APPLIED PHYSICS 157

ACTIVITY 09
NEURAL NETWORKS

By : Manalang, Johnenn R.

TABLE OF CONTENTS

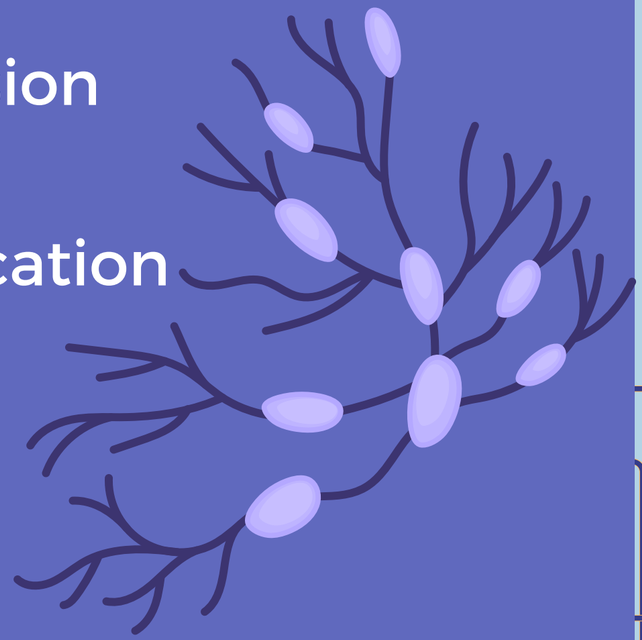
01 Objectives

02 Overview

06 Neural Network for Regression

10 Neural Network for Classification

14 Reflection



OBJECTIVES

01.

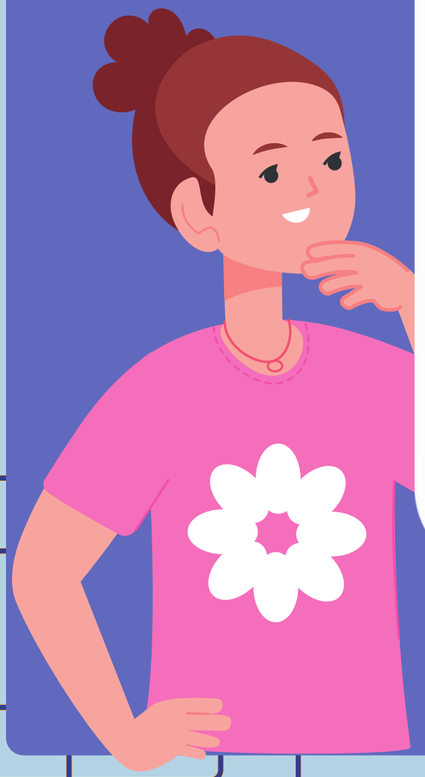
Discuss how Neural Networks work.

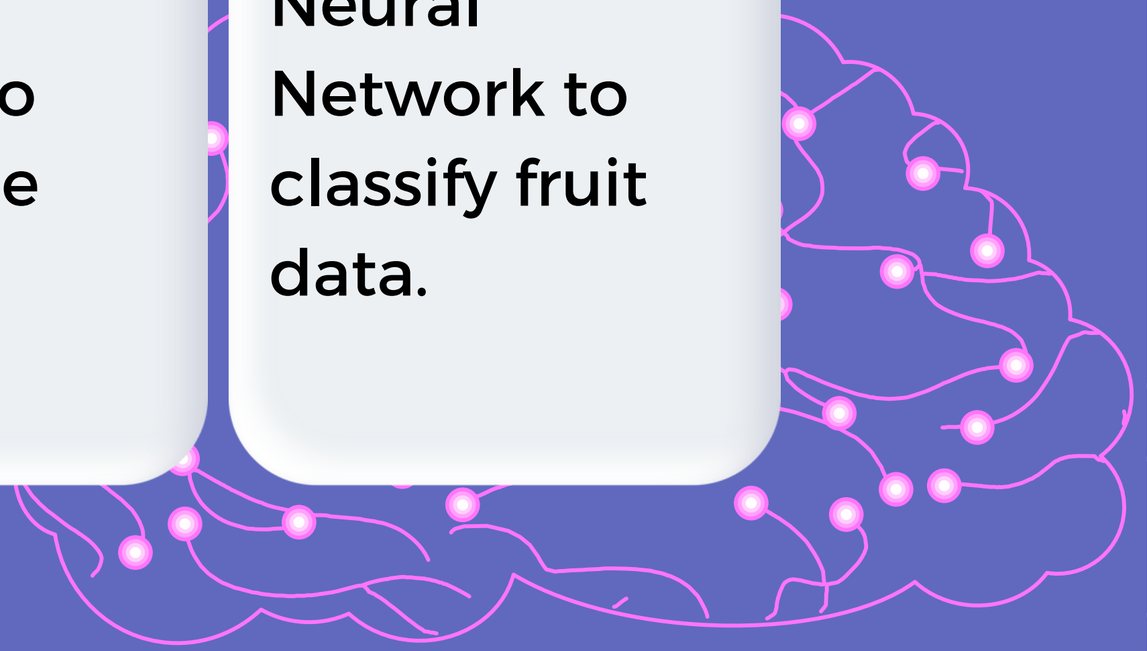
02.

