Chem. Senses 23: 531–539, 1998

Action Potential Classifiers: a Functional Comparison of Template Matching, Principal Components Analysis and an Artificial Neural Network

J.P. Stitt[1,2], R.P. Gaumond[2], J.L. Frazier[1] and F.E. Hanson[3]

[1]Department of Entomology, Pennsylvania State University, University Park, PA 16802, [2]Bioengineering, Pennsylvania State University, University Park, PA 16802 and [3]Department of Biological Sciences, University of Maryland Baltimore County, Baltimore, MD 21250-5398, USA

Correspondence to be sent to: J.P. Stitt, Department of Entomology, Pennsylvania State University, University Park, PA 16802, USA. e-mail: jps120@psu.edu

Abstract

Multiunit neural activity occurs often in electrophysiological studies when utilizing extracellular electrodes. In order to estimate the activity of the individual neurons each action potential in the recording must be classified to its neuron of origin. This paper compares the accuracy of two traditional methods of action potential classification—template matching and principal components—against the performance of an artificial neural network (ANN). Both traditional methods use averages of action potential shapes to form their corresponding classifiers while the artificial neural network 'learns' a nonlinear relationship between a set of prototype action potentials and assigned classes. The set of prototypic action potentials and the assigned classes is termed the training set. The training set contained action potentials from each class which exhibited the full range of amplitude variability. The ANN provided better classification results and was more robust in analysis of across-animal data sets than either of the traditional action potential classification methods.

Introduction

Background

In insects, chemosensory neurons are grouped together into distinct organs called sensilla that are located on mouthpart appendages. Although each chemosensory cell acts independently and responds to different compounds, it is often not feasible to record electrophysiologically from single sensory neurons. Instead, responses of several neurons are recorded from the entire sensillum via a single electrode, and therefore appear multiplexed on a single record. Accordingly, action potential (AP) classifiers must be employed to identify single neuronal APs in the multi-neuronal signal. The AP frequencies in the separated channels then quantify the activity levels of the individual sensory neurons.

Several AP classification techniques have been developed to separate similar-shaped APs in single recordings. Wheeler (1999) groups such schemes into template matching and principal components schemes. Hanson *et al.* (1986) developed a template-matching algorithm that was later incorporated into a software package (SAPID) (Smith *et al.*, 1990) that is often used in insect chemosensory studies. Each of these classification algorithms has limitations, however, and therefore we have developed a new design that employs an artificial neural network (ANN) (Wasserman, 1989). The ANN is 'trained' using prototype action potentials from averages of individual trials, thereby relieving the operator from making decisions about AP classes, an ever-present source of bias.

Classification problem

We developed this classifier to determine the neural coding performed by the two taste organs, the lateral and medial maxillary styloconica, that are the primary contributors to the feeding decision center of the larval tobacco hornworm, *Manduca sexta* (Waldbauer, 1962; deBoer and Hanson, 1987). Each of these sensilla contains four taste neurons, so the across-fiber activity levels of eight sensory neurons (16 bilaterally) provide the primary inputs to the feeding behavior control center (deBoer and Hanson, 1987; Zacharuk and Shields, 1990).

Previous investigations have identified chemical compounds that selectively activate individual neurons within a sensillum (Peterson *et al.* 1993). Therefore we will refer to each identified neuron using the name of that chemical compound (e.g. medial inositol neuron), and to the corresponding chemical as the 'reference compound'. Electrophysiological records of the eight medial and lateral taste neurons are identified by reference compounds in Figure 1.

Discernible differences exist among the observed shapes

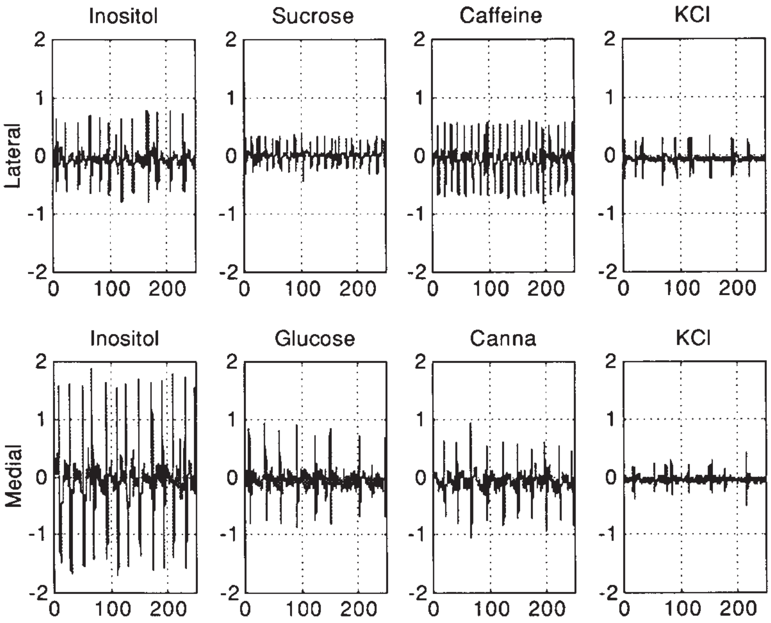

Figure 1 Segments (0.25 s) of AP trains elicited by the application of specific reference compounds for each of the eight taste neurons that constitute the two maxillary styloconic taste receptors. The plots along the top row were recorded from the lateral sensillum while those along the bottom row are associated with the medial sensillum.

