Adaptive Learning : automatically stop training earlier layers as training progresses

P.Deepika
Department of Computer Science
IIT Hyderabad
cs14btech11027@iith.ac.in

S.Kumuda Priya
Department of Computer Science
IIT Hyderabad
cs14btech11029@iith.ac.in

K.Reshma
Department of Computer Science
IIT Hyderabad
cs14btech11042@iith.ac.in

Abstract

We came up with a method to estimate a **threshold** , such that if the norm of the gradients coming into a layer is less than the threshold , then we stop training that layer or backpropagating any further.

1 Vanishing Gradient Problem

The gradient tends to get smaller as we move backward through the hidden layers.The phenomenon is known as the vanishing gradient problem.

1.1 Speed of Learning for hidden layers

To visualize the learning speed of each layer , we compute the $\delta \mathrm{n}_{ij}$ = δC/$\delta \mathrm{w}_{ij}$, for each neuron in that layer and calculate the norm || δn || of all neurons in that layer.

1.2 Visualize the learning speed of each layer

From the graph Figure 1 , we can observe that speed of learning for the final layer is much faster than earlier hidden layers.

1.3 Cause for Vanishing Gradient

As seen in the figure 2 , the gradient from shallower layers are computed by multiplying smaller and smaller values there by causing vanishing gradients.

2 Automatically stop training earlier layers as the training progresses.

For a set of fixed parameter of network , a constant threshold is maintained. If the norm of gradients from a layer < threshold , then the gradients are set to zero and training is continued.

2.1 How to set the threshold

From observations , we found that threshold is dependent on the weight scale , no of hidden layers , training Iterations.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

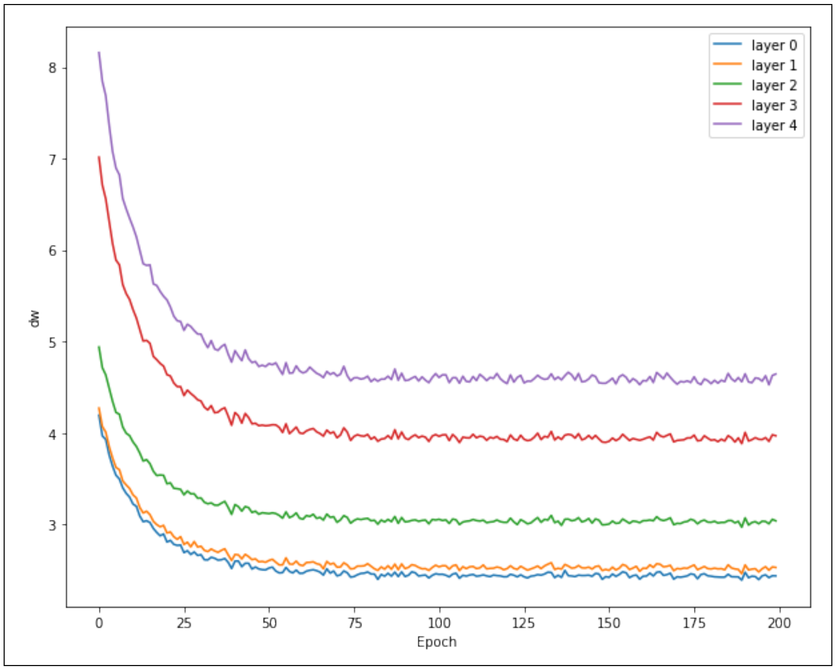
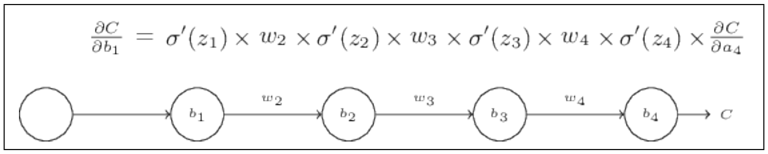


Figure 1: δC/δw Vs Epochs



$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

Figure 2: Cause for Vanishing Gradient

2.2 Weight Scale

If the weight scale is very high, the network faces exploding gradient problem. If the weight scale is very low, the network faces vanishing gradient problem.

