**8**.

We always prefer less complex network, which has a smaller architecture. Thus, we can use the information to re-train the network with a smaller architecture that matches the Effective Number of Weights found. In addition, we can use this information check if a larger network leads to the same Effective Number of Weights so that we can check whether larger network will lead to better result.

**9**.

The temperature T plays a crucial role in controlling the evolution of the state of the system with regard to its sensitivity to the variations of the penalty function. As T gets smaller, the probability of the transition to new state becomes smaller. If T decreases to certain level, the probability of transition is negligible so that the system becomes stable, meaning we reach the state of convergence.

**10**.

To determine the sensitivity of each of the 13 attributes. We can use the Backpropagation to calculate the derivative of sum squared error with respect to each of the 13 inputs. If we find that some of these derivatives are much smaller than the maximum derivative, then we can consider removing those inputs. After removing the potentially irrelevant inputs, we retrain the network and compare the performance with the original network. If the performance is similar, then we accept the simplified network.

We can also try to look for collinearities in the 13 attributes since it will help us to reduce the number of input and reduce the complexity of the network. To perform the testing of collinearities, we can calculate the Pearson correlation coefficient between each of the inputs and see if there exist collinearities. When variables are highly correlated with each other, they can cause problems such as over fitting because they are explaining almost the same variability in the outcome. If we found collinearities among the 13 attributes, we will remove input with smaller sensitivity to simply the network.

**11**.

By using two outputs (0,1) and (1,0), respectively, to represent classes A and B, than a single output (0) and (1), we are in effect introducing extra free parameter in our network. In general, using more free parameters in a model will make data fitting faster since it is always easier to use more parameters in regression. Thus, using the two-outputs representation will speed up the learning convergence. However, we will also be more likely to run into the issue of over fitting.

It hold true for using a thermometer scale to represent ordinal variables than a single output.

**12**.

It is possible to change the order of preference of the classifiers by introducing different costs for Type I and Type II errors. In the graph, C has the highest True positive rate (TPR). If we assign very high cost for Type II error than cost of Type I error, C might be always preferred over A and B regardless the False positive rate (FPR). On the other hand, since B has both lower TPR and higher FPR then A, B can never be preferred over A. But since B has smaller FPR then C, B can be preferred over C if the cost Type I error is much higher than Type I error.