ELSEVIER

Journal of Nuclear Materials 264 (1999) 141–154

journal of nuclear materials

Nuclear fuel pellet inspection using artificial neural networks

Shahla Keyvan [*], Xiaolong Song, Mark Kelly

Department of Nuclear Engineering, University of Missouri-Rolla, Rolla, Missouri 65409, USA

Received 19 August 1997; accepted 22 June 1998

Abstract

Nuclear fuel must be of high quality before being placed into service in a reactor. Fuel vendors currently use manual inspection for quality control of fabricated nuclear fuel pellets. In order to reduce workers' exposure to radiation and increase the inspection accuracy and speed, the feasibility of automation of fuel pellet inspection using artificial neural networks (ANNs) is studied in this paper. Three kinds of neural network architectures are examined for evaluation of the ANN performance in proper classification of good versus bad pellets. Two supervised neural networks, back-propagation and fuzzy ARTMAP, and one unsupervised neural network called ART2-A are applied. The results indicate that a supervised ANN with adequate training can achieve a high success rate in classification of fuel pellets. © 1999 Elsevier Science B.V. All rights reserved.

1. Introduction

The quality of fabricated nuclear fuel pellet must be checked before the pellet is inserted into zirconium cladding fuel rod in order to minimize the potential for pellet–cladding interaction (PCI). Fuel pellet defects such as crack may increase the risk of PCI [1]. Improper shape, e.g., improper 'cupping' of the pellet end, may allow distortion and swelling due to uneven rates of heat generation within the pellet [2,3]. PCI could lead to a fuel element failure, allowing radioactive fission products to enter the reactor coolant. A high coolant fission product concentration (0.5–1.0 μCi I/g of coolant) from leaking fuel rods may necessitate an unscheduled reactor shutdown for corrective action [4,5]. An unscheduled reactor shutdown may cost a utility an amount approaching three million dollars per week, making fuel integrity an important economic issue.

Westinghouse currently claims [6] that 99.995% of their fuel rods are leak-free for their in-core service life. The importance of defect-free fuel is emphasized when considering a core containing 51 000 fuel rods, even a 99.995% success rate means that one should expect to find two or three defective rods during each refueling outage. Union Electric's Callaway nuclear power plant currently uses Westinghouse fuel and has found, on an average, 4.4 defective fuel rods per refueling outage [5].

The current practice of pellet inspection by humans is tedious, subject to inconsistencies, and prone to error. In addition, manual inspection is cumbersome since the inspector must keep the pellet at arm's length and wear glasses to protect the lens of the eye from radiation. Fuel pellet inspection is complicated by the allowance of some small degree of chipping and cracking. The inspection process, as currently performed, is essentially a judgment call.

Quality control in fuel fabrication at Asea Brown and Boveri Combustion Engineering Nuclear Fuel (ABB) relies on human inspection of the manufactured fuel pellets before insertion into the zirconium fuel tubes. The pellets are taken from the pellet sizing machine, dumped onto a rack, shaken into rows and then viewed as a group. The entire group is rotated 90° four time to provide the inspector with a 360° view of each pellet. The sides of the pellet are examined for cracks, chips, and unusual markings, e.g., water stains and machine banding. The ends of the pellets are checked for defects if edge misalignment is noticed on the pellet–pellet interface when viewing the pellets from the side.

Both Westinghouse [7] and Siemens [8] use manual inspection to check for fuel pellet defects. Siemens

[*] Corresponding author.

PII: S 0 0 2 2 - 3 1 1 5 (9 8) 0 0 4 6 4 - 4

attempted to use machine vision to automate the process in 1982, but abandoned their method because it was too slow.

Framatome currently manufactures fuel assemblies and does not itself manufacture uranium dioxide fuel pellets. The pellets are purchased from Siemens and are loaded into fuel rods at the Framatome Commercial Nuclear Fuel Plant. The pellets are inspected prior to shipment by Siemens, and inspected manually at the Framatome plant before insertion into the fuel rods.

General Electric (GE) currently uses manual inspection [9]. GE attempted developing an automatic sorter and inspection system using laser techniques in 1984. The inspection system pushed the pellets through in a single stream and scanned the pellet circumference with a laser. An algorithm developed by GE analyzed the reflected beam. GE obtained three patents, named, Tray Loader in 1984, Automated Inspection System in 1985, and Optical Inspection System in 1985. However, their system is currently not in use due to a high good pellet rejection rate, high cost, and maintenance difficulties. GE claims to have spent 30 man years developing their system.

The objective of this work is to evaluate the feasibility of using Artificial Neural Networks (ANNs) for pellet image classification. This paper presents the results of the classification performance of three ANNs, namely, backpropagation, fuzzy ARTMAP, and adaptive resonance theory ART2-A paradigm. The above-mentioned networks are selected because of their general acceptance by the community, pattern recognition capabilities, and suitability of the learning algorithm to be automated. In the following sections, ANNs, data acquisition, and corresponding inspection results are described.

2. Artificial neural networks

ANNs are biologically inspired algorithms which attempt to simulate certain aspects of human intelligence, particularly the ability to learn and generalize. ANNs are often used for pattern recognition and classification type problems which do not readily yield themselves to solution by explicit sequential algorithms. This type of problem may require complex data translation where no pre-defined mapping function exists, or may require a 'best guess' as an output when presented with noisy inputs [10].

The specific characteristics of an ANN are a result of the network paradigm utilized. The network paradigm is specified by the network architecture and neuro-dynamics. The network architecture defines the arrangement of processing elements (PEs) and their interconnections. This establishes which PEs are interconnected, inputs to and outputs from PEs, the group or layers of PEs, and how the information flows in the network. For example in a feedforward network, the information will flow strictly from the input to the output (see Fig. 1).

The PEs (also called neurons) have a number of inputs which are modified by adaptive coefficients (weights) and generate an output signal (see Fig. 2). Neuro-dynamics specifies how the inputs to the PE are going to be combined together, and what type of function or relationship is going to be used to develop the output, and how the weights are going to be modified.

The learning mechanism which handles modifications to the weights and any other organization of the network can be classified under supervised learning, unsupervised learning, and self-supervised (reinforcement) learning. Supervised learning takes place when the network is trained using pairs of inputs and desired outputs. ANNs learn to associate presented outputs with presented inputs by adjusting the weights on the neuron–neuron connections. They may be able to produce the correct outputs when presented with new inputs after being properly trained. In unsupervised learning, inputs are entered and the network is able to organize its own categories. Self-supervised learning adds the feedback to unsupervised learning to correct errors in the pattern recognition process.

The spectrum of different paradigms is quite extensive. For example, the network architectures range from simplistic one-layer to hierarchical networks. In addition, there are a large number of algorithms for modification of the adaptive coefficients. The various existing paradigms have their limitations and strengths, hence one must identify the suitable application areas for which they lend themselves.

It is necessary to explain some of the ANN related terminologies used in this work.

Features: A feature is a unique measurable attribute of a pattern. A feature vector is a collection of features that completely describe a pattern. This feature vector is presented to the ANN as input during the training process.

