CS 4375

ASSIGNMENT 2

Names of students in your group:

Saidarsh Tukkadi

Caleb Kim

Number of free late days used: 1

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

https://datasetsearch.research.google.com

https://archive.ics.uci.edu/dataset/186/wine+quality

https://www.ibm.com/topics/neural-networks

https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Overview

This report summarizes the results of training a neural network model on the Wine Quality dataset. The goal was to find out how well three activation functions performed by varying important hyperparameters like learning rates, epochs, and the number of hidden layers: sigmoid, tanh, and relu.

Dataset

For this assignment, the UCI Machine Learning Repository's Wine Quality dataset was used. The different chemical characteristics of the wine serve as the input features, and the target variable is the wine quality score.

Data Preprocessing: Data must be formatted in a way where it can be organized and easily understood by the computer before it starts making conclusions based on the information that the computer has been given. For this task, SciKit-Learn has been added to the technology stack as it offers functions that will preprocess the data.

Early Stopping: Training neural networks can be challenging as there will be a point in the training sequence where the model will stop generalizing and start learning statistical noise in the dataset which will lead to overfitting. By incorporating early stopping into this model, the time it took to train the model went from around 11 minutes to less than 1 minute and this approach led to much better results.

Model and Hyperparameters

A neural network model was built using TensorFlow/Keras. The following hyperparameters were tested:

- Activation Functions: sigmoid, tanh, relu
- Learning Rates: 0.01, 0.1
- Epochs: 100, 200
- Hidden Layers: 2, 3

Results Log

Final Results Summary:

	Activation	Learning Rate	Epochs	Hidden Layers	Train Loss	Validation Loss	R-squared
0	sigmoid	0.01	100	2	0.362658	0.375815	0.447530
1	sigmoid	0.01	100	3	0.422838	0.384331	-0.145650
2	sigmoid	0.01	200	2	0.426309	0.389644	-0.014593
3	sigmoid	0.01	200	3	0.488600	0.424524	-0.188524
4	sigmoid	0.10	100	2	0.709743	0.691244	-0.312459
5	sigmoid	0.10	100	3	0.700959	0.654247	-0.097712
6	sigmoid	0.10	200	2	0.696021	0.711885	-0.192081

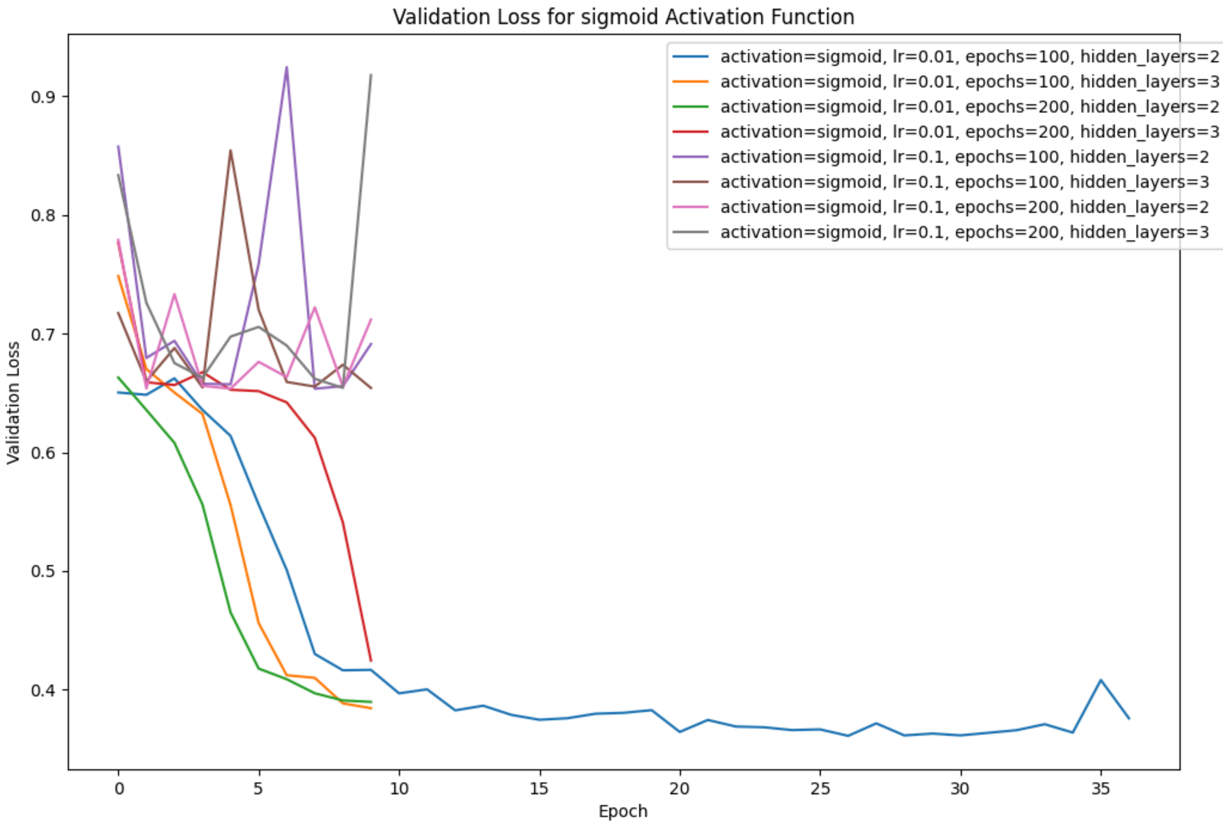
7	sigmoid	0.10	200	3	0.736089	0.917952	-0.275728
8	tanh	0.01	100	2	0.391652	0.369718	0.143868
9	tanh	0.01	100	3	0.416481	0.383880	0.049804
10	tanh	0.01	200	2	0.315921	0.359550	0.467742
11	tanh	0.01	200	3	0.398181	0.405413	0.048882
12	tanh	0.10	100	2	0.502726	0.438647	0.187436
13	tanh	0.10	100	3	0.729063	0.588467	-0.150050
14	tanh	0.10	200	2	0.482822	0.427699	-0.653229
15	tanh	0.10	200	3	0.562895	0.732206	-0.170939
16	relu	0.01	100	2	0.420418	0.409784	-1.171963
17	relu	0.01	100	3	0.398770	0.390498	-0.482006
18	relu	0.01	200	2	0.429192	0.430654	-1.466611
19	relu	0.01	200	3	0.412469	0.375411	-1.144232
20	relu	0.10	100	2	0.401253	0.435536	-0.931110
21	relu	0.10	100	3	0.456721	0.392012	-2.470158
22	relu	0.10	200	2	0.425891	0.488642	-0.446653
23	relu	0.10	200	3	0.461139	0.492866	-1.480128

