Inhalt

9 Zusammenfassung

Trotz ihrer theoretischen Leistungsfähigkeit haben neuronale Netze bisher die in sie gesetzten Erwartungen bezüglich der Klassifikation meßtechnisch erfaßter dreidimensionaler Datensätze in der Praxis nicht erfüllt. Das Ziel dieser Arbeit war deshalb, zu untersuchen, wo die Schwachstellen der bisherigen Ansätze lagen und Verbesserungsmöglichkeiten aufzuzeigen.

Dazu wurden zunächst verschiedene profilgebende Meßverfahren untersucht und anhand zweier konkreter Anwendungsbeispiele – Patronenhülsen und Honstrukturen – miteinander verglichen. Es hat sich gezeigt, daß im Prinzip die geeigneten Meßmethoden zur Verfügung stehen, wenn auch für Anwendungen in der Produktion aus Zeitgründen vom Ideal eines kompletten Flächenscans abgewichen werden muß und statt dessen eine Kombination aus Profilschnitt und Grauwertbild eingesetzt wird.

Nach dem derzeitigen Stand der Technik ist es in der Praxis keine Problem, zu einer gegebenen Trainingsstichprobe einen passenden Klassifikator zu finden. Ein gutes Abstraktionsvermögen kann dagegen mit den praxisüblichen Algorithmen i. allg. nicht garantiert werden. Die zentrale Aussage des Abschnitts 4 dieser Arbeit ist die, daß das daran liegt, daß die Komplexität des Klassifikators nicht an das Problem angepaßt wird. Üblicherweise werden Klassifikatoren mit zu vielen Freiheitsgraden eingesetzt.

Im Rahmen der Arbeit wurden daher neue Konzepte zu Konstruktion von Klassifikatoren entwickelt. Insbesondere

- eine quantisierte Form des Backpropagation-Algorithmus als ein Ansatz für einen Lernalgorithmus, der zuerst Klassifikatoren mit der geringst möglichen Komplexität generiert,
- symmetrische neuronale Netze, die ein Problem der Klassifikation in *n* Klassen auf eine Klassifikation in 2 Klassen reduzieren, sowie
- eine Verallgemeinerung der Hopfield-Netze mit mehr als zwei diskreten Zuständen je Neuron, bei der die Netzgewichte nicht explizit gespeichert, sondern in funktionaler Form dargestellt sind.

Diese Ansätze wurden in einem auf Neuro-Fuzzy basierenden Expertensystem zusammengefaßt, das die Zuordnung von Patronenhülsen ermöglicht. In den Voruntersuchungen zu dieser Arbeit hat sich gezeigt, daß neuronale Netze alleine dazu nicht in der Lage sind. Weiterhin wurde die praktische Anwendbarkeit der vorgeschlagenen Algorithmen zur Beurteilung von Honstrukturen demonstriert.

Wichtig für das Gelingen der Klassifikation war die Verwendung echter Profilinformation. So konnte etwa bei der Zuordnung der Patronenhülsen der Öffnungswinkel des Einschlagkegels als Merkmal herangezogen werden, was mit einfachen Grauwertbildern nicht möglich ist. Bei der Beurteilung von Honstrukturen sind die nach DIN 4776 aus dem Höhenprofil berechneten Rauheitsparameter wichtige (und weithin anerkannte) Merkmale und auch bei der Detektion von Blechmantel spielt die Kombination aus Reflexions- und Profilinformation eine wichtige Rolle. Die subjektive Komponente, die bei der Erstellung von Grauwertbildern aus Faxfilmabdrücken durch die manuelle Wahl der Beleuchtung unvermeidlich ist, läßt sich durch die Verwendung der Höheninformation vollständig eliminieren. Weiterhin vereinfacht sich die Anwendung von Bildverarbeitungsalgorithmen, da keine Probleme mit Kontrast- und Beleuchtungsinhomogenitäten auftreten. Da eine normgerechte Rauheitsmessung Voraussetzung für eine breite Akzeptanz ist, gibt es kurzfristig keine Alternative zum mechanischen Tastschnittverfahren. Abbildung 54 auf Seite 86 zeigt, daß sich aus der Höheninformation durch geeignete Rechenverfahren eine Darstellung erzeugen läßt, die mit den Faxfilmaufnahmen vergleichbar ist. Umgekehrt ist das nicht möglich[11].

