CS F425 – DEEP LEARNING

Assignment 1

Sathvik Bhaskarpandit[1], Ruban S[2], and Deepti Kumar[3]

[1] 2019A7PS1200H
[2] 2019A7PS0097H
[3] 2018B5A70790H

Table of Contents

Deep Learning Models

We have implemented 24 different deep learning models for the assignment in Keras. Each is a neural network-based architecture with variations in layers, neurons, and activation functions. Early stopping is used to reduce training time.

We used Glorot uniform initialization, Cross-Entropy Loss, and Adam optimizer for training our models for all the below-mentioned networks.

The initial three networks are to test the effect of adding neurons onto the model (shallow network with one layer)

- Number of neurons = 16
- Number of neurons = 128
- Number of neurons = 1024

We look at the effect of activation functions on our model, namely Tanh, $\mathrm{Sigmoid}$, and ReLU. Note that these operate only on the hidden layers, and we always use softmax before the output layer for classification:

- ReLU
- Sigmoid
- Tanh

We also look at the effect of adding a layer to our network.

We finally looked into a few unconventional neural networks and compared their performance with the above networks.

- Number of neurons = 128, Number of layers = 5, Activation function = Sigmoid, Batch size=128
- Number of neurons = 1024, Number of layers = 5, Activation function = ReLU, Batch size = 128
- Number of neurons = 128, Number of layers = 1, Activation function = ReLU, Batch size = 64
- Number of neurons = 128, Number of layers = 1, Activation function = ReLU, Batch size = 256
- Number of neurons = 128, Number of layers = 1, Activation function = ReLU, Batch size = 512
- Number of neurons = 128, Number of layers = 1, Activation function = ReLU, Batch size = 1024

The Fashion MNIST dataset provided consists of 60,000 28×28 images of clothing items. To feed the images into our neural network, we flatten each image into a 784-dimensional vector.

Accuracy and Log Loss of the tested DL Models

Batch Size	Number of layers	Number of units	Activation Function	Train Accuracy	Test Accuracy	Train Log Loss	Test Log Loss
128	1	16	ReLU	0.792	0.768	0.553	0.653
		128		0.879	0.846	0.326	0.456
		1024		0.886	0.857	0.307	0.428
		16	Sigmoid	0.817	0.806	0.525	0.558
		128		0.825	0.814	0.505	0.520
		1024		0.826	0.813	0.476	0.518
		16	Tanh	0.791	0.782	0.608	0.634
		128		0.788	0.778	0.579	0.609
		1024		0.809	0.798	0.519	0.553
	2	16	ReLU	0.830	0.802	0.460	0.563
		128		0.902	0.869	0.262	0.394
		1024		0.908	0.874	0.250	0.378
		16	Sigmoid	0.822	0.809	0.513	0.548
		128		0.842	0.830	0.437	0.474
		1024		0.838	0.825	0.434	0.472
		16	Tanh	0.780	0.766	0.589	0.621
		128		0.804	0.793	0.532	0.564
		1024		0.777	0.768	0.602	0.622
	5	128	sigmoid	0.826	0.814	0.472	0.507
		1024	ReLU	0.917	0.884	0.228	0.348
64	1	128	ReLU	0.866	0.842	0.381	0.483
256		128		0.895	0.862	0.281	0.433
512		128		0.908	0.865	0.242	0.463
1024		128		0.920	0.858	0.214	0.591

Observations and Explanations

Computation time

The average time for training smaller units with one hidden layer took around 3 minutes. The training time increased with the increase in the network's complexity (number of layers and number of neurons per layer).

Effect of Neurons in each hidden layer

As we increase the number of neurons, the non-linearity of the model increases, and the decision boundary becomes complex. Too many neurons and the decision boundary separates each and every data point in the training set, which may not generalize well to new unseen data.

