Neural Comput & Applic (1996)4:72–82

Neural
Computing
& Applications

Interpretation and Knowledge Discovery from the Multilayer Perceptron Network: Opening the Black Box

Marilyn Lougher Vaughn

Computing Information Systems Management Group, Cranfield University, RMCS, Swindon, UK

This paper interprets the outputs from the multilayer perceptron (MLP) network by finding the input data features at the input layer of the network which activate the hidden layer feature detectors. This leads directly to the deduction of the significant data inputs, the inputs that the network actually uses to perform the input/output mapping for a classification task, and the discovery of the most significant of these data inputs. The analysis presents a method for providing explanations for the network outputs and for representing the knowledge learned by the network in the form of significant input data relationships. During network development the explanation facilities and data relationships can be used for network validation and verification, and after development, for rule induction and data mining where this method provides a potential tool for knowledge discovery in databases (KDD).

Keywords: Data mining; Explanation facilities; Interpretation; Knowledge discovery; Rule induction; Validation and verification

1. Introduction

The multilayer perceptron (MLP) network is one of the most widely used neural networks in the field of neural computing. The MLP came into use in the mid 1980s with the development of the backpropagation learning algorithm by several independent researchers in the field [1,2] and it has been used extensively in many different kinds of problems [3–6]. This paper examines how a multilayer perceptron performs the input/output mappings for a classification task with binary input data and the analysis is extended to the general case of a multilayer perceptron that classifies continuously valued inputs to continuously valued outputs in the ranges -1.0 to $+1.0$. The paper starts from first principles and provides a useful introduction to neural computing.

2. A Simple Classification Task

A simple recognition problem with binary input data is used as an example to demonstrate how a MLP with one hidden layer performs a classification task. The network is required to recognise the ten digits when each digit is represented exactly as shown in Fig. 1 [7], where a white pixel has a network input value $+0$ and a black pixel has a network input value $+1$.

The architecture of the MLP network that performs the task is shown in Fig. 2. The network is a fully connected, feedforward network with three layers of artificial neurons, an input layer, a hidden layer and an output layer. The input layer has 20 neurons, one for each input pixel, and acts as an input buffer, passing the pixel input values to each neuron in the hidden layer across the first layer of network connections. Each of the ten neurons in the output layer of the network indicates the digit that is to be recognised, for example, when the digit two is presented at the input layer the second output neuron is required to output the value $+1$

Correspondence and offprint requests to: M. Lougher Vaughn, Computing Information Systems Management Group, Cranfield University, RMCS, Shrivenham, Swindon SN6 8LA, UK.

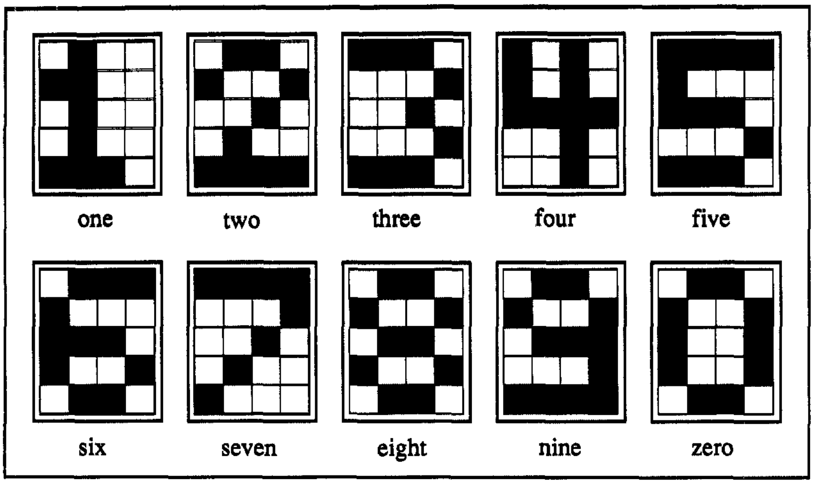

Fig. 1. Binary representation of ten digits.