Die für die Praxistauglichkeit eines Klassifikators entscheidende Eigenschaft ist das Abstraktionsvermögen, also das Verhalten auf nicht gelernten Merkmalsvektoren. Die Abweichung des zu erwartenden Fehlers auf nicht gelernten Daten vom Klassifikationsfehler auf der Stichprobe ist um so kleiner, je kleiner die Komplexität des Klassifikators ist. Dies ist die zentrale Aussage des MDL-Prinzips. Durch die Einführung der symmetrischen neuronalen Netze konnte die Anzahl der Freiheitsgrade wesentlich reduziert und damit das Abstraktionsvermögen verbessert werden. Um ein Maß für die Komplexität eines trainierten (symmetrischen) neuronalen Netzes zu erhalten, wurde eine quantisierte Version des Backpropagation-Algorithmus vorgeschlagen. Die hier beschriebene Modifikation des Hopfield-Netzwerks erlaubt es, aus dem von einem symmetrischen neuronalen Netz berechneten Ähnlichkeitsmaß eine scharfe Äquivalenzrelation zu gewinnen.

Die Leistungsfähigkeit der genannten Algorithmen wurde sowohl an simulierten Testdatensätzen als auch an Messungen von Patronenhülsen und Hontexturen erprobt. Mit den Simulationen konnte die Überlegenheit der symmetrischen neuronalen Netze nachgewiesen werden. Das so berechnete Ähnlichkeitsmaß eignet sich auch für die Vorselektion bei der Klassifikation von Patronenhülsen. Für die endgültige Auswahl wird ein Neuro-Fuzzy-Expertensystem eingesetzt, das auch den Korrelationskoeffizienten als Kriterium nutzt. Bei der Beurteilung von Honstrukturen wurde zunächst die Bestimmung des Honwinkels automatisiert. Um eine gute Akzeptanz zu erreichen, wurde die Berechnung durch eine geeignete Visualisierung (Abbildung 64, Seite 91) überprüfbar gemacht. Voraussetzung für die automatische Beurteilung der Gleichmäßigkeit der Riefenscharen ist die Detektion einzelner Honriefen. Dies wurde durch Anwendung des MDL-Prinzips erreicht. Ebenfalls mittels MDL wurde eine Segmentierung der einzelnen Honriefen erreicht.

Es ist zu erwarten, daß durch konsequente Anwendung der beschriebenen Methoden weitere Verbesserungen bei der automatischen Klassifikation erreichbar sind. Bei der Archivierung der Patronendaten kann MDL dabei helfen, nur relevante Merkmale zu speichern und dadurch gleichzeitig eine Datenreduktion zu erzielen. Bei den Honstrukturen können nach einer geeigneten Merkmalsextraktion symmetrische neuronale Netze die nach 8.8.1 detektierten Defekte klassifizieren.

Weitere Verbesserungen sind von den optischen Meßverfahren zu erwarten, die durch Bereitstellung zusätzlicher Reflexionsdaten die automatische Detektion von Blechmantelbildung erleichtern würden.

[11] Ist das Streuverhalten der Oberfläche exakt bekannt, kann das Höhenprofil theoretisch aus dem Grauwertbild rekonstruiert werden [41]. Praktische Bedeutung hat dies allerdings (noch) nicht.

