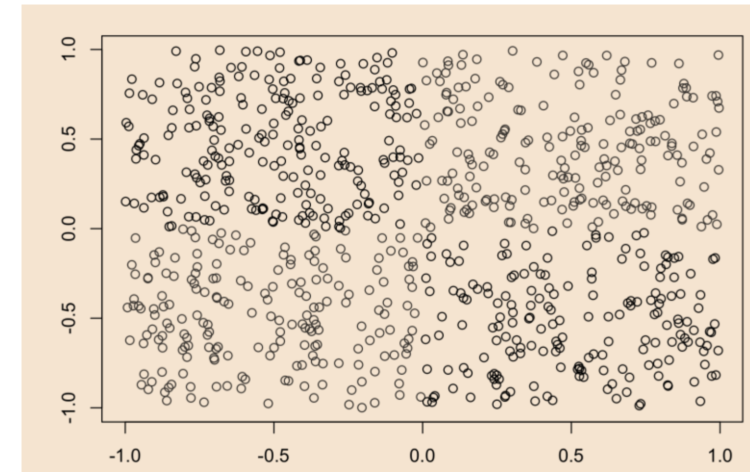
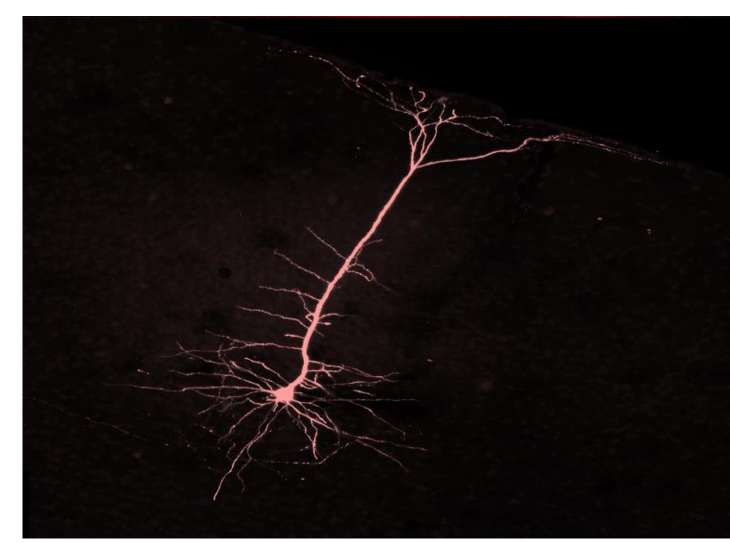
NEURAL NETWORKS