Feature space: Feature space is a set of feature values that determines a pattern class, from which the feature vector is determined.

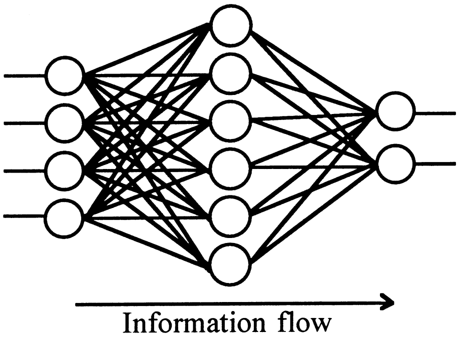


Fig. 1. A feedforward network.

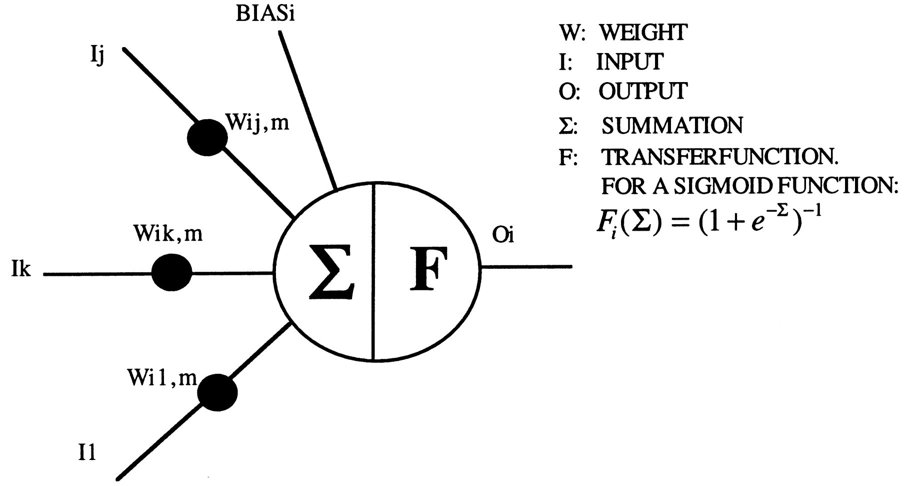


Fig. 2. A processing element.

Generalization: Generalization is the ability of a network to model the behavior of a specific relationship. This capability can be measured by the performance of the network for examples outside of the training set. According to Tishby et al. [11] "various researchers have shown that there is often little connection between the error at the end of the training process, namely the training error, and the network's ability to generalize outside of the training set".

Vigilance parameter: Vigilance parameter $\rho (0 \leqslant \rho \leqslant 1)$ is used by the adaptive resonance theory family of neural networks in order to determine if an input pattern is 'similar' to the existing prototype. The vigilance parameter (ρ) sets the criterion for matching, under the exact same conditions, lower vigilance leads to coarser categories and higher vigilance to finer categories. The optimum ρ is obtained by trial and error.

3. Data acquisition

ABB, a nuclear fuel fabrication plant in Hematite, Missouri, proved the sample nuclear fuel pellets for this study. A total of 252 pellets with various defects were selected for this research work. Each pellet was photographed four times at rotations of 90°. This created four images labeled a, b, c, and d. Although pellets are randomly selected in this analysis, the image labeled a may appear more often than others. This is because when taking the photos from each pellet, the first picture was taken directly from the defective area, hence image labeled a appears quite often in our sample images for analysis. The resultant black and white negatives were scanned into the computer in 256 grayscale mode. The grayscale intensity value of 255 is assigned to the lightest (white) pixel and the intensity value of 0 is assigned to the darkest (black) pixel. Measurements on scanned pellet images revealed that at a resolution of 150 × 150 pixels the pellets within the scanned image had dimensions of 56 × 45 pixels. It was determined that each image should be cropped so that the entire pellet surface would be contained within four images, with little overlap. A crop size of 56 × 33 allowed for the trimming of shadows on the extreme pellet circumference while covering the entire pellet [12]. Fig. 3 shows several kinds of pellet defects and their scanned images.

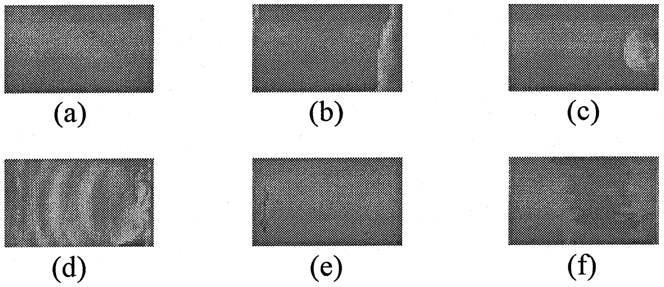


Fig. 3. Fuel pellet images: (a) Good pellet; (b) Large end defect; (c) Chip; (d) Rippled surface; (e) Circumferential Crack; (f) Banded.

Photo images can be processed by assigning color scale values to each pixel and representing each pixel as an input to an empirical modeling tool, such as an ANN. This approach would result in a very large number of inputs. In the application of ANNs, a large number of input data may cause complications for proper pattern classification [13]. One technique, feature extraction, helps alleviate such a problem. Feature extraction is analogous to data compression. By reducing the dimensionality of an image to a few salient features, one avoids irrelevant (noisy) input, and increases the speed of classification and pattern recognition which in turn increases the inspection speed.

Multiple features whose values are expected to be sensitive to the presence of defects within a pellet image were selected for extraction from the pellet image files. Table 1 lists the 39 features extracted from each image file. In this work the Matlab software with image processing toolbox were used for the purpose of converting the image files into ASCII files containing grayscale integer data. Mean grayscale intensity, grayscale intensity variance, grayscale intensity skew (a measure of histogram asymmetry about its mean), and grayscale intensity kurtosis (an indication of departure from a normal distribution) provided the grayscale intensity information on each image [14,15]. The mean, variance, skew, and grayscale intensity kurtosis are described in Eqs. (1)–(4), respectively. The variance of grayscale intensity is least for a good pellet and greatest for a rippled surface pellet. The mean intensity for the crack image is

