

Gebruikershandleiding

Neural Vision 2.0

Data Visualisatie en Data Analyse met Neurale Netwerken

Jeroen Sprenger
Rob van Stee
Michiel van Wezel

mei 1999

CWI BIBLIOTHEEK



3 0054 00117 6909

Inhoudsopgave

1	Inleiding	4
1.1	Installatie & systeemeisen	5
1.1.1	Registry instellingen	6
1.2	Globale structuur en werking van de programmatuur	6
1.3	Structuur van de gebruikersinterface	13
2	Korte inleiding artificiële neurale netwerken	15
2.1	Enige biologie van natuurlijke neurale netwerken	15
2.2	Enige 'biologie' van artificiële neurale netwerken	16
2.3	Typen artificiële neurale netwerken	18
2.3.1	McCulloch-Pitts netwerken	19
2.3.2	De Hebb leerregel	19
2.3.3	De Perceptron	20
2.3.4	Feed Forward Netwerken	21
2.3.5	De Backpropagation leerregel	21
3	Tabblad 1: Dataset	23
3.1	Een bestand openen	23
3.2	Het Bestandsmenu	24
3.3	Bewerken	25
3.4	Ontbrekende waarden	25
4	Tabblad 2: het trainen van de neurale netwerken	28
4.1	Sanger subtabblad	28
4.2	ICA subtabblad	30
4.3	Fscnn netwerk	30
4.4	Labels	31

5	Tabblad 3: de polygon-plot	34
5.1	Het projectieveld	34
5.1.1	Kiezen doelen in projectie	35
5.2	De polygonplot	36
5.3	Tijdsanalyse	36
6	Tabblad 4: Statistieken	41
6.1	Statistieken per variabele	41
6.2	Statistieken per cluster	42
6.3	Het histogram	43
7	Uitgewerkte voorbeelden	44
7.1	Inleiding	44
7.2	Beschrijving van de Boston dataset	45
7.3	Analyse van de Boston dataset	46
7.4	Beschrijving van de FRAME3 dataset	54
7.5	Analyse van de FRAME3 dataset	55
8	Gebruikte technieken	61
8.1	Het Sanger neurale netwerk	61
8.1.1	Principale componenten analyse	61
8.1.2	Het Sanger neurale netwerk	63
8.2	Wiskundige beschrijving van PCA en het Sanger netwerk . . .	65
8.3	Het ICA neurale netwerk	68
8.3.1	Algemene uitleg en voorbeelden	68
8.3.2	Wiskundige uitleg	71
8.4	Het frequency-sensitive competitive learning neurale netwerk	75
8.5	Variabelenselectie met behulp van variantie en entropy	78
8.5.1	Het uitrekenen van de variantie	78
8.5.2	Het uitrekenen van de entropy	78
8.6	De eenvoudige statistieken	79
	Bibliografie	81
A	Lijst met trefwoorden	83
B	Beschrijving van het ASCII dataset file format (*.ndf files)	86

C	Artificiële Neurale Netwerken	88
C.1	Inleiding	88
C.2	Leren en veranderen	90
C.3	De werking van Artificiële Neurale Netwerken	93
C.4	De praktijk van het in kaart brengen van veranderingsprocessen	98
C.5	Beloften en zwakten van neurale netwerken	103
C.6	Bibliografie	106

Hoofdstuk 1

Inleiding

In dit rapport wordt het software-pakket 'Neural Vision 2.0' beschreven dat is ontwikkeld door het Centrum voor Wiskunde en Informatica te Amsterdam. Het pakket 'Neural Vision 2.0' is een uitbreiding op het in 1996/1997 ontwikkelde 'Neural Vision 1.0'.

Neural Vision 2.0 is bedoeld voor visualisatie en clustering van hoog-dimensionale data. Bij het ontwikkelen van Neural Vision 2.0 is het de doelstelling geweest om een laagdrempelig instrument te creëren voor het analyseren en inzichtelijk maken van multidimensionale datasets, ofwel het uitvoeren van verkennende data-analyses. Hiertoe zijn verschillende algoritmen geïmplementeerd om data uit een hoog-dimensionale ruimte te projecteren op een twee dimensionaal vlak. Na deze projectie zullen eventueel aanwezige structuren in de data vaak duidelijk zichtbaar worden.

De user interface van Neural Vision 2.0 bevat gebruikersvriendelijke voorzieningen als een spreadsheet voor data management, automatische substitutie van ontbrekende waarden en een muis geïntegreerde aansturing van de polygon-plot analyse.

De structuur van dit rapport is als volgt. De volgende secties in dit hoofdstuk beschrijven hoe het pakket geïnstalleerd dient te worden (sectie 1.1) en wat de globale structuur en de gebruiksmogelijkheden van de programmatuur zijn (secties 1.2 en 1.3). Hoofdstukken 3 t/m 6 beschrijven de afzonderlijke modules waaruit het pakket is opgebouwd. In hoofdstuk 7 worden ter illustratie enkele voorbeeld analyses uitgevoerd met Neural Vision 2.0. Hoofdstuk 8 beschrijft de neurale netwerk technieken die in het pakket zijn geïmplementeerd. Dit hoofdstuk is slechts bedoeld voor de geïnteresseerde lezer en wordt gevolgd door een literatuuropgave. Voor een korte inleiding

in het verschijnsel neurale netwerken wordt verwezen naar hoofdstuk 2.

Een lijst met trefwoorden kan men vinden in appendix A. Appendix (B) beschrijft het gedefiniëerde file format voor ASCII-databestanden. Tenslotte wordt in appendix (C), geschreven door Walter Baets en Leon Brunenberg, achtergrondinformatie verstrekt over de filosofie inzake het gebruik van neurale netwerken voor het in kaart brengen van veranderingsprocessen en het meten van leren, een mogelijk toepassingsgebied van Neural Vision 2.0.

Neural Vision 2.0 is ontwikkeld met Borland C++ Builder 3.0. Borland componenten die met Neural Vision 2.0 meegeleverd worden zijn intellectueel eigendom van Borland en zijn auteursrechtelijk beschermd. Zij mogen alleen gebruikt worden door het Neural Vision programma. Zij mogen niet los gebruikt worden of in een applicatie waar men geen licentie voor aangeschaft heeft. Hetzelfde geldt voor de Builder component 'topgrid' die gebruikt is voor het spreadsheet-deel van Neural Vision 2.0 en intellectueel eigendom is van Top Support. Ook deze component mag niet los van Neural Vision 2.0 of in andere applicaties zonder licentie worden gebruikt. Het deel van Neural Vision dat door het CWI ontworpen is, is intellectueel eigendom van het Ministerie van Verkeer en Waterstaat, Dienst Weg- en Waterbouwkunde, gevestigd te Delft.

1.1 Installatie & systeemeisen

Het installatiepakket bestaat uit vier diskettes. Op de eerste diskette staat setup.exe, waarmee InstallShield gestart kan worden door het vanuit de windows verkenner dubbel aan te klikken. Dit is een standaard installatieprogramma, dat via enkele Engelstalige dialoogboxen de installatie in goede banen leidt. Als het gestart wordt, verschijnt er een dialoogbox die aangeeft dat de installatie gestart wordt. Er loopt een getal van 0 tot 100%. Daarna moet ingevuld worden in welke directory het programma geïnstalleerd moet worden. Hier kan goed de suggestie die het programma zelf geeft gebruikt worden. Hierna worden een voor een alle vier de diskettes ingelezen. Tenslotte verschijnt er een afsluitende dialoogbox, waarin de gebruiker kan aangeven of hij direct het programma wil opstarten.

De directory waarin Neural Vision 2 is geïnstalleerd, bevat drie subdirectories: datasets, examples en template. In de directory examples staan enkele voorbeeldbestanden in verschillende formaten, die in Neural Vision 2 ingeladen kunnen worden. De directory datasets is aan het begin leeg en is

bedoeld voor bestanden die de gebruiker zelf aanmaakt. De directory template bevat een database, die door Neural Vision 2 intern gebruikt wordt. Neural Vision 2 draait op PC's waar Windows 95 of hoger geïnstalleerd is.

1.1.1 Registry instellingen

Neural Vision maakt onder `HKEY_LOCAL_MACHINE\Software` een sleutel `Borland` aan. Hieronder komen twee sleutels, `BLW32` en `Database Engine`. Onder deze twee sleutels komen alle instellingen van de Borland Database Engine terecht. De internationale instellingen komen onder `BLW32` te staan. Onder `Database Engine` staat `Settings`, en daaronder `DRIVERS` en `SYSTEM`. Onder `DRIVERS` staan de instellingen voor `DBASE`, `FOXPRO`, `MSACCESS` en `PARADOX`, terwijl onder `SYSTEM` onder andere de instellingen voor de weergave van datums, getallen en de tijd staan. Tenslotte staan onder `SYSTEM\INIT` nog wat algemene instellingen.

Er is nog een andere plek waar instellingen verschijnen. Namelijk onder `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\SharedDLLs` komen verwijzingen naar allereerst alle DLLs uit de directory `C:\Program Files\Common Files\Borland\BDE` en bovendien de bestanden uit `C:\Windows\System` die Neural Vision 2.0 daar gezet heeft: `bde-admin.cpl`, `borlndmm.dll`, `cp3240mt.dll`, `ctl3d.dll`, `ctl3d32.dll` en `vcl35.dll`.

1.2 Globale structuur en werking van de programmatuur

Zoals in de inleiding reeds is gemeld, is Neural Vision 2.0 bedoeld voor de verkennende analyse van multidimensionale datasets. Bij het ontwerp van Neural Vision 2.0 is de laagdrempeligheid van het gebruik telkens een uitgangspunt geweest: Neural Vision 2.0 poogt een intuïtief inzicht te geven in de verborgen structuren die in een dataset aanwezig zijn. In deze sectie wordt in grove lijnen uitgelegd hoe Neural Vision 2.0 gebruikt dient te worden. Allereerst wordt echter duidelijk gemaakt hoe Neural Vision 2.0 omgaat met verschillende typen data.

Datatype	NV2 benaming	Doet mee in			
		NN's	labelling	Miss. val. repl.	Statist.
Kwalitatief nominaal	tekst	-	X	-	-
Kwalitatief ordinaal	ordinaal numeriek	X	X	X	deels
Kwantitatief discreet	numeriek	X	X	X	X
Kwantitatief continu	numeriek	X	X	X	X

Tabel 1.1: Rol van verschillende variabelentypen in Neural Vision 2.0

Typen data en behandeling in Neural Vision 2.0

Zoals de lezer wellicht bekend is bestaan er verschillende typen data. Grofweg kan men data die men in de praktijk tegenkomt onderverdelen in twee typen: kwantitatieve en kwalitatieve data. Beide typen variabelen zijn weer onder te verdelen in sub-klassen. Kwantitatieve variabelen kan men verder onderverdelen in discrete en continue variabelen. Kwalitatieve variabelen kan men verder onderverdelen in ordinale en nominale variabelen.

Voorbeelden van kwantitatieve variabelen zijn: lengte van een persoon, aantal blaadjes aan een boom, bandenspanning etc. Deze kwantitatieve variabelen kan men objectief meten op een interval-schaal. Op deze interval-schaal geldt de metriek der natuurlijke of continue getallen. Kwantitatieve variabelen kan men onderverdelen in discrete en continue variabelen. Om iedere variabele even zwaar te laten meetellen in de analyse worden de waarden zodanig herschaald dat het gemiddelde 0 en het bereik 1 is.

Het tweede type data is kwalitatieve data, ook wel categorische data genoemd. Voorbeelden van kwalitatieve variabelen zijn kleur, vorm en smaak. Een kwalitatieve variabele zegt iets over een eigenschap van een object, maar een kwalitatieve variabele kan niet als getal gemeten worden. Kwalitatieve variabelen kan men verder onderverdelen in ordinale en nominale variabelen. Bij ordinale variabelen is er sprake van een impliciete ordening (bijvoorbeeld smaak \in {erg vies, vies, gemiddeld, lekker, erg lekker}). Bij nominale variabelen is dit niet het geval (bijvoorbeeld vorm \in { rond, vierkant, driehoekig}).

De wijze waarop Neural Vision 2.0 de diverse variabelentypen behandelt is weergegeven in tabel 1.1. In deze tabel zijn de namen die in Neural Vision 2.0 gebruikt worden om de verschillende variabelentypen te onderscheiden weergegeven en is aangegeven welk type variabele in welke programma-onderdeel gebruikt wordt.

Het variabelentype 'ordinaal numeriek' behoeft wat extra toelichting. Zoals de naam al weergeeft betreft dit variabelentype ordinale variabelen, maar

deze dienen wel numeriek te zijn, dus te worden gerepresenteerd door getallen! Niet numerieke ordinale variabelen beschouwt Neural Vision 2.0 altijd als tekst-variabelen (nominaal). De 'ordinaal numerieke' variabelen worden standaard meegenomen als invoervariabelen voor de neurale netwerken. Ook kan men op een 'ordinaal numerieke' variabele 'missing value replacement' uitvoeren. Beide bewerkingen mag men echter alleen uitvoeren als de metriek die geldt op de getallen die gebruikt worden om de waarden voor de ordinale variabele te coderen niet erg verschillend is van de standaard-metriek der natuurlijke getallen. Is men hier niet van overtuigd dan verdient het aanbeveling om een 'ordinaal numerieke' variabele te behandelen als een tekst-variabele!

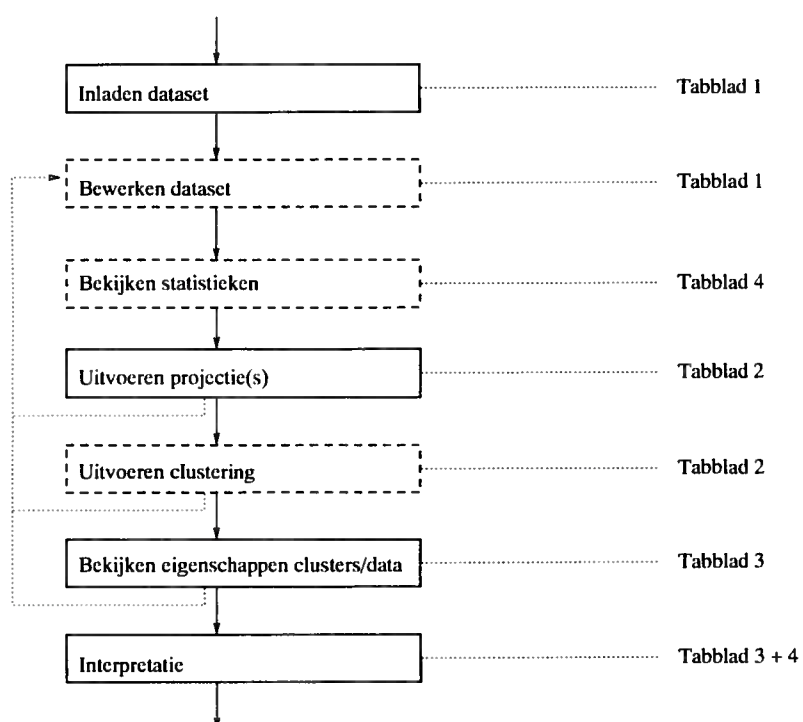
Nog een waarschuwing is op zijn plaats! Wanneer een ingelezen dataset een nominale variabele bevat die met door getallen gecodeerd is mag deze variabele **nooit** worden meegenomen in een projectie! Een dergelijke variabele dient altijd te worden getypeerd als een tekst-variabele, ondanks het feit dat ze door getallen gerepresenteerd wordt.

Gebruik van de programmatuur

In Neural Vision 2.0 zijn twee neurale netwerken voor de projectie van multi-dimensionale data geïmplementeerd: het Sanger Netwerk (bekend uit Neural Vision 1.0) en het ICA-netwerk. (ICA staat voor Independent Component Analysis.) De werking van deze netwerken wordt in detail besproken in hoofdstuk 8. Met behulp van deze netwerken kan een hoog dimensionale dataset op een optimale manier op twee dimensies worden afgebeeld en vervolgens worden bekeken. Deze eerste fase van de analyse is bedoeld voor het ontdekken van structuur in de dataset.

Ook is er een neuraal netwerk geïmplementeerd voor het creëren van een clustering. Als in de data-projectiefase duidelijk clusters te onderscheiden zijn, zullen deze clusters in de clusteringfase worden teruggevonden. Men kan de eigenschappen van de clusters vervolgens nader bekijken met een speciaal type plot: de 'polygonplot'. Hier wordt later nog op in gegaan.

Het programma dient gebruikt te worden als geïllustreerd in figuur 1.1. In deze figuur is een workflow schema weergegeven. De onderdelen die hierin omlind zijn met stippellijnen kunnen als optioneel beschouwd worden. De gebruikersinterface van NV2 is onderverdeeld in tabbladen. Bij ieder onderdeel is aangegeven welk tabblad voor dit onderdeel gebruikt dient te worden. De verschillende onderdelen uit dit workflow schema worden hieronder toegelicht.



Figuur 1.1: Workflow schema voor Neural Vision 2.0

Inlezen dataset : In deze fase wordt een databestand ter analyse ingeladen in Neural Vision 2.0. Men kan verschillende gangbare typen databestanden inladen. De ingeladen dataset wordt getoond op een spreadsheet.

Bewerken dataset : In deze fase kan de dataset bewerkt worden. Zo kan men in deze fase randwaarden in de dataset verwijderen/aanpassen en kunnen ontbrekende waarden worden gesubstitueerd. Verder kunnen met behulp van de spreadsheet een aantal operaties op de rijen en kolommen van de dataset uitgevoerd worden.

Bekijken statistieken : In deze fase kan men bepaalde eigenschappen van de dataset (statistieken) bekijken die veel beknopte informatie geven over de dataset.

Uitvoeren projecties : In deze fase worden de numerieke en ordinale variabelen uit de dataset geprojecteerd op het twee dimensionale vlak. Vaak zal in deze twee dimensionale projectie veel structuur te zien zijn. Nabijheidsrelaties tussen de samples uit de dataset zijn in een dergelijke afbeelding vaak duidelijk zichtbaar. De 'scree plot' geeft een indruk van de betrouwbaarheid van de projectie. De projectie is betrouwbaar als de scree plot links sterk dalend is. Op deze projectiefase wordt hieronder nog teruggekomen.

Uitvoeren clustering : In deze fase verdeelt men de data in de dataset in een aantal clusters. Het bepalen van dit aantal gebeurt aan de hand van de data-projectie.

Bekijken eigenschappen clusters/data : De eigenschappen van alle gevonden clusters worden in deze fase gevisualiseerd middels het eerder genoemde polygon-plot. Ook kan men voor ieder cluster een aantal statistieken bekijken. Voorts kan men de punten in de dataprojectie labellen (van kleuren voorzien) om de inhoud van de clusters te onderzoeken.

Interpretatie : In deze fase dienen de uitkomsten van de analyse te worden geïnterpreteerd. Deze fase hangt nauw samen met de vorige fasen.

Wenken voor de projectiefase

Als de projectiefase uitgevoerd is kan vaak een interpretatie aan de assen van de projectie gegeven worden. Dit doet men door de punten in de data-projectie te labelen op de diverse numerieke variabelen. Als nu het verloop (gecodeerd door het kleurverloop) van deze numerieke variabele samenvalt met een van de assen dan is er sprake van een belangrijke variabele die een grote bijdrage levert aan de as in kwestie. Door te kijken welke variabelen er een grote rol spelen in een dimensie kan men vaak een interpretatie aan de dimensie geven. In hoofdstuk 7 wordt hier een voorbeeld van gegeven.

Soms is in datasets een zogenaamde ‘afhankelijke’ variabele aanwezig. In dit geval is men geïnteresseerd in een verband tussen de ‘onafhankelijke’ variabelen in de dataset en de ‘afhankelijke’ variabele. Dergelijke datasets zijn vaak afkomstig van regressie- of classificatievraagstellingen. Alhoewel Neural Vision primair bedoeld is voor verkennende ongesuperviseerde data analyse, kunnen verbanden tussen onafhankelijke en afhankelijke variabelen toch worden onderzocht en gevisualiseerd.

Bij een numerieke afhankelijke variabele dient men hiervoor een data-projectie te genereren waarbij slechts de onafhankelijke variabelen geprojecteerd worden. Nadat de projectie gemaakt is dienen de assen te worden geïnterpreteerd op de hierboven beschreven manier. Vervolgens kan men onderzoeken of er een sterk verband bestaat tussen de afhankelijke variabele en de onafhankelijke variabelen door het labelen op de afhankelijke variabele. Als hieruit blijkt dat het verloop van de afhankelijke variabele samenvalt met het verloop van een of beide dimensies uit de projectie, dan is er een correlatie tussen deze dimensie(s) en de afhankelijke variabele. Dit duidt erop dat de dataset tenminste enigszins de juiste informatie bevat voor het maken van een voorspellingsmodel. Wanneer verschillende waarden voor de afhankelijke variabelen kris kras over de plot zijn verdeeld duidt dit op een zwak (of afwezig) verband. Ook hiervan zullen we in hoofdstuk 7 een voorbeeld zien.

Een praktisch probleem bij het labelen op een numerieke afhankelijke variabele is dat het niet mogelijk is te labelen op een numerieke variabele die niet in de projectie wordt meegenomen. De gebruiker kan dit probleem omzeilen door eigenhandig de numerieke afhankelijke variabele onder te verdelen in een aantal intervallen (bijv. door afronding of het aanmaken van klassen {zeer laag, laag, gemiddeld, hoog, zeer hoog}) m.b.v. een programma als Excel van Microsoft. Deze nieuwe variabele kan vervolgens als tekst-variabele

meegenomen worden in de analyse en gebruikt worden voor labelling.

Heeft men een nominale afhankelijke variabele dan kan het verband ook worden onderzocht. Dit doet men door het maken van een projectie en wederom het labelen met de afhankelijke variabele. Als de waarde van de afhankelijke variabele (dat is een klasse indeling bij nominale variabelen) erg afhankelijk is van de gevonden dimensies, dan zullen de datapunten die tot dezelfde klassen behoren in de plot bij elkaar clusteren. Ook hiervan zullen we in hoofdstuk 7 een voorbeeld zien.

Men kan zich afvragen of het niet verstandiger is om de afhankelijke variabele in de projectie mee te nemen wanneer deze numeriek is. Het *wel* meenemen van de afhankelijke variabele levert het gevaar op dat deze overgerepresenteerd gaat worden in een van de twee gevonden dimensies. Dit zou leiden tot het samenvallen van de kleuring op de afhankelijke variabele en de dimensie waarin deze sterk is gerepresenteerd, veroorzaakt door het feit dat de gevonden dimensie niets anders is dan de afhankelijke variabele. De gebruiker zou ten onrechte de conclusie trekken dat de afhankelijke variabele dus te modelleren valt met behulp van de onafhankelijke variabelen.

Anderzijds is het zo dat de afhankelijke variabele verantwoordelijk is voor een deel van de totale variantie in de dataset. Mochten er in de onafhankelijke variabelen factoren (dit zijn potentiële dimensies die door Neural Vision worden gevonden) zitten die geschikt zijn voor het modelleren van de afhankelijke variabele en factoren die hiervoor niet geschikt zijn, dan is de kans dat de geschikte factoren in een van de twee gevonden dimensies terecht komen kleiner omdat de 'niet geschikte' factoren een relatief groter deel van de variantie in de totale dataset voor hun rekening zullen nemen. Echter vanwege het in de vorige alinea genoemde bezwaar is het te prefereren de afhankelijke variabele niet mee te nemen. Neural Vision leent zich dan ook beter voor het visualiseren van hoog-dimensionale datasets dan voor het zoeken van verbanden tussen variabelen.

Keerzijde van het niet meenemen van de afhankelijke variabele is dat deze verantwoordelijk is voor een deel van de totale variantie in de dataset. Mochten er in de onafhankelijke variabelen factoren (dit zijn potentiële dimensies die door Neural Vision worden gevonden) zitten die geschikt zijn voor het modelleren van de afhankelijke variabele en factoren die hiervoor niet geschikt zijn, dan is de kans dat de geschikte factoren in een van de twee gevonden dimensies terecht komen kleiner omdat de 'niet geschikte' factoren een relatief groter deel van de variantie in de totale dataset voor hun rekening zullen nemen.

1.3 Structuur van de gebruikersinterface

The screenshot shows the 'Dataset' tab of the Neural Vision 2.0 software. The window title is 'Neural Vision 2.0 [C:\Program Files\Borland\VCBuilder3\Projects\Dataset\shpmd02.dbf]'. The menu bar includes 'Bestand', 'Bewerken', 'Netwerken', 'Labels', 'Tijdanalyse', and 'Info'. The main area displays a table with the following columns: 'INVOLOW', 'FSCNN', 'DIRECTIE', 'DEKLAAG', 'LEEFTIJD', 'OVERLAGING', 'ASFODKTE', 'FUNDMAT', 'FUNDKITE', 'VWV_JR', and 'ESAL_JR'. The table contains 20 rows of data.

INVOLOW	FSCNN	DIRECTIE	DEKLAAG	LEEFTIJD	OVERLAGING	ASFODKTE	FUNDMAT	FUNDKITE	VWV_JR	ESAL_JR
1	4	10	1	233	0	0	371,250	44500		
2	4	13	1	215	1	238	139645	119396		
3	4	9	2	202	2	213	276518	315230		
4	4	14	2	171	3	158	451500	514710		
3	3	8	1	269	0	0	166000	167400		
4	3	13	2	228	4	220	294865	326146		
2	4	12	1	244	5	192	668780	762408		
1	4	13	2	154	6	210	456388	519028		
8	4	10	1	370	6	131	487438	567073		
1	4	10	1	244	7	125	223423	191036		
3	4	8	1	285	7	122	877183	1083296		
9	4	14	1	291	0	0	277240	316054		
2	4	12	2	150	8	268	629440	717562		
4	4	18	2	231	0	0	622400	709536		
2	4	14	1	332	0	0	288225	325577		
3	4	11	1	210	6	423	562500	641250		
2	4	5	1	284	9	183	647508	738159		
3	4	10	1	197	6	147	643823	732963		
4	4	12	1	234	7	122	1134360	1327201		
5	3	10	1	322	7	110	1400000	1638000		
5	3	11	1	235	2	172	819000	1011485		
5	4	13	1	289	7	127	1320000	1544400		
5	4	13	1	198	6	150	384750	1082966		
6	4	4	1	125	10	360	1618750	1368156		
6	3	12	2	750	0	0	1618750	1399156		
6	3	7	1	323	7	48	1750000	2161250		
6	3	8	1	456	7	110	1222500	1522130		
6	3	10	2	333	0	0	742195	916487		
7	3	9	1	276	7	100	720000	883200		
7	4	10	1	269	7	114	1046250	1292119		

Figuur 1.2: Overzicht van de userinterface na het openen van een dataset.

De structuur van de programmatuur reflecteert de structuur van het in figuur 1.1 weergegeven workflow schema. Dit is te zien in figuur 1.2. In deze figuur is het beginscherm te zien dat de gebruiker bij het opstarten van Neural Vision 2.0 op zijn scherm krijgt, na het openen van een dataset. Dit scherm bestaat uit een aantal tabbladen en een menu-structuur. De respectievelijke tabbladen zijn

Dataset : dit tabblad is bedoeld voor het inladen en het bewerken van een dataset.

Netwerken : dit tabblad is bedoeld voor het creëren van een data-projectie en een clustering door middel van het trainen van de verschillende neurale netwerken.

Polygonplot : dit tabblad is bedoeld voor de polygon-plot analyse: de bestudering en grafische weergave van de gevonden clusters en aparte datapunten.

Statistieken : op dit tabblad wordt een aantal eigenschappen van de dataset gerepresenteerd, zoals histogrammen van de variabelen, gemiddelden, entropie en standaard deviatie van de verschillende variabelen, etcetera.

In de volgende hoofdstukken zal Neural Vision 2.0 gedetailleerd worden besproken. Hierbij wordt tabblad voor tabblad behandeld. Allereerst volgt er echter een inleiding over kunstmatige neurale netwerken.

Hoofdstuk 2

Korte inleiding artificiële neurale netwerken

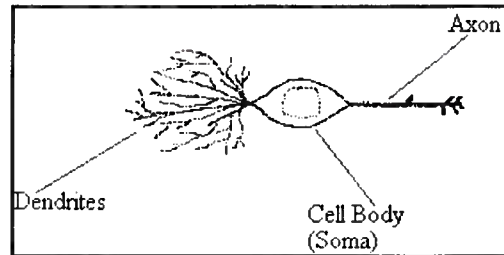
De bedoeling van dit hoofdstuk is om een inleiding te geven in de theorie van artificiële neurale netwerken. De basiselementen en grondslagen van artificiële neurale netwerken worden besproken. Dit stuk is een vrije, bewerkte vertaling van <http://home.clara.net/ml1/ai/nn/>.

