Bayesian Selection of Important Features for Feedforward Neural Networks

Kevin L. Priddy Steven K. Rogers Dennis W. Ruck Gregory L. Tarr
Matthew Kabrisky

Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB OH 45433

March 2, 1992

Accepted For Publication in Neurocomputing: Paper Number: 92-92.

Abstract

This paper presents a probability of error based method of determining the saliency (usefulness) of input features and hidden nodes. We show that the partial derivative of the output nodes with respect to a given input feature yields a sensitivity measure for the probability of error. This partial derivative provides a saliency metric for determining the sensitivity of the feedforward network trained with a mean squared error learning procedure to a given input feature.

1 Introduction

The multilayer perceptron has received a great deal of attention for implementation in pattern recognition tasks [5] [9] [15] [16] [17]. The results recently presented by Ruck [19] and Gish [4] have demystified the multilayer perceptron, showing that it approximates a Bayes optimal discriminant function [21][6]. This result allows us to find an understandable relationship between the output of the neural network and an input feature.

The next section will illustrate the relationship between the probability of error and an input feature. The third section will compare the result obtained in Section II with a previously used saliency metric. Finally, the last section will discuss the results obtained.

2 Computing the Saliency

The saliency (relevance) of a given input feature in determining classification is of great interest in pattern recognition because of the problems encountered by large numbers of measurements (features) [1] [2] [3] for classification. Often the object to be classified is measured using various criteria (features) with the resultant description recorded as a vector. What is often assumed during the classification process is that each feature is equally relevant. In actuality, this is almost never the case because some features can be

correlated, some scaled duplicates, or others totally irrelevant for classification. We will now present an approach to measure the usefulness of a given feature in the classification process.

The output of the feedforward network in the limit can be represented by [19]:

$$\mathrm{z}_j = \mathrm{P}(\mathrm{C}_j|\vec{x}) \tag{1}$$

where

$$\begin{aligned} \mathrm{z}_j &\equiv \text{output of node } j \text{ on output layer} \\ \mathrm{P}(\mathrm{C}_j|\vec{x}) &\equiv \text{the a posteriori probability of class } \mathrm{C}_j \text{ given input vector } \vec{x} \\ \vec{x} &\equiv \text{input feature vector} \end{aligned}$$

The result stated above is dependent on the outputs of the neural network being trained to be one and zero where one node will have a value of one and all others a value of zero for a given input class. In addition, the result requires that the neural network has converged to a final solution. We will now relate the a posteriori probability to an error criterion which will be used to measure the saliency of a given input feature.

From Bayesian statistics we know that [10]:

$$\sum_j \mathrm{P}(\mathrm{C}_j|\vec{x}) = 1$$

The probability of error for an output node (j) given an input vector $\vec{x}$ can be computed as:

$$\mathrm{P}_{error}(j, \vec{x}) = 1 - \mathrm{P}(\mathrm{C}_j|\vec{x})$$

or in equivalent form:

$$\mathrm{P}_{error}(j, \vec{x}) = \sum_{k \neq j} \mathrm{P}(\mathrm{C}_k|\vec{x}) \tag{2}$$

Based upon (1) and (2), when we take the partial derivative of the summed outputs $\sum_{k \neq j} \mathrm{z}_k$ of the feedforward network we are in actuality taking the partial derivative of P_{error}. By taking the partial derivative of the summed outputs $\sum_{k \neq j} \mathrm{z}_k$ with respect to an input feature we can measure the sensitivity of the output to the input feature. This provides a meaningful metric for feature saliency. Now let's take the derivative of the probability of error, P_{error}, with respect to a given input feature, x_i, to measure its saliency.

$$\begin{aligned} \frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} &= \frac{\partial}{\partial \mathrm{x}_i} \sum_{k \neq j} \mathrm{P}(\mathrm{C}_k|\vec{x}) \\ &= \frac{\partial}{\partial \mathrm{x}_i} \sum_{k \neq j} \mathrm{z}_k \end{aligned} \tag{3}$$

