Maya Hussein

CSCE-4604

October 2022

Lab Assignment 2

1 Introduction

This lab assignment tackles the understanding and application of multinomial classification using fully connected layers and Convolutional Neural Network (CNN). The aim of the lab is to be able to classify handwritten numbers from 0-9 using multinomial classification. Below are the detailed description of each network type.

2 Part I: Deep Forward Neural Networks

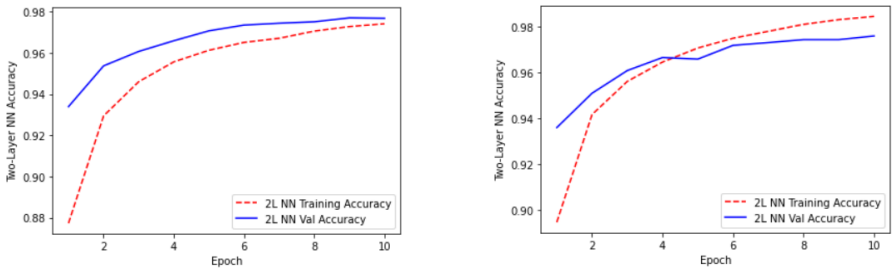
2.1 Activation Models

2.1.1 ReLU Activation Function

The below accuracy vs. Epoch graphs are modeled using a ReLU first dense layer with 128 neurons in the first layer and 10 in the second as illustrated in the notebook. The accuracy of the model with softmax resulted in a 97.68% accuracy, while model using sigmoid resulted in 97.59%. Though both had similar accuracy percentages and the sigmoid was minimally higher, using a softmax formulation as an output layer is more ideal, as the softmax distributes the outputs as a total of probabilities of all outputs whereas sigmoid is usually considered for binary classification. Since we are trying to identify

the handwritten number into a classification from 0-9, it is more reasonable to use softmax as an output layer. It is also good to note that though both don't seem to overfit their data.

**Also note that all the models use L2 Kernel Regularizer with a Dropout layer of rate 0.2 unless stated otherwise as it is the regularizer that deemed most fit for the model. The learning rate is also not specified as the model will automatically yield the most fit learning rate as it is training.



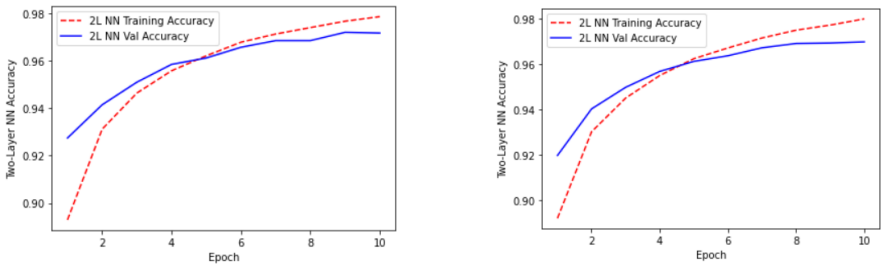
(a) ReLU with Softmax (b) ReLU with Sigmoid

Figure 1: ReLU with Second Layer

2.1.2 Tanh Activation Function

The second batch of outcomings used a different first dense layer, tanh also with 128 and 10 neurons, respectively. The accuracies of both networks were 97.24% on several runs. However, even though both models don't overfit, it is clear that the model with Sigmoid as a second layer has higher overfitting than the softmax model as there is greater variance in model (b). In other-

words, in comparison between both models, the tanh with softmax model is a better choice; however, since it has a lower accuarcy than the ReLU model, the ReLU model with softmax is more favorable.