Allereerst wordt een beknopte inleiding gegeven op biologische neurale netwerken (hersenen). Vervolgens wordt duidelijk gemaakt hoe biologische neurale netwerken zijn geschematiseerd tot artificiële neurale netwerken. De werking van verschillende typen artificiële neurale netwerken wordt uitgelegd.

Zoals gezegd geeft dit hoofdstuk een beknopte en algemene inleiding op artificiële neurale netwerken. Technische details zijn hierbij vermeden. De technieken die in Neural Vision zijn geïmplementeerd worden hier dan ook niet nauwgezet besproken. De lezer die hierin is geïnteresseerd wordt verwezen naar hoofdstuk 8.

2.1 Enige biologie van natuurlijke neurale netwerken

Het menselijk brein is een van de meest complexe, zonet het meest complexe, orgaan in het menselijk lichaam. Het brein bestaat uit een verbluffende hoeveelheid (ca. 10.000.000.000.000) individuele cellen. Deze cellen worden zenuwcellen of neuronen genoemd. Elk neuron is verbonden met een aantal (variërend van tienduizenden tot minder dan honderd) andere neuronen. Alle



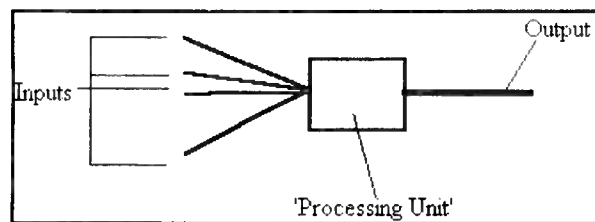
Figuur 2.1: Een natuurlijk neuron.

cellen in het brein hebben overeenkomsten met elkaar. Allemaal hebben ze nl. 'dendrieten': verbindingen met andere cellen die de invoer van het neuron vormen. Ook heeft iedere cel een 'axon': een smal buisje dat vele vertakkingen kan hebben en dat de uitvoer van de cel vormt. Dit 'axon' is verbonden met de dendrieten van andere cellen. Een biologisch neuron kan er dus uitzien zoals is weergegeven in de onderstaande figuur:

Een neuron geeft een signaal af (dit wordt ook wel vuren genoemd) wanneer voldoende cellen die met de dendrieten van het neuron verbonden zijn ook vuren. Het resulterende signaal wordt dan over de axon van de cel gestuurd. Hierdoor kunnen andere neuronen getriggerd worden om te vuren. Door het karakter van de axonen en dendrieten zodanig aan te passen dat zij signalen versterken of afzwakken is het brein in staat om zijn gedrag aan te passen en daardoor te leren. Dit is het gesimplificeerde gedrag van een biologisch neuron. Het feit dat de enorme hoeveelheid cellen in het brein zich op deze manier afhankelijk van elkaar gedragen leidt tot het emergent complex gedrag van het brein als geheel. Dit emergente gedrag van het brein is verantwoordelijk voor de gedachten en beelden in het menselijk brein.

2.2 Enige 'biologie' van artificiële neurale netwerken

Artificiële neurale netwerken (ANN's) zijn het computationele equivalent van de huidige biologische modellen voor de werking en samenhang van neuronen. Omdat een groot deel van de werking van de hersenen nog met mysteries is omgeven, is dit computationele equivalent nog geenszins gelijkwaardig aan het menselijk brein: in tegenstelling tot dit laatste complexe systeem zijn



Figuur 2.2: Een kunstmatig neuron.

ANN's slechts in staat relatief eenvoudige taken uit te voeren.

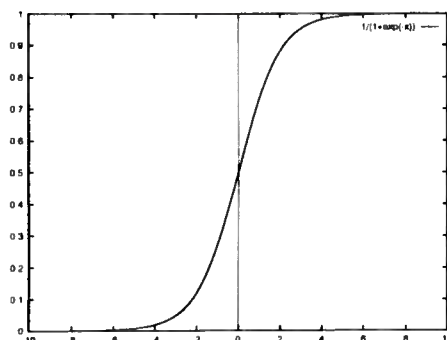
ANN's bestaan uit vele afzonderlijke 'units', analoog aan de neuronen waaruit het brein bestaat. Deze units zijn onderling verbonden door middel van 'verbindingen' of 'links' waarmee een 'gewicht' is geassocieerd. Deze verbindingen zijn de equivalenten van axonen en dendrieten in natuurlijke neuronen. Via een verbinding kan een unit een signaal van een andere unit ontvangen of naar een andere unit versturen. Een typisch kunstmatig neuron kan er dus als volgt uitzien:

In ANN's neemt een unit zijn gewogen invoer (dat is de som van de afzonderlijke invoeren vermenigvuldigd met de gewichten van de respectievelijke verbindingen) en past hier een eenvoudige mathematische bewerking (de zgn. transfer function) op toe. Op de mogelijkheden voor deze transfer functie wordt hieronder nog teruggekomen. Het resultaat van deze transfer functie wordt over de uitvoerverbinding gestuurd, en kan door andere units worden opgepikt als een deel van hun invoerpatroon.

Er zijn verschillende keuzes mogelijk voor de transferfunctie. De eerste mogelijkheid is de zgn. step function. Deze functie is 0 voor alle invoerwaarden beneden een bepaalde drempelwaarde, 1 voor alle waarden daarboven. Een andere mogelijkheid is de identiteitsfunctie. Bij deze functie is de uitvoerwaarde identiek aan de invoerwaarde. De laatste veelgebruikte transferfunctie is de sigmoïde functie of squashing functie. Deze functie heeft het bereik $[0 : 1]$ en heeft een *S*-achtige vorm. Een voorbeeld van een sigmoïde functie is gegeven in figuur 2.3.

Net als biologische neurale netwerken vindt het leren in artificiële neurale netwerken plaats door het aanpassen van het karakter van de verbindingen tussen de units, dus door het aanpassen van de gewichten.

De beschrijving hierboven is uiteraard een grove simplificatie van de werking van een neurale netwerk in het algemeen. Om totale werking van een



Figuur 2.3: Sigmoide functie gebruikt in meerlaags feedforward neuraal netwerk.

neuraal netwerk te doorgronden moet men uiteraard het gedrag van de units als een collectief bestuderen. Dit is een zeer complex vakgebied dat volop in ontwikkeling is. Hopelijk heeft deze korte inleiding de lezer echter enig idee gegeven over de werking van natuurlijke en artificiële neuronen.

Nu de basisbegrippen zijn geïntroduceerd, kan verder worden gegaan met het geven van beschrijvingen van de belangrijkste typen neurale netwerken. We behandelen hier de typen die van historisch of praktisch belang zijn.

2.3 Typen artificiële neurale netwerken

Zoals reeds is gemeld zijn artificiële neurale netwerken een computationele representatie van hoe een natuurlijk brein gedacht wordt te werken. Er zijn vele wetenschappers, waaronder psychologen, fysiologen, informatici etc., die gepoogd hebben om een computationeel model op te stellen voor het menselijk brein. Sommigen van hen hebben systemen ontwikkeld die een (meestal gering) deel van de hersenen nabootsen. Anderen hebben op neuron-niveau naar het probleem gekeken, en hebben algoritmen ontwikkeld waarmee het leergedrag van biologische neuronen nagebootst kan worden.

Dit deel van dit hoofdstuk is bedoeld om de lezer een elementair inzicht te geven in enkele bekende netwerken en algoritmen. De netwerken en leerregels die aan bod zullen komen zijn:

- McCulloch-Pitts netwerken

- Hebb leerregel
- Perceptrons
- Feed forward netwerken (multilayer perceptron netwerken)
- Backpropagation leerregel

2.3.1 McCulloch-Pitts netwerken

McCulloch-Pitts neurale netwerken worden over het algemeen gezien als de eerste modellen voor biologische neurale netwerken die in electronika geïmplementeerd werden. Omstreeks 1943 (toen het artikel [MP43] gepubliceerd werd) hadden McCulloch en Pitts ontdekt dat door het onderling verbinden van verschillende 'units' een grote computationele kracht kon worden verwezenlijkt. In deze periode waren de termen 'kunstmatige intelligentie' (artificial intelligence) en 'neuraal netwerk' nog niet geboren.

Dit netwerk gebruikte een eenvoudige stap-functie als transferfunctie. Zoals al uitgelegd is, zegt deze functie dat als de totale invoer voor een unit boven een bepaalde waarde uit komt, dat de unit dan 'aan' (1) is, terwijl deze anders 'uit' (0) is.

Dergelijke netwerken worden zelfs vandaag de dag, 55 jaar na introductie nog gebruikt; Meestal is dit in logische circuits, wat ook de originele toepassing was.

2.3.2 De Hebb leerregel

De Hebb leerregel is de eerste leerregel die ooit ontworpen is voor neurale netwerken. De regel is vernoemd naar Hebb, die hem in 1949 publiceerde in [Heb49]. De regel wordt gebruikt om de gewichten van een neuraal netwerk zodanig aan te passen dat de daadwerkelijke uitvoer van een neuraal netwerk de verwachte uitvoer representeert. Deze leerregel wordt ook nu nog in vele typen neurale netwerken toegepast.

Omdat dit een van de eerste leerregels was, is het ook een van de eenvoudigste leerregels om toe te passen. De regel schrijft voor dat als twee units, A en B, dezelfde activatie hebben (beide 'aan' of 'uit'), het gewicht van de verbinding tussen deze twee units dient te worden verhoogd. Als nu de uitvoer van unit A, die bijdraagt aan de invoer voor unit B, vaak gelijk is aan de uitvoer van unit B, wordt het gewicht tussen A en B verhoogd. Daarmee

wordt ook de kans verhoogd dat dit gedrag zich vaker voordoet, zodat unit A unit B gaat imiteren. Een hoog gewicht van een verbinding tussen twee units duidt dus op een hoge correlatie van het gedrag van deze twee units. Als transfer functie hebben deze netwerken vaak een identiteitsfunctie of een stapfunctie.

2.3.3 De Perceptron

De Perceptron [Ros58] is omstreeks dezelfde tijd ontwikkeld als de McCulloch-Pitts neuronen, door Frank Rosenblatt verbonden aan Cornell Aeronautical Laboratories.

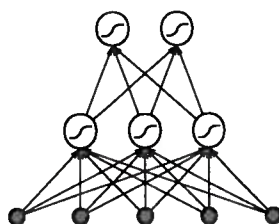
De perceptron was een van de eerste netwerken die specifiek gebaseerd was op een deel van het menselijk lichaam, te weten het netvlies van het oog. De invoer van het netwerk werd gevormd door een matrix van lichtgevoelige sensoren. Deze sensoren waren gedeeltelijk verbonden door middel van gewogen verbindingen met een laag van associatie-units (A-units genoemd). De A-units waren op hun beurt verbonden met 'response units' (R-units genoemd), die de uitvoer van het systeem vormden.

De bedoeling van het netwerk was om een invoerbeeld te classificeren als zijnde een van een aantal bekende klassen. Iedere klasse werd hierbij gerepresenteerd door middel van een van de 'R-units'.

De leerregel die men gebruikte voor dit netwerk was de 'perceptron leerregel', een van de eenvoudigste leerregels in de ANN gemeenschap. De leerregel komt op het volgende neer:

- Als de uitvoer van het netwerk gelijk is aan de gewenste uitvoer, doe niets.
- Als een uitvoer ongewenst aan is (1), verlaag dan alle actieve gewichten met één. Een actief gewicht is een gewicht die een unit verlaat die 'aan' is.
- Als een uitvoer ongewenst uit is (0), verhoog dan alle actieve gewichten met één.

Dit is een voorbeeld van een 'supervised' leerregel, waarbij het netwerk kan leren omdat het feedback krijgt over zijn prestaties. Naast 'supervised' leerregels bestaan er ook 'unsupervised' leerregels waarbij we niet weten wat de correcte uitvoer dient te zijn en het ANN dus zelf-organiserend zal moeten opereren. Dergelijke typen netwerken zijn in NeuralVision 2.0 gebruikt.



Figuur 2.4: Plaatje van een feedforward neuraal netwerk.

2.3.4 Feed Forward Netwerken

Het meerlaags feedforward neuraal netwerk is een extensie van de perceptron die al is besproken. Het heet een meerlaags ANN omdat er één of meer verborgen lagen met knopen in zitten. Deze lagen heten verborgen omdat de units in deze lagen niet direct door de buitenwereld beïnvloed worden noch direct de buitenwereld beïnvloeden. De units in de verborgen lagen ontvangen hun invoer van andere units in het ANN en sturen hun uitvoer ook weer naar andere units in het ANN. Figuur 2.4 laat een typisch feedforward ANN zien.

Het gebruik van meerlaags feedforward ANN's is niet beperkt tot classificatie toepassingen, zoals bij de originele perceptron. Meerlaags feedforward ANN's zijn breder toepasbaar en worden veel gebruikt in o.a. regressie, proces-controle en classificatie.

Zoals gemeld bestaan de meerlaags netwerken uit verschillende lagen. In het algemeen is het zo dat de units in de invoerlaag alleen verbonden zijn met de units in de eerste verborgen laag. De units in deze laag zijn verbonden met de units in de tweede verborgen laag, etc. tot aan de uitvoerlaag. Het netwerk is dus echt in lagen georganiseerd en de units in een bepaalde laag genereren de invoer voor de units in de volgende laag. .

2.3.5 De Backpropagation leerregel

Meerlaags feedforward ANN's worden normaal gesproken getraind met behulp van de Backpropagation leerregel. Het leerproces kan worden onderverdeeld in vier fasen: presentatie van een patroon, vergelijking van de uitvoer van het netwerk met de gewenste uitvoer, berekening van de partiele afgeleiden van de error functie met betrekking tot de gewichten van het netwerk en aanpassing van deze gewichten.

De invoerlaag wordt gebruikt om data aan het netwerk te presenteren. De units in de invoerlaag representeren elk een specifiek kenmerk van een object. De activaties van de units in de invoerlaag dienen als invoer voor de units in de verborgen laag. Elke unit in de verborgen laag gebruikt zijn gewogen invoer als argument voor een niet-lineaire transferfunctie. Als transferfunctie in een MLFF netwerk wordt vaak een sigmoïde functie gebruikt, weergegeven in figuur 2.3. De activaties van de units in deze verborgen laag dienen weer als invoer voor de units in de uitvoerlaag. De units in de uitvoerlaag gebruiken hun gewogen invoer ook weer als argument voor een niet-lineaire transferfunctie.

Aan de hand van de uitvoerwaarden wordt een vergelijking gemaakt tussen de gewenste en daadwerkelijke uitvoer. Na deze vergelijking vindt de terugpropageerfase (engels: backpropagation) plaats. In deze fase wordt de error (dat is het verschil tussen gewenste en daadwerkelijke uitvoer) teruggepropageerd door het ANN en worden alle gewichten in het netwerk zodanig aangepast dat het verschil tussen gewenste en daadwerkelijke uitvoer iets kleiner wordt. Deze procedure wordt patroon voor patroon herhaald en uiteindelijk resulteert een netwerk dat zich gedraagt zoals de gebruiker wenst.

Dit type netwerk was een van de eerste ANN's die uit meerdere lagen bestond. Hierdoor zijn de mogelijkheden van het netwerk aanmerkelijk uitgebreider dan de mogelijkheden van de eenvoudigere netwerken die eerder besproken zijn, hetgeen het onderzoek naar ANN's halverwege de 80-er jaren een sterke impuls heeft gegeven.

Hoofdstuk 3

Tabblad 1: Dataset

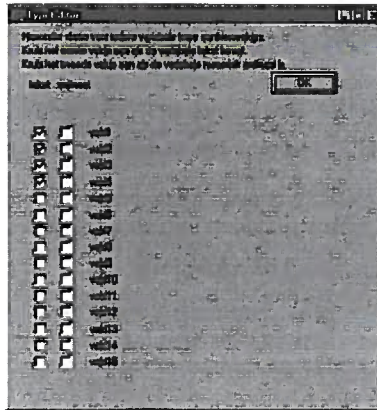
Dit eerste tabblad is bedoeld voor het inlezen en bewerken van een dataset. Hieronder worden de diverse functies van Neural Vision 2.0 die op dit tabblad betrekking hebben één voor één toegelicht.

3.1 Een bestand openen

Een bestand kan worden geopend via **Bestand / Open**. De eerste keer dat een bestand geopend wordt, zal er een dialogbox verschijnen waarop de gebruiker moet aangeven wat de types zijn van de verschillende variabelen. Deze dialogbox is weergegeven in figuur (3.1).

In deze dialog moet aangekruist worden welke variabelen tekst (nominaal) zijn en welke ordinaal numeriek zijn. Ordinaal numerieke variabelen zijn variabelen zijn weergegeven door geheeltallige nummers waarop een ordening gedefiniëerd is, maar waarvoor geldt dat de metriek niet dezelfde is als bij de natuurlijke getallen. Alle niet-aangekruiste variabelen worden opgevat als numerieke variabelen. Hierna verschijnt de dataset op het dataset tabblad. Het scherm ziet er nu zo uit als in figuur 1.2. Tekstkolommen worden grijs weergegeven, omdat ze niet worden gebruikt door de neurale netwerken in hun analyse.

De gebruiker kan nu naar keuze gegevens veranderen, operaties op de hele dataset uitvoeren of direct de dataset gaan analyseren met behulp van neurale netwerken. De relevante menu-opties voor bewerkingen op de dataset worden hieronder besproken.



Figuur 3.1: Type-editor waarmee de typen van de variabelen aangegeven kunnen worden

3.2 Het Bestandsmenu

Nieuw maakt een nieuw data bestand aan in dBase-format. Er verschijnt een tabel met vier kolommen en 1 rij, waarin de gebruiker gegevens kan gaan invoeren.

Open opent een bestand. Er worden twee formaten ondersteund: dBase (versie 3 en 4), en NDF (Neural Vision Definitions File). Een bestand van het type NDF is een tekstbestand (ASCII), dat bestaat uit een tabel met data met daarboven enkele extra gegevens. Bij het inlezen van zo'n bestand wordt het geconverteerd naar dBase-formaat.

Voor een uitgebreide beschrijving van het type NDF wordt de lezer verwezen naar de appendix.

Opslaan als... opent een dialoogbox, waarin een naam kan worden ingetypt waaronder het bestand moet worden opgeslagen.

Afdrukken leidt tot een submenu: met **Dataset** kan de data worden afgedrukt. Met **Sanger projectie** kan de Sanger projectie geprint worden. Met **ICA projectie** de ICA projectie. Met **polygonplot** de polygonplot. Met **statistiekentabel** de tabel met eenvoudige statistieken die op het vierde tabblad is weergegeven. Tenslotte kan met **histogram** het histogram afgedrukt worden dat op dat moment op het vierde tabblad te zien is.

Met **Sluiten** kan het huidige bestand afgesloten worden.

Afsluiten sluit Neural Vision 2.0 af.

3.3 Bewerken

In de spreadsheet kan de data bewerkt worden op een standaard wijze. Dit houdt in dat de spreadsheet op de standaard windows manier reageert op de shift-, control- en altoetsen, alsmede op de muisknoppen.

Rijen kunnen weggehaald of toegevoegd worden met behulp van de knoppenbalk onderaan het scherm (gebruik de + en de -). Het verwijderen van een record moet bevestigd worden door het aanklikken van de ok-knop in het dialoogvenster dat dan verschijnt. Om hetzelfde te doen met kolommen, moet uit het menu **Bewerken** de optie **Getalkolom toevoegen**, **Tekstkolom toevoegen** of **Kolommen verwijderen** gekozen worden. Deze laatste optie werkt alleen maar als er ook daadwerkelijk kolommen zijn geselecteerd om te verwijderen. Anders gebeurt er niets.

Met **Kolommen in/uitschakelen** worden de geselecteerde kolommen omgeschakeld: ingeschakelde kolommen worden uitgeschakeld en andersom. Kolommen die uitgeschakeld zijn, doen niet mee in de analyse door de netwerken. **Rijen in/uitschakelen** doet hetzelfde voor rijen. Rijen en kolommen die uitgeschakeld zijn, worden met een grijze kleur weergegeven.

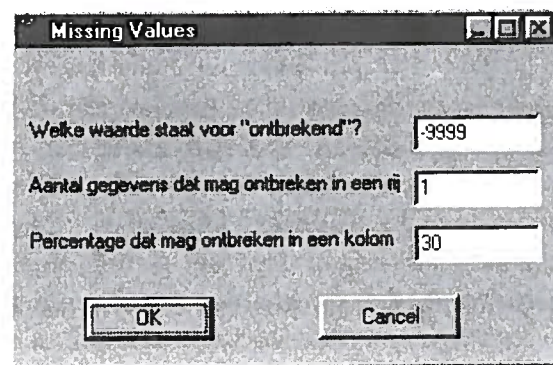
In het menu **Bewerken** staat verder nog de optie **Rijen verwijderen**, waarmee alle geselecteerde rijen verwijderd worden, en opties om de geselecteerde rijen of kolommen te kopiëren. Ook kan hier de selectie omgedraaid worden, zodat alle rijen of kolommen die geselecteerd waren dat niet meer zijn en andersom.

Met **Kolommen sorteren** kunnen de kolommen gesorteerd worden op variantie of entropie. Met de laatste optie, **Variabelentypes**, kunnen de types van de variabelen worden veranderd als daar in eerste instantie een vergissing in is gemaakt.

Hier is ook de mogelijkheid om tekstkolommen als getalkolommen te laten lezen, of getalkolommen als tekst. Dit heeft echter geen invloed op het onderliggende type in de database, maar alleen op de manier waarop Neural Vision de gegevens uitleest. Het kan handig zijn om bepaalde (ordinale) variabelen als tekst op te vatten, zodat er op gelabeld kan worden.

3.4 Ontbrekende waarden

Wanneer in de dataset gegevens ontbreken (missing values), is het mogelijk deze te schatten met behulp van de Missing Value Replacement optie in het



Figuur 3.2: Het Missing Values-dialogvenster

menu **Bewerken**.

Door in de menu-optie "Bewerken" vervolgens "Ontbrekende waarden" te kiezen wordt het dialoogvenster in figuur 3.2 zichtbaar. Hierin kunt u drie getallen invullen:

- het getal dat de ontbrekende gegevens representeert.
- het aantal kolommen dat in een observatie (rij) mag ontbreken (rijen-afbreekwaarde).
- het percentage aan rijen dat in een kolom mag ontbreken (kolom-afbreekwaarde).

U moet in het eerste veld hetzelfde getal invullen als het getal dat in uw data ingevuld is in de plaats van ontbrekende gegevens: als in uw data voor elke ontbrekende waarde 999 is ingevuld, moet u hier 999 invullen. Het is noodzakelijk dat alle ontbrekende waarden gerepresenteerd worden door hetzelfde dummy-getal. Het tweede en derde veld geven aan hoeveel gegevens er in een rij of kolom mogen ontbreken. De rijen-afbreekwaarde geeft aan hoeveel kolommen in een rij (observatie) mogen ontbreken: wanneer bij kolommen een 4 staat betekent dit dat er maximaal 4 kolommen in elke rij ontbrekende waarden mogen bevatten. Is aan deze eis voldaan, dan wordt een schatting van de ontbrekende waarde gemaakt met behulp van meervoudige lineaire regressie. Als onafhankelijke variabelen in deze regressie worden de kolommen gebruikt waarvoor in de rij met de ontbrekende waarde wel waarden zijn ingevuld. Wanneer er teveel gegevens ontbreken in een rij wordt

de hele rij door de gemiddelden over de hele dataset vervangen - ook de wel waargenomen waarden! Dit om waarnemingen van te geringe kwaliteit op een neutrale manier in de dataset op te nemen. Bij de kolom-afbreekwaarde wordt een percentage gehanteerd: indien er meer dan het ingestelde percentage van de observaties in een kolom ontbreekt, wordt de variabele in deze kolom als onvoldoende gedetermineerd beschouwd. In dat geval worden alle waarden in deze kolom vervangen door het gemiddelde van de wel gemaakte observaties. Uiteraard worden de waarden wel geschat wanneer er minder als het ingestelde percentage aan observaties in een kolom ontbreekt. Indien alle observaties in een rij of kolom ontbreken wordt alle data in deze rij danwel kolom vervangen door 0 of, indien 0 de ontbrekende waarde representeert, door 99999.

Deze twee afbreekwaarden dienen om te voorkomen dat wanneer er teveel gegevens ontbreken, er een zinloze schatting wordt gemaakt. U wordt aangeraden de standaard ingevulde waarden slechts te wijzigen indien hoogst noodzakelijk.

Wanneer de meervoudige lineaire regressie een waarde oplevert die buiten het oorspronkelijke bereik van de variabele in kwestie ligt wordt deze waarde afgekapt op het minimum of maximum.

Ook dient men er rekening mee te houden dat bij het berekenen van het gemiddelde van een kolom het getal dat oorspronkelijk de missing value aangaf (uiteraard) niet wordt meegenomen.

Hoofdstuk 4

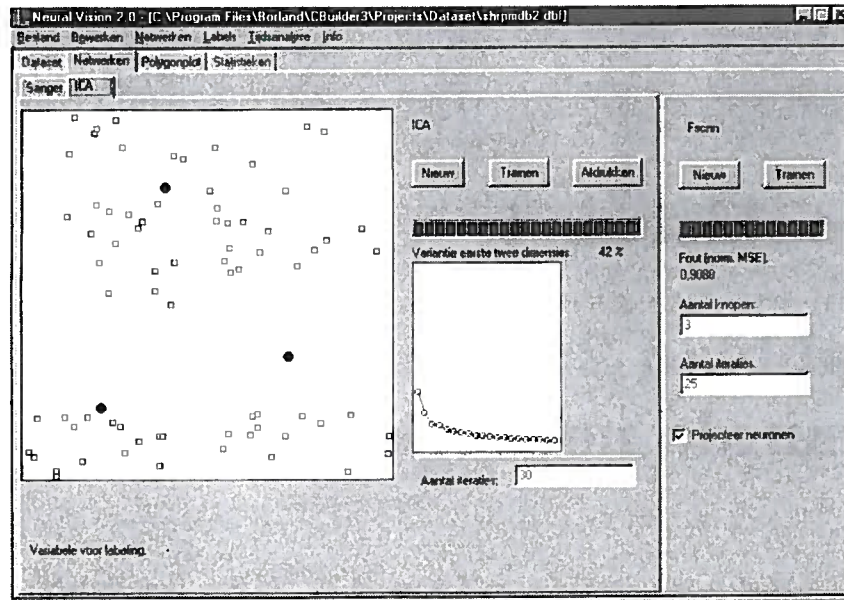
Tabblad 2: het trainen van de neurale netwerken

Dit tabblad is weergegeven in figuur (4.1). Indien een dataset geopend is kunnen de drie verschillende neurale netwerken getraind worden met behulp van dit tabblad. Met de Sanger- en ICA-netwerken kan de data geprojecteerd worden op twee dimensies. Het fscnn-netwerk (frequency sensitive competitive neural network) zoekt clusters in de data. Het Sanger- of ICA-net kan gekozen worden door het desbetreffende subtabblad aan te klikken. De gegevens van het fscnn-netwerk zijn altijd zichtbaar op dit tabblad. De theorie achter deze netwerken wordt in hoofdstuk 8 nader toegelicht. Op dit punt is het verstandig nog een keer te benadrukken dat tekst (nominale) variabelen niet mee worden geprojecteerd door de neurale netwerken en ook niet door het fscnn-netwerk gebruikt worden.

4.1 Sanger subtabblad

Zoals reeds vermeld dient het Sanger netwerk voor de projectie van data. Het Sanger netwerk poogt zoveel mogelijk van de in de data aanwezige informatie in twee dimensies te 'proppen'. Het Sanger netwerk wordt in detail besproken in hoofdstuk 8.