A multilayer perceptron is composed of layers which consist of input nodes (x), output nodes (z), weights (w), and thresholding elements (θ's). The output of a three layer neural network as depicted in Figure 1 is typically represented by [20]:

$$\mathrm{z}_j = \mathrm{f}_h(\sum_m \mathrm{x}_m^2 \cdot \mathrm{w}_{mj}^3 + \theta_j^3) \tag{4}$$

where

$$\begin{aligned}
\mathrm{z}_j &\equiv \text{output of node } j \text{ on third layer} \\
\mathrm{w}_{mj}^3 &\equiv \text{weight value from node } m \text{ on second layer to node } j \text{ on third layer} \\
\mathrm{x}_m^2 &\equiv \text{output of node } m \text{ on second layer} \\
\theta_j^3 &\equiv \text{threshold for node } j \text{ on layer 3} \\
\mathrm{f}_h &\equiv \text{sigmoid function}
\end{aligned}$$

Now if we substitute for z_k by using (4) in (3) we obtain

$$\frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} = \frac{\partial}{\partial \mathrm{x}_i} \sum_{k \neq j} \mathrm{f}_h(\sum_m \mathrm{x}_m^2 \cdot \mathrm{w}_{mk}^3 + \theta_j^3) \tag{5}$$

Using the fact that the derivative of the sigmoid function, f_h, is simply the sigmoid times one minus the sigmoid [20] [11] we obtain

$$\frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} = \sum_{k \neq j} \mathrm{z}_k \cdot (1 - \mathrm{z}_k) \cdot \frac{\partial}{\partial \mathrm{x}_i}(\sum_m \mathrm{x}_m^2 \cdot \mathrm{w}_{mk}^3 + \theta_j^3) \tag{6}$$

where

$$\begin{aligned}
\mathrm{z}_k &\equiv \text{output of node } k \text{ on layer 3} \\
\mathrm{x}_i &\equiv \text{feature } i \text{ on input layer} \\
\mathrm{x}_m^2 &\equiv \text{output } m \text{ on layer 2}
\end{aligned}$$

Applying the derivative yields

$$\frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} = \sum_{k \neq j} \delta_k^3 \cdot \sum_m \mathrm{w}_{mk}^3 \cdot \frac{\partial}{\partial \mathrm{x}_i}(\mathrm{x}_m^2)$$

where

$$\delta_k^3 = \mathrm{z}_k \cdot (1 - \mathrm{z}_k)$$

Continuing the process yields:

$$\begin{aligned}
\frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} &= \sum_{k \neq j} \delta_k^3 \cdot \sum_m \mathrm{w}_{mk}^3 \cdot \delta_m^2 \cdot \frac{\partial}{\partial \mathrm{x}_i}(\sum_n \mathrm{x}_n^1 \cdot \mathrm{w}_{nm}^2 + \theta_m^2) \\
&= \sum_{k \neq j} \delta_k^3 \cdot \sum_m \mathrm{w}_{mk}^3 \cdot \delta_m^2 \cdot \sum_n \mathrm{w}_{nm}^2 \cdot \delta_n^1 \cdot \mathrm{w}_{in}^1
\end{aligned} \tag{7}$$