MACHINE LEARNING - MODULE 2





NINO PHILIP RAMONES | GITHUB
2020 - 05616
JULY 3, 2023

NEURAL NETWORKS

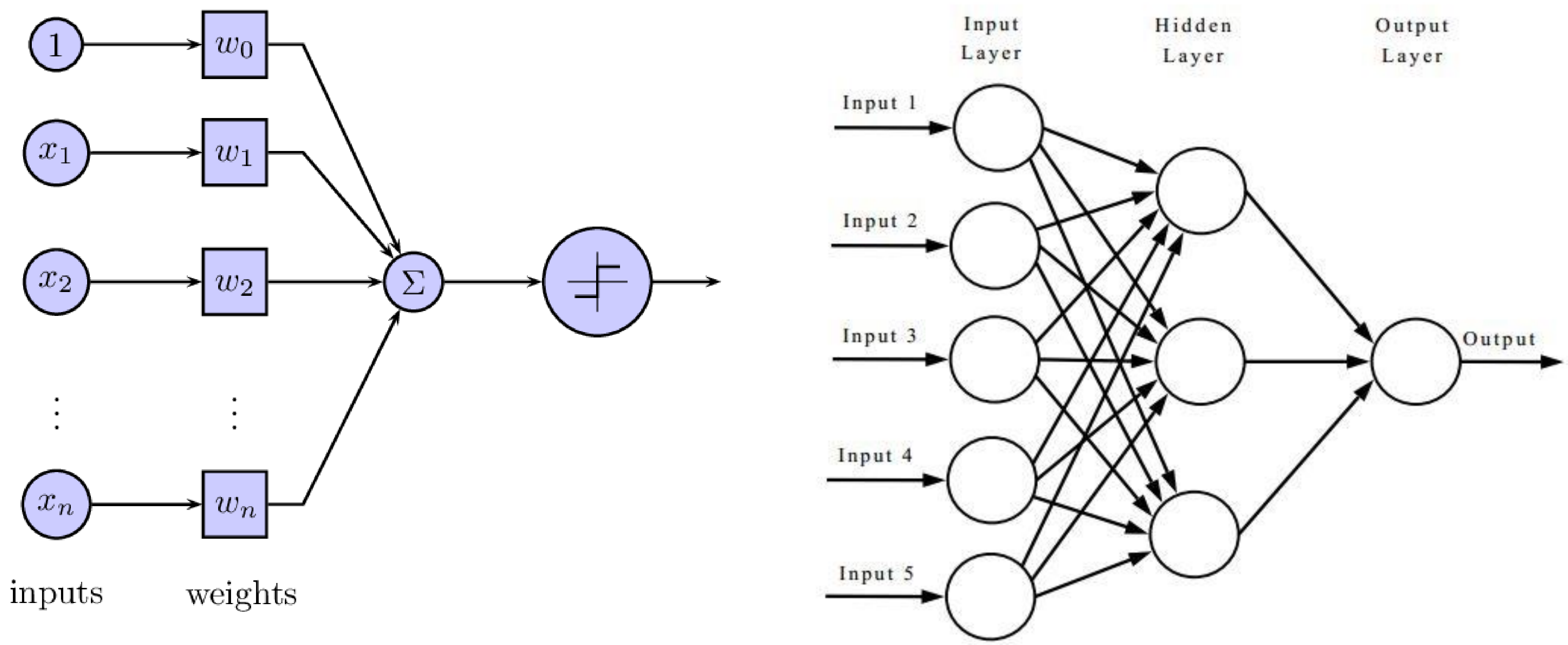


Figure 1. The model of a simple perceptron (left) compared to neural network (right). Neural networks can be thought as a multilayered perceptron due to the presence of the hidden layer. [This Photo](#) by Unknown Author is licensed under [CC BY-SA](#).

Neural networks (NN) can be thought as a **model of multilayered perceptron** with the first layer being the **input layer**, and the nodes on the first layer are connected to a **series of hidden layers** and so on until it goes up to the **output layer**. Compared to perceptron, the number of input layer is the number of features including the bias with the number of classes as the number of neurons on the output layer.