Een nieuw Sanger netwerk kan worden aangemaakt door op de knop Nieuw te klikken. Indien er al een Sanger netwerk in het geheugen aanwezig is, volgt een venster waarin gevraagd wordt of het bestaande netwerk gewist kan worden. Wanneer een nieuw netwerk wordt aangemaakt, wordt



Figuur 4.1: tabblad 2

een willekeurige projectie gekozen. Die verschijnt in het projectieveld (het grote vierkant uiterst links). Een nieuw netwerk kan tevens worden aangemaakt door middel van de optie Nieuw in het submenu Sanger van het menu Netwerken

Elk vierkantje in de projectie stelt een sample van de dataset voor. De samples kunnen gelabeld worden zodat verschillende groepen samples in de dataset duidelijk van elkaar te onderscheiden zijn. Zie de sectie over labels aan het eind van dit hoofdstuk voor meer details.

Is er eenmaal een (nieuw) Sanger netwerk aangemaakt dat kan dit getraind worden. De knop Trainen kan nu worden aangeklikt. De voortgang van het trainen wordt weergegeven in de progressiebalk. Vlak onder deze balk wordt de variantie van de eerste twee dimensies aangegeven, een maat voor de kwaliteit van de projectie. De kwaliteit wordt tevens visueel gerepresenteerd als een grafiek: de scree-plot Hoe hoger een punt op het verticale as ligt, hoe hoger de variantie van de bijbehorende dimensie.

Tijdens het trainen verandert de projectie van de dataset. Als na een trainingscyclus de projectie nog niet gestabiliseerd is, kan een volgende trainingscyclus worden uitgevoerd om het netwerk verder te trainen.

Onderaan het scherm zijn twee leerparameters in te stellen. Bij **Het aantal iteraties** kan de duur van een trainingscyclus worden veranderd. Er kunnen alleen gehele getallen groter dan 1 worden ingevoerd. Wordt een niet toegestane waarde ingevoerd, dan verschijnt de oude waarde weer op het scherm. Tevens kan de **Leersnelheid** worden aangepast.

Deze twee waarden zijn standaard zo ingesteld dat één trainingscyclus een goed resultaat geeft. De knop **Afdrukken** stuurt de projectie naar de printer.

4.2 ICA subtabblad

De gebruikersinterface van het ICA-netwerk is vrijwel het zelfde als die van het Sanger netwerk. Het enige verschil is dat bij het ICA netwerk geen leersnelheid kan worden ingesteld.

Achter de schermen verschillen deze netwerken echter wel substantieel van elkaar. De beide methoden worden in hoofdstuk 8 gedetailleerder toegelicht. Hier wordt volstaan met te zeggen dat de interpreteerbaarheid van de resulterende dimensies bij het ICA netwerk waarschijnlijk beter zal zijn dan bij het Sanger netwerk. Dit is een reden om het ICA netwerk doorgaans te prefereren boven het Sanger netwerk.

4.3 Fscnn netwerk

Door op de knop **Nieuw** in fscnn deel van het scherm (geheel rechts) te drukken wordt een nieuw netwerk aangemaakt. Indien er al een fscnn-netwerk in het geheugen aanwezig is, volgt een venster waarin gevraagd wordt of het bestaande netwerk gewist kan worden.

Indien er al een Sanger- of ICA-netwerk aangemaakt is worden de neuronen van het nieuwe fscnn-netwerk op het projectieveld geprojecteerd. Het projecteren van de neuronen kan worden in- en uitgeschakeld door de optie **Projecteer neuronen** te klikken.

Het aantal verborgen knopen dat het netwerk bevat kan worden ingesteld. Dit aantal komt overeen met het aantal clusters waarnaar het netwerk gaat zoeken. Standaard zijn dit er vijf.

Analoog aan het Sanger- en ICA-netwerk kan ook bij het fscnn-netwerk het aantal iteraties dat één trainingscyclus duurt worden ingesteld. Minder

iteraties betekent een kortere trainingstijd, maar kan er voor zorgen dat het netwerk nog niet is uitgetraind na één cyclus. Er is dan een volgende cyclus nodig. Een netwerk is niet uitgetraind zolang de neuronen nog bewegen door de projectie heen en de foutmaat (MSE, mean squared error) nog afneemt.

Merk op dat voor alle drie de netwerken de opties **Nieuw** en **Trainen** ook beschikbaar zijn in de submenu's van het menu **Netwerken**.

De gebruiker moet erop bedacht zijn dat meerdere runs van het fscnn netwerk kunnen resulteren in verschillende clusteringen. Dit is met name het geval wanneer van een erg hoog-dimensionale ruimte op het tweedimensionale vlak geprojecteerd wordt. De kwaliteit van de verschillende oplossingen, af te lezen aan de MSE fout, zal echter in alle gevallen vrijwel gelijk zijn.

4.4 Labels

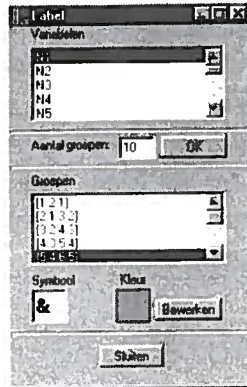
Om de dataset te kunnen analyseren kunnen samples gelabeld worden. Dat wil zeggen: alle samples kunnen worden ingekleurd. Dit heeft als gevolg dat van elk sample in de projectie duidelijk wordt tot welke groep deze behoort.

Voor numerieke variabelen zijn deze groepen gedefinieerd als intervallen in het bereik van een variabele over alle samples in de dataset. Het aantal van deze intervallen kan worden ingesteld. Ook bij tekst variabelen (nominale variabelen) is er sprake van groepen. Het aantal groepen is het aantal verschillende waarden dat deze variabele aanneemt in de dataset.

Als derde optie kan er gelabeld worden op fscnn-neuronen. Elk cluster krijgt een bepaalde kleur toegewezen. Samples die zich een bepaald cluster bevinden krijgen de kleur van dat cluster. Merk op dat er alleen op fscnn-neuronen gelabeld kan worden indien er een fscnn-netwerk is aangemaakt.

Samples kunnen zowel gelabeld worden met symbolen als met kleuren. Dit kan gedaan worden door één van deze twee opties te selecteren in het menu **Labels**. De labeling kan worden aan- en uitgezet door op de optie **Toon Labels** in dit menu te klikken. Zolang er geen groepen zijn gedefinieerd kan deze optie niet worden aangeklikt. Om deze groepen te definiëren, dient de optie **Bewerken...** gekozen te worden in dit menu. Het venster wat dan verschijnt is weergegeven in figuur 4.2. Het labellen met symbolen kan van pas komen als men een gelabelde plot wil printen op een niet-kleuren printer. Men moet echter in gedachten houden dat er slechts 32 symbolen gereserveerd zijn voor het labellen die cyclisch gebruikt worden en dat bij meer dan 32 groepen dus meerdere groepen door hetzelfde symbool kunnen

worden aangeduid.



Figuur 4.2: Het Labels | Bewerken... venster.

Door in de bovenste lijst te klikken kan de variabele worden geselecteerd waarop gelabeld moet worden. Alle variabelen die hiervoor beschikbaar zijn staan in deze lijst, zowel de numerieke, ordinaal numerieke als de nominale (tekst) variabelen. Indien een fscnn-netwerk in het geheugen aanwezig is, verschijnt de optie fscnn-neuronen onderaan de lijst.

Wanneer een variabele wordt geselecteerd, verschijnt in de onderste helft van het scherm een lijst van groepen. Dit is een lijst van intervallen in het geval van numerieke variabelen, een lijst van strings in het geval van nominale variabelen of een lijst van neuronen in het geval van labelen op clusters. Door op een element van een lijst te klikken worden de kleur en het symbool van de groep zichtbaar. Standaard worden de groepen van blauw naar rood gekleurd voor lage, respectievelijke hoge waarden in het variabelebereik. Het is echter ook mogelijk een eigen kleur te definiëren voor een groep. Dit gebeurt door te klikken of de knop *Bewerken* of te dubbel-klikken op het kleurvlakje. Het symbool kan ook worden veranderd.

Als er een lijst van groepen (intervallen) verschijnt, kan het aantal groepen worden ingesteld in het veld daarvoor, halverwege het scherm. Het aantal groepen bij fscnn-labeling en tekst variabelen kan niet worden ingesteld. Dit gaat (alleen) automatisch.

In figuur 4.2 is variabele N1 geselecteerd. Het bereik van deze variabele is verdeeld in tien intervallen. Deze tien intervallen (groepen) zijn weergegeven

in de onderste lijst. In deze lijst is het interval [5,4 : 6,5] geselecteerd. Dit interval kan in de projectie worden aangegeven als het &-tekenen of als de door het venster getoonde kleur.

De kleur van een groep kan gewijzigd worden door op het kleurvakje te dubbelklikken of door op de knop **Bewerken** te klikken. Er verschijnt dan een dialoogvenster waarmee de kleur van de geselecteerde groep veranderd kan worden. Er moet dus van te voren een groep geselecteerd zijn om de kleur van te bewerken. Om de kleur van een andere groep te veranderen dient dus eerst de betreffende andere groep geselecteerd te worden. Daarna kan de kleuren-editor geopend worden.

Hoofdstuk 5

Tabblad 3: de polygon-plot

Het derde tabblad heeft als doel meer informatie te geven over de tot stand gekomen projectie en clustering. Daartoe kunnen eigenschappen van samples en neuronen worden gevisualiseerd met behulp van de polygonplot. Het tabblad is afgebeeld in figuur 5.1

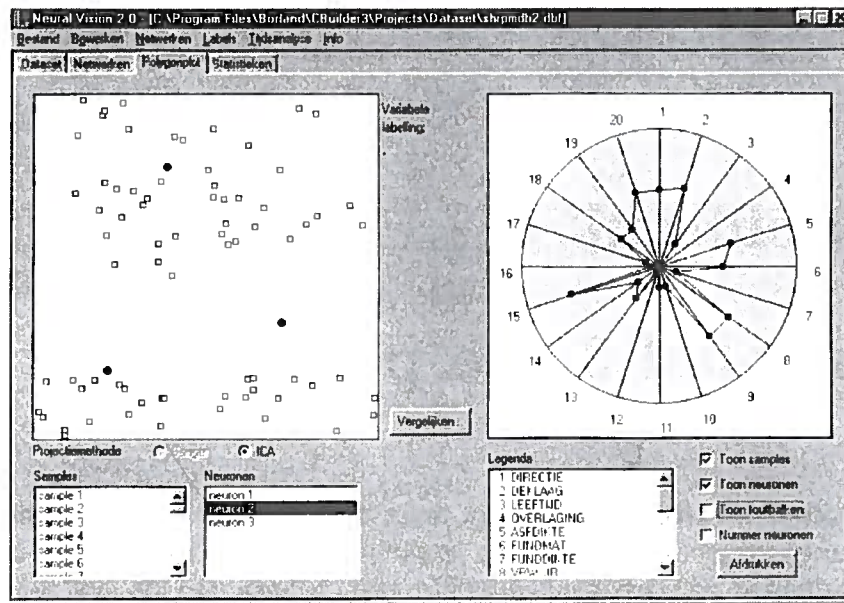
5.1 Het projectieveld

Links op dit tabblad staat het projectieveld. Indien de bijbehorende netwerken aanwezig zijn in het geheugen, kan zowel de projectie van het Sanger- als die van het ICA-netwerk getoond worden. De gewenste projectie worden gekozen door het aanklikken van één van de knoppen onder het projectieveld. Om meer inzicht te krijgen in de eigenschappen van bepaalde samples of neuronen kunnen deze geselecteerd worden om getoond te worden op de polygonplot.

Samples en neuronen kunnen op verschillende manieren geselecteerd worden. Namelijk door met de linker muisknop te klikken op een sample of neuron in het projectieveld. Tevens kan dit door een sample of neuron te selecteren in de lijst van samples (linksonder) of lijst van neuronen (midden onder).

Een geselecteerd sample zal in het projectieveld zwart worden ingekleurd en verschijnen op de polygon-plot. Tevens zal dit sample geselecteerd worden in de spreadsheet op het eerste tabblad. Er kunnen meerdere samples tegelijk worden geselecteerd.

Een geselecteerd neuron zal in het projectieveld blauw worden ingekleurd



Figuur 5.1: tabblad drie

en verschijnen op de polygon-plot. Net als bij samples, kunnen ook meerdere neuronen tegelijk worden geselecteerd.

Door met de linker muisknop op een positie te klikken waar geen sample of neuron staat, worden alle selecties ongedaan gemaakt. Door te klikken op een reeds geselecteerd sample of neuron zal de selectie ervan ongedaan gemaakt worden.

5.1.1 Kiezen doelen in projectie

Door met de rechter muisknop te klikken op een willekeurige plaats in het projectieveld, kan een terugprojectie worden uitgevoerd. Op het scherm verschijnt een virtueel sample. Dit sample wordt gerepresenteerd als een groen vierkantje. Ook op de polygonplot verschijnen de waarden van de variabelen van het virtuele sample. Deze variabelewaarden zijn de twee dimensionale coördinaten van het projectieveld teruggerekend naar de oorspronkelijke n-dimensionale dataset. De gehanteerde methode van terugprojectie benadert de oorspronkelijke dataset zo goed mogelijk. Omdat tijdens de projectie echter informatie verloren is gegaan, kan het zijn dat de polygonplot van het

virtuele sample enigszins afwijkt van de plot van nabij gelegen samples.

5.2 De polygonplot

Om de eigenschappen van de samples en neuronen te visualiseren is er de polygonplot, rechtsboven op het tabblad. Elke spaak in het wiel van de polygonplot representeert een numerieke variabele. De nummers rond de plot verwijzen naar de nummers in de legenda rechtsonder op het scherm. Wanneer een sample getoond wordt op de polygonplot worden de waarden van de variabelen van dat sample als rode vierkantjes getoond. Hoe groter de afstand van het vierkantje tot het midden van de polygonplot, hoe groter de waarde van de variabele van het sample. De minimale waarde over de hele dataset van een variabele komt overeen met het midden van de plot. De maximale waarde wordt aangegeven door een punt op de rand van de plot. Onderaan staan nog een aantal opties die in- en uitgeschakeld kunnen worden:

- **toon samples:** toont, indien ingeschakeld, de geselecteerde samples op de polygonplot.
- **toon neuronen:** toont, indien ingeschakeld, de geselecteerde neuronen op de polygonplot.
- **toon foutbalken:** toont, indien ingeschakeld, foutbalken bij de neuronen in de polygonplot. De weergegeven foutbalken representeren de 95 % voorspellingsintervallen, en worden slechts bepaald voor numerieke variabelen. Bij een scheve verdeling kunnen de eindpunten van de foutbalken buiten de polygonplot komen te liggen.
- **nummer neuronen:** toont, indien ingeschakeld, nummers bij de neuronen in het projectieveld.

5.3 Tijdsanalyse

Deze optie is toegevoegd om samples in de projectie door de tijd te kunnen volgen. Het idee is de ontwikkeling van (eigenschappen van) samples te volgen door de tijd.

Er zijn bepaalde objecten. Daarvan wordt de waarde van een vaste set eigenschappen (variabelen) op verschillende momenten in de tijd gemeten.

INVROW	FSCNN	VELD2	VELD3	VELD4	VELD5	VELD6	VELD7	VELD8
1		1	1	0.5	0.4	0.1	0	ins-sets
1		2	1	0.7	0.3	0	0	ins-sets
1		3	1.5	1.1	0.4	0.1	0.1	ins-sets
1		4	2	1.4	0.6	0.1	0.1	ins-sets
1		5	2.5	1.8	0.7	0.1	0.1	ins-sets
1		6	3.1	2.1	0.8	0.1	0.1	ins-sets
1		7	3.6	2.4	1	0.1	0.1	ins-sets
1		8	4.1	2.8	1.1	0.2	0.2	ins-sets
1		9	4.6	3.1	1.3	0.2	0.2	ins-sets
1		10	5.1	3.5	1.4	0.2	0.2	ins-sets
2		10	4.9	3	1.4	0.2	0.2	ins-sets
3		10	4.7	3.2	1.3	0.2	0.2	ins-sets
4		10	4.6	3.1	1.5	0.2	0.2	ins-sets
5		10	5	3.6	1.4	0.2	0.2	ins-sets
6		10	5.4	3.9	1.7	0.4	0.4	ins-sets

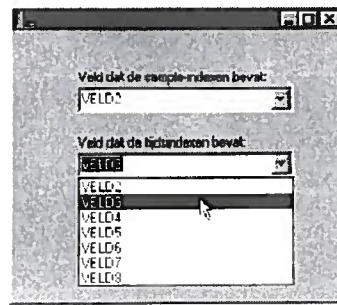
Figuur 5.2: fragment van de gebruikte dataset

Het resultaat van een zo'n waarneming heet een sample. Er zijn meerdere samples (groep) van hetzelfde object gekoppeld aan verschillende tijdstippen. In een dataset staan samples van verschillende objecten. Daarom moet het programma objecten en tijdstippen kunnen onderscheiden. Ieder sample krijgt een object-index en tijdsindex. De combinatie van deze twee indices maakt het sample uniek en identificeerbaar. Samples met dezelfde object-index vormen een groep van waarnemingen van het object in de tijd. De twee kolommen met de object-index en de tijdsindex worden aan het programma kenbaar gemaakt.

De procedure zal verduidelijkt worden met een voorbeeld, uitgaande van de volgende, fictieve dataset. Deze dataset is ontstaan door aan de iris dataset twee kolommen toe te voegen: VELD2 voor de objectindices en VELD3 voor de tijdsindices. De rest van de variabelen (VELD4 t/m VELD8) is overgenomen van de iris dataset. In de dataset is voor de groep samples met index 1 een relatie aangebracht tussen de waarde van VELD4 en de tijdsindex. Deze relatie zal later terug te vinden zijn in de projectie. Een deel van deze dataset is te zien in figuur 5.2.

Om de kolommen met groepen en tijdsindices te kunnen definiëren is er de optie **Bepaal kolommen** in het menu **Tijdsanalyse**. Na selecteren van deze menuoptie verschijnt de dialoog die is weergegeven in figuur 5.3

In het bovenste veld kan met de muis de gewenste kolom worden gekozen voor object-indices, ofwel de groepen. In het onderste veld moet de kolom die de tijdsindices bevat gekozen worden. Beide indices worden niet meegenomen in de analyse. Wil men bijvoorbeeld tijd toch mee-projecteren, dan dient de



Figuur 5.3: Bepaal kolommen

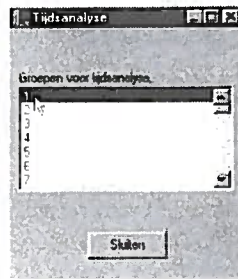
kolom die als tijdsaanduiding dienst doet te worden gekopieerd. Deze nieuwe kolom kan dan mee worden geprojecteerd.

De gebruiker dient bewust te zijn van het feit dat de neurale netwerken voor projectie getraind worden met de data verkregen op *alle* tijdstippen. Al deze data is dus in een plaatje te zien. Wil men afzonderlijke projecties voor de verschillende tijdstippen van waarneming, dan dient men hiervoor zelf verschillende databestanden aan te maken: één voor elk tijdstip in de oorspronkelijke dataset. Als een dergelijke analyse uitgevoerd wordt moet men er wel op bedacht zijn dat iedere projectie opgeschaald wordt tot beeldvullend formaat. Dit kan de interpretatie van de projecties in relatie tot elkaar bemoeilijken.

Een andere complicerende factor is in dit geval het feit dat de verschillende plots een verschillende oriëntatie kunnen hebben. Dit wordt veroorzaakt doordat de assen in verschillende trainingsruns van de neurale netwerken verwisseld danwel gespiegeld kunnen worden (overigens zonder kwaliteitsverlies in termen van verklaarde variantie). Door een aantal keer een nieuw neurale netwerk aan te maken en te trainen verkrijgt men meestal echter snel een plot met de gewenste oriëntatie.

Voor het voorbeeld wordt de kolom VELD2 gekozen als voor de object-indices en de kolom VELD3 als kolom voor tijdsindices. Deze kolommen worden opgeslagen in het configuratiebestand. Het is daarom niet altijd nodig deze procedure uit te voeren. Merk op dat beide kolommen niet worden meegenomen in de analyse en dus ook niet op de polygonplot verschijnen.

Zijn beide kolommen vastgelegd dan kan worden ingesteld welke groep samples getoond moet worden in de tijdsanalyse. Dit kan worden gedaan via de optie Bepaal groep in het menu Tijdsanalyse. Na het selecteren van

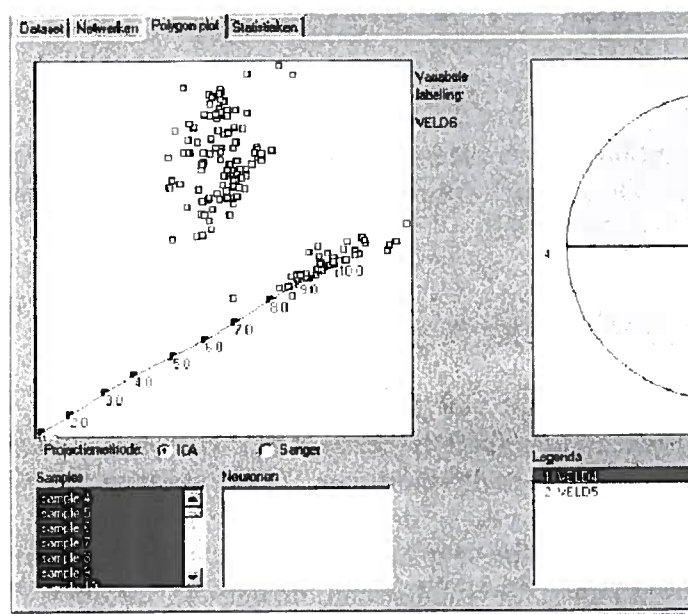


Figuur 5.4: Bepaal groep

deze menuoptie verschijnt de dialoog uit figuur 5.4. In het voorbeeld wordt de sample groep met index 1 gekozen.

Uit de lijst kan de groep gekozen worden waarvan het verloop in de tijd getoond moet worden. Tenslotte kan door het aanklikken van de optie **Teken analyse** in het menu **Tijdsanalyse** de analyse getoond worden. Dit gebeurt in het projectieveld op tabblad drie.

Alle samples van object 1 worden verbonden door een lijn. Deze lijn loopt van de laagste tijdsindex naar de hoogste. De tijdsindex staat als een getal naast het sample aangegeven, zoals is te zien in figuur 5.5. De samples die in de tijdsanalyse verbonden worden, worden geselecteerd zodat ze ook op de polygonplot verschijnen. Op de polygonplot is het verloop van VELD4 duidelijk zichtbaar. De samples kunnen met elkaar vergeleken worden, door ze afwisselend wel of niet te selecteren op de polygonplot.



Figuur 5.5: De tijdsanalyse getekend in de projectie op tabblad drie.

Hoofdstuk 6

Tabblad 4: Statistieken

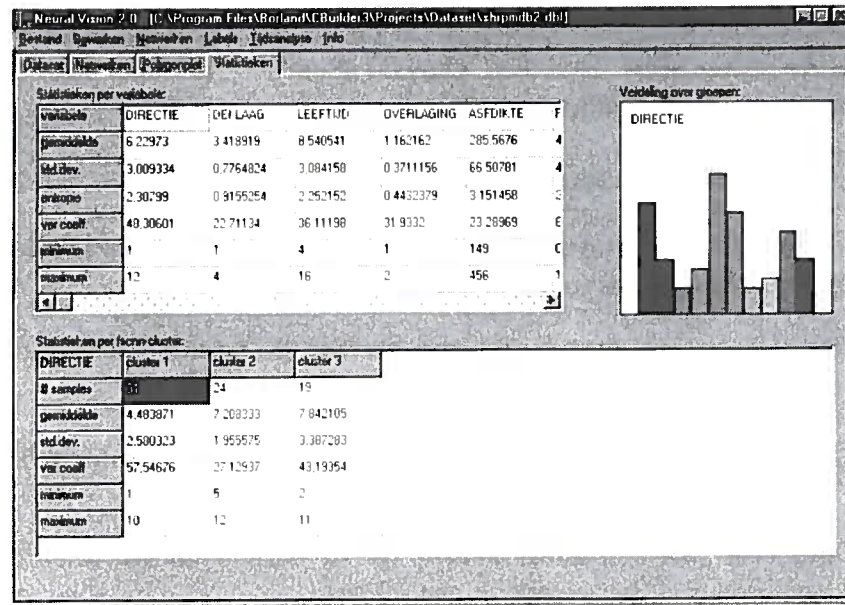
Het vierde tabblad geeft meer inzicht in de eigenschappen van zowel dataset als projectie. Dit gebeurt door het geven van aanvullende statistieken in twee tabellen en een histogram.

6.1 Statistieken per variabele

In figuur 6.1 is tabblad vier (Statistieken) te zien. De tabel linksboven geeft statistieken voor alle numerieke variabelen in de dataset. De verschillende variabelen staan in de kolommen. Door op een variabelenaam te klikken (in de bovenste rij) kan de actuele variabele worden geselecteerd. De actuele variabele wordt gebruikt voor het labelen van de samples in de projecties op de andere tabbladen. Tevens bepaald de actuele variabele de waarde van de statistieken in de tabel Statistieken per cluster.

De tabel Statistieken per variabele toont de volgende statistieken:

- **variabele**: de naam van variabele staat boven aan elke kolom
- **gemiddelde**: de gemiddelde waarde van een variabele, genomen over alle samples in de dataset.
- **std.dev.**: de standaard deviatie van een variabele, genomen over alle samples in de dataset.
- **entropie**: de entropie van een variabele.
- **var. coeff**: de variatie coefficient. Gedefinieerd als $\text{gemiddelde}/\text{std.dev} * 100 \%$.



Figuur 6.1: tabblad vier

- **minimum:** de minimale waarde die een variabele aanneemt, genomen over alle samples in de dataset.
- **maximum:** de maximale waarde die een variabele aanneemt, genomen over alle samples in de dataset.

Voor categorische variabelen wordt alleen het minimum en maximum getoond omdat de overige statistieken niet gedefinieerd zijn op dit soort variabelen.

6.2 Statistieken per cluster

De tabel linksonder geeft statistieken per cluster voor een bepaalde variabele. Deze tabel bevat alleen waarden als een fscnn-netwerk is getraind. Een mogelijkheid om deze variabele te selecteren is het klikken op de variabele-naam in de bovenste tabel. Als dus door op de tabel te klikken een variabele wordt geselecteerd wordt, zal ook de labeling van de samples veranderen. De variabele kan ook geselecteerd worden door deze te selecteren via de optie **Bewerken...** in het menu **Labels**.

Het tabelelement linksboven toont de actuele variabelenaam. Een kolom bevat de statistieken voor een bepaald cluster. De tabel Statistieken per cluster toont de volgende statistieken.

- **# samples**: het aantal samples in een bepaald cluster.
- **gemiddelde**: de gemiddelde waarde van de actuele variabele, genomen over alle samples in een cluster.
- **std.dev.**: de standaard deviatie van de actuele variabele, genomen over alle samples in een cluster.
- **var. coeff**: de variatie coefficient. Gedefinieerd als $\text{gemiddelde}/\text{std.dev} * 100 \%$.
- **minimum**: de minimale waarde die de actuele variabele aanneemt, genomen over alle samples in een cluster.
- **maximum**: de maximale waarde die de actuele variabele aanneemt, genomen over alle samples in een cluster.

6.3 Het histogram

Deze grafiek toont voor de actuele variabele de verdeling over de intervallen die gebruikt worden voor het labelen van de data. Hoe meer samples er in een groep vallen, hoe hoger de balk voor de groep is. Voor elke groep is er één balk. Standaard zijn er tien intervallen gedefinieerd en zal dus ook het histogram tien balken kennen. De grootste balk is altijd geschaald op $2/3$ van de hoogte van het hele figuur. Voor nominale variabelen toont het histogram het aantal samples met dezelfde waarde. De actuele variabele en het aantal intervallen kunnen worden bepaald via de optie **Bewerken...** in het menu **Groepen**. De naam van actuele variabele staat linksboven in het histogram.

Hoofdstuk 7

Uitgewerkte voorbeelden

7.1 Inleiding

In dit hoofdstuk worden twee voorbeelden gegeven van data-analyses met Neural Vision 2.0. De eerste dataset die hiervoor gebruikt wordt is de 'Boston' dataset, bekend van de eerste versie van Neural Vision. Als tweede dataset wordt de zogenaamde 'FRAME3' dataset geanalyseerd. Deze dataset is afkomstig uit de RWS organisatie en representeert dus een belangrijk toepassingsdomein.

Dit hoofdstuk zal vooral ingaan op het analyseren van een dataset en minder op het omgaan met het eerste tabblad. Daarvoor wordt verwezen naar hoofdstuk 3.