of the APs produced by the four neurons of a sensillum. These differences are easily seen in Figure 2, which contains a plot of the averaged AP shapes ('exemplar APs') obtained from an ensemble of hundreds of APs from many trials of responses to the reference compound by the same insect. The time course of each of the four exemplar APs for the medial sensillum is depicted in Figure 2. The representation of each AP corresponds to the ensemble average at 32 amplitude samples (with linear interpolation between samples) taken at 100 μs intervals (10 kHz sampling rate) for 3.2 ms. The exemplar AP of each of the four neurons has a distinct shape, with the inositol neuron producing the largest amplitude AP and the KCl neuron producing the smallest. While classification of the exemplar APs is a simple task, classification of the individual APs which constitute the ensemble is much more difficult because of their variability. The amplitude distributions of each AP type overlap one or more of the others at some of the 32 sample points. As is evident in Figure 2, the peak value offers the largest difference in amplitude for the exemplars, yet there is considerable overlap within the amplitude distributions (Figure 3). The objective of the present study was to identify a classification algorithm that possesses the greatest separation ability despite these overlaps in AP shape. In determining a 'best' method for classification, an additional consideration was that the method should require little or no operator involvement during the routine classification process to optimize speed and minimize potential operator bias.

One problem for classifiers is the change of AP amplitude during a trial. For the medial inositol neuron, for example, the phasic or transient response occurs within the first 50–150 ms of stimulation and then settles into the steady state, or tonic, phase of the trial (Figure 4). During the phasic portion, a steady increase in AP amplitude can be observed in conjunction with a time-dependent firing rate. The increasing amplitude will result in misclassification of the first few APs of an inositol trial because they resemble the exemplar AP of the glucose neuron more closely than the inositol neuron. The conventional method of dealing

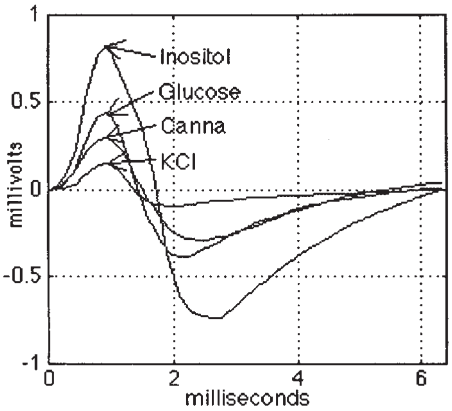

Figure 2 The ensemble averages of the four classes of APs associated with the medial taste sensillum. All of the waveforms are biphasic and have returned to the baseline within 6.4 ms (64 sample points). The first 32 samples of each waveform are used as templates for the template matching classifier and to form the covariance matrix of the PC classifier.

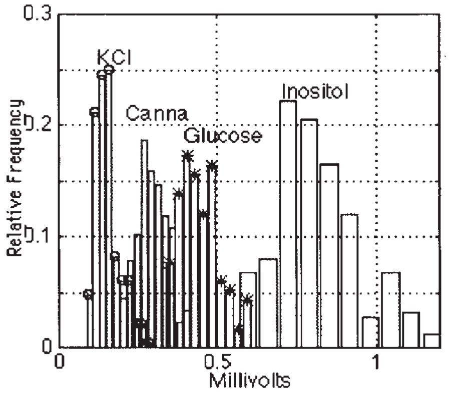

Figure 3 Peak amplitude relative density functions of the four medial ensembles. The mode of each ensemble's histogram occurs at a distinct voltage level; however, there is overlap between the tails of the four distributions.

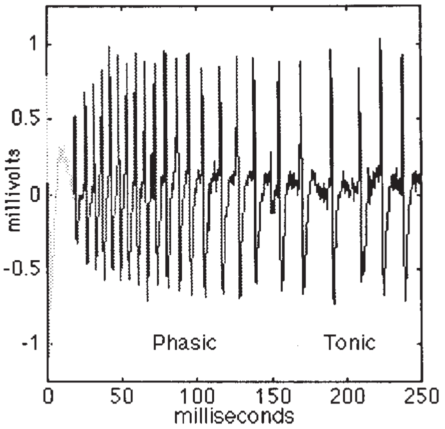

Figure 4 The first 0.25 s of a typical medial inositol AP train. The onset artifact that occurs during the first 25 ms of the sequence is ignored. The phasic portion occurs during the interval immediately following the onset artifact to ~150 ms. The tonic phase begins at ~150 ms and lasts for the remainder of the stimulation.

with this problem has been to avoid the phasic portion by ignoring the first 50–150 ms of a trial (Frazier and Hanson, 1986; Roessingh *et al.*, 1997). Thus, accommodating to changing AP shapes would also be a desirable property of a classification method.

Here we compare the results of these three classification methods operating on the same set of APs, taking data from both the phasic and tonic phases of each recording. In addition, we compare across-animal applicability of the classification techniques to determine robustness of the various classification schemes.

Conventional classification methods

Many different AP classification techniques are available, ranging from simple threshold discriminators to complex software algorithms (Frazier and Hanson 1986; Wheeler, 1999). Here we discuss two of the most widely used computer techniques, template matching and principal components analysis, and compare them to a novel method which employs an ANN.

