**CS 390 – FALL 2021**

**PURDUE UNIVERSITY**

**Lab 2**

**Name:** Nalin Ahuja

**Login**: ahuja15

**PUID:** 0030482364

October 8, 2021

**Student Information**

* Name: Nalin Ahuja
* Email: [ahuja15@purdue.edu](mailto:ahuja15@purdue.edu)

**Project Information**

* Git Repository: [Link](https://github.com/nalinahuja/CS390-NIP/tree/main/lab2)

**Resources****Used**

* CS 390-NIP Lecture Slides: [Lecture 5](https://docs.google.com/presentation/d/e/2PACX-1vQEEb5fOl9V6OmP__M2jDJCjevKhcyhNM-CX2_JK-8w3H5XhnMB0hHQFfE4OeGebSfuQZsBcrol-YHb/pub?start=false&loop=false&delayms=3000#slide=id.p), [Lecture 6](https://docs.google.com/presentation/d/e/2PACX-1vRiPhLmER7I6obbtf1mTkOUzEHJuABgLkoIe1GXkRtxNPDCw-W0H3szB7juHNu4G9lkGvfxS1B3S2ed/pub?start=false&loop=false&delayms=3000#slide=id.p)
* API Documentation: [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/)
* Batch Normalization: [Link](https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/)

**Lab Milestones**

* TensorFlow Artificial Neural Network [5]
  + Fully functioning artificial neural network using Keras and TensorFlow.
    - Can be added to pipeline when .
* TensorFlow Convolutional Neural Network [10]
  + Fully functioning convolutional neural network using Keras and TensorFlow.
    - Can be added to pipeline when .
* Accuracy [58]
  + Fully functioning convolutional neural network using Keras and TensorFlow.
    - Achieves 99.37% classification accuracy for mnist\_d dataset [10].
    - Achieves 92.11% classification accuracy for mnist\_f dataset [10].
    - Achieves 74.71% classification accuracy for cifar\_10 dataset [11].
    - Achieves 55.21% classification accuracy for cifar\_100\_c dataset [12].
    - Achieves 41.97% classification accuracy for cifar\_100\_f dataset [12 + 3 EC].
* Pipeline [10]
  + Pipeline can be configured to use the cifar\_10, cifar\_100\_c, and cifar\_100\_f datasets [6].
  + Bar plots for ANN and CNN accuracies for all datasets can be found [here](https://github.com/nalinahuja/CS390-NIP/tree/main/lab2/plots) [4].

**Lab Questions**

*How is a CNN superior to a standard ANN for image processing?*

CNNs are superior to standard ANNs because they can preserve the spatial relationship between pixels using techniques like convolutions to produce sub-tensors which are learned by the CNN. This means that CNNs can extract more useful information from images compared to ANNs.

*Why do we sometimes use pooling in CNNs?*

Pooling is a technique used in CNNs after convolutional layers to reduce the dimensionality of an image tensor by removing unnecessary information from the input tensor. This helps make the image tensor easier to process for layers downstream in the network.

*Why do you think the cifar* *datasets are harder than mnist?*

The *cifar* dataset contains inputs with more dimensions and more classes than the *mnist* dataset. Due to the higher complexity of the *cifar*

**CNN Model Accuracy Optimization**

* Dropout layers were added to the network structure between dense layers to reduce overfitting and improve classification accuracy.
* Model complexity was progressively increased to promote higher accuracy. This was done by using more convolutional layers and increasing dense layer sizes.
* Batch normalization was added to the network structure after each pooling layer to encourage the model to converge faster by standardizing the inputs into a layer for each mini batch.

**CNN Hyperparameter Values**

* Epochs: 10
* Dropout: 0.20
* Learning Rate: 0.001
* Optimizer Function: Adam
* Max Pooling Layer Size: [2, 2]
* Convolutional Layer Kernel Size: [3, 3]
* Loss Function: Categorical Cross Entropy
* Activation Function: ReLU for all hidden layers, SoftMax for the output layer
* Convolutional Layer Structure
* Dense Layer Structure

**ANN Model Information**

*ANN Model Accuracy Plot*

Chart, bar chart

Description automatically generated

*ANN Model Metrics*

Training metrics and classification figures for the ANN model can be viewed for each dataset [here](https://github.com/nalinahuja/CS390-NIP/tree/main/lab2/out/ann).

**CNN Model Information**

*CNN Model Accuracy Plot*

*Chart, bar chart

Description automatically generated*

*CNN Model Metrics*

Training metrics and classification figures for the CNN model can be viewed for each dataset [here](https://github.com/nalinahuja/CS390-NIP/tree/main/lab2/out/cnn).