De gebruikte datasets zijn terug te vinden in de map `examples` die wordt meegeïnstalleerd met het Neural Vision 2 pakket. Van te voren moet worden opgemerkt dat de resultaten van het trainen van een neurale netwerk kunnen verschillen van dit voorbeeld. Onder meer de projectie kan er anders uitzien dan de afbeeldingen hier gepresenteerd. Dit is het gevolg van het feit dat de netwerken ongevoelig zijn voor spiegelingen.

De Boston dataset zal worden gebruikt om het Sanger netwerk en het fscnn-netwerk te demonstreren. Tevens wordt met deze dataset gedemonstreerd hoe het analyseren van de data met de tabbladen drie (Polygonplot) en vier (Statistieken) in z'n werk gaat. De FRAME3 dataset zal worden geanalyseerd met behulp van het ICA netwerk. Allereerst zal nu de Boston dataset geanalyseerd worden, vervolgens de FRAME3 dataset.

7.2 Beschrijving van de Boston dataset

Deze dataset bestaat uit 506 samples van elk 14 variabelen. Ieder sample betreft een cluster huizen in Boston. De variabelen geven sociaal geografische indicatoren weer. De variabelen uit de dataset zijn:

1. hoofdelijke criminaliteitsratio per deelgemeente (CRIM)
2. deel van de woongrond verdeeld in percelen groter dan 25.000 vierkante voet (ZN)
3. deel van de grond in gebruik door industrie en nijverheid (geen middenstand) in de deelgemeente (INDUS)
4. boolese variabele die aangeeft of perceel aan rivier ligt: 1=ja, 0=nee (CHAS)
5. concentratie aan nitraat-oxide in aantal deeltjes per 10 miljoen (NOX)
6. gemiddeld aantal kamers per pand (RM)
7. percentage van door de eigenaren bewoonde huizen van voor 1940 (AGE)
8. gewogen afstanden tot vijf werkgelegenheidscentra in Boston (DIS)
9. index voor toegankelijkheid van uitvalswegen (RAD)
10. onroerend-goed belasting per \$ 10.000 (TAX)
11. leerling/leraar verhouding per deelgemeente (PTRATIO)
12. $1000(B_k - 0.63)^2$ met B_k het percentage Amerikanen van Afrikaanse afkomst in de deelgemeente (B)
13. percentage van de bevolking met lagere status (LSTAT)
14. mediaan van de prijs voor huizen die bewoond worden door de eigenaren (MEDV)

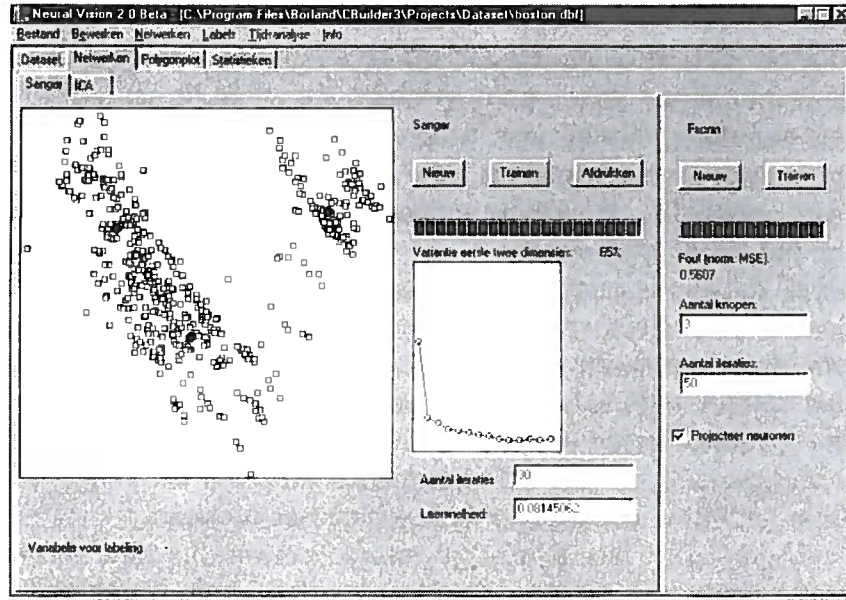
7.3 Analyse van de Boston dataset

De 'Boston' dataset is interessant om te analyseren vanwege de sociale relevantie ervan. Zoals uit de analyse blijkt zijn er in de dataset verschillende clusters aanwezig, met elk zeer herkenbare eigenschappen.

De analyse wordt begonnen door het openen van de dataset. Dit gaat door het aanklikken van de optie **Openen** in het menu **Bestand**. De dataset **Boston.dbf** wordt geselecteerd.

Als de dataset is geopend kan er een Sanger netwerk aangemaakt worden om de data te projecteren op een twee-dimensionale ruimte. Het aanmaken van een Sanger netwerk geschiedt door op tabblad 2 (**Netwerken**) op sub-tabblad **Sanger** te klikken op de knop **Nieuw**. Er verschijnt een willekeurige lineaire projectie geprojecteerd op twee dimensies. Het aanmaken van een nieuw Sanger netwerk kan ook geschieden door het selecteren van de optie **Nieuw** in het sub-menu **Sanger** in het menu **Netwerken**.

Vervolgens dient het zojuist gecreëerde netwerk getraind te worden. Dit kan met behulp van de knop **Trainen** op het eerder omschreven tabblad. Het zelfde effect heeft de optie **Trainen** in het sub-menu **Sanger** in het menu **Netwerken**. Het netwerk wordt nu getraind met de parameters die onder de scree-plot staan aangegeven. Na het trainen ziet het scherm er als volgt uit:



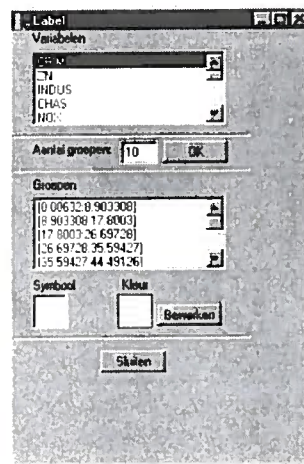
Figuur 7.1: tabblad twee met een getraind Sanger netwerk

De scree-plot laat duidelijk zien dat veel van de variantie in de dataset in één dimensie kan worden weergegeven. De scree-plot loopt namelijk erg schuin af aan de linkerzijde.

In de projectie zien we duidelijk een aantal clusters. Linksbovenaan is een groot cluster te zien, net als rechtsonderaan. Dit laatste cluster is weer verdeeld in twee kleinere clusters. Ook deze twee clusters bevatten elk weer kleinere clusters.

De data wordt nu nader geanalyseerd. In het eerste plaats door het labelen van de data. Hierbij worden de datapunten gekleurd. In de tweede plaats door gebruik te maken van de polygonplot en de aanvullende statistieken.

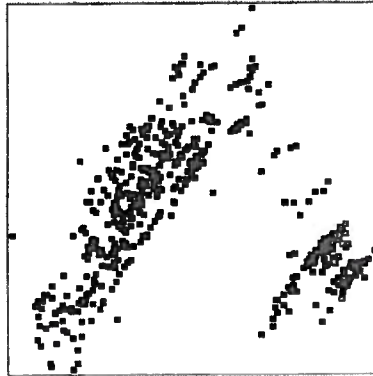
Zoals eerder uitgelegd kunnen de samples in de projectie worden gekleurd. Door het kiezen van de optie *Bewerken...* in het menu *Groepen* verschijnt het volgende dialoog venster:



Figuur 7.2: Het dialoog venster voor het labelen van data.

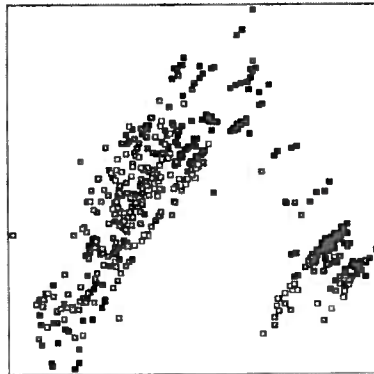
Eén van de veertien variabelen kan nu worden geselecteerd om een kleuring (labeling) aan te brengen in de projectie. Hierbij is het altijd zo dat de kleinste waarden van die variabele gerepresenteerd worden door blauwachtige kleuren. De hoogste waarden nemen roodachtige kleuren aan. Dit is na te gaan door het aanklikken van verschillende groepen in de lijst op het venster.

Wordt nu bijvoorbeeld de variabele CRIM gekozen als label, dan ontstaat de kleuring zoals afgebeeld in het volgende (zwart-wit) figuur. In deze figuur is blauw de overheersende kleur, behalve rechtsonder in de projectie. Waar enkele groene, oranje en zelfs rode tinten zijn waar te nemen. Hieruit kan de conclusie worden getrokken dat criminaliteit in deze huizenclusters is geconcentreerd. Het mini-clustertje rechtsonder springt eruit.



Figuur 7.3: De Sanger projectie gelabeld op CRIM.

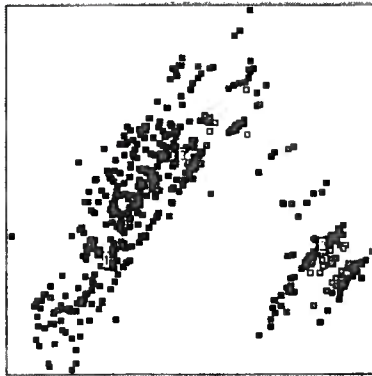
Als vervolgens de variabele AGE (die iets uitdrukt over de leeftijd van de huizen) gekozen wordt als label, dan worden de verschillen nog duidelijker. Rechtsonder bevinden zich veel oude huizen. Oude huizen zijn rood gekleurd, nieuwe blauw. In de top van de grote wolk bevinden zich ook veel oude huizen. Onderaan in deze wolk zijn vooral nieuwere huizen te vinden. Dit is te zien in de onderstaande figuur:



Figuur 7.4: De Sanger projectie gelabeld op AGE.

De laatste variabele die in dit voorbeeld gebruikt wordt om te labelen is B. Deze variabele is een (getransformeerde) maat voor het aantal Amerikanen

van Afrikaanse afkomst per buurt. Hoe lager de waarde van deze variabele, hoe sterker deze groep is gerepresenteerd in de desbetreffende buurt. Erg duidelijk is te zien dat deze bevolkingsgroep is geconcentreerd in één wijk, namelijk het mini-cluster rechts-onderaan: de blauw gekleurde samples.



Figuur 7.5: De Sanger projectie gelabeld op de variabele B.

Langzaam aan begint zich een beeld te vormen van wat de verschillende clusters representeren. Het cluster rechts-onderaan representeert de binnenstad van Boston, waarbij het mini-cluster met veel Amerikanen van Afrikaanse afkomst en een hoge criminaliteit een typisch Amerikaans 'ghetto' representeert. Dit zijn conclusies die we op basis van de tot nu toe uitgevoerde analyse kunnen trekken (ze zijn in ieder geval erg waarschijnlijk). Deze informatie was niet van tevoren aanwezig in de dataset.

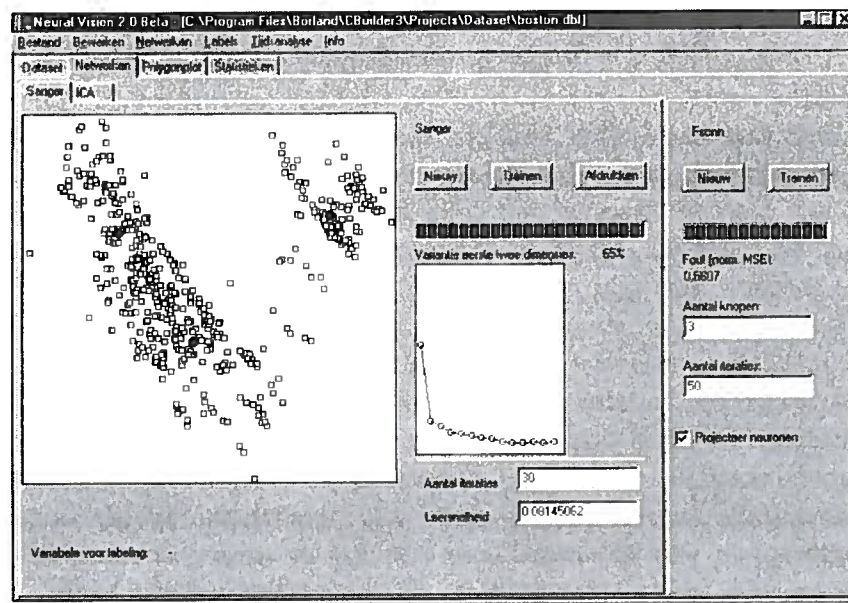
Deze conclusie is niet de enige die door labeling aan de dataset kan worden verbonden. Het kan ook duidelijk worden gevisualiseerd dat de binnenstad een goede toegang geeft tot uitvalswegen, dat er veel belasting wordt betaald, dat de huizen relatief weinig waard zijn, enzovoort. In het oog springende eigenschappen van het platteland zijn dat er weinig criminaliteit is, dat er grote percelen zijn en dat de lucht een laag NOX gehalte heeft.

De volgende stap in dit voorbeeld is het trainen van een fscnn netwerk met drie units. Zoals in deze handleiding eerder behandeld is, zal iedere unit representatief worden voor een deel van de gegevens. Hopelijk wordt één unit representatief voor het platteland, één voor de buitenwijken en één unit voor de binnenstad.

Om een fscnn netwerk aan te maken, dient gebruik gemaakt te worden van de knop **Nieuw**, geheel links op het tabblad **Netwerken**. Ook kan gekozen worden voor de optie **Nieuw** in het sub-menu **Fscnn** in het menu **Netwerken**. In de projectie verschijnt nu een initieel netwerk met vijf neuronen. Dit aantal wordt op drie gezet door het getal vijf in drie te veranderen in het veld **Aantal verborgen knopen** rechtsmidden op het tabblad.

Het aangemaakte netwerk kan nu worden getraind door te klikken op de knop **Trainen**. Eveneens kan gekozen worden voor de optie **Trainen** in het sub-menu **Fscnn** in het menu **Netwerken**. Het trainen van het netwerk wordt gevisualiseerd door de neuronen die door de projectie heen bewegen.

Na het trainen van het netwerk kan de data in de projectie worden gelabeld met de units van het fscnn netwerk. Hierbij krijgt iedere unit zijn eigen kleur toegewezen. Elk sample krijgt nu de kleur van het dichtstbijgelgen neuron. Als neuron 1 met de kleur geel gelabeld wordt, in plaats van het standaard blauw, is de volgende afbeelding het gevolg:

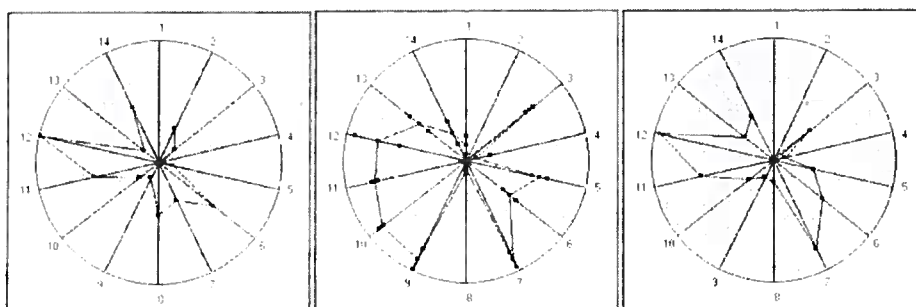


Figuur 7.6: De Singer projectie gelabeld met fscnn neuronen.

Het platteland wordt nu geel (i.p.v. blauw) gekleurd (unit 1), de binnen-

stad groen (unit 2), de buitenwijken rood (unit 3). De gewichten vectoren van deze units zijn nu representatief geworden voor deze drie groepen. De volgende stap is het visualiseren van deze vectoren met behulp van de polygonplot op tabblad drie.

Door het selecteren van een neuron verschijnt deze op de polygonplot. Een neuron kan geselecteerd worden door op het neuron te klikken in het projectieveld. Tevens kan selectie plaatsvinden door op het gewenste neuron te klikken in de lijst van neuronen rechtsonder het projectieveld. Hieronder zijn de visualisaties te zien. Merk op dat bij neuron twee de foubalken worden getoond. Dit wordt gedaan door het aanklikken van de optie Toon foubalken onder de polygonplot.



Figuur 7.7: De polygonplot representatie van neuron 1 (l), 2 (m) en 3 (r).

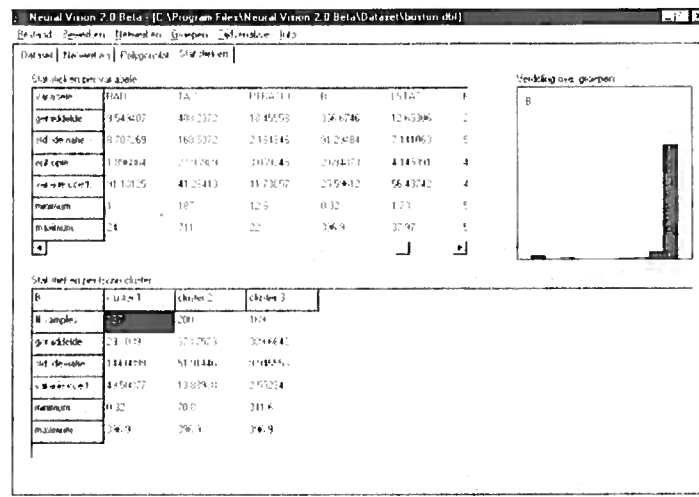
Uit deze visuele representatie kunnen de karakteristieke kenmerken van de verschillende groepen eenvoudig worden afgeleid. Duidelijk is te zien dat groepen 1 en 3 (platteland en buitenwijken) relatief veel overeenkomsten hebben en dat groep 2 (binnenstad) er erg van verschilt. De kenmerken van groep 1 (platteland) zijn: hoge waarde variabele B (laag aantal Amerikanen van Afrikaanse afkomst), lage waarde voor variabele AGE (dus jonge huizen), relatief hoge waarde variabele ZN (dus grote percelen), enzovoort.

De meest in opvallende eigenschappen van de binnenstad (groep 2) zijn de hoge waarden van variabele RAD, TAX en PTRATIO (respectievelijk toegankelijkheid uitvalswegen, belasting, leraar/leerling verhouding), de relatief hoge van variabele CRIM (criminaliteit), enzovoort.

Als laatste kan gezegd worden dat groep 3 (buitenwijken) redelijk overeenkomt met groep 1 (platteland). Er zijn echter wel kenmerken die deze

groepen van elkaar onderscheiden. Met name de in waarde van variabele AGE (percentage van door de eigenaren bewoonde huizen van voor 1940), variabele INDUS (deel van de grond in gebruik door de industrie). en variabele DIS (afstand tot werkgelegenheden). Deze vergelijking kan goed gemaakt worden door beide neuronen tegelijk te tonen op de polygon plot.

De analyse kan worden voortgezet door een blik te werpen op het tabblad Statistieken. Als de data gelabeld wordt op variabele B (percentage Amerikanen van Afrikaanse afkomst) ziet het tabblad er als volgt uit:



Figuur 7.8: De statistieken op tabblad vier.

De tabel linksboven laat de statistieken per variabelen zien. Opvallend is dat de standaarddeviatie vrij hoog is (91.29) voor de variabele B. Het histogram laat zien dat er maar weinig samples zijn met een lage waarde voor deze variabele, er zijn dus maar weinig deelgemeenten met een hoog percentage Afro-Amerikanen in Boston. De eerdere conclusie dat veel Afro-Amerikanen in de binnenstad wonen wordt gestaafd door de tabel linksonder: de gemiddelde waarde voor de variabele in cluster 1 (291,093) is lager dan de in andere twee clusters.

7.4 Beschrijving van de FRAME3 dataset

Deze dataset bevat 225 patronen (samples). Ieder patroon bevat 11 numerieke- en 4 tekstvariabelen. De numerieke variabelen zijn:

intmvt : verkeersintensiteit in voertuigen per etmaal

lberm : lengte bermen in kilometers

brberm : breedte van de bermen in kilometers

oppkruid : oppervlakte kruidachtige vegetatie in ha.

opphout : oppervlakte houtachtige vegetatie in ha.

oppwater : oppervlakte water-gangen en waterpartijen

lgschaw : lengte geluidsschermen aarden wallen

lgschnaw : lengte geluidsschermen anders dan aarden wallen

lgscherm : totale lengte geluidsschermen (som van de twee voorgaande)

pbermc0 : percentage van obstakel vrije bermen

oppberm : oppervlakte bermen, water-gangen en waterpartijen.

De tekst-variabelen zijn:

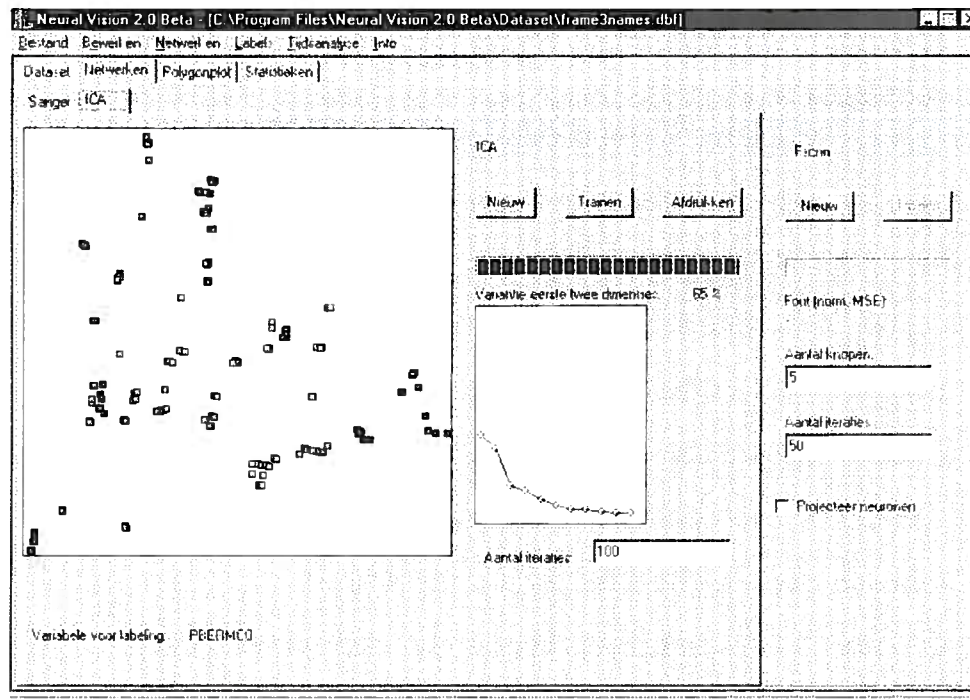
budgetlm : budget voor landschap en milieu

budgetdisc : tekst-variabele die hoogte van budget aangeeft

dienstkring : dienstkring

jaar : jaar.

Deze dataset is oorspronkelijk samengesteld om te pogen de variabele 'budgetlm' (budget voor landschap en milieu) te modelleren met de andere numerieke variabelen als onafhankelijke variabelen. In deze analyse zal het gebruik van budgetlm als 'afhankelijke variabele' ook nog onder de aandacht gebracht worden. Hiernaast kunnen met behulp van Neural Vision enkele andere interessante conclusies getrokken worden. De gebruiker wordt aangeraden om de analyse zelf uit te voeren tijdens het lezen van het voorbeeld.



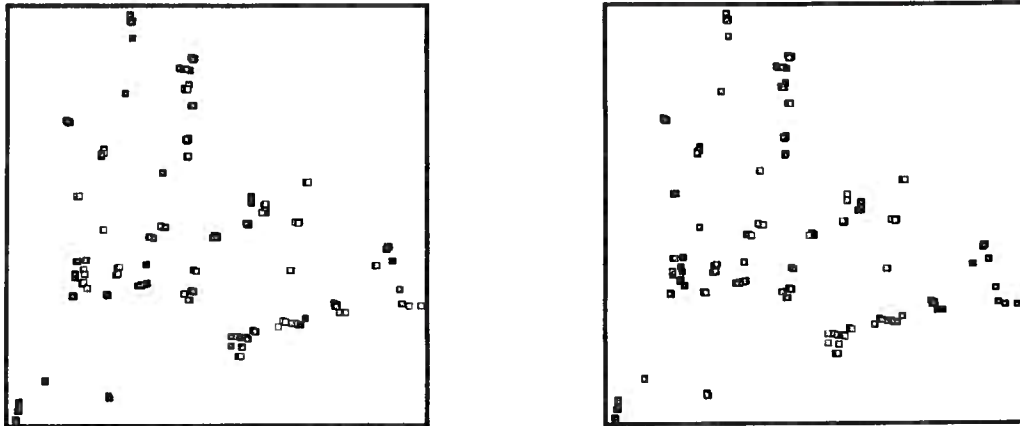
Figuur 7.9: ICA projectie van frame3 dataset.

7.5 Analyse van de FRAME3 dataset

De FRAME3 dataset wordt in het geheugen geladen via het bestandsmenu. Vervolgens wordt op het tweede tabblad een ICA netwerk aangemaakt. Dit heeft het scherm weergegeven in figuur (7.9) tot gevolg. De gebruiker kan bij herhaling van het experiment een andere projectie krijgen vanwege het eerder genoemde feit dat het ICA net ongevoelig is voor spiegelingen en verwisseling van de assen.

Over deze projectie kunnen een aantal zaken worden opgemerkt. Ten eerste kan uit de scree plot opgemaakt worden dat de oorspronkelijke 11 dimensionale dataset twee intrinsieke dimensies bevat. Van de variantie in de dataset wordt 65 % in de 2 dimensionale plot weergegeven. Dat houdt in dat de projectie een betrouwbaar beeld geeft van de originele dataset en dat de assen van de projectie de verborgen variabelen zullen representeren.

We gaan nu eerst een interpretatie aan de assen van de projectie pogen



Figuur 7.10: Projecties gelabeld op variabelen *intmvt* en *pbermc0*

te geven door het labelen op de oorspronkelijke variabelen die meedoen in de projectie. Vervolgens worden nog wat algemene conclusies getrokken. We beginnen bij de horizontale as.

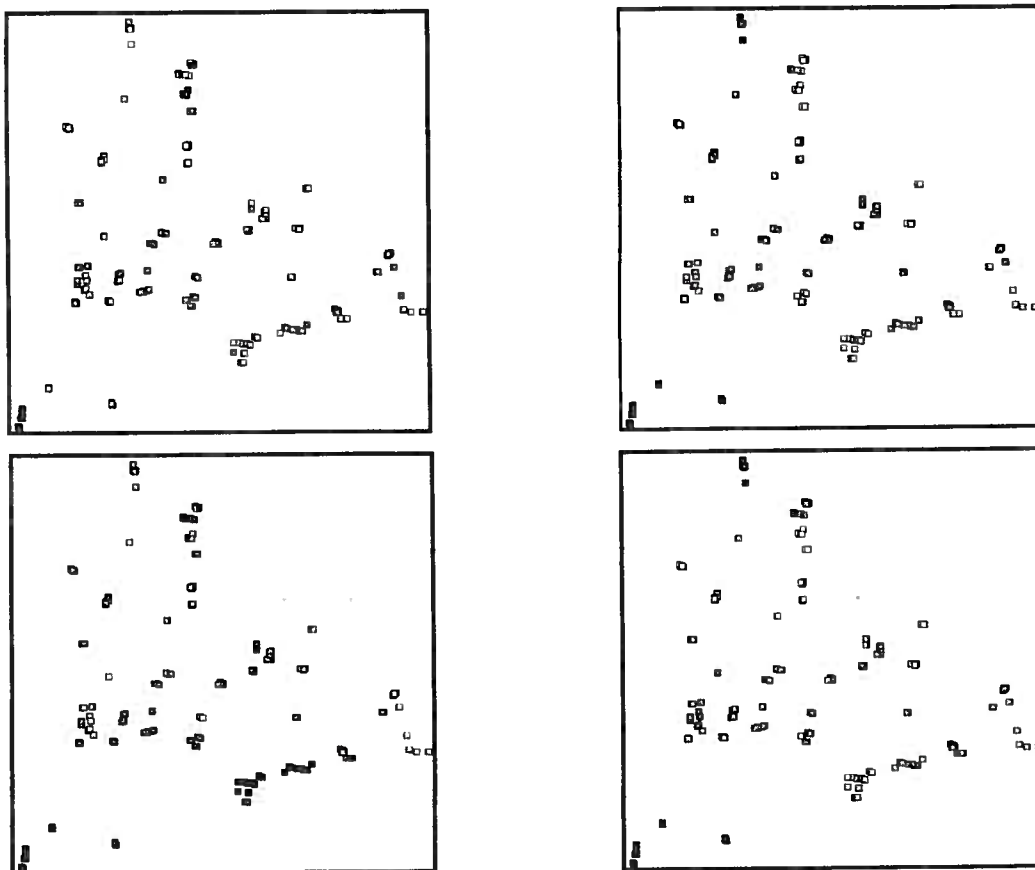
Als we de punten in de projectie labelen op de diverse variabelen, dan valt op dat het verloop van twee variabelen sterk samenvalt met de horizontale as in de projectie. Dit zijn 'intmvt' (verkeersintensiteit in voertuigen per etmaal) en 'pbermc0' (percentage van obstakel-vrije bermen). Hierbij geldt dat 'intmvt' toeneemt naar rechts toe, terwijl 'pbermc0' duidelijk afneemt, omdat hier de blauw-tinten aan de rechterzijde overheersen. De plots gelabeld op deze variabelen zijn weergegeven (in zwart wit) in figuur (7.10). Helaas is in deze plot het kleur-effect niet goed te zien. De gebruiker wordt dan ook aangeraden het voorbeeld zelf te herhalen.