Table 1
List of features extracted from each pellet image

No.	Parameter	Description
1	m	Mean grayscale intensity of matrix
2	v	Grayscale intensity variance of matrix
3	s	Grayscale intensity skew
4	k	Grayscale intensity kurtosis
5	pa	Number of pixels above intensity 130
6	pc	Number of pixels between intensity 80 and intensity 130
7	pb	Number of pixels below intensity 80
8	x1a	1st moment of inertia, pixels above intensity 130, binary
9	y1a	1st moment of inertia, pixels above intensity 130, binary
10	x1b	1st moment of inertia, pixels below intensity 80, binary
11	y1b	1st moment of inertia, pixels below intensity 80, binary
12	x2a	2nd moment of inertia, pixels above intensity of 130, binary
13	y2a	2nd moment of inertia, pixels above intensity of 130, binary
14	x2b	2nd moment of inertia, pixels below intensity of 80, binary
15	y2b	2nd moment of inertia, pixels below intensity of 80, binary
16	x1	1st moment of inertia, grayscale
17	y1	1st moment of inertia, grayscale
18	x2	2nd moment of inertia, grayscale
19	y2	2nd moment of inertia, grayscale
20	x1inv	1st moment of inertia, inverted grayscale
21	y1inv	1st moment of inertia, inverted grayscale
22	x2inv	2nd moment of inertia, inverted grayscale
23	y2inv	2nd moment of inertia, inverted grayscale
24	x1ano	1st moment of inertia, pixels above a threshold, binary, alternative origin
25	y1ano	1st moment of inertia, pixels above intensity 130, binary, alternative origin
26	x1bno	1st moment of inertia, pixels below intensity 80, binary, alternative origin
27	y1bno	1st moment of inertia, pixels below intensity 80, binary, alternative origin
28	x2ano	2nd moment of inertia, pixels above intensity 130, binary, alternative origin
29	y2ano	2nd moment of inertia, pixels above intensity 130, binary, alternative origin
30	x2bno	2nd moment of inertia, pixels below intensity 80, binary, alternative origin
31	y2bno	2nd moments of inertia, pixels below intensity 80, binary, alternative origin
32	x1no	1st moment of inertia, grayscale, alternative origin
33	y1no	1st moment of inertia, grayscale, alternative origin
34	x2no	2nd moment of inertia, grayscale, alternative origin
35	y2no	2nd moment of inertia, grayscale, alternative origin
36	x1invno	1st moment of inertia, inverted grayscale, alternative origin
37	y1invno	1st moment of inertia, inverted grayscale, alternative origin
38	x2invno	2nd moment of inertia, inverted grayscale, alternative origin
39	y2invno	2nd moment of inertia, inverted grayscale, alternative origin

slightly lower than the mean for a good image. This is to be expected due to the effect of the presence of the dark pixels (hence, lower intensity values on the scale of 0–255) in the crack portion of the image. The mean intensity for the rippled surface pellet image is considerably higher due to many light areas (hence, higher intensity values on the scale of 0–255) on the surface of the pellet.

$$\text{mean} = \frac{\sum i}{n}, \tag{1}$$

$$\text{variance} = \frac{\sum (i-m)^2}{n}, \tag{2}$$

$$\text{skew} = \frac{\sum ((i-m)/\sigma)^3}{n}, \tag{3}$$

$$\text{kurtosis} = \frac{\sum ((i-m)/\sigma)^4}{n-3.0}, \tag{4}$$

where, the summation is performed over all image pixels, and i is the pixel intensity, n the total number of pixels, m the mean, and σ the intensity standard deviation.

Other geometrical moment parameters were selected for extraction in order to provide information about defect type, orientation, and size [14]. For example, the first moment of inertia (center of mass) has the potential to provide information about the presence or absence of defects. It would be expected that a chip free pellet image center of mass calculation would yield coordinates that are roughly in the center of the pellet image. When large defects such as chips, shears, or ripples appear, the center of mass location would slowly migrate. This is why the first moment of inertia is selected as a feature. Geometrical second moment of inertia is selected as a feature because it has the potential to provide information about crack orientation.

The first and second moment of inertia about the x-axis shown in Fig. 4 are calculated from Eqs. (5) and (6), respectively. Note, the first moment of inertia (center of mass) is calculated using the pixel intensity as the 'mass' in the standard center of mass equation.

$$x_1 = \frac{\sum (yi\,\mathrm{dA})}{\sum (i\,\mathrm{dA})}, \tag{5}$$

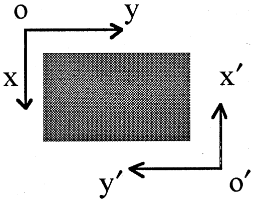


Fig. 4. Pellet image coordinate system.

$$x_2 = \sum (y^2 i\,\mathrm{dA}), \tag{6}$$

where the summation is performed over all image pixels with y being the pixel number from the origin on the y-axis in Fig. 4, i representing the pixel intensity, and $\mathrm{dA} = 1$.

The feature extraction programs use a Cartesian coordinate system with the origin taken as the upper left hand corner of the image with the positive x-axis emanating downward (along the rows) from the origin O. The positive y-axis emanates toward the right (along the columns) from this origin. Other features are extracted at new origin O′ (see Fig. 4). Use of an alternate origin is suggested by the necessity of creating a parameter set that is sensitive to defects in any location within the pellet image. The variability in the possible location of defects suggests that second moment of inertia feature variables be examined using two origins.

Parameters calculated using pixel grayscale intensity are biased to favor the contribution of the lighter pixels due to white being represented as 255 and black as 0. For inverted features, each pixel's intensity was inverted by subtracting it from 255, the maximum intensity.

Parameter calculation with inversion is suggested due to the fact that dark defects such as cracks and chips, when inverted, have more influence on first and second moment parameters. For this reason, parameters were also calculated using inverted intensities (255 − intensity).

The use of thresholds in calculating moments of inertia was motivated by a desire to increase the sensitivity of these variables to the presence of defects. Image pixel intensities were examined to determine appropriate values for the upper and lower pixel thresholds used for many of the routines in the feature extraction programs. A lower threshold intensity (= 80) and an upper threshold in the intensity region 130–145 was determined to be appropriate [12].

Figs. 5–7 show the pattern created by the trend line connecting these 39 feature parameters for good pellet, crack pellet and rippled surface pellet, respectively. These patterns clearly illustrate how the extracted features change from pellet to pellet, hence creating a unique pattern for each type of defect. Although the features calculated here do not uniquely identify object, they do reduce the number of inputs from 1848 inputs (56 × 33 pixels) per image to less than 40, more than a 98% reduction in input file size.

4. Application of backpropagation and fuzzy ARTMAP

Backpropagation Network (BPN) [16] and fuzzy ARTMAP [17,18] are supervised neural networks. A supervised network is one that has sample input and associated correct outputs given to it simultaneously as training examples so that error terms can be calculated and weights adjusted.