10 Summary

10.1 Introduction

Technical surfaces are normally classified by comparing certain roughness values to specified tolerance limits. The surface is classified ‚OK‘ if the value lies within the given interval and ‚NOT OK‘ otherwise. It is thus assumed that there is a known correspondence between the roughness values and the functional behavior of the surface. Since this is not normally the case, statistical methods have to be applied. Conventional statistical methods can only be applied when the quality of the surface is determined by a few simple parameters. These are severe limitations:

- The surface must be characterized by only a few parameters, as the required sample size grows exponentially with respect to the number of parameters.
- The parameters have to be simple (e.g. expressed by an easily computable real number). Some texture parameters (e.g. those used in the Hon-Atlas), pose an classification task in itself.

In the forensic sciences one has to deal with large numbers of classes. The task here is to match an object against a large number of objects stored in an archive.

The Perceptron, which was developed in 1958 by F. Rosenblatt, was believed to be a solution to such problems. Those expectations could not be met and a paper of Minsky and Papert, which showed the limitations of the Perceptron, ended this period of neurocomputing. In the 1980s multilayer perceptrons, which do not suffer from the limitations of the original Perceptron and the backpropagation algorithm gave a new momentum to the research.

The commonly available development tools make it easy to train neural nets so that they perform well on the given training sample. But up to now there are only few successful applications of neural nets to real world problems. This is because multilayer perceptrons are universal classifiers which can easily be trained to correctly classify the training sample and - after some iterations - the test sample, but still perform poorly on unlearned samples.

This is not an inherent limitation of the neural approach. There are results from statistical decision theory which allow the estimation of the error rate on unlearned samples based on training sample size and net complexity. This is similar to MDL (Minimum Description Length) method from image analysis. The MDL approach is to select that model that would give the shortest codelength to encode both the model and the data. The main goal of this thesis is to make those results available to the practitioner.

10.2 Classification: State of the Art and Possibilities for Improvements

Figure 1 shows a typical setting for automatic classification using neural nets. A representative sample is manually classified. The objects are measured and characteristic features are extracted from the measured data. To manually classify the examples additional tests which may be time consuming or destructive may be necessary. It is thus not guaranteed, that the extracted feature vector contains the full information about the correct class. To test the classification accuracy of the trained network, an additional set of randomly selected and manually classified examples - the so called test set - is required.

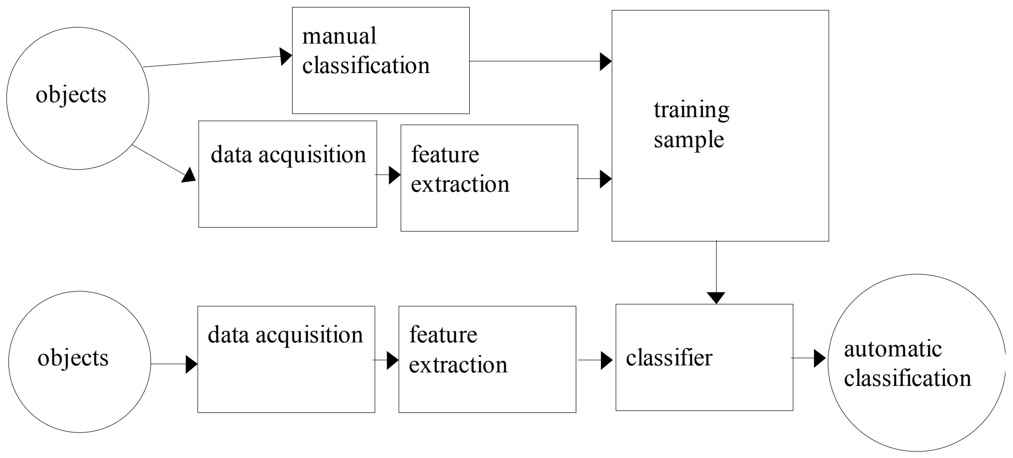

Figure 1: Automatic Classification using Neural Nets

Feature extraction is necessary to reduce the dimension of the input vector which determines the size of the first layer of the neural net and to incorporate *a priori* knowledge about the specific problem. This feature vector is presented to a multilayer perceptron as shown in figure 2. Here the neuron Y_1 in the so called hidden layer calculates the scalar product of the augmented input vector $x = (1, x_1, \ldots x_n)$ and its weight vector $w = (w_0, w_1, \ldots w_n)$. w_0 is called the bias. The output value is then calculated according to $z = \sigma(wx)$, where the output function σ is usually taken as $\sigma(x) = \frac{1}{1+e^{-x}}$.

