Training Neural Networks Using Variable L2 Regularization

David Handy, Scott Jeffery, Benjamin Lampe

Table of Contents

Table of Figures

1. Abstract

In order to more quickly arrive at an accurate but generic solution, neural networks can use a variable level of regularization to initially create a generic solution by using a large regularization constant, and then reducing the constant to a very small value near the end of the training period to allow the network to custom fit the training data better, once a generic function has been learned.

Networks with decaying values that start at some maximum regularization value and end at some minimum achieve better accuracy on inputs outside the training data than networks that statically remain either at the same maximum or minimum regularization value. They do not saturate as quickly, allowing for the training period to be more effective.

2. Introduction

Given enough epochs, most neural networks can approach 100% accuracy on the data used to train them. However, whether this translates to an accurate network depends on how well that training generalizes to inputs not present in the training data. Networks with a high degree of accuracy on training data, but a low degree of accuracy on other data are said to have over-fitted to the training data.

Regularization seeks to avoid over-fitting by constraining the network into functions that tend to generalize better, essentially creating smoother functions. However, it is difficult to know how much to regularize. Overly large regularization values create very smooth output functions, but they also minimize the effect of the training data on the final network. This means that the average error of a network could still be very high, even after the network has trained until saturation. Overly small regularization values do not sufficiently prevent the network from over-fitting to its training data.

This paper explores the idea of using variable regularization values, where the value is high in the initial epochs to avoid premature saturation and over-fitting, but decays to a very low value by the end of the training period, allowing the training data to customize the generic function that the network learned at the beginning of the training period.

The intent of using such variable regularization values is to increase the overall accuracy and generalization of the network while decreasing the length of the training period needed to achieve such results.

3. Methods

All tests were performed on 3-layer networks with 784 neurons in the input layer, 100 neurons in the hidden layer and 10 neurons in the output layer. The input neurons mapped to the 784 pixels in a 28x28 pixel image, while the ten output neurons mapped to the digits 0-9. The desired behavior was that given an input of a picture of a hand-written digit, the output neuron corresponding to that digit would have an activation of 1, while the other nine output neurons had an activation of 0.

Training was performed against the MNIST handwriting data set. The network trained on 50,000 images, and after each epoch, it evaluated its accuracy against an additional 10,000 images. The

network trained for 400 epochs, using the Quadratic cost function.

The network used L2 regularization as part of its cost function to avoid over-fitting. Four profiles were used for the value of λ:

- A static regularization of $\lambda_{max}=5.0$
- A static regularization of $\lambda_{min}=0.001$
- A linearly decaying regularization of $\lambda_e=\lambda_{e-1}-\frac{\lambda_{max}-\lambda_{min}}{E}$ where λ_e is the regularization constant at epoch e, E is the total number of epochs, and $\lambda_1=\lambda_{max}$
- An exponentially decaying regularization of $\lambda_e=\lambda_{min}+\frac{\lambda_{max}-\lambda_{min}}{d^{e-1}}$, where d is the decay rate, in this case $d=1.05$

The value of λ as epochs progress is illustrated in Figure 1.