Backpropagation is a multilayered feedforward network. As shown in Fig. 1, it consists of one input layer, one output layer and a minimum of one hidden layer. The learning takes place using a pre-defined set of input–output example pairs and a two phase propagate–adapt cycle. The input pattern is applied as stimulus to the first layer of the network units. Sigmoid function is commonly used (see Fig. 2) for output function of a unit in BPN application. The pattern is propagated through each upper layer until an output is generated. This output is then compared to the desired output, and the error signal is computed for each output unit. The error signal is propagated backward from the output layer to each node in the intermediate layers. However, each unit in the intermediate layer receives only a portion of the total error signal, based on the relative contribution the unit has made to the original output. This process repeats until each node in the network has received an error signal that describes its relative contribution to the total error. Based on the error signal received, connection weights are updated by each unit to cause the network to converge toward a state that allows all the training pattern to be encoded. The learning rule used in backpropagation is called the Generalized Delta Rule. The Generalized Delta Rule minimizes the root mean square (RMS) error between the actual predicted output and the desired output by modifying network weights. Suppose we have a three layer network. When the pth pattern is applied to the input layer, the net input to the jth node in the hidden layer is: $\mathrm{net}_j = \sum_{i=0}^{n} W_{j,i}x_{p,i}$, where

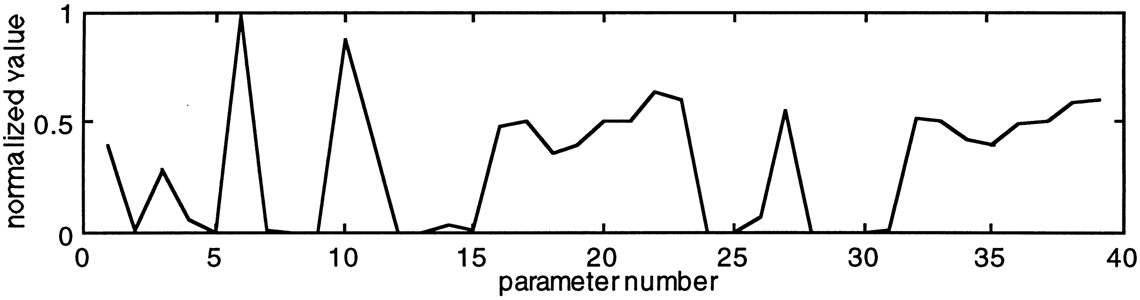


Fig. 5. Good fuel pellet feature parameters.

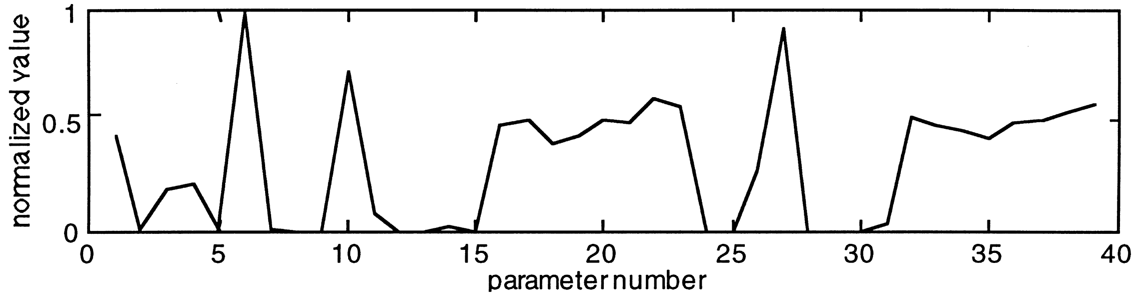


Fig. 6. Feature parameters of fuel pellet with crack defect.

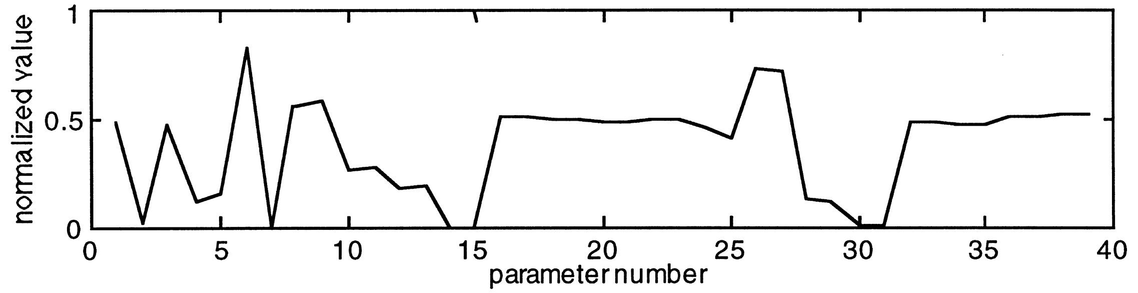


Fig. 7. Feature parameters of fuel pellet with rippled surface.

$w_{j,i}$ is the weight value of the connection from the ith input node to the jth hidden layer node. $x_{p,i}$ is the input value to the ith node in the input layer for pattern p. The output of the jth node in the hidden layer is

$$x_{p,j} = f\left(\sum_{i=0}^{n} w_{j,i} x_{p,i}\right),$$

where f is generally a sigmoid function.

Thus the net input to the kth node of the output layer $\mathrm{net}_k = \sum_j w_{k,j} x_{p,j}$, where again $w_{k,j}$ is the weight value of the connection from the jth hidden layer node to the kth output layer node. The output of the kth node of the output layer is

$$o_{p,k} = f\left(\sum_{j} w_{k,j} x_{p,j}\right),$$

where f is generally a sigmoid function. Suppose the desired output of the kth node of output layer is $d_{p,k}$, the corresponding squared error is

$$l_{p,k}^2 = |d_{p,k} - o_{p,k}|^2.$$

The total error (for the pattern p) is given by $E_p = \sum_k (l_{p,k})^2$. One way to minimize E_p is to use the gradient descent method. Since $o_{p,k}$ depends on the network weights, E_p is also a function of the network weights. According to the gradient descent method, the direction of weight change of w should be in the same direction as $-\partial E_p/\partial w$. We calculate the value of $\partial E_p/\partial w_{k,j}$ for each connection from the input layer to the output layer. Similarly, we calculate the value of $\partial E_p/\partial w_{j,i}$ for each connection from the input layer to the hidden layer. The connection weights are then changed by using the values so obtained [16].

The main advantages of BPN are its generalization capability and noise tolerance. Generalization means that, given several different input vectors all belonging to the same class, a BPN will learn to keyoff of significant similarities in the input vectors and irrelevant features will be ignored. If a BPN is inadequately trained on a particular class of input vectors, subsequent identification of members of that class may be unreliable. Also, there are no clear guidelines for choosing the number of neurons in the hidden layer, and even the number of hidden layers required. Generally, decision on the number of hidden layer and neurons are made based on trial and error seeking as few hidden-layer units as possible without losing the accuracy of the system. One hidden layer with optimum number of neurons may be adequate in many applications. In this first attempt in examining BPN performance for this application, it was decided to first use one hidden layer, but optimize the number of neurons in that hidden layer by trial and error. To fine tune the BPN performance in future application, the effect of more hidden layer will be examined. Developing a BPN software program is not challenging and this has led to its popularity. Commercial software are available for BPN network. The software 'Matlab' can also be used for developing BPN software. In this application, the software for Backpropagation neural network was obtained from University of Nevada, School of Medicine. This BPN utilizes a Sigmoid function for a unit transfer function. Also, due to fast convergence in our application, there was no need to use a bias node. The training time using a Power Macintosh 8500/120 was less than 5 min for a total of 120 training set with each set containing 39 inputs and training process of 5000 epochs (at which the output RMS error stabilized at a value of 0.31).

Fuzzy ARTMAP network is selected in this work due to its capability of handling non-stationary stochastic signal as well as supervised learning. Fuzzy ARTMAP has a more complicated structure than a BPN network. Although, it incorporates the same learning strategy as a BPN, i.e., the supervised method, it has extra features like self-organization and self-stabilization which a BPN lacks.