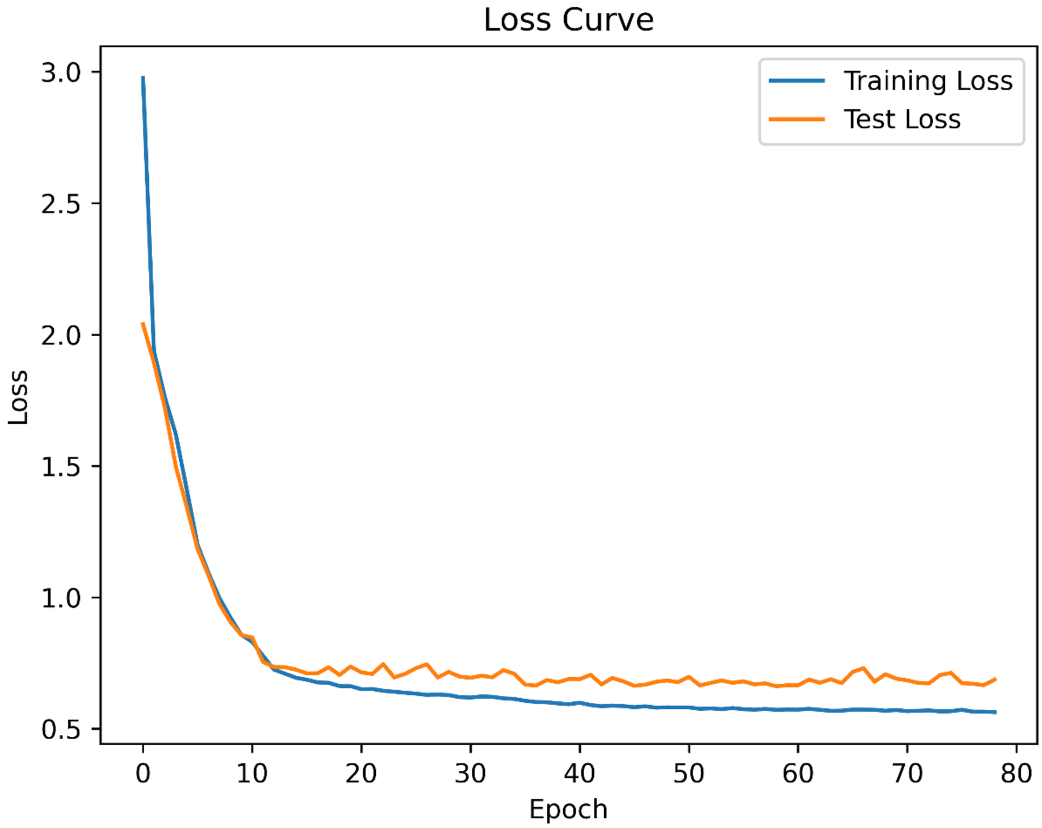
Here, we compare the performance of 1 hidden layer network, with ReLU function as activation, batch size = 128, with the increase in neurons.

1024 units network converges the fastest, and 16 units converges the slowest.

