**Lab 3 – Neural Network – MLP - ANSWERS**

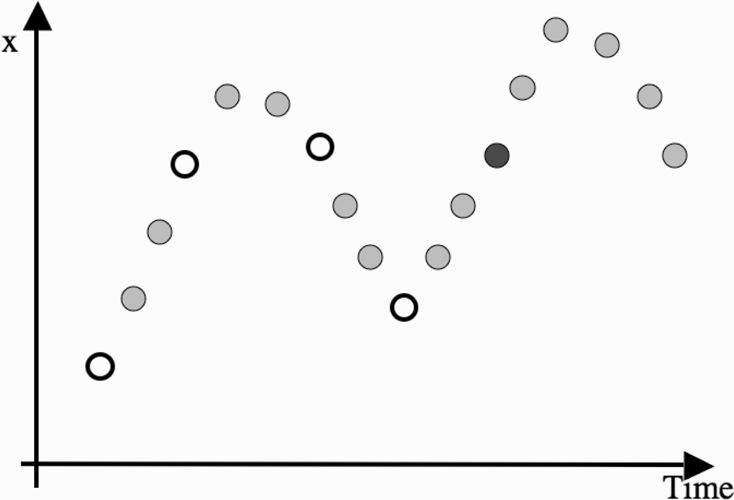
1. Study Section 4.4.4. Download the dataset PNOz.dat and run the MLP. See whether you can get something similar to Fig. 4.16.

**Ans.** The key is to understand how to create the data. The general equation for time series prediction is



Use the fact that starting at the current time, we need to go back k steps with unit time of

Tau (see figure below): To Be explained in the class.



Now, you can generate all the data needed using this method.

1. Use the knowledge you gained from #1 to solve the following problem:

Suppose that the local power company wants to predict electricity demand for the next 5 days. They have the data about daily demand for the last 5 years. Typically, the demand will be a number between 80 and 400.

1. Describe how you could use an MLP to make the prediction. What parameters would you have to choose, and what do you think would be sensible values for them?

**Ans.** The prediction can be made by creating a training, validation and testing set using the approach used in problem #1 above; and the training an MLP. We would need to select k and tau. Typical values would be k = 4, tau = 2.

1. If the weather forecast for the next day, being the estimated temperatures for daytime and nighttime, was available, how would you add that into your system?

**Ans.** The weather information can be added as a new input to the neural networks. Also, we would need to create data relatingthe estimated temperatures for daytime and nighttime with the power used, if such data were not in the power usage data.

1. Do you think that this system would work well for predicting power consumption? Are there demands that it would not be able to predict?

**Ans.** It will work to a reasonable extent. Yes, there are demands it would not be able to predict – e.g. upward or downward trends within summers or within winters. We would, however, know that average usage in summer or winter would be more than other seasons.

1. Modify the code to allow another hidden layer to be used. You will have to work out the gradient as well in order to compute the weight updates for the extra layer of weights. Test this new network on the Pima Indian dataset that was described in Section 3.4.4.

Ans:

The equations of Back Propagation algorithm remains consistent when we add more hidden layers. This means that the back propagated error for the first hidden layer (from the output) calculated using the error (i.e. deltas) can be used in the same way for the 2nd hidden layer (from the output). We just need to use different notation for the 2nd hidden layer and also use the input for the 2nd hidden layer etc.

Thus, for the backward Phase, we have:

Delta at the output,



Back propagated delta at the 1st hidden layer,





The weight update equations also remain similar,

For weights between 1st hidden layer and output layer,





1. A recurrent network has some of its outputs connected to its own inputs, so that the outputs at time *t* are fed back into the network at time *t* + 1. This can be a different way to deal with time-series data. Modify the MLP code so that it acts as a recurrent network, and test it out on the Palmerston North ozone data on the book website.

Ans. In this case, only the output node’s equation will change as we need to add the previous value of each output node to itself as shown below. There will be no change to back propagation or weight updates equations; however since yk values will change, it will affect delta and weight values but the equations will not change except for the yk equation (note that the outputs of the previous value is fed back with unity weights i.e. we are not assuming any weight update for the recurrent path).