Program a Neural Network to learn a sine function.

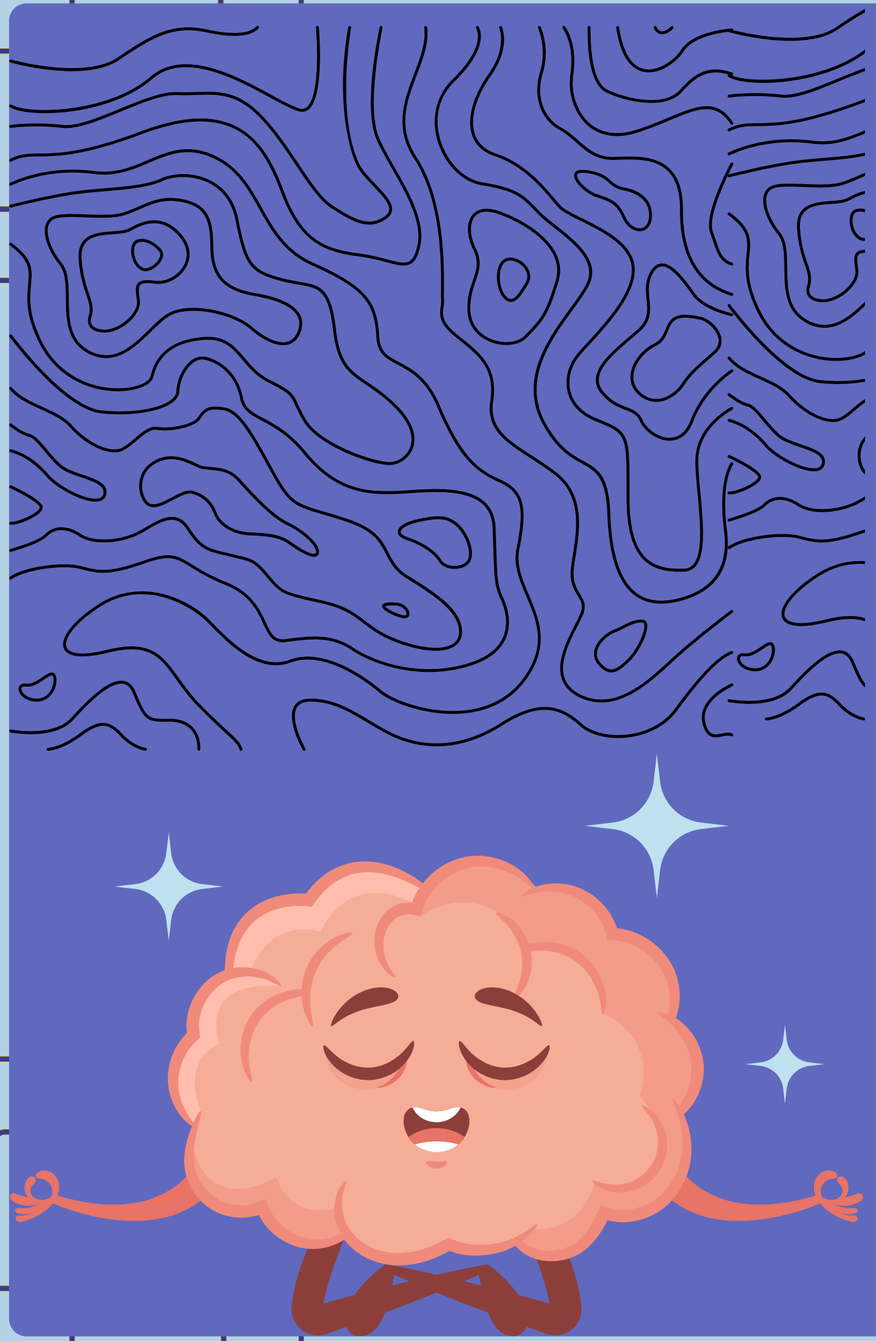
03.

