A Novel Approach to Training Neurons with Biological Plausibility

Bernadette Garner
CSSE
Monash University
Clayton Vic. 3168
Australia
Bmg@csse.monash.edu.au

The dominant approach to training artificial neurons is to search iteratively for single numeric values that satisfy the training conditions based on error. A novel method of fully training neurons that will generalize has been developed that does not find single numeric values for weights, but instead finds regions in the weight-space that satisfy the training conditions. This method provides an alternative to the biologically implausible backpropagation for training feedforward neural networks, as it allows the McCulloch-Pitt neuron model to be used as a basis for a fast training algorithm that dynamically allocates neurons into the network.

Keywords: biological neuron training, constraints, SOM.

1. Introduction

Experiments have shown that neurons use the step function as the activation function [1]. In other words, neurons are either activated or not. Despite recent speculation that the biological neuron is not a step function, no experimentation has demonstrated that this is the case [3,12].

When Rosenblatt [8] started training artificial neurons, he trained the neuron's weight values as single numeric values that attempt to satisfy the training conditions by iteratively modifying the weight values to eliminate error between the desired and the actual output of the neuron. This has remained the dominant method of training neurons.

Because there was no method of training neurons to learn more complex data sets than those that are linearly separable, the perceptron had to be modified to use the sigmoid function, hence data values that are close to boundary conditions may not accurately cause the network to fire as required.

This method of training neurons learns by finding relationships between the neuron's incoming connection weights and the neuron's threshold to produce the required output for the input the neuron is to classify. The algorithm learns by modifying the connection weights by ranges of response, which can be likened to quantization.

The algorithm being discussed in this paper uses the step function, and adds neurons into the network as required thus building its own architecture. When a single neuron cannot learn a particular input, one or more additional neurons are added to the ANN to enable the network to learn the input. This is similar to the biological brain, which can add connections between neurons in a number of ways [10]. Also this training algorithm learns in a single pass.

In this paper, section 2 defines the neuron to be used and describes how neurons are trained with the method proposed in this paper. Section 3 demonstrates that the neuron is fully trained, in other words that the data the neuron learnt can be recovered and generalize; and section 5 shows that the neuron can generalize. Finally, the conclusions are in section 6.

2. New Method of Training Neurons

This neuron being used by this algorithm is defined as the McCulloch-Pitt neuron [1], where **x** is the input vector i, n the number of inputs into the neuron, and **w** is the weight vector associated with the incoming weights of the neuron. T is the threshold of the neuron and O is the neuron's output.

$net_n = \mathbf{x_i} \cdot \mathbf{w}$

where $\mathbf{x} \in \{0, 1\}^n$ and $\mathbf{w} \in \mathbf{R}^n$.

$$O = \begin{cases} 1, & if \quad net_n \geq T \\ 0, & otherwise \end{cases} \tag{1}$$

There are a number of input connections into a neuron and each weight connection can be considered a dimension in the weight-space.

It has been established, through the use of sensitivity analyses, that there are many values the weights can take for each input connection into the neuron that solve the learning problem that any neural network may learn [8]. Traditional neuron training methods only find one of these weight values.

So then the region in the weight space that will allow the neuron to fire can be considered a volume. This is called the activation volume.

Relationships can be constructed based on equation (1) and the input vector as constraints. The constraints define relationships between the weights associated with each neuron's input connection weights and the neuron's threshold. If the neuron is to be 0 then the weighted summed input is < than the neurons threshold, then the neuron will not fire, and if the output is to be 1 the weighted sum will be ≥ T and the neuron fires. This summarized in the following relationships.

$\mathbf{x_i} \cdot \mathbf{w} \geq T \rightarrow 1$

or

$\mathbf{x_i} \cdot \mathbf{w} < T \rightarrow 0$

where i is the pattern that the neuron is currently learning.

What the neuron can learn is based on whether adding the constraint satisfies all the previously learnt constraints. Each time a constraint is applied to the weight space of a neuron the volume that causes the neuron to be activate is reduced. Hence, if a neuron learns $\mathbf{x_i}.\mathbf{w} < T$, it eliminates the possibility the neuron activates when $\mathbf{x_i}.\mathbf{w} \geq T$ and vice versa.