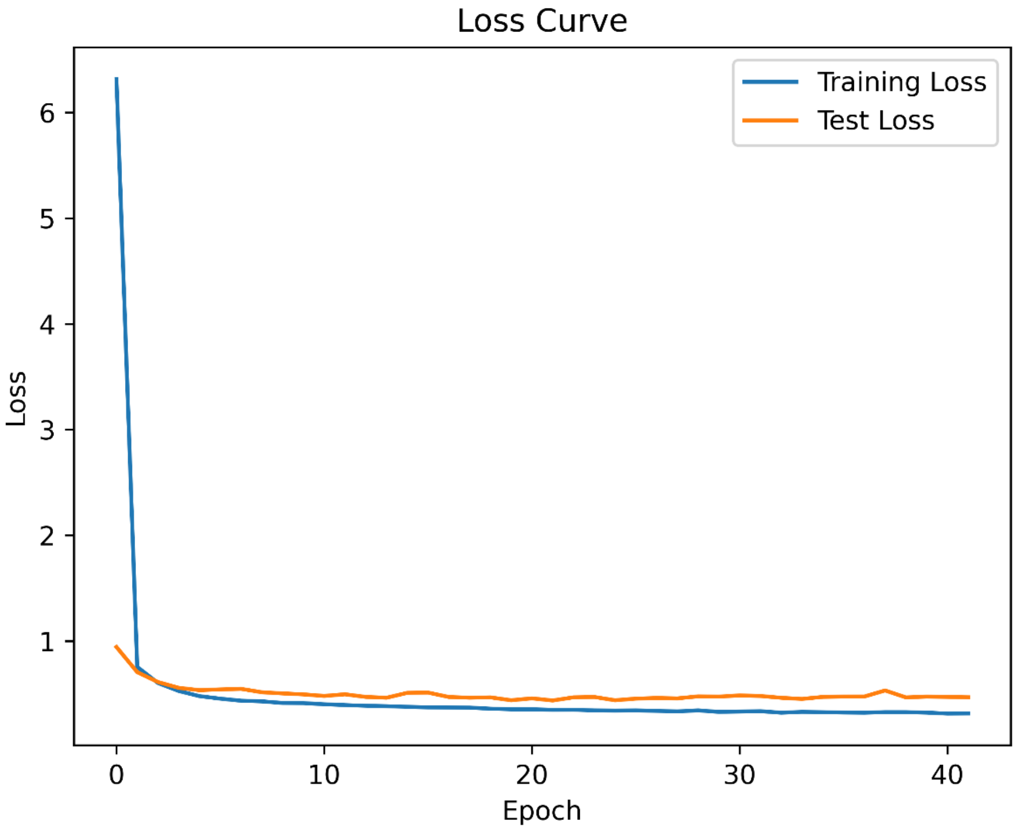
Number of units	Train Accuracy	Test Accuracy
16	0.792	0.768
128	0.879	0.846
1024	0.886	0.857

