Deep Learning Final Project

Nicolas Rubert
Rutgers University
School of Engineering
Introduction To Deep Learning
ECE 435
Nar126@rutgers.edu

ABSTRACT
Data will be loaded into four distinct convolutional neural networks (CNN) where each configuration will be evaluated and ranked using time and accuracy as the parameters. This will evaluation is done using the Pytorch Library.

CCS Concepts
• CCS → Computing methodologies → Machine learning → Deep Learning → LeNet-5 Model

Keywords
Machine Learning; Deep Learning; Python; Pytorch; LeNET-5; Batch Normalization; Dropout; Convolutional Neural Network;

1. Introduction
The objective of this project is to evaluate which of the four conventual neural networks (CNN) is the fastest and most accurate. The four configurations of the convolutional neural networks are as followed:

1. Fully connected with dropout, convolutional layers with batch normalization
2. Fully connected with dropout, convolutional layers without batch normalization
3. Fully connected without dropout, convolutional layers with batch normalization
4. Fully connected without dropout, convolutional layers without batch normalization

2. Dataset

Before applying the machine learning algorithms to predict which applicants are approved for the loan, we must first understand the data. The data will provide the necessary values needed for the system to calculate the outcome.

2.1 MNIST Dataset

The data that will be used as to test our dataset is from the Modify National Institute of Standards and Technology or MNIST. The dataset is which includes a massive database with handwritten numbers that is used to validate machine learning models. **Figure 1** below shows the dataset visualized. The dataset contains 50,000 images. This amount is sufficient to learn and test the performance of each LeNet-5 configuration.

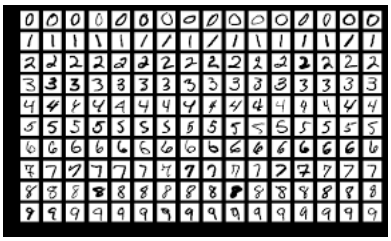


Figure 1

3. Model and Configurations

3.1 LeNet-5

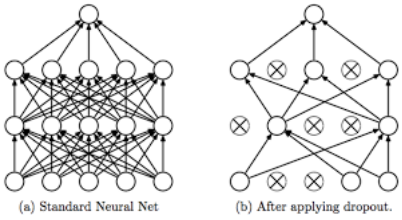
The LENET-5 learning model is a Convolutional Neural Network (CNN) model created in the late 90's. The five in the name of the model refers to the five layers that has learnable parameters. There are 3 convolutional layers with two fully connected layers. There is also a layer used for classification layer that takes the images and puts them in their corresponding category.

3.2 Batch Normalization

Batch normalization is a method used to speed up the process and stabilize the data when being trained and tested. This is done by normalizing batches of data. Normalizing refers to standardizing data to a similar scale. This can be done by adjusting data such as averaging through propagation.

3.3 Dropout

Dropout refers to ignoring certain nodes in the neural network. This normalization practice does it by ignoring certain nodes randomly. Although this seems counter intuitive using dropout nodes helps combat over fitting by using.



Dropout Visualized

3.4 FC with Dropout, Convolutional Layers with Batch Normalization

The first configuration consists of the standard Lenet-5 learning model with an addition of dropout layers and batch normalization layers. The dropout layers are included with the fully connected layers. Three additional batch normalization layers were included with the convolutional layers.

3.5 FC with Dropout, Convolutional Layers without Batch Normalization

Like before, this configuration is the same but without the batch normalization layers. This means that model will not scale the data before training and testing which could lead to a longer time to achieve results.

3.6 FC without Dropout, Convolutional Layers with Batch Normalization

This third configuration contains the convolutional layers with batch normalization. It does not include a dropout layer after the fully connected layer.

3.7 FC without Dropout, Convolutional Layers without Batch Normalization

The final configuration is having fully connected layers without dropout layers. The convolutional layers do not include batch normalization layers as well. By having neither dropout layer to prevent overfitting and the batch normalization layers to scale the data this could be the worst performing configuration.