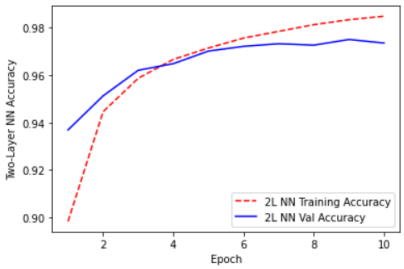
(a) Tanh with Softmax

(b) Tanh with Sigmoid

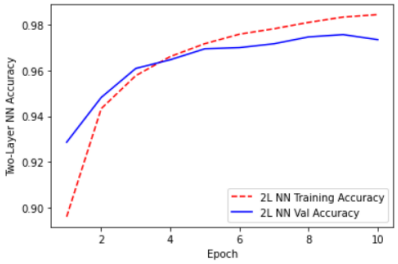
Figure 2: Tanh with Second Layer

2.2 Regularizers

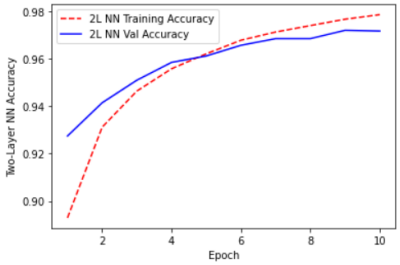
According to the graphs, with L2 kernel regularization and dropout, the model seems to have the highest accuracy through multiple runs, with accuracies 97.51% for L2 Regularizer without Dropout, 97.36% for L1 Regularizer, and 97.51% for L1L2 Regularizer, and 97.68% for adding Dropout Regularization layer with L2 regularizer. It is noted through the graphs that the model with dropout has the least overfitting as the number of epochs increase.



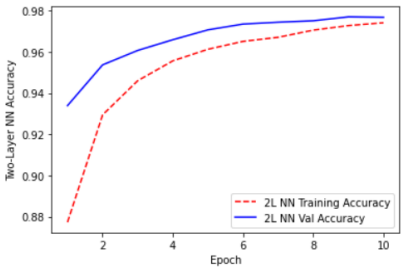
(a) L1 Kernel Regularizer



(b) L2 Kernel Regularizer



(c) L1L2 Regularizer

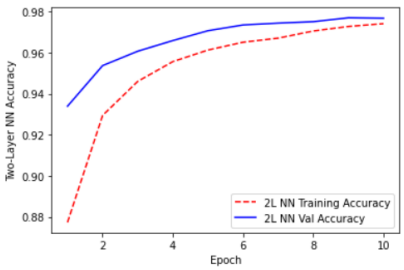


(d) Dropout Regularizer

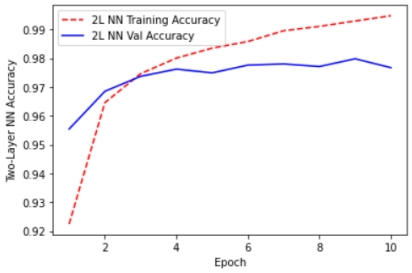
Figure 3: ReLU with Softmax Layer

2.3 Optimizers

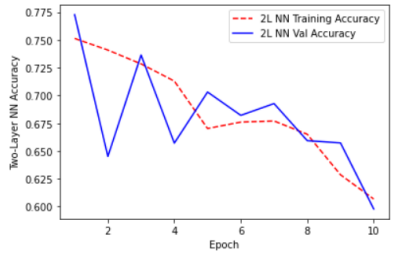
Throughout the optimizers used, the changes in accuracies were drastic. The following were the optimization techniques used and the their accuracy levels. Stochastic Gradient Descent, which was the optimizer used in the best model retrieved, with accuracy 97.68%, Adam with 59.78%, and another 59.78% for RMSprop Optimizer. It is evident that the SGD produces the best results. The above models all had a standard rate of 1e-1. When changing the learning rate on the best model (SGD) from 1e-1 to 3e-1 the model tend to have a slightly lower accuracy (97.67%) and greater overfitting. It also clear that the Adam and RMSProp Optimizers seem to cause the models to underfit as the models accuracies are very low and are haphazard.