In an example of a 2 input neuron, if the input vector is [1 1] and this causes the neuron to be activated, then the constraint produced is $w_1 + w_2 \geq T$. If the neuron is to learn logical AND, then the constraints produced are $\{w_1 + w_2 \geq T, 0 < T\}$.

A diagram of the weight space of this is shown in Figure i of what the neuron has learnt.

The neuron may then learn to classify more patterns. If the neuron has to learn [1 0] produces an output of 0, then the weight space in Figure ii is produced.

The neuron can still learn that either the input vector [0 1] causes the neuron to be activated or not. If it fires then the weight-space will remain unchanged but if it learns that the neuron doesn't fire then the weight space will be reduced to $\{W_1 + W_2 \geq T, 0 < T, W_1 < T, W_2 < T\}$. In this case the neuron has learnt to only be activated if both input 1 AND input 2 are present. If the neuron learns input vector [0 1] causes the neuron to be activated, then the neuron has learnt that only input 2 is required to activate it.

This algorithm learns data in a single pass, as it does not use iteration. The algorithm forms prepositional logic functions between the neuron's input connections, which is comparable to biological neurons which exhibit the property of forming different types of inhibitory and excitatory neurons [5,6,10,14].

3. Full Training and Generalization

Using this training method for neurons, it is possible to recover what the neurons have learnt. For instance if a 2-input neuron has learnt the following constraints $\{w_1 + w_2 < T, 0 < T, w_1 \geq T\}$, and it receives input vector [1 1], using equation (1), the output can be deduced $1.w_1 + 1.w_2 = w_1 + w_2 < T$ therefore the neuron will output 1. Hence from this example, it is seen that the network is fully trained and the network can output what it has been previously taught.

Generalization is one of the most interesting properties of neurons as it allows neurons to output responses to data is has not been trained with. Beale et al [1] define generalization as the property that allows neurons to handle noisy input.

If a neuron is trained with input vectors with associated output

$[1\ 1] \rightarrow 0$

$[0\ 0] \rightarrow 0$

$[1\ 0] \rightarrow 1$

Then the following constraint will be produced $\{w_1 + w_2 < T, 0 < T, w_1 \geq T\}$. During training the input vector [0 1] was not seen, but it is meant to output 0. For proper generalization, we need to be able to determine what the output of the neuron will be when [0 1] is applied to the input. When the input [0 1] is applied to the neuron, it is found we want to find what relationship $0.w_1 + 1.w_2 = w_2$ has to the threshold T.

It can be deduced from the constraints that since $0 < T$ and $w_1 \geq T$ that $0 < T \leq w_1$ however $w_1 + w_2 < T$, therefore $w_2 < T - w_1$, so $w_2 < T$, which causes the neuron not to be activated.

Hence the neuron is fully trained and can generalized when trained with this method.

4. Training a Network of Neurons

Whether a neuron can learn a data pattern forms a natural criterion for adding a neuron to the network. Hence the algorithm will dynamically allocate neurons into the network only as required to learn features in the data set [7].

Because the network can learn input vectors immediately, the network can learn in a single pass. However, since it is a single pass training algorithm it requires input vectors to be sorted for it, as the rules learnt by the network depends on the order that data is presented.

The brain exhibits such a sorting in the form of brain map features on the surface of the cortex [11]. These features can be sorted artificially using SOMs [1].

Using this method of training neurons does not experience the local minima problem that artificial neural networks (ANN) exhibit since it does not feed error back or rely on gradient descent. It also produces relationships between the weights and thresholds that can be easily interpreted and adds neurons as required.

This training algorithm uses binary data, as the biological brain does. For ANN floating point data is converted to binary. The training algorithm has been implemented in prolog and provided exciting results for feedforward problems considered difficult for networks trained with backpropagation. Data sets used include the twin spiral data set [4], data sets from the UCI Repository of Machine learning databases [2] and a number of binary data set problems.

5. Conclusions

This method of training neurons explains how networks of binary data can be learnt by the simple McCulloch-Pitt neuron model and leads to a simple method of training networks. The method entails cutting planes through the neuron’s weight space and then removing the regions that from the volume which cause the neuron to be activated. To learn the input vectors it converts them to constraints which allows this method to be implemented on computers, but it is assumed that planes are constructed by modifying the relationships between the neuron's weights in biological weights. This is beyond the scope of the paper. This algorithm could possibly explain why brain maps exist on the surface of the cortex.