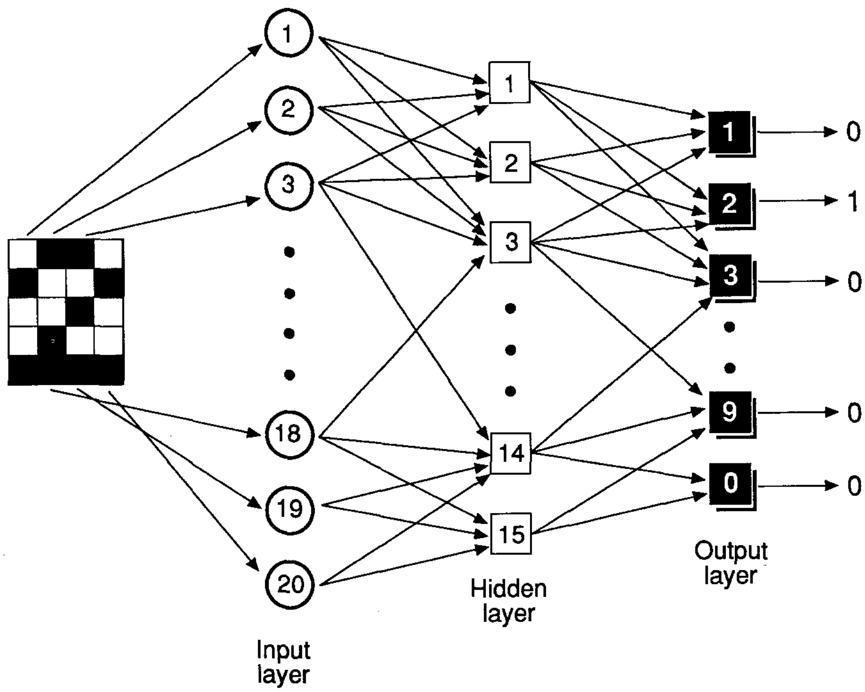

Fig. 2. The multilayer perceptron network for the digit recognition.

and the remaining output neurons are required to output the value +0. In this way, the output values of the ten output layer neurons indicate the class to which a digit belongs. An arbitrary choice of 15 hidden layer neurons is sufficient to demonstrate how the MLP performs the classification task but, in practise, the optimum number of hidden layer neurons is determined empirically [8].

The square [9] nodes in Fig. 2 indicate that the artificial neurons in the hidden and output layers process inputs received from neurons in the layer below. A processing neuron multiplies each input value, x_i, from neuron i, by a weight, w_i, representing the synaptic strength of the connection, as shown in Fig. 3. The combined input sum, $\Sigma\, x_i w_i$, is then input to an activation function that determines the activation level of the processing neuron. A widely used activation function is the sigmoid activation function, shown in Fig. 4, a continuous and differentiable function which produces an activation level between +0.0 and +1.0 depending on the value of the combined input sum and the processing neuron's threshold value, T. When the combined input sum is above T and approaches $+\infty$ the activation level

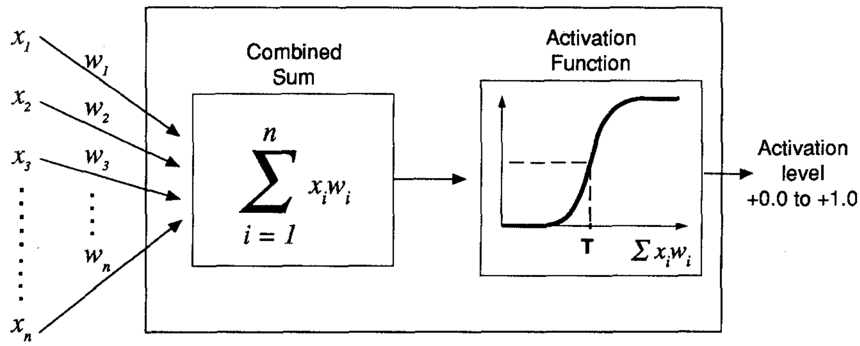

Fig. 3. Artificial processing neuron with activation function.

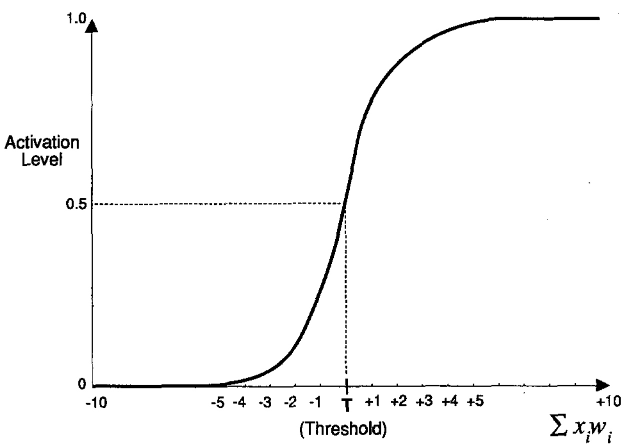

Fig. 4. Sigmoid activation function with threshold T.

$\rightarrow +1.0$; when the combined input sum is below T and approaches $-\infty$ the activation level $\rightarrow +0.0$; when the combined input sum is exactly at the threshold the activation level is $+0.5$.

