Rahul Thapa

Summary:

In this paper, the authors propose back-propagation procedure in neural networks which minimizes the total error by adjusting the weights of the connections in the network. Before back propagation, the neural networks such as perceptron does not have a true hidden layer because their states are completely determined by the input. Therefore, they will not be able to learn hidden important features of the task at hand. In contrast, the procedure that uses back-propagation has true hidden units which can capture some useful features because of synaptic modification. The state of these hidden units is not specified by the task.

In the paper, the authors mention the use of a specific nonlinear activation function (sigmoid) in order to obtain the real-valued output of a hidden unit using inputs from the previous layer. However, the author also clarifies that any non-linear activation function can be used if it has a bounded derivative. This way, the output in the final layer can be calculated by the sequence of feed forward method of applying activation function to the input of preceding layer after multiplying by the weight of the edge or connection. The total error then can be calculated by averaging the square of the difference between the exact and approximate value of all the output units. The goal is to minimize this error and one method that paper mentions for this is to use gradient descent. This requires calculating partial derivative of error with respect to each weight in the network. There are two passes through the network to achieve this. First is the forward pass as described above and the second is backward pass or back-propagation. The author describes that the backward pass starts by computing the partial derivative of the error with respect to the actual state for each output unit. The main goal is to find the derivative of cost function with respect to the weights. Therefore, the author uses chain rule to calculate this partial derivative as shown in equation (4), (5), and (6). Basically, the idea is to use the partial derivative of error with respect to output for all unit in the last layer to find the partial derivative of error with respect to output in the penultimate layer. This process can then be repeated to backwards calculating the partial derivate of error with respect to weight along the way.

The paper mentions that during back propagation, the weights can be changed along the way, after every input-output case. Updating weights in this way saves memory. However, the authors followed a different approach. They collected the partial derivate over all the input-output cases and finally changed the weight at the end of that process. The authors mention a simple and easy approach of updating weight proportional to the partial derivative which can further be improved by using acceleration method given in equation (9).

Strengths:

Despite being a very old paper and not having as many background works to reference and build upon, the paper does a good job of explaining the mathematical equations used in the back-propagation. The paper gives background of traditional neural network and the feed

forward mechanism first instead of directly jumping into the jargon of back-propagation, which makes it easier for reader to follow and understand the concepts well.

Weaknesses:

One main weakness of the paper is that the graphics are not very clear. I understand that it is because the paper is very old, and they may not have as good of a graphics tool back then. Also, the paper can be divided into sections to make it easier for reader to read and follow.

Confusions:

I was confused about their actual experiment. The graphics was not very clear, and I found the description a little bit vague. In the figure 1, the paper talks about the network which learned to detect symmetry in input vector, and I could not completely follow their methodology.

Discussion Questions:

1. How can a synchronous iterative net be equivalent to layered net?
2. How does the acceleration method describe in equation 9 of the paper work? Visually, how will you represent it?
3. How does backpropagation help in detection of symmetry, as described in figure 1 of the paper?