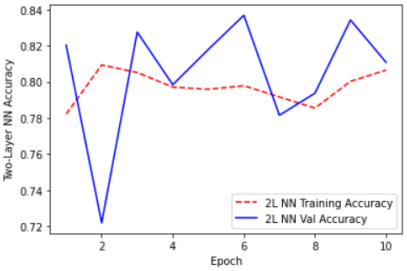
(a) SGD Optimizer



(b) SGD Optimizer with 3e-3 Learning Rate



(c) Adam Optimizer



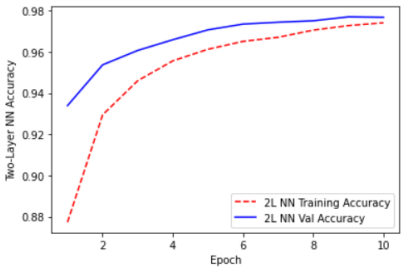
(d) RMSprop Optimizer

Figure 4: ReLU with Softmax Layer

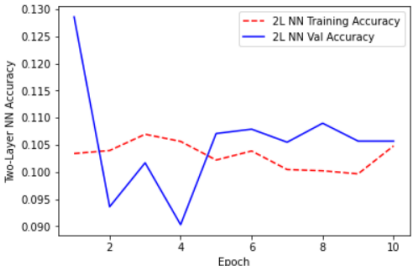
2.4 Loss Functions

Two Loss Functions were tested: a. Sparse Categorical Cross-Entropy and b. Poisson Loss Functions. The accuracies of the graphs were as follows: graph (a) an accuracy of 97.68% and graph (b) an accuracy of 10.5%. The graph of the validation accuracy has no semantics in regards to the training, which is likely to underfit. Other loss functions such as the MSE and the cosine similarity couldn't be tested as they are regression methods and the output of the problem is probabilistic. Moreover, other probabilistic functions such as KL Divergence and the Binary Cross Entropy as they both are measures

of binary output, which is not the case in this problem.



(a) Sparse Categorical Cross-Entropy Loss Function



(b) Poisson Loss Function

Figure 5: ReLU with Softmax Layer

3 Part II: Convolutional Neural Networks

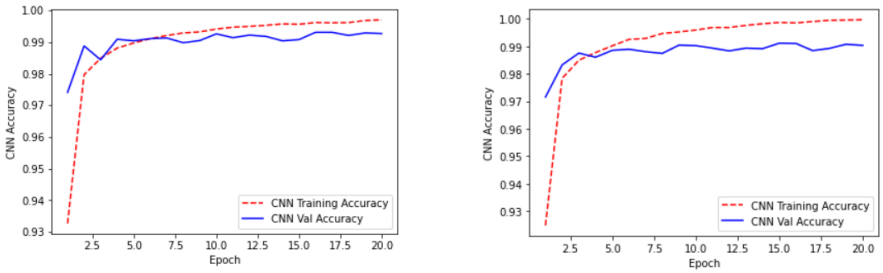
3.1 Activation Models

3.1.1 ReLU Activation Function

The below accuracy vs. Epoch graphs are modeled using a ReLU first dense layer with 128 neurons in the first layer and 10 in the second as illustrated in the notebook. The accuracy of the model with softmax resulted in a 99.27% accuracy, while model using sigmoid resulted in 99.03%. Though both had similar accuracy percentages and the sigmoid was minimally lower, again using a softmax formulation as an output layer is more ideal, as the softmax distributes the outputs as a total of probabilities of all outputs whereas sigmoid is usually considered for binary classification. Since we are trying to identify the handwritten number into a classification from 0-9, it is more reasonable to use softmax as an output layer. It is also good to note that though both don't seem to overfit their data, though the sigmoid model seems to have higher overfitting.

**Also note that all the models use L2 Kernel Regularizer with Dropout Layer and rate (0.2), 24 filters, 3x3 kernel in the first layers, 36 filters and 3x3 kernel in the second layer, and all pool sizes were 2x2 unless stated otherwise as this combination has produced the higher accuracy that deemed most fit

for the model.