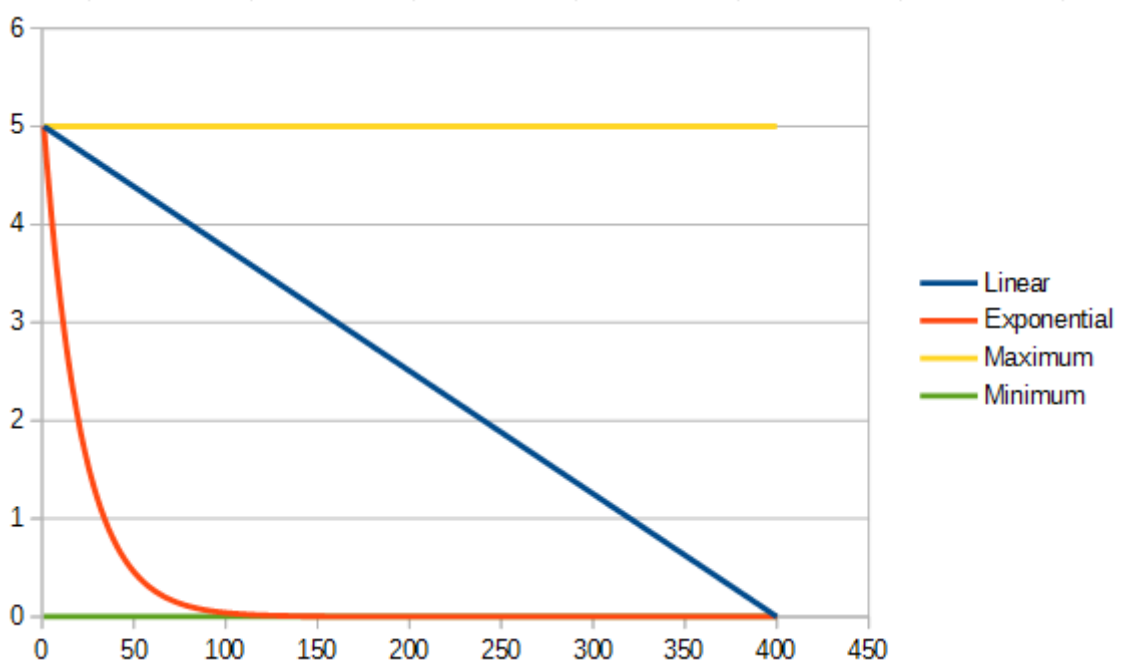

Figure 1: Regularization constant λ as epochs progress

For each of the regularization profiles, the network was initialized with the same set of randomly generated weights, so that any observed differences were not artifacts of the network approaching a different local optimum based on its initial weights.

However, in order to avoid learning behaviors that may be unique to a given initial set of weights and its nearby local optima, five sets of initial weights were randomly generated and the experiment was repeated for each of the five. The results of the five sets of experiments were then averaged and the data presented in this document is the average of those results.

At each epoch, four data points were collected: the average value of the cost function given each input from the training data (hereafter referred to as training cost), the percentage of inputs from the training data that resulted in correct output (hereafter referred to as training accuracy), the average value of the cost function given each input from the evaluation data (hereafter referred to as evaluation cost), and the percentage of inputs from the evaluation data that resulted in correct output (hereafter referred to as evaluation accuracy).

4. Results

4.1 Training Cost

Training costs are tracked in order to determine when the network has learned the training data so well that continuing to train on that data set is not likely to improve the network, or, in other words, the network has become saturated against that data set. The network is considered saturated when the slope of the training cost curve approaches zero.

As seen in Figure 2, the network using the maximum regularization value λ_{min} saturates near the 50^{th} epoch, meaning the remaining 350 epochs do not significantly improve the performance of the network relative to the training data.

The network using the static minimum regularization is the next to saturate at roughly 100 epochs, followed by the exponential decay network. 400 epochs was insufficient for the linear decay network to saturate. Future work should expand the number of epochs beyond 400 to find out when the linear decay network would saturate.