6. Acknowledgement

Thank you to Prof. Bala Srinivasan, Monash University, for many useful comments on this text.

7. References

[1] R. Beale and T. Jackson, Neural Computing, an introduction, (IOP, 1992).

[2] C.L. Blake, and C.J. Merz, (1998). *UCI Repository of machine learning databases*, http://www.ics.uci.edu/~mlearn/MLRepository.html Irvine, CA: University of California, Department of Information and Computer Science.

[3] Carandini and Ferster, Membrane Potential and Firing Rate in Cat Vi, J. Neurosci, 20(1) (2000) 470-484.

[4] S. Fahlman and C. Lebiere, The Cascade-Correlation Architecture, Technical Report CMU-CS-90-100, Carnegie Mellon University, 1991.

[5] M.M. Francis, K.I Choi, B.A Horenstein, and R.L. Papke Sensitivity to Voltage-Independent Inhibition Determined by Pore-Lining Region of the Acetylcholine Receptor, Biophys J, Vol. 74(5) (1998) 2306-2317.

[6] J.A. Freeman, D.M. Skapura, *Neural Networks algorithms, applications, and programming techniques*. (Clear Lake, Addison-Wesley, 1992).

[7] B.M. Garner, A New Training Algorithm for Feedforward Neural Networks, in: accepted IC-AI'02 (2002).

[8] J. Hertz, A. Krough, R.G. Palmer, Introduction to the theory of neural computation (Redwood City, Addison-Wesley, 1991).

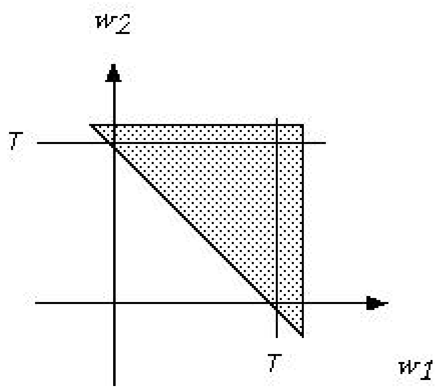
[9] C.C. Klimasauskas, Neural Nets Tell Why, Dr. Dobb's Journal, April (1991).

[10] K. McDonald, Pictures reveal how Nerve Cells form Connections to Store Short- and Long-Term Memories in Brain, (http://ucsdnews.ucsd.edu/newsrel/science/mccell.htm, 2001)

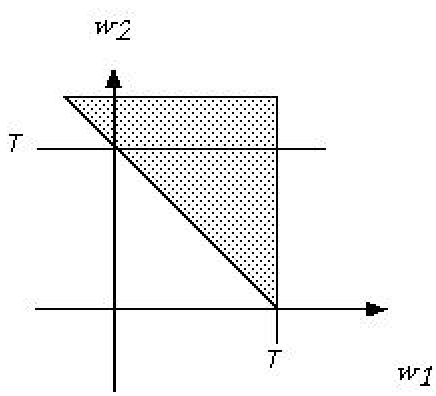
[11] K. Obermayer, T. Sejnowski, G.G. Blasdel, Neural pattern formation via a competitive Hebbian mechanism, Behavioural Brain Research 66 (1995) 161-167.

[12] B. Perks, Synaptic "kiss-and-run" could explain speed of neurotransmission (http://news.bmn.com/news/story?day=011012&story=1, 2001)

[14] J. Szentagothai, On General Theories of Brain Organization and Brain Function, Naturwissenschaften 72(6) (1985) 303-309.



(i)



(ii)

Figure i. The weight-space for a 2-input neuron that has learnt $\{W_1 + W_2 \geq T, 0 < T\}$

Figure ii. The weight-space for a 2-input neuron that has learnt $\{W_1 + W_2 \geq T, 0 < T, W_1 < T\}$

Bernadette Garner is a Ph.D. candidate at Monash University, Australia, in the field of neural networks. She worked in the telecommunications industry and ran a small research group in a company she worked for.