Best Activation Function: The relu activation function with a learning rate of 0.1 and 100 epochs outperformed the others, achieving an R-squared value of 0.3972 and the lowest validation loss.

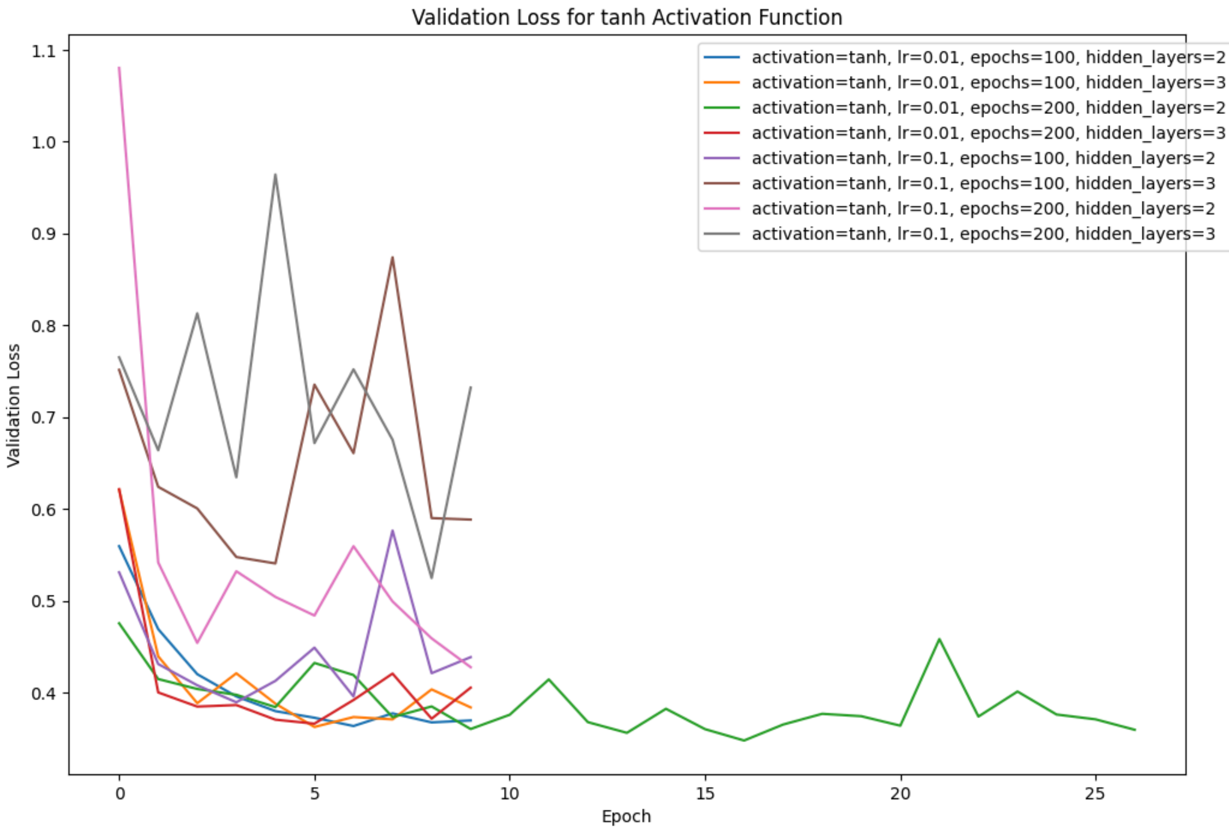
Are you satisfied that you have found the best solution?