In the following discussions, all exemplars and APs are represented as 32-element column vectors whose elements are a sequence of amplitude samples over 3.2 ms of an AP train of one second duration. Vectors are shown as lower-case bold-faced characters (e.g. $\mathbf{x}_j$ the *j*th AP of a trial) and matrices are set as upper-case bold-faced characters (e.g. $\mathbf{X}$ is a 32 × M matrix, where M is the number of APs in the corresponding trial). The transpose of a vector or matrix is indicated by a superscript T (e.g. $\mathbf{x}_j^{\mathrm{T}}$ represents a row vector).

Template matching

The template matching method of AP sorting involves determining the minimum Euclidean distance

$$\min_{i\in\{1,2,3,4\}}\left\|\mathbf{x}_j-\mathbf{e}_i\right\|=\left\{\sum_{k=1}^{32}\left(\mathbf{x}_j[k]-\mathbf{e}_i[k]\right)^2\right\}^{0.5}$$

between the set of four template APs ($\mathbf{e}_i$) and the *j*th AP ($\mathbf{x}_j$). This involves calculating the sum of the squared differences between the unknown AP $\mathbf{x}_j$ and $\mathbf{e}_i$ at each of the 32 sample points. This calculation is repeated for all APs of a trial.

The exemplar APs, discussed above, serve as the templates for each sensillum's neural responses. SAPID (Smith *et al.*, 1990) is a commercially available template matching program for the DOS/Windows platform.

Principal components

The method of principal components analysis (PCA) employs the discrete Karhunen–Loeve expansion (Fukunaga, 1990) or Hotelling transformation, and uses the exemplar APs of each sensillum to determine a small number of features that can be used to classify APs. The values of the features are determined by the degree to which an AP that is to be classified resembles (projects onto) each of the principal components (PC) that have been selected. The PCs onto which we are projecting are the eigenvectors of the composite correlation matrix having the N largest corresponding eigenvalues.

An estimate of the individual covariance matrix for the *i*th neuron is obtained by summing over all M_i of the outer products

$$\mathbf{C}_i=\left(\frac{1}{M_i}\right)\sum_{m=1}^{M_i}\left(\mathbf{w}_{m,i}-\mathbf{e}_i\right)\cdot\left(\mathbf{w}_{m,i}-\mathbf{e}_i\right)^{\mathrm{T}}$$

of the differences between each AP ($\mathbf{w}_{m,i}$) in the *i*th class ensemble and the exemplar that is associated with the *i*th class. The composite covariance matrix is formed by summing the individual covariance matrices

$$\mathbf{C}_i = \sum_{i=1}^{4} \mathbf{C}_i$$

for each of the constituent neurons of a sensillum. Next, the composite covariance matrix is orthogonalized [see Gram-Schmidt orthogonalization in Strang (1988)] to produce a matrix whose columns represent an orthonormal basis of the composite covariance matrix's vector space. Each column is an eigenvector, $\mathbf{p}_k$, $k = 1, \ldots, N$, of the composite covariance matrix, and the eigenvectors with the largest corresponding eigenvalues are selected as the PCs.

The objective when using the PC method is to reduce the dimensionality of the classification problem while retaining a sufficient amount of information to adequately classify the transformed APs. The value of each eigenvalue (variability) serves as a measure of the amount of information that is represented by the corresponding eigenvector. Figure 5 is a plot of the amount of information (variability) contained in the first *N* PCs and the corresponding percentage of APs that are misclassified. For example, the first five PCs contain 98% of the information and result in a misclassification of 7.2% of the total ensemble of APs. The percentage of misclassified APs converges to the lower bound of 7% by using the first six PCs.

The inner product of an AP and the principal components

$$f_k = \mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{p}_k \qquad k = 1, \ldots, N \quad j = 1, \ldots, M$$

represents the degree to which the AP projects onto each PC. The amount that an AP projects onto the *k*th PC represents the value of the *k*th feature (f_k).

Artificial neural network classifier

Artificial neural networks represent a collection of information processing procedures that are capable of adapting their internal states to 'learn' a relation between a set of inputs and the corresponding outputs. There are many different architectures and training algorithms which can be employed to form an ANN (Wasserman, 1989; Zurada, 1992).

For our AP classifier we implemented a two-layer feed-forward ANN using the error back-propagation supervised training procedure (see Appendix for more details). The ANN is composed of two layers of processing elements (PEs) (see Figure 6). In the literature, PEs are also referred to as (artificial) neurons or perceptrons (Wasserman, 1989; Zurada, 1992).

This version of ANN functions in two modes. The first mode is a training mode where the ANN is presented with a training set with which the training algorithm adjusts the internal state of the ANN until the error between the actual outputs and the desired outputs is minimized. Figure 7 plots the training set that was used to develop our ANN classifier. The training set was composed of thirty-four 36-element association vectors. The first 32 elements of each association vector constitute a prototype AP, the remaining four elements representing the desired output associated with the prototype (e.g. if the prototype is an inositol AP, the desired output vector would be [1 –1 –1 –1]).