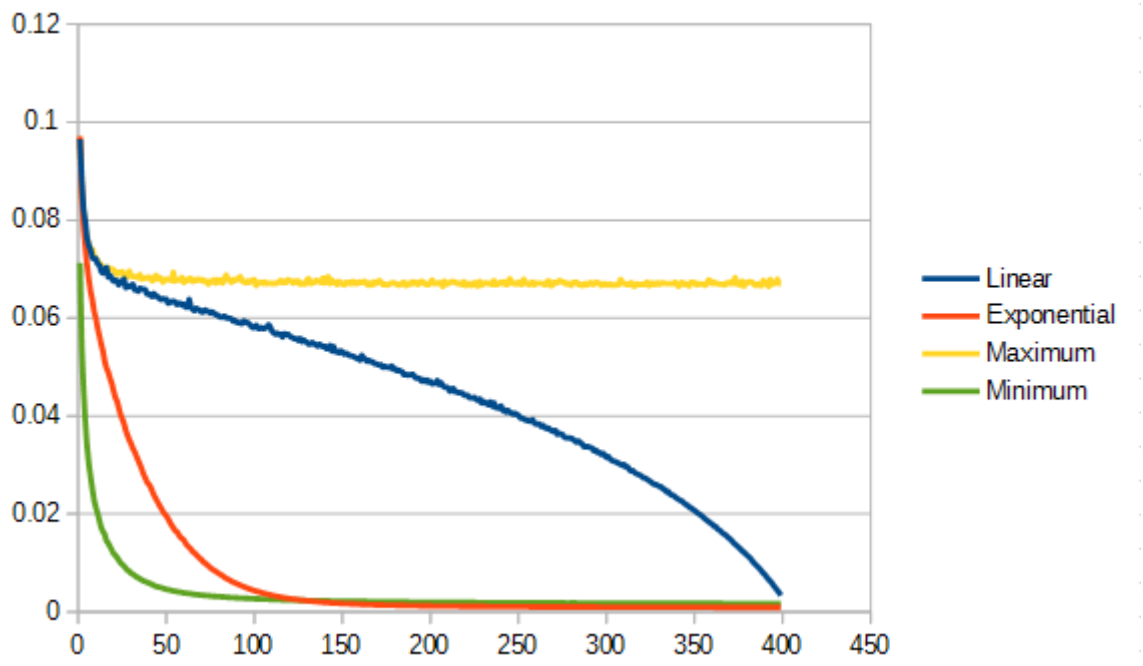

Figure 2: Training Cost

4.2 Training Accuracy

Training accuracy is tracked for two desirable traits. A higher final accuracy means the network was able to train better, while arriving at a high accuracy early means the network trained more quickly and perhaps could complete its training early, thus saving time.

As seen in Figure 3, the minimum, linear decay and exponential decay networks eventually arrived at a nearly identical level of accuracy, while the maximum regularization network was significantly less accurate.

The minimum, maximum and exponential decay networks both learned relatively quickly, while the linear decay network did not level off until near the end of the training period.

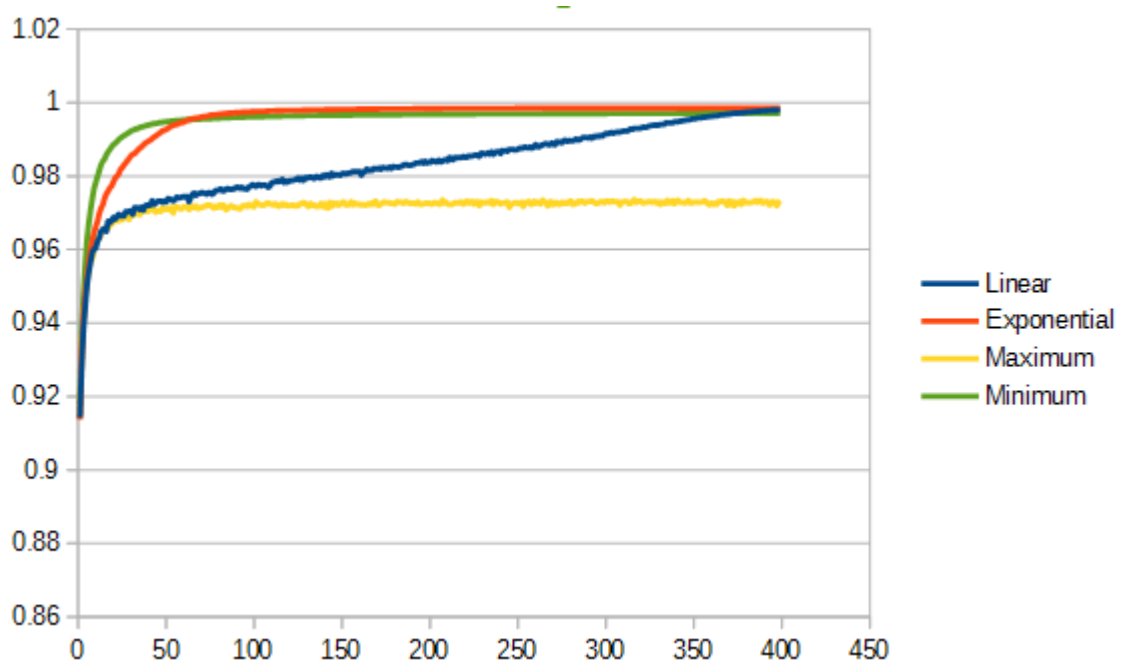

Figure 3: Training Accuracy

4.3. Evaluation Cost

The evaluation cost is tracked to determine whether the training is still effective. While the training cost measures saturation in the network, the evaluation cost measures the average error against non-training data. This is distinct from evaluation accuracy because the accuracy measurement treats the results as a binary success or failure, while the evaluation cost differentiates between near misses (where the error was just beyond the threshold of what is considered a correct answer) and fairly large errors where the output was nowhere near what was expected. Essentially, evaluation cost is a measure of convergence with the actual function that the network is seeking to learn.

As seen in Figure 4, the linear decay and maximum regularization networks actually increase in cost at the beginning of the training period. However, as the linear decay network moves away from the maximum regularization, its evaluation cost starts decreasing at an accelerating rate.

The maximum regularization network plateaus fairly high, indicating it never approaches convergence with the target function. The other three networks appear to reach similar levels of convergence, though the linear decay network's curve still has a steep slope at the end of the training period, indicating that if allowed to continue training, it could possibly surpass the exponential decay and minimum regularization networks.

