Regularization's Influence on the Performance Of Network With Projection Shortcuts

Lizhen Zhu 516030910570 Team #5

1 Introduction

We add penalty term to the loss function to improve the performance of the network with projection shortcuts. If the performance is better, according to the message conveyed in the penalty term, we can verify some insight about shortcut is true. We use L1-norm and L2-norm, and design some new penalty terms based on the similarity between parameters to do the experiment. The result is that traditional L1-norm and L2-norm regularization can reduce over fitting rate but can't improve the performance on validation rate. By constraint parameters' similarity, validation accuracy can be improved to some extent when hyper-parameters are well-defined.

2 Motivation

When I train a classification model using network with shortcuts, the model is always over fitting. We know add penalty term can prevent over fitting. So we want to try different penalty terms to decrease over fitting rate at first.And we think penalty term, which use knowledge we have already known, can not only limit the simplicity of the model but also restrict the model in other ways, the similarity of parameters, for example. So we want to find out different penalty terms' influences on network with projection shortcut. Also, the results can give us some insight about shortcut.

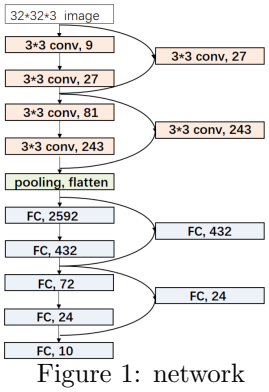


Figure 1: network

3 Experiments & Results

We do our research on cifar-10 dataset which has $32 * 32$ images divided into ten completely mutually exclusive classes. We randomly choose 20000 imagines as training data and 1000 imagines as validation data used to evaluate the

model's performance. We design a network which is illustrated in the picture on the right. For every results in the experiment, we run three times and average them.

First, we don't add any penalty to the loss function and the result turn out to be overfitting. The accuracy of training data become very close to one, while the validation accuracy becomes worse and worse after it comes up to sixty percent.

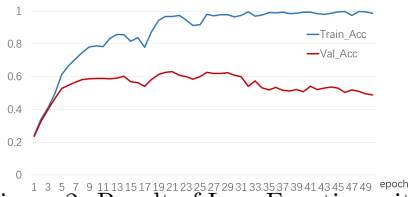
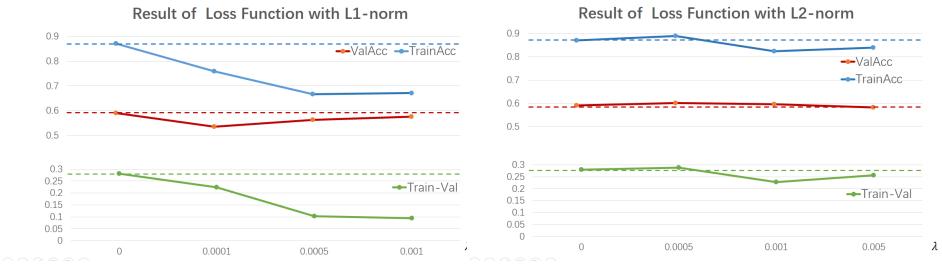


Figure 2: Result of Loss Function without any Penalty

3.1 L1-norm, L2-norm

We add L1-norm $\lambda \sum_{\omega} |\omega|$ to the loss function.As the λ become larger. the train accuracy and the over fitting rate become smaller, which illustrate the L1-norm can reduce the over fitting rate to some extent. However, the validation accuracy becomes smaller at first and then becomes larger and the finally result is still a little worse than the original validation accuracy. We think maybe in the back propagation the parameters are updated in terms of whole matrices, and radical in preventing overfitting. So although the overfiting rate decrease obviously, the validation accuracy are also affected.

We add L2-norm $\lambda \sum_{\omega} \omega^2$ to the loss function. The validation accuracy is not obviously affected. And the over fitting rate decreases not so obviously as L1. We think it performs like this because L2-norm updates the parameters proportionally which is much soft and better.



(a) L1-norm (b) L2-norm

Figure 3: Result of Loss Function with L1,L2

3.2 Orthogonal Regularization

We design some ways to evaluate the difference of parameters. For matrices, because in the process of calculating the result, they multiply the vertices in

network so we can know every column of the matrices do the same thing to the vertices in network. We eevaluate two matrices operation's difference by the results of each two column's dot production. The form goes like this $\omega_1^T \cdot \omega_2$, and we have to minimize every number in the matrix. The result is that the train and validation accuracy behave in the same way, in some points they are much better than the origin accuracy. We can know that if the parameters in the left part and the shortcut part are different to some extent the performance will be better.

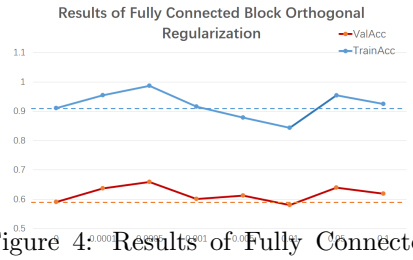


Figure 4: Results of Fully Connected Block Orthogonal Regularization

3.3 Kernel Difference Regularization And Combine two Ways of Regularization

For convolutional kernels, they multiple every corresponding position and then add them all, so we think using the sum of squares of the differences in corresponding positions is proper. The formula is like this $\sum(\omega_{i_{i,j}} - w_{2_{i,j}})^2$, where ω_1 means convolutional kernels in shortcut and ω_2 means convolutional kernels in left part of the network. We first add kernel difference regularization alone and then add it with L2-norm which performs better in restricting the complexity of parameters. We can see from the result below that when the differences of kernels and the hyper=parameters of L2-norm between kernels are in some specific point the result will be better.

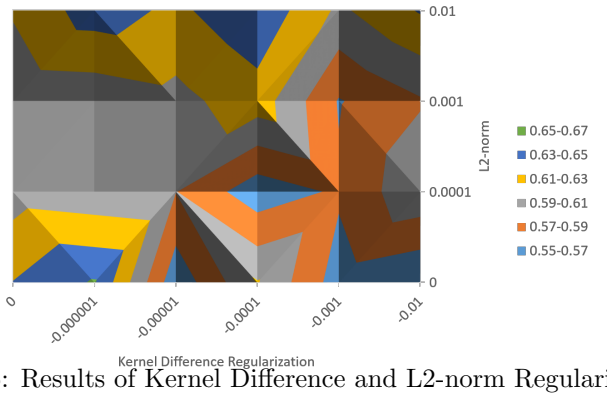


Figure 5: Results of Kernel Difference and L2-norm Regularization

4 Code

Our code can be seen in `https://github.com/Ldhlwh/RegularShortcut`.

5 Reflection

There are still some drawbacks about this research:

1. The valence of the result is kind of large, for we don't do the experiment enough time.
2. More preliminary experiments should be done before we decide which value of λ, the hyper parameter of the penalty term, should be chosen.
3. Orthogonal regularization and kernel difference regularization should be added to the loss function together to see the results.
4. If the experiment can be improved, we many derive the mathematical formula between λ and the accuracy.