Task 1 (Regression)

A brief description of the dataset.

N/B; An API to the UCI Machine Learning Repository is required to fetch the dataset by running; !pip3 install -U ucimlrepo. Source - https://archive.ics.uci.edu/dataset/1/abalone

Dataset is Abalone (ID: 1). Number of Instances: 4177, Number of Features: 8 (7 numerical and one nominal). Features include;

In [22]:
```
# variable information
abalone.variables
```

Out[22]:

	name	role	type	demographic	description	units	missing_values
0	Sex	Feature	Categorical	None	M, F, and I (infant)	None	no
1	Length	Feature	Continuous	None	Longest shell measurement	mm	no
2	Diameter	Feature	Continuous	None	perpendicular to length	mm	no
3	Height	Feature	Continuous	None	with meat in shell	mm	no
4	Whole_weight	Feature	Continuous	None	whole abalone	grams	no
5	Shucked_weight	Feature	Continuous	None	weight of meat	grams	no
6	Viscera_weight	Feature	Continuous	None	gut weight (after bleeding)	grams	no
7	Shell_weight	Feature	Continuous	None	after being dried	grams	no
8	Rings	Target	Integer	None	+1.5 gives the age in years	None	no

The dataset is about predicting the age (the target, computed by adding 1.5 to Rings column) of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope. Other measurements, which are easier to obtain, are used to predict the age.

A clear description of the four models you tried.

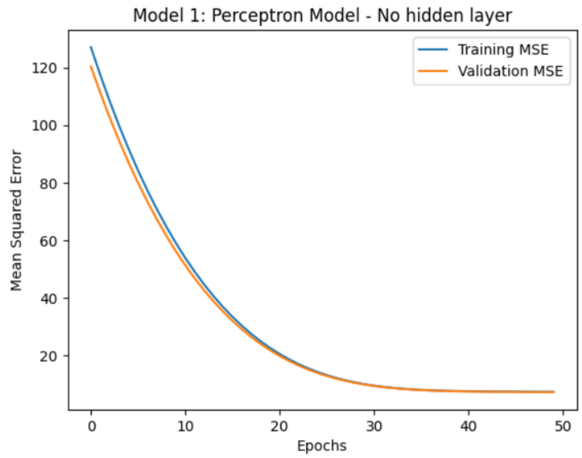
Model 1: Perceptron Model - This model represents a single-layer perceptron with a linear activation function. It is trained for 50 epochs with the mean squared error (MSE) as the loss function and the Adam optimizer for optimization. It has no hidden layer therefore directly maps inputs to outputs.

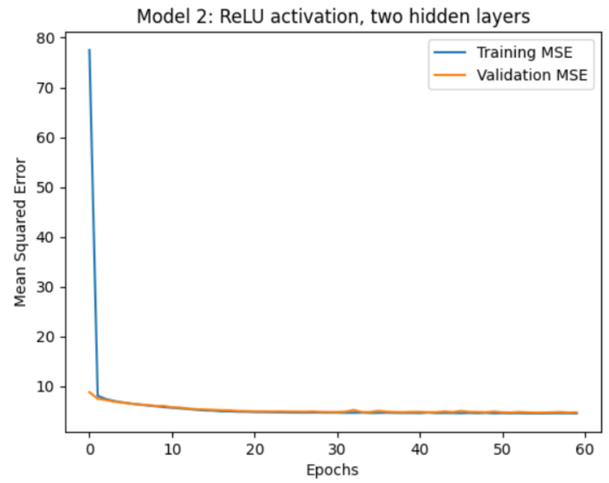
Model 2: ReLU activation – This model has two hidden layers with 20 nodes for the first hidden layer and 25 nodes for the second hidden layer. It utilizes ReLU activation function in the hidden layers. It also uses Adam optimizer and MSE loss function and trained for 60 epochs.

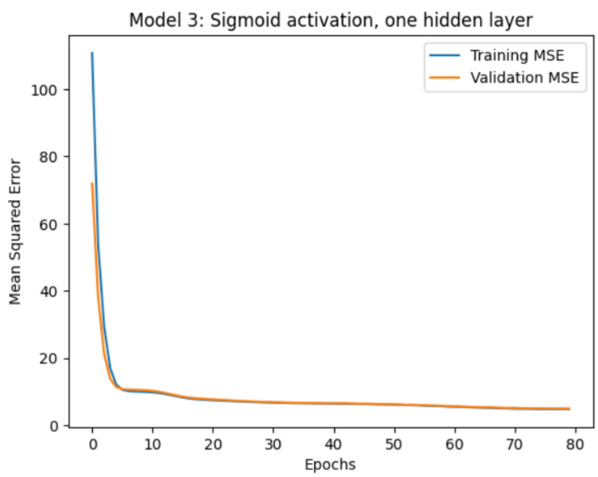
Model 3: Sigmoid activation – This model has one hidden layer with 32 nodes and uses sigmoid activation function in its hidden layer. The model is trained for 80 epochs with the Adam optimizer and MSE loss function.