Denkt men na over deze combinatie van variabelen dan komt men al snel tot de conclusie dat het verband dat m.b.v. Neural Vision te zien is erg logisch is: bij een hoge verkeersintensiteit zullen veel borden, stoplichten, praatpalen, verkeersgeleidingssystemen, etc. zorgen voor veel obstakels in de berm, waardoor variabele 'pbermc0' laag uit zal vallen. De horizontale dimensie drukt dus 'drukke van de weg in voertuigen en objecten' uit!

De interpretatie van de verticale as verloopt op een zelfde manier. Op deze as blijken vier variabelen te overheersen. Deze variabelen zijn

lberm : lengte bermen in kilometers

opkruid : oppervlakte kruidachtige vegetatie in ha.



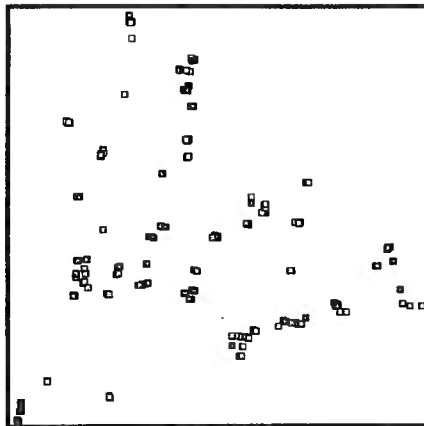
Figuur 7.11: ICA projecties gelabeld op variabelen lberm (linksboven), oppkruid (rechtsboven), opphout (linksonder) en oppberm (rechtsonder).

opphout : oppervlakte houtachtige vegetatie in ha.

oppberm : oppervlakte bermen, water-gangen en waterpartijen.

Ook hier is eenvoudig in te zien waarom deze variabelen in één dimensie terecht komen in de projectie: ze zijn allemaal representatief voor hetzelfde begrip : grootte van de berm. Deze verborgen dimensie representeert dus de grootte van de berm en die neemt toe naarmate een punt zich verder boven in de plot bevindt. De ICA projecties gelabeld op deze variabelen zijn te zien in figuur (7.11).

Nu we de dimensies in de plot hebben benoemd is het interessant om wat



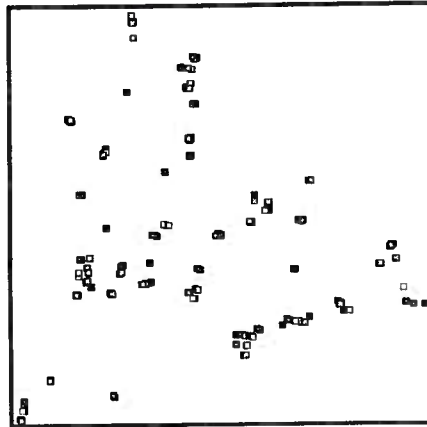
Figuur 7.12: ICA projecties gelabeld op variabele dienstkri

verder naar de dataset te kijken. Zo springt meteen in het oog dat bij wegen met grote 'drukke' de spreiding in lengte kleiner is dan bij wegen met kleine 'drukke'. Echt interessant wordt het wanneer er labelen op de tekst variabele 'dienstkri' (dienstkring). Dit heeft het beeld tot gevolg dat is weergegeven in figuur (7.12).

Hoewel de dienstkring-informatie niet in de projectie meegenomen wordt zien we dat iedere dienstkring een klein groepje puntjes in de projectie vormt, waarbij ieder punt een meting tijdens één jaar representeert. We kunnen uit deze projectie eenvoudig nabijheidsrelaties tussen dienstkringen en eigenschappen van dienstkringen bestuderen (hiervoor is het wellicht makkelijk de kleuren tijdelijk door symbolen te vervangen).

Zo zien we bijvoorbeeld dat de dienstkringen Amsterdam (asd), Haarlem (hrl), Dordrecht (dor) en Delft (del) in de plot dicht bij elkaar terecht komen, hetgeen betekent dat de in de eigenschappen 'grootte van de berm' en 'drukke van de weg' erg op elkaar lijken: de grootte van de berm is middelmatig, de drukke van de weg is hoog! De dienstkringen Deltakust (dk) en Zeeuws Vlaanderen (zv) scoren lage waarden op deze beide dimensies. De dienstkringen Friesland (fri), Drenthe (dre), 'gln' (Geleen) en 'aie' hebben gemeen dat ze grote bermen hebben en een lage verkeersintensiteit.

Labelen we vervolgens op 'jaar' (figuur 7.13) dan zien we dat in veel dienstkringen de grootte van de bermen de laatste twee jaar aanzienlijk is toegenomen!



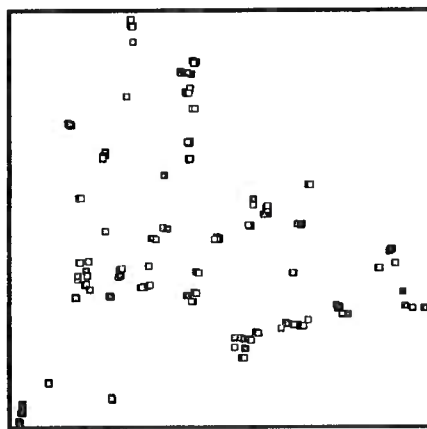
Figuur 7.13: ICA projecties gelabeld op variabele jaar

Deze dataset is oorspronkelijk bedoeld als regressie-dataset waarbij 'budgetlm' de afhankelijke variabele was. Omdat budgetlm niet is meegenomen in de analyse kunnen we deze helaas niet gebruiken voor labelling. Wel kunnen we de tekst-variabele 'budgetdisc' gebruiken voor het labelen van de projectie. Dit heeft het beeld in figuur 7.14 tot gevolg.

Uit deze labeling kunnen we de haast triviale conclusie trekken dat de budgetten voor landschap en milieu hoger neigen te worden naarmate de bermen groter en de wegen drukker worden. Een interessantere conclusie is dat dienstkring Haarlem (hrl) die erg veel op dienstkringen Amsterdam (ams) en Delft (del) lijkt een aanzienlijk lager budget heeft! Kennelijk spelen hier ander factoren dan grootte van de berm en drukte van de weg een rol!

Hoewel uit deze dataset nog meer conclusies getrokken kunnen worden, stoppen we hier met de analyse. We hopen de lezer een indruk te hebben gegeven van de gebruiksmogelijkheden van Neural Vision.

Voor toepassingen als het in kaart brengen van veranderingsprocessen binnen organisaties, wordt verwezen naar appendix C. Voor dit soort toepassingen kan de tijdsanalyse (beschreven in hoofdstuk 5) een belangrijk hulpmiddel zijn.



Figuur 7.14: ICA projecties gelabeld op tekst-variabele budgetdisc

Hoofdstuk 8

Gebruikte technieken

8.1 Het Sanger neurale netwerk

In deze sectie zal het Sanger neurale netwerk beschreven worden. Het Sanger neurale netwerk voert een transformatie op de data uit die bekend staat als principale componenten analyse (PCA). Voordat de werking van het neurale netwerk wordt toegelicht wordt aandacht besteed aan PCA.

8.1.1 Principale componenten analyse

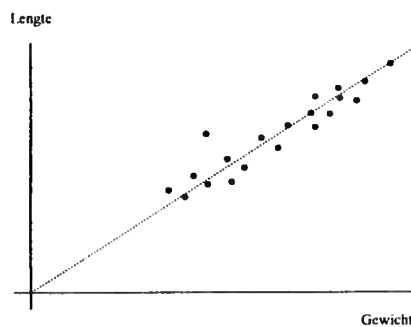
Een veel voorkomend probleem binnen de data-analyse is het reduceren van de dimensionaliteit van data. Gegeven een verzameling hoogdimensionale data, poogt men bij dimensionaliteitsreductie een transformatie te vinden waarmee zoveel mogelijk van de intrinsieke informatie in de data in zo min mogelijk variabelen wordt gestopt. Formeler gezegd komt dit op het volgende neer: stel dat we een p -dimensionale vector \mathbf{x} hebben die we willen reduceren tot een m -dimensionale vector, met $m < p$. We zoeken een methode die dit doet met behoud van zoveel mogelijk informatie in de m -dimensionale vector. Bij PCA wordt de optimale lineaire transformatie \mathbf{T} uitgevoerd om dit doel te bereiken. Optimaal betekent hier dat de fout bij terugtransformatie van de m dimensionale vectoren naar p -dimensionale vectoren minimaal is. Dit impliceert dat de m dimensies zoveel mogelijk van de variantie in de originele data bevatten.

Een principale-componententransformatie is eigenlijk niets anders dan het roteren van het originele assenstelsel waarin de data is ingebed. De nieuwe

assen (vanaf nu principale-componentenassen genoemd) staan loodrecht op elkaar en verklaren zoveel mogelijk variantie in zo weinig mogelijk dimensies.

Principale componenten analyse is ook bekend onder de naam Karhunen-Loèvetransformatie. Het is een van de bekendste en oudste technieken binnen de multivariate data-analyse. PCA is voor het eerst geïntroduceerd in 1901 door Pearson [Pea01] in een biologische context. Later is het onafhankelijk daarvan herontdekt door Karhunen in 1947 [Kar47], en gegeneraliseerd door Loève in 1963 [Loe63].

We geven een voorbeeld van PCA aan de hand van figuur (8.1). Hierin zien we van een aantal personen lengte en gewicht tegen elkaar uitgezet. Tussen deze twee dimensies is echter een duidelijk verband, zodat de variantie zich hoofdzakelijk in een richting bevindt. Als we nu de data op deze richting afbeelden blijft veel van de in de data aanwezige informatie behouden. Als we multidimensionale data op twee dimensies afbeelden kunnen we deze multidimensionale data vervolgens visualiseren.

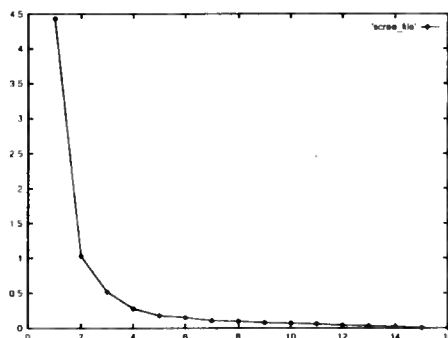


Figuur 8.1: Voorbeeld van een PCA-toepassing. De meeste variantie in de dataset bevindt zich in de richting van de gestippelde lijn, ofwel: de gestippelde lijn is de eerste principale component.

Een vraag die zich opwerpt bij dimensionaliteitsreductie is: hoeveel dimensies hebben we nodig om de data te beschrijven. Doorgaans wordt dit probleem opgelost door middel van een zgn. scree-plot. Hierin wordt voor iedere principale component de hoeveelheid variantie die deze verklaart tegen zijn nummer uitgezet. Van links naar rechts krijgt men op deze manier een dalende grafiek te zien, omdat de eerste principale component altijd meer variantie zal bevatten dan de tweede, deze weer meer dan de derde, etc. In

veel gevallen zal er een elleboog te zien zijn in deze grafiek. Vanaf deze elleboog wordt alleen nog maar ruis gemodelleerd door de principale componenten. Het gebruik bij dimensionaliteitsreductie is dan ook om als nieuw aantal dimensies het aantal te kiezen waarbij in de grafiek zich de elleboog bevindt.

Een voorbeeld van een scree-plot is te zien in figuur (8.2). Het betreft hier een scree-plot gemaakt na een PCA-transformatie van de 16-dimensionale 'zoo'-dataset. Dit is een 16-dimensionale dataset die 101 samples bevat. Uit de scree-plot is te zien dat er twee à drie dimensies nodig zijn voor een accurate representatie van deze 16-dimensionale data.



Figuur 8.2: Voorbeeld van een scree-plot. Dit is een scree-plot van de 'zoo'-dataset. Het benodigde aantal dimensies is hier 2 à 3

8.1.2 Het Sanger neurale netwerk

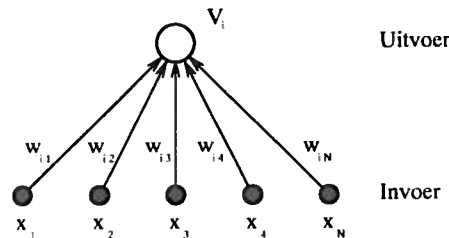
Nu verduidelijkt is wat PCA doet, kunnen we de werking van het Sanger-netwerk uitleggen.

Stel dat we een n -dimensionale dataset tot onze beschikking hebben, en daarop willen we een PCA uitvoeren met behulp van een Sanger neurale netwerk. Hiertoe dienen we eerst een Sanger-netwerk te creëren. Een Sanger-neuraal netwerk bestaat uit een aantal knopen. Een dergelijk neuron is weergegeven in figuur (8.3). We zien een aantal zwarte rondjes, waarop de invoer patronen (x) aangeboden gaan worden, een aantal verbindingen waarmee gewichten w zijn geassocieerd en een wit rondje dat de uitvoer V van het neuron gaat berekenen op een manier zoals hier onder weergegeven.

Bij de inputverbindingen van de neuronen wordt een N -dimensionale inputvector $\mathbf{x} = (x_1, \dots, x_N)$ uit de dataset aangeboden. Vervolgens gaat ieder neuron een outputsignaal afgeven, dat bepaald is door zijn activatiefunctie (ook wel transferfunctie genoemd). In het geval van Sangernetwerken is dit een lineaire activatiefunctie. Deze functie neemt een gewogen som van alle invoerwaarden die het neuron ontvangt. De output V_i van neuron i wordt dus gegeven door

$$V_i = \sum_{j=1}^N w_{ij} x_j. \quad (8.1)$$

Hierin is $\mathbf{w}_i = (w_{i1}, \dots, w_{iN})$ de gewichtenvector van het neuron i . Dit neuron kan N -dimensionale inputvectoren aan.



Figuur 8.3: Een neuron uit een Sangernetwerk

In een Sangernetwerk worden de patronen uit de dataset één voor één aangeboden aan het neurale netwerk. Iedere keer dat een patroon is aangeboden, worden de gewichten van het netwerk aangepast. Dit heet het leren van het netwerk. Dit leren gaat middels de volgende leerregel:

$$\Delta \mathbf{w}_{ij} = \eta V_i \left(x_j - \sum_{k=1}^i V_k \mathbf{w}_{kj} \right) \quad (8.2)$$

Hierin is η de leersnelheid van het neurale netwerk. Na het vele malen aanpassen van de gewichten zal het i -de neuron uit het neurale netwerk representatief worden voor de i -de principale component. De output van het i -de neuron als inputpatroon μ wordt aangeboden bij de inputlaag is de coördinaat van patroon μ op de principale component i . Bij data visualisatie op een tweedimensionaal vlak hoeven we alleen maar de outputs van de eerste twee neuronen te gebruiken; Deze bepalen de coördinaten van de patronen op het vlak opgespannen door de eerste twee principale componenten.

8.2 Wiskundige beschrijving van PCA en het Sanger netwerk

In deze sectie geven we een formelere beschrijving van PCA en bewijzen we dat het Sanger neurale netwerk een principale componenten analyse uitvoert. De sectie heeft de volgende indeling. Eerst wordt aangetoond dat de principale componenten gelijk zijn aan de eigenvectoren van de covariantiematrix van de data. Vervolgens wordt aangetoond dat door middel van het leerproces de gewichtenvectoren van het Sanger neurale netwerk loodrecht op elkaar gaan staan en convergeren naar de richtingen van de eigenvectoren van de covariantiematrix.

Laat \mathbf{x} een p -dimensionale random vector die de data set representeert. We nemen aan dat \mathbf{x} een gemiddelde van 0 heeft:

$$E[\mathbf{x}] = 0 \quad (8.3)$$

met E de statistische verwachtingsoperator. Als \mathbf{x} een gemiddelde heeft ongelijk aan 0, trekken we eerst de gemiddelde waarde van \mathbf{x} af alvorens verder te gaan.

Laat \mathbf{C} de covariantiematrix, bij data met een gemiddelde van 0 gedefiniëerd door

$$\begin{aligned} \mathbf{C} &= E[\mathbf{x}\mathbf{x}^T] \\ &= \langle \mathbf{x}\mathbf{x}^T \rangle \end{aligned} \quad (8.4)$$

Laat \mathbf{a} nu een willekeurige vector ter lengte 1, waarop \mathbf{x} geprojecteerd gaat worden. Deze projectie Y wordt gegeven door

$$Y = \mathbf{x}^T \mathbf{a} = \mathbf{a}^T \mathbf{x} \quad (8.5)$$

onder de randvoorwaarde

$$\|\mathbf{a}\| = 1 \quad (8.6)$$

Omdat E een lineaire operator is, volgt dat voor de projectie Y geldt

$$E[Y] = \mathbf{a}^T E[\mathbf{x}] = 0 \quad (8.7)$$

Hieruit volgt dat de variantie σ^2 van Y gelijk is aan

$$\begin{aligned} \sigma^2 &= E[Y^2] \\ &= E[(\mathbf{a}^T \mathbf{x})(\mathbf{x}^T \mathbf{a})] \\ &= \mathbf{a}^T E[\mathbf{x}\mathbf{x}^T] \mathbf{a} \\ &= \mathbf{a}^T \mathbf{C} \mathbf{a} \end{aligned} \quad (8.8)$$

We kunnen nu de principale componenten verder beschrijven. We brengen in herinnering dat de principale componenten loodrecht op elkaar staande vectoren zijn die zoveel mogelijk variatie bevatten. Dientengevolge kunnen we de eerste principale component Y_1 definiëren als de projectie

$$Y_1 = \mathbf{a}_1^T \mathbf{x} \quad (8.9)$$

waarvoor geldt dat de coëfficiënten \mathbf{a}_1 zo gekozen zijn dat de variantie $\sigma^2 = \mathbf{a}_1^T \mathbf{C} \mathbf{a}_1$ maximaal is onder de randvoorwaarde

$$\mathbf{a}_1^T \mathbf{a}_1 = 1 \quad (8.10)$$

Het maximaliseren van σ^2 onder deze randvoorwaarde kan met behulp van de multiplicatorenstelling van Lagrange worden herschreven tot een ander maximalisatieprobleem, t.w. het maximaliseren van

$$\mathbf{a}_1^T \mathbf{C} \mathbf{a}_1 - l_1 (\mathbf{a}_1^T \mathbf{a}_1 - 1) \quad (8.11)$$

met l_1 een constante die de (onbepaalde) Lagrange-multiplicator genoemd wordt. Door middel van differentiaalrekening kunnen we dit probleem herschrijven tot

$$\mathbf{C} \mathbf{a}_1 = l_1 \mathbf{a}_1 \quad (8.12)$$

Dit is een eigenwaarde-probleem. De oplossing hiervoor wordt verkregen door voor \mathbf{a}_1 de eigenvectoren van \mathbf{C} te nemen met de grootste eigenwaarde l_1 . De eigenwaarde is dan meteen de variantie van bijbehorende principale component. Dit kan men zien door beide leden van (8.12) links te vermenigvuldigen met \mathbf{a}_1^T .

De tweede principale component Y_2 moet weer een maximale variantie hebben, maar onder twee randvoorwaarden: hij moet loodrecht liggen op de eerste principale component en lengte 1 hebben. We zoeken dus

$$Y_2 = \mathbf{a}_2^T \mathbf{x} \quad (8.13)$$

waarvoor geldt dat de coëfficiënten \mathbf{a}_2 zo gekozen zijn dat de variantie $\sigma^2 = \mathbf{a}_2^T \mathbf{C} \mathbf{a}_2$ maximaal is onder de randvoorwaarden

$$\mathbf{a}_2^T \mathbf{a}_2 = 1 \quad (8.14)$$

en

$$\mathbf{a}_2^T \mathbf{a}_1 = 0 \quad (8.15)$$

Wederom gebruik makend van Lagrange-multiplicatoren, differentiaalrekening en algebra, blijkt dat \mathbf{a}_2 gegeven wordt door de eigenvector van \mathbf{C} met de 2^e grootste eigenwaarde.

In het algemeen zoeken we voor de j -de principale component naar een lineaire combinatie $Y_j = \mathbf{a}_j^T \mathbf{x}$ welke loodrecht staat op alle eerdere principale componenten (i.e. $\mathbf{a}_j^T \mathbf{a}_i = 0 \quad \forall i < j$) en welke een maximale variantie heeft. Met behulp van Lagrange-multiplicatoren kan worden aangetoond dat \mathbf{a}_j de j -de grootste eigenvector van \mathbf{C} is.

Nadat we het begrip principale component hebben verduidelijkt gaan we verder met het bewijzen dat de gewichtenvectoren van het Sangernetwerk convergeren naar de principale componenten richtingen. Dit bewijs bestaat eigenlijk uit twee delen. In het eerste deel bewijzen we dat de gewichtenvectoren van een geconvergeerd Sanger netwerk loodrecht op elkaar staan (net als de principale-componentenrichtingen). In het tweede deel bewijzen we dat de gewichtenvectoren in de richtingen met maximale variantie van de data wijzen. Het bewijs verloopt via inductie.

Merk op dat de gemiddelde verandering in de gewichten gegeven wordt door

$$\begin{aligned} \langle \Delta w_{ij} \rangle / \eta &= \left\langle \sum_p w_{ip} x_p x_j - \sum_p w_{ip} x_p \sum_{k=1}^i \sum_q w_{kq} x_q w_{kj} \right\rangle \quad (8.16) \\ &= \sum_p w_{ip} C_{pj} - \sum_{k=1}^i \left[\sum_{pq} w_{kq} C_{pq} w_{ip} \right] w_{kj} \\ &= \mathbf{C} \mathbf{w}_i - \sum_{k=1}^{i-1} \left[\mathbf{w}_k^T \mathbf{C} \mathbf{w}_i \right] \mathbf{w}_k - \left[\mathbf{w}_i^T \mathbf{C} \mathbf{w}_i \right] \mathbf{w}_i \end{aligned}$$

Neem nu aan dat de gewichtenvectoren $1, 2, \dots, i-1$ reeds geconvergeerd zijn naar de corresponderende eigenvectoren, dus $\mathbf{w}_k = \pm \mathbf{C}^k$ voor $k < i$. Omdat in het algemeen geldt dat $\mathbf{x} - (\mathbf{y}^T \mathbf{x}) \mathbf{y}$ de projectie is van \mathbf{x} op de ruimte loodrecht op een eenheidsvector \mathbf{y} , kunnen vgl. (8.16) herschrijven als

$$\langle \Delta \mathbf{w}_i \rangle / \eta = (\mathbf{C} \mathbf{w}_i)^\perp - \left[\mathbf{w}_i^T \mathbf{C} \mathbf{w}_i \right] \mathbf{w}_i \quad (8.17)$$

met $(\mathbf{C} \mathbf{w}_i)^\perp$ de projectie van $\mathbf{C} \mathbf{w}_i$ op de deelruimte orthogonaal met de eerste $i-1$ eigenvectoren.

Stel nu dat \mathbf{w}_i een component heeft die niet in de deelruimte loodrecht op de eerste $i-1$ eigenvectoren ligt. Voor deze component geldt dat de eerste term van het rechterlid van (8.17) gelijk is aan nul, terwijl de tweede term

ervoor zorgt dat de bedoelde component langzaam wordt afgebouwd naar nul. Hierdoor zal \mathbf{w}_i in de deelruimte loodrecht op $\mathbf{w}_1, \dots, \mathbf{w}_{i-1}$ komen te liggen. Nu moeten we nog bewijzen dat de i -de gewichtenvector convergeert naar de i -de eigenvector. In de deelruimte loodrecht op de gewichtenvectoren $1, 2, \dots, i-1$ (waar \mathbf{w}_i dus in komt te liggen) reduceert (8.16) tot

$$\langle \Delta \mathbf{w}_i \rangle / \eta = \mathbf{C} \mathbf{w}_i - [\mathbf{w}_i^T \mathbf{C} \mathbf{w}_i] \mathbf{w}_i \quad (8.18)$$

omdat $\mathbf{w}_k^T \mathbf{C} \mathbf{w}_i = \mathbf{w}_i^T \mathbf{C} \mathbf{w}_k = \mathbf{w}_i^T l_k \mathbf{w}_k = 0$. Vergelijking (8.18) heeft de vorm van Oja's leerregel, waarvoor bewezen kan worden dat de gewichten convergeren naar de maximale eigenvector (zie voor dit bewijs [HKP91] pp. 203-204). Met andere woorden: de gewichten van neuron i zullen convergeren naar de i -de eigenvector van de covariantiematrix \mathbf{C} .

8.3 Het ICA neurale netwerk

ICA is een afkorting van 'Independent Component Analysis'. Independent component analysis is recent ontwikkeld als een uitbreiding op Principal Component Analysis (PCA). Independent Component Analysis wordt in de literatuur ook wel 'Non-linear PCA' genoemd. De eerste publikatie op het gebied van ICA is [HJA85]. Andere referenties zijn [Com94, OKH97, KOW⁺97]. Er bestaan diverse algoritmen voor ICA. Het algoritme dat in Neural Vision is geïmplementeerd is afkomstig uit [KOW⁺97].

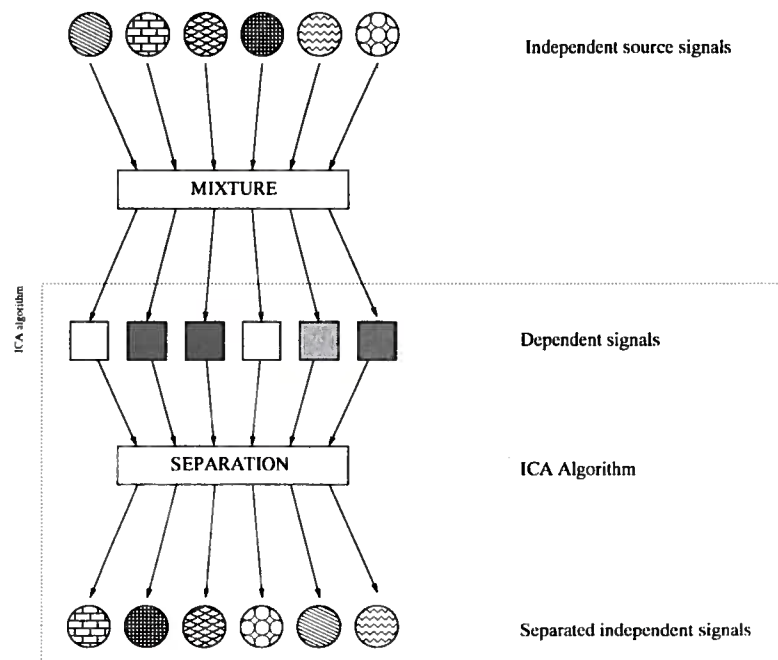
Het ICA neurale netwerk is het tweede netwerk dat in Neural Vision is geïmplementeerd voor dataprojectie. Dit type netwerk zal op twee manieren worden belicht. In sectie (8.3.1) zal een informele uitleg aan de hand van een zeer eenvoudig voorbeeld gegeven worden. In sectie (8.3.2) wordt een meer gedetailleerde en wiskundige uitleg van het ICA-netwerk gegeven.

8.3.1 Algemene uitleg en voorbeelden

In deze sectie wordt eerst in niet-technische termen uitgelegd wat ICA is. Vervolgens worden enkele voorbeeldtoepassingen van ICA uit de literatuur gegeven.