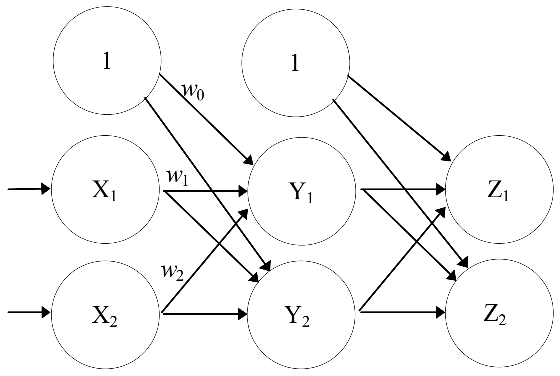

Figure 2: Multilayer Perceptron

A multilayer Perceptron can learn by adapting its weights. A widespread method for adapting the weights is the well known backpropagation algorithm, which can be shown to be equivalent to gradient descent. To apply a neural net to a classification task, each class is associated with a unit vector $e_i = (0, \ldots, 1, 0, \ldots 0)^T$. A new feature vector is assigned to class i if the output of neuron Z_i exceeds a given threshold. Classification is thus reduced to function approximation.

Another type of neural net is the Hopfield network. It is a single layer recurrent neural network with a symmetric weight matrix where each neuron can take on the two states +1 and -1. The neurons are updated sequentially and the new state is determined by comparing $\sum_j w_{ij} z_j$

to a threshold θ_i. The new state is +1 if the sum is greater than the threshold, else the new state is -1. It can be shown that the Hopfield network converges with probability 1 to a stable state. The Hopfield network has no associated learning law. The weight matrix is calculated in advance and it will be shown how a modification of the Hopfield net can be used for classification.

A different approach to classification uses fuzzy set theory, which was developed in 1965 by Lotfi A. Zadeh. It can be seen as a generalization of classical logic and can deal with imprecise and uncertain information. It is suited to model the knowledge of human experts as in the following example.

Example An expert classifies according to the rule "if the roughness value is high or the trough number is low, then part is to be rejected".

Fuzzification The membership value of the measured roughness value *x* and the trough number *y* of the set *A* of high roughness values and the set *B* of low trough numbers is computed:

$$\mu_A(x) = \begin{cases} 0, & x \le 1 \\ x-1, & 1 < x < 2, \\ 1, & x \ge 2 \end{cases} \mu_B(y) = \begin{cases} 1, & y \le 10 \\ 2 - \tfrac{y}{10} & 10 < y < 20. \\ 0, & y \ge 20 \end{cases}$$

With this definition of the membership function μ_A, roughness values above 2 μm are known to be too high, values below 1 μm can be accepted. In between the membership function is monotonically increasing and continuous in *x*.

Rule Evaluation The truth value *z* of the statement "the part is to be rejected" is calculated using the fuzzy OR operator *s*: $z = s(\mu_A, \mu_B)$, where *s* is defined as $s(a,b) = \dfrac{a+b}{1+ab}$.

De-fuzzification The part is rejected if $z > \dfrac{1}{2}$.

Automatic classification using fuzzy logic can be implemented with neural nets by choosing appropriate weights and output functions. By carefully selecting the rules, good performance can be achieved. But selecting the weight manually is very time consuming. Only a "fine-tuning" of the weights can be done automatically using learning algorithms such as backpropagation.