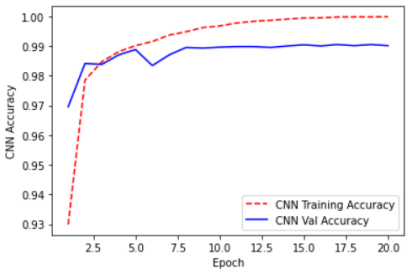


(a) ReLU with Softmax (b) ReLU with Sigmoid

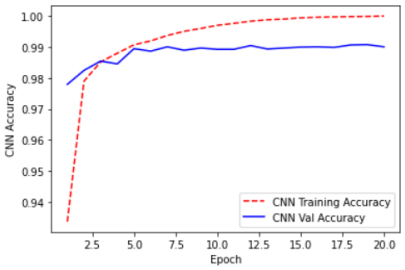
Figure 6: ReLU with Second Layer

3.1.2 Tanh Activation Function

The second batch of outcomings used a different first dense layer, tanh also with 128 and 10 neurons, respectively. The accuracies of graph (a) was 99.01% and graph (b) was 99.00% on several runs. However, even though both models don't overfit, it is clear that the model with sigmoid as a second layer has slightly higher overfitting than in model (b). In other words, in comparison between both models, the tanh with softmax model is a better choice for the softmax's ability to distribute probabilities over multiple classes; however, since it has a lower accuracy than the ReLU model, the ReLU model with softmax is more favorable.



(a) Tanh with Softmax

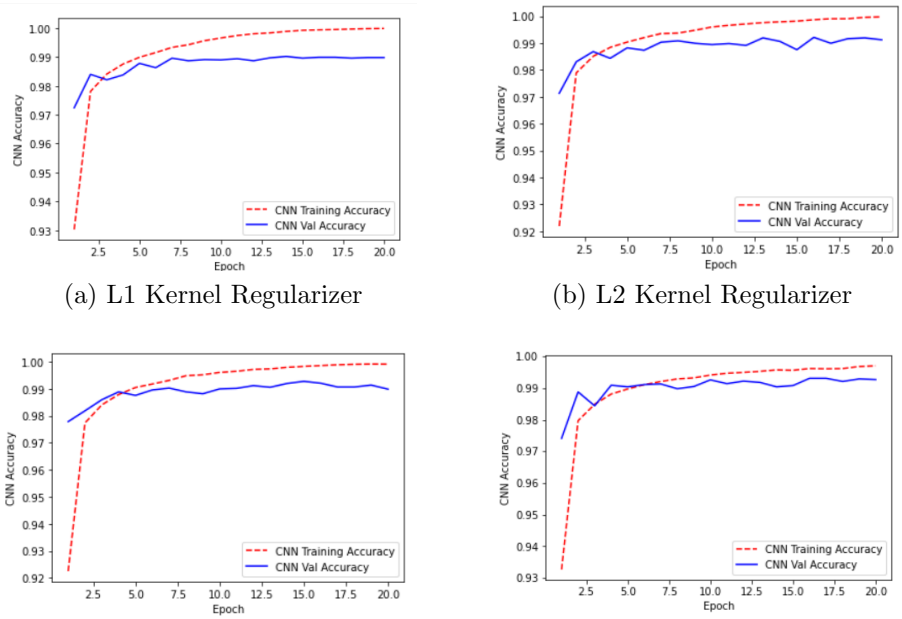


(b) Tanh with Sigmoid

Figure 7: Tanh with Second Layer

3.2 Regularizers

According to the graphs, with L2 kernel regularization, the model seems to have the highest accuracy through multiple runs, with accuracies 99.12% for L2 Regularizer without Dropout, 98.98% for L1 Regularizer, and 98.98% for L1L2 Regularizer, and 99.27% for adding Dropout Regularization layer with L2 regularizer. It is noted through the graphs that the model with dropout has the least overfitting as the number of epochs increase.