Uitleg ICA

Independent component analysis poogt een vector met afhankelijke toevalsvariabelen te transformeren naar een vector met onafhankelijke toevalsvari-

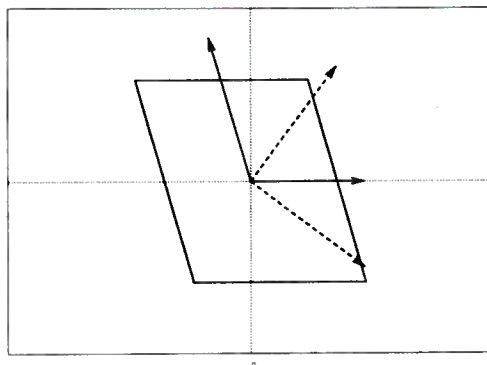


Figuur 8.4: Schematische weergave van toepassing ICA-algoritme. Zie tekst voor meer uitleg.

abelen. Hierbij wordt de aanname gedaan dat iedere afhankelijke toevalsvariabele een lineaire combinatie is van de onafhankelijke toevalsvariabelen. Independent component analysis is nuttig omdat in de praktijk verschillende variabelen die men meet meestal niet onafhankelijk zijn, terwijl men wel in die onafhankelijke variabelen is geïnteresseerd.

Het idee achter ICA is schematisch weergegeven in figuur (8.4). Boven in deze figuur zien we een aantal onafhankelijke variabelen weergegeven als rondjes. Deze variabelen worden middels een 'mixture'-proces met elkaar 'vermengd', waarna een aantal afhankelijke variabelen resulteren (weergegeven als vierkantjes). De taak van een ICA algoritme is het terugvinden van de onafhankelijke variabelen uit hun mixture. Dit is weergegeven in het onderste deel van de figuur. De gevonden onafhankelijke componenten kunnen in volgorde verschillen van de oorspronkelijke onafhankelijke variabelen.

Bij PCA zoekt men een beperkt aantal richtingen in de data die zoveel mogelijk van de variantie in de data verklaren. Bij ICA zoekt men een aantal



Figuur 8.5: De theoretische basisvectoren van ICA (ononderbroken lijn) en PCA (onderbroken lijn). In dit voorbeeld zijn de datavectoren uniform verdeeld in het parallellogram.

richtingen in de data die statistisch onafhankelijk zijn. Een eenvoudig voorbeeld van Independent Component Analysis is gegeven in figuur 8.5. In deze figuur zien we een parallellogram waarin de data uniform verdeeld is. Klaarblijkelijk zijn de twee richtingen in deze data niet statistisch onafhankelijk. Immers, de verdeling van y is afhankelijk van de waarde van x . In deze figuur zijn de projectierichtingen van zowel PCA (onderbroken lijn) als ICA (ononderbroken lijn) weergegeven. Het is duidelijk dat de ICA projectierichting de data beter beschrijft.

Voorbeelden ICA

Om de lezer meer vertrouwd te maken met ICA worden hieronder enkele voorbeelden van ICA toepassingen afkomstig uit de literatuur gegeven.

Verwerking van ECG signalen In deze toepassing, beschreven in [BMO98] worden ECG signalen verwerkt. ECG signalen zijn doorgaans ‘vervuild’ met signalen die gegenereerd worden door ademhaling, elektroden etc. De medicus is hierin niet geïnteresseerd en zij bemoeilijken interpretatie. Aangezien het hier allemaal onafhankelijke signalen betreft kan men ICA gebruiken om ze van elkaar te scheiden.

Scheiden van beelden uit hun mixture In deze toepassing, afkomstig uit [Hyv97], worden originele beelden teruggevonden uit hun mixtures.

Een voorbeeld ziet men in figuur 8.6. In deze figuur zijn boven vier mixtures van beelden weergegeven. Onder zijn de beelden te zien die door een ICA neuraal netwerk geëxtraheerd zijn.

Scheiden van geluiden Men kan met een ICA algoritme verschillende geluiden die door elkaar heen zijn opgenomen (bijvoorbeeld twee mensen die door elkaar heen praten) van elkaar scheiden.

8.3.2 Wiskundige uitleg

In deze sectie wordt een wat gedetailleerdere uitleg gegeven van ICA. Deze uitleg volgt de lijn van [KOW⁺97]. In ICA wordt aangenomen dat er M bronsignalen zijn met een gemiddelde van 0, gegeven door $s_k(1), \dots, s_k(M)$, $k = 1, 2, \dots$. Deze bronsignalen zijn scalair van aard en zijn onderling statistisch onafhankelijk voor elke k . Formeel houdt de onderlinge onafhankelijkheid in dat het produkt van de marginale kansdichtheden van de individuele bronnen gelijk is aan de gezamenlijke (multivariate) kansverdeling van de individuele bronnen:

$$p[s_k(1), \dots, s_k(M)] = \prod_{i=1}^M p_i[s_k(i)]. \quad (8.19)$$

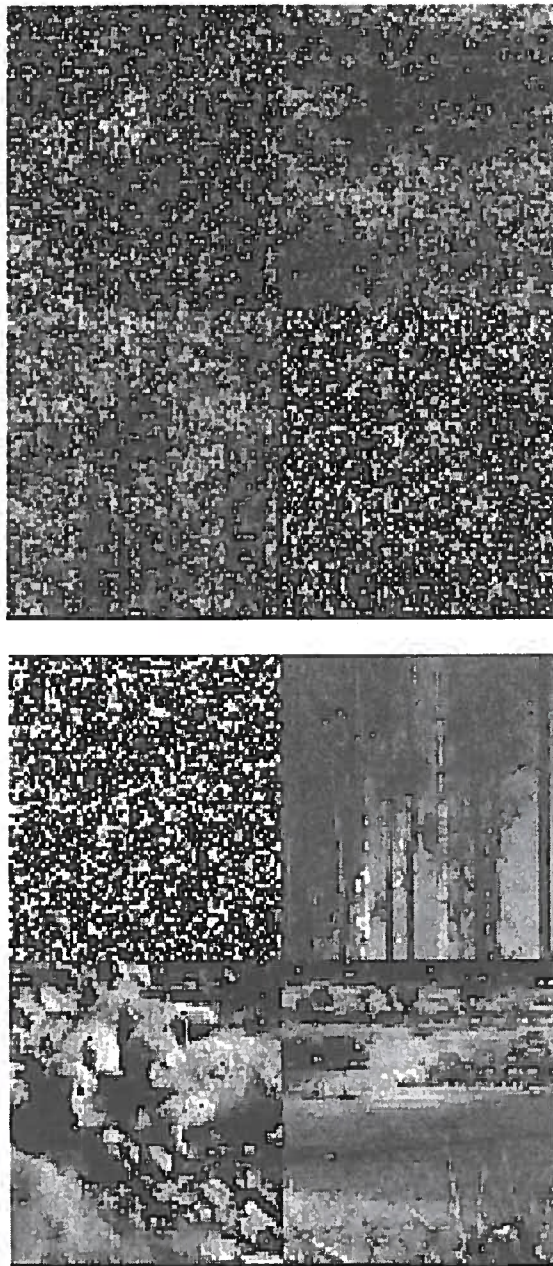
We nemen aan dat de originele bronnen niet te observeren zijn maar dat we slechts over een verzameling ruizige lineaire mixtures $x_k(1), \dots, x_k(M)$ beschikken, met

$$x_k(j) = \sum_{i=1}^M s_k(i)a(ij) + n_k(j). \quad (8.20)$$

In vectorvorm luidt deze vergelijking

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \sum_{i=1}^M s_k(i)\mathbf{a}(i) + \mathbf{n}_k \quad (8.21)$$

Hier is $\mathbf{s}_k = [s_k(1), \dots, s_k(M)]^T$ de bronvector die bestaat uit de M bronsignalen, en $\mathbf{A} = [\mathbf{a}(1), \dots, \mathbf{a}(M)]$ is een constante $L \times M$ "mixing matrix". \mathbf{n}_k representeert de ruisvector. Deze ruisvector wordt in de formulering 8.21 vaak achterwege gelaten omdat het gewoonlijk onmogelijk is om ruis van de bronsignalen te onderscheiden.



Figuur 8.6: Voorbeeld-toepassing van ICA: scheiden van beelden

Het probleem bij Independent Component Analysis is het terugvinden van de oorspronkelijke bronsignalen uit hun ruizige mixtures. Dit terugvinden wordt bereikt middels een 'separation matrix' \mathbf{B} , zodat

$$\mathbf{y}_k = \mathbf{B}\mathbf{x}_k \quad (8.22)$$

een schatting wordt voor \mathbf{s}_k (ofwel $\mathbf{y}_k = \mathbf{s}_k$). Het probleem wordt nu het vinden van de matrix \mathbf{B} .

Het vinden van matrix \mathbf{B} kan aanmerkelijk eenvoudiger gemaakt worden door 'pre-whitening' toe te passen op de data alvorens te zoeken naar \mathbf{B} . Dit houdt een transformatie in die de covariantiematrix gelijk maakt aan de eenheidsmatrix. Hierna kan matrix \mathbf{B} orthogonaal genomen worden ($\mathbf{B}\mathbf{B}^T = \mathbf{I}$, ofwel de rijen hebben lengte 1).

Een praktisch probleem bij ICA-algoritmen is dat verificatie van de onafhankelijkheidsconditie 8.19 onmogelijk is omdat de betrokken kansverdelingen onbekend zijn. Men kan hierdoor geen gebruik maken van 'mutual information' en moet met een approximatie werken. In de referentie [Com94] zijn diverse van dergelijke approximaties geïntroduceerd. Deze worden vaak 'contrast functions' genoemd. Een veel gebruikte contrast function is de kurtosis, gedefiniëerd door

$$\text{cum}[s(i)^4] = E\{s(i)^4\} - 3[E\{s(i)^2\}]^2. \quad (8.23)$$

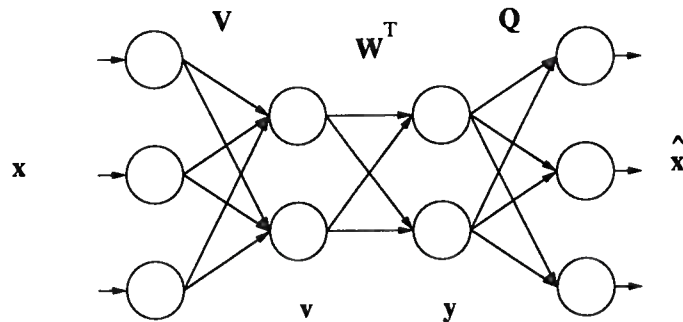
Als $s(i)$ Gaussisch is, dan is zijn kurtosis $\text{cum}[s(i)^4] = 0$. Signalen met een negatieve kurtosis worden vaak 'sub-Gaussisch' genoemd en hebben kansverdelingen die 'platter' zijn dan de Gaussische verdeling. Signalen met een positieve kurtosis worden vaak 'super-Gaussisch' genoemd en hebben kansverdelingen die 'scherper' zijn dan de Gaussische verdeling.

Voor 'pre-whitened' data wordt vaak de volgende contrast function gebruikt

$$J_1(\mathbf{y}) = \sum_{i=1}^M |\text{cum}[y(i)^4]| = \sum_{i=1}^M |E\{y(i)^4\} - 3[E\{y(i)^2\}]^2|. \quad (8.24)$$

Als matrix \mathbf{B} orthonormaal is en de input-vectoren \mathbf{x} zijn pre-whitened dan geldt dat $E\{y(i)^2\} = 1$ zodat $\text{cum}[y(i)^4] = E\{y(i)^4\} - 3$. Nu wordt dus het criterium 8.24 gemaximaliseerd wanneer de som van de vierde-orde momenten

$$J_2(\mathbf{y}) = \sum_{i=1}^M E\{y(i)^4\} \quad (8.25)$$



Figuur 8.7: Structuur van het gebruikte ICA neurale net.

wordt gemaximaliseerd voor bronnen met een negatieve kurtosis en geminimaliseerd voor bronnen met een positieve kurtosis. In de algoritmen die beschreven staan in [KOW⁺97] wordt deze contrastfunctie 8.25 gebruikt.

Beschouw figuur 8.7. In deze figuur is een ICA-netwerk weergegeven dat uit vier lagen met knopen en drie lagen met gewichten bestaat. De activatiefuncties van de knopen zijn lineair. De eerste laag met gewichten (\mathbf{V}) zorgt voor 'pre-whitening' van de data. De tweede laag met gewichten (\mathbf{W}) is de 'separation' layer. Deze laag extraheert de onafhankelijke signalen. De laatste laag met gewichten (\mathbf{Q}) is de 'mixing' layer. Deze laag transformeert de onafhankelijke signalen terug naar de oorspronkelijke, 'vermengde' data. In de figuur representeren de rondjes (knopen) de vectoren die na iedere bewerking resulteren.

Het trainen van het ICA-netwerk vindt plaats in drie fasen. Eerst worden de gewichten (\mathbf{V}) van de pre-whiteninglaag geschat. Vervolgens worden de gewichten (\mathbf{W}) van de separationlaag geschat. Tenslotte worden de gewichten (\mathbf{Q}) van de mixinglaag geschat. Hieronder zal iedere fase in meer detail worden uitgelegd.

Schatten van de pre-whiteninggewichten (\mathbf{V})

Voor deze fase wordt vaak standaard-PCA gebruikt. In dit geval is de pre-whiteningmatrix \mathbf{V} gegeven door

$$\mathbf{V} = \mathbf{D}^{-1/2} \mathbf{E}^T \quad (8.26)$$

met hierin $\mathbf{D} = \text{diag}[\lambda(1), \dots, \lambda(M)]$ een $M \times M$ -diagonaal matrix waarin $\lambda(i)$ de i -de grootste eigenwaarde van de datacovariantiematrix $\mathbf{E}\{\mathbf{x}\mathbf{x}^T\}$ re-

presenteert en $\mathbf{E} = [\mathbf{c}(1), \dots, \mathbf{c}(M)]$ een matrix waarvan de i -de kolom $\mathbf{c}(i)$ de i -de grootste eigenvector van de covariantiematrix $\mathbf{E}\{\mathbf{x}\mathbf{x}^T\}$ representeert.

Schatten van de separationgewichten (\mathbf{W})

Het schatten van deze gewichten gebeurt middels een iteratief algoritme. Dit algoritme is

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k \mathbf{v}_k \mathbf{g}(\mathbf{y}_k^T) + \gamma_k \mathbf{W}_k (\mathbf{I} - \mathbf{W}_k^T \mathbf{W}_k). \quad (8.27)$$

Hierin is k een tijdsindex en zijn μ_k en γ_k twee leerparameters. In NV2 hebben zij waarden van 0.001 en 0.5 respectievelijk. Verder representeert \mathbf{v}_k de ge-pre-whitende versie van \mathbf{x}_k , dus $\mathbf{v}_k = \mathbf{V}\mathbf{x}$. Voorts is $\mathbf{y}_k = \mathbf{W}_k \mathbf{v}_k$ en is $\mathbf{g}(\mathbf{y}_k^T)$ een functie die een niet-lineaire transformatie g toepast op elk element van zijn argument. Deze transformatie is gegeven door $g(t) = t^3$. Er kan worden bewezen (zie hiervoor [KOW⁺97]) dat door deze leerregel de foutfunctie 8.25 geminimaliseerd wordt.

Schatten van de mixing gewichten (\mathbf{Q})

Het schatten van de mixing gewichten \mathbf{Q} is de laatste fase van het ICA trainen. Dit schatten kan men doen door de foutfunctie

$$\mathbf{E}\{\|\mathbf{x} - \mathbf{Q}\mathbf{y}\|^2\} \quad (8.28)$$

te minimaliseren. De gradiënt van $\|\mathbf{x} - \mathbf{Q}\mathbf{y}\|^2$ m.b.t. de matrix \mathbf{Q} is gegeven door $-2(\mathbf{x} - \mathbf{Q}\mathbf{y})\mathbf{y}^T$. Dit leidt tot het gradiënt descent-algoritme

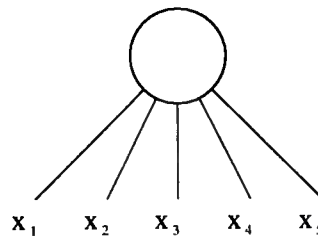
$$\mathbf{Q}_{k+1} = \mathbf{Q}_k + \mu_k (\mathbf{x}_k - \mathbf{Q}_k \mathbf{y}_k) \mathbf{y}_k^T \quad (8.29)$$

voor het leren van matrix \mathbf{Q} . Hierin is μ_k de leersnelheidparameter.

8.4 Het frequency-sensitive competitive learning neurale netwerk

Dit type netwerk is van een andere aard dan de vorige twee typen. Het is niet bedoeld voor dataprojectie, maar voor het uitvoeren van een clusteranalyse, ofwel het geforceerd indelen van de data in een aantal groepen. Het is geïntroduceerd in [MJ95].

Een frequency-sensitive competitive learning neuraal netwerk bestaat uit een aantal neuronen, waarvan er een is weergegeven in figuur (8.8). Ieder van deze neuronen heeft een geassocieerde gewichtenvector met dezelfde dimensionaliteit als de data waarmee het netwerk getraind gaat worden. Deze gewichten krijgen willekeurige beginwaarden.



Figuur 8.8: Een neuron uit een fscnn. Dit neuron is bedoeld voor het clusteren van 5-dimensionale data.

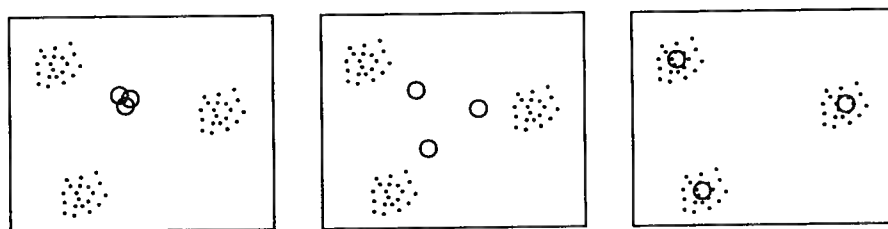
Het trainen van een dergelijk netwerk geschiedt als volgt. Eén voor één worden alle patronen uit de dataset aangeboden, in een willekeurige volgorde. Nadat een patroon \mathbf{x} is aangeboden wordt berekend welke van de neuronen zijn gewichtenvector het dichtst bij het aangeboden patroon heeft liggen (in euclidische zin). Dit neuron heet de *winnaar*, hetgeen verklaart dat dit een *competitive* netwerk genoemd wordt. Als een neuron vaker dan gemiddeld deze competitie wint wordt er bij de daadwerkelijke afstand tot het aangeboden patroon een kunstmatige "straf term" opgeteld. Deze straf term zorgt ervoor dat het neuron een kleinere kans maakt om de "competitie" te winnen. Vanwege deze eigenschap wordt een dergelijk netwerk *frequency-sensitive* genoemd. Er wordt ook gezegd dat dergelijke neuronen een 'geweten' hebben, omdat zij ook collega-neuronen af en toe laten winnen. Het nut van deze maatregel zal hieronder worden verduidelijkt.

Als de winnaar i^* bekend is, wordt de gewichtenvector \mathbf{w}_{i^*} van de winnaar een klein beetje opgeschoven in de richting van het aangeboden patroon. Dit gebeurt met de leerregel

$$\Delta \mathbf{w}_{i^*} = \eta (\mathbf{x} - \mathbf{w}_{i^*}). \quad (8.30)$$

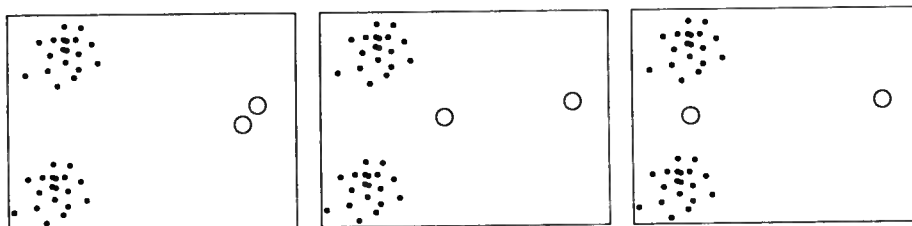
waarin η wederom de leersnelheid is. Zonder bewijs vermelden we hier dat er een geassocieerde kostenfunctie is waarop leerregel (8.30) gradient descent uitvoert.

Het effect van deze trainingsprocedure is geïllustreerd in figuur (8.9). In deze figuur zijn de neuronen van een fscnn-netwerk te zien als drie open rondjes. Dit netwerk bestaat uit drie neuronen. Ook zien we drie dataclusters in de figuur, voorgesteld door de drie wolken met zwarte puntjes. Duidelijk is te zien dat de neuronen de data 'opzoeken'. De eigenschappen van de drie clusters kan men nu bestuderen door te kijken naar de gewichtenvectoren van de neuronen die zich in die clusters bevinden.



Figuur 8.9: Trainingsproces van een fscnn netwerk.

In figuur (8.10) is weergegeven wat er zou gebeuren als een dergelijk netwerk niet "frequency-sensitive" zou zijn. We zien hier dat een van de twee neuronen voortdurend de competitie wint, en dat de het andere neuron ongebruikt blijft.



Figuur 8.10: Trainingsproces waarbij een neuron voortdurend de competitie wint.

8.5 Variabelenselectie met behulp van variantie en entropy

Neural Vision 2.0 biedt de mogelijkheid om numerieke variabelen te selecteren op basis van variantie of entropy. Beide grootheden zijn indicatief voor de hoeveelheid informatie die de numerieke variabele bevat. Hieronder wordt voor zowel variantie als entropy uitgelegd hoe deze uitgerekend worden.

8.5.1 Het uitrekenen van de variantie

De variantie van een variabele x waarvan een sample x_1, \dots, x_N bekend is wordt uitgerekend met behulp van de formule

$$\text{variantie}(x) = \sum_{i=1}^N \frac{(x_i - \bar{x})^2}{(N-1)}. \quad (8.31)$$

Hierin is \bar{x} de gemiddelde waarde van x in het sample.

8.5.2 Het uitrekenen van de entropy

De entropy van een variabele x waarvan een sample x_1, \dots, x_N bekend is wordt als volgt uitgerekend. De variabele wordt eerst gediscrètiseerd in M intervallen $I_j, j = 1, \dots, M$, waarbij $M = \lfloor N/2 \rfloor$. Vervolgens wordt van ieder interval I_j de kans berekend dat $x \in I_j$ als

$$P(x \in I_j) = \frac{|\{x_k | x_k \in I_j\}|}{N}. \quad (8.32)$$

De intervalindices $j = 1, \dots, M$ kan men zien als mogelijke waarden van een discrete toevalsvariabele a , waarbij geldt dat de kans

$$P(a = j) = P(x \in I_j) = \frac{|\{x_k | x_k \in I_j\}|}{N}. \quad (8.33)$$

De entropy van deze discrete toevalsvariabele a wordt nu uitgerekend met behulp van de formule

$$\text{entropy}(a) = - \sum_{j=0}^M P(a = j) \log P(a = j). \quad (8.34)$$

De lezer vraagt zich waarschijnlijk af waarom de variabele x gediscrèteerd dient te worden alvorens de entropie kan worden uitgerekend. Voor een continue variabele x gebruikt men normaal de fomule

$$h(x) = - \int_{-\text{inf}}^{\text{inf}} f(x) \log f(x) dx, \quad (8.35)$$

waarbij $f(x)$ de kansdichtheid van x weergeeft. Omdat deze kansdichtheid onbekend is kan deze fomule niet gebruikt worden.

8.6 De eenvoudige statistieken

In Neural Vision 2.0 is een tabblad ingericht voor eenvoudige beschrijvende statistieken. Deze statistieken zijn :

Gemiddelde van een numerieke variabele x : Deze wordt uitgerekend met de fomule

$$\bar{x} = \frac{1}{N} \sum_i x_i.$$

Standaarddeviatie van een numerieke variabele x : De standaarddeviatie σ_x is de wortel van de variantie en wordt uitgerekend met de fomule

$$\sigma_x = \sqrt{\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{(N - 1)}}.$$

Variatie-coëfficient: Gegeven door de fomule

$$\frac{\sigma_x}{\bar{x}} \times 100\%$$

Gemiddelde binnen een groep: Dit gemiddelde \bar{x}_g voor groep g is berekend door

$$\bar{x}_g = \frac{1}{N_g} \sum_{i=1}^{N_g} x_{ig}$$

waarbij N_g het aantal elementen binnen groep g aangeeft en x_{ig} het i -de element van groep g is.

Variatie binnen een groep: Deze variatie is voor groep g berekend met de formule

$$\frac{1}{N_g - 1} \sum_{i=1}^{N_g} (x_{ig} - \bar{x}_g)^2$$

Variatie-coëfficiënt binnen een groep: Gegeven door de formule

$$\frac{\sigma_{x_g}}{\bar{x}_g} \times 100\%$$

Bibliografie

- [BMO98] A. K. Barros, A. Mansour, and N. Ohnishi. Adaptive blind elimination of artifacts in ecg signals. In *Proceedings of I&ANN*, Spain, 1998.
- [Com94] P. Comon. Independent Component Analysis, a new concept ? *Signal Processing, Elsevier*, 36(3):287–314, April 1994. Special issue on Higher-Order Statistics.
- [Heb49] D. O. Hebb. *The organisation of behavior*. Wiley, New York, 1949.
- [HJA85] J. Herault, C. Jutten, and B. Ans. Détection de grandeur primitives dans un message compsite par une architecture de calcul neuromimétique en apprentissage nonsupervisé. In *Proceedings of the GRETSI conference*, Nice, 1985.
- [HKP91] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, 1991.
- [Hyv97] A. Hyvrinen. One-unit learning rules for independent component analysis. In *Advances in Neural Information Processing System 9 (NIPS*96)*. MIT Press, 1997.
- [Kar47] H. Karhunen. Uber Lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fenn.*, Ser. A.I. 37, 1947. see translation by I. Selin, The Rand Corp., Doc. T-131, 1960.
- [KOW⁺97] J. Karhunen, E. Oja, L. Wang, R. Vigário, and J. Joutsensalo. A class of neural networks for independet component analysis. *IEEE Transactions on Neural Networks*, 8(3):486–504, May 1997.

- [Loe63] M. Loeve. *Probability Theory*. Springer, New York, 3 edition, 1963.
- [MJ95] Jianchang Mao and Anil K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296–317, March 1995.
- [MP43] Warren McCulloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [OKH97] E. Oja, J. Karhunen, and A. Hyvaerinen. From neural principal components to neural independent components. *Lecture Notes in Computer Science*, 1327:519–??, 1997.
- [Pea01] K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2 (sixth series):559–572, 1901.
- [Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

Bijlage A

Lijst met trefwoorden

clusteren: Het in groepen verdelen van data.

dataset: Een verzameling data bestaande uit een aantal metingen (samples). Iedere meting bestaat vaak uit een aantal variabelen.

dialogoog: Window met knoppen, menu's, iconen e.d.

entropy: Maat voor de hoeveelheid informatie die een variabele bevat.

fscnn: Frequency sensitive competitive neural network: netwerktype dat gebruikt kan worden voor het clusteren en comprimeren van data.

gewicht: Getal dat is geassocieerd met een verbinding in een neurale netwerk en dat bepaalt met hoeveel het signaal dat over die verbinding gaat wordt vermenigvuldigd.