where

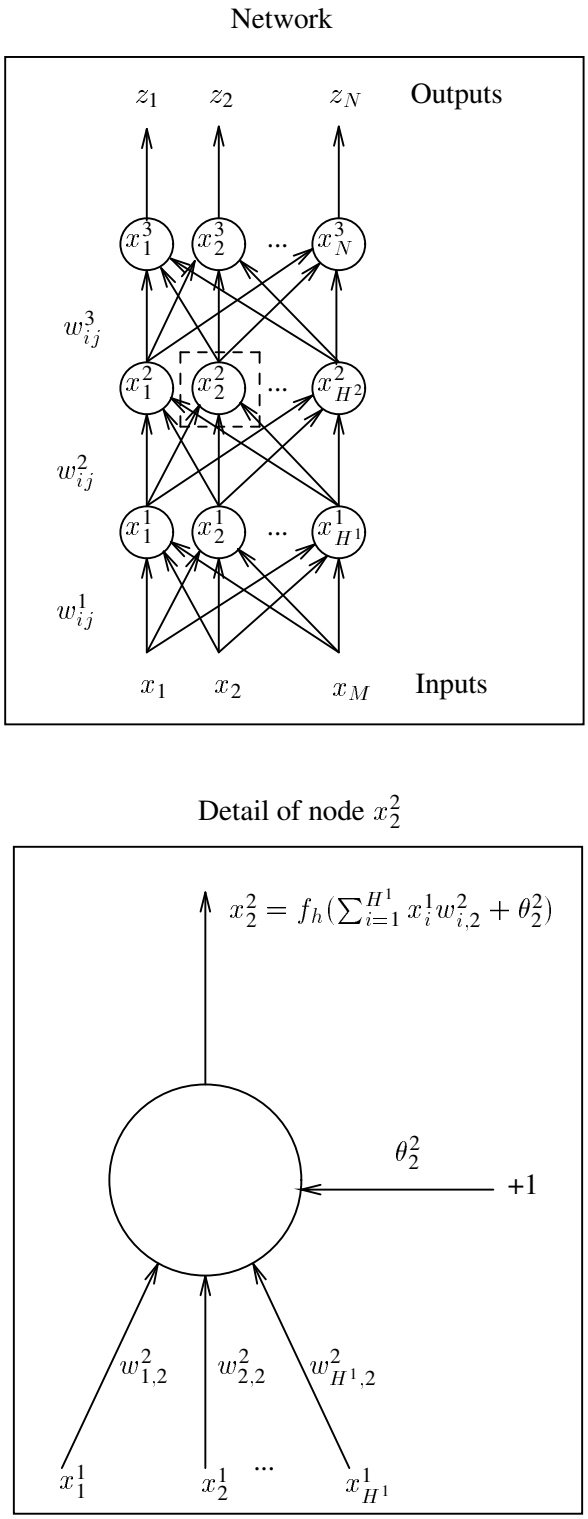

Figure 1: Two hidden layer feedforward network with accompanying expansion of a hidden node

$$
\begin{aligned}
m &\equiv \text{number of nodes on second hidden layer} \\
n &\equiv \text{number of nodes on first hidden layer} \\
\delta_m^2 &= \mathrm{x}_m^2 \cdot (1 - \mathrm{x}_m^2) \\
\delta_n^1 &= \mathrm{x}_n^1 \cdot (1 - \mathrm{x}_n^1)
\end{aligned}
$$

Note that the derivative of an ouput node (z_j) of a feedforward network with respect to a given input node (x_i) can be generalized to any number of layers as shown below:

$$\frac{\partial \mathrm{z}_j}{\partial \mathrm{x}_i} = \delta_j^N \cdot \sum_m \mathrm{w}_{mj}^N \delta_m^{N-1} \sum_n \mathrm{w}_{nm}^{N-1} \delta_n^{N-2} \cdots \sum_p \mathrm{w}_{po}^3 \delta_p^2 \sum_q \mathrm{w}_{qp}^2 \delta_q^1 \mathrm{w}_{iq}^1 \tag{8}$$

Also note that if the feedforward network uses linear transfer functions on the output nodes rather than sigmoidal transfer functions then δ_j^N = 1.

Now that we have a relationship established between an input feature and the probability of error, we can define a Bayesian-based saliency metric Ω_i for feature x_i of the input feature vector $\vec{x}$. The saliency metric presented is based on the magnitude of the changes in P_{error} as the feature x_i is varied over its range of values. Because we are measuring the sensitivity of the P_{error} to changes in x_i, a feature which causes little or no change in P_{error} is of little relevance while a feature which causes substantial changes in P_{error} is very relevant and should be retained.