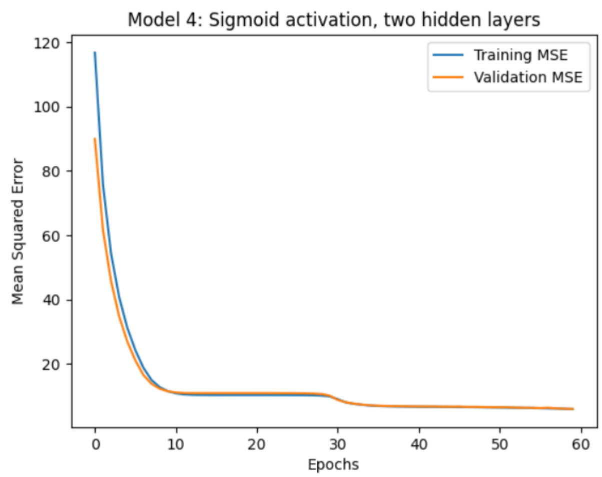
Model 4: Sigmoid activation – This model has two hidden layers with 20 nodes for the first hidden layer and 25 nodes for the second hidden layer. It utilizes sigmoid activation function in its hidden layer. The model is trained for 60 epochs with the Adam optimizer and MSE loss function.

Four graphs, one for each model.









A table of minimum validation errors.

```
Model               Minimum Training Error
1:Perceptron Model              7.344
2:ReLU activation               4.502
3:Sigmoid activation            4.794
4:Sigmoid activation            5.541
=========================================
Model               Minimum Validation Error
1:Perceptron Model              7.273
2:ReLU activation               4.696
3:Sigmoid activation            4.928
4:Sigmoid activation            5.663
=========================================
```

Discussion of the results.

The Perceptron Model demonstrates higher errors compared to the other models. The relatively high errors suggests that the linear model might not be expressive enough to capture the complexities in the data. The ReLU activation model (Model 2) achieves the lowest errors on both training and validation data among the models tested. The model is able to learn more complex relationships within the data by adding two hidden layers with ReLU activations.

The models with sigmoid activation, the one hidden layer model outperformed the two hidden layer model as seen with their validation errors. This shows that adding more hidden layers doesn't always guarantee improved performance. In this case, the architecture might be more prone to overfitting. Overall, the results shows that the ReLU activation model (Model 2) performs the best among the architectures tested while the Perceptron Model (Model 1) lags behind due to its simplicity.

Task 2 (Classification)

A brief description of the dataset.

Dataset 2: ID: 40983 (Wilt Data Set)

URL;
https://www.openml.org/search?type=data&sort=runs&status=active&qualities.NumberOfClasses=%3D_2&id=40983

This data set involved detecting diseased trees in Quickbird satellite images. The high-resolution QuickBird images of 165 different area were acquired in 27 August 2012 to detect the diseased trees. The QuickBird images contains four 2.4 m resolution MS bands; G, R, NIR and PAN band.

Task: The dataset is about the detection of Pine Wilt Disease (PWD) infected trees and non-affected. The number of instances (rows) in the data set is 4839, and the number of features (columns) is 6.

The features:

1. GLCM_Pan: GLCM mean texture (Pan band)
2. Mean_G: Mean green value
3. Mean_R: Mean red value
4. Mean_NIR: Mean NIR value
5. SD_Pan: Standard deviation (Pan band)

Class (**Target**): ‘infected’ correspond ‘1’, and ‘non-affected’ to 0

A clear description of the four models you tried.

Model 1: ReLU activation, one hidden layer.

Hyperparameters: 1 hidden layer, 30 nodes per layer, ReLU activation function, trained for 60 epochs.

Model 2: ReLU activation, two hidden layers

Hyperparameters: 2 hidden layers, 42 nodes per layer, ReLU activation function, trained for 80 epochs.

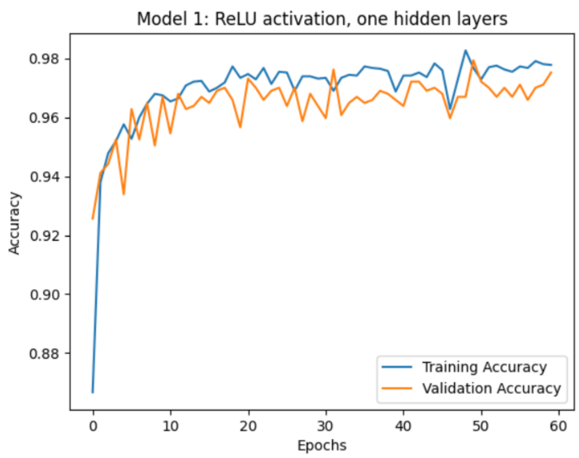
Model 3: Sigmoid activation, one hidden layer