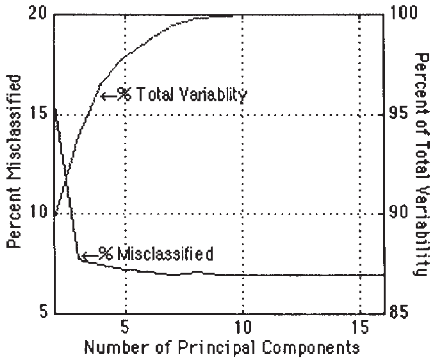

Figure 5 This figure shows plots of both the percent misclassified and percentage of total variability (information) as a function of the number of principle components (PC) used to form the classifier. Both parameters have reached their optimal bounds by the sixth PC.

The second mode of ANN operation is as a feed-forward classifier. The internal state of the ANN remains fixed. The individual APs of unknown origin that have been elicited by more complex chemical mixtures are classified using I/O relations that were 'learned' during the training mode. All of the PEs perform the same mathematical operation. Each input element *x*[*i*] is multiplied by the corresponding connection weight *w*[*i*]. The sum over all *x*[*i*] serves as the argument to the activation function, which in our case is the hyperbolic tangent function [i.e. $g(\mathbf{x}) = \tanh(\mathbf{x} \cdot \mathbf{w})$]. The value produced by the activation function serves either as the input to each PE of the next layer [$y_j = g(\mathbf{x})$] or as the output of the ANN [$z_k = g(\mathbf{y})$]. The result produced by the *j*th hidden layer PE is the weighted sum of all of the input values. The value produced by the *k*th output PE (z_k) is a weighted sum of the values produced by all hidden layer PEs (i.e. each output is a composite function of all of the input elements).

$$z_k = g\left(\sum_j g_j(\mathbf{x})\right)$$

Methods

As the main goal of this study was to improve AP classification, asn ensemble of typical APs was compiled to provide an objective comparison of the performance of each classifier. APs were extracted from multiple 1 s trials that were generated by each of four reference neurons. The ensemble consisted of 775 APs of known origin (249 inositol; 116 glucose; 177 canna; 233 KCl) that were

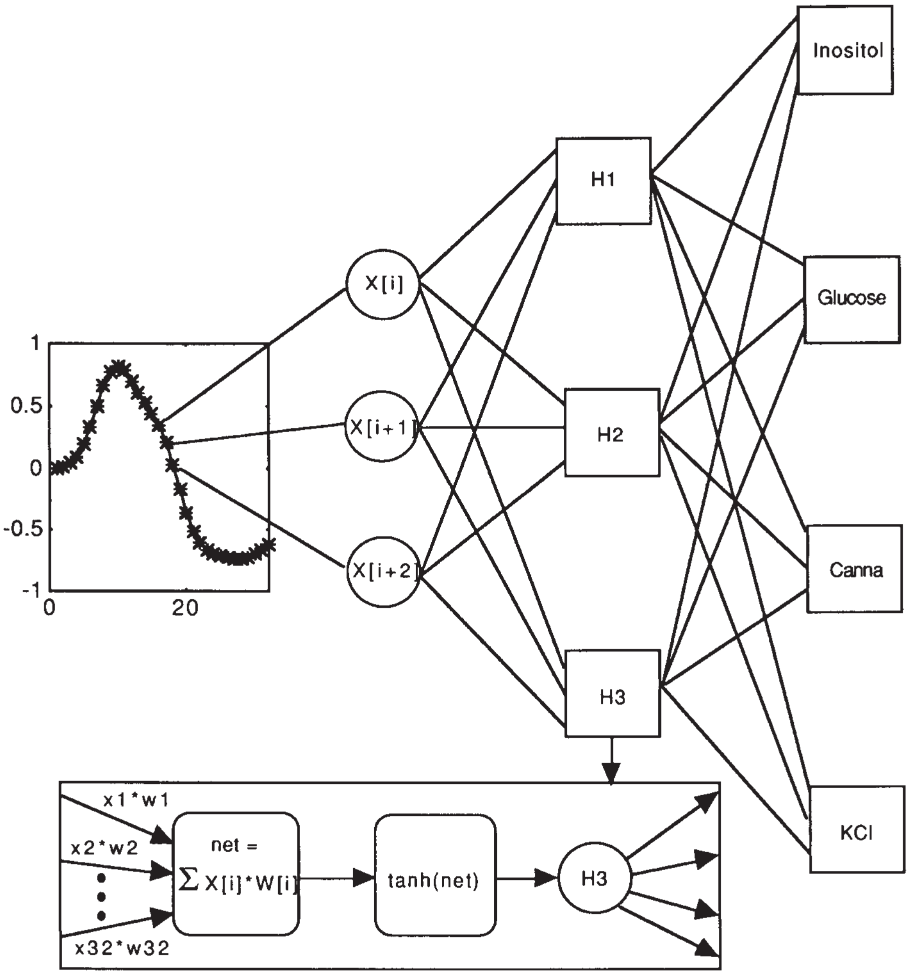

Figure 6 Diagram of the architecture of the two-layer feed-forward ANN classifier. The circles that represent the nodes of the input layer are simply storage registers for the 32 sample points of a AP. The squares on the diagram represent processing elements (PE). The hidden layer contains three PEs labeled H1, H2 and H3. The output layer contains four PEs labeled with their corresponding class. The node labeled H3 is expanded to illustrate the internal components of a PE. Each PE sums the weighted inputs from the registers of the preceding layer. The result (net) serves as the argument to the activation function which in our case is the hyperbolic tangent function. The output of the activation function is stored in a register for use by the succeeding layer.

recorded from the medial styloconica of a single animal. The APs that were extracted from each trial were visually inspected to eliminate any obvious noise-corrupted APs (e.g. temporally superimposed APs). The entire ensemble was presented to each classifier (i.e. template matching, PCA and ANN).