A Concept for Classification in the Case of Large Class Numbers

To construct classifiers with good generalization properties even in the case of large class numbers and / or only few available training samples per class, the following concepts are investigated:

- The MDL principle is used to avoid overfitting.
- Neural nets are trained to classify pairs of features vectors into the two classes similar / dissimilar.
- Where appropriate, crosscorrelation of surface data is used.
- A modified Hopfield network is used to convert a fuzzy similarity measure into a proper classification.
- Fuzzy logic is used to incorporate expert knowledge.

10.3 Data Acquisition

In order to classify technical surfaces, the surface topography has to be measured.

The most widely used measurement principle is the contacting stylus. Standardized roughness values are based on this method and it is robust enough to be used in a production environment. Due to the limited measurement speed, it is normally only used for line scans. When additional texture information is needed, a grayscale image is acquired using a CCD camera.

The auto focus sensor is an optical point sensor. Laser light is focussed on the surface and the reflected light is detected on a quadrant photo diode. A control loop keeps the objective lens at a fixed distance from the surface. The position of this lens is then measured. To measure roughness values, a special filtering of the data is necessary. Changes in the intensity of the reflected light is used to mask off spikes which occur at profile slopes above ca. 7°.

A white light interferometer is a fast 3D sensor. A reference plane scans in z direction at about 3 µm / s. A CCD camera is used to detect interference when reference path and object path match. The *z* position of this match is stored for each pixel.

Triangulation methods include fringe projection, where parallel fringes are projected onto the surface. A CCD camera detects the distortion of the fringes and the profile is calculated in real time. Microscopic fringe projection has been successfully used to measure honed surfaces.

In scanning confocal microscopes, a rotating Nipkow disk performs the x/y-scan on the surface. The intensity $V(z)$ of the reflected light attains a maximum at a distance z_0 from objective lens to surface. A scan in *z* direction is performed and the *z* position of the intensity maximum recorded for each profile point. Scanning confocal microscopy was used to measure the firing pin indentations in gun shells.

10.4 Construction of a Classifier using the MDL Principle

The problem of determining the optimum topology for a classification task is similar to selecting a suitable polynomial degree for curve fitting. Figure 3 shows an example of a set of data points together with a fitted polynomial of degree 3. The data has been generated by sampling a sine function with added noise. As can be seen in Figure 4, a polynomial of degree 9 fits the data better but it gives a poor representation of the underlying sine function. The same happens, when a neural network is overtrained. It classifies the given training data correctly but performs poorly on data that is not part of the training set.

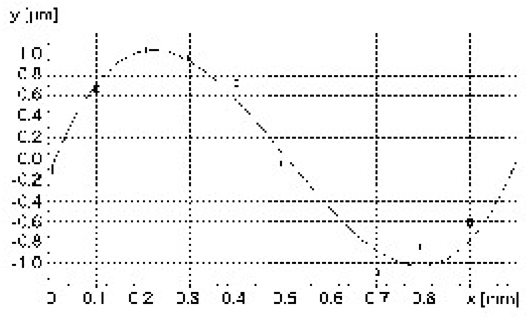

Figure 3: Fitted Polynomial of Degree 3

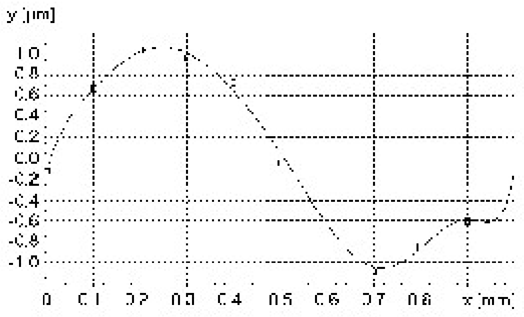

Figure 4: Fitted Polynomial of Degree 9

In the following it will be shown, how to find a good compromise between fitting the sampled data and achieving a good generalization. To do this formally, a few definitions are in order.

Definition 5 A *decision function* is a mapping $\phi\colon \mathbf{R}^d \to \{0,1\}$, where $\mathbf{R}^d$ is the *d*-dimensional feature space. In the context of neural nets, ϕ is a trained net. A feature vector $x \in \mathbf{R}^d$ is classified as a member of a given class, if and only if $\phi(x) = 1$.