(a) L1 Kernel Regularizer

(b) L2 Kernel Regularizer

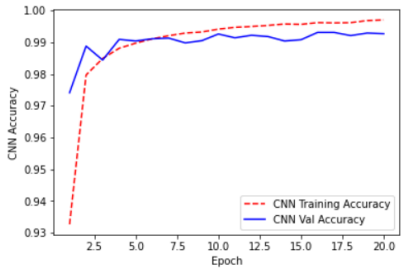
(c) L1L2 Regularizer

(d) Dropout Regularizer

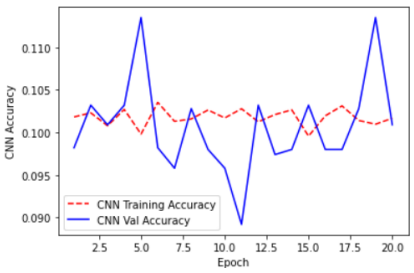
Figure 8: ReLU with Softmax Layer

3.3 Filter Sizes

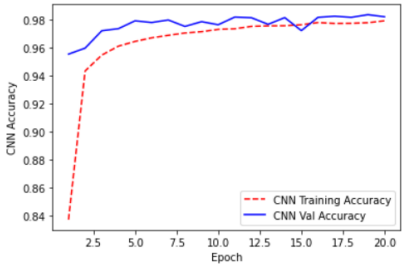
As shown in the graphs, the requested amount of filters yielded to the most accurate model with 99.27% accuracy, whereas, after changing the filters to 36 and 48 in the first and second layers, respectively, the model has become underfit with 10.09% accuracy. However, when changing once again the stride length to match the number of filters, the accuracy percentage has inclined to 98.20%, which is a much better accuracy from keeping the stride length as default.



(a) 24 and 36 filters in first and second layers



(b) CNN Network with 36 and 48 filters in first and second layers with no stride length specified

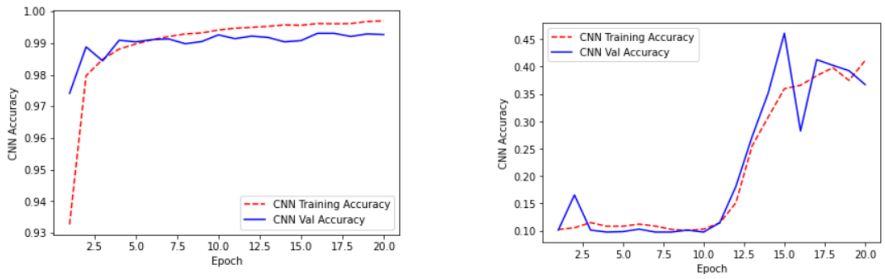


(c) CNN Network with 36 and 48 filters in first and second layers with stride = (3,3);

Figure 9: ReLU with Softmax

3.4 Pooling Layers

The model seemed most ideal when the pooling is standardized. As shown in graph(a); allover 2x2 pooling resulted in an 99.27% accuracy, while using different pools have resulted in graph(b) with an accuracy of 36.70% which is an underfit to the data.



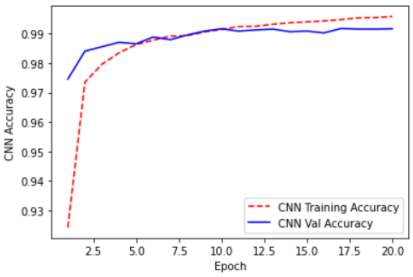
(a) 2x2 Pooling All Over L1 (b) 3x3 Pooling then 2x2 Pooling

