Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping

Satyam Kumar (B19BB060)

Jan, 2021

Abstract

Earlier it is believed that backprop nets with excess hidden units generalize poorly. So, in this paper it is shown that excel capacity also generalize well if it is trained with backprop and early stopping. While training on real examples it become clear that: -

1.) Overfitting vary significantly in different regions of the model. Excess capacity allows better fit to the regions of high non-linearity, and backprop often avoids overfitting of the region of low non-linearity.
2.) Big nets learn task in subcomponents and Early stopping can stop training the large net when it generalizes comparably to a smaller net.

In this paper it is also shown that conjugate gradient can yield worse generalization because it overfits regions of low non-linearity when learning to fit regions of high non-linearity.

1.) Introduction

Earlier, we think Restricting net capacity prevents overfitting because the net has insufficient capacity to learn models that are too complex. But once training of large networks becomes feasible, it is noted that overfitting does not occur with excess capacity. Even it is observed that Large nets generalize well.

In this paper it is also shown that Multi-layer-perceptrons with excess capacity often do not overfit. On the contrary, we observe that large nets often generalize better than small nets of sufficient capacity. **Backprop** appears to use excess capacity to better fit regions of high non-linearity, while still fitting regions of low non-linearity smooth.

Also, if **Early stopping** is used, training of the large net can be halted when the large net's model is similar to models learned by smaller nets.

2.) The Problem

In this paper, the problem we are going to discuss is Overfitting. How we can control our model from being overfitted. Different techniques we are going to use in this paper to overcome overfitting such as **Backpropagation**, **Early Stopping** and **Conjugate Gradient.** We also get rid of certain believes like if we use excess hidden units our model will overfit even after using backpropagation, overfitting is a global phenomenon, large model generalizes poorly.

3.) Overview

I started of reviewing the research paper in order to understand the objectives and focus. The main objective of this paper is trying to train some real-life data and find relation between overfitting and net size, problems used in this case are: - NETtalk, 7 and 12 bit parity, an inverse kinematic model for a robot arm, vision data used to learn to steer an autonomous car, etc. It shows that excess capacity does not hurt (means even if we use large MLPs using backprop our model does not overfit) through these problems. Along with that, earlier it was believed that overfitting is a global phenomenon but, in this paper, it is shown experimentally that overfitting may occurs locally i.e., overfitting is different in different region.

4.) Pros

So basically, this paper is based on experiment results. Data is train on multiple real-life problems. Along with this, different approaches are used to train models, so that we can conclude finally that how overfitting depends upon other factors, like size of hidden layers, hidden units, backprop, early stopping, conjugate gradient. In this paper it is analysed that what nets of different size learn while they are trained. We compared input/output behaviour of nets at different stages of learning on large samples of test patterns.

If two nets make the same predictions for all test cases, they have learned the same model (even though each model is represented differently), and the **squared error** between the two models is zero. If two nets make different predictions for test cases, they have learned different models, and the squared error between them is large. This is not the error the models make predicting the true labels, but the difference between predictions made by two different models. Two models can have poor generalization (large error on true labels), but have near zero error compared to each other if they are similar models. But two models with good generalization (low error on true labels) must have low error compared to each other.

4.) Cons

In this paper they are using only limited problems, so it may also possible that their result may be biased also. It may possible that their accuracy is only on training data and not on testing data (as it may generally happens in case of supervised learning). There is no categorisation of overfitting as we know it is of different types like: -

- Noise learning on the training set: when the training set is too small in size, or has fewer representative data or too many noises. This situation makes the noises have great chances to be learned, and later act as a basis of predictions. So, a well-functioning algorithm should be able to distinguish representative data from noises

- Hypothesis complexity: the trade-off in complexity, a key concept in statistic and machining learning, is a compromise between Variance and Bias. It refers to a balance between accuracy and consistency. When the algorithms have too many hypotheses (too many inputs), the model becomes more accurate on average with lower consistency.

5.) Suggestions

As discussed in this paper about Early stopping, as this strategy is used to avoid the phenomenon “learning speed slow-down”. This issue means that the accuracy of algorithms stops improving after some point, or even getting worse because of noise-learning.

As seen in Figure 1, where the horizontal axis is epoch, and the vertical axis is error, the blue line shows the training error and the red line shows the validation error.

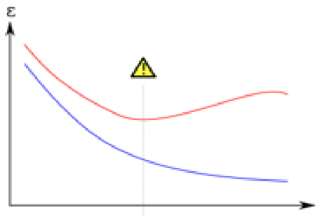

Figure 1. Validation error vs testing error

If the model continues learning after the point, the validation error will increase while the training error will continue decreasing. So, we need to stop at the exact point to stop training.

To find out the point to stop learning, the obvious way is to keep track of accuracy on the test data as our network trains. In another word, we compute the accuracy at the end of each epoch and stop training when the accuracy on test data stops improving.

6.) References

[1] https://elitedatascience.com/overfitting-in-machine-learning

[2] Paris G., Robilliard D., Fonlupt C. (2004) Exploring Overfitting in Genetic Programming. Artificial Evolution, International Conference, Evolution Artificielle, Ea 2003, Marseilles, France, October. DBLP, pp.267-277.

[3] C. Darken and J.E. Moody. Note on learning rate schedules for stochastic optimization. In *Advances in Neural Information Processing Systems,* volume 3, pages 832- 838. Morgan Kaufmann, 1991.