Program a Neural Network to classify fruit data.





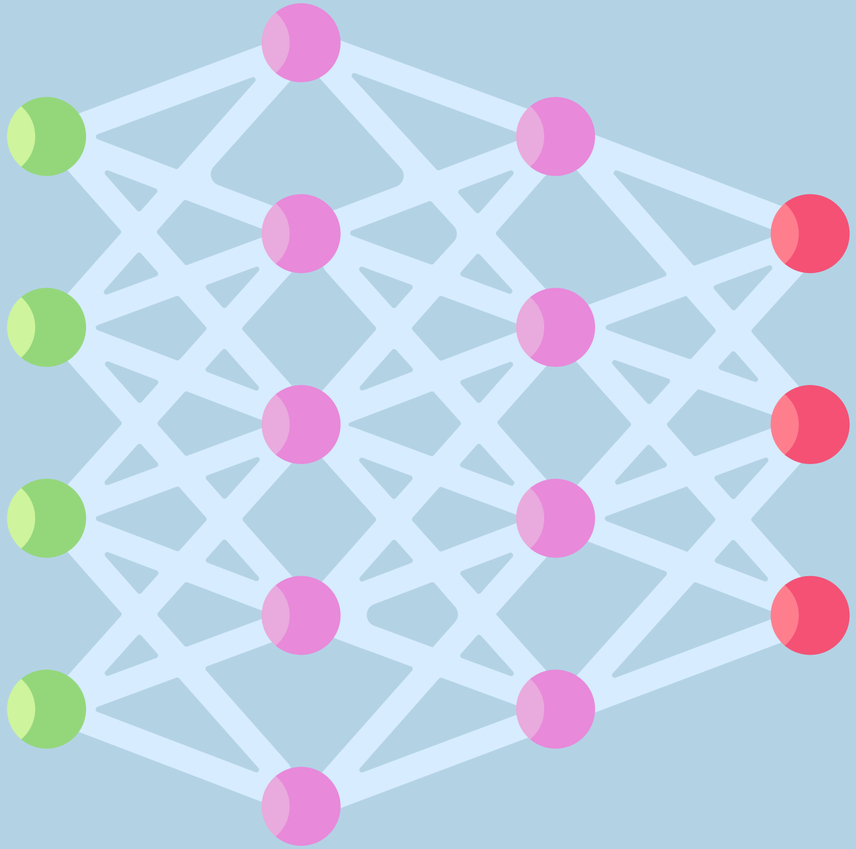
OVERVIEW



Neural networks (NN) are machine learning models inspired by the brain. They consist of interconnected artificial neurons that learn from data to recognize patterns and make predictions. Neural networks have wide-ranging applications, including image recognition, natural language processing, and medical diagnosis, revolutionizing industries with automation and data-driven decision-making. In this activity, we will use NN in regression and classfication.

HOW TO: NEURAL NETWORK

Neural networks are composed of interconnected layers of artificial neurons that process and transmit information. Each neuron applies a mathematical transformation to its inputs and passes the result to the next layer. The network learns by adjusting the weights and biases of these transformations based on the error between its predictions and the desired output.