Loss vs Epochs plots

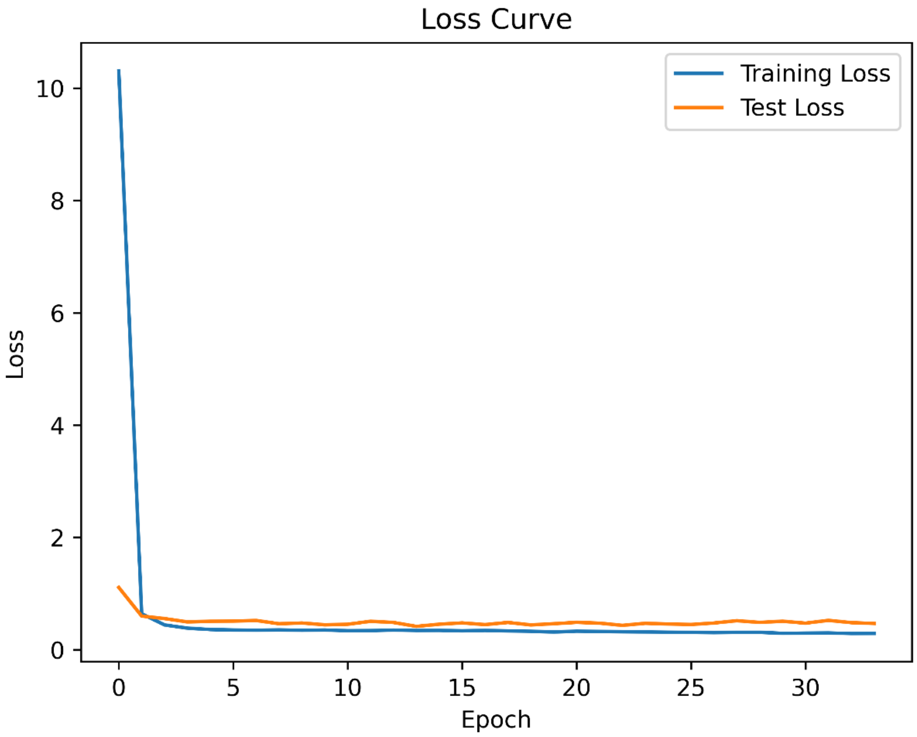
Number of units: 16



Number of units: 128



Number of units: 1024



Effect of Hidden Layers

We see the increase in both train and test accuracy after increasing the number of layers for a shallow neural network on Fashion MNIST. This is possibly due to the increase in the number of decision regions ($O\left(\binom{n}{l}^{d(l-1)} \cdot n^d\right)$ Where n is the number of neurons in each hidden layer, l is the number of hidden layers, and d is the number of input features), which increases non-linearity and decreases misclassification. However, with too many layers, we risk having too many decision regions and parameters that overfit the model.

Empirically, greater depth does seem to result in better generalization for a wide variety of tasks. Having multiple layers makes the network more eager to recognize specific features of input data.

This suggests that using deep architectures expresses a useful prior over the space of functions the model learns.

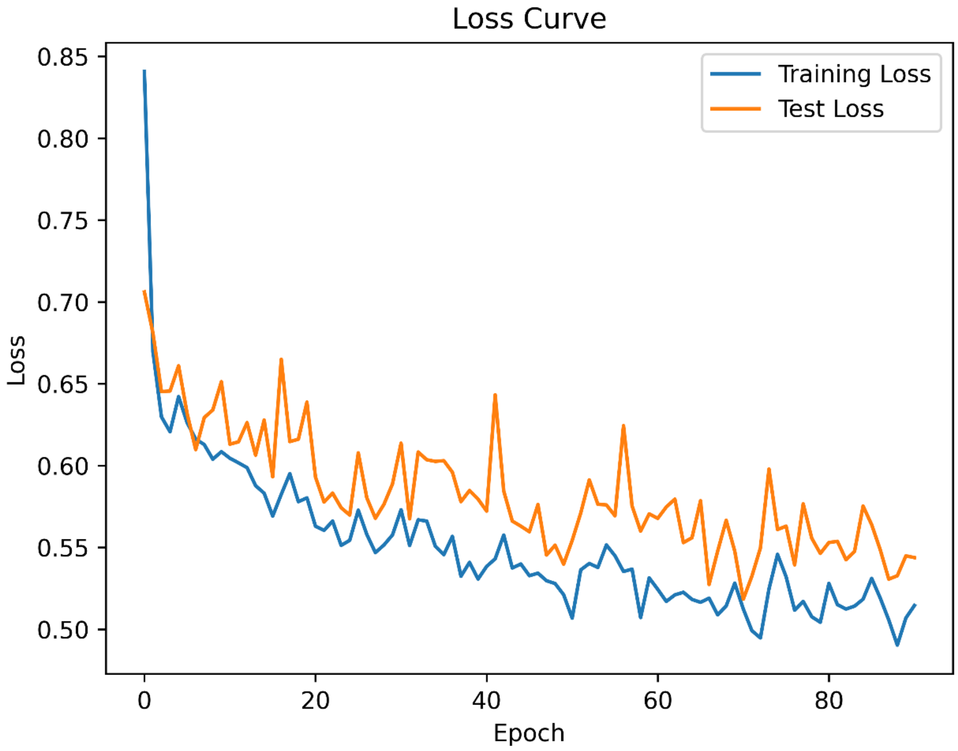
Here, we compare the performance in 128 units per hidden layer network, with sigmoid function as activation. batch size = 128

The effect of adding a layer is more significant than adding neurons to a single hidden layer.