The saliency (Ω_i) can be calculated as follows: Take each training vector ($\vec{x}$) in the training set S. For the ith feature, sample at various locations over the expected range of the feature while holding all other features in $\vec{x}$ constant. Now compute the magnitude of the partial derivative of the output z_k for the sample values. Sum the magnitude of the partial derivative of P_{error} over the outputs (j), the sampled values of x_i (D_i), and the training set S. See [18] for a discussion of this method of forming Ω_i.

$$\Omega_i = \sum_j \sum_{\vec{x} \epsilon \mathrm{S}} \sum_{\mathrm{x}_i \epsilon \mathrm{D}_i} \left| \frac{\partial \mathrm{P}_{error}(j, \vec{x})}{\partial \mathrm{x}_i} \right| \tag{9}$$

which can be rewritten as:

$$\Omega_i = \sum_j \sum_{\vec{x} \epsilon \mathrm{S}} \sum_{\mathrm{x}_i \epsilon \mathrm{D}_i} \left| \sum_{k \neq j} \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \tag{10}$$

where

$$
\begin{aligned}
j &\equiv \text{output nodes index} \\
k &\equiv \text{output nodes index (other than node } j\text{)} \\
\mathrm{D}_i &\equiv \text{Set of sample points for feature } \mathrm{x}_i
\end{aligned}
$$

We now present the metric Λ_i which is a simplification of (10).

$$\Lambda_i = \sum_j \sum_{\vec{x} \epsilon \mathrm{S}} \sum_{k \neq j} \sum_{\mathrm{x}_i \epsilon \mathrm{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \tag{11}$$

It can be shown (see Appendix) that (11) is an upper bound for (10). Thus the proposed metric, Λ_i is more sensitive to changes in the P_{error} as x_i is varied.

The saliency metric can be readily calculated by substituting the result from (8) into (11). The required weights and outputs are available in the normal feedforward algorithm and the required storage for intermediate results is quickly inserted into the feedforward algorithm.

3 Comparison With A Previous Saliency Metric

Now that we've established saliency from a probability perspective, how does it compare with a previously used saliency metric? Ruck [18] proposed a saliency metric which is given below:

$$\tilde{\Lambda}_i = \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{k} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \tag{12}$$

Note the similarity between the Ruck saliency metric and the metric defined in (11). Ruck's metric is actually the same as that obtained from the probability of error derivation to within a factor of n-1 where n is the number of output nodes for the feedforward network. To illustrate this relationship, consider the following:

$$\begin{aligned}
\Lambda_i &= \sum_{j} \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{k \neq j} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \\
&= \sum_{k \neq 1} \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| + \sum_{k \neq 2} \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| + \cdots + \sum_{k \neq n} \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \\
&= (n-1) \sum_{\vec{x}\epsilon \mathbf{S}} \sum_{k} \sum_{\mathrm{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathrm{z}_k}{\partial \mathrm{x}_i} \right| \\
&= (n-1) \cdot \tilde{\Lambda}_i
\end{aligned} \tag{13}$$

4 Results

The next question to be answered is whether or not this saliency measure is useful. Ruck [18] showed that indeed the saliency measure is useful and gives similar results when compared to conventional statistical techniques [12, 13, 14]. To illustrate the efficacy of this saliency metric, results are presented in this section. The classification task is to identify targets from non-targets in forward looking infrared (FLIR) and laser radar (LADAR) images where the targets consist of tanks, trucks and armored personnel carriers (APCs). A set of nine FLIR and eight LADAR features, given in Table 1, are extracted for this purpose. The features chosen were previously shown to be effective using conventional techniques [12, 13, 14]. These features are then ranked using a probability of error criterion. That is, each feature was used individually to determine whether the object was a target or not and the probability of error was calculated. The feature with the lowest probability of error was given the highest rank and *vice versa*. This ranking provides the baseline for evaluation of the utility of the saliency measure developed.