As an additional validation of classifier performance, we compared classification results of the ANN with SAPID Tools. A subset of the ensemble was used to form a spike train which was presented to SAPID for evaluation.

In addition to improving the classification technique, we were interested in determining whether the classifiers could be used to sort APs recorded across multiple trials of the reference compounds. The classifiers were formed using derived prototypical waveforms (e.g. ensemble averages) and a small subset of raw spikes. The chronology of the individual APs was preserved to evaluate the across-trial applicability of the classifiers.

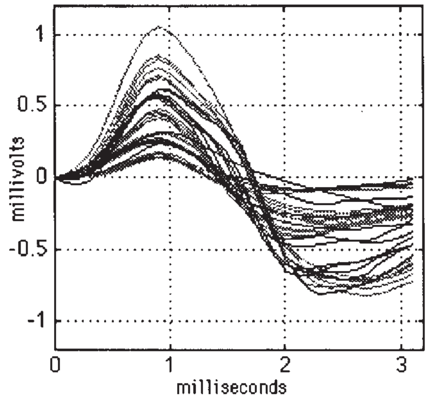

Figure 7 Plot of the ANN training set. APs plotted using dark gray lines are individual APs while the waveforms shown with light gray lines are trial averages (see Figure 9 and Appendix). These individual APs are raw waveforms that occurred during the phasic region of a recording elicited by the associated reference compound.

Further, we were interested in determining if the classifiers that were formed by data elicited from one animal could be used to classify APs that were produced by different animals. This would suggest that the reference neurons produce APs that are a characteristic of the species. Recordings from several different animals were used to evaluate the across-animal applicability of the classifiers. The APs were extracted from multiple trials and from a wide range of reference compound concentrations.

Results

Tonic response

The three AP classification methods were compared by presenting all of the individual waveforms that comprise the four reference compound ensembles to each of the classifiers. Table 1 contains the classification results for each reference neuron's ensemble. The percentage correctly classified by each classification method is shown under the corresponding column-title. The ANN technique performed extremely well in separating all four AP types. The other two techniques identified inositol and KCl APs equally well but misclassified many of the glucose and canna APs. This is not surprising considering the high degree of overlap in the distributions of peak amplitudes of these two APs (Figure 3). All of the techniques performed perfectly when applied to the inositol-evoked APs. APs produced by the KCl referenced neuron were also classified extremely well by all three methods.

Both the template matching and PCA methods showed a sharp increase in misclassifications of the glucose and canna APs. Both methods produced roughly the same percentage of errors, although the PCA was slightly weaker when classifying glucose APs. The fact that misclassification occurred between glucose and canna APs is not surprising. Referring back to Figure 3, the high degree of overlap of the peak amplitude (the maximum separation of distributions occurs here) distributions predicts problems. What is surprising is how well the ANN classifier performed in the presence of this distribution overlap.

The ANN's superior performance can be attributed to the use of the training set (see Figure 7). The training set consists of multiple prototype APs for each reference neuron. Each neuron's set of APs exhibits variability at practically each sample point. This variability within the training sets allows the ANN to 'learn' some representation of the amplitude distributions within each neuron's prototype set and between the reference neurons' sets. On the other hand, the PCA and template methods use only the average values, represented by the exemplars, to produce classification results.

Phasic response

Table 2 contains the results for APs from the phasic period. The ANN classifier outperformed both template matching and PCA when presented with phasic APs. Canna APs have a pronounced increase in amplitude during the phasic period, and therefore it is not surprising that they are often misclassified by all three methods. Again, the ANN performed best because it uses a training set that includes this variability. The glucose and KCl APs do not change shape appreciably during the phasic period and therefore are classified as well as in the tonic period.

Table 1 Comparison of results of the three classification methods: tonic-phase responses of the four chemosensory neurons of the medial styloconicum of a tobacco hornworm larva (animal H-6) to six trials of each of the four reference compounds

Reference neuron ensemble	Total no. of tonic phase APs	Template % correct	PCA % correct	ANN % correct
Inositol	182	100	100	100
Glucose	74	91	88	99
Canna	152	88	89	95
KCl	186	98	99	98

The body of the table contains the number of each type of APs presented to the classifiers and the percentage correctly classified by each of the three methods. Reference compounds and their concentrations are: 1 mM inositol, 100 mM glucose, canna extract and 2.5 M KCl.

Table 2 Comparison of classification results of the three methods on the phasic-phase of the medial sensillum of animal H-6

Neuron	Total no. of phasic APs	Template % correct	PCA % correct	ANN % correct
Inositol	67	87	87	96
Glucose	42	92	92	86
Canna	25	56	56	80
KCl	47	100	98	96

Results are tabulated as percent correctly classified.

