

DS5220 - Final Project Report

Kevin Russell

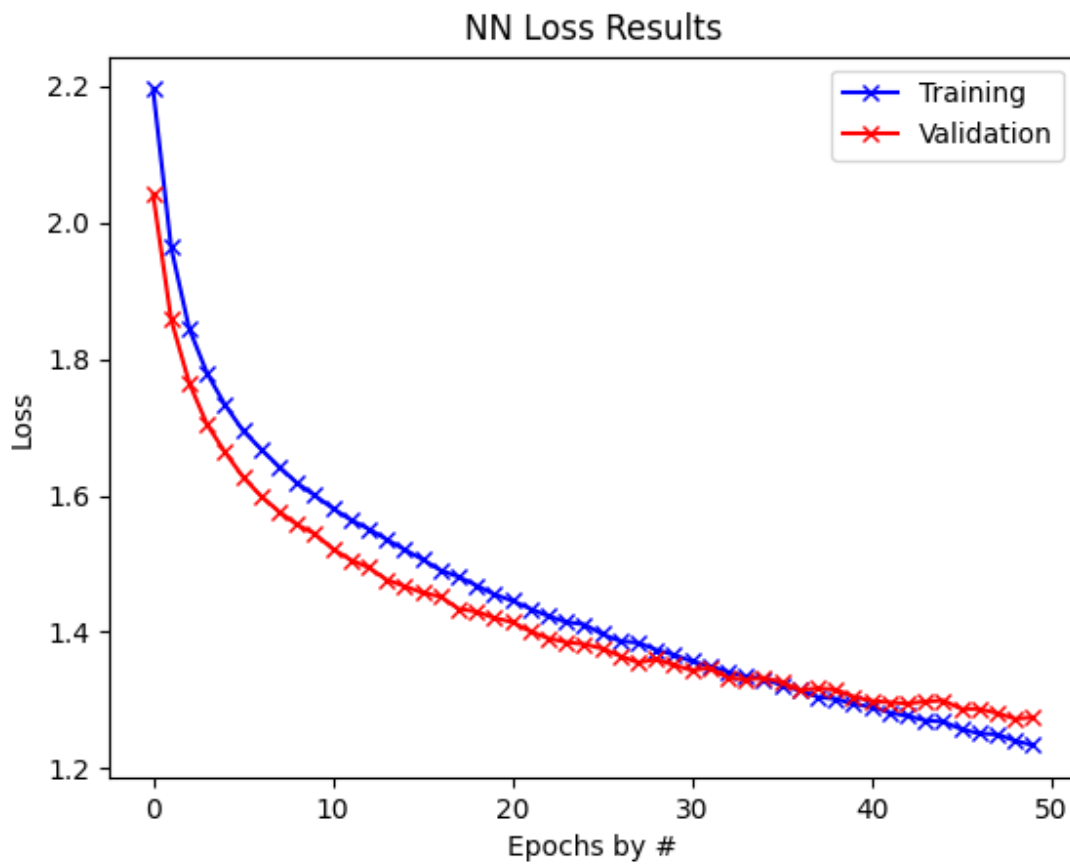
2022-05-01

Introduction

For my final project, I chose to utilize neural networks and convolution neural networks in order to classify the CIFAR-10 data set.

