ECE 8870 Project 1

Qazi Zarif Ul Islam

Pawprint: qzic2d

Neural Networks

University of Missouri

University of Missouri-columbia

02/29/2024

1 Technical Description

The project described henceforth is one that aims to understand the working 'principle' of a neural network. In principle, a neural network has several core components; the inputs and outputs, the computational units known as 'neurons', the weights, the activation function, an optimizer and an error function. A neural network's components work together to form composite functions that ultimately combine to generate a final, function. Thus objective of 'understanding' the working principle of a neural network implies the attempt to understand how the core components behave in different configurations. With this aim in mind, this project contains two parts; Part 1- 'Sanity Check': In this part, 4 multi-layered perceptrons of different configurations were employed to classify a synthetic, linearly separable classes of 3 classes. Part 2- 'MLP vs CNN': In this part, the prediction performance of a convolutional neural net (CNN) and an MLP (Multi-layered perceptrons) were compared.

In the following sections will start with an explanation on the operation of a neural network. Then, activation functions and their derivatives shall be mathematically described and illustrated. Then, the algorithms used in this project for optimizing the neural networks shall be described, which are **Stochastic Gradient Descent (SGD)** and **Adaptive Momentum (Adam)**. Finally the so called 'learning task' with respect to the datasets in both parts will be defined.

1.1 Operation of Multi-layered Perceptrons

A more in-depth treatment of the operation is done in ...

1.2 Operation of Convolutional Neural Networks

1.3 Activation functions

1.4 Optimizers

1.4.1 Stochastic Gradient Descent

1.4.2 Adaptive Momentum

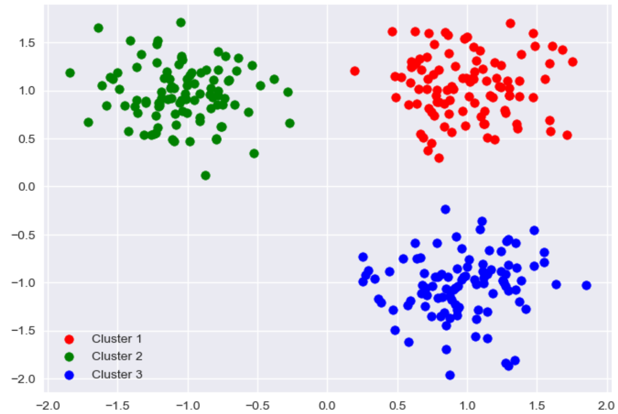

Figure 1: Synthetic data

2 Experiments & Results

The experiments were run in two Parts; Part 1- A classification task to classify a dataset containing 3 classes and Part 2- A classification task to classify the 'MNIST' dataset.

2.1 Training details

2.2 Part 1

In this portion of the report, we delve into the experiments conducted and the corresponding results obtained. Our primary objective was to ascertain the functionality of the developed program. A secondary goal involved investigating the effects of various activation functions: **Rectified Linear Unit (ReLU)**, **Sigmoid**, and **Hyperbolic Tangent (Tanh)** and the optimizers: **Stochastic Gradient Descent** and **Adaptive Momentum (Adam)**.

2.2.1 Data

The dataset utilized for this analysis, depicted in Figure 1, was specifically crafted to be linearly separable. This design choice was made to simplify the verification process, as creating a discriminant function for a linearly separable dataset is relatively straightforward for a Neural Network employing non-linear activation functions. To this end, multiple architectures were explored, each incorporating one of the three aforementioned activation functions

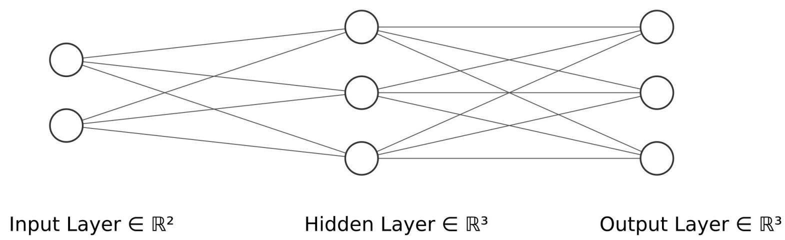

Figure 2: MLP with 1 hidden layer

across all hidden layers. This approach facilitated a comparative analysis of the activation functions' performance within the framework of Multi-Layer Perceptron (MLP) models.

2.2.2 Model Architectures

The 'base model' was a 1 layered model shown in figure 2 that used the sigmoid activation function across its hidden layers, . This architecture was designed with the minimum number of neurons required to create 3 functions in the hidden layer which were supposed to have separated the 3 classes.