In the saliency test 100 networks were trained on the same data with different initial weights and a different order of presentation of the training data. Histograms of the ranking of three input features were

FLIR	LADAR	Feature	Description
X	X	Complexity	Ratio of border pixels to total object pixels
X	X	Length/Width	Ratio of object length to width
X	X	Mean Contrast	Contrast ratio of object's mean to local background mean
X		Maximum Brightness	Maximum brightness on object
X		Contrast Ratio	Contrast ratio of object's highest pixel to its lowest
X	X	Difference of Means	Difference of object and local background means
X	X	Standard Deviation	Standard deviation of pixel values on object
X		Ratio Bright Pixels/Total Pixels	Ratio of number of pixels on object within 10% of maximum brightness to total object pixels
X	X	Compactness	Ratio of number of pixels on object to number of pixels in rectangle which bounds object
	X	Length	Length of object in pixels
	X	Height	Height of object in pixels

Table 1: Features Evaluated in FLIR and LADAR Data Sets

constructed and an example is shown in Figure 2. The three features plotted are representative distributions. The ranking of the FLIR and LADAR features by the probability of error criterion and the saliency measure is shown in Table 2 and Table 3 respectively. Note that the two methods of ranking the FLIR data are in agreement for the top two and bottom two features, while the LADAR rankings are in general agreement with the probability of error criterion. As a further test of the validity of the saliency measure, the accuracy of classifiers designed using the top three features from each ranking were compared. Table 4 shows these results. Note that the differences between the top three features of either ranking for all three types of classifiers are negligible. Hence, the saliency measure does provide a useful measure of the significance of input features.

Feature	P_e Criterion Rank	Saliency Measure Rank
Length/Width	7	7
Standard Deviation	2	5
Maximum Brightness	5	2
Compactness	0	1
Complexity	8	8
Mean Contrast	6	3
Contrast Ratio	4	4
Ratio Bright Pixels/Total Pixels	1	0
Difference of Means	3	6

Table 2: Ranking of FLIR Features

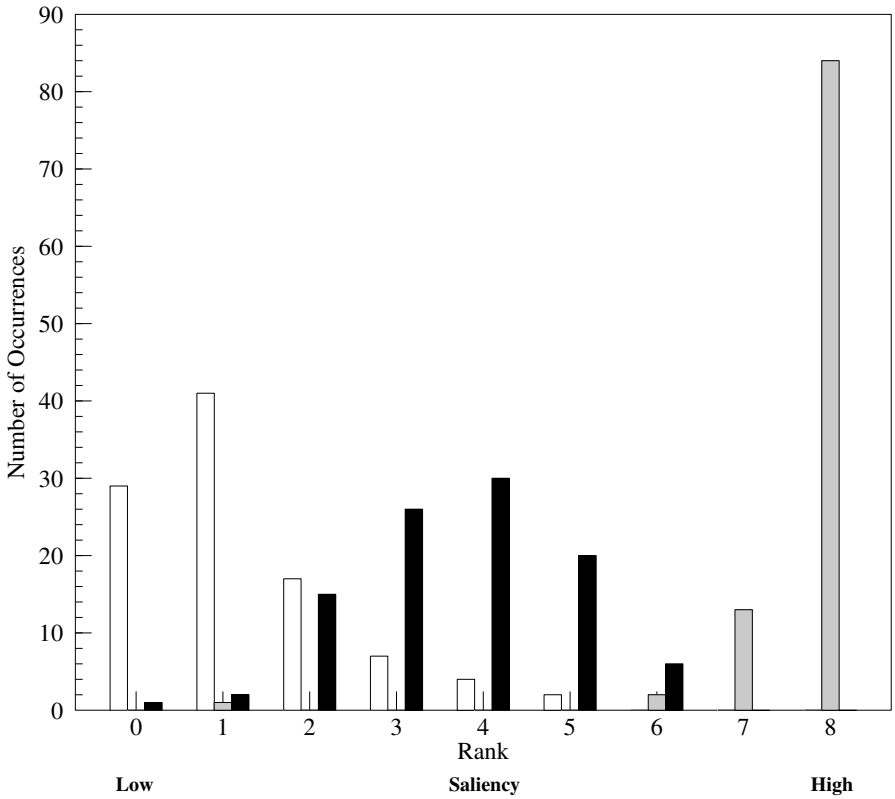

Figure 2: Rank Histograms for Three FLIR Imagery Features. Each feature is plotted using a different fill pattern. The FLIR feature represented by the unfilled bars shows little saliency and, thus, did not provide much information for classifying the input patterns. The darkly shaded bars represent a FLIR feature with a moderate degree of saliency providing some information for classification. The feature shown with the lightly shaded bars has a large saliency and provided much information for classifying the FLIR objects.

