Friday, 15 July

CFI, IIT Madras

Neural Networks

SRINIVASAN K

Analytics + CVI

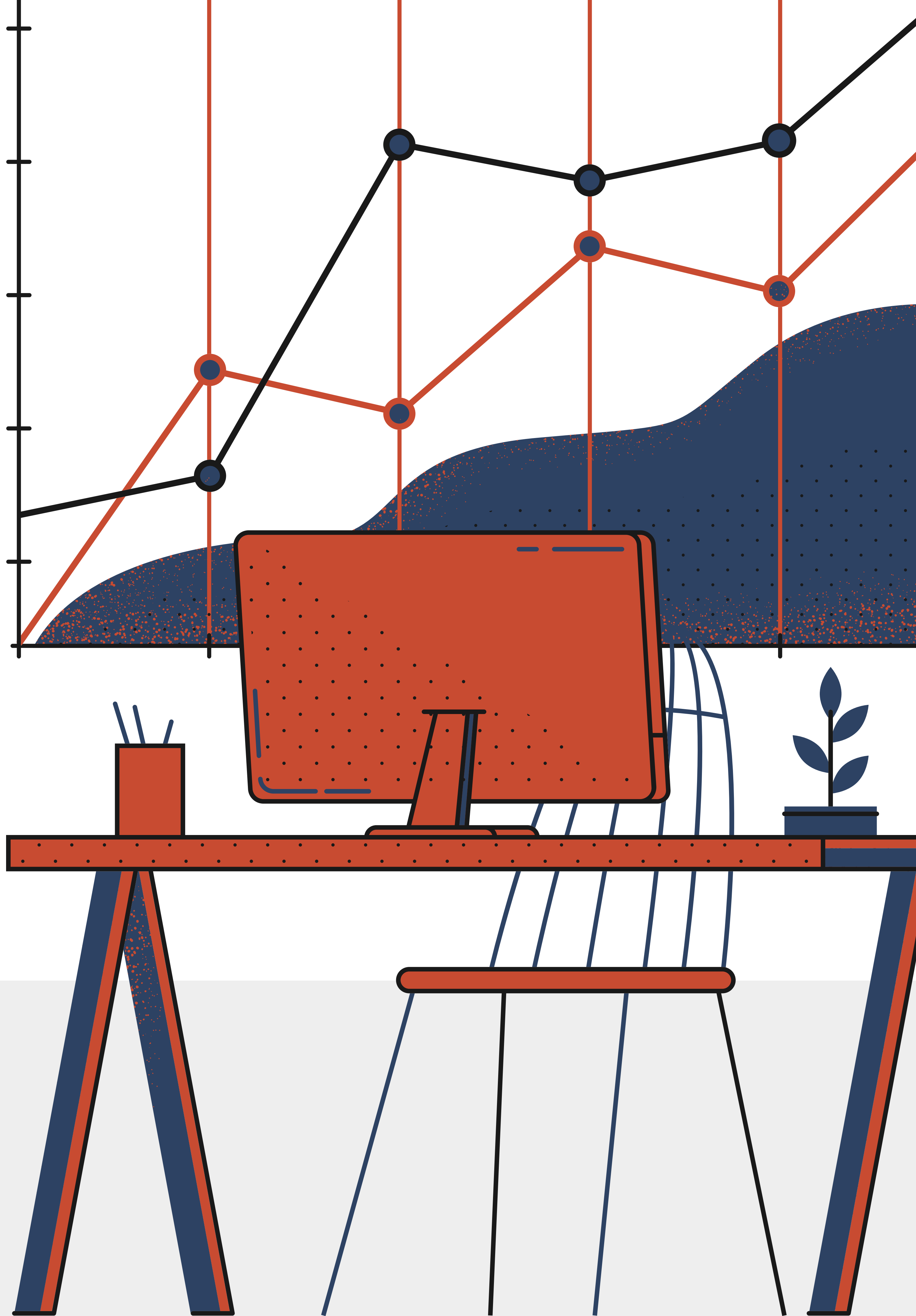


Table of Contents

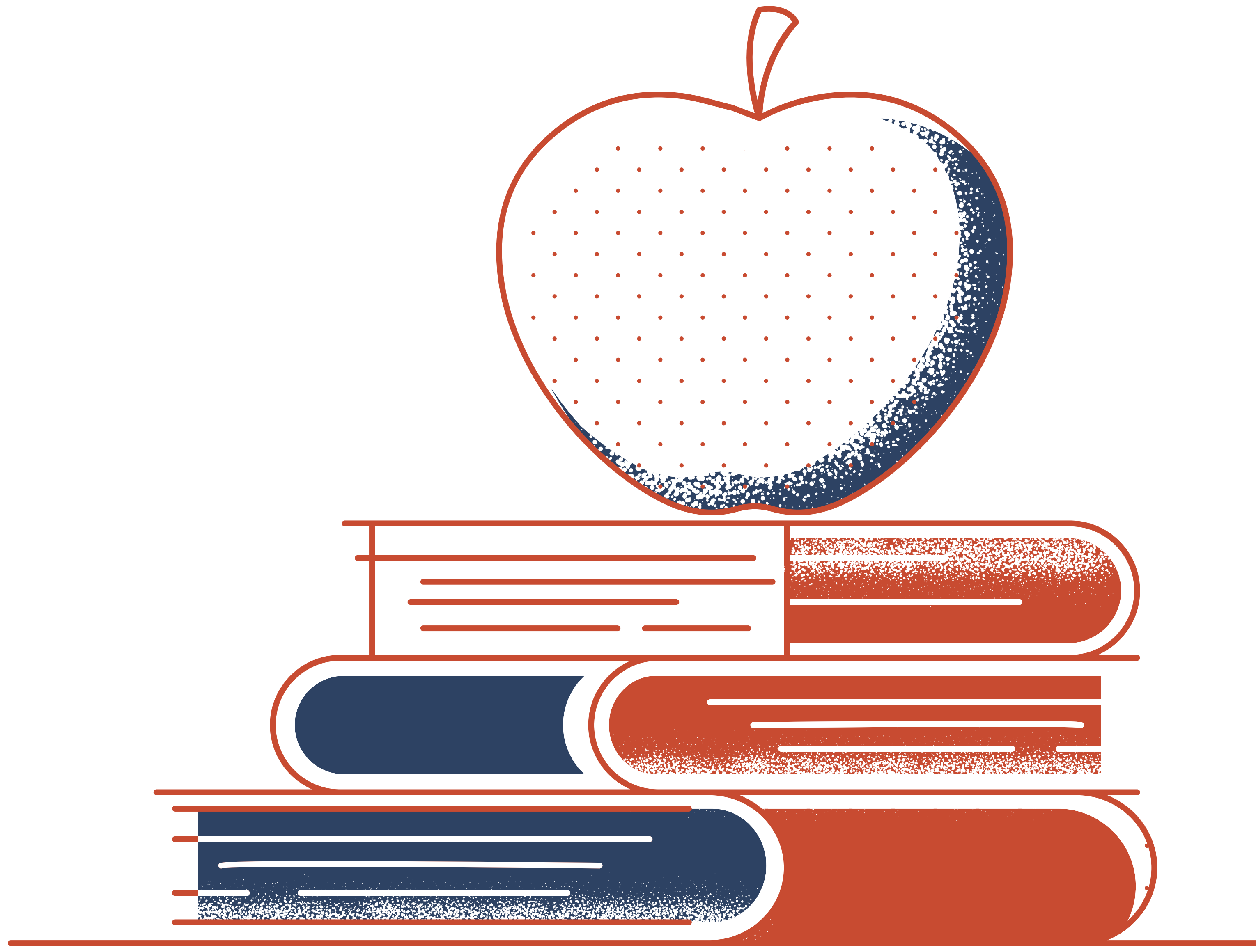
Parts of a Neural Network Code

- Initialization
- Forward Pass
- Loss
- Back Propagation
- New Forward Pass

Cost/Loss Function

A cost or loss function is what allows us to get a metric of how our model is performing. It basically finds the difference between the actual value of some parameter and the predicted value of that parameter by our network.

In our model we will use the mean squared error (MSE) loss. This is one of the most basic loss metrics and is computationally simple as well.