2.3 No of hidden layers

We observed that with increase in hidden layers, more number of times norm of gradients of a layer is going below threshold.

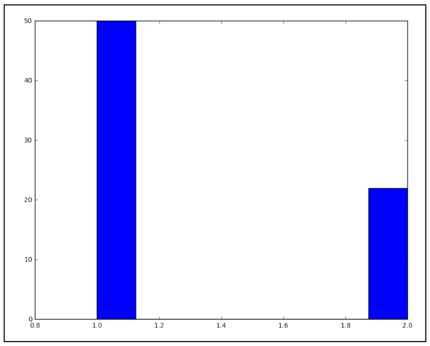


Figure 3: No of times norm went below threshold for each layer with no of hidden layers = 5

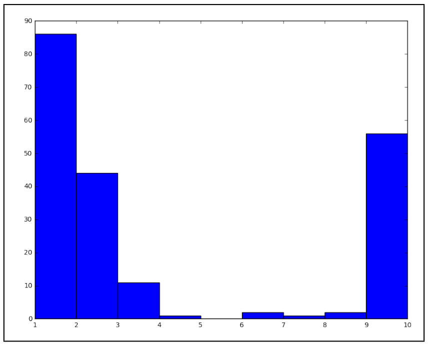


Figure 4: No of times norm went below threshold for each layer with no of hidden layers = 10

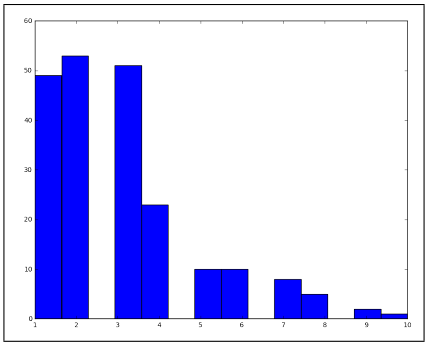


Figure 5: No of times norm went below threshold for each layer with no of hidden layers = 15

2.4 Training Iterations

Next important factor is training iterations. When training iterations are increased, the model is prone to overfitting and also the gradients of shallow layers decrease enormously.From Figure 6 , we can observe this.

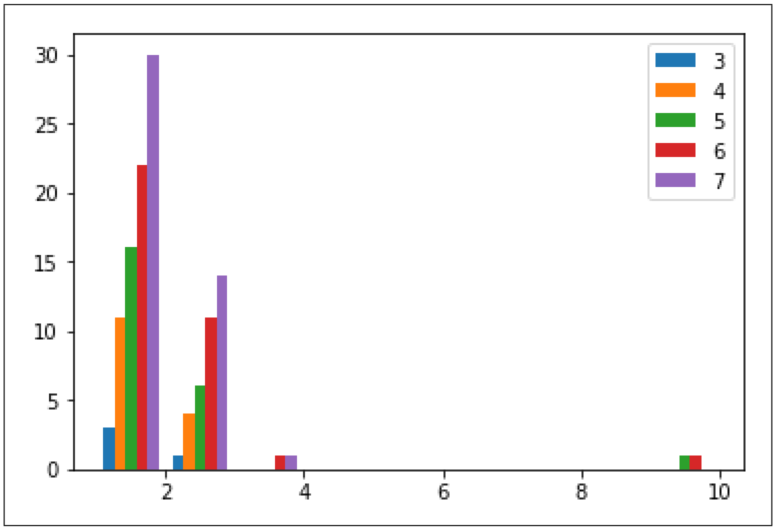


Figure 6: No of times norm went below threshold for each layer in each epoch for 10 hidden layers