3 more '2 (hidden) layered models' (figure 3) were created to compare firstly, the activation functions and secondly, to compare 2 layered models to the base 1 layered model.

2.2.3 Results

The epoch-accuracy (validation set) plots for all the architectures are shown in figure 4. The epoch-loss plots for the training and validation sets are shown in figure 5 and figure 6. The adam optimizer was able to optimize all models such that they achieved near 100% accuracy whereas SGD was only able to optimize the models with 2 layers using the Hyperbolic Tangent and the Rectilinear Unit function. The Adam optimizer optimized the models faster. If compared on the two models where the SGD achieved near 100% accuracy, the Adam optimizer optimized the Tanh model in 5 epochs and the ReLu model in 12 epochs where as the SGD took close to a 100 epochs for the ReLu model and about 55 epochs for the Tanh model. Remarkably, the 1 layered sigmoid model spiked in accuracy at the 62nd epoch and started decreasing (and kept decreasing until the end of the experiment) at the 70th epoch. During these times (62nd to 70th epoch), the average loss over all a batch never

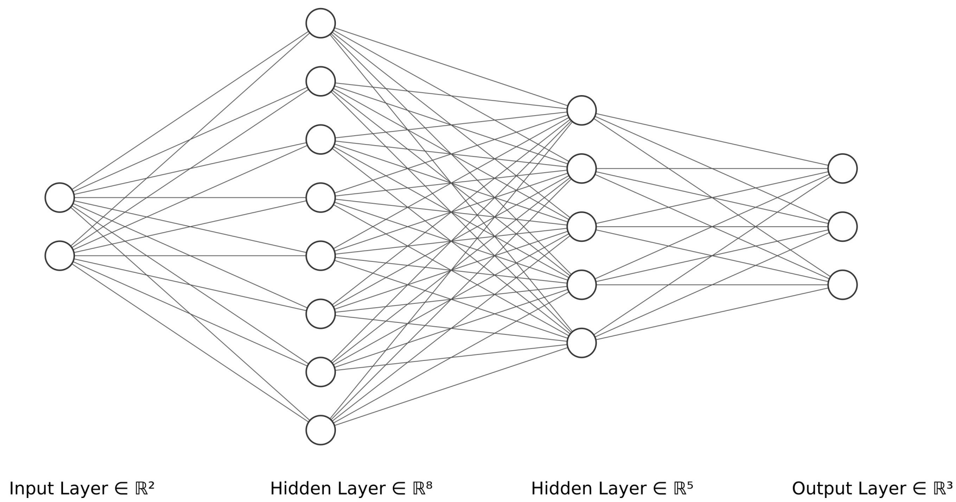

Figure 3: MLP with 2 hidden layers

changed for this model. In fact, the loss never changed throughout training.

The confusion matrices in figure 7 indicate that the sigmoid model with 1 hidden layer classified every datapoint as class 1 whereas the one with 2 hidden layers classified all datapoints belonging to class - 2 as class - 0. The confusion matrix in figure 8 represents the conclusion matrix for architectures, but optimized with the adam optimizer.

