COMP 551 Assignment 3

Emi Baylor, Amy Hynes, Karla Gonzalez

April 19, 2020

1 Abstract

In this project, we investigated the performance of two neural networks - a multi-layer perceptron (MLP) and a convolutional neural network (CNN) - on an image classification task using the CIFAR-10 dataset. The MLP model achieved 55% accuracy, and the CNN achieved 71% accuracy. We found that the number of epochs a model is run for and the learning rate had large impacts on the accuracy of these models.

2 Introduction

The goal of this project was to compare the performance of two neural networks, MLP and CNN, on an image classification task using the CIFAR-10 dataset. The task is categorical classification where the goal is to correctly classify images into ten classes.

A multi-layer perceptron (MLP) is a network which maps sets of input data onto a set of outputs in a feed-forward manner. A MLP is composed of interconnected nodes in multiple layers (input, hidden, and output layers), with each layer fully connected to the preceeding and succeeding layers. The outputs of each node are weighted units followed by a nonlinear activation function to distinguish the data that is not linearly separable.

A convolutional neural network (CNN) is a variant of the multi-layer feed-forward neural networks, mostly used for image classification and segmentation. A CNN is designed to take an input image and process it in multiple arrays by considering local and global stationary properties. Similar to the MLP, the CNN is a network stacked into a number of layers, where adjacent layers are connected by a set of learnable weights and biases. The major difference is that in a CNN, each layer is represented as input and output feature maps by applying a convolution operation to capture different perspectives on features.

We ran our models multiple times, each time systematically modifying the model in order to obtain the highest accuracy possible. We took multiple approaches and were able to modify our entropy cost function, the number of epochs the model traverses, and the size of our model (which includes number and size of the layer composition). Additionally, for the CNN model, we also tested the effect of varying the number of convolutional and linear layers.

Having run these experiments, we obtained our highest accuracy of 70.55% on the test dataset and 85.02% on the training set using the CNN model run for 18 and 20 epochs respectively with a scheduled learning rate. The MLP model achieved 54.51% accuracy on test set and 94.56% accuracy on training set run for 16 and 19 epochs respectively using a learning rate of 0.001.

3 Discussion of Related Work

An article by Chang et al. claims a 14% test error rate for a MLP model and an 11% error for a CNN on the CIFAR-10 dataset [1]. Both of these models used more complex architectures and techniques than seen in this project, but the potential for great results using these models as the base exists.

A paper by Mukkamala et al. explores the effect of different optimization functions on the performance of a MLP on the CIFAR-10 dataset [3]. They found that stochastic gradient descent (SGD) produced better results than the adam optimizer, although adam stabilized earlier. SGD took about 150 epochs to reach peak performance, while adam plateaued around 40 epochs.

An article by Zhang et al. explores the idea of a hybrid of the two models used in our project [5]. The two algorithms, which behave very differently, were integrated using a rule-based decision fusion approach for the classification of very fine spatial resolution remotely sensed imagery. This MLP-CNN classifier achieved promising performance, consistently outperforming MLP and CNN working independently.

4 Dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Among them, the training batches contain exactly 5000 images from each class [2]. The data was loaded and normalized using PyTorch [4].

5 Approach

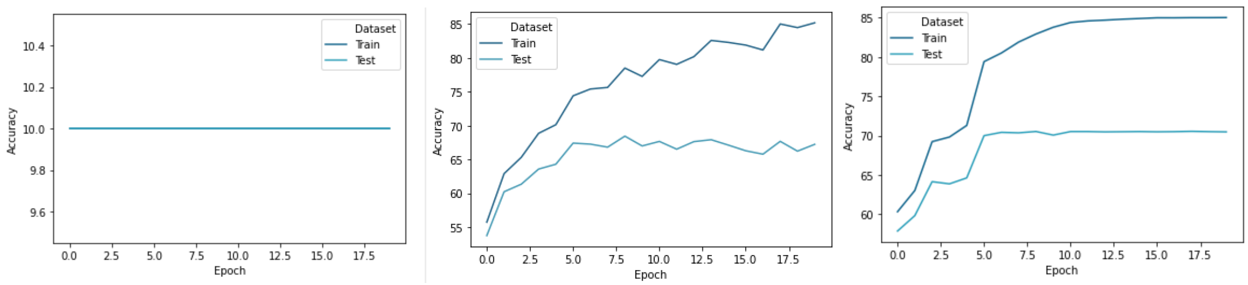
The data was loaded and normalized using PyTorch. A testing pipeline was implemented to experiment with changing different hyperparameters such as the number of epochs, the size of our model and the learning rate. These experiments were run on both the MLP and the CNN. When we explored the effects of changing the size of our model, we varied the number of layers used, as well as the size of the layers. The learning rate refers to the tuning parameter in an optimization algorithm that determines the step size at each iteration while minimizing the loss function. We used adam for the optimization function of the CNN, and SGD for the MLP. Adam is an adaptive learning rate optimization algorithm that is designed specifically for training deep neural networks. Our MLP model uses ReLU as the activation function for the hidden layers and a softmax classifier on the output layer.