Inter-animal applicability

A classifier that is sufficiently robust to accommodate across-animal variability would eliminate the need for generating a training set on each animal and would reduce operator bias by applying the same criteria to all data sets. The three classifiers discussed above were formed with APs from the animal designated A-1. The results for inositol recordings from five different animals were analyzed. The first animal, B-1, was recorded at the same time using the same chemical preparation of inositol as animal A-1. The next three animals (B-2, B-3 and B-4) were recorded at four different concentrations (0.3, 1.0, 1.5 and 3.0 mM) of inositol as part of a separate dose–response study. The recordings from the last animal, B-5, were obtained during another experiment using inositol at an extremely high concentration (100 mM). Recordings from animal B-5 were

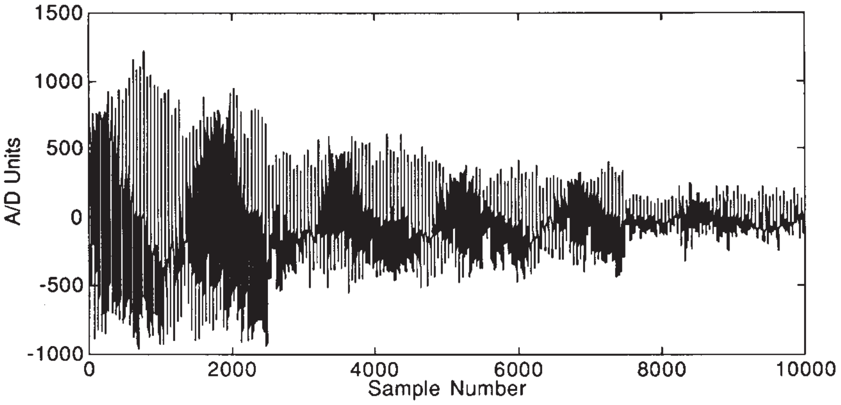

Figure 8 The spike train that was used for SAPID Tools analysis. To form the simulated 1 s sequence, 50 APs were selected from each of the four class ensembles. The first 0.25 s contains inositol APS, the second 0.25 s contains glucose APs, followed by canna and finally KCl.

obtained bilaterally, from both the left and right medial sensilla. These data thus provide responses across a wide concentration range, across animals, and from both left and right sides.

Each of the classifiers was presented all of the inositol data from all five animals. All three methods were able to correctly classify 100% of the APs in some of the trials. Several of the trials contained APs that resulted in increased levels of misclassification. The lowest percentage of correctly classified APs within any trial for the ANN classifier was 91%, for PCA it was 2% and for the template method 3%. Thus the ANN was consistently superior across all concentrations, across animals, and from both left and right sides than were the other two methods.

SAPID comparison

As an additional validation of the ANN classifier's performance, the ANN results were compared with the classification results produced by SAPID Tools. The evaluation was performed by presenting a test set consisting of 200 APs of known origin to both the SAPID classifier and the ANN classifier. The test set was composed of 50 APs from each of the four class ensembles. The 50 APs from a given class were selected by taking every *q*th AP from the class ensemble. The value of *q* was determined by dividing the total number of APs in a class ensemble by 50 (the desired number of spikes per ensemble) and then rounding off to the nearest integer value. The APs that constitute a class ensemble are stored in the order that they were extracted from the raw AP train and from the first trial to the last. Selecting every *q*th spike results in roughly equal numbers of APs from both phases of a trial and from all of the trials that constitute an ensemble.

For the SAPID evaluation an input file was generated in the required format. The file was composed of four subsequences, one for each of the four classes of APs. Each subsequence was constructed by concatenating the corresponding class APs in chronological order. The 'spike train' (see Figure 8) was formed by merging together the four subsequences.

SAPID correctly extracted 200 spikes after the proper threshold was set. SAPID requires the entry of three parameters prior to classification: (i) template deviation (TD); (ii) spike deviation (SD); and (iii) minimum number of APs (MC) to form a template. To find an optimal set of parameters requires manually searching the entire three-dimensional parameter space. To minimize the search time, the SAPID documentation recommends setting the value of SD to be 1.5 times the value of TD. The minimum number of spikes to form a template was set at four spikes. This value was found to minimize the number of classification errors and fixing this value further reduced the search process.

The value of TD was incremented through the range of integers from 5 to 20. At each increment of TD, the value of SD was incremented through a range of values centered at one and a half times TD. Any set of parameters that resulted in the formation of four templates was examined further. The parameter settings {TD = 6, SD = 10, MC = 4} resulted in the minimum percentage of APs misclassified and was selected as the optimal set of parameters. Using the optimal set of parameters, SAPID misclassified 18% of the APs in the test set. When presented with the same test set, the ANN classifier misclassified only 6% of the APs.

An experienced SAPID Tools user can employ all of the available tools to classify all of the APs. The manual classifications are subjective and require a great deal of operator time. For example, classification of six trials elicited by one of the reference compounds required over 1 h when using SAPID. The same set required a couple of minutes using the ANN classifier. One of our goals in this study was to eliminate operator bias and produce real-time results that are available to the operator as the electrophysiological recordings are being made. Our method classifies all of the extracted APs within several seconds without the need for operator input.