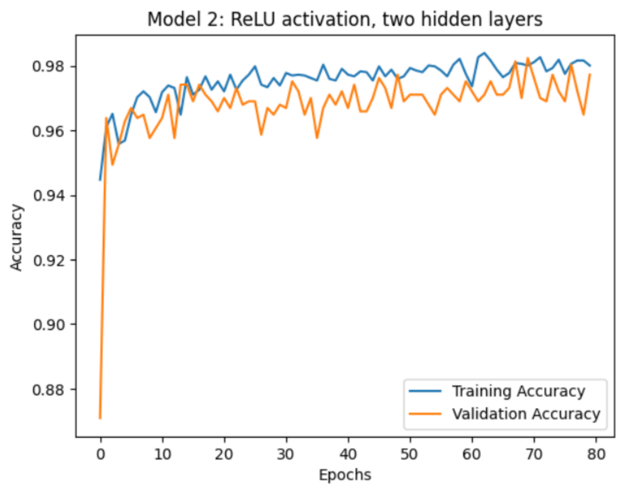
Hyperparameters: 1 hidden layer, 30 nodes per layer, sigmoid activation function, trained for 60 epochs.

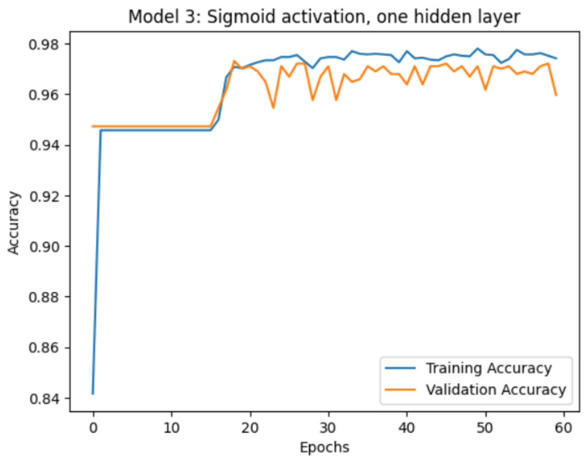
Model 4: Sigmoid activation, two hidden layers

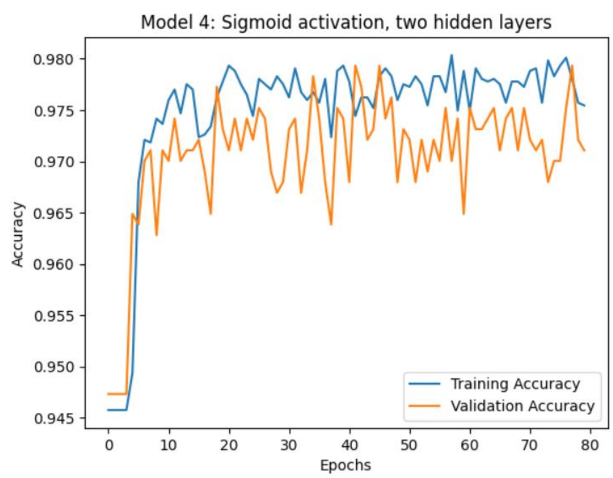
Hyperparameters: 2 hidden layers, 42 nodes per layer, sigmoid activation function, trained for 80 epochs.

Four graphs, one for each model.









A table of maximum validation accuracies.

```
Model           Maximum Training Accuracy
1:ReLU activation Model                  0.983
2:ReLU activation 2HL                    0.984
3:Sigmoid activation                     0.978
4:Sigmoid activation 2HL                 0.980
=========================================
Model           Maximum Validation Accuracy
1:ReLU activation Model                  0.979
2:ReLU activation 2HL                    0.982
3:Sigmoid activation                     0.971
4:Sigmoid activation 2HL                 0.973
=========================================
```

Discussion of the results.

Comparing the hidden layers

For both ReLU and sigmoid activation functions, models with two hidden layers achieve slightly higher maximum training and validation accuracies compared to models with one hidden layer. However, the improvement in accuracy is relatively marginal, suggesting that adding an extra hidden layer does not significantly enhance the model's performance in this scenario.

Comparing the activation functions.

The ReLU activation model consistently achieves higher maximum training and validation accuracies compared to the sigmoid activation model with the same hyperparameters. This suggests that ReLU activation might be more effective in capturing the underlying patterns in this dataset. Similarly, for models with two hidden layers, the ReLU activation model outperforms the sigmoid activation model in terms of both maximum training and validation accuracies.

Conclusion; while adding an extra hidden layer may slightly improve the model's performance, the choice of activation function appears to have a more significant impact. In this specific scenario, ReLU activation consistently leads to higher maximum training and validation accuracies compared to sigmoid activation, regardless of the number of hidden layers.