For the MLP network shown in Fig. 2 to perform the digit classification task, the values of the 450 network connection weights and 25 thresholds must be found that produce the required network outputs at the output layer neurons corresponding to each input digit. This is done by supervised training using the backpropagation learning algorithm [1,2,4] in which the network is shown examples of the training inputs with the associated network outputs, the training input/output pairs. After 1000 presentations of the training pairs [10], the classifying output neurons, in the network shown in Fig. 2, have activation levels exceeding $+0.90$ and the other output neurons have activation levels less than $+0.06$ when the corresponding digit vector is input to the trained network. In practice [8,11], further optimisation would take place at this stage of training but these activations are sufficiently developed to explain how the MLP achieves the classification task.

3. Explaining how the MLP Network Performs the Classification Task

To find how the MLP network performs the input/output mapping, the network outputs are first examined for one of the digits, the digit eight, an arbitrary choice. When the vector (0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0) representing the digit eight image is presented to the trained network the 8th output neuron has an activation level of $+0.918$, as shown in Fig. 5. The reasons for this are now examined.

A high activation level at a processing neuron means that the combined input sum to that neuron significantly exceeds the neuron's threshold value, T, as shown in Fig. 4. It can be seen in Fig. 5 that there is an extra input neuron, the bias neuron, which inputs a value of $+1.0$ to each processing neuron in the hidden and output layers. The bias neuron contributes an extra input to the combined sum at each processing neuron and has the effect of setting the threshold, T, to $+0.0$ and the bias connection weight to $-T$. So, with the bias input, *a high activation level at a processing neuron means that the combined input sum at the neuron significantly exceeds a threshold of* $+0.0$. For this reason, the combined input sum at the 8th output neuron is first examined.

3.1. Discovery of the Hidden Layer Feature Detectors

The high activation level of $+0.918$ at the 8th output neuron in Fig. 5 is the output activation from the sigmoid of a combined input sum of $+2.441$. On further examination, this sum is found to be made up of a positive part which is $+5.2797$ and a negative part which is -2.8687. This shows that the positive classification at the eighth output

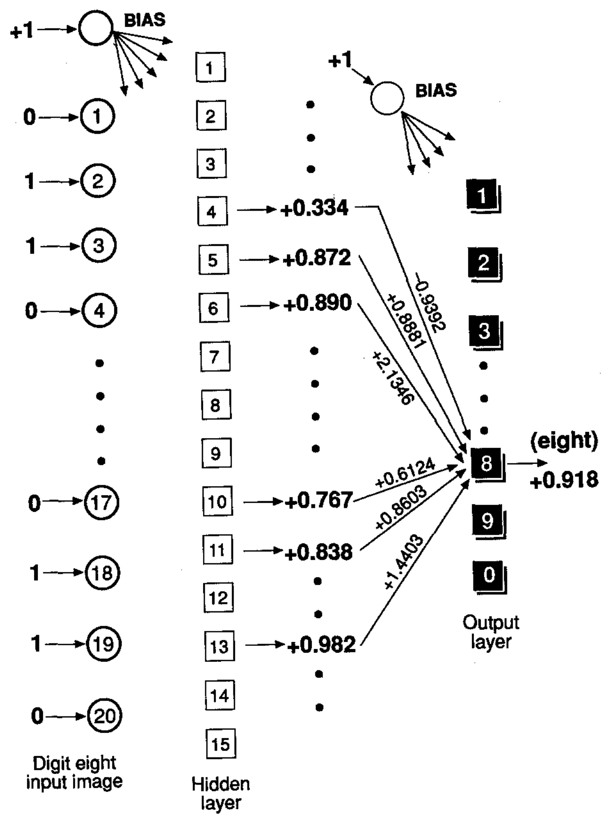

Fig. 5. Highest activation levels at the hidden and output layer neurons in response to the digit eight.

neuron is completely determined by the positive part of the combined input sum and the negative part only contributes a dampening effect. For this reason the positive part of the combined input sum at the eighth output neuron is examined further.

As explained in Sect. 2, the combined input sum at an output neuron is the sum of contributions x_iw_i from each of the hidden layer neurons in the layer below, where x_i is the activation at the hidden layer neuron, i, and w_i is the connection weight to the output neuron. All of the hidden layer neurons have positive activation levels, by definition, because the activation functions at the hidden layer are sigmoidal. This means that *contributions to the positive part of the combined input sum at the eighth output layer neuron come from hidden layer neurons which connect to the output layer neuron with positive weights.* (The negative, or dampening, contributions come from hidden layer neurons connecting to the output neuron with negative weights.)

As shown in Fig. 5, the 5th, 6th, 10th, 11th and 13th neurons have the highest activation levels in the hidden layer and are the only neurons that connect to the 8th output neuron with positive weights. (The next highest hidden layer activation, +0.3342, at the 4th hidden layer neuron connects to the 8th output neuron with a negative weight, −0.9392.) This means that the 5th, 6th, 10th, 11th and 13th hidden layer neurons contributed all of the positive part of the combined input sum at the 8th output neuron and determine the positive output classification for the digit eight. These neurons are known as the *hidden layer feature detectors* [4,12] for the reasons given in Sect. 3.3. The positive part of the combined input sum leading to the high activation levels at the five feature detectors is further investigated.

