**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/lab1)

*Resources Used*

* CS 390-NIP Lecture Slides: [Lecture 2](https://docs.google.com/presentation/d/e/2PACX-1vSSsqahjBfVnIw5tnGWIrzH-9Wn3Gg-IsgF0CkIOezip6KW9K4SH0BjlvYBfOwVGrZ0Ilp-qmf__V3k/pub?start=false&loop=false&delayms=6000#slide=id.p), [Lecture 3](https://docs.google.com/presentation/d/e/2PACX-1vTvpapjxL7BxZ3XDseofyBqViHy2UYB82INz1fjYncbrkNooWjmQOwmypqPF15VskYC_dJccgDV6pPC/pub?start=false&loop=false&delayms=3000#slide=id.p)
* API Documentation: [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/)
* Backpropagation Example: [Link](https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/)
* Confusion Matrix Guide: [Link](https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826)

*Lab Milestones*

* Custom Neural Network
  + Fully functioning two-layer neural network.
    - Can be added to pipeline when .
    - 88.17% accurate with mini batches of size 100.
  + Implemented sigmoid and sigmoid derivative functions.
  + Implemented train function with correct backpropagation.
* TensorFlow Neural Network
  + Fully functioning two-layer neural network using Keras and TensorFlow.
    - Can be added to pipeline when .
    - 98.23% accurate with Adam optimizer.
* Pipeline
  + Added input value range normalization from to .
  + Multiclass confusion matrix and labeled F1 scores printed along with accuracy.

*Lab Summary*

* Custom Neural Network

Implemented this custom neural network and its various features after understanding the intended structure of the model. This neural network uses the sigmoid activation function for all neurons and the backpropagation function has been correctly implemented after gaining a thorough understanding from online resources. The output of the neural network is one-hot encoded after the model prediction is received by the caller so that the output of the model shows a clear, singular prediction. The NumPy library is extensively used in the neural network to decrease the time mathematical operations require when executing code.

This neural network was trained over 15 epochs at a learning rate of 0.001, achieving 88.17% accuracy with mini batches of size 100. The model outputs accuracy metrics, a confusion matrix where the X dimension is the true class, and the Y dimension is the predicted class and labeled F1 scores for each class.

* TensorFlow Neural Network

Implemented this custom neural network and its various features after understanding the intended structure of the model. The neural network uses the ReLU activation function for the dense hidden layer with 512 neurons and the SoftMax activation function for the dense output layer with 10 neurons. This model configuration was determined after reviewing the class slides on basic classification. The categorical cross entropy loss function was used for this model since it is best suited for categorical data classification and the Adam optimizer function was used for gradient decent. The output of the neural network is one-hot encoded after the model prediction is received by the caller so that the output of the model shows a clear, singular prediction.

This neural network was trained over 15 epochs at a learning rate of 0.001, achieving 98.23% accuracy with the Adam optimizer. The model outputs accuracy metrics, a confusion matrix where the X dimension is the true class, and the Y dimension is the predicted class and labeled F1 scores for each class.

*Program Outputs*

* Guesser Algorithm

Calendar

Description automatically generated

* Custom Neural Network

Calendar

Description automatically generated­

* TensorFlow Neural Network

Calendar

Description automatically generated with medium confidence