Adaptive Resonance Theory (ART) [1] represents a family of ANNs which self-organize categories in response to arbitrary sequences of input patterns in real time for pattern recognition. A class of these networks, called ART 1, which is unsupervised, can be used only for binary patterns. ART 2, which is also an unsupervised class, responds to both binary and analog patterns. The class 'fuzzy' ART is similar in architecture to ART 1, however, fuzzy operators are added in order to handle analog patterns without losing the advantages of ART 1 architecture. The class ARTMAP ('predictive' ART) is built upon the basic ART designs, while incorporating supervision in the learning process. It has the capability of handling non-stationary stochastic signal as well as supervised learning.

ART network architecture consists of two major subsystems called the attentional subsystem and orienting subsystem. The attentional subsystem as shown in Fig. 8 consists of two layers of PEs, where all the processing of input pattern takes place. The two layers of the attentional subsystems are the F1 layer (input layer) and the F2 layer (output layer). The neurons in both layers are fully interconnected. Patterns of activity that develop over the nodes in the two layers of the attentional subsystems is called the short-term memory (STM) traces because they exist only in association with a single application of an input vector. The weights associated with the bottom–up and top–down connections between F1 and F2 layers are called long-term memory (LTM) traces

[1] All ART neural network software programs and feature extraction programs are written in-house. The reader should refer to Refs. [18,19] for complete algorithm to develop an ART software program.

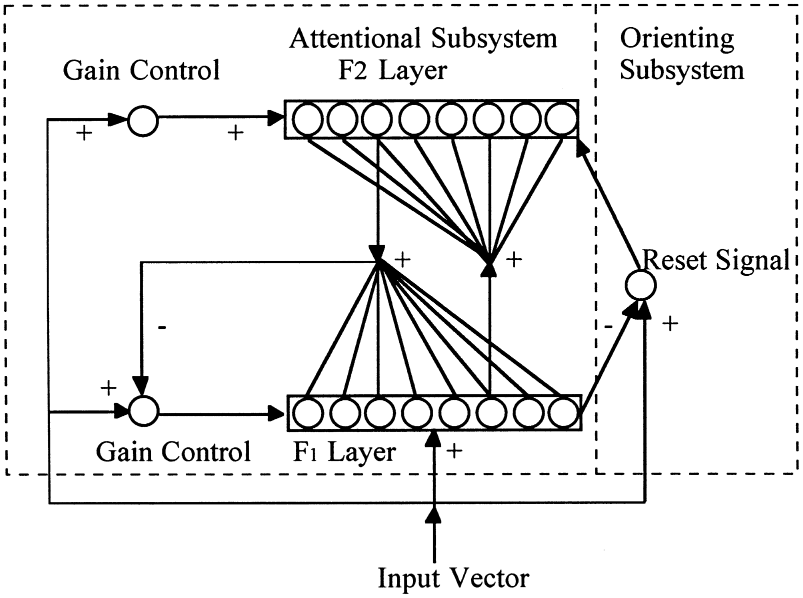


Fig. 8. General ART Network.

because they encode information that remains a part of the network for an extended period of time. The Gain Control in ART networks is a mechanism which acts to adjust overall sensitivity to the input patterns and to coordinate the difficult functions of the ART subsystems.

The orienting subsystem acts as a novelty detector. The orienting subsystem in ART networks is responsible for controlling the fineness or coarseness of the pattern being recognized. It has only two input signals and one output signal. The two inputs are the input data pattern and the overall activity in F1. The single output of the orienting system goes to F2 as a reset wave (see Fig. 8). The resonance and reset is accomplished in the orienting subsystem through a very important parameter called the vigilance 'ρ'. The vigilance is a parameter which points to the degree to which the system discriminates between different classes of input patterns. It can have any decimal value between 0 and 1. For a given set of patterns to be classified, a large value of ρ will result in finer differentiation between patterns than a smaller value of ρ.

Fuzzy ARTMAP (see Fig. 9) is built with a pair of fuzzy ART modules, Fuzzy ARTa and Fuzzy ARTb. These networks are designed as generalized ART except the set-theoretic operation of intersection ($\cap$) is replaced by the fuzzy set theory conjunction($\wedge$). This fuzzy set operator makes fuzzy ARTMAP capable of handling both analog and binary data and makes the ART neurodynamic equations much simpler. The setup description of the two modules of fuzzy ARTMAP makes the understanding of the complete network much easier. As seen in Fig. 9, the two fuzzy ART modules a and b are connected with an inter-ART module called the Mapfield Layer. The Mapfield associates the fuzzy ARTa recognition categories with the fuzzy ARTb recognition categories. If a match occurs at the Mapfield layer then learning takes place. If a mismatch occurs, then the match tracking process is initiated. Match tracking is a method of forcing the mismatched fuzzy ARTa winning node to reset by incrementing ρ_a (vigilance parameter) by a small value, such that the current winning node fails the resonance condition in the subsequent trial.

During the supervised learning the Fuzzy ART$_a$ module receives a stream (a) of input patterns and Fuzzy ART$_b$ receives a stream (b) of input patterns, where (b) is the correct prediction, given (a). These modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time.

The Fuzzy ART$_a$ complement coding preprocessor transforms the M$_a$-vector a into a 2M$_a$-vector $A = (a, a^c)$ at the Fuzzy ART$_a$ field F_0^a, where a^c is the complement of vector a (i.e., $a_i^c = 1 - a_i$ for each normalized element i of the vector a). A is the input vector to Fuzzy ART$_a$ field F_1^a. The input to F_1^b, like Fuzzy ART$_a$, is the 2M$_b$-vector $B = (b, b^c)$, where b^c is the complement of vector b. For each input (I) in the 0–1 range presented to the network, the net value at the output is compared as

$$\mathrm{net}_j(I) = \frac{|I \wedge W_j|}{\alpha + |W_j|},$$

where $\alpha = 0.00001$, I is input vector, W_j are adaptive weights or LTM traces (initialized to 1), $\wedge$ is the fuzzy And operator.

