Emperical Analysis of General utility problem in Machine Learning

Lawrence B. Holder
holder@cse.uta.edu

Department of Computer Science Engineering University of Texas at Arlington
Box 19015, Arlington, TX 76019-0015

1 Technical Updates

1.) The paper describes the optimal classification error which tells us about the error incurred during training of them model and therefore lets us decide what would be the best time to stop the training and therefore achieve the optimal result for accuracy while achieving a low error rate.We, see that the error curve also followed the curve of general utility problem that is achievement of peak followed by gradual decline in performance as the training progressed.

Now, when considering the case of training of a neural network, it can be noisy and therefore when performance is plotted on the graph,the performance of a model on a validation dataset may go up and down many times .It may happen that the validation error still goes further down after it has begun to increase and hence if we would have had stopped at a earlier point, then the global optimum of the validation error would not be reached and hence we cannot get the actual peak of least error of the model and therefore the best performance of the model cant be obtained.

It is seen that real validation error curves almost always have more than one local minimum and therefore the following method can be can be used to find the global optimum of the model:-

- In this method, We need to use a variable to store the parameter's that we getting at some point .Every time the error on the validation set is better than what the current parameter in the variable attain,we store a copy of the current model parameters inplace of the old parameters in the variable .We then continue moving forward and keep on training the model simultaneously the process of updating the variable with the best parameter continues.

- When the training algorithm terminates, we return the parameters stored in the variable, rather than the latest parameters that we are currently on or the initial parameters.In this way we are able to get the best parameters of the model giving the best performance.

Pros of the above change:-

- This method ensures that we don't stop at the first signs of overfitting but move on to achieve the global maximum of performance before overfitting starts.

- The above method is not only applicable to neural network but most of the models and neither is this method time consuming nor does it need high computing power.
- Generally its observed that some delay in stopping is almost always a good idea and results indicate that a slower criteria for stopping i.e which stop later than others, on the average lead to improved generalization compared to faster ones.
- Also,it is seen that such a process not only applies to some of the models but to a vast range of models and domains.Therefore, a generalization of such a process is feasible both theoretically and practically.
- when implementing the above process , it sure that we are going to get the best parameters from across a range of data that is being learnt and not only a part of it.Now, it is also possible to optimize the above process to consider only a part of the model performance and pick the best parameters in it.

2.) Another important method is to also including a validation set while training the model as the best performance will be achieved when the model has least error on the validation set.If we are only taking the training set and finding the least error parameters of the training data error graph.Then,in most cases it is not the parameter that gives the best performance as overfitting in the data may have started well beyond that point.

From such a set of data where there is also validation set along with training set , by plotting a graph between the two, we can get an insight into the different techniques we can use to improve performance i.e. to find the optimum loss or error. A couple of them are:-

- If accuracy of the training set is much better than the validation set, you are probably overfitting and we can use techniques like regularization. and therefore the
- If the accuracy of the training and validation are both low, you are probably underfitting and we can probably increase the capacity of your network and train more or longer.
- If there is an inflection point when training goes above the validation, we might be able to use early stopping in that part of training. which infact is the process described in the point (1) above.

Pros of the above change:-

- One of the most prominent benefits in the having a validation set being that we can improve the accuracy or performance and also keep it under control in some circumstances as in some cases in the paper, it was seen that the accuracy was even overstepping the maximum accuracy/performance of the model.

3.)The expression for the classification error $R(L)$ in terms of the bias $B(L)$ and variance $V(L)$ is given where R* is the Bayes optimal classification error is given as:

$$R(L) = B(L) + V(L) + R^* \tag{1}$$

but what sample we have may not always be clean and hence value of the output will be different form the original one .

- Therefore the value of the classification error will be different from what has been shown in the paper so at that instant the value of ***Total error=Bayes error + how much your model is worse than bayes error*** rather than Bias + Variance +Bayes error and this may depend on your model and the inherent nature of the noise that it has.

Pros of the above change:-

- One of the most important benefits of this is that the above method/formulae does not idealise one equation for cases where there is high noise with one where there is low noise and hence, we are able to find better solution to the error and hence better performance.