Discussion

We have presented a new method for separating APs generated by multiple chemosensory neurons that are mixed together within a single recording. The separation is performed by a classifier that employs a two-layer feed-forward ANN with three hidden-layer neurons. Template matching and PCA, two well-known classifiers, are discussed and applied to the same data set for comparison. The ANN classifier is flexible enough to handle both tonic and phasic region APs and is more robust than the conventional methods in dealing with across-animal trials.

The conventional approach in classifying APs within a single-channel recording is to focus on the tonic phase of the recording. This avoids the problem of changing AP shape that is observed during the phasic response portion of an AP train. The ANN classifier performed better than either the PCA or the template method when classifying APs that occurred during the tonic phase. By following the conventional approach of focusing only on the tonic phase and ignoring the phasic response some information is lost which may be important. For example, Dethier (1973) proposed that feeding decisions by blowflies could be made on the basis of this phasic information. Therefore we attempted to develop a method that would utilize all of the available information. Accordingly, the ANN training set was augmented with phasic APs (inositol and canna) to form a classifier that was an improvement over the other two methods when classifying both phasic and tonic APs. The performance of the resulting ANN was relatively robust when challenged with phasic APs. This flexibility in forming the ANN's training set, in conjunction with the ANN's ability to learn amplitude distributions rather than simply averages, are two significant advantages the ANN possesses over the PCA and template methods.

The robustness of the ANN is also demonstrated by its superior applicability across animals and concentrations. The ANN formed with data from one animal can classify data from other animals (and at other stimulus concentrations) much more reliably than can the PCA and template techniques. This could be an important feature to consider in the design of a simple and quick classification process. Once the ANN is trained, it is ready to classify new data (from representative animals of the same species) without retraining. These results also suggest a biologically important corollary, namely that the unique AP shapes produced by these chemosensory neurons are characteristic of the species.

We anticipate that further improvements in precision may be achieved by using temporal information and iterative processing.

References

deBoer, G. and **Hanson, F.E.** (1987) *Differentiation of role of chemosensory organs in food discrimination among host and non-host plants by larvae of the tobacco hornworm,* Manduca sexta. Physiol. Entomol., 12, 387–398.

Dethier, V.G. (1973) *Electrophysiological studies of gustation in lepidopterous larvae. II. Taste spectra in relation to food plant discrimination*. J. Comp. Physiol., 82, 103–134.

Frazier, J.L. and **Hanson, F.E.** (1986) *Electrophysiological recordings and analysis of insect chemosensory responses*. In Miller, J.R. and Miller, T.A. (eds), Insect/Plant Interactions. Springer-Verlag, New York.

Fukunaga, K. (1990) Statistical Pattern Recognition. Academic Press, San Diego, CA.

Hanson, F.E., Kogge, S. and **Cearley, C.** (1986) *Computer analysis of chemosensory signals*. In Birch, M., Payne, T. and Kennedy, C. (eds), Mechanisms of Insect Olfaction. Oxford University Press, Oxford.

Peterson, S.C., Hanson, F.E. and **Warthen, J.D.** (1993) *Deterrence coding by a larval* Manduca *chemosensory neuron mediating rejection of a non-host plant,* Canna generalis *L*. Physiol. Entomol., 18, 285–295.

Roessingh, P., Städler, E., Baur, R., Hurter, J. and **Ramp, T.** (1997) *Tarsal chemoreceptors and oviposition behavior of the cabbage root fly (*Delia radicum*) sensitive to fractions and new compounds of host-leaf surface extracts*. Physiol. Entomol., 22, 140–148.

Smith, J.J.B., Mitchell, B.K., Rolseth, B.M., Whitehead, A.T. and **Albert, P.J.** (1990) *SapidTools: microcomputer programs for analysis of multi-unit nerve recordings*. Chem. Senses, 15, 253–270.

Strang, G. (1988) Linear Algebra and its Applications. Saunders/HBJ, Orlando, FL.

Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T. and **Alkon D.L.** (1988) *Accelerating the convergence of the back-propagation method*. Biol. Cybernet., 59, 257–263.

Wasserman, P.D. (1989) Neural Computing. Wiley, New York.

Waldbauer, G.P. (1962) *The growth and reproduction of maxillectomized tobacco hornworms feeding on normally rejected non-solanaceous plants*. Entomol. Exp. Appl., 5, 147–158.

Wheeler, B.C. (1999) *Real time techniques for automatic discrimination of single units*. In Nicolelis, M. (ed.), Methods for Neural Ensemble Recordings. CRC Press, in press.

Zacharuk, R.Y. and **Shields, V.** (1990) *Sensilla of immature insects*. Annu. Rev. Entomol., 36, 331–354.

Zurada, J.M. (1992) Introduction to Artificial Neural Systems. West, St Paul.

Accepted June 3, 1998

Appendix: Artificial Neural Network—Development of Current Configuration

Several characteristics are used to distinguish the architecture of an ANN: (i) feed-forward or recurrent architecture; (ii) the number of layers of processing elements (note that the set of input nodes of a feed-forward ANN is not counted as a layer because the nodes provide only storage, not processing); and (iii) method of training used to alter its internal state, i.e. either supervised or unsupervised training. Supervised training employs a training set that includes the prototype inputs to be learned and the desired outputs that the ANN must assign to each prototype. The training set for an unsupervised ANN contains only prototypes; the ANN self-organizes to produce outputs for each group of prototypes. In our case the *prototype* APs are formed by averaging all APs that are produced by a reference neuron during a single trial. The *exemplar* APs, which are discussed throughout the paper, are formed by averaging all of the prototype APs associated with a given reference neuron.

