CS390 FALL 2021

Lab 2

Name: Divay Gupta

Email: gupta576@purdue.edu

Repository Link: black-tul1p/CS390

Completed Lab Sections:

- **Custom Neural Net**
  - Accuracy – 93.23%
  - Implemented the sigmoid and sigmoid derivative functions.
  - Implemented the train function using backpropagation properly.
  - Fully functioning 2-layer neural net with minibatches of size 100.
- **TensorFlow Neural Net**
  - Accuracy – 98.27%
  - Implemented a 2-layer neural net using Keras
  - Used the sigmoid and SoftMax activation functions, in that order
  - Used the Adam optimizer
- **Pipeline**
  - Normalized image data values from 0-255 to a range of 0.0-1.0
  - Implemented a F1 score table and a confusion matrix with accuracy values

Resources Used:

- **Lecture Slides**
  - Slides 5 – Convolutional Neural Networks, Pooling, Optimization
  - Slides 6 – Convolutional Neural Networks 2, CNNS using TensorFlow and Keras
- **API Documentation Links**
  - TensorFlow
  - Keras
- **Miscellaneous Links**
  - Using TensorFlow's Batch Normalization Correctly
  - Building a Convolutional Neural Network (CNN) in Keras

Lab Implementation Summary:

- **TensorFlow ANN (*tf_net*)**
  - Implemented using TensorFlow and Keras
  - Fully functional [5]
  - Accuracy:

MNIST_d	:	95.20%	
MNIST_f	:	82.58%	
CIFAR_10	:	10.01%	
CIFAR_100_C	:	05.02%	
CIFAR_100_F	:	01.00%	

- **TensorFlow CNN (*tf_conv*)**
  - Implemented using TensorFlow and Keras
  - Fully functional [10]
  - Accuracy:

MNIST_d	:	99.35%	[10]
MNIST_f	:	92.81%	[10]
CIFAR_10	:	74.59%	[11]
CIFAR_100_C	:	52.96%	[12]
CIFAR_100_F	:	39.71%	[12]

- **Pipeline**
  - Can use the TensorFlow ANN by setting ALGORITHM to "*tf_net*"
  - Can use the TensorFlow CNN by setting ALGORITHM to "*tf_conv*"
  - Can train the neural nets on the following datasets:
    - MNIST_d
    - MNIST_f
    - CIFAR_10 [2]
    - CIFAR_100_C [2]
    - CIFAR_100_F [2]

Questions:

1. How is a CNN superior to standard ANNs for image processing?

CNNs are superior to standard ANNs since they are able to preserve the spatial relationships between the pixels using a set of convolution operations like max pooling, which produce sub-tensors which are then learned. Thanks to this, CNNs learn recurrent patterns in the images better than ANNs.

2. Why do we sometimes use pooling in CNNs?

Pooling is a convolutional operation sometimes used in CNNs to reduce the dimensionality of the image tensor by only keeping relevant information, reducing the number of parameters to be learned. This reduces the processing work of the downstream layers, reducing network complexity.

3. Why do you think the CIFAR datasets are harder than MNIST?

CIFAR datasets contain more types (classes) of images and the images have a higher dimensionality than MNIST datasets, making it much more complex. The larger variation and complexity in input in CIFAR datasets also makes it harder for neural nets to learn the patterns in the dataset accurately.

CNN Accuracy Optimization:

- Added batch normalization to the network model after pooling layers to standardize the inputs to each layer for each mini-batch, which helps reduce the number of training epochs required for training.
- An option of dropout layers was added to the network model to improve accuracy reduce overfitting
- Added more convolutional layers and larger dense layers to increase network model complexity to be able to handle the complex CIFAR datasets.