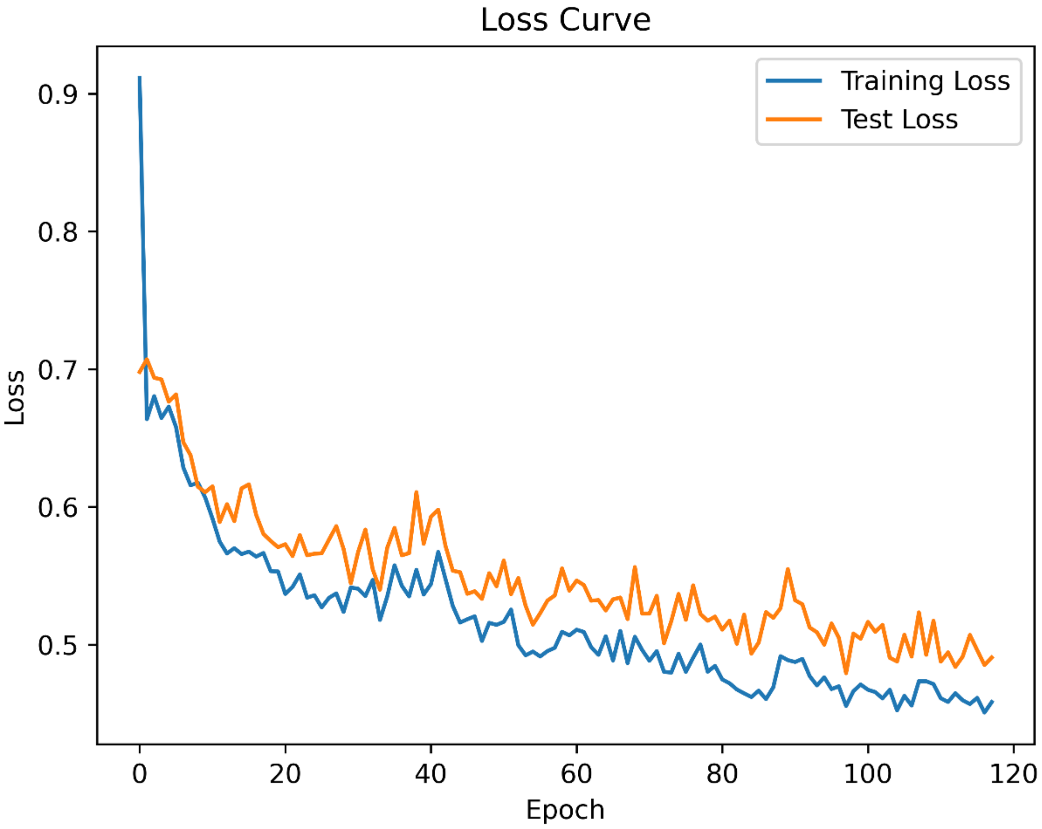
Number of layers	Train Accuracy	Test Accuracy
1	0.825	0.814
2	0.842	0.830
5	0.826	0.814