3.2. Discovery of the Input Features of the Digit Eight

The combined input sum at a hidden layer neuron is the sum of contributions x_iw_i from each of the input layer neurons in the layer below, where x_i is the pixel input value from the input layer neuron, i, and w_i is the connection weight to the hidden layer neuron. Each of the input pixel values representing the digit eight is either +0 or +1, exactly. The contribution of a white input pixel with input value +0 to the weighted sum is +0.0 so that the white pixels in the input image make no contribution to the combined input sum, and, consequently, take no part in the output classification.

This means that *contributions to the positive part of the combined input sum at a hidden layer neuron come from black input pixels, with an input value of* +1, *which connect to the hidden layer neuron with positive weights.* (Negative, or dampening, contributions to the combined sum at a hidden layer neuron come from black input pixels, with an input value of +1, which connect to the neuron with negative weights.) The positive and negative connection weights to each hidden layer neuron are shown in Fig. 6, in a grid dimension similar to the input digits [4], where a black pixel signifies a positive connection weight and a white pixel signifies a negative connection weight.

The (black) input pixels with an input value of +1 that connect to the hidden layer neuron with positive (black) connection weights are shown in Fig. 7. These pixels represent a subset, or *input feature*, of the digit eight input vector that determine the positive part of the combined input sum that activates the 5th hidden layer neuron. The four features of the digit eight at the 6th, 10th, 11th and 13th hidden layer neurons, respectively, are also shown in Fig. 7.

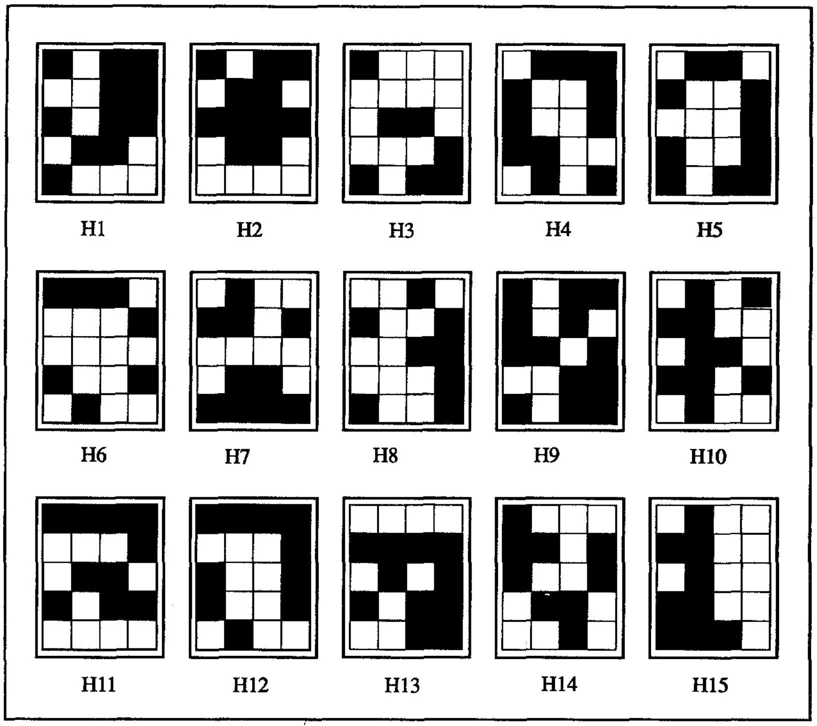

Fig. 6. Connection weights from the input layer to each hidden layer neuron (a black pixel represents a positive weight and a white pixel represents a negative weight).

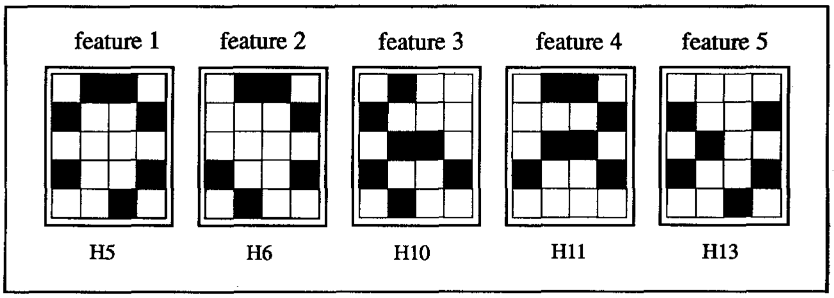

Fig. 7. Input features of the digit eight detected by the hidden layer neurons indicated.