A *classifier* is a function $\phi_n\colon \mathbf{R}^d \times \left(\mathbf{R}^d \times \{0,1\}\right)^n \to \{0,1\}$. Here $\left(\mathbf{R}^d \times \{0,1\}\right)^n$ is the space of all training sets of cardinality *n*. The input consists of *n* pre-classified feature vectors and one feature vector which is to be classified. The output is the prediction of the class of the unclassified vector. Equivalently, a classifier can be seen as a function $\phi_n\colon \left(\mathbf{R}^d \times \{0,1\}\right)^n \to C\colon \left((X_1, Y_1), \ldots, (X_n, Y_n)\right) \mapsto \hat{\phi}_n$, which maps a training set to a decision function. C is the set of all decision function which can be implemented by the selected network topology.

The *error probability* of a decision function ϕ is $L(\phi) := P(\phi(X) \neq Y)$. This is simply the probability that a vector which is drawn at random according to the underlying probability distribution is incorrectly classified.

The *empirical error probability* is defined by $\hat{L}_n(\phi) := \frac{1}{n}\sum_{i=1}^{n} I_{\{\phi(X_i) \neq Y_i\}}$, where *I* is the indicator function. Only this error can be measured during training.

The *approximation error* of a class *C* of decision functions is defined by $\inf_{\phi \in C} L(\phi) - L^*$. Here L^* is the Bayes risk (i. e. error probability of the optimum Bayes decision function). The approximation error measures how well the optimum decision function can be approximated by members of the class *C*. It is a falling function of *C*: $C \subset C' \Rightarrow L(C) \geq L(C')$.

The *estimation error* of a classifier is $\mathrm{E}\{L(\hat{\phi}_n)\} - \inf_{\phi \in C} L(\phi)$, where the expectation E is taken over all training sets of size *n*.

Obviously: $\mathrm{E}\left\{L(\hat{\phi}_n)\right\} = \left(\mathrm{E}\left\{L(\hat{\phi}_n)\right\} - \inf_{\phi \in C} L(\phi)\right) + \left(\inf_{\phi \in C} L(\phi) - L^*\right) + L^*$, i.e. the expected classification error is the sum of estimation error, approximation error and Bayes risk. The Bayes risk is determined by the joint distribution of features and classes. In the following, $L^* = 0$ will be assumed.

With neural nets it is easy to reduce empirical and approximation error by implementing large classes *C*, but this results in a greater estimation error. Ideally, approximation error and estimation error should be of the same order of magnitude. To achieve this, the sum $\hat{L} + r$ of empirical error and a complexity penalty is minimized, rather than minimizing only the empirical error as in the backpropagation algorithm. In the context of MDL (Minimum Description Length), *r* is the code length necessary to describe the members of *C* and $\hat{L}$ is the code length necessary to reconstruct the correct classes of the training vectors, given the prediction of the selected decision function. Since exactly minimizing $\hat{L} + r$ requires run time exponential in code length, a more practical solution shall be proposed. The *Quantum Backpropagation Algorithm* is a variation of the backpropagation algorithm where the length of the code representing the resulting decision function is proportional to the run time:

Algorithm QBP 1

Let $q_n; n \in \mathbf{N}$ be a falling sequence with $\lim_n q_n = 0$ and $\sum_n q_n = \infty$, e.g. $q_n = \frac{1}{n}$.

1. **Initialization:** set all weights to zero.
2. **Backpropagation:** for every weight w_i calculate the change $\Delta w_i := \operatorname{sign}(\Delta w_i' + \varepsilon_{ni}) \cdot q_n$, where n is the number of the current iteration, $\Delta w_i'$ is the weight change as calculated by the standard backpropagation algorithm and ε_{ni} is a small random number to assure that the term in brackets is not zero after initialization.
3. **Update:** set the new weights to $w_i + \Delta w_i$.
4. **Termination:** if $n \cdot n_w > c_{\min}$ then stop, else continue with step 2. n_w is the number of weights, $c_{\min}$ is the minimum sum of empirical error and complexity term $n \cdot n_w$ encountered so far.

