Machine Learning Project Report

Hikmet Orhun Guley

orhun.guley@ozu.edu.tr

Abstract

In this project, I design 3 different neural networks on TensorFlow using CIFAR-10 dataset to classify the type of a objects.

The CIFAR-10 data: The dataset contains images of various images of some objects—such as airplanes, automobiles, birds, cats, deer, and dogs. The problem is an image classification problem, which can be solved by properly designed neural networks.

1 Introduction

My focus in this project is to gain experience by designing convolutional neural network on the CIFAR-10 dataset. The problem is famously known image classification problem. In this project, I designed a neural network by using TensorFlow framework. The goal was to classify images of some objects. In this report, I mostly analyze the relationship between number of epochs and accuracy of the model. In addition, I tried 2 different regularization techniques to prevent overfitting. These techniques are L1 regularization and L2 regularization. The dataset is more challenging than the MNIST-Fashion dataset I worked before. The comparison is made for both training accuracy and testing accuracy.

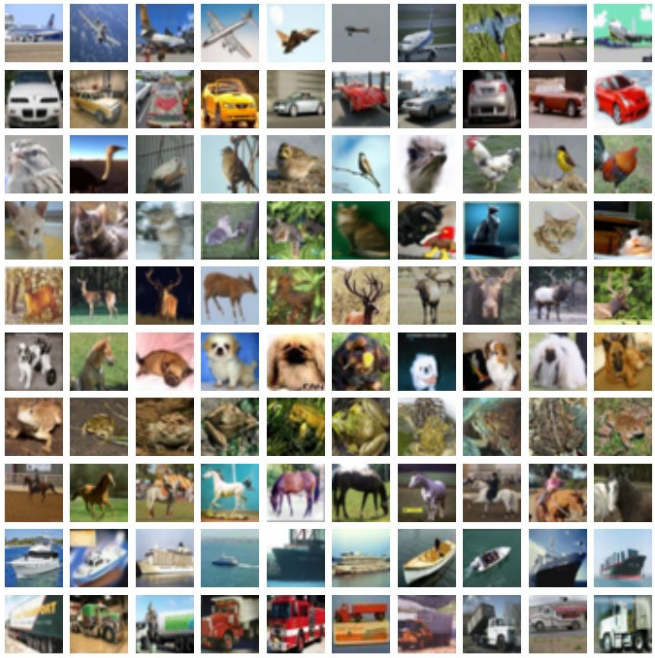
2 Problem and Dataset Information

I have used the data from cs.toronto.edu which is Toronto University repository. The CIFAR-10 dataset contains images of various objects and animals—such as frogs, horses, ships, trucks, a total of 10 classes.

The CIFAR-10 dataset consists of 60000 32x32x3 color images in 10 classes, with 6000 images per class. These classes are;

- T-shirts

- Trousers
- Pullovers
- Dresses
- Coats
- Sandals
- Shirts
- Sneakers
- Bags
- Ancle boots



- **Training set** is consist of **50000** images. (85%)
- **Tests set** is consist of **10000** images. (15%)

I have used one neural network architecture to classify images. For optimization, Adam Optimizer is used to minimize the loss, making use of TensorFlow framework.

3 Neural Network Design, Optimizations and Regularization Algorithms

In this section, I discuss which optimization algorithm I used to minimize de loss, the designs of the neural network and the regularization methods I used to improve accuracy.