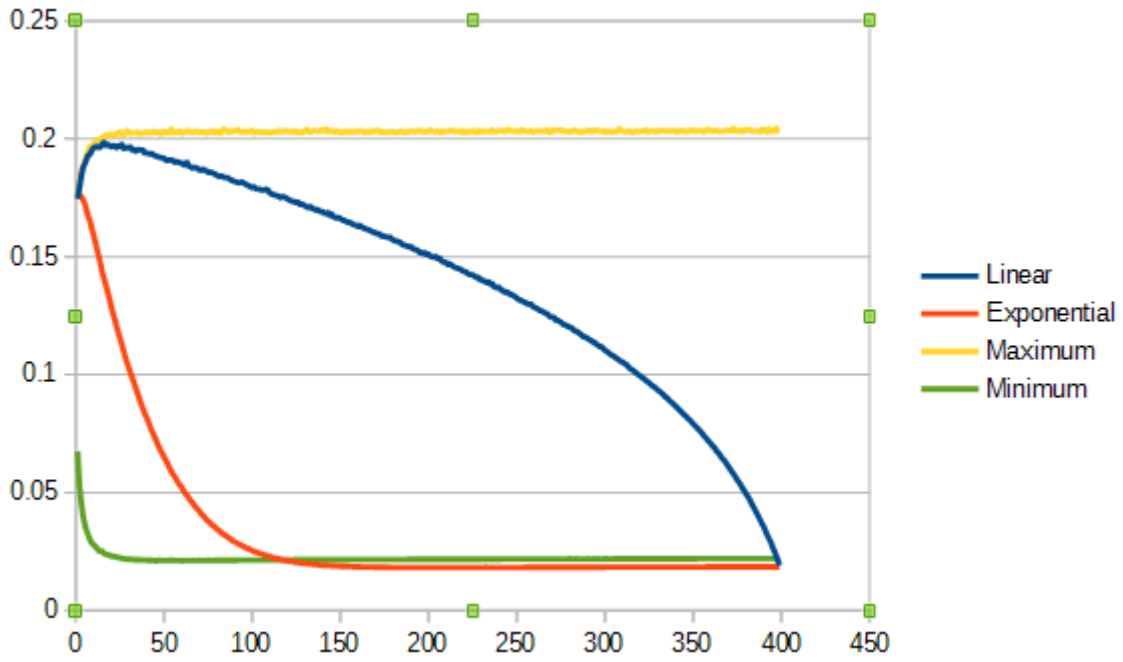

Figure 4: Evaluation Cost

4.4. Evaluation Accuracy

The evaluation accuracy measures how well the networks' training generalizes to inputs not present in the training data. A network's training is considered successfully generalized of the ratio of training accuracy to evaluation accuracy is near 1. Maximizing this number is the entire goal of training. Ideally, a network would be able to approach 100% accuracy in as few training epochs as possible.

Figure 5 shows that the maximum regularization network quickly plateaus as the worst performer for this network, followed by the minimum regularization network. The exponential decay network plateaus slightly later with a higher accuracy than the minimum network. The linear decay network continues to increase in accuracy over the course of the training period and surpasses the exponential decay network near the end of the training period. It does not appear to plateau yet, and performing the experiment with more epochs would indicate at what accuracy the linear decay network is would plateau.