Feature	P_e Criterion Rank	Saliency Measure Rank
Length	2	6
Height	1	3
Length/Width	7	5
Compactness	0	2
Complexity	5	4
Blob Std. Dev.	4	1
Difference of Std Dev	6	7
Difference of Means	3	0

Table 3: Ranking of LADAR Features

Classifier	Best 3 P_e Features	Best 3 Saliency Measure Features
Bayesian w/ Parzen Windows	92.5%	93.4%
Nearest Neighbor	92.7%	92.5%
Multilayer Perceptron	90.7%	90.2%

Table 4: Classification Accuracies of Two Feature Subsets

5 Discussion

In this paper, we have shown that the Ruck saliency metric, while chosen intuitively, was actually a metric which measured the sensitivity of the probability of error to a given input feature. The results presented here have shown the saliency metric to be consistent and useful. Indeed, when compared with the traditional statistical approach to ranking input features, the proposed metric performed as well and produced similar overall results. This result was to be expected because the intent of the paper was to show the connection between the Bayesian properties of the multilayer perceptron and traditional Bayesian statistical methods. This result has significance because it further illustrates that the multilayer perceptron is performing as a Bayes optimal discriminant function. Furthermore, the saliency metric can be used to determine the usefulness of input features and hidden nodes.

One of the issues which should be looked at when considering the efficacy of the proposed saliency metric is its ability to handle correlated data. With this in mind, the correlation matrices for the FLIR and LADAR data sets are given in Table 5 and Table 6. Using the Student's t Distribution [7] to find the minimum value for the correlation coefficient **r** which will reject the hypothesis that the **r** value is due to chance results in a $\mathbf{r}_{min}$ of $\pm$ 0.101 for the FLIR data and a $\mathbf{r}_{min}$ of $\pm$ 0.117 for the LADAR data. Thus, if $\mid \mathbf{r} \mid \geq 0.101$ for FLIR or $\mid \mathbf{r} \mid \geq 0.117$ for LADAR, then the correlation value is not due to chance and is significant. The results given in Tables 5 and 6 show that some correlation does exist between the input features and yet the saliency metric was able to obtain results equal to the three most popular statistical methods. On closer examination of the correlation matrices it became evident that, in this example, the backpropagation neural network was able to choose one feature as significant while suppressing correlated features. While correlation should not be used to infer causality, e.g. two features may be correlated but have nothing to do with the classification decision, it is useful in determining if features do influence one another in the classification process.

One limitation of the saliency metric is that it is computed over the entire range of each feature. Not all portions of the range of a given feature are near a class boundary which makes them ineffective in determining relevancy. One solution would be to measure the saliency metric near the class boundaries in a given classification problem. If there is a good understanding of the a priori probability distributions, this information could be used to measure the saliency near the class boundaries. Another possible method of determining the decision boundaries is measuring the distance of a feature vector to the hyperplanes of the nodes in a multilayer perceptron. The activation of a node in a multilayer perceptron is actually the distance of the input feature vector to the hyperplane of the neuron multiplied by the magnitude of the weight vector associated with the node. However, knowledge of the distance from the feature vector to the hyperplane of each node is a necessary but not sufficient condition for determining the actual classification boundary. This is true because the output nodes are taking nonlinear combinations of the input feature vector to ultimately determine the classification. The problem is a difficult one and requires some preparation and analysis of