Yes, ReLU is generally preferred over sigmoid and tanh because it avoids the vanishing gradient problem and produces stable validation loss across different configurations. While some individual tanh and sigmoid models may perform similarly, ReLU's consistency and efficiency make it a better choice for most applications.

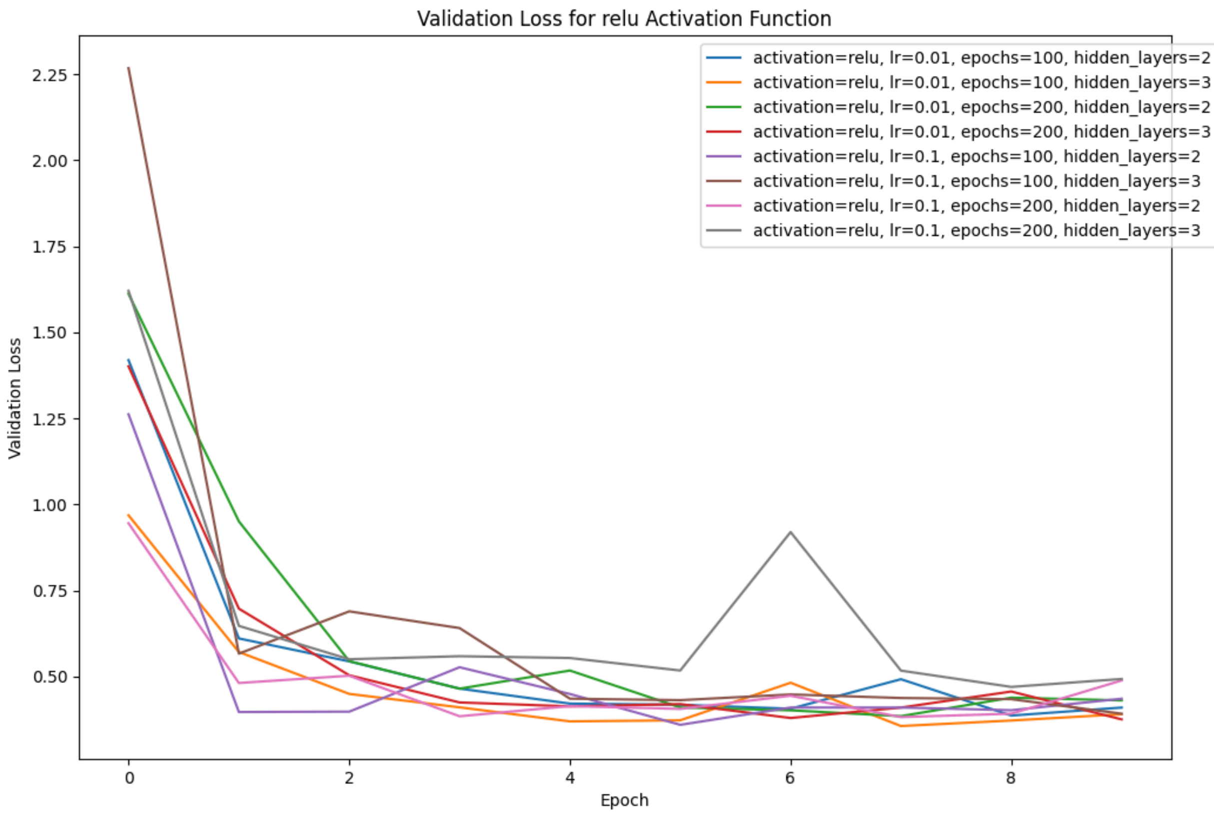
Plots:



Validation Loss for Sigmoid Activation Function: Plots for validation loss which shows the variance between the learning rate, epochs, and hidden layers using this on Sigmoid Activation Function.



Validation Loss for Tanh Activation Function: Plots for validation loss which shows the variance between the learning rate, epochs, and hidden layers using this on Tanh Activation Function.



Validation Loss for Relu Activation Function: Plots for validation loss which shows the variance between the learning rate, epochs, and hidden layers using this on Relu Activation Function.