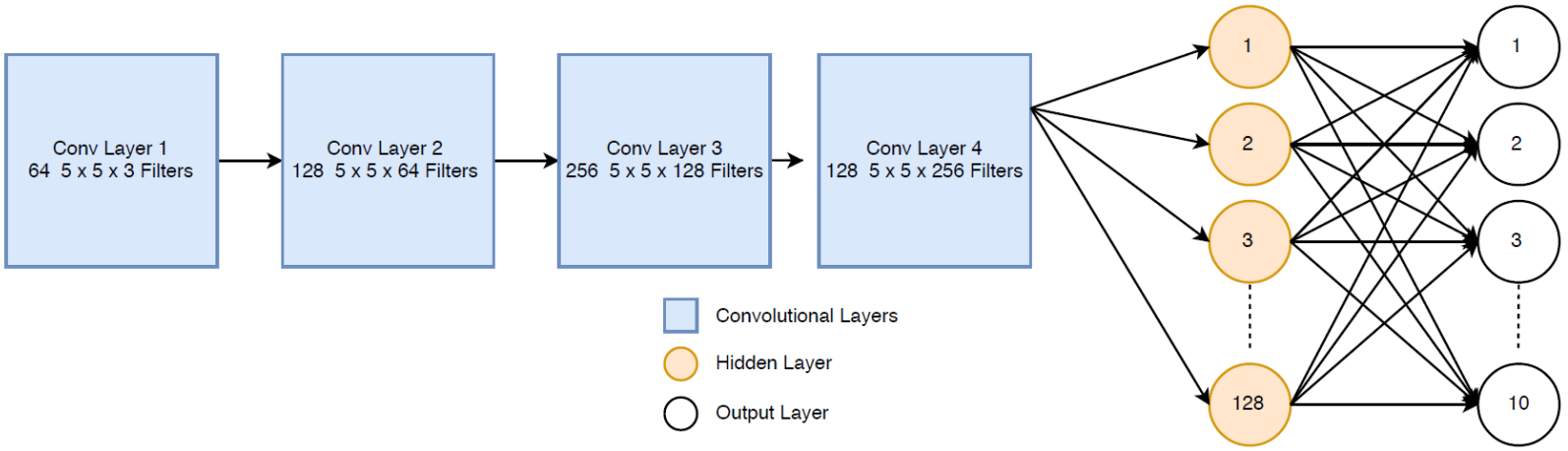
3.1 Adam Optimization Algorithm

I used Adam Optimizer to minimize the loss function since it achieves good results fast. That is one of the reasons that Adam Optimizer is one of the most popular one in the field of deep learning. Adam Optimization algorithm is some variation of stochastic gradient descent, which we have not covered yet but with Prof. Şensoy's suggestion, I used as the optimization engine in my neural networks.

3.2 Neural Network with 4 Convolutional Layers and One Hidden Layer

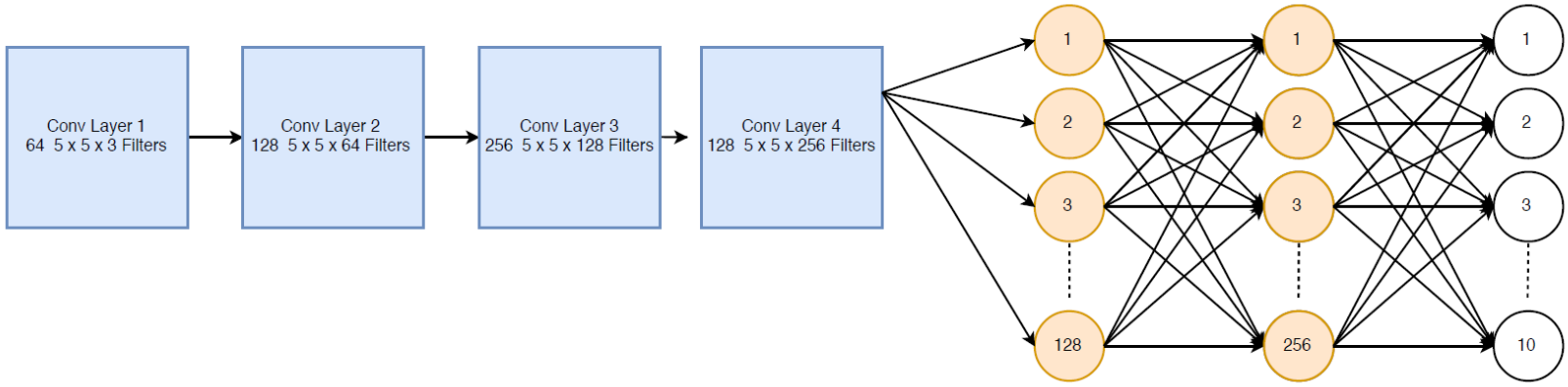
In my first design, I built a multi-layer convolutional neural network with 4 convolutional layers and one fully connected layer. The first convolutional layer is consist of 64 5 x 5 x 3 filters. The second convolutional layer is consist of 128 5 x 5 x 64 filters. The third convolutional layer is consist of 256 5 x 5 x 64 filters. The fourth convolutional layer is consist of 128 5 x 5 x 256 filters. The output of the convolution layers are processed by the activation function ReLU, which merely ensures that the output is positive because negative values are set to zero. After that process, maxpooling is applied to the output. The hidden layer computes has 128 fully connected neurons. You can see the architecture of first neural network below:

- The dataset is consist of 32 x 32 x 3 images.