ICA: Niet-lineaire projectiemethode die poogt een vector met afhankelijke toevalsvariabelen te transformeren naar een vector met onafhankelijke toevalsvariabelen.

independent component analysis: data transformatie waarbij afhankelijke toevalsvariabelen worden getransformeerd naar onafhankelijke toevalsvariabelen.

knoop: Ander woord voor neuron of unit. In staat tot het uitvoeren van kleine berekeningen met invoer die wordt ontvangen over invoerverbindingen. Het resultaat van deze berekening bepaalt de output van de knoop.

kwalitatieve data: Niet kwantitatieve data. Data die kwalitatieve eigenschappen van objecten beschrijft, bv. vorm, geur etc.

kwantitatieve data: Data die een uitspraak doet over een kwantitatieve eigenschap van een object en op een intervalschaal is gemeten waarop de natuurlijke metriek der getallen geldt. Een kwantitatieve variabele wordt continu genoemd wanneer deze een niet-aftelbaar aantal waarden aan kan nemen.

label: Klasse-informatie die met een sample kan worden geassocieerd. Het aantal mogelijke klassen moet eindig zijn.

leerregel: Regel die gebruikt wordt voor het aanpassen van de gewichten tijdens het leren van een neurale netwerk.

neurale netwerk: Autonoom parallel gedistribueerd adaptief informatieverwerkend systeem, geïnspireerd op de menselijke hersenen.

neuron: Ander woord voor knoop of unit. In staat tot het uitvoeren van kleine berekeningen met invoer die wordt ontvangen over invoerverbindingen. Het resultaat van deze berekening bepaalt de output van het neuron.

nominale data: Kwalitatieve data zonder impliciete ordening (bv. vorm \in { rond, vierkant, driehoekig}).

ordinale data: Kwalitatieve data met impliciete ordening (bv. smaak \in {erg vies, vies, gemiddeld, lekker, erg lekker}).

polygon plot: Visuele weergave van een multidimensionaal patroon door middel van een op een spinnweb gelijkend raderwiel.

principale componenten: Loodrecht op elkaar staande richtingen binnen een dataset die een maximale hoeveelheid variantie verklaren.

projectiemethode: Methode om data op een kleiner aantal dimensies af te beelden.

samann: Niet-lineaire neurale projectiemethode, die poogt de oorspronkelijke afstanden in de data zo goed mogelijk te behouden. Was onderdeel van Neural Vision 1.0.

sample: Een meting (rij, patroon). Meerdere patronen met dezelfde dimensionaliteit kunnen een dataset vormen.

sanger: Lineaire neurale projectiemethode die convergeert naar de principale componenten.

unit: Zie *knoop*.

variabele: Meetbare eigenschap. Voorbeeld: lengte van een persoon. In de context van de Neural Vision data analyse tool zal een variabele vaak numeriek of ordinaal numeriek zijn.

variantie: Maat voor de hoeveelheid informatie die een variabele bevat.

verbinding: Loopt tussen twee neuronen in een neuraal netwerk. Overdracht van informatie tussen de neuronen verloopt via een verbinding.

Bijlage B

Beschrijving van het ASCII dataset file format (*.ndf files)

Het .ndf file format is een zelf gedefiniëerd file format voor dataset bestanden. Ndf bestanden zijn ASCII bestanden, dus zij kunnen eenvoudig bewerkt worden in een text editor of tekstverwerker.

De beschrijving van dit file format gaat het makkelijkst aan de hand van een voorbeeld. Daarom is hieronder is een deel van de file IRIS.NDF weergegeven, die bij de installatie is inbegrepen.

```
nof_patterns 150
nof_inputs 4
nof_labels 1
var_names p_length p_width s_length s_width plant_sort
var_types 0 0 0 0 1
5.1 3.5 1.4 0.2 Iris-setosa
4.9 3.0 1.4 0.2 Iris-setosa
4.7 3.2 1.3 0.2 Iris-setosa
.
.
6.5 2.8 4.6 1.5 Iris-versicolor
5.7 2.8 4.5 1.3 Iris-versicolor
6.3 3.3 4.7 1.6 Iris-versicolor
.
.
6.5 3.0 5.2 2.0 Iris-virginica
6.2 3.4 5.4 2.3 Iris-virginica
5.9 3.0 5.1 1.8 Iris-virginica
```

De eerste vijf regels van deze file bevatten informatie over de dataset:

nof_patterns is een parameter die het aantal samples (ook wel patronen genoemd) in de dataset aangeeft. De iris dataset bevat 150 samples, dus **nof_patterns** is 150.

nof_inputs is een parameter die aangeeft hoeveel numerieke input variabelen (ofwel features) in de dataset aanwezig zijn. Bij de iris dataset zijn dit er 4.

nof_labels is een variabele die aangeeft hoeveel tekstvariabelen (ofwel nominale variabelen) er in de dataset zitten. In het geval van de iris dataset is dat er 1, t.w. de ondersoort waartoe een plantje gerepresenteerd door een sample behoort.

var_names zijn de namen van de variabelen. een naam mag slechts uit de symbolen [a..z] bestaan en geen spaties, tabs, komma's etc. bevatten!

var_types bevat de types van de variabelen: 0 betekent een numerieke waarde, en 1 betekent een nominale waarde. Het aantal nullen en enen dient te sommeren tot de waarden die bij **nof_inputs** en **nof_labels** gegeven zijn.

Onder deze regels met systeemparemeters volgen de eigenlijke data. Voor iedere regel geldt dat eerst een naam van een patroon is gegeven (in het geval van de iris dataset is deze niet aanwezig), vervolgens de invoer variabelen gescheiden door spaties (in het geval van de iris dataset zijn dit er vier) en als laatste de tekstvariabelen, ook gescheiden door spaties (in het geval van de iris dataset is dit er een).

Men kan eenvoudig een ASCII file converteren tot een .ndf file, door zelf de bovenste regels met de juiste bijbehorende waarden in te typen. Men dient er hierbij wel op te letten dat namen van variabelen en tekstvariabelen uit een woord bestaan. Is dit niet het geval, dan worden delen van de namen als waarden van variabelen ingelezen, hetgeen niet wenselijk is. De meest eenvoudige oplossing voor dit probleem is het toevoegen van underscores (.) om de woorden 'aan elkaar te plakken'. Het is niet toegestaan om bijzondere tekens zoals \ en / te gebruiken in een variabelenaam.

Bijlage C

Artificiële Neurale Netwerken

Het in kaart brengen van veranderingsprocessen en het meten van leren

Walter Baets

Leon Brunenberg

C.1 Inleiding

In dit hoofdstuk wordt ingegaan op de filosofie om neurale netwerken te gebruiken in het kader van 'Organisational Learning' projecten. Dit hoofdstuk is al eerder gepubliceerd door Dr. Walter Baets in het Handboek voor Effectief Opleiden. Dit hoofdstuk is bedoeld voor de lezer die bij het gebruik van de tool 'Neural Vision' behoefte heeft aan achtergrond-informatie.

Dit hoofdstuk wil de lezer tonen hoe artificiële neurale netwerken in de praktijk gebruikt kunnen worden voor het in kaart brengen van veranderingsprocessen en het meten van leren. Na een inleiding wordt dieper ingegaan op wat verstaan wordt onder veranderen en leren, hoe neurale netwerken precies werken en waarom zij een interessant alternatief kunnen bieden. Aan de hand van een voorbeeld zal blijken hoe neurale netwerken precies gebruikt worden, hoe een 'kaart' er dan uitziet en hoe men leren meet. Tenslotte worden ook de beperkingen en mogelijkheden van het gebruik van neurale netwerken toegelicht.

Veranderingsprocessen zijn in modern management eerder regel dan uitzondering. Er gaat dan ook heel wat aandacht naar hoe organisaties (con-

tinue) kunnen veranderen en welke rol het management hierbij kan spelen. Kreten als de 'lerende organisatie' (Swieringa en Wierdsma, 1990) en 'lerend management' (Van Den Broeck, 1994) zijn aan de orde van de dag. Het idee, eind jaren zeventig en begin jaren tachtig, dat rationele modellen een verklaring konden geven van de wereld en hoe we die zouden kunnen verbeteren heeft plaats gemaakt voor een misschien wel net zo overdreven opvatting dat de werknemer, omgeven door een minder 'sturende' vorm van management, in een vertrouwenwekkende omgeving uitstijgt tot nieuwe hoogten. Zijn dit de eerste tekenen van een renaissance van het positivistisch mensbeeld?

Het probleem blijft echter wel bij de manager die zijn team moet leiden, die aangeworven wordt voor, of belast wordt met, veranderingsprocessen en die misschien het risico van zelfsturende veranderingsprocessen niet zo ziet zitten. Hiermee wil zeker niet gezegd zijn dat organisaties of processen niet zelf regulerend kunnen zijn. Verderop zal zelfs gesuggereerd worden dat dit nu precies de reden is waarom artificiële neurale netwerken zo interessant zijn, maar dit is voor het ogenblik twee stappen te ver.

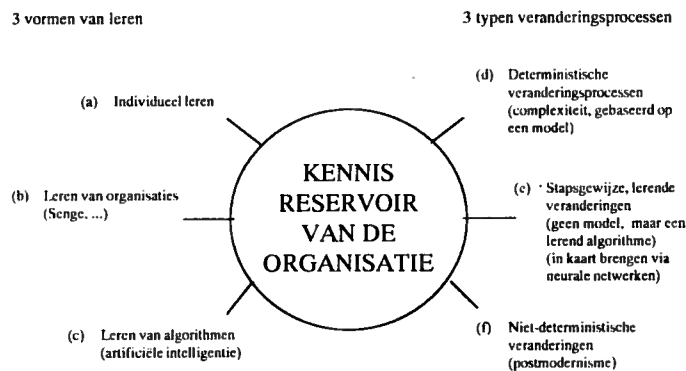
Ongeacht onze overtuiging omtrent de onderliggende redenen voor veranderingsprocessen of het leren op zich, lijkt het voor de hand te liggen dat managers of veranderingsteams best baat kunnen hebben bij tools die hen inzicht geven in het veranderingsproces. Zoals Shakespeare al zei is de perceptie de waarheid; bijgevolg zal men enkel veranderingsprocessen in kaart kunnen brengen via percepties van de betrokken personen, en niet via objectieve waarnemingen. Het falen van de gekende kwantitatieve methoden in zulke omstandigheden heeft de aandacht hiervan een beetje afgewend. Het bedrijfsleven is driftig aan de slag gegaan met veranderingsprocessen, zonder evenwel een duidelijk beeld te hebben (of zelfs maar te betrachten) omtrent het veranderingsproces, laat staan na afloop te kunnen meten of verandering (in percepties) plaats gevonden heeft.

De techniek van artificiële neurale netwerken heeft in de ingenieurswetenschappen al veel successen geboekt. Daarenboven lijkt deze techniek een behoorlijk potentieel te hebben om veranderingsprocessen in kaart te brengen (Baets, 1993; Baets et al., 1995; Venugopal and Baets, 1994). In deze bijdrage wordt getracht inzicht te geven in wat artificiële neurale netwerken nu precies zijn, waar de analogie naar bedrijfsprocessen zit en hoe ze gebruikt kunnen worden voor het in kaart brengen van veranderingsprocessen.

C.2 Leren en veranderen

Hoewel onze maatschappij al eeuwen georganiseerde systemen van leren kent (bijvoorbeeld scholen) blijken we het nog steeds niet echt eens te zijn over hoe we het best leren, laat staan veranderen. Misschien zijn we het zelfs niet echt eens over wat leren nu precies is en op welk organisatieniveau het plaats grijpt. Teneinde een basis te leggen voor een goed wederzijds begrip is enige beschrijving voor deze bijdrage dan ook wenselijk.

Figuur 1 geeft een overzicht hoe leren en veranderen met elkaar te maken zouden kunnen hebben via wat men 'bedrijfsintelligentie' (corporate intelligence) kan noemen.



Figuur C.1: Leren, kennis en veranderen

In figuur 1 komen drie vormen van leren voor: individueel leren, leren van een organisatie en leren van algoritmen, die elk het kennis-reservoir van het bedrijf aanvullen. Vanuit het kennis-reservoir kunnen drie typen veranderingsprocessen aangestuurd worden: deterministische veranderingsprocessen, stapsgewijze en lerende processen, en niet-deterministische processen.

Wellicht is het meeste bekend over individueel leren (a). Gedurende eeuwen zijn daarvoor scholen opgericht, en ook een behoorlijk stuk van de opleidingen binnen bedrijven is gebaseerd op het concept van individueel leren: de pedagogie is een wetenschap op zich.

Reeds enkele jaren oud, doch recent volop in de belangstelling gekomen, zijn de ideeën omtrent het leren op organisatieniveau (b) (Argyris and Schön, 1978). Een doorbraak op dit gebied werd internationaal wellicht bewerkstelligd door Senge (1990) en in Nederland door Swieringa en Wierdsma (1990). Groepen en ondernemingen vertonen kwaliteiten die de groep in staat stelt (en via de groep het individu) meer te leren dan wat een individu zou kunnen leren zonder groep. Het blijft echter de vraag of leren een individueel, dan wel een groepsproces is (Nonaka, 1994).

Tenslotte zijn er recent doorbraken te melden in de theorie en toepassingen van lerende algoritmen (c). Binnen de artificiële intelligentie gaat veel aandacht naar het fenomeen dat 'systemen', van welke aard dan ook, lerend gedrag vertonen. Artificiële neurale netwerken zijn hier slechts een voorbeeld van (Dayhoff, 1990). Experimenten met zulke lerende algoritmen of met zelfregulerende systemen (Morin, 1990a en b) geven heel wat inzicht in de manier waarop systemen niet begeleid leren.

Deze drie vormen van leren (individueel leren, groepsleren, lerende algoritmen) dragen bij tot het creëren van een reservoir van kennis, waarop de onderneming zijn acties kan steunen. Deze vergaarbak wordt wel eens 'corporate intelligence' genoemd: het echte onderscheidende vermogen tussen bedrijf A en bedrijf B. Om die reden hebben sommigen wel eens geschreven dat het vermogen tot continu leren wellicht het enige duurzame competitieve voordeel is dat bedrijven hebben (de Geus, 1988) of, nog verder, dat het vermogen tot het creëren en beheren van kennis het enige duurzame competitieve voordeel zou zijn (Nonaka, 1994).

Zoals reeds gesuggereerd is het deze 'bedrijfsintelligentie' die de basis vormt voor succesvolle acties van een bedrijf, maar dus tevens de basis voor het ondernemen van veranderingsprocessen. Net zoals het creëren van deze bedrijfsintelligentie op verschillende manieren kan gebeuren, zo zijn er ook verschillende veranderingsmodellen bekend. Figuur 1 geeft ook hiervan een overzicht.

Ons klassiek economisch en management denken heeft succes-jaren gekend voor wat men 'wetenschappelijk' management is gaan noemen. Deterministisch, bij voorkeur model-matig denken was de gangbare vorm van het bekijken van bedrijfsprocessen. Economische processen werden vereenvoudigd om gevat te kunnen worden in meestal lineaire, deterministische en niet-dynamische modellen. De realiteit blijkt echter anders in elkaar te zitten. Chaos theorie (Cohen en Stewart, 1994) complexiteitstheorie (Nicolis en Prigogine, 1989) en onderzoek naar onder andere kunstmatig leven (Lang-

ton, 1989) hebben ons inzicht op dit punt drastisch veranderd. Het Santa Fe Institute (Bulletin of the Santa Fe Institute) heeft hierin baanbrekend werk verricht. We hebben nu een beter inzicht in de beperkingen van datgene waarvan we gedurende jaren dachten dat het de oplossing was voor het verbeteren van management. Maar zijn we het nu eens over hoe beter management er uit ziet? Niet echt. Ook hier kunnen we ruwweg drie gedachtengangen herkennen wat betreft veranderingsprocessen.

Allereerst is er veel aandacht voor deterministische (d), maar complexe veranderingsprocessen. Deze onderscheiden zich van het klassieke modelmatige denken door de 'vorm' van de vergelijkingen achter het model. Deze kunnen niet-lineair en dynamisch zijn, waarbij vooral bij dat dynamische karakter gedacht kan worden aan het inplannen van feed-forward mechanismen. De complexiteitstheorie is hiervan een voorbeeld, evenals de systeem dynamica (gelanceerd in de jaren zeventig en nu weer populair) en kennis-systemen. Men gaat er bij deze benadering van veranderingsprocessen van uit dat er een deterministisch veranderingsmodel moet bestaan wat in een of andere vorm kan gevat en beschreven worden door complexe vergelijkingen of regels.

Een tweede kijk op veranderingsprocessen, stapsgewijze of lerende processen (e), aanvaardt nog wel het bestaan van een (deterministisch) model, doch men aanvaardt ook dat het onmogelijk is een gedetailleerd model of een gedetailleerde voorstelling te bereiken. In deze benadering is het belangrijker een werkbaar veranderingstraject te hebben dan het best mogelijke. Men aanvaardt dat verandering stapsgewijs verloopt en dat gedurende het veranderingsproces zelf ook leren (en aanpassen) kan plaats grijpen. Het debat spitst zich hier toe op de vraag of het nu beter is een vereenvoudigde maar gedetailleerde beschrijving van de werkelijkheid te hebben, dan wel een alles omvattende maar minder gedetailleerde voorstelling, die overigens ook nog kan veranderen. Het gebruik van neurale netwerken om deze processen in kaart te brengen en 'lerend' te veranderen is hiervan een toepassing en zal verderop in deze bijdrage in detail bekeken worden.

Een laatste benadering is die van niet-deterministische veranderingsprocessen (f). In deze benadering is er geen model achter de werkelijkheid en ook niet achter veranderingsprocessen en is het dus tijdverlies hiernaar te zoeken. Sommige visies op het leren van organisaties (o.a. Swieringa en Wierdsma) steunen hierop. Hoewel dit een contradictio in terminis lijkt kan men deze veranderingsprocessen wel trachten te (bege)leiden. Deze gedachtschool noemt men dan het postmodernisme (Hassard en Parker, 1993;

Hassard, 1993). In deze bijdrage zal hier verder niet op worden ingegaan.

Samengevat: leren wordt niet opgevat als het opstapelen van bruikbare kennis. Leren is eerder het bruikbaar maken van percepties en ervaringen. Wij weten niet hoe kennis er precies uit ziet, enkel dat er verschillende vormen zijn (passief en actief) en dat kennis wellicht gedistribueerd opgeslagen zit. Bedrijfsintelligentie is een pool van bedrijfskennis die wellicht de motor is achter veranderingsprocessen. Het zou nuttig zijn een idee te hebben van bedrijfsprocessen, zoals die gepercipieerd worden door de belanghebbenden, alvorens ze te veranderen. Maar gezien al het voorgaande is dit niet echt mogelijk indien we onze (klassieke) deterministische denkwijze trachten te gebruiken.

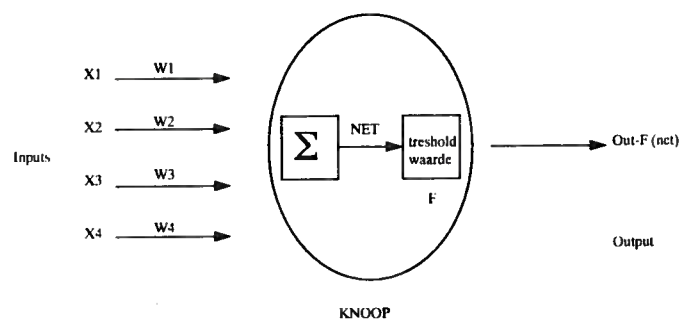
Welke rol kunnen Artificiële Neurale Netwerken (ANN) nu spelen bij deze leer- en veranderingsprocessen? Uit figuur 1 blijkt de centrale rol die de bedrijfsintelligentie speelt en het belang van deze intelligentie voor veranderingsprocessen. Hoewel in deze figuur leren en veranderen eigenlijk artificieel gescheiden zijn is er al eerder gesuggereerd dat in werkelijkheid leren en veranderen dikwijls samengaan. Neurale netwerken zijn interessant wanneer men het bestaan van een deterministisch model aanvaardt, doch ook inziet dat werkbaarheid belangrijker is dan detail. Overigens blijkt het in werkelijkheid zeer moeilijk deterministische (kwantitatieve) modellen te bouwen wanneer de gegevens percepties zijn in plaats van harde gegevens. In veranderingsprocessen gaat het nu juist dikwijls om percepties.

C.3 De werking van Artificiële Neurale Netwerken

Artificiële Neurale Netwerken (ANN) trachten de architectuur van het biologisch neurale systeem (onze hersens) te modelleren. Biologische neurale netwerken zijn samengesteld uit eenvoudige, zeer hecht geconnecteerde 'processors' die we neuronen noemen. De connecties worden gemaakt door de uitgaande takken die we axons noemen. Deze takken vormen dan op hun beurt weer connecties met de andere neuronen en deze connecties noemen we synapses of knopen. Wanneer een bepaald neuron een stimulus ontvangt en wanneer de som van deze stimuli een bepaalde grenswaarde overschrijdt, dan lanceert dit neuron op zijn beurt een stimulus naar een ander (naburig) neuron. De bedoeling van Artificiële Neurale Netwerken is deze biologische

concepten te gebruiken om krachtige nieuwe computer methodieken te ontwikkelen.

ANN bestaan uit een groot aantal niet-lineaire rekenelementen die we knopen (synaps) noemen. Deze knopen zijn hecht ingebed in een netwerk en onderling met elkaar verbonden. Knopen nemen een of meerdere inputwaarden (stimuli), combineren deze waarden in een enkele waarde, en transformeren deze vervolgens in een output waarde. Figuur 2 toont een knoop naar analogie met een biologisch neuron.

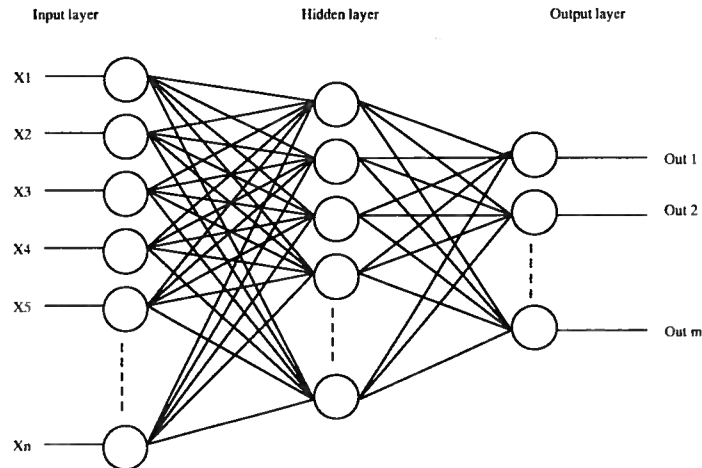


Figuur C.2: Symbolische voorstelling van een knoop

In figuur 2 worden een set van inputs (X_1 tot X_n) naar een knoop gestuurd. Elke input wordt vermenigvuldigd met een wegingscoëfficiënt (W_1 tot W_n). Elke wegingscoëfficiënt is gerelateerd aan de sterkte van een connectie (synapsis). Vervolgens worden alle inputs opgeteld. Deze sommatie noemen we 'NET'. Het blokje met label F krijgt NET binnen. Indien NET de drempelwaarde overschrijdt dan wordt de OUT-knoop geactiveerd. Erg vergelijkbaar dus met wat onze hersenen doen.

De kracht van neurale computing komt voort uit de connectie van verschillende neuronen in een netwerk; een artificieel neurale netwerk. De eenvoudigste vorm is een set neuronen geordend in n niveau (layer). Wanneer we een aantal enkelvoudige niveaus in cascade zetten krijgen we een multilayer netwerk. Figuur 3 geeft een voorbeeld van zo een multilayer netwerk, dat in dit geval bestaat uit drie niveaus. De knopen van de verschillende niveaus

zijn hecht verbonden. De input knopen ontvangen de signalen en zenden deze verder door het netwerk, layer voor layer.



Figuur C.3: Meerlaags netwerk

Zowel het aantal niveaus en neuronen als de gewichten die worden toegekend aan de verbindingen tussen de neuronen kunnen zo gekozen worden dat het neurale netwerk zo goed mogelijk de bestudeerde dataset benadert.

Artificiële Neurale Netwerken (ANN's) worden gekenmerkt door drie eigenschappen: de structuur van het netwerk (de topologie), hoe en wat het netwerk berekent (reken-eigenschap) en hoe en wat het netwerk leert berekenen (lerend vermogen). Leren is het proces waarbij een aantal input waarden herhaaldelijk aan het netwerk gepresenteerd worden en waarbij de netwerkgewichten worden aangepast zodanig dat gelijke inputs uiteindelijk dezelfde outputs geven. Er bestaan gesuperviseerde en niet-gesuperviseerde leeralgoritmen. Gesuperviseerde algoritmen koppelen elke input waarde met een te bereiken output waarde en een 'leraar' die informatie geeft over de gemaakte fouten. Bij een niet-gesuperviseerd leeralgoritme worden er enkel inputs gegeven aan het netwerk. De output wordt bepaald door het netwerk gedurende het leerproces zelf. Niet-gesuperviseerd leren creëert interne mo-

dellen die patronen zoeken in de input gegevens, zonder verdere informatie hierbij te ontvangen.

Het grote aantal proceselementen maakt neurale computing sneller dan conventionele computing. Door hun parallelle karakter zijn neurale netwerken robuust en fout-bestendig. Onder fout-bestendigheid verstaan we dat de precisie van het resultaat bijna niet wijzigt wanneer n der knopen zou uitvallen. Op basis van wat we vandaag weten over neurale netwerken, blijkt dat neurale netwerken zichzelf aanpassen in structuur en gegeven gewichten om een hogere precisie te geven. Voor meer informatie over neurale netwerken wordt verwezen naar Dayhoff (1990) of Nelson en Illingworth (1991).

Indien we ANN toepassen op de problematiek van paragraaf 2, dan wordt het mogelijk processen in kaart te brengen, via de percepties van de verschillende stakeholders. De ANN zijn in staat op basis van waargenomen percepties te leren van een situatie en uit een brede, vage massa percepties orde te creëren. Op deze wijze dragen ANN bij tot de vermeerdering van de 'bedrijfsintelligentie'. In deze rol zijn ANN enigszins het verlengde van kennissystemen. Bij kennissystemen heeft men echter nog zeer concrete regels nodig en wil men die ook heel bewust nastreven. ANN dragen ertoe bij passieve kennis (tacit knowledge) om te zetten in actieve kennis, zonder het karakter van de kennis al te zeer te veranderen.

Zoals eerder reeds vermeld gaan leren en veranderen dikwijls samen. Indien ANN bijdragen tot het creëren van bedrijfsintelligentie, via het in kaart brengen van processen, dan kunnen zulke maps eveneens gebruikt worden voor het begeleiden van veranderingsprocessen. ANN geven niet het volledig gedetailleerde beeld, maar wel een leefbaar, correct en holistisch beeld van een specifiek bedrijfsproces. Indien men dan dit beeld kan visualiseren in een aansprekende software interface bijvoorbeeld, kan dit door verschillende mensen gebruikt worden als input voor een bepaald veranderingsproces. De belanghebbenden kunnen in een niet confronterende omgeving kennis maken met elkaars 'kennis', gerelateerd aan een specifiek bedrijfsproces. Eenvoudig gezegd: de betrokkenen kunnen een idee hebben van wat er allemaal speelt rond een specifiek bedrijfsproces vanuit verschillende gezichtspunten, gebaseerd op de percepties, en ze kunnen hiermee ook nog spelen (dus leren). Verder in deze bijdrage komen we terug met een zeer concreet voorbeeld.

Voor de volledigheid dient hier kort verwezen te worden naar andere toepassingen van ANN. Neurale netwerken worden in de ingenieurswetenschappen veelvuldig gebruikt voor patroonherkenning (weervoorspelling, spraakherkenning, schriftherkenning, fraudebestrijding, procescontrole) of robotica

(gebaseerd op het lerend vermogen van neurale netwerken). ANN worden veel minder gebruikt in bedrijfstoeepassingen maar ook daar treedt verandering in op. Een aantal financiële en marketing toepassingen worden door bedrijven uitgetest (Venugopal en Baets, 1994 a en b).

Een eerste mogelijke toepassing is het voorspellen van marktaandelen en/of verkoopvolumes. Marktsegmentatie met neurale netwerken is een ander voorbeeld. In dezelfde lijn ligt de identificatie van potentieel geïnteresseerde klanten (klantpatronen). Dit sluit aan bij direct marketing problemen waarbij een bedrijf erin geïnteresseerd is om op tijd een potentiële klant te kunnen identificeren. Een toepassing waarbij gebruik gemaakt wordt van de kwaliteit van neurale netwerken om met perceptiegegevens te kunnen werken is brandimaging. Het meten van merkgevoeligheid (percepties) is hierbij een sleutelprobleem.

In de financiële sector is er ook ervaring opgedaan met een aantal toepassingen. Het voorspellen van beurzen (marktgedrag) spreekt natuurlijk het meest tot de verbeelding. Duidelijke successen zijn geboekt in het voorspellen van bedrijfsfalingen, waarbij gebruik wordt gemaakt van de kracht van neurale netwerken indien er kwalitatieve variabelen betrokken zijn en indien er gegevens ontbreken (robuust gedrag van neurale netwerken). In dezelfde lijn zijn er successen geboekt bij kredietscoring.