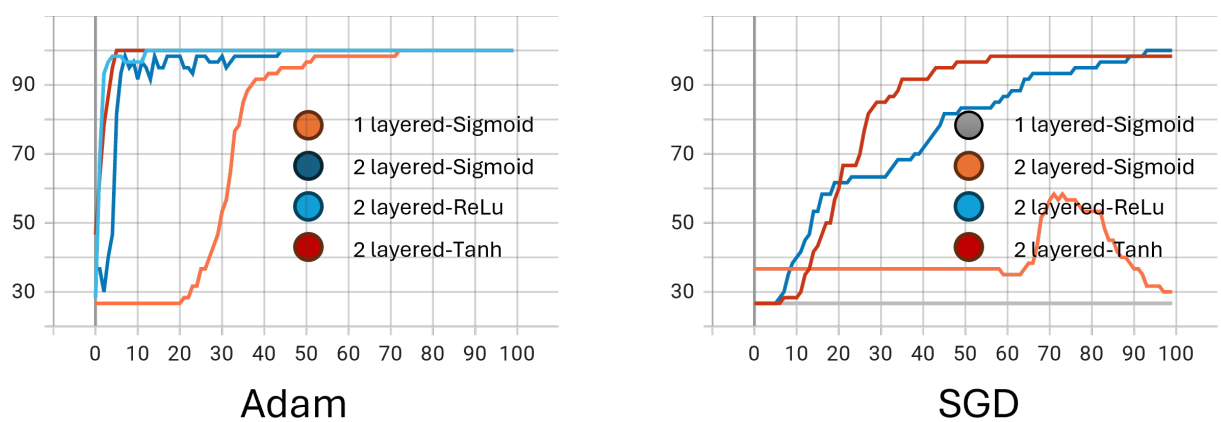

Figure 4: Accuracy on validation set

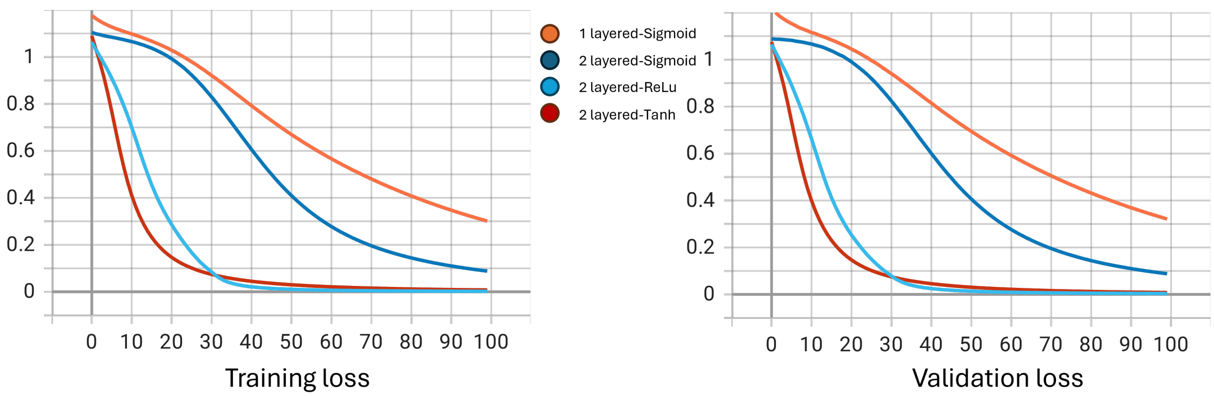

Figure 5: loss Adam

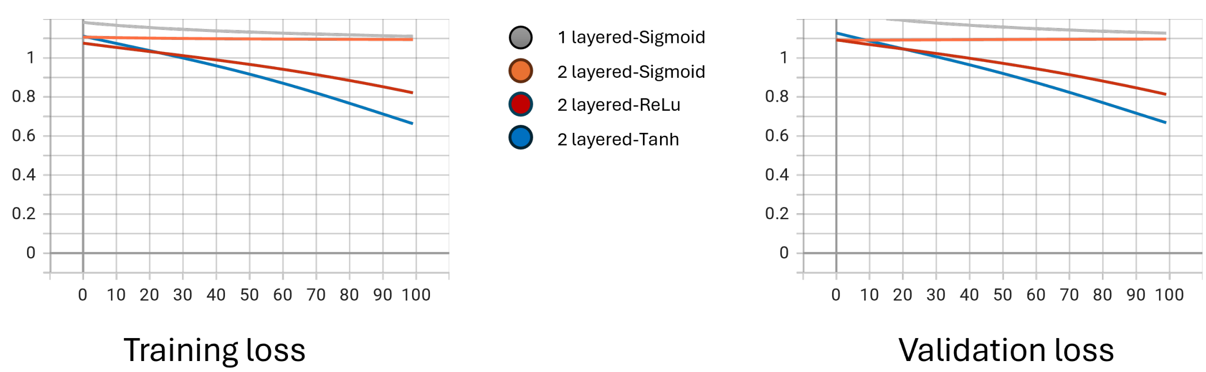

Figure 6: loss SGD

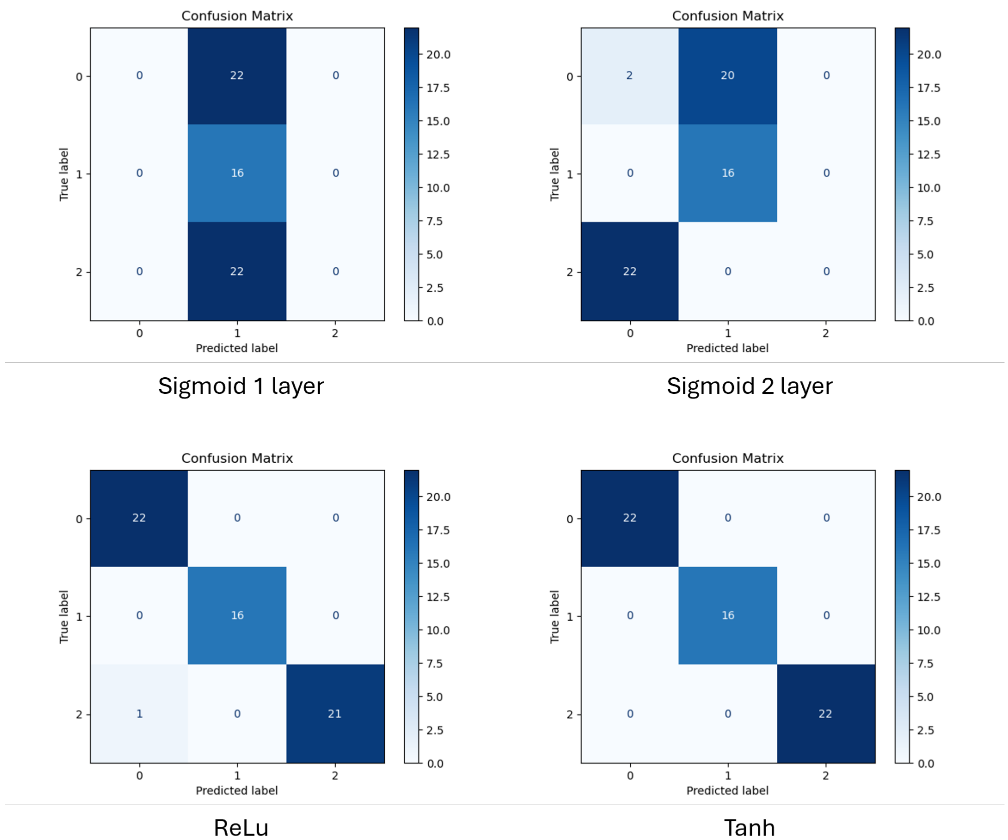

Figure 7: Confusion matrices for architectures optimized with **SGD**

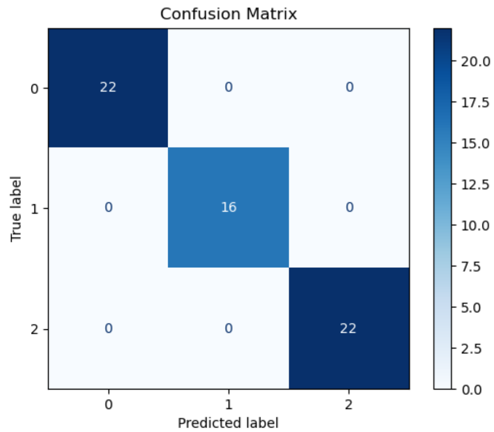

Figure 8: Confusion matrix for all architectures optimized with **Adam**

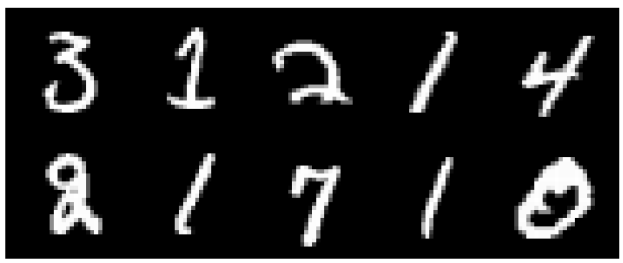

Figure 9: Samples from the MNIST dataset

2.3 Part 2

2.3.1 Data

The MNIST dataset (Samples of the dataset shown in figure 9 was used in this part of the project to compare the efficacy (loosely defined) and usefulness (loosely defined) of a CNN and an MLP. The Modified National Institute of Standards and Technology (MNIST) dataset is a widely used collection of 70,000 handwritten digits, serving as a cornerstone for research in machine learning and image processing. Comprised of 60,000 training images and 10,000 testing images, each labeled with its corresponding digit (0-9).

2.3.2 Model Architectures

A **Convolutional Neural Network** and a

2.3.3 Results

3 Conclusions and future work

The aim was not to attain high classification performance but rather to see the system in action.

Disregarded output classes sometimes, which benefitted performance. The merit of a classifier based on fuzzy sets lies within the interplay of the rules, membership functions and domain knowledge. The domain knowledge

The systems 1 and 2 had 3 fuzzy subsets and with the the better rule bases in system 3, a performance.

Thus, partitioning the overlapped region by having more subsets number and shape of fuzzy subsets over output domain could've been changed