Feature	0	1	2	3	4	5	6	7	8
0	1.000	0.028	0.170	-0.243	-0.254	-0.033	0.154	-0.258	-0.036
1	0.028	1.000	0.863	-0.178	-0.485	0.829	0.793	-0.435	0.863
2	0.170	0.863	1.000	-0.207	-0.568	0.703	0.712	-0.494	0.761
3	-0.243	-0.178	-0.207	1.000	0.024	-0.132	-0.373	0.315	-0.061
4	-0.254	-0.485	-0.568	0.024	1.000	-0.524	-0.593	0.501	-0.468
5	-0.033	0.829	0.703	-0.132	-0.524	1.000	0.757	-0.339	0.950
6	0.154	0.793	0.712	-0.373	-0.593	0.757	1.000	-0.608	0.644
7	-0.258	-0.435	-0.494	0.315	0.501	-0.339	-0.608	1.000	-0.270
8	-0.036	0.863	0.761	-0.061	-0.468	0.950	0.644	-0.270	1.000

Table 5: FLIR Input Feature Correlation Matrix: 0 - Length/Width, 1 - Std Deviation, 2 - Max Brightness, 3 - Compactness, 4 - Complexity, 5 - Mean Contrast, 6 - Contrast Ratio, 7 - Ratio Bright Pixels/Total Pixels, 8 - Difference of Means

Feature	0	1	2	3	4	5	6	7
0	1.000	0.328	0.653	-0.171	-0.492	0.494	0.427	0.199
1	0.328	1.000	-0.403	-0.286	-0.113	0.228	0.159	-0.043
2	0.653	-0.403	1.000	0.004	-0.401	0.378	0.270	0.214
3	-0.171	-0.286	0.004	1.000	0.255	-0.238	-0.200	-0.168
4	-0.492	-0.113	-0.401	0.255	1.000	-0.136	-0.629	-0.509
5	0.494	0.228	0.378	-0.238	-0.136	1.000	0.023	-0.125
6	0.427	0.159	0.270	-0.200	-0.629	0.023	1.000	0.734
7	0.199	-0.043	0.214	-0.168	-0.509	-0.125	0.734	1.000

Table 6: LADAR Input Feature Correlation Matrix: 0 - Length (m), 1 - Height (m), 2 - Length/Width Ratio, 3 - Compactness, 4 - Complexity, 5 - Blob Std Dev, 6 - Abs Value(Blob Std Dev - Background Std Dev), 7 - Abs Value(Blob Mean - Background Mean)

the classification problem at hand before a reduced or more efficient data set for saliency measurement can be found.

6 Conclusion

In this paper, a technique for ranking the importance of features for a multilayer perceptron has been developed from a Bayesian perspective. The saliency measure has been shown empirically to be both consistent and useful. When compared with the traditional method of ranking input features by the probability of error criterion, it performed as well and resulted in a similar ranking for the input features.

One extension to the saliency measure is its application to the hidden nodes in a multilayer perceptron. The outputs of the hidden nodes are inputs to the following layer; hence, the saliency of the hidden nodes can also be computed. On-going research into this area will, hopefully, provide a method for automatically sizing the hidden layers in a multilayer perceptron.

Appendix

Proof of Upper Bound For Ω_i

We wish to show that Λ_i is an upper bound for Ω_i.
Starting with Equation 10 in the text we have

$$\Omega_i = \sum_j \sum_{\vec{x} \epsilon \mathbf{S}} \sum_{\mathbf{x}_i \epsilon \mathbf{D}_i} \left| \sum_{k \neq j} \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \tag{14}$$

For the purposes of this proof consider the absolute value term in Equation 14 as the variable

$$\gamma_i = \left| \sum_{k \neq j} \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \tag{15}$$

We now show that γ_i is bounded.
Because $\mathbf{z}_k$ is a sigmoid function, each $\frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i}$ exists and is finite.
Therefore

$$\sum_{k \neq j} \left| \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \leq C \tag{16}$$

and by the triangle inequality [8] γ_i is bounded by Equation 16

$$\gamma_i \leq \sum_{k \neq j} \left| \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \leq C \tag{17}$$