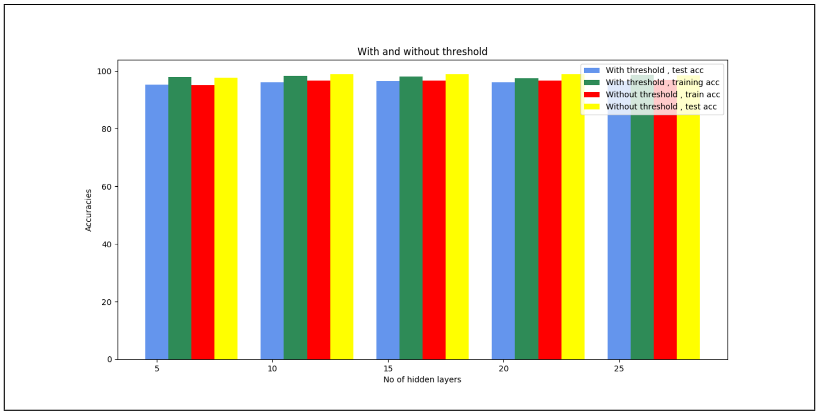


Figure 7: Accuracies Vs no of hidden layers with and without threshold.

2.5 Threshold vs training and test accuracies

The test accuracy of models with and without threshold are same but the train accuracy of models without threshold is high(overfitting).From Figure 7, we can observe this.

3 Adapting threshold based on number of hidden layers and train iterations.

For the first half training iterations, the threshold is zero. During this time the least norm of gradients value for each layer is stored. For next half training iterations, threshold is specific to a layer:

$$\text{Threshold} = \text{min\_grad} * (\text{nHiddenLayers} - i) / (\text{nHiddenLayers})$$

This update equation helps us in maintaining lower threshold values at deeper layers and higher threshold at shallower layers.

So that deeper layers will go lesser number of times below the threshold than shallower layers.

4 Computational Efficiency

For 12 layer network
percentage of computations decreased due to constant threshold : 26.85%
percentage of computations decreased due to varying threshold : 15.64%

test accuracy with constant threshold : 96.1%
test accuracy with varying threshold : 99%
test accuracy without threshold : 98.8%

train accuracy with constant threshold : 96.3%
train accuracy with varying threshold : 95.6%
train accuracy without threshold : 97.63%

By maintaining a constant threshold, at the cost of 1-2 % loss in test accuracy we can obtain around 25% decrease in computations.

By maintaining varying threshold, with around 15% decrease in computations we can even get 0-1 % increase in test accuracy.

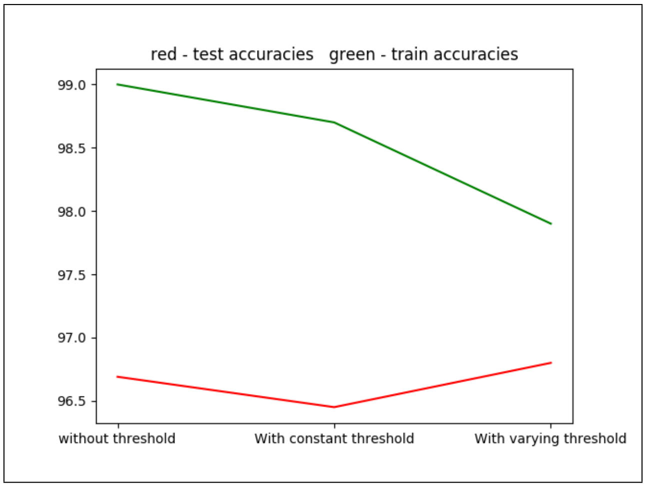


Figure 8: Accuracies Vs no of hidden layers with and without threshold.

5 Final Comparison

As we can see from Figure 8 training accuracy is decreasing and test accuracy is increasing in case of varying threshold in comparison with without threshold . Our varying threshold model gives better generalisation to the input data.

References

`http://neuralnetworksanddeeplearning.com/chap5.html`