Loss vs Epoch plots

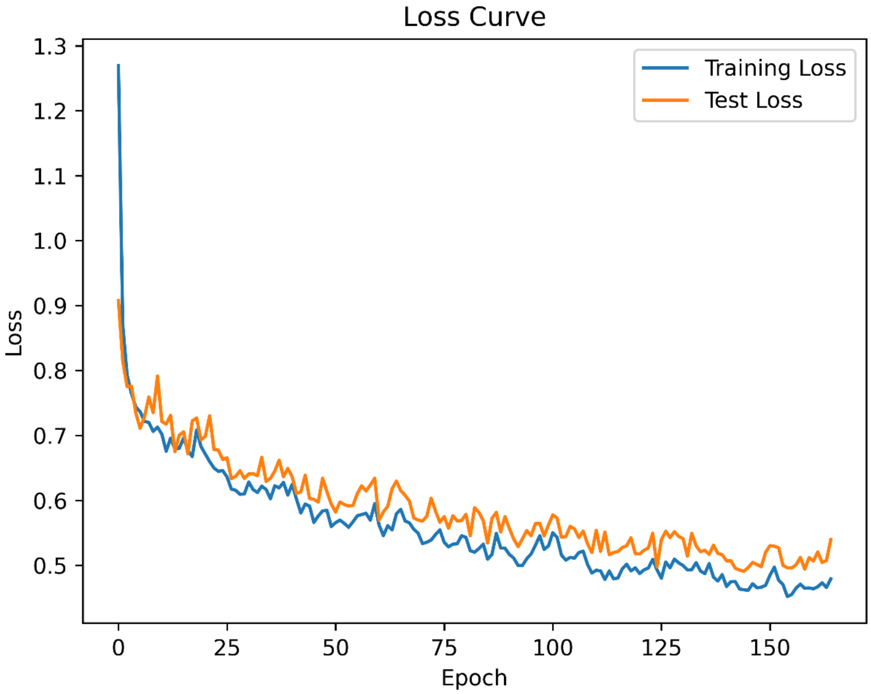
Number of layers: 1



Number of layers: 2



Number of layers: 5



Effect of Activation Function

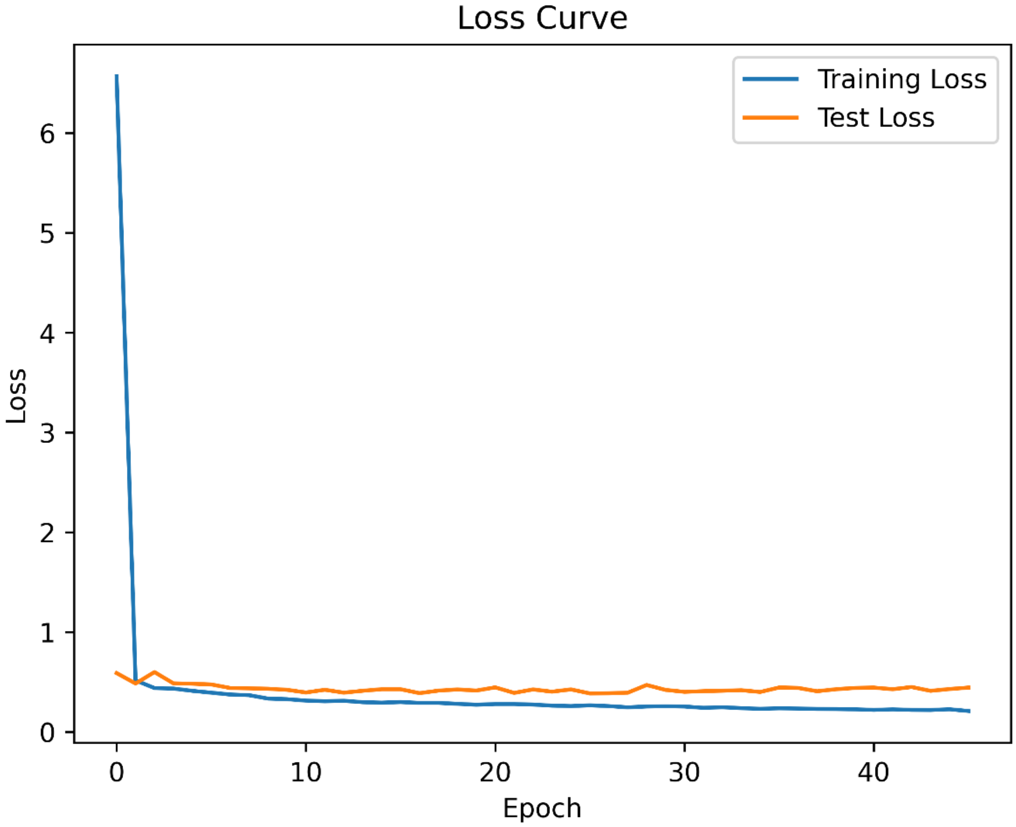
ReLU activation performs the best of all three, while tanh performs the worst of them all. This is because the ReLU function does not saturate for larger weights while both tanh and sigmoid do. Tanh saturates fastest among all three, which might be the reason for lower accuracy.