Combining Equations (14 - 17) yields

$$\Omega_i = \sum_j \sum_{\vec{x} \epsilon \mathbf{S}} \sum_{\mathbf{x}_i \epsilon \mathbf{D}_i} \left| \sum_{k \neq j} \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \leq \sum_j \sum_{\vec{x} \epsilon \mathbf{S}} \sum_{\mathbf{x}_i \epsilon \mathbf{D}_i} \sum_{k \neq j} \left| \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \tag{18}$$

Rearranging the order of summation and choosing the right hand side, yields Equation 11 in the article.

$$\Lambda_i = \sum_j \sum_{\vec{x} \epsilon \mathbf{S}} \sum_{k \neq j} \sum_{\mathbf{x}_i \epsilon \mathbf{D}_i} \left| \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_i} \right| \tag{19}$$

Hence, Λ_i is an upper bound for Ω_i in the text.

References

[1] Pierre A. Devijver and Josef Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall International, London, 1982.

[2] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[3] Donald H. Foley. Considerations of sample and feature size. *IEEE Transactions on Information Theory*, IT-18:618–626, September 1972.

[4] Herbert Gish. A probabalistic approach to the understanding and training of neural network classifiers. In *Proceedings of International Conference on Acoustics Speech and Signal Processing 90*, pages 1361–1364, 1990.

[5] R. Paul Gorman and Terrence J. Sejnowski. Learned classification of sonar targets using a massively parallel network. *IEEE Transactions on Acoustics Speech and Signal Processing*, 36(7):1135–1140, July 1988.

[6] William Y. Huang and Richard P. Lippmann. Comparisons between neural net and conventional classifiers. In *Proceedings of IEEE First International Conference on Neural Networks*, San Diego, CA, June 1987.

[7] Sam Kash Kachigan. *Statistical Analysis*. Radius Press, 1986.

[8] Steven R. Lay. *Analysis with an Introduction to Proof.* Prentice Hall, New Jersey, 2nd edition, 1990.

[9] Richard P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, pages 47–63, November 1989.

[10] Paul L. Meyer. *Introductory Probability and Statistical Applications*. Addison-Wesley Pubishing Company, 2nd edition, 1970.

[11] Steven K. Rogers and Matthew Kabrisky. *An Introduction to Biological and Artificial Neural Networks for Pattern Recognition*, volume TT 4. SPIE Optical Engineering Press, 1991.

[12] Michael C. Roggemann. *Multiple Sensor Fusion for Detecting Targets in FLIR and Range Images*. PhD thesis, Air Force Institute of Technology, June 1989.

[13] Michael C. Roggemann, J.P. Mills, M. Kabrisky, S.K. Rogers, and J.A. Tatman. An approach to multiple sensor target detection. In *Proceedings of SPIE Conference on Sensor Fusion*, volume 1100, 1989.

[14] Michael C. Roggemann, J.P. Mills, M. Kabrisky, S.K. Rogers, and J.A. Tatman. Multiple sensor tactical target detection in FLIR and range images. In *Proceedings of Tri-Service Data Fusion Symposium*, Johns Hopkins University, May 1989.

[15] Michael W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on Neural Networks*, 1(1):28–43, March 1990.

[16] Dennis W. Ruck. Multisensor target detection and classification. Master's thesis, Air Force Institute of Technology, 1987.

[17] Dennis W. Ruck. Multisensor fusion target classification. In *SPIE Symposium on Optics, Electro-Optics, and Sensors*, 1988.

[18] Dennis W. Ruck, Steven K. Rogers, and Matthew Kabrisky. Feature selection using a multilayer perceptron. *The Journal of Neural Network Computing*, 2(2):40–48, Fall 1990.

[19] Dennis W. Ruck, Steven K. Rogers, Matthew Kabrisky, Mark E. Oxley, and Bruce W. Suter. The multilayer perceptron as an approximation to a Bayes optimal discriminant. *IEEE Transactions on Neural Networks*, 1(4), 1990.

[20] David E. Rumelhart, James L. McClelland, and the PDP Research Group. *Parallel Distributed Processing*, volume 1: Foundations. MIT Press, 1986.

[21] Julius T. Tou and Rafael C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley Pubishing Company, 1st edition, 1974.