The maximum of $\mathrm{net}_j(I)$ is selected. For this node J, it is checked if the vigilance (ρ) parameter is met, that is

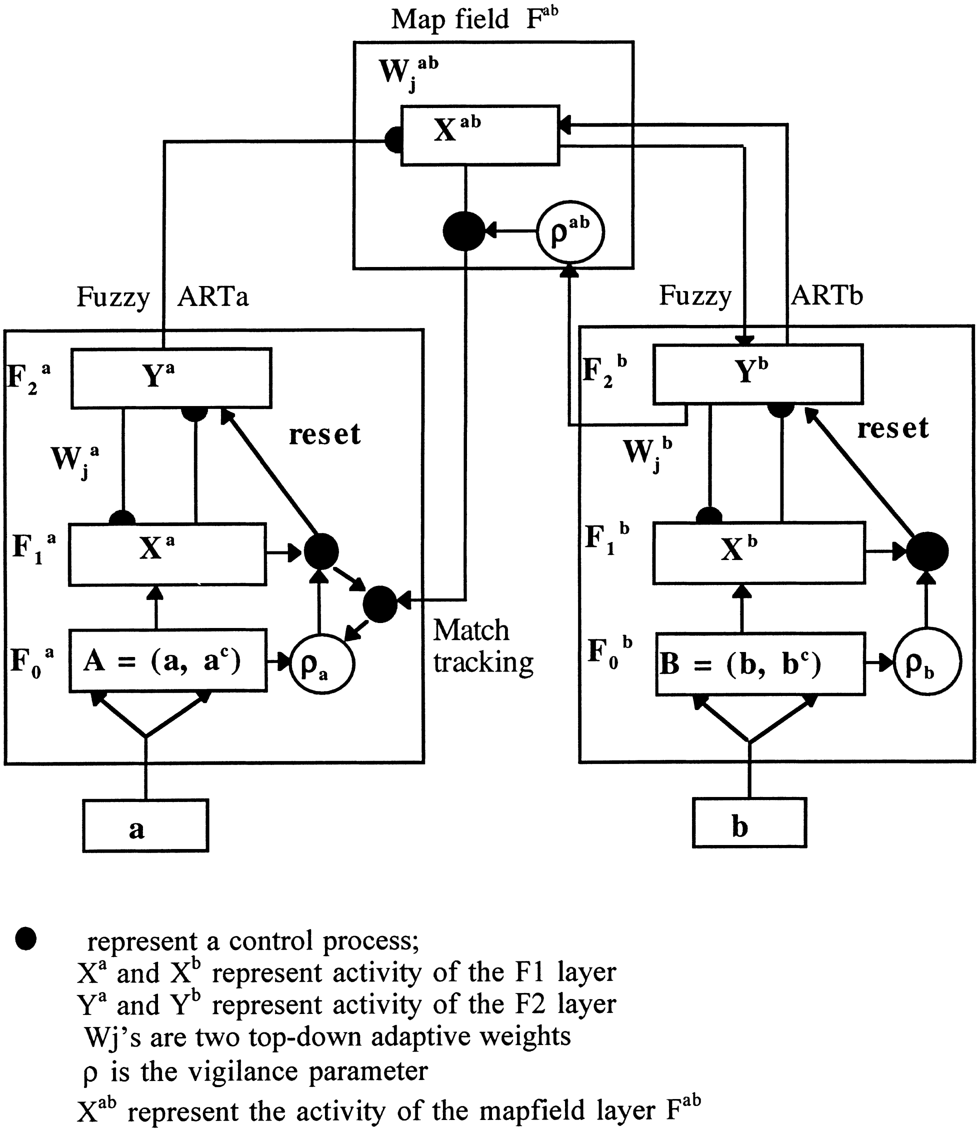

Map field F^{ab}
W_j^{ab}
X^{ab}
ρ^{ab}
Fuzzy ARTa
$F_2{}^a$
Y^a
$W_j{}^a$
reset
$F_1{}^a$
X^a
$F_0{}^a$
$A = (a, a^c)$
ρ_a
Match tracking
a
Fuzzy ARTb
$F_2{}^b$
Y^b
$W_j{}^b$
reset
$F_1{}^b$
X^b
$F_0{}^b$
$B = (b, b^c)$
ρ_b
b
● represent a control process;
X^a and X^b represent activity of the F1 layer
Y^a and Y^b represent activity of the F2 layer
Wj's are two top-down adaptive weights
ρ is the vigilance parameter
X^{ab} represent the activity of the mapfield layer F^{ab}

Fig. 9. Architecture of fuzzy ARTMAP neural network.

$$\frac{|I \wedge W_j|}{|I|} \geqslant \rho$$

resulting in node J being committed, else reset occurs at the selected output node and is inhibited for further competition for input I. This process is repeated until the vigilance criteria equation is satisfied. If the output node satisfies the vigilance criteria then learning takes place using [2]

$$W_j^{\text{new}} = (I \wedge W_j^{\text{old}}).$$

In order to evaluate and compare the performance of BPN and fuzzy ARTMAP networks, the same number of training sets and testing sets were applied to both. First a total of 30 sets of data are randomly selected, 15 good pellets and 15 pellets with various defects (5 banded, 4 end defect, and 6 chipped pellets) for training. Similarly, 20 good sets and 20 bad sets are randomly selected (5 banded, 4 end defect, and 11 chipped pellets) for testing. It may seem not necessary to have more teasing data than training data, however, the same 40 test data are needed again when increasing the training data from 30 to 120 in the second trial. Two output nodes are set for the backpropagation and the fuzzy

[2] Presentation of more detail algorithm of ART family of neural netwoeks is quite lengthy and is certainly beyond the scope of this paper. The interested reader should refer to Refs. [18,19] for further information.

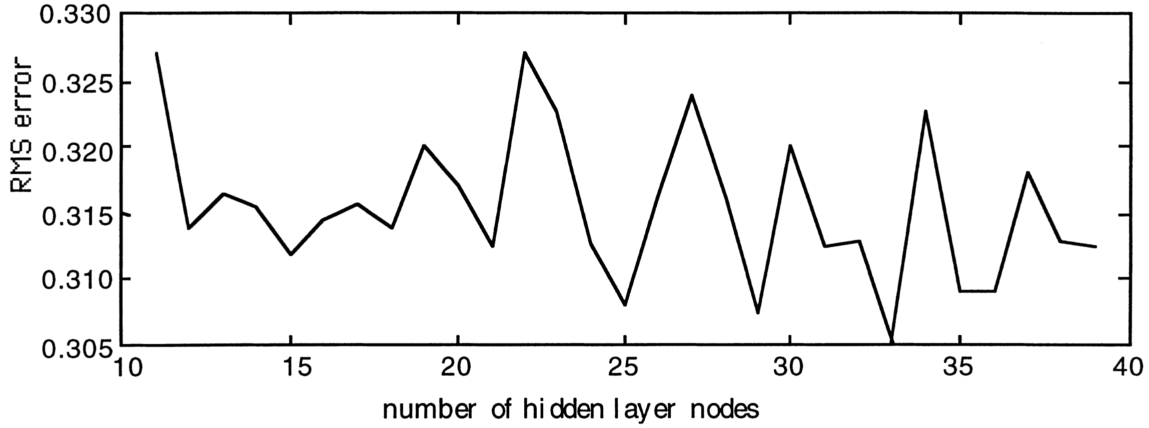


Fig. 10. Output RMS error versus number of node in hidden layer.

ARTMAP networks. Although, the BPN network could work just as well with one output node in this application, the two output node was preferred in order to create an output display that is easy to read.