The output is the net which achieved the minimum $c_{\min}$. It is coded by the $n \cdot n_w$ sign bits.

Algorithm QBP 2

Set $q_1 := 1$ and $q_{n+1} := \begin{cases} \alpha q_n, & \operatorname{sign}(w_{n+1}) = \operatorname{sign}(w_n) \\ \beta q_n, & \operatorname{sign}(w_{n+1}) \neq \operatorname{sign}(w_n) \end{cases}$, where $\alpha > 1$ and $0 < \beta < 1$, e.g. $\alpha = \sqrt{5} - 1$ and $\beta = \frac{1}{\sqrt{5}}$. Here the modulus of the weight change is adapted for each weight individually. This results in a faster convergence. Again, the q_n and thus the resulting net can be reconstructed from the sign bits, and $n \cdot n_w$ is a proper complexity term.

10.5 Calculation of Similarity Measures using Neural Nets

With large class numbers, the approach, where each class is assigned to a neuron in the output layer is not practical, as this would result in neural nets with very high complexity. Instead, the simpler task of calculating a measure of similarity is considered first. Given two feature vectors x and x', a measure of similarity $\mu(x, x') \in [0,1]$ is to be calculated, which satisfies $\mu(x, x) = 1$ (reflectivity) and $\mu(x, x') = \mu(x', x)$ (symmetry).

To do this, a symmetric neural net is constructed (Figure 5). The feature vectors are input to two identical nets. From the Euclidean distance $d = \sum (z_i - z_i')^2$ of the respective output vectors the similarity measure is calculated as $\mu = f(d, a)$, where $f(d, a) = \frac{1}{1 + d^a}$, $a > 0$.

The training set consists of pairs of feature vectors. Each pair is associated a desired output value $s = 1$ or $s = 0$, depending on whether or not both vectors are members of the same class. The error $\mu - s$ is propagated back through the net and the weight changes $\Delta w, \Delta w'$ in the upper and lower half calculated as in the standard backpropagation algorithm. The weights are updated according to $w_{\text{new}} := w_{\text{old}} + \Delta w + \Delta w'$, which leaves the symmetry intact.

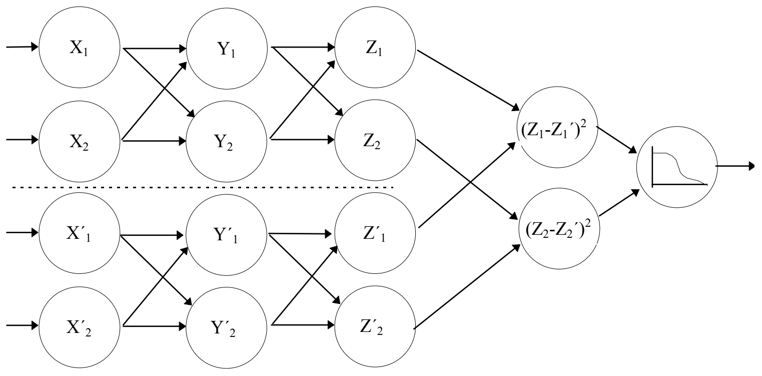

Figure 5: Symmetric Neural Net

If the feature vectors are transformed using the resulting neural net, a nearest neighbor classifier can be used to carry out the classification. Another approach is to use a modification of the Hopfield net. In this modified Hopfield network, each neuron can take on one of *n* different states, where *n* is the number of classes. Each vector in the training set and each unclassified vector is associated with a neuron in the Hopfield net. The states of the neurons associated with vectors from the training set are fixed to their known classes. Of the neurons associated with unclassified vectors, one neuron at a time is selected at random and its state z_i updated, so that the energy function $E_i(z) := -\frac{1}{2}\sum_{\substack{j \\ z_i = z_j}} w_{ij} + \frac{1}{2}\sum_{\substack{j \\ z_i \neq z_j}} w_{ij} + \theta_{iz_i}$ is minimized. The weights are defined by a similarity measure: $w_{ij} = \mu(x_i, x_j) - \frac{1}{2}$. The iteration will converge to a stable state, which yields a classification for the feature vectors.