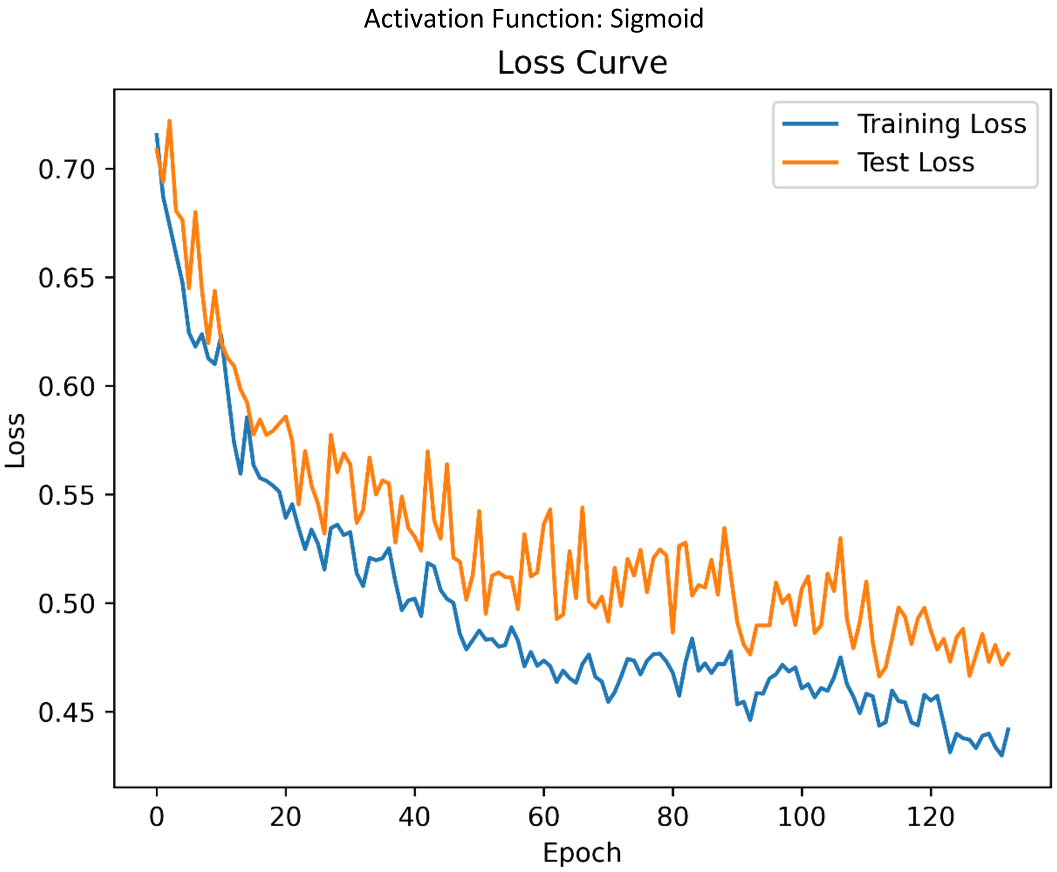
Here, we compare the performance of activation functions of a network with 1024 units per hidden layer (2 hidden layers), with batch size = 128

Activation function	Train Accuracy	Test Accuracy
ReLU	0.908	0.874
sigmoid	0.838	0.825
tanh	0.777	0.768