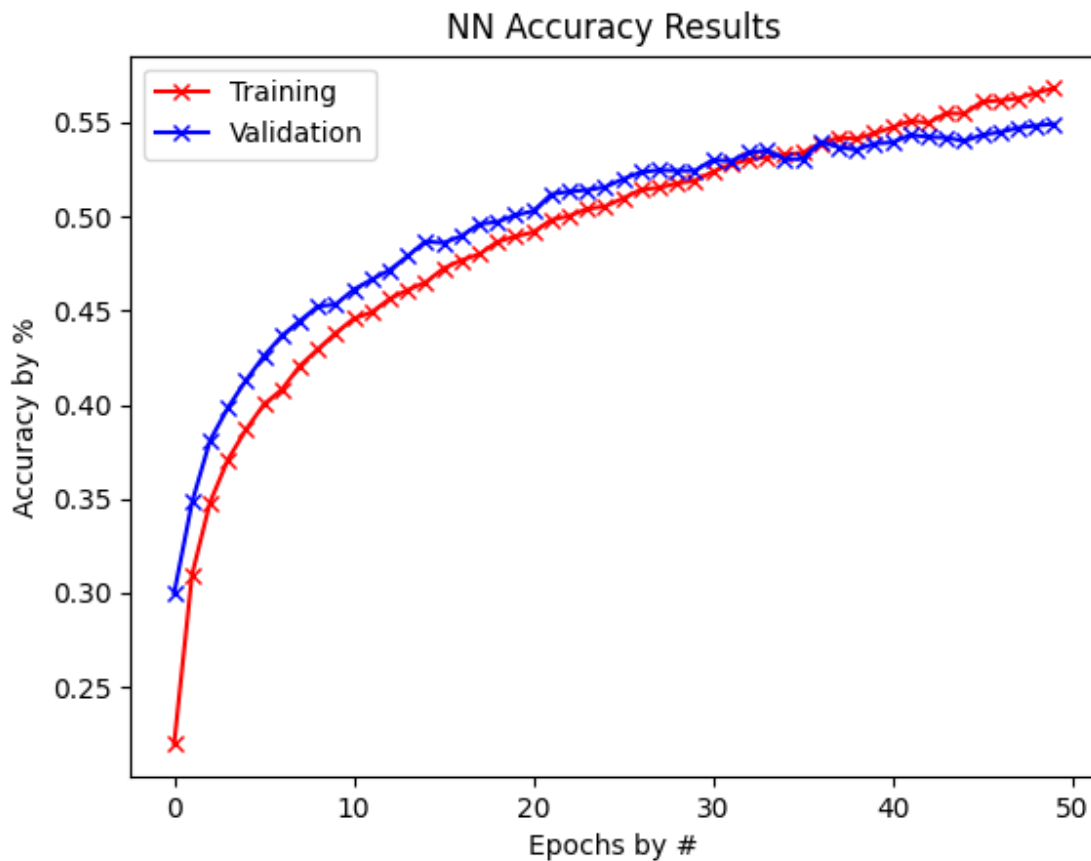
Neural Networks

Loss Analysis



For my neural network, the accuracy loss is comparable to the training loss except when the epoch reached 38 and greater. That is when the training loss started to become better than the validation loss. I have synthesized many different configurations and found that dropout with higher features seems to have better accuracy than having many Linear functions with relu activation functions. I also found that .3 dropout yielded the best performance for higher feature set linear combinations. Additionally, it seemed as if the dropout acted like a regularization because without it there was noticeable differences between the training accuracy and the validation accuracy.

Accuracy Analysis



Accuracy was consistently better than the training accuracy. This could be due to the lag-time of training accuracy calculation and the validation accuracy calculation. However, validation accuracy started declining in rate starting on the 15th epoch, but was still consistently better than training accuracy until the 38th epoch. In total, the runtime of this neural network was 50 epochs, which converged successfully on my machine.