3.3 Neural Network with 4 Convolutional Layers and Two Hidden Layer

The second neural network built a multi-layer convolutional neural network with 4 convolutional layers and two fully connected layer. As the first one, the convolutional layers have the same size. I just add more hidden layer before the output layer, which has 256 neurons. You can see the architecture of second neural network below:



3.4 LASSO (L1) Regularization

Since the accuracy on the test data did not improve and did not have a high value, I tried to apply regularization methods as it is asked. The first regularization method I implemented is L1 regularization. After applying L1 regularization, the accuracy did not improve. Maybe I implemented it wrong, or it is not a suitable one for CIFAR-10 dataset. You can see the loss function of L1 regularization below;

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

3.5 Ridge L2 Regularization

Since the L1 regularization did not help the accuracy, I implemented L2 regularization. Actually, L2 regularization help the accuracy to improve. I was able to increase to accuracy to %70-%75. You can see the loss function of L1 regularization below;

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

4 Results

In this section, I observe the relationship between accuracy and number of epochs, for both training and testing datasets. In addition, I compare the accuracy results between L1 and L2 regularization methods.

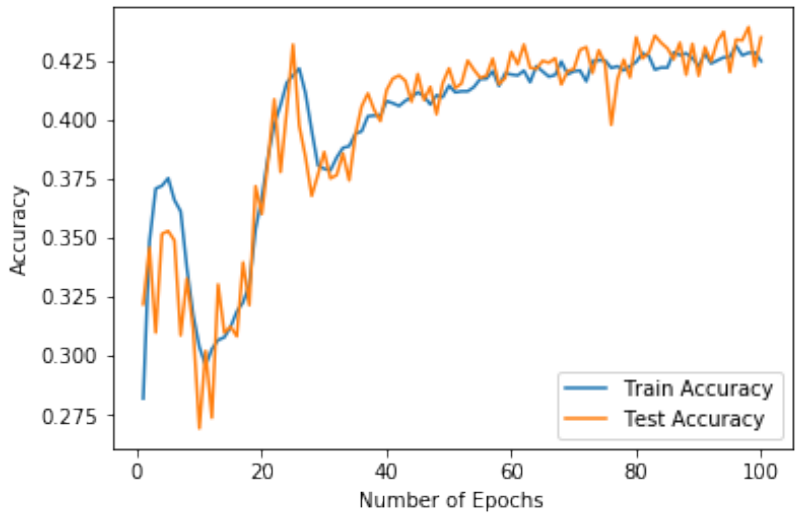
As it is mentioned in the project 2 assignment pdf, I tried to implement the **save/restore** property of TensorFlow. Unfortunately, it did not work on my laptop. The same code works on my friend's laptop but in my laptop, it gave me error.

In the first 2 experiments, I used the 1st convolutional neural network design with 4 convolutional layers and one fully connected layer. I trained the data for 100 epochs. In the first one, I used L1 regularization method and in the second experiment, I used L2 regularization method.

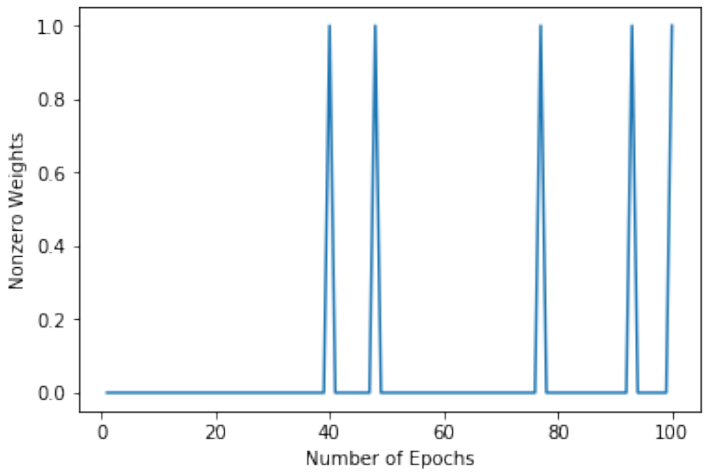
For the first 2 experiments, 100 epochs are applied and the batch size is 100. Adam Optimizer is used in both networks and also for activation functions, ReLU function is preferred.