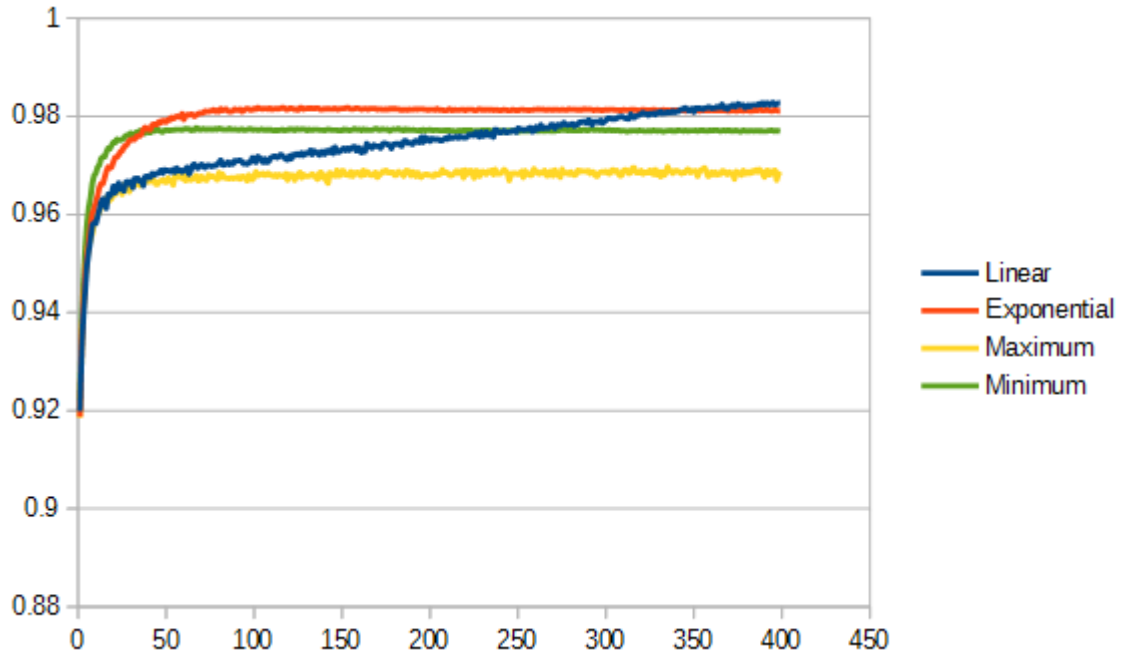

Figure 5: Evaluation Accuracy

5. Conclusions

It is evident from Figure 2 that both of the static regularization networks saturate fairly quickly, while the exponential decay network is able to delay saturation a little longer. The linear decay network never saturates in the 400 epochs for which it was allowed to train. This indicates that a variable decaying regularization constant has a positive effect on avoiding premature saturation.

While all but the maximum regularization network are able to train to a comparable level of accuracy against the training data, there is a significant difference between the networks in terms of evaluation accuracy. Figure 5 clearly demonstrates that the exponential decay network was significantly more effective at generalizing its training that the minimum regularization network, and though it is difficult to extrapolate past the end of the training period, it appear that the linear decay network eventually achieve equal or greater accuracy when compared to the exponential decay network. This seems to indicate that it is desirable to maintain a larger regularization constant so long as it eventually decays to a small value.

There is an evident trade-off in the two decay methods. The exponential decay method is able to approach its maximum evaluation accuracy fairly quickly, while the linear decay algorithm takes much longer to approach its maximum. However, it appears that the linear decay network would eventually outperform the exponential decay network, given enough epochs.

It would seem that approaching the minimum regularization quickly minimizes the number of epochs needed to train, while avoiding prematurely approaching the minimum regularization has a positive influence on eventual maximum accuracy. Decreasing the decay rate on the exponential decay algorithm would likely find a better balance of these two properties, where it would learn faster than

the linear algorithm, as it would approach the minimum regularization sooner, but it would eventually generalize better, as it avoided approaching the minimum regularization too soon.

6. Future Work

6.1 Increased number of epochs

This experiment should be redone with a longer training period to attempt to observe the behavior of the linear decay network beyond 400 epochs. It may be that this is impossible, since the linear decay is designed to adjust its slope based on the number of epochs, such that it always arrives at the minimum regularization value on the last epoch. In that case, the behavior observed in this experiment may be normal for any number of epochs, but further work could confirm or disprove that.

6.2 Adjust Exponential Decay Rate

This test was run with an exponential decay rate of $d=1.05$ (at each iteration e, $\lambda_e = \lambda_{min} + \frac{\lambda_{max} - \lambda_{min}}{d^{e-1}}$).

A smaller decay rate would move the curve seen in Figure 1 closer to the linear decay line. Such a curve may find a middle ground between the $d=1.05$ exponential decay network's fast learning and the linear decay network's apparent superior eventual accuracy.

6.3 Experiment with Other Decay Functions

The only decay functions observed in this experiment were linear and exponential. Other decay functions might prove to have interesting properties, though, such as one that decayed along some sort of a sigmoid curve, remaining near the maximum regularization value until fairly suddenly moving from it to the minimum.

6.4 Verify Results are not Based on Better Average

While the results indicate that the decaying regularization networks outperformed the static regularization networks, it is possible that the average regularization value of the decay networks was simply a better value for λ. This could be verified by re-running the experiment with a static value for λ equal to the average of the decay functions (2.5 for the linear decay function and 0.26 for the exponential decay function).

6.5 Compare to an Un-regularized Network

For comparison, a network with no regularization should be included with the results to prove that the regularization is actually improving the network performance.