A test is performed to see if a BPN would successfully identify bad pellets. To get a good BPN structure for this special purpose, different architectures of backpropagation with one hidden layer and the same input and output nodes are examined. The number of nodes in the hidden layer ranged from 15 to 40. Training proceeded for 5000 epochs, at which the output RMS error stabilized at approximately 0.31. Best performance was achieved when using 33 nodes in the hidden layer (i.e. minimum RMS error), as shown in Fig. 10. For each pellet image the 39 features described in Table 1 are extracted and normalized (i.e. a value between 0 and 1) and then used as input to the neural network. The normalization process consisted of two steps. The first step was to calculate the value of the 39 features for a solid white pellet (i.e. intensity of 255 for all pixels) as well as for a solid black pellet (i.e. intensity of 0 for all pixels). The second step was to use the maximum value of each feature from either the solid white or solid black pellet as the normalization devisor for the corresponding feature for each pellet examined in this application. The final network architecture consisted of 39 nodes for inputs layer (using the 39 normalized feature), 33 for hidden layer and 2 nodes for the output layer. The results of Pass/Fail classification of BPN are given in Table 2. In the first trial using 15 training samples, the BPN performs well with a 100% success rate at identifying 20 good pellets, a 65% success rate for bad pellets, and an overall success rate of 82.5%.

The Pass/Fail classification of the fuzzy ARTMAP result is also given in Table 2. The vigilance factor for Fuzzy ARTMAP is set to 0.85 based on the optimal classification performance in this study. Fuzzy ARTMAP performs with 85% correct rate at identifying 20 good pellets, 75% for bad pellets, and an overall success rate of 80%.

Table 2
Results of pass/fail classification

Pellet image type	Number of training sets	Number of testing sets	Correct classification
Backpropagation			
Good	15	20	20/20
Bad	15	20	13/20
Total	30	40	33/40 (82.5%)
Backpropagation			
Good	60	20	20/20
Bad	60	20	16/20
Total	120	40	36/40 (90%)
Fuzzy ARTMAP			
Good	15	20	17/20
Bad	15	20	15/20
Total	30	40	32/40 (80%)

To check the effect of the number of training data on the classification outcome, the training sets are increased to 120 (60 for good and 60 for bad pellets (22 banded, 12 end defect, and 26 chipped pellets)) for both backpropagation and fuzzy ARTMAP. Testing data are still the same original 40 sets. The overall performance of backpropagation is improved from 82.5% to 90% correct rate as shown in Table 2. Increasing the training sets does not improve the performance of fuzzy ARTMAP. It must be mentioned that although the number of training sets is increased, there are not enough sample pellets for certain types of defects (i.e. cracks) for training and testing by BPN or fuzzy ARTMAP. More training sets of all types of defects are necessary to obtain the optimum performance of the supervised neural networks. In addition, in this first examination of the neural networks performance for pellet inspection, the objective was to recognize good versus bad (i.e. defective) pellets apart. Hence, it was not necessary to identify the type of defect. Therefore, defective pellets were selected randomly and no attempt was made to make an orderly training set with equal sample of each type of defect. In fact, the abundant number of chipped pellet resulted in utilization of more of them both in training and testing.

5. Application of unsupervised ART2-A neural network

ART2-A ('algorithmic' ART), an unsupervised network, is a special case of ART2 paradigm which emphasizes the intermediate and fast learning rates, hence accelerating the learning process by 3 orders of magnitude. Fig. 11 shows the architecture and the learning mechanism of ART2-A network. ART2-A has three fields: F_0, F_1, and F_2 (see Fig. 11). The output of the F_0 field is the vector I defined by

$$I = \mathrm{normal}(f(\mathrm{normal}(I_0))),$$

where I_0 is the input vector of dimensionality M, and normal is an operator defined by

$$\mathrm{normal}(x) = x/\|x\|$$

and $f(\ )$ is a piecewise linear function:

$$f(x) = \begin{cases} 0 & \text{if } 0 \leqslant x < \theta, \\ x & \text{if } x \geqslant \theta, \end{cases}$$

with $0 < \theta \leqslant (M)^{-1/2}$.

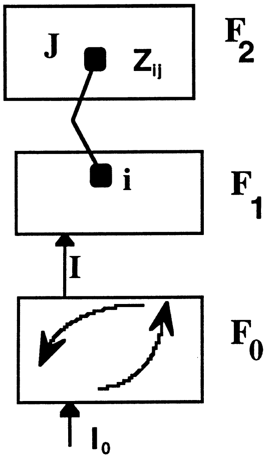
A test is performed to determine the feasibility of using ART2-A to carry out the pass/fail inspection. Forty good pellets and 40 bad pellets (7 banded, 7 end defect, and 26 chipped pellets) are randomly selected for ART2-A application. Each input file to the ART2-A network consists of four sets of data, two sets of good pellet and two sets of bad pellet data. Each set consists of the same normalized 39 features described in Table 1. ART2-A evaluates each file, and based on the vigilance input, assigns an integer to each image representing ART2-A network classification. A correct identification with the optimum vigilance is a report of '0011' – two zeros indicating a group containing two of a kind, followed by two ones indicating a group of two of another kind. Since we are testing good versus bad classification, and output of 0012 is also considered as a correct classification. This is because 0 represents a good pellet and anything other than 0 represents a defective pellet. When the two defective pellets do not have the same defects, then they are classified separately also, i.e., an output of '0012' instead of '0011'. It should be mentioned that two sets of each category (good and bad) must be used for proper evaluation of ART2-A network performance. This is because if using only one set, then with high enough vigilance factor one can always separate the two category apart, resulting in a meaningless test and 100% success rate.

Table 3 shows the testing results using ART2-A. As mentioned above, there are four sides (images), a, b, c and d, for each pellet. The last alpha letter in the pellet name refers to the pellet side, and the number in the file name refers to the pellets number, e.g., p2a refers to pellet number 2, side a. All applied data are randomly selected in this test. Results in Table 3 indicates a success rate of classification of 16/20 (80%) for the unsupervised network ART2-A.

To check the impact of the number of features on the unsupervised ART2-A network performance, a new test is performed using nine selected features corresponding to features number 3–11 in Table 1. These inputs are selected based on two criteria: (a) they must represent the most sensitive features according to the feature patterns in Figs. 5–7, and (b) select only one set of moment of inertia features to avoid possible noisy input data. The result is again an 80% success rate of correct classification as shown in Table 4. Reducing the number of features from 39 to 9 clearly does not improve the overall performance in separating good versus bad (defective) pellets. Although performance is not improved, the result indicates that an overall smaller vigilance ρ is required than when using all 39 features. This indicates that patterns are not too similar to require a fine category recognition (i.e., high vigilance ρ) as was the case for the first test with 39 feature input.

6. Conclusions

The results of network classification in this study show that a supervised network is required for proper classification of fuel pellets. Although the overall success rate is 90% for BPN network (see Table 2), the fact that



$$I = N(F(N(I_0)))$$

Where: **I** is the vector output of the field Fo
Io is the input vector of dimensionality M.
F is the piecewise linear function:

$$f(x_i) = \begin{cases} x_i & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

N is a 'normal' operator
$\mathbf{Z_{ij}}$ is LTM for input i to node j

Long term memory (LTM) is made of two components, the bottom-up adaptive coefficient Z_{ij} and the top-down adaptive coefficient Z_{ji}.