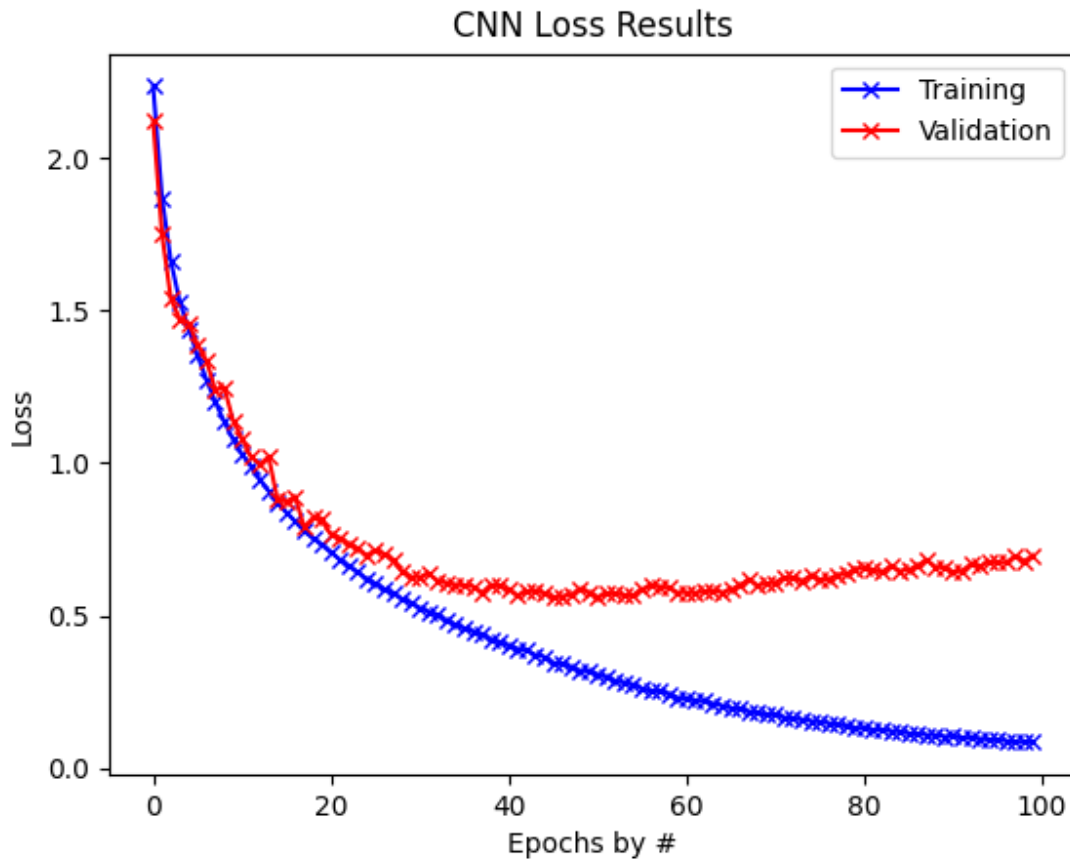
Confusion Matrix



The confusion matrix shows that the accuracy ranges from 32% to 74% for correct classification. It looks like the network had trouble classifying cat, dog and bird.