FUNCTION FITTING

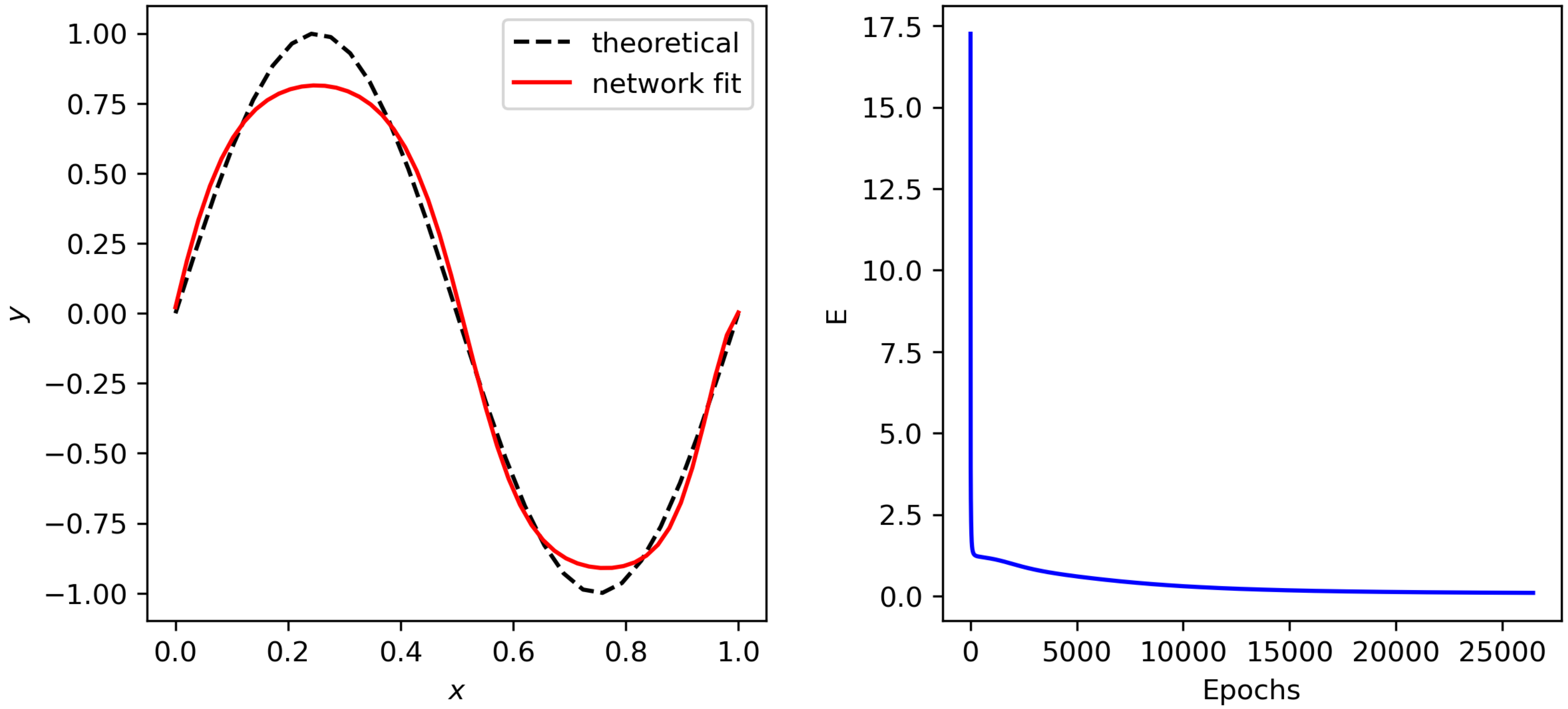


Figure 2. Neural network trained to fit a data through a sine wave function.

A neural network was designed to test it as **curve fitting model**. Initializing a sine wave across the interval 0 and 1, data points were generated from a **Gaussian distribution**. The hidden node was removed such that the NN only had the **input and output node**. A **sigmoid function** was used as the activation function, defined from -1 to 1 as the range of a sine wave is contained in this interval. The corresponding **derivative of the sigmoid** were used and the NN was able to **converge** at around 25000 epochs. The theoretical and network fit was compared and a close resemblance was obtained.