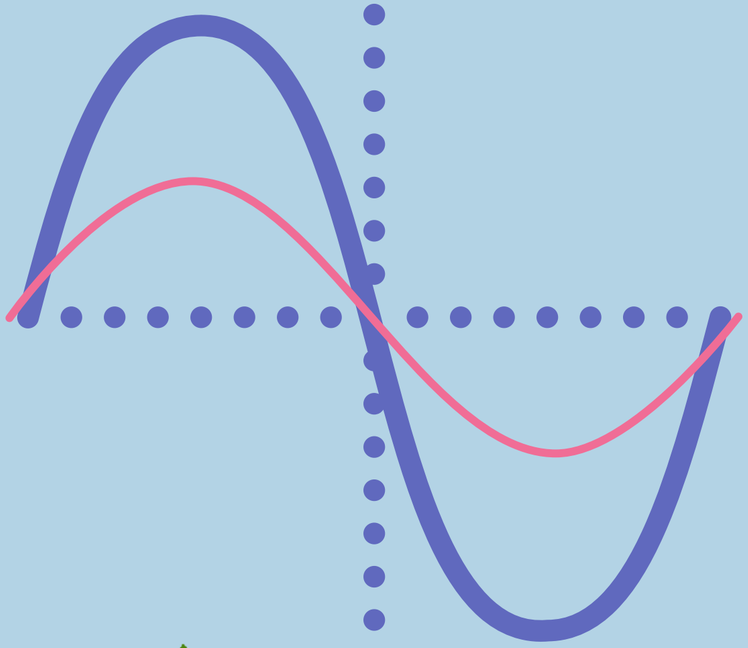


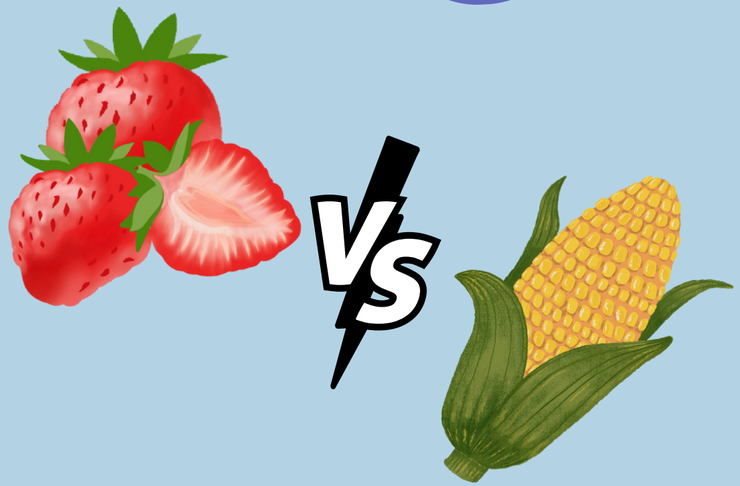
ERROR BACKPROPAGATION

During training, a neural network forwards input data through its layers to generate an output. The error between the predicted and target output is computed using a loss function. Through backpropagation, this error is propagated backward through the network, attributing it to each neuron's contribution. By updating the weights and biases using an optimization algorithm like gradient descent, the network minimizes the error. This iterative process enables the network to learn and improve its predictions by adjusting its parameters based on the computed gradients.

NN AS A CURVE FITTING TOOL

Neural networks serve as powerful curve-fitting tools capable of capturing complex relationships between input and output data. They excel at tasks such as regression, classification, and pattern recognition. With their ability to learn from data, neural networks have found diverse applications in various fields. In the following slides, we will demonstrate NN for regression by programing it to learn a sine function and for classification by programming it to classify fruit data.