3.3. The Concatenation of the Input Features

The input features, as shown in Fig. 7, exhibit considerable overlap and redundancy which gives the MLP network its well known properties of robustness and graceful degradation. *By concatenating all of the five input features together, the original image of the digit eight is reconstructed* showing that all of the black input pixels in the digit eight input vector make a contribution to the network classification of the digit eight. This demonstrates that the MLP network, shown in Fig. 2, shares the classification task of the digit eight between the 5th, 6th, 10th, 11th and 13th hidden layer. For this reason, these hidden layer neurons are known as the hidden layer feature detectors.

The input features for all of the ten input digits are found in the same way and for all of the digits the original image is reconstructed when the input features for each digit are concatenated. It is only the black pixels in the input features that contribute to the output classification since, as explained in Sect. 3.2, the white input pixels make no contribution to the combined input sum at the hidden layer. The concatenation of the input features has revealed the input data at the input layer that the network actually uses to make the output classification and these inputs are called the *significant inputs* in this paper. In the classification of the digit eight, for example

concatenation (feature 1, feature 2, feature 3, feature 4, feature 5)

= (0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0)

= the significant inputs for the digit eight

identifying the ten significant inputs, as indicated, that the MLP network actually uses to classify the digit eight out of the possible 20 inputs in the input vector. In a general classification problem not all of the non-zero input data are necessarily significant.

4. Discovery of the Significant Inputs for a General Classification Problem

An example of a general classification problem using binary input data (see Sect. 4.2 for continuously valued input data) is the mapping of life assurance client data into classes of life assurance risk. An example of client data is shown in Table 1 [13], where the binary input values of +0 and +1 represent client data attributes within ranges of data categories such as gender, age, weight, height and medical history. Client profiles vary and different client profiles can map to the same classification output.

After training a MLP network on a representative client data set the input features of any particular client input data profile in the training set that produces an output life assurance risk classification can be discovered using the method explained in Sect. 3. It is only the siginficant inputs that are present in the input features since, by definition, these are the inputs that the network uses in making the classification. This means that for the general problem, *the concatenation of the input features determines the significant inputs in the input layer data.* This is one of the main results of this paper, since the significant inputs are generally unknown.

The significant inputs can be found in a similar way for a MLP with binary input values -1, $+1$ and for the general case of a MLP that classifies continuously valued inputs to continuously valued outputs in the ranges -1.0 to $+1.0$, as discussed in Sect. 4.3–4.5.

Table 1. Life assurance client data profile

Attributes	Attribute categories	Attribute value
gender	female	0
	male	1
age	under 30	0
	30–49	1
	50 and over	0
weight/height	over weight/height	0
	normal weight/height	1
	underweight/height	0
sedentary occupation	yes (1), no (0)	1
previous policy holder	yes (1), no (0)	1
hazardous pursuits	yes (1), no (0)	0
prescription drugs/ medicine	yes (1), no (0)	1
daily alcohol consumption	high	0
	medium	1
	low	0
smoking habits	non smoker	1
	less than 10/day	0
	over 10/day	0

4.1. Uses of the Analysis

The analysis leading to the discovery of the significant inputs presents a method for interpreting the network classification outputs and for providing full, or limited, explanation facilities at any stage of network development and use, as shown in Sect. 5. The knowledge learned by the network from the training data is presented in the form of significant input data relationships in Sect. 6. During training and testing the data relationships and explanation facilities can be used for network validation and verification, as discussed in Sect. 6.1, and after network development for knowledge discovery and rule induction, as discussed in Sect. 6.2 and 6.3.

Some of the significant inputs are more important to the network than others in making an output classification and the discovery of the most significant inputs is presented in Sect. 7. This is another main result of this paper and the enhanced knowledge can be incorporated into the explanation facilities, the significant input data relationships and the rules.

4.2. Discovery of the Significant Continuously Valued Inputs from +0.0 to +1.0

For a MLP network with continuously valued inputs between +0.0 and +1.0 the significant continuously valued inputs are found as follows. The input features in the continuously valued inputs, x_i, between +0.0 and +1.0, are the positive x_i, which connect to the feature detectors with positive weights, w_i. As in the case of binary input data, the concatenation of the input features determines the significant continuously valued inputs, x_i, which the network actually uses in making the input/output mapping.

4.3. Discovery of the Significant Binary Input Values −1 and +1

For a MLP network with binary input values −1 and +1 the significant binary inputs are found as follows. The input features in the binary valued inputs, x_i, −1 and +1, are the positive x_i, with input value +1, which connect to the feature detectors with positive weights, w_i, *and* the negative x_i, with input value −1, which connect to the feature detectors with negative weights, w_i. In the above digit classification example, the white input pixels have the input value −1, so that white pixels with negative connection weights to a feature detector are also present in the input feature. The concatenation of the input features determines the negative as well as the positive significant binary valued inputs, x_i, which the network actually uses in making the input/output mapping.