Learning:

$$Z_J^{(new)} = \begin{cases} I & \text{if J is uncommitted} \\ \text{normal}(\beta(\psi) + (1-\beta)Z_J^{(old)}) & \text{if J is committed} \end{cases}$$

where

$$\psi_i = \begin{cases} I_i & \text{if } Z_{JI}^{(old)} > \theta \\ 0 & \text{otherwise} \end{cases}$$

$0 \leq \beta \leq 1$ ($\beta = 1$ for fast learning),

$0 < \theta \leq (M)^{-1/2}$, M is the number of input data points, i.e. dimension of the input vector I_0.

Fig. 11. The architecture and neuro-dynamics of ART2-A network.

it is safer to miss a good pellet than to miss catching a bad pellet, an 80% success rate for bad pellet test is not exciting. On the other hand, missing too many good pellets is also not acceptable economically. Therefore, the high success rate (100%) for good pellet classification of BPN network is very exciting. The results obtained here are generally encouraging, considering this is a first attempt in testing the neural networks performance in pattern recognition of good versus defective pellet. In addition, due to lack of sufficient cracked pellet (only 3 cracked pellets in the 252 sample pellets), crack image was not included in tests presented in this paper. In view of the fact that some of the features described here are sensitive to crack defect, it is expected that training with more samples of all type of defects would improve the classification performance of supervised neural networks. Work is in progress (with support from the Department of Energy) using machine vision techniques and fuzzy logic for pellet inspection and comparison with ANN performance.

Table 3
Results of ART2-A network performance using 39 features

Number	Good pellets (file name)		Bad pellets (file name)		Vigilance parameter	ART2-A reports	Correct?
1	p2a	p4a	p4c	p14b	0.855	0012	Yes
2	p16a	p24a	p51b	p73b	0.77	0011	Yes
3	p88a	p140a	p75b	p88c	0.87	0010	No
4	p218a	p62a	p122b	p145b	0.87	0012	Yes
5	p72a	p82a	p146b	p200a	0.9	0011	Yes
6	p227a	p115a	p201a	p202a	0.1	0011	Yes
7	p126a	p133a	p208a	p214a	0.9	0012	Yes
8	p158a	p165a	p229a	p230a	0.96	0111	No
9	p171a	p178a	p231a	p232a	0.9	0001	No
10	p184a	p209a	p233a	p236a	0.99	0000	No
11	p145a	p174a	p129a	p130a	0.84	0011	Yes
12	p1a	p3a	p17a	p25a	0.8	0011	Yes
13	p26a	p91a	p28a	p90a	0.88	0011	Yes
14	p15a	p18a	p116a	p118a	0.8	0011	Yes
15	p79a	p218a	p66a	p67a	0.85	0011	Yes
16	p20a	p23a	p7a	p74a	0.8	0011	Yes
17	p36a	p42a	p47a	p49a	0.84	0111	No
18	p97a	p107a	p50a	p54a	0.805	0011	Yes
19	p56a	p61a	p57a	p69a	0.82	0011	Yes
20	p63a	p110a	p60a	p64a	0.89	0012	Yes

Table 4
Results of ART2-A network performance using 9 features

Number	Good pellet (file name)		Bad pellet (file name)		Vigilance parameter	ART2-A reports	Correct?
1	p2a	p4a	p4c	p14b	0.1	0011	Yes
2	p16a	p24a	p51b	p73b	0.5	0011	Yes
3	p88a	p140a	p75b	p88c	0.5	0011	Yes
4	p218a	p62a	p122b	p145b	0.65	0012	Yes
5	p72a	p82a	p146b	p200a	0.77	0011	Yes
6	p227a	p115a	p201a	p202a	0.1	0011	Yes
7	p126a	p133a	p208a	p214a	0.7	0012	Yes
8	p158a	p165a	p229a	p230a	0.96	0001	No
9	p171a	p178a	p231a	p232a	0.96	0001	No
10	p184a	p209a	p233a	p236a	0.96	0000	No
11	p145a	p174a	p129a	p130a	0.665	0012	Yes
12	p1a	p3a	p17a	p25a	0.67	0011	Yes
13	p26a	p91a	p28a	p90a	0.85	0011	Yes
14	p15a	p18a	p116a	p118a	0.8	0011	Yes
15	p79a	p218a	p66a	p67a	0.65	0011	Yes
16	p20a	p23a	p7a	p74a	0.7	0011	Yes
17	p36a	p42a	p47a	p49a	0.8	0101	No
18	p97a	p107a	p50a	p54a	0.7	0011	Yes
19	p56a	p61a	p57a	p69a	0.86	0011	Yes
20	p63a	p110a	p60a	p64a	0.69	0012	Yes

Acknowledgements

This research is sponsored by the University of Missouri Research Board.

References

[1] S. Glasstone, A. Sesonske, Nuclear Reactor Engineering, Chapman and Hall, London, 1994.

[2] D.R. Olander, Fundamental Aspects of Nuclear Reactor Fuel Elements, Technical Information Center Energy Research and Development Administration, Springfield, VA, 1985.

[3] R.G. Cochran, N. Tsoulfanidis, The Nuclear Fuel Cycle: Analysis and Management, American Nuclear Society, La Grange Park, IL, 1992.

[4] NUREG-0800, Standard review plan for the review of safety analysis reports for nuclear power plants, July 1981.

[5] M. Brwon, Personal interview on Union Electric Callaway's fuel, 6/21/95.

[6] Westinghouse World Wide Web Page, http://www.nist.-gov:8102/doc/Win/Westinghouse_Electric_Corporation.html, July 1995.

[7] R. Pautler, Personal interview on Westinghouse fuel inspection, 7/7/95.

[8] B. McLees, Personal interview on Siemens pellets inspection methods, 6/16/95.

[9] D. Landry, Personal interview on GE automation project, 1/5/95.

[10] J.A. Freenman et al., Neural Networks: Algorithm, Application and Programming Techniques, Addison–Wesley, Reading, MA, 1992.

[11] N. Tishby, E. Levin, S. Solla, Consistent inference of probabilities in layered networks: Prediction and generalization, in: Proceedings of International Joint Conference on Neural Network, IEEE TAB, June 1989, pp. II403–II409.

[12] M.L. Kelly, Nuclear fuel pellet inspection using machine vision and artificial neural networks, MS thesis, University of Missouri-Rolla, December 1995.

[13] M.B. Christopher, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[14] A.K. Jain, Fundamental of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[15] C.R. John, The Image Processing Handbook, CRC Press, Boca Raton, 1992.

[16] K. Mehrotra, C.K. Mohan, S. Ranka, Elements of Artificial Neural Networks, MIT, Cambridge, MA, 1997.

[17] C.T. Lin, C.S. George Lee, Neural Fuzzy Systems: A neuro-Fuzzy Synergism to Intelligent Systems, Prentice-Hall PTR, Upper Saddle River, NJ, 1996.

[18] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional MAPs, IEEE Transactions on Neural Networks 3 (5) (1992).

[19] G. Carpenter, S. Grossberg, D.B. Rosen, ART2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition, Neural Networks, 4, 1991.