Figure 10: ReLU with Softmax

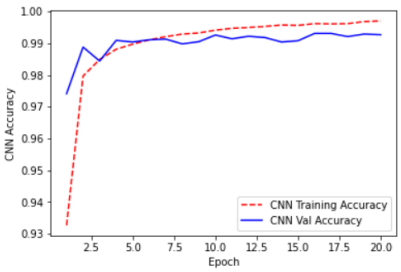
3.5 Optimizers

Throughout the optimizers used, the changes in accuracies were drastic. The following were the optimization techniques used and the their accuracy levels. Stochastic Gradient Descent with 3e-1 as a learning rate, which was the optimizer used in the best model retrieved, with accuracy 99.27%, Adam with 9.80%, and another 10.32% for RMSprop Optimizer. It is evident that the SGD produces the best results. The above models all had a standard rate of 3e-1. When changing the learning rate on the best model (SGD)

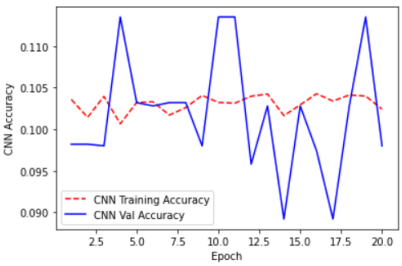
from 3e-1 to 1e-1 the model tend to have a slightly lower accuracy (99.17%) and greater overfitting. It is also evident that the Adam and RMSProp are underfitting to the data and have very low accuracies.



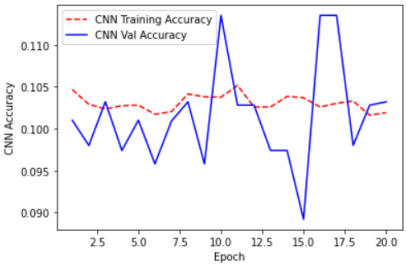
(a) SGD Optimizer with 1e-1



(b) SGD Optimizer with 3e-1 Learning Rate



(c) Adam Optimizer

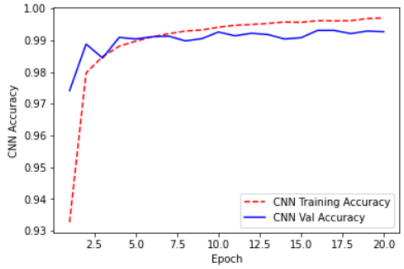


(d) RMSprop Optimizer

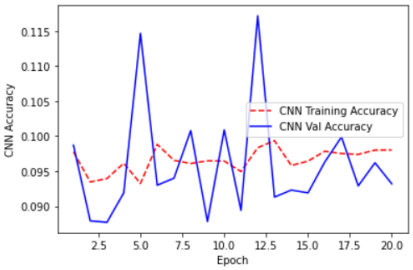
Figure 11: ReLU with Softmax Layer

3.6 Loss Functions

Two Loss Functions were tested: a. Sparse Categorical Cross-Entropy and b. Poisson Loss Functions. The accuracies of the graphs were as follows: graph (a) an accuracy of 99.27% and graph (b) an accuracy of 9.32%. The graph of the validation accuracy has no semantics in regards to the training, which is likely to underfit. Other loss functions such as the MSE and the cosine similarity couldn't be tested as they are regression methods and the output of the problem is probabilistic. Moreover, other probabilistic functions such as KL Divergence and the Binary Cross Entropy as they both are measures of binary output, which is not the case in this problem.



(a) Sparse Categorical Cross-Entropy Loss Function



(b) Poisson Loss Function

Figure 12: ReLU with Softmax Layer

3.7 Machine Learning Interpretability

What does the plot show in each entry? What can you conclude from the plot? Is the model focusing on meaningful features? Why? Why not? Are there features in certain digits that confuse the model?

The plot shows the handwritten digit and its corresponding value. For instance, the handwritten number 2, corresponds to its value 2, which is the highlighted value in red next to the hand written number. The model is indeed focusing on meaningful features because the aim of this problem is to be able to identify handwritten numbers which the SHAP is able to do based on the values in red. There might be certain outliers as to the SHAP; however, the model seems accuarate although there are a few light blue values on certain numbers such as 5 taking the value of 3, which can confuse the model as they look alike.

Figure 13: MLI Plot