Artificiële neurale netwerken imiteren de werking van ons menselijk verstand, vooral door het parallelle gedistribueerde karakter. De oplossing ligt in het aanvaarden dat we de exacte vergelijkingen wellicht niet kunnen neerschrijven en dat we gebruik maken van een methode die 'geassimileerd denken' simuleert. In deze functie hebben neurale netwerken hun bruikbaarheid al getoond (Fletcher en Goss, 1993; Baets, 1993). Zij worden dan gebruikt om een voorstelling te vormen (te creëren) van een complex systeem, zonder a priori deze complexiteit te willen beperken, maar daarbij dan wel te aanvaarden dat we niet alle detailverbanden zullen kennen. Een rijker maar toch nog beheersbaar systeem in de vorm van complexiteit wordt verkregen in ruil voor minder informatie op detailniveau.

Het is niet de bedoeling in deze bijdrage op alle mogelijke toepassingen in te gaan. Wel willen we verder organisatieveranderingsprocessen leren beschouwen, zoals we bij het begin van deze bijdrage beschreven hebben. Paragraaf 4 beschrijft hoe neurale netwerken kunnen bijdragen tot het in kaart brengen van 'mind sets' aan de hand van een concreet voorbeeld. Verder gaan we in op het meten van veranderingen (leren).

C.4 De praktijk van het in kaart brengen van veranderingsprocessen

De 'heilige graal' waar we met zijn allen naar op zoek zijn als bedrijf, is de bedrijfsintelligentie (zie figuur 1). Hoe kunnen we een idee krijgen van deze intelligentie? Hoe kunnen we die voorstellen, opbouwen, uitbreiden en 'tonen' aan anderen? Dit heeft te maken met onze inzichten in wat kennis is en hoe het voorgesteld kan worden. Kennistechnologie heeft hierop reeds een aantal antwoorden geformuleerd.

Kennistechnologie gaat er impliciet nog steeds van uit dat kennis te vatten is in regels (een iets soepelere vorm van een vergelijking). Cognitief onderzoek lijkt dit echter niet te bevestigen (Rumelhart and McClelland, 1989). Kennis wordt opgeslagen in een netwerkstructuur, waarbij kennis niet als ondeelbaar element bestaat (kennis is dus niet het atoom van de bedrijfsintelligentie) doch slechts in een netwerkstructuur kan opereren. In parallelle processen worden knopen tegelijkertijd aangesproken en het netwerk levert dan een antwoord op een bepaalde vraag. Niemand weet exact welke knopen wat bevatten en welke bijdrage zij leveren aan het antwoord.

Indien deze netwerkstructuur in de praktijk blijkt te bestaan, lijkt het voor de hand te liggen de bedrijfsintelligentie ook op die wijze op te bouwen en te behandelen. Hierbij vallen we dan terug op de techniek van neurale netwerken, zoals eerder besproken. Deze techniek zullen we nu verder toelichten, maar we gaan daarbij in deze bijdrage niet verder in op de bedrijfsintelligentie. We willen ons beperken tot een klein onderdeel hiervan: het bedrijfsproces.

Figuur 1 toegepast op een bedrijfsproces suggereert dat managen bestaat uit het creëren en beheren van kennis omtrent bedrijfsprocessen om er vervolgens veranderingsprocessen mee te begeleiden. We zijn er dus in geïnteresseerd de bedrijfsprocessen te kunnen beschrijven. Maar bedrijfsprocessen zijn niet objectief; bedrijfsprocessen kunnen niet met vergelijkingen beschreven worden. Verschillende stakeholders hebben een verschillende visie op een bepaald bedrijfsproces.

Een artificieel neuraal netwerk tracht een bewegend beeld van een bedrijfsproces te maken, bekeken vanuit de percepties van de belanghebbenden. Hierbij maakt een neuraal netwerk passieve (tacit) kennis expliciet, beheert deze kennis maar laat ook toe kennis te creëren. Indien een manager proactief wil gaan beheren dan zal hij toch van de ervaring van zoveel mogelijk mensen moeten leren. Hij zal nieuwe ervaringen moeten kunnen inpassen

in de ervaringswereld van de individuen (hun individuele mentale modellen) en in de mentale modellen die bestaan op bedrijfsniveau. Deze bewegende beelden van een bedrijfsproces (of van de bedrijfsintelligentie) kunnen de manager helpen bij het verbeteren van veranderingsprocessen en het aanvaardingsproces binnen een organisatie iets gemakkelijker maken.

Hoe kunnen we nu bedrijfsprocessen meten en in kaart brengen, gebruik makende van de percepties van de betrokken partijen?

Om dit te illustreren werken we met een voorbeeld en beschrijven de stappen die hierbij doorlopen worden. Laten we er vanuit gaan dat een bedrijf een veranderingsproces wil doorvoeren, waarbij kwaliteitsborging centraal staat, doch ook heel wat nevenelementen van minder objectieve aard een rol spelen. Het zou bijvoorbeeld om een chemisch bedrijf kunnen gaan. Dit bedrijf wil een bepaald productieproces vernieuwen. Hierbij zijn zij verantwoordelijk voor de kwaliteit van het geproduceerde en er is uiteraard druk om dit bij voorkeur tegen lagere prijzen te kunnen produceren. Er zijn echter nog tenminste twee andere stakeholders. Vooreerst zijn er de milieunormen waaraan het bedrijf gevolg zou moeten geven. Het zou best kunnen dat deze ingaan tegen de eerder genoemde doelen. Overigens zijn die normen niet altijd even concreet en is er speelruimte in de toepassing ervan. Dit wil zeggen dat de toepassing afhankelijk is van de perceptie die beide partijen (bedrijf en overheid) hebben omtrent de toepassing van die milieunormen. Tenslotte is er zeker nog een derde partij: de omwonenden. De beleving van de omwonenden van de installatie en de in gebruik name van het nieuwe productieproces is bijna uitsluitend gebaseerd op percepties.

Dit is een typisch voorbeeld waarbij neurale netwerken kunnen helpen het bedrijfsproces of het veranderingsproces in kaart te brengen. Samengevat zijn er onduidelijke, meervoudige en dikwijls tegengestelde doelen en zijn er meerdere partijen betrokken in het proces. De elementen zijn niet altijd objectief meetbaar. Soms is er a priori eerder een gebrek aan vertrouwen tussen de verschillende betrokken partijen. Deze partijen zijn het dikwijls niet eens over definities, zoals kwaliteit, leefbaarheid, enzovoort, of erger nog, er bestaan nu eenmaal verschillende definities van hetzelfde begrip (zoals bij het begrip kwaliteitsborging).

In deze gevallen heeft men er belang bij op een niet confronterende wijze een 'foto' te kunnen maken van het probleem of het proces. Indien men de verschillende partijen om de tafel zou brengen, loopt dit dikwijls uit op het versterken van vooraf ingenomen en tegengestelde standpunten. Men komt zelden tot een consensus. Overigens is het bij zulke processen behoorlijk

moeilijk om verbanden te leggen tussen verschillen fenomenen. Men weet wel wat er belangrijk is, maar men weet zelden hoe de verschillende elementen elkaar beïnvloeden. Indien we deze 'foto' dan aan alle betrokkenen kunnen tonen en indien zij daarmee aan de slag kunnen om elkaars percepties te begrijpen zonder onmiddellijk in stellingen-oorlogen te verzeiken, is er toegevoegde waarde.

Via verschillende wegen (bijvoorbeeld reeds bestaand materiaal), interviews, interne nota's) wordt informatie verzameld over die elementen die een rol spelen in het proces. Daarbij wordt niet gevraagd naar de invloed van de factoren op het proces of naar de verbanden tussen elementen omdat verbanden gebaseerd zullen zijn op de persoonlijke mentale map van de betrokken persoon (Kim, 1993). We zoeken enkel de issues op een zo autonome mogelijke wijze (te vergelijken met de rol van het atoom in chemische processen). Onder het uiteindelijke doel is de mentale map van de betrokken personen te beïnvloeden, willen we op dit ogenblik vermijden dat de huidige mentale map interfereert met het mogelijk gewenste doel.

Om deze elementen te valideren, worden ze getest met een aantal betrokken partijen. Na validering wordt een vragenlijst opgebouwd waarin gevraagd wordt naar het belang van de verschillende issues. Het belang van de issues wordt aangegeven op een Likert schaal, bijvoorbeeld van 1 tot 10.

Via niet-gesuperviseerd leren wordt er een mogelijke structuur in de gegevens gezocht. De orde en richting die uit dit proces komen, kunnen verschillend zijn van wat er vandaag als logisch aanvaard wordt. Bijvoorbeeld zou kunnen blijken dat iets wat de marketing-afdeling altijd belangrijk vond helemaal niet meer voorkomt als iets wezenlijks voor het hele proces. Het zou bijvoorbeeld ook kunnen blijken dat hoewel verschillende partijen verschillende definities van kwaliteit hanteren, er over praktische issues toch een verstandhouding blijkt te bestaan. Uiteindelijk zal een voorstelling toelaten een aantal mogelijke structuren te identificeren, die uit de observaties zelf gegroeid zijn (en niet vanuit een bepaalde mentale map opgelegd werden). Deze verschillende structuren zijn a priori even waarschijnlijk. Enkel de bruikbaarheid zal hierin onderscheid maken.

Dan zijn er twee vragen om verder te gaan: verdere analyse met gesuperviseerde algoritmen; het in kaart brengen van het proces zoals het zich voordoet.

Indien we kiezen om verder te gaan in de analyse, dan is de volgende stap een aantal van deze structuren met gesuperviseerde algoritmen verder in te vullen. Hierbij wordt dan niet enkel ingegaan op een structuur, maar

tevens op de onderlinge beïnvloeding van verschillende variabelen. In dit stadium trachten we dus het proces bloot te leggen, eveneens enkel gebaseerd op de waarnemingen, niet vertrekkende van vooropgezette structuren. Dit netwerk geeft dan de 'ins', 'outs' en 'throughs' van het veranderingsproces, zoals gepercipieerd door de verschillende stakeholders. Deze stap geeft meer informatie omtrent de achtergrond van het veranderingsproces. Het brengt dit proces niet noodzakelijk in kaart, in de zin van de eerder genoemde 'bedrijfsfoto'. Het betreft eerder detail opnemen, die daarom niet minder interessant zijn.

De tweede weg die gekozen kan worden is het veranderingsproces in kaart te brengen, met het doel dit te kunnen communiceren en hiervan te kunnen leren. Om dit in de praktijk mogelijk te maken moet er een instrument gemaakt worden en praktisch gebeurt dit door het gebruik van interactieve software tools. Een bijkomend voordeel hiervan is dat men het 'resultaat' gemakkelijk kan verspreiden onder veel mensen binnen een organisatie, gebruik makende van elektronische netwerken. Hierdoor is het mogelijk effect veel groter dan wat verkregen kan worden door bijvoorbeeld workshops. Speciaal als het dus gaat om veranderingsprocessen waarbij veel mensen (en groepen) betrokken zijn heeft dit een bijkomend voordeel.

Deze kaart tracht niet de grootste gemene deler van de percepties te vinden, maar houdt wel degelijk de verscheidenheid in het netwerk. Hierin ligt duidelijk een sterkte van neurale netwerken, vergeleken met andere mapping technieken (zoals cognitieve mapping van Eden, 1988). Cognitieve mapping, bijvoorbeeld, zoekt actief naar een consensus waar die niet noodzakelijk hoeft te bestaan. Het zou best kunnen dat een veranderingsproces moet doorgaan zonder voorafgaand akkoord onder alle partijen. Speciaal in dit geval is het van belang de diversiteit te bewaren en te tonen, en niet van bij aanvang te streven naar een consensus (een gemiddelde).

Het resultaat van het getrainde netwerk is feitelijk een matrix van waarden, waarbij alle beschouwde issues bijvoorbeeld in de kolommen staan en een vooraf bepaald aantal groepen van waarnemingen op de rijen. Men dient dus vooraf een keuze te maken hoeveel issues men wil gebruiken, indien dit minder zou zijn dan het totale aantal. Verder dient men ook een beslissing te nemen over hoeveel groepen men wil in de waarnemingen. Deze groepen zijn te vergelijken met de clusters in een clusteranalyse. Hoe meer groepen je toelaat, hoe meer diversiteit je meeneemt, maar hoe minder begrijpelijk het wordt. Indien je kiest voor minder groepen verlies je natuurlijk detail maar wordt het 'gedrag' wel iets meer voorspelbaar. Ook hier is geen enkele keuze

a priori goed of slecht. Het gebruik zal dikwijls uitwijzen wat het beste aantal is. Overigens bestaan er ook algoritmen die hierbij hulp kunnen bieden.

De sleutel van het getrainde netwerk is dus zoals gezegd de matrix van waarden. Deze matrix kan nu op verschillende manier worden voorgesteld. De meest eenvoudige is gebruik makende van een grafenvoorstelling. Men kan zich hierbij een fietswiel met spaken voorstellen. Op het einde van elke spaak staat een issue. Deze voorstelling kan dermate geprogrammeerd worden dat men dan kan gaan inklikken op een bepaald issue. Dan vraagt het programma een waarde voor dit issue in te geven op een schaal van bijvoorbeeld 1 tot 10. Het systeem geeft dan aan in welke groep(en) van gedragspatroon men kan zitten. Indien men een aantal issues aanklikt en waarden ingeeft, dan zal het aantal groepen teruggebracht worden tot n enkele. Dit alles is natuurlijk afhankelijk van de grootte van het probleem. Dan is het netwerk in staat alle andere issues af te beelden als vectoren op de verschillende spaken. Men krijgt dus onmiddellijk visueel het totale patroon van de percepties van de beschouwde persoon omtrent het bedrijfsproces. Hiermee kan men dan gaan 'spelen' en inzicht krijgen in de verschillende beelden die stakeholders hebben betreffende hetzelfde bedrijfsproces.

Indien de grootte van het probleem belangrijk is, zullen er slechts een beperkt aantal spaken kunnen afgebeeld worden. Verschillende mogelijkheden bestaan dan: men kiest voor de belangrijkste, de minst belangrijke, een gemiddelde van de verschillende groepen, enzovoort. De gebruiker kan dit dan vrij kiezen en ook afwisselen.

De mogelijke uitbreiding van dit voorstellingsstelsel is dan de vrijheid van de ontwerper. Indien de financiële middelen beschikbaar zijn kan zelfs gedacht worden aan multimedia toepassingen. Dit draagt enkel bij tot de communicatiewaarde en niet zozeer tot de kwaliteit van het netwerk. Deze interface is echter wel belangrijk indien men met dit instrument verandering wil ondersteunen. Hoe beter de processen gecommuniceerd kunnen worden, hoe effectiever men mag verwachten dat de veranderingen zullen zijn.

Veranderen (of leren) is waar we eigenlijk naar op zoek zijn. We hebben een instrument en methoden gebruikt om iets te creëren waarbij het gemakkelijker wordt veranderingsprocessen te begeleiden. Misschien willen we dan ook wel weten of verandering is opgetreden.

Om te kunnen leren moet een individu (of een organisatie) impulsen krijgen, geconfronteerd worden met 'observaties'. Het inpassen hiervan in ieders mentale map zal dan eventueel leiden tot leren: het aanpassen van de mentale map. Veranderingsprocessen zijn op zich eveneens observaties, die aanleiding

kunnen geven tot leren. Men kan zich dus perfect een loop voorstellen, waarbij veranderen een aanleiding wordt voor leren. Dit illustreert de gekende 'single' en 'double' loop leerfenomenen (Kim, 1994). Hierop gaan we in deze bijdrage niet verder in (o.a. in Baets, 1995b, is hierover meer geschreven).

Wat echter wel belangrijk is, is het probleem van het meten van veranderen, dus indirect het meten van leren uit veranderingsprocessen. Indien het inderdaad waar is dat men veranderingsprocessen langs percepties in kaart kan brengen, dan ligt het ook voor de hand dat op die wijze een meetinstrument gebouwd kan worden om verandering (leren) te kunnen meten.

We kunnen op een bepaald ogenblik een map construeren die een bepaalde set percepties toont. Op basis hiervan kan men dan een veranderingsproces trachten te begeleiden. Op een later tijdstip kan men dan dezelfde map (dezelfde structuur) terug trainen, met percepties van het nieuwe ogenblik. Vergelijking van beide mappen zal toelaten vast te stellen of er verandering is opgetreden. Door het vergelijken van de twee mappen kan men dan vaststellen waar welke veranderingen opgetreden zijn. Dit geeft uiteraard geen kwantitatieve meting en geeft zelfs geen richting van leren aan (positief of negatief). Men kan zich echter wel afvragen of andere meetmethoden van leren dit wel correct weergeven en of het überhaupt mogelijk is bijleren en afleren van elkaar te onderscheiden. Men kan zich afvragen of men ooit objectief in cijfers kan meten of wat er geleerd is. Tot nu toe hebben we dat echter al voor enkele tientallen (zo niet honderden) jaren gedaan.

Binnen bedrijfsveranderingsprocessen, waar (continue) verandering op zich een belangrijk doel geworden is, lijken neurale netwerken een monitoring instrument te zijn waarmee verandering getoond kan worden. Als we ervan uit gaan dat verandering op zich geen goede en slechte richting kent (Senge, 1990) is dit wellicht, in combinatie met het gebruik gedurende het veranderingsproces zelf, een goede manier om een bedrijf te ondersteunen.

C.5 Beloften en zwakten van neurale netwerken

Zoals reeds eerder gezegd staan neurale netwerken in bedrijfstoepassingen nog in de kinderschoenen. Deze toepassingen zitten in de klassieke functionele domeinen (marketing, finance). Hoewel neurale netwerken hier goed werk kunnen leveren, lijkt de toegevoegde waarde niet helemaal daar te liggen.

De toegevoegde waarde van neurale netwerken ligt daar waar andere methoden falen of niet helemaal slagen. Hoewel het moeilijk is dit in enkele zinnen samen te vatten spreken we dan toch over processen waarbij percepties belangrijk zijn, processen waarbij we complexe situaties volledig (maar misschien niet in detail) in kaart willen brengen, of situaties waarbij we erin geïnteresseerd zijn om passieve (tacit) kennis expliciet te maken. Indien we deze drie gebieden even toelichten, zien we ook onmiddellijk hoe lang de weg is die we nog moeten afleggen.

Percepties zijn een belangrijk element in bedrijfsvoering. In de nabije toekomst zullen we wellicht verder onderzoek zien voor het gebruik van neurale netwerken bij klassieke problemen waarbij percepties belangrijk zijn. Hierbij kan dan gedacht worden aan toepassingen in de marketing, zoals brand imaging, en kwaliteitsmanagement. De toepassingen zijn nogal rechttoe-rechtaan. De kans op succes is groot en de moeilijkheidsgraad van de ontwikkeling onbeheersbaar.

Een tweede groep van toepassingen waar we in de toekomst neurale netwerken nuttig kunnen gebruiken betreft het in kaart brengen van complexe situaties. In de management literatuur is men het er meer en meer over eens dat management in de toekomst meer zal lijken op omgaan met onzekerheden, complexiteit en chaos, dan op het plan-en-controle mechanisme van het verleden. Daarop zijn we vandaag nog niet echt voorbereid.

Complexiteitstheorie (en tevens ook chaos theorie) komt maar net om de hoek kijken, maar wij rationele managers doen behoorlijk ons best om hen terug om de hoek te trappen. Het lijkt wel een beetje struisvogel gedrag. We voelen ons niet gemakkelijk met onzekerheden en tevens zijn we (en worden we nog steeds in ons opleidingsbestel) doordrenkt van het besef dat men enkel kan managen wat men expliciet kan maken. Alleen, expliciet maken wil dan in de praktijk zeggen: vatten in lineaire, niet-dynamische systemen. Dat kunnen dan vergelijkingen zijn (simulatie modellen), maar ook kennis regels (expert systemen) of zelfs afgesproken regels (evalueren van personeel). Daarbij stellen we ons niet echt de vraag of de afgesproken regels (vergelijkingen) niet een oversimplificatie zijn van de realiteit.

Wellicht zullen we naar voorstellingen moeten die de gehele complexe realiteit beschrijven (simuleren). De prijs die we daarvoor betalen is een gebrek aan detail. We zijn vandaag echter niet ver genoeg om detail 'vergelijkingen' te schrijven van de complexe realiteit. Per definitie kunnen neurale netwerken een goede en complete beschrijving leveren van complexe situaties, als we daarbij aanvaarden dat we een stuk detail weggeven. In dat geval kunnen

we zelfs langs neurale netwerken complexe situaties simuleren. Maar zover is bedrijfskundig onderzoek nog lang niet. In de toekomst zullen we hopelijk ook onderzoek zien in neurale netwerken en complexiteitstheorie, toegepast op bedrijfssituaties. Binnen de bedrijfskunde is wellicht de universiteit van Aix-Marseille (Frankrijk) de enige in Europa die een in dit verband specifiek research center heeft dat ook nog tamelijk actief is. Hopelijk zien we hier in de toekomst een evolutie in.

Vanuit de hoek van cognitieve wetenschappen en de organisatie wetenschappen zien we interesse in kennismanagement groeien. Zonder hierop in detail te willen ingaan ligt er een terrein open in onderzoek naar hoe men passieve kennis (tacit knowlegde) kan gebruiken in een bedrijf, zonder het passieve karakter te verliezen (Nonaka, 1994). Uit eigen onderzoek (bijvoorbeeld Baets, 1993, 1995) blijken neurale netwerken hier een interessante mogelijkheid te bieden. Doch dit staat eveneens nog in de kinderschoenen. Hopelijk laat de ontwikkeling in 'organisational learning' (en de onderzoekscentra die hierover worden opgericht op verschillende plaatsen) enige ruimte om ook dit aspect aandacht te geven. De toekomst zal dit uitwijzen.

Conclusies trekken lijkt me op dit ogenblik iets te ver te gaan en het zou tevens suggereren dat we al heel veel bereikt hebben. Hopelijk staan we aan het begin van een evolutie die in de toekomst successen zal kennen. Ik zou het hier dan ook beperkt willen houden tot het samenvatten van de tendensen.

Neurale netwerken zijn beloftevol voor toepassingen waarbij percepties een belangrijke rol spelen of waarbij we geen exacte vergelijkingen kunnen neerschrijven (meestal in marketing en finance). Onderzoek naar deze toepassingen is goed op weg.

Neurale netwerken zijn beloftevol voor het in kaart brengen en begeleiden van organisatie veranderingsprocessen. Alleen hierbij staan we nog in de kinderschoenen en zal er meer bewijs vergaard moeten worden om dit ook wetenschappelijk aan te tonen.

Boeiend onderzoek ligt te wachten naar het gebruik van neurale netwerken als ondersteuning van management in complexe situaties, alsook bij kennismanagement (niet in de zin van rule-based systemen). Hopelijk zien we hier in de toekomst een belangrijke toegevoegde waarde van neurale netwerken.

C.6 Bibliografie

- Ahalt, S.C., Krishnanmurthy, A.K., Chane, P., and D.E. Melton, 'Competitive Learning Algorithms for Vector Quantization', Neural Networks 3, pp. 277-290, 1990.
- Argyris C and Schbn D, 1978, Organizational Learning: A Theory of Action Perspective, Reading, MA: Addison-Wesley),
- Baets W, 1992, 'Aligning information Systems with Corporate Strategy', The journal of Strategic Information Systems, Vol 1, Nr 4
- Baets W, 1993, "IT for organisational change: beyond business process engineering", Business Change and Re-engineering (The Journal of Corporate Transformation), Vol 1, Nr 2, Autumn
- Baets W, 1995a, Ecosip study day on Business Process Reengineering and Information Technology (Paris): Business Process Re-engineering: A corporate mind set
- Baets W, 1995b, 'The corporate mind-set as a precursor for Business Process (Re-) engineering: About knowledge, perceptions and learning', in Information and the 21th century organisation: organisational innovation and transformation, Baets W and Gailiers R (eds), forthcoming
- Baets W, 1995c, 'Artificiele Neurale Netwerken', Handboek Effectief Opleiden, Delwel, Juni
- Baets W, Brunenberg L, van Wezel M and Venugopal V, 1995, 'Corporate Cognitive Mapping: Mapping of corporate change processes', Fourth International Conference on Artificial Neural Networks, Cambridge, UK
- Baets W and Venugopal V, 1994, EURO Xill/OR 36 Conference (Glasgow): No doubt, neural networks are alternative tools for statistical techniques: but yet...
- Cohen j en Stewart 1, 1994, The collapse of chaos, Viking
- Dayhoff J, 1990, Neural Network Architectures, Von Nostrand Reinhold Book, NY

- De Geus A, 1988, "Planning as learning", Harvard Business Review, March/April, pp 70-74
- Eden C, 1988, 'Cognitive mapping', European Journal of Operations Research, 36, pp 1-13
- Fletcher and Goss, 1993, "Forecasting with neural networks", Information and Management, Vol 24, Nr 3, March, pp 159-167
- Fritzsche B, 1994, 'Growing Cell Structures - A Self-Organizing Network for Unsupervised and Supervised Learning', Neural Networks, Vol 7, Nr 9, pp 1441- 1460
- Haeckel S and Nolan R, 1993, 'Managing by wire', Harvard Business Review, Sept-Oct, Vol 71, Nr 5
- Hassard J, 1993, Sociology and Organization theory: Positivism, Paradigms and Postmodernity, Cambridge University Press
- Hassard J and Parker M, 1993, Postmodernism and organizations, London, Sage
- Kaufman L, P.J. Rousseeuw 'Finding Groups in Data: an Introduction to Cluster Analysis'. Wiley Series in Probability and Mathematical Statistics, Wiley, 1990.
- Keen P, 1993, 'Information Technology and the management difference: A fusion map, IBM Systems Journal, Vol 32, Nr 1
- Kim D, 1993, 'The link between individual and organizational learning', Sloan Management Review, Fall
- Krishnaiah, P.R. and L.N. Kanal, eds. 'Classification, Pattern Recognition, and Reduction of Dimensionality'. Handbook of Statistics, 2, Amsterdam: North holland, 1982.
- Krzanowski, W.j., 'Principles of Multivariate Analysis'. Oxford statistical science series, 3, Oxford University Press, 1988.
- Langton C (ed), 1989, Artificial life, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings vol 6, Redwood City CA, Addison Wesley

- Morin E, 1990a, Sciences avec conscience, Paris, Fayard
- Morin E, 1990b, Introduction A la pens6d complexe, Paris, Esf
- Nelson M en Illington W, 1991, A practical guide to Neural Nets, Addison-Wesley, New York
- Nicolis C en PrigOgine I, 1989, Exploring Complexity, Freeman
- Nonaka I, 1994, 'A dynamic theory of Organizational Knowledge creation', Organization Science, Vol 5, NR 1, February
- Ritter H.J. and Kohonen T, 1989, 'Self organizing semantic maps', Biological Cybernetics, 61, pp 241-254
- Rumelhart D and McClelland J, 1986, Parallel Distributed Processing: Exploration in the Microstructure of cognition, Vol 1: Foundations, Cambridge, MA: MIT Press
- Rumelhart D and McClelland J, 1986, Parallel Distributed Processing: Exploration in the Microstructure of cognition, Vol 2: Psychological and Biological Models, Cambridge, MA: MIT Press
- Santa Fe Institute, Bulletin of the Santa Fe Institute, 1987 - present
- Senge P M, 1990, The Fifth Discipline: The Art & Practice of the Learning Organization, Doubleday, NY
- Sammon John W. Jr. 'A nonlinear mapping for data structure analysis'. IEEE Transactions on Computers, C-18(5):401-409, May 1969.
- Swieringa J en Wierdsma A, 1990, Op weg naar een lerende organisatie, Wolters- Noordhoff
- Ultsch, A. 'Self organized feature maps for monitoring and knowledge acquisition of a chemical process'. in 5. Gielen, B. Kappen (eds.) 'Proceedings of the International Conference on Artificial Neural Networks' (ICANN93), pp 864-867. London. Springer-Verlag, 1993.
- Van Den Broeck H, 1994, Lerend Management, Lannoo Scriptum

- Venugopal V and Baets W, 1994a, 'Neural Networks and Statistical Techniques in Marketing Research: A conceptual comparison', Marketing Intelligence and Planning, Volume 12, Number 7
- Venugopal V and Baets W, 1994b, 'Neural Networks and their Applications in Marketing Management', The journal of Systems Management, September 1994
- Venugopal V and Baets W, 1995, 'An integrated intelligent support system for Organisational Learning - A conceptual framework', The Learning Organization, Fall
- Windrow B and Stearns S, 1985, Adaptive Signal Processing, Englewood Cliffs, NJ, Prentice Hall