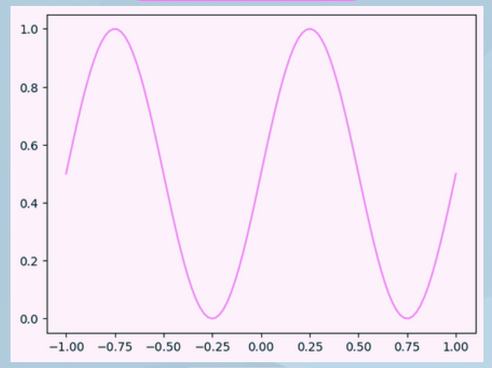


NEURAL NETWORK FOR *Regression*

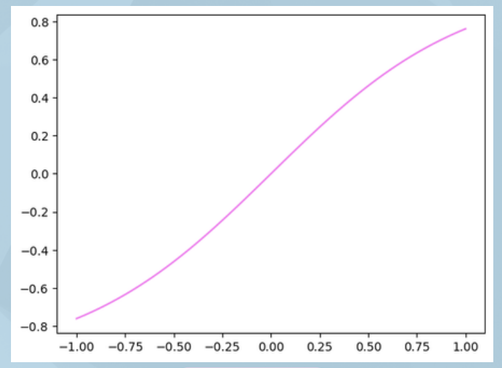
NN FOR REGRESSION

MODEL

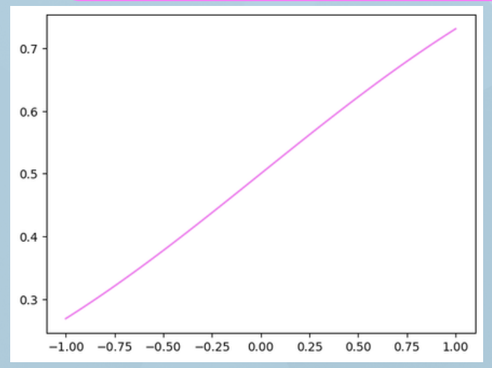
ACTIVATION FUNTIONS



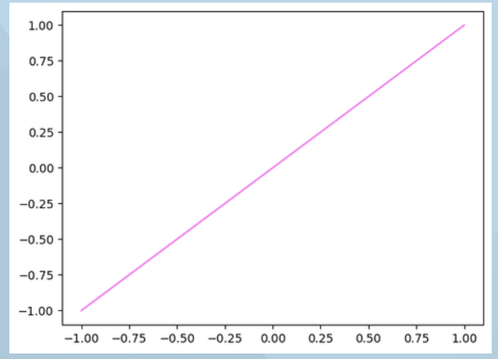
sine



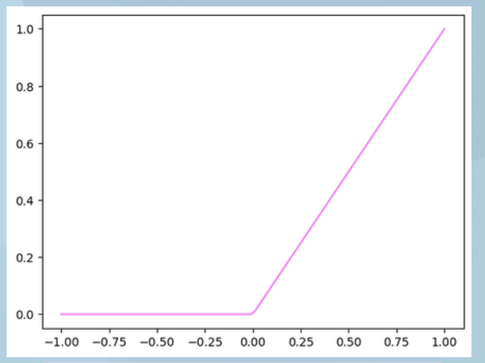
tanh



sigmoid



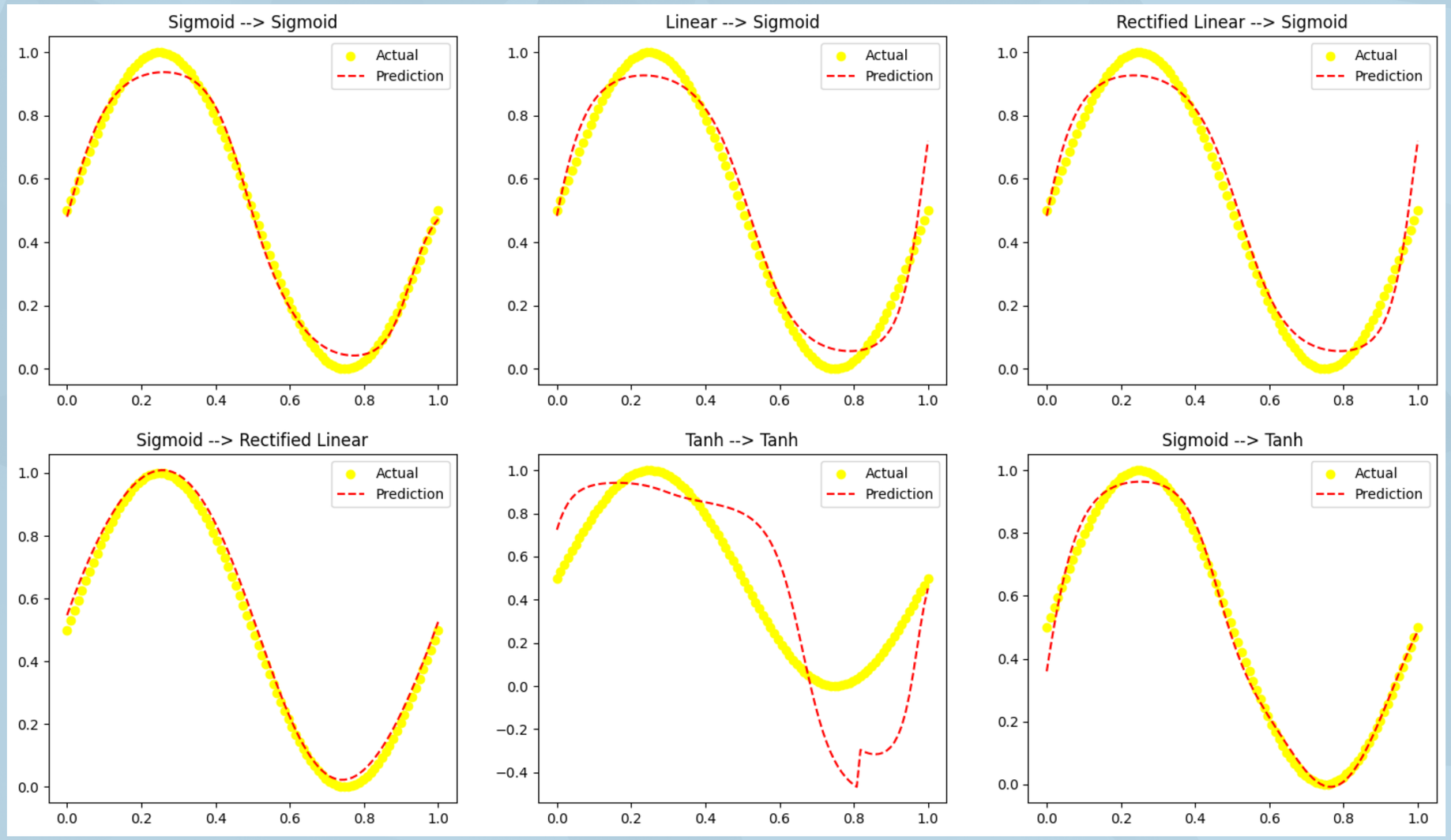
linear



rectified linear

In Neural Network, different activation functions can be used for the hidden and output layers. In our case, the model function is the sine function and the activation functions