While calculating the test set accuracy, it gave me the in "ResourceExhaustedError" some trials. As a solution, I wrote a function which calculates the accuracy in 20 steps. As 10000 test set images with dimensions 32 x 32 x 3 is challenging for my laptop, I implement a loop that takes 500 images in one iterations. In that way, I am able to calculate the test set accuracy in 20 steps.

You can see the graphs for 1st experiment (L1 regularization) below;

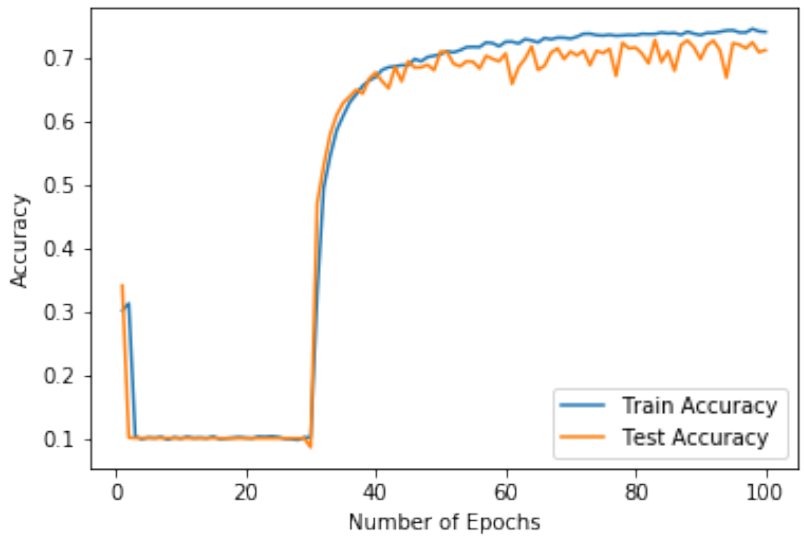


L1 regularization in CIFAR-10 dataset did not give me good results. As you can see from the graph, the maximum test set accuracy that first experiment can be reach is around 41%. The graph below shows the number of zero weight in the first experiment. One of the main differences between L1 and L2 regularizations are sparsity. To analyze that difference I implemented a graph that shows the number of zero coefficients in the weights. The graph can be seen below;

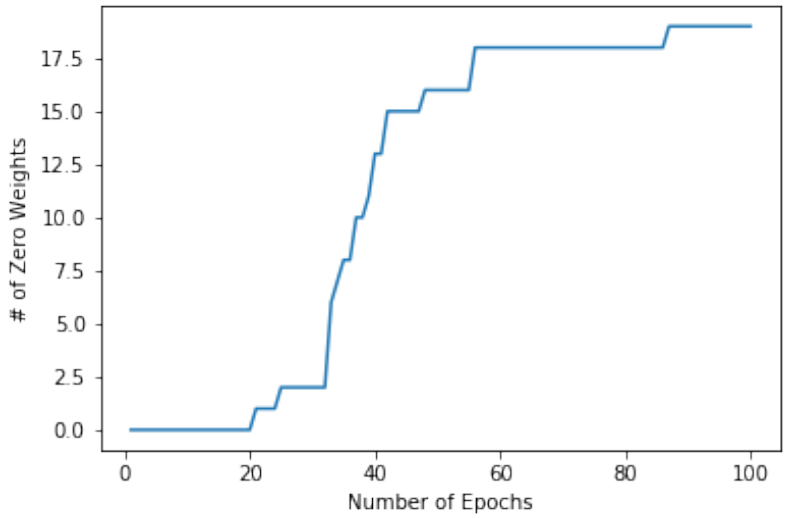


The number of zero coefficients are not like that I expected. Of course, there is a possibility that I implemented the L1 regularization wrong.