4.4. Discovery of the Significant Continuously Valued Inputs from −1.0 to +1.0

For a MLP network with continuously valued inputs between −1.0 and +1.0 the significant continuously valued inputs are found as follows. The input features in the continuously valued inputs, x_i, between −1.0 and +1.0 are the positive x_i, which connect to the feature detectors with positive weights, w_i, *and* the negative x_i, which connect to the feature detector with negative weights, w_i. As in the case of binary inputs −1 and +1, both negative x_i as well as positive x_i can be present in the input features and the concatenation of the input features determines the negative and the positive continuously valued significant inputs, x_i, which the network actually uses in making the input/output mapping.

4.5. Discovery of the Feature Detectors for Activations from −1.0 to +1.0

The hidden layer neurons and output layer neurons in a MLP network have activation values in the range −1.0 to +1.0 when a symmetric activation function, such as tanh [14,15] is used at the processing neuron. Activations in this range from hidden layer neurons affect the discovery of the feature detectors. In this case the feature detectors are the hidden layer neurons with positive activation values, x_i, which connect to the classifying output neuron with positive weights, w_i, *and* the hidden layer neurons with negative activation values x_i, which connect to the neuron with negative weights, w_i.

5. Interpretation and Explanation Facilities from the MLP Network

The analysis leading to the discovery of the significant inputs in Sect. 4 presents a method for interpreting the network classification outputs and for providing full, or limited, explanation facilities at any stage of use. Without loss of generality the classification of the digit eight is used as an example. The analysis directly provides the following interpretation of the network outputs when the digit eight input vector is presented at the input layer:

> The 8th output neuron in the output layer has a high activation level of +0.918 (and all other neurons in the output layer have the low activations)
>
> ⇒ Class 8
> because
>
> the 5th neuron in the hidden layer detects feature 1 with an activation level of +0.872
>
> and
>
> the 6th neuron in the hidden layer detects feature 2 with an activation level of +0.890
>
> and
>
> the 10th neuron in the hidden layer detects feature 3 with an activation level of +0.767
>
> and
>
> the 11th neuron in the hidden layer detects feature 4 with an activation level of +0.838
>
> and
>
> the 13th neuron in the hidden layer detects feature 5 with an activation level of +0.982
>
> because
>
> concatenation (feature 1, feature 2, feature 3, feature 4, feature 5)
> = (0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0)
> = the significant inputs for the input data

In general, the significant inputs are a subset of the input data and the inclusion of the meaning of each of the significant inputs in the interpretation or explanation facilities will enhance the meaning for the user. For the general user of the network, it may be appropriate to provide more limited explanations omitting the information about the hidden layer nodes, for example, as follows:

> The 8th output neuron at the output layer has a high activation level of +0.918 (and all other neurons in the output layer have the low activations)
>
> ⇒ Class 8
> because
> the significant inputs at the input layer
> = (0,**1**,**1**,0,**1**,0,0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0)

Again, the inclusion of the meaning of each of the significant inputs in the explanations will further enhance the meaning for the user.

6. Knowledge Discovery for the MLP Network

Using the method presented in Sect. 4, the significant inputs can be discovered from all of the training input data at any stage of the training process. The knowledge learned by the network from the training data is embodied in the data relationships between the significant inputs in each input training vector and the associated network outputs, as follows:

> significant training inputs ⇒ associated network outputs

In the example of the training input vector for the digit eight, the following data relationship is deduced from the analysis:

> significant training inputs =
> (0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0)
>
> ⇒ network outputs (⇒ Class 8)

This relationship is non linear due to the effect of the sigmoid functions at each processing layer of neurons, and can be explicitly stated for each network output as a function of the connection weights and thresholds. These relationships can be validated and verified during training with the assistance of the domain experts.

6.1. Network Validation and Verification

At any stage of network training the explanation facilities, discussed in Sect. 5, reveal the knowledge discovered about the hidden layer feature detectors and the significant input data relationship for each training input/output pair and this knowledge can be used to validate and verify the network outputs with the help of domain experts. For example, the MLP may have learned some of the input/output mappings from invalid data relationships, in which case, action can be taken to improve the network learning, such as including more training examples in the training process. The explanation facilities at all stages of network development will show the network developer how the network uses the hidden layer neurons to make the mappings and will enhance the understanding of the network.

During network testing the explanation facilities and data relationships can be used to verify that the network is correctly generalising the knowledge learned by the network from the training process. For example, testing is not expected to reveal new significant input data relationships since network generalisation is inferred from the network knowledge acquired during training.