that we used are the hyperbolic tan, sigmoid, linear, and rectified linear. Note that for a function to be chosen as an activation function, it should be differentiable or piecewise differentiable.

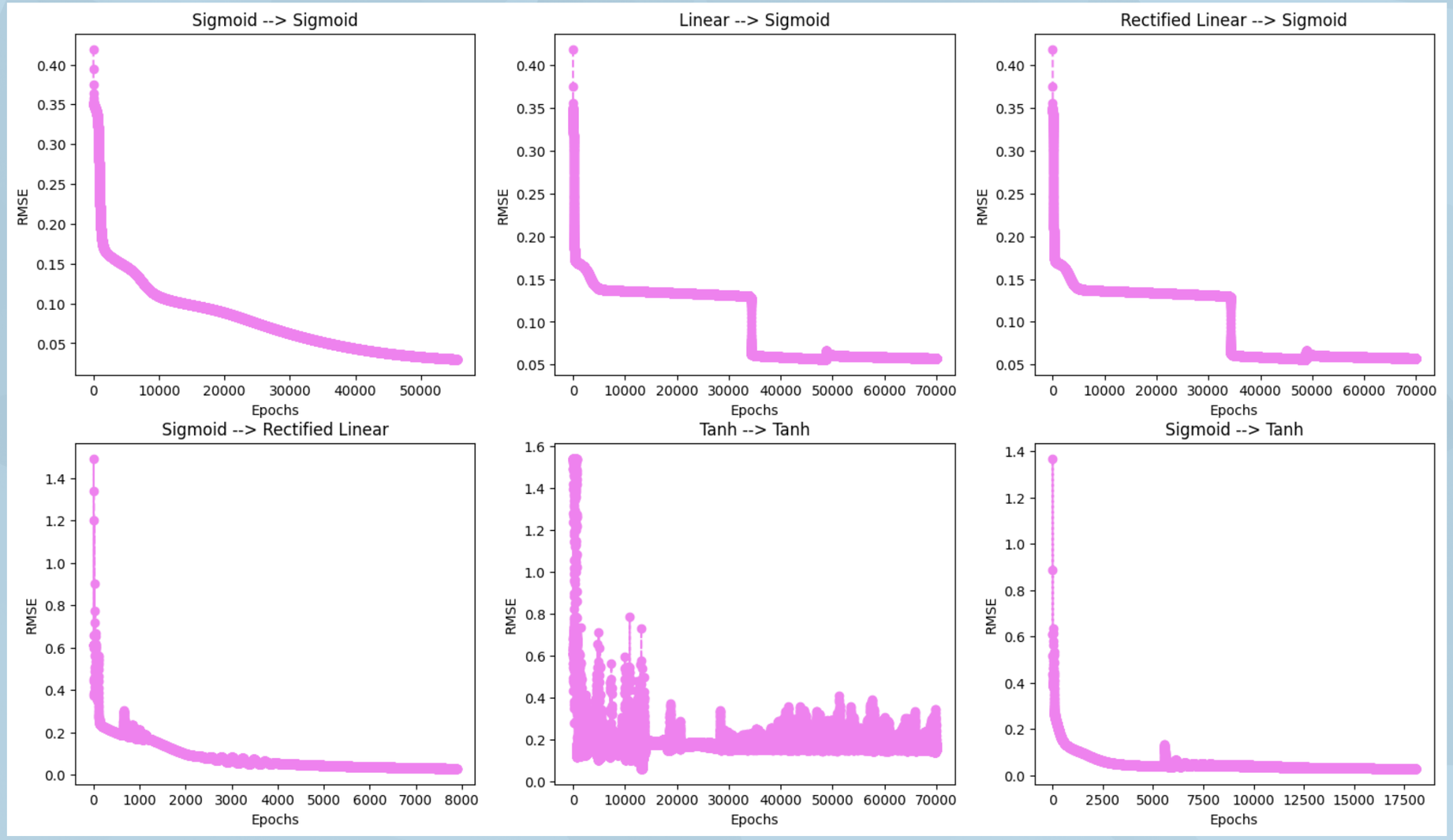
NN PREDICTION



The output plots with different combinations of activation functions for the hidden and output layers are shown. As we can see, the neural network successfully learned and predicted the sine function. However, the tanh-tanh combination seems unusual. We will explain this further in the next slide.