4. Method Description

All models are using the same dataset along with the same number of images to train/test. This is done to ensure there is no bias. To download the dataset code provided from previous homework assignments will be used. Once the database is downloaded there will be code to prep the data to be used for testing and training.

4.1 Experimental Procedure

To test each model configuration, they all need to have the same training and testing data and parameters. As established the dataset that will be used is the one provided by MNIST. The learning rate is 0.5 steps with 50 epochs with a batch size of 128. Two performance metrics will be used to compare the models. The first being accuracy and the second time. The best model configuration will include a high accuracy in testing in the lowest amount of time. The results collected will be recorded on **Table 1.**

5. Results and Analysis

5.1 Results

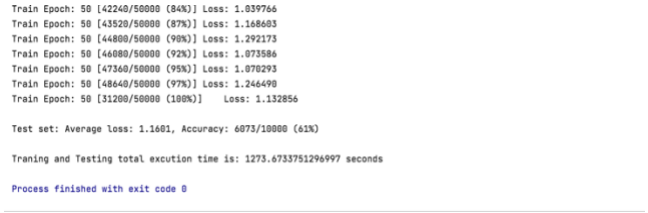
Table 1

Algorithm	Accuracy	Time
Fully connected with dropout, convolutional layers with batch normalization	**69 %**	**1316 seconds**
Fully connected with dropout, convolutional layers without batch normalization	**60 %**	**1224 seconds**
Fully connected without dropout, convolutional layers with batch normalization	**70 %**	**1418 seconds**
Fully connected without dropout, convolutional layers without batch normalization	**61 %**	**1336 seconds**

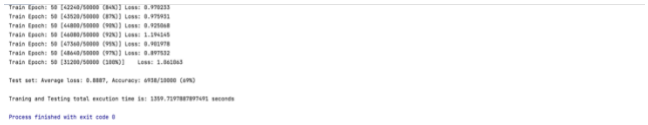
5.2 Analysis

The worst configuration was the fully connected with dropout layer and the convolutional layers without batch normalization. Although it was the fastest due to the dropout layer the lack of normalization of the data provides by the batch normalization played a rule in how accurate this configuration is.

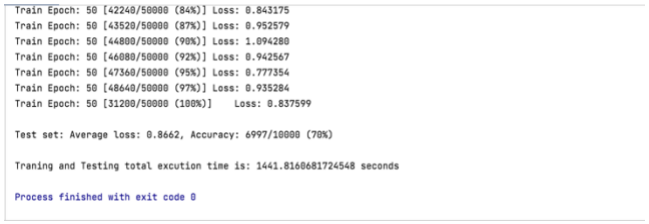
The last configuration in **Table 1** which includes the fully connected layers without dropout and batch normalization was the second worst performing configuration. It took more time than the worst one for only a marginal increase in accuracy.



The second-best performing configuration contains the fully connected layers with dropout and convolutional layers with batch normalization. This configuration is just marginally worse when it comes to training and testing accuracy but completes it in over a minute and a half quicker than the most accurate configuration. Overall, the this should be the most optimal configuration for this type of analysis.



The best performing in terms of accuracy is the fully connected layers without dropout layers and the convolutional layers with batch normalization. The batch normalization helps standardize the data. By having no dropout layer each “neuron” is used to help classify the images. This means that every layer in the model is being used to train and test.



6. CONCLUSION

The model configuration that would provide great performance while using less resources is the fully connected layers with a dropout layer and the convolutional layers with batch normalization. Although not the most accurate configuration it provided marginally worse accuracy but was quicker by almost two minutes. This configuration leverages the dropout layer to prevent overfitting and batch normalization for keeping having the data in a common scale.

7. REFERENCES

[1] MNIST Database*: http://yann.lecun.com/exdb/mnist/*

[2] LeNet-5: http://yann.lecun.com/exdb/lenet/

[3] Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov; https://jmlr.org/papers/v15/

[4] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Sergey Ioffe Christian Szegedy* : https://proceedings.mlr.press/v37/ioffe15.pdf

[5] Pytorch Tutorial:https://pytorch.org/tutorials/

[6] Stanford CS231: http://cs231n.stanford.edu/