6.2. Knowledge Discovery in Databases (KDD)/ Data Mining

When the training and testing of the MLP network is completed, the significant training input data relationships, as presented in Sect. 6, embody the knowledge that the network has learned from the training process and previously unknown, or hidden, data relationships may be revealed. The analysis leading to this discovery provides a method for knowledge discovery in databases (KDD), or data mining.

6.3. Rule Induction for the MLP Network

When the MLP network development is completed, the analysis leading to the discovery of the significant inputs, presented in Sect. 4, also provides a method for rule induction. As an example, the following rule is induced for the significant inputs in the training vector for the digit eight:

> If
> the significant input data at the input layer
> = (0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0,**1**,0,0,**1**,0,**1**,**1**,0)
> then
> the 8th output neuron in the output layer has an activation level of +0.918 (and all other neurons in the output layer have the low activations)
> ⇒ Class 8

Meaning can be attributed to each of the significant inputs at the input layer to enhance the meaning of the rule. Similar rules can be induced for all of the training input data that map to the same classification output and similar rules can be induced for all classification outputs.

7. Discovery of the Most Significant Data Inputs

Some of the inputs that the MLP network uses in making an input/output mapping are more important to the network than others and these are the *most* significant inputs. For example, the discovery that some symptoms are more important than others to a network making a medical diagnosis provides valuable information for the domain expert which can be put to strategic use [15]. The most significant inputs *can* be found directly, in recall mode, by removing the significant inputs in turn from the input vector and finding the change in the activations at the output layer neurons. Combinations of significant inputs can also be examined in a similar way. However, the number of possible combinations is combinatorially large and a more strategic approach is to find the most significant inputs from within each feature in order of feature detector significance. This also contributes to an enhanced understanding of the network, and the knowledge of the most significant inputs and features can be incorporated into the explanation facilities, significant input data relationships and rules, discussed in Sect. 5 and 6.

7.1. Discovery of the Most Significant Feature Detectors

Each of the hidden layer feature detectors makes a different contribution to the output classification depending on the contribution $x_i w_i$ to the positive part of the combined input sum at the classifying output neuron, where x_i is the activation at the hidden layer neuron, i, and w_i is the connection weight to the output neuron. These contributions are shown in Table 2, where it can be seen that the contributions of the 2nd and 5th feature detectors to the combined input sum at the 8th output neuron are more than twice the contribution of the 1st and 4th feature detectors and more than four times the contribution of the 3rd feature detector.

A measure of the significance of the feature detectors to the output classification is the change in the output activation when the corresponding feature detector is deleted. For example, if the 2nd feature detector, H6, in Fig. 5 is deleted, the combined sum at the 8th output neuron drops from +2.411 to +0.511 with a corresponding 31.9% reduction in the classifying output from +0.918 to +0.625. Similarly, removing the four other feature detectors produces the results shown in Table 2. The ranking of the features in order of largest percentage reduction at the output classification is: 2nd feature (31.9%), 5th feature (20.5%), 1st feature (8.8%), 4th feature (8.1%), 3rd feature (4.7%).

7.2. Discovery of the Most Significant Inputs from Within the Input Features

A strategic way to find the most significant inputs is to first rank the significant inputs within each feature in order of feature detector significance and then selectively remove the significant inputs, and combinations of significant inputs, in the same order from the input vector to find the effect on the activation at the classifying output neuron. The most significant inputs within each feature are the ones with the highest positive connection weights to the feature detector since these make the highest contribution to the positive part of the combined sum at the feature detector. For example, within the most significant 2nd feature detector, the 4th and 7th significant inputs in the digit eight image have the highest connection weights.

If a significant input is also present in the other input features, its removal from the input layer also decreases the activations at the other corresponding feature detectors, all of which contribute to a reduction in the output classification. However, if the significant input is not present in a feature, its removal from the input vector can increase the feature detector activation if the connection weight

Table 2. Effect of deleting hidden layer feature detectors on 8th output neuron activation

Feature	Hidden layer feature detector	Activation at feature detector	Connection weight to 8th output neuron	Contribution to weighted sum	Activation drop (from +0.918) at 8th output neuron
feature 1	H5	+0.8722	+0.8881	+0.7746	−0.081 (8.8%)
feature 2	**H6**	**+0.8900**	**+2.1346**	**+1.8998**	**−0.293 (31.9%)**
feature 3	H10	+0.7665	+0.6124	+0.4694	−0.043 (4.7%)
feature 4	H11	+0.8381	+0.8603	+0.7210	−0.074 (8.1%)
feature 5	H13	+0.9824	+1.4403	+1.4149	−0.187 (20.5%)

to the detector is negative, lessening the drop in activation at the classifying output neuron.