Notation: hidden layer --> output layer

RMSE VS. EPOCH



Notation: hidden layer --> output layer

Plotting RMSE vs. Epoch allows us to see if the network is converging to a solution. A desirable trend is a gradual decrease in error over epochs, indicating that the network is learning and improving its predictions. Following this trend, we can observe that the sigmoid-sigmoid combination performs better. On the other hand, the tanh-tanh combination seems to struggle in converging to a solution, as we can observe multiple peaks instead of a decreasing trend.

NEURAL NETWORK FOR *Classification*

FRUIT CLASSIFICATION

OR

For this classification, I will use the dataset that I previously used for the Perceptron. I have already extracted two features: (1) eccentricity and (2) greenness from the dataset. I have saved the values for the strawberry and corn in two separate CSV files and have set aside half of the data from each class as the training set and the remaining half as the test set. Similar to regression, I will also try different combinations of hidden and output activation functions and compare their performance.

ACCURACY

ACCURACY = NO. OF CORRECT PREDICTIONS/TOTAL NUMBER OF PREDICTIONS

Combinations:

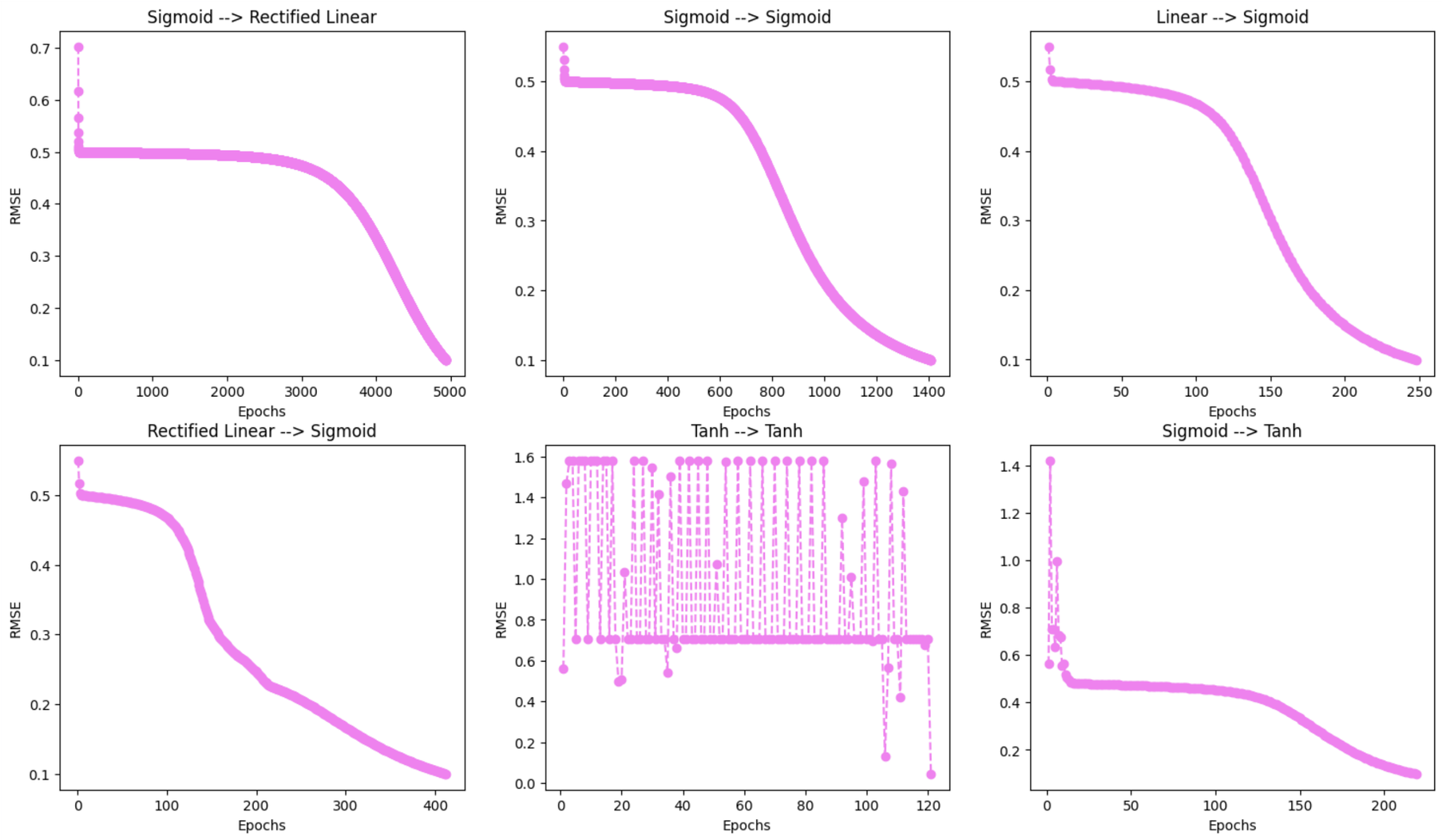
- **Sigmoid, Rectified Linear**
- **Sigmoid, Sigmoid**
- **Linear, Sigmoid**
- **Rectified Linear, Sigmoid**
- **Tanh, Tanh**
- **Sigmoid, Tanh**