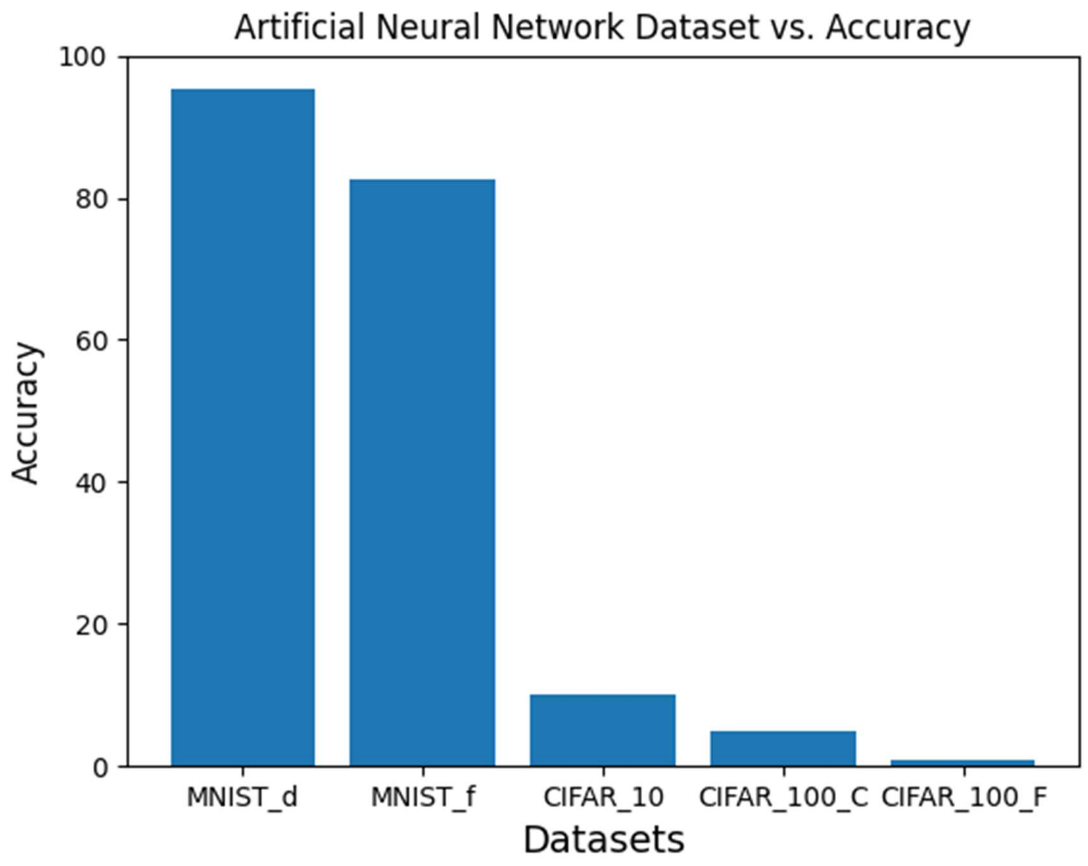
Hyperparameters:

- Epochs: 10
- Learning rate: 0.001
- Dropout rate: 0.20
- Optimizer Function: Adam

- Loss Function: Categorical Cross Entropy
- Max Pooling Kernel Size: [2, 2]
- Convolutional Layer Kernel Size: [3, 3]
- Activation Functions: ReLU, SoftMax (output only)

ANN Metrics

Accuracy Plot

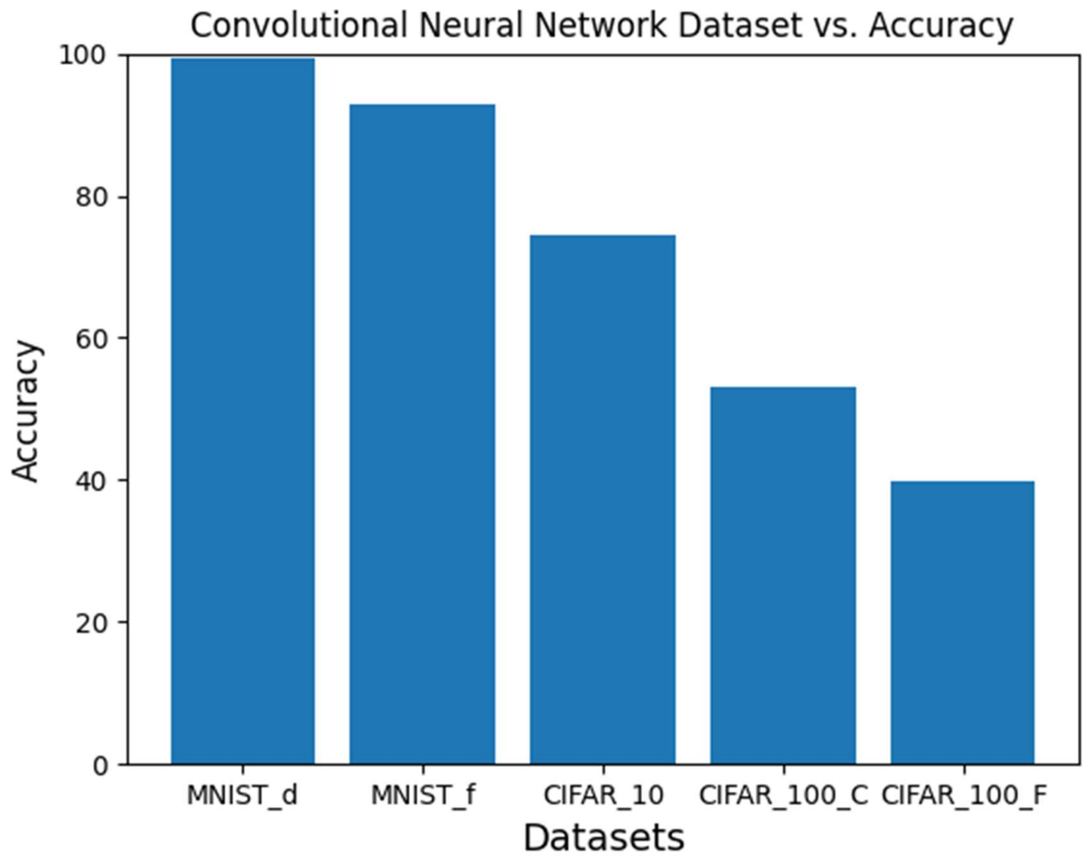


Model and Dataset-Specific Metrics

Can be found here

CNN Metrics

Accuracy Plot



Model and Dataset-Specific Metrics

Can be found here