8. Summary and Conclusion

The main results of this paper are the discovery of the significant inputs, the data inputs that the MLP network actually uses to make an input/output mapping for a classification task, and the discovery of the most significant, or important, of these data inputs to the network. The analysis leading to this discovery presents a method for interpreting the network outputs and for providing full, or limited, explanation facilities at any stage of network development and use. The knowledge learned by the network from the training data is presented in the form of significant input data relationships and the analysis provides a potential tool for knowledge discovery in databases (KDD), or data mining. During training, the data relationships and explanation facilities can be used for network validation and verification, and after training for rule induction. The analysis is extended to the general case of a MLP that classifies continuously valued inputs to continuously valued outputs in the ranges -1.0 to $+1.0$.

The author concludes that the training input data is a constituent part of the MLP network knowledge since the network acquires knowledge in the training process that is represented by the data relationships discovered from the significant training inputs. Network generalisation is inferred from the network knowledge acquired during training.

9. Further Developments

The analysis presented in this paper can be extended readily to analyse the most significant connection weights in the network, in which case the analysis provides a network development tool for optimal brain damage [16]. By analysing both the negative and the positive part of the combined sum at the output layer neuron, it is possible to interpret any level of activation at an output neuron. This analysis could be used to provide a diagnostic tool in network development, for example, in explaining why an output neuron has not made a particular input/output mapping.

It is the intention of the author to implement the knowledge discovery presented in this paper and further developments for a range of MLP networks developed in the following application areas: medical diagnosis [17]; voice recognition [18]; speech recognition [19]; and multi sensor data fusion [20]. The results of this further research will be presented subsequently.

Acknowledgements. The author wishes to acknowledge the input feature results obtained carefully by Fiona Cotter in her BSc(Hons) CIS Project [10], under the author's guidance.

References

1. Rumelhart DE, Hinton GE, Williams RJ. Learning Internal representations by Error Propagation. In: Rumelhart DE, McClelland JL and PDP Research Group (eds), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations. MIT Press, Cambridge, MA, 1986
2. Werbos P. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD diss. Harvard University, Cambridge, MA, 1974
3. Lisboa PGJ. Neural Networks Current Applications. Chapman & Hall, London, 1992
4. Dayhoff J. Neural Network Architectures. An Introduction. Van Nostrand Reinhold, New York, 1990
5. Spiegelhalter DJ, Taylor CC (eds). Machine Learning, Neural and Statistical Classification. Ellis Horwood, Chichester, 1994
6. Vaughn ML. An Introduction to Neural Computing. International CIS Journal 1989; 3(2).
7. The International Journal of Neural Networks Research & Applications 1989; 1(1) (Learned Information (Europe) Ltd, Oxford, England)
8. Hecht-Nielsen R. Neurocomputing. Addison-Wesley, Wokingham, UK, 1990
9. Wasserman PD. Neural Computing Theory and Practice. Van Nostrand Reinhold, New York, 1989
10. Cotter F. An Investigation into the Properties of Artificial Neural Network Architectures that Achieve Speaker Recognition. BSc(Hons) Command and Control, Communications and Information Systems, Project Report. Cranfield University, RMCS, 1992
11. Best Practice Guidelines for Developing Neural Computing Applications. DTI, London, 1994
12. Lippmann RP. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine 1987; April: 4–22
13. Cassell I. An Investigation into Neural Computing as an Approach to Modelling Life Assurance Underwriting. BSc(Hons) Command and Control, Communications and Information Systems, Project Report, Cranfield University, RMCS, 1991
14. Refenes AN, Azema-Barac M. Neural Networks in Financial Asset Management. Neural Computing & Applications 1994; 2(1)
15. Khabaza T, Shearer C. Data Mining with Clementine. IEE Colloquium, Knowledge Discovery in Databases. London, February 1995
16. le Cun Y. Generalisation and Network Design Strategies. Technical Report CRG-TR-89-4, University of Toronto, Department of Computer Science, 1989

17. Cavill SJ. The Use of Artificial Neural Networks in the Assessment and Diagnosis of Lower Back Pain. MSc Project Report, Cranfield University, RMCS, 1994
18. Vaughn ML, Van Schalkwyk H, King RA. The Application of Artificial Neural Networks to Provide Safe Access to Driver Information Systems and Other Non-critical Automotive Functions. 8th Int Conf "Automotive Electronics" IEE, London, 1991
19. Campbell MI. Speaker Independent Phoneme Recognition from the TIMIT Database using TESPAR. A comparison of the MLP and LVQ Artificial Neural networks as Classifiers. MSc Project Report, Cranfield University, RMCS, 1994
20. Lison AC. Using Neural Networks for the Prediction of Helicopter Airframe Load Spectra from Flight Parameter Data. MSc Project Report, Cranfield University, RMCS, 1994