RESULT:

```
Summary of the classification:
Accuracy (Corn):  100.0 %
Accuracy (Strawberry):  100.0 %
```

For all the combinations of activation functions, the resulting accuracy for the classification of both the corn and strawberry is 100%. For the classification, I used a learning rate of 0.0005, 10 hidden nodes, 3 input nodes, and 100 data points.

RMSE VS. EPOCH

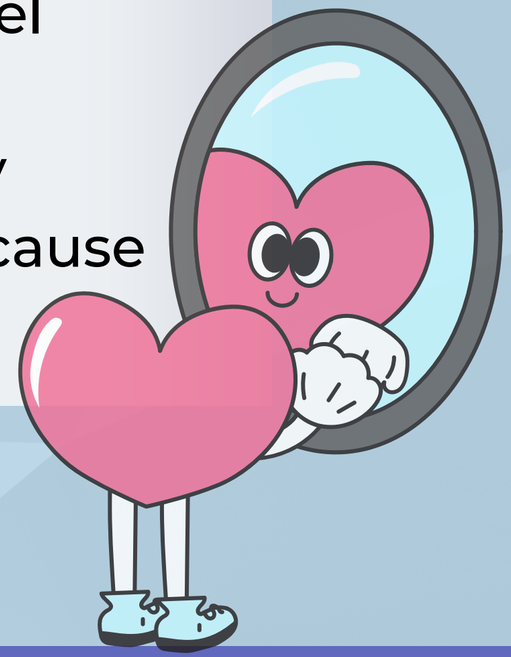


The plot of RMSE vs Epoch is shown here. As we can observe, all the models converged, but they did so at different rates. In general, if a neural network takes fewer epochs to converge, it is considered considerably better. This is because it requires fewer computational resources and less training time, and it helps prevent overfitting. The number of epochs required for convergence can vary depending on the quality of the data, the complexity of the problem, and the chosen parameters.

Notation: hidden layer --> output layer

REFLECTION

This is the cause of my stress for weeks! Neural networks is not that easy to understand at first. I had to consult several references just to understand how to do it. Luckily, I have my labmates who has helped me absorb the concepts. Special thanks to Jonabel and Julian for sharing their codes with me. I also used some inspiration from the code of gpgloria from github. They really helped me survive this activity and survive this semester (because this is the last activity I will submit for the sem!)



REFERENCES

1. https://github.com/gpgloria/Physics-Projects/blob/main/Applied%20Physics%20186/ML5%20-%20Neural%20Networks.ipynb?fbclid=IwAR1Y9Aj962a52jKSch2A9Vp4Gp5hUHbEZBoWZoSkZm_I61tjCaoQM3VPNVc_aem_AcZuY4-vFT4rVtNGgwPW7F0YA2Ycg1tcmZZSovdXfviYGb7y_o0KeZGllDYgjYIdg1c
2. https://uvle.upd.edu.ph/pluginfile.php/888692/mod_resource/content/1/ML5%20-%20Neural%20Networks.pdf

REPORT GRADE

Criteria	Score
Technical Correctness	35
Quality of Presentation	35
Self Reflection	30
Initiative	10
TOTAL	100

EVALUATION

Overall, I know that I have accomplished all the tasks included in this activity. I have successfully demonstrated how the Neural Network is applied in regression and classification. I also used several combinations of hidden and output layers for comparison.