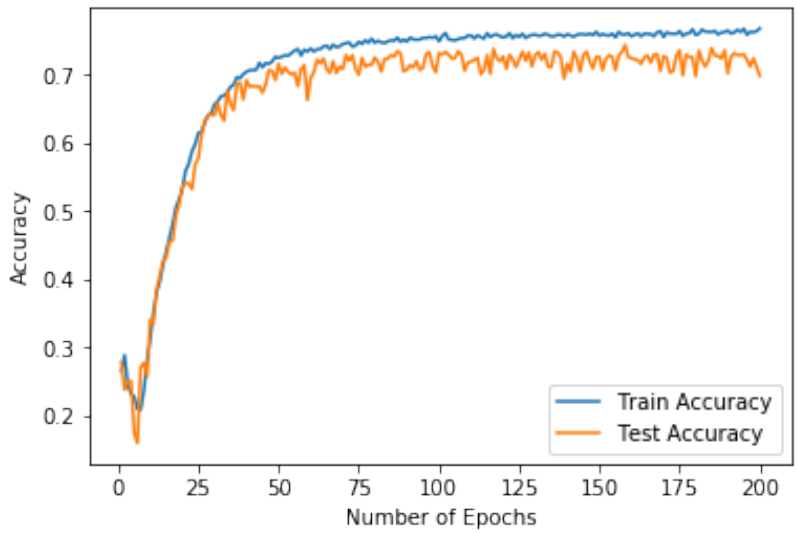
As I mentioned before, in the second experiment, I used L2 regularization. You can see the graphs below;



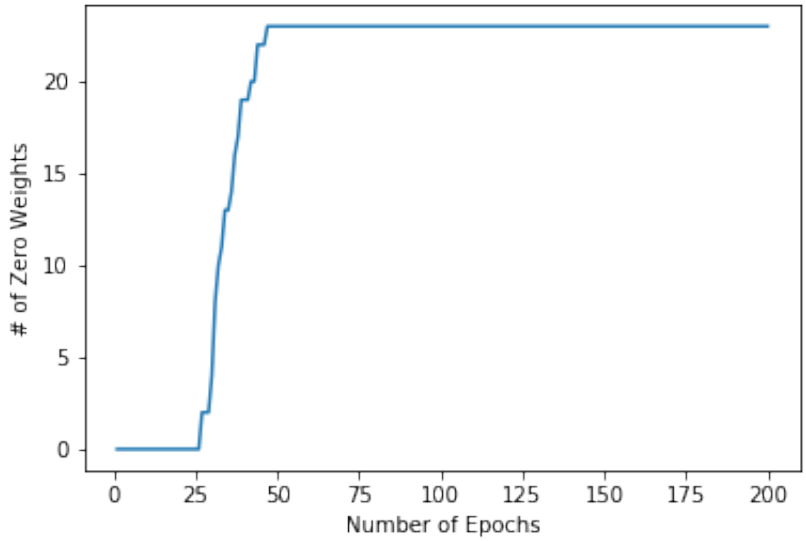
Maximum accuracy reached in the 82nd epoch, which is %72.88. L2 regularization for CIFAR-10 dataset makes a huge difference in terms of test and train set accuracy. Below, you can see the graph that shows the number of zero coefficients in the weights;



For 3rd experiment, I used a different neural network architecture which has 4 convolutional layers and two fully connected hidden layers. In this experiment, I set the number of epochs to 200. You can see the accuracy graph below;



Maximum accuracy reached in the 157th epoch, which is % 74.32. If we compare experiment 2 with experiment 3, we can observe that there is not much a difference between experiment 2 and experiment 3 in terms of accuracy. You can see the graph that shows the number of zero coefficients in the weights;



5 Conclusion

As conclusion, for each experiments I have designed, I calculate the accuracy of the train and test sets and compare with each other. I also constructed graphs that show the number of zero weights in the neural networks I designed.

In this project, I am able to see that L2 regularization gives better results than the L1 regularization in terms of accuracy. If we compare the two experiments with L2 regularization, we can conclude that 100 epoch is enough for this experiments. Increasing number of epoch does not make much of a difference. Different approaches can be done in order to increase the accuracy. Applying data augmentation and dropout methods can help to increase accuracy.

Designing different types of neural networks and analyzing the performance of these networks help me to understand which parameters are affecting the performance measures like accuracy and loss. By making these kind of project, I hope to get better at this field.

References:

http://christopher5106.github.io/deep/learning/2016/09/16/about-loss-functions-multinomial-logistic-logarithm-cross-entropy-square-errors-euclidian-absolute-frobenius-hinge.html

https://www.cs.toronto.edu/~kriz/cifar.html

https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c