6 Results

Our CNN was implemented with 2 convolutional layers and 3 linear layers. The linear layers had shapes of $[2048, 128], [128, 128], [128, 10]$. Our MLP was implemented with 3 layers with shapes 3 layers of $[2048, 512], [512, 128], [128, 10]$. For both classifiers, a batch size of 4 was used. When the learning rate was scheduled it was initialized at 0.001 and multiplied by 0.1 every 5 epochs.

CNN Results

Learning Rate	Train Max Accuracy %	Train Max Accuracy Epoch	Test Max Accuracy %	Test Max Accuracy Epoch
0.01	10.0	All	10.0	All
0.001	85.18	19	68.44	9
Scheduled	85.02	20	70.55	18



(a) Learning Rate = 0.01 (b) Learning Rate = 0.001 (c) Learning Rate = Scheduled

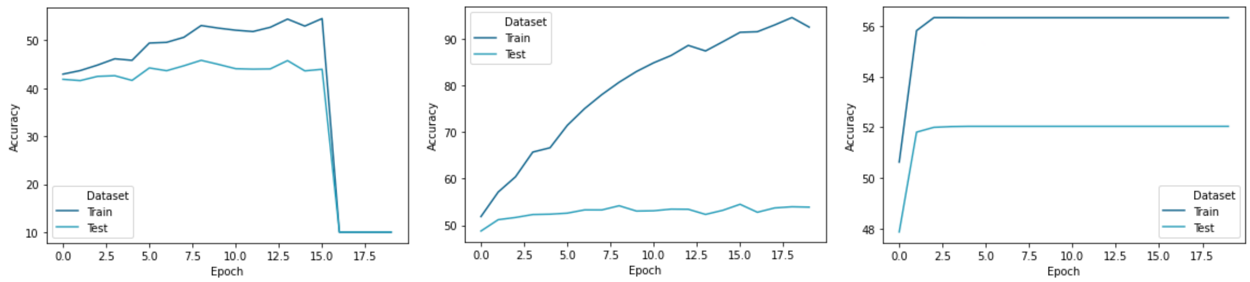
Figure 1: CNN accuracy over 20 epochs

The CNN model performs best with the scheduled learning rate with an accuracy of 71% on the test set. This is an improvement over the 0.001 learning rate, as the scheduled learning rate seems to reduce overfitting on the training set, leading to better test performance. As seen in figure 1(b) & (c), the accuracy on the test set stops improving after approximately 5 epochs, but the training accuracy continues to improve up to 20. This implies the model begins to overfit when the test set accuracy stops increasing.

For the learning rate of 0.01, the loss was approximately constant, around 2.3 regardless of epoch. This indicates that the learning rate is too high for the model to gain any useful information over each epoch. As seen in figure 1(a), this model performed at random chance (10% accuracy) on both the train and test sets, and performance did not improve over the epochs.

MLP Results

Learning Rate	Train Max Accuracy %	Train Max Accuracy Epoch	Test Max Accuracy %	Test Max Accuracy Epoch
0.01	54.54	15	45.83	8
0.001	94.56	19	54.51	16
Scheduled	56.32	3 & 4	52.04	5-20



(a) Learning Rate = 0.01 (b) Learning Rate = 0.001 (c) Learning Rate = Scheduled

Figure 2: MLP accuracy over 20 epochs

The MLP model performs best with a learning rate of 0.001 for an accuracy of 55% on the test set. This learning rate also saw the best performance on the training set, with an accuracy of 95%. This indicates that the model suffers from overfitting which is reflected in figure 2(b).

As with the CNN model, the learning rate of 0.01 produced poor results, suggesting it may be too large of a learning rate. This model appears to start off weaker than the other models, but the performance is not too bad until the 16^{th} epoch when the accuracy on both the train and test sets drops. The scheduled learning rate produced similar results as the loss stopped changing after epoch 2, and the accuracies for both the train and test sets do not improve after that point as seen in figure 2(c). This suggests we may have found a local minimum which is different from the global minimum we found with the 0.001 learning rate. The model did not overfit with the scheduled learning rate.

7 Discussion & Conclusion

Learning rates have a large impact on the accuracy of these models. The difference between our worst and best performance using the CNN model was 61%, and the only parameter changed was the learning rate. Scheduled learning rates can help with overfitting, something neural networks are susceptible to. It is not surprising that the CNN achieves better results than the MLP, since the intended use of a CNN is image classification. Overall, these models perform very well at a complex image classification task. A classifier which randomly guesses each class with equal probability would achieve 10% accuracy on this classification task. Both neural networks implemented in this project perform significantly better than this baseline.

A possible direction for future investigation would be to use a dropout weight parameter in the MLP model to help reduce overfitting. Another area for investigation would be running 100+ epochs with a smaller learning rate to see if further improvement on test set accuracy can be achieved. This requires more computational power than we could access for this project.

8 Statement of Contributions

Emi implemented the CNN and MLP models. Amy assisted with the MLP model, and she and Karla wrote the report.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Batch-normalized maxout network in network. *CoRR*, abs/1511.02583, 2015.

[2] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 2009.

[3] Mahesh Chandra Mukkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds. 08 2017.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Ce Zhang, Xin Pan, Huapeng Li, Andy Gardiner, Isabel Sargent, Jonathon Hare, and Peter Atkinson. A hybrid mlp-cnn classifier for very fine resolution remotely sensed image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:133–144, 06 2018.