Loss vs Epoch plots

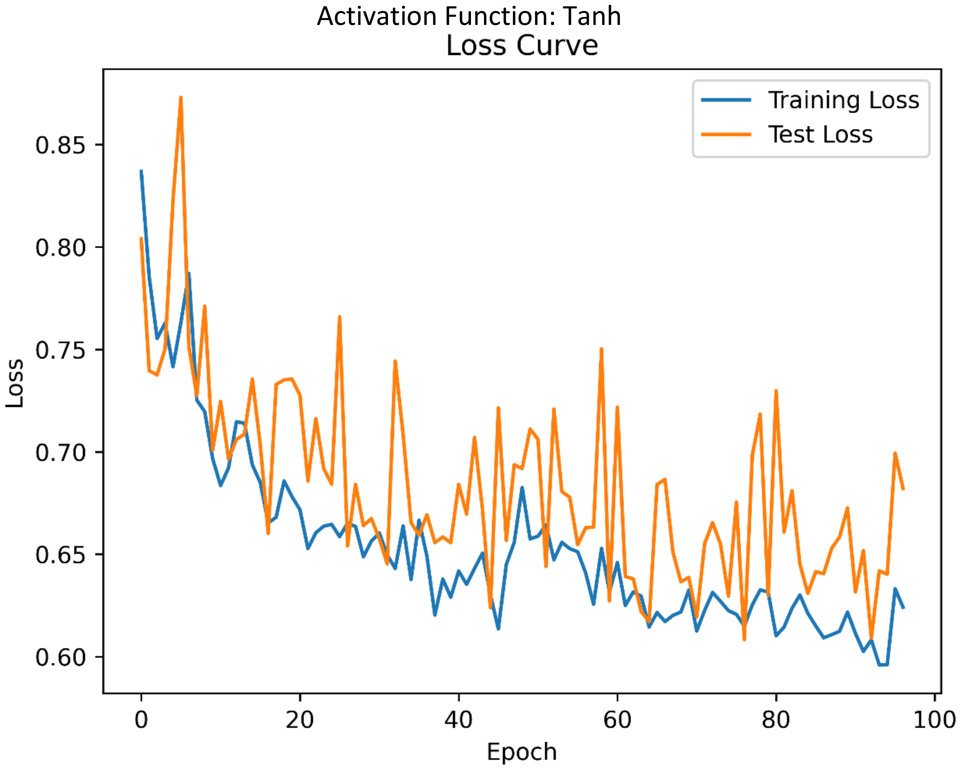
Activation Function: ReLU





Activation Function: Sigmoid

Loss Curve



Activation Function: Tanh

Loss Curve

Conclusion

From this, we conclude that the optimal set of hyperparameters (the set of hyperparameters that gave the highest accuracy on test data) is:

- Batch size: 128
- Number of hidden layers: 5
- Number of neurons per layer: 1024
- Activation function: ReLU

We also conclude that the worst set of hyperparameters (the set of hyperparameters that gave the least accuracy on test data) is:

- Batch size: 128
- Number of hidden layers: 1
- Number of neurons per layer: 128
- Activation function: Tanh

References

- Dataset - Fashion MNIST | Kaggle
- Keras Documentation - Module: tf.keras | TensorFlow Core v2.6.0

Glossary

Name	Description
Activation Function	The activation function defines the output of that node given an input or set of inputs.
ADAM Optimizer	Adam optimization is a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments.
Batch Size	Batch size refers to the number of training examples utilized in one iteration.
Cross-Entropy Loss	Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1.
Decision Boundary	A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous.
Deep Learning	Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning.
Early Stopping	Early stopping is a form of regularization used to avoid overfitting when training a learner with an iterative method
Epoch	Epoch indicates the number of passes of the entire training dataset the machine learning algorithm has completed.
Fashion MNIST	Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples.
Glorot Uniform Initialization	The goal of Glorot Initialization is to initialize the weights such that the variance of the activations is the same across every layer.
Keras	Keras is an open-source software library that provides a Python interface for artificial neural networks.
Layer	A layer is a structure in the model's architecture, which takes information from the previous layers and then passes information to the next layer.
Neural Network	A neural network is a network or circuit of neurons