10.6 Implementation of a Neuro-Fuzzy Expert System

The knowledge base of a fuzzy expert system consists –as does a conventional expert system– of rules of the form $P_1 \wedge P_2 \wedge \ldots \wedge P_n \to C$, where P_i are the premisses. If all the premisses are true, then the rule states that the conclusion *C* is true. Each rule is associated with a degree of validity $v \in [0,1]$. Also each statement *A* is associated with a truth value in [0, 1]. Negation and operators such as AND and OR are differentiable functions:

NOT: $n(x) = 1 - x$

AND: $t(x, y) = \dfrac{xy}{xy - x - y + 2}$

OR: $s(x, y) = \dfrac{x + y}{1 + xy}$

The knowledge base of a fuzzy expert system is constructed by a human expert. In a neuro-fuzzy system it can be modified using learning algorithms such as backpropagation as follows. Set $f(x,a) := 2\left(a - \frac{1}{2}\right)x + \min\{1, 2(1-a)\}$. Then $f(x,0) = 1 - x$, $f\left(x, \frac{1}{2}\right) = 1$ and $f(x,1) = x$. With appropriate selection of $a_i \in \{0, \frac{1}{2}, 1\}$, $t(f(x_1, a_1), \ldots f(x_n, a_n))$ can represent any conjunction of statements and their negations. The initial selection of a_i can be adapted via backpropagation.

10.7 Classification of Gun Shells

Matching of firing pin indentations in gun shells in forensic laboratories is a typical task of classification into a large number of classes. This task is currently carried out manually by forensic experts, which can be very time consuming. A method for automatic pre-selection of best matching shells is proposed here.

A part holder and motorized x/y-stages allow for automated measurements of up to 25 shells. The measurement time for an area of 2×2 mm^2 on the primer of a shell varies between 4 and 6 minutes, depending on the measuring method. Standard methods are used for leveling and alignment of the measured data. Special software routines are used to reproduce the SYBE features currently used by the German BKA. This is necessary to use the method to match measured shells against the ca. 5000 shells archived at the BKA. Furthermore cross-correlation of measurement data is used as a similarity measure and the methods from sections 5 and 6 are applied.

10.8 Classification of Honed Surfaces

The honed surfaces of cylinder bores are currently classified by DIN/ISO roughness values and by comparing grayscale images of the texture to pre-classified images in the so called Hon-Atlas, which is mainly a collection of samples.

To automate this, a cylinder test device has been constructed. It uses a contacting stylus to measure roughness values according to the standards. A CCD camera is used to acquire a grayscale image and the texture is analyzed. The honing angle is calculated by integrating the power spectrum along radii and determining the maximum. The honing angle is then used to do an affine transformation such that the troughs become parallel to the co-ordinate axes. This allows for a quick visual verification.

Special algorithms based on MDL have been developed to detect troughs and trough-defects. This is done by calculating a Hough transform of the binarized grayscale image (or a Radon transform of the actual profile data) and detecting local maxima. Each local maximum in the transformed data corresponds to a candidate trough. MDL is used as a significance measure. To do this a two stage coding of the data is considered. First the angle, position and width of the troughs are stored. Then the data inside and outside the troughs is coded separately. This reduces the code length since the entropy in these areas is smaller than the overall entropy. A detected maximum is considered significant, if coding the corresponding trough separately reduces the total code length.

To detect trough defects, this principle is taken one step further. A segment of a trough is considered a defect, if coding its length and position within the trough and excluding it from the trough area further reduces the code length. This results in a very robust algorithm without the need for manually setting any acceptance thresholds.