Feed-forward ANNs, which are composed of no more than two hidden layers, have been shown to be universal nonlinear approximators, capable of approximating any relation between input and output spaces (Zurada, 1992). Currently there are no rules available to predetermine the number of hidden layers or the number of neurons per hidden layer that ultimately will be necessary to adequately implement an input–output relation. Thus, large numbers of trials were necessary to arrive at an acceptable architecture. The number of nodes that make up the input layer is fixed by the number of samples that constitute an AP (32 in this case), and the number of output neurons also is fixed by the desired number of categories (four in this case, representing the four unique AP classes).

Various two-layer and three-layer feed-forward ANNs were tested during the development process. The number of neurons per hidden layer was varied from 3 to 100 neurons. In theory the more neurons per layer, the more likely the ANN is to learn the given relationship. On the other hand, increasing the number of neurons increases the amount of time required to achieve some minimum level of error used to terminate the learning process. The architecture that was finally selected consisted of a two-layer ANN with three hidden-layer neurons. Figure 6 is a diagram of the ANN architecture (for simplicity, only 3 of the 32 input nodes are shown) that was determined to be the best for our classification problem.

It is important to consider the settings of the various learning parameters when designing the ANN. The list of parameters includes: (i) learning coefficient (η); (ii) momentum factor (μ); (iii) range of initial connection weight values; (iv) epoch size; and (v) learning rate transition. Large values of η cause large steps to be taken during the learning process while small values result in small changes in the interconnection weights. It might seem desirable to routinely use large steps to expedite the learning process; however, if the class separation is very narrow, large values will cause the ANN to oscillate in the area of the separation, never minimizing the overall error level enough to terminate the training process. Setting the value of η very small will ensure that the ANN can distinguish between classes that are very similar, but it may take an unacceptably long time to find the boundary between the classes. These parameters appear to be application-specific and require searching the domain of potential values to find the best settings.

To determine a nearly optimal set of parameter values, the parameter space was partitioned and the end points of the subsets were used as parameter settings. The regions that resulted in the lowest squared-error and fastest training time were further subdivided, and the end points of each subregion were used as training parameters. The process was repeated at increasingly finer levels until acceptable levels of squared-error and training time were achieved. As an example from the current application, the learning rate coefficient (η) is a continuous variable with range [0,1]. One method of determining an optimal value for η is partitioning the range into 10 equal subintervals while holding all other parameters constant. The interval between 0 and 0.1 resulted in a trained network. The subinterval (i.e. [0, 0.1]) was partitioned into 10 equal subintervals. The process of subdividing was continued until no ANN performance improvement was achievable. The final value of η for our application was 0.039 for the hidden-layer neurons and 0.019 for the output-layer neurons. The time consuming search for an optimal value of η can be eliminated by using a modified back-propagation algorithm that constantly adjusts the value of η as training progresses (Vogl, 1988). Use of this algorithm not only eliminated the tedious task of searching for an optimal value of η but also was found to converge faster when compared with the standard back-propagation algorithm employing a fixed value for η.

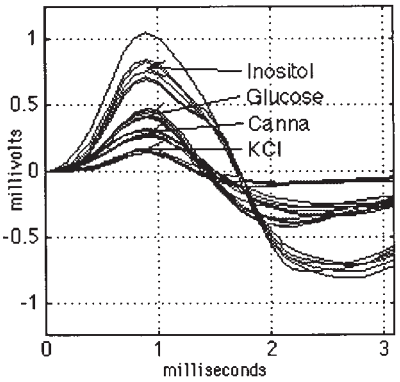

Figure 9 The individual prototype APs that were used as the original training set for the ANN classifier. Each of the prototypes is an average AP of a 1 s trial of the associated reference compound.

Our system was developed using C-code modules with parameter modifications expedited with NeuralWare (Aspen Technologies, Pittsburgh, PA) to graphically display the effects of parameter modifications on the ANN. We then moved it to MATLAB (MathWorks, Boston, MA), which allowed us to implement the entire system, from data acquisition (via .mex files) through behavior prediction, within a single cross-platform environment that also supports a graphical user interface (GUI) to make the operation user friendly. End-users are not limited to a few predefined displays, but can use built-in MATLAB functions to manipulate and display data in any convenient manner. MATLAB is cross-platform, and therefore this data analysis system can be used with any hardware platform for which a MATLAB run-time program is available.

In addition to the values of the various parameters employed in training, the content of the training set is important if the ANN classifier is to be maximally effective. The training set must contain a sufficient number of examples to permit the ANN to learn the distributions associated with each AP type. However, as the size of the training set grows, the training time will increase significantly. To minimize training time while including the inherent variability of each AP type, we averaged all of the APs from a given trial to form a single 'prototype AP' for that trial (see Figure 9). Training the ANN with these prototypes resulted in very good classifications within the tonic region of an AP train, but it did not perform as well during the phasic portion of the same AP train. However, by adding individual APs (see Figure 7) that occurred early in the phasic region to the original training set, the modified ANN correctly classified all phasic inositol APs compared with only 65% by the original ANN.