CLASSIFICATION DATASET

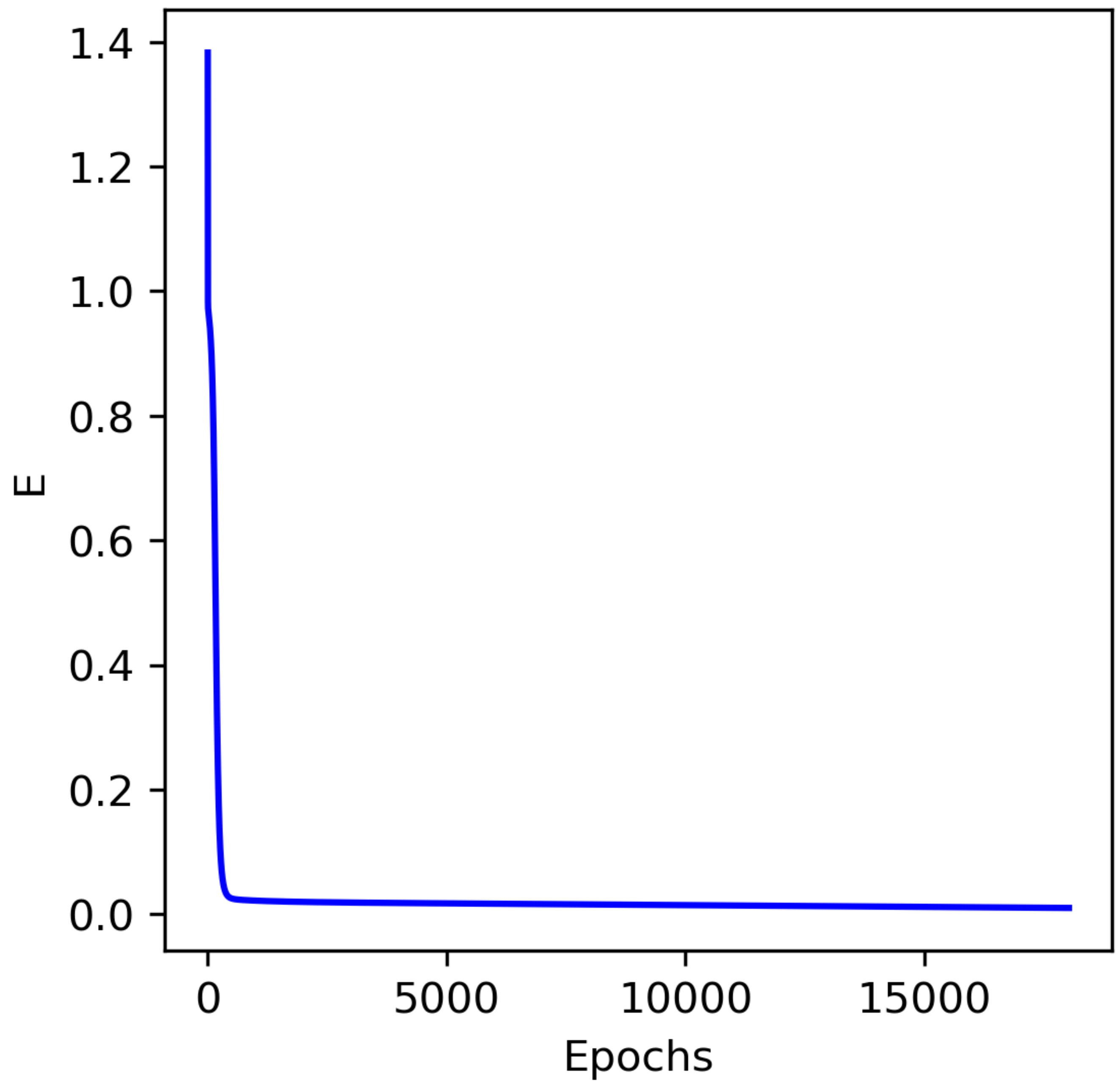


Figure 3. Convergence of the neural network as a classifier. The error loss was plotted against the epoch or iteration count.

Using the same algorithm for feature extraction on the Perceptron activity, the **mean RGB** of different flower images from Kaggle --- roses, chrysanthemums, and butterfly peas ---- were used as **training and prediction data**. The expected output of the NN was set to be at 0.3, 0.6, and 0.9, respectively. Since three features were used, **four input nodes** including the bias were built, along with **three hidden nodes**, and **one output node**. At a **learning rate** of 0.1 and **error threshold** at 0.01, the sigmoid function is optimal for the **convergence** of the data set. At around ~15000 epochs, the NN was able to converge, which implies the network was able to **classify** the given data.

REFLECTION

Understanding the neural network algorithm from scratch was challenging due to the number of mathematical implications I need to consider. I wish I could have done better in writing the code, but I was happy that I was able to grasp and execute the neural network as a curve fitting model and classifier. For this activity, I would give myself a score of **90 / 100** ! I believe there is still a room for improvement in my output given that the activity was given to us in a short amount of time ☹

REFERENCES | GITHUB

1. M. Soriano, Applied Physics 157 – Neural Networks, 2023.
2. But what is a neural network? | Chapter 1, Deep learning – YouTube
3. Iris Flower Dataset | Kaggle