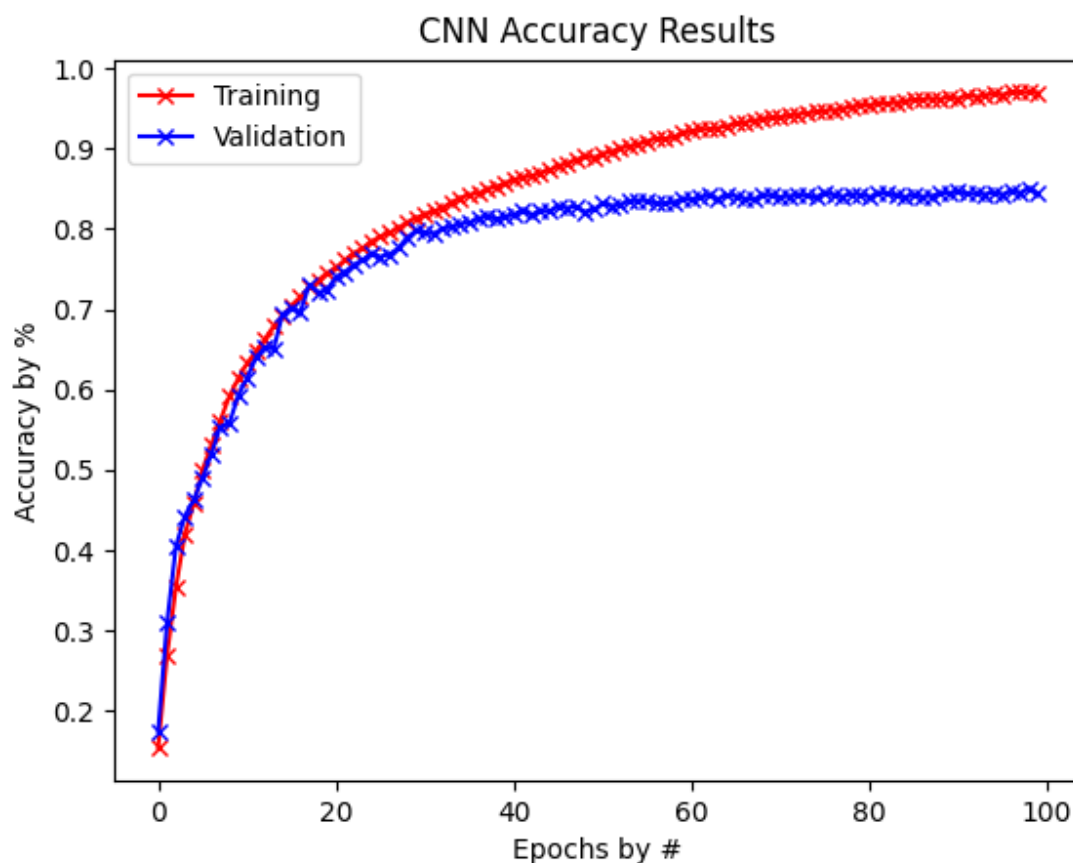
Convolution Neural Networks

Loss Analysis



For my convolution neural network, the accuracy loss is comparable to the training loss except when the epoch reached 20 and greater. That is when the training loss started to become better than the validation loss. I have synthesized many different configurations and found that performing a convolution to a normalization to a relu yields incredible performance. Additionally, it seems that adding more than 2-3 convolutions in a row before a max pooling yields worse results. Another note is that a dropout of 25% works great as regularization after it is applied to a max pooling. It took a while to figure out a set of channel changes for the convolutions as well. I do believe that there are a lot of improvements to be made for the model in terms of weight initiation, loss function, channel/feature dimension optimization, and finding the right balance in layers with application. I do wonder whether adding the original value before applying a sequential chunk would yield better performance. This is something to be explored in the future. Overall, I am not happy with the loss being stuck at the 0.5 range, but I do think this is a good first attempt at convolution neural networks. I also don't like how inefficient this model is in terms of handling memory.

Accuracy Analysis



Accuracy was consistently the same as training accuracy until the 80% range. It seems that I hit some sort of blocker at the 80% accuracy point where training accuracy was getting better while validation accuracy kept the same. It is as if the model had issues differentiating the low level, medium level, and high level features to accurately classify the image. Like, I wonder if the network needed to be deeper, whether the weights needed to be initiated a certain way, and/or whether the network needed to overlay the image features over the processed features from convolution and pooling transformations. The model went through 100 epochs that converged successfully on my machine.

Confusion Matrix



The confusion matrix shows that the accuracy ranges from 74% to 92% for correct classification. It looks like the network had trouble classifying cat, dog and bird like it did in the regular neural network. These three classes must be the most troublesome classes to get high accuracy.

Conclusion

Convolution Neural Networks had the best accuracy overall with an overall accuracy of 84% at best whereas the NN model had 54% at best. Neural network had a smoother convergence at 50 epochs than convolution neural networks had at 100 epochs. Ideally, Convolution Neural Networks would have been better at 30 epochs at its current build. If the CNN did stop at 30 epochs then CNN would have had better convergence

than the NN model with better accuracy. There is still much to improve whether optimizing the forward operations, weights, cost function, and layers.

Source Code

https://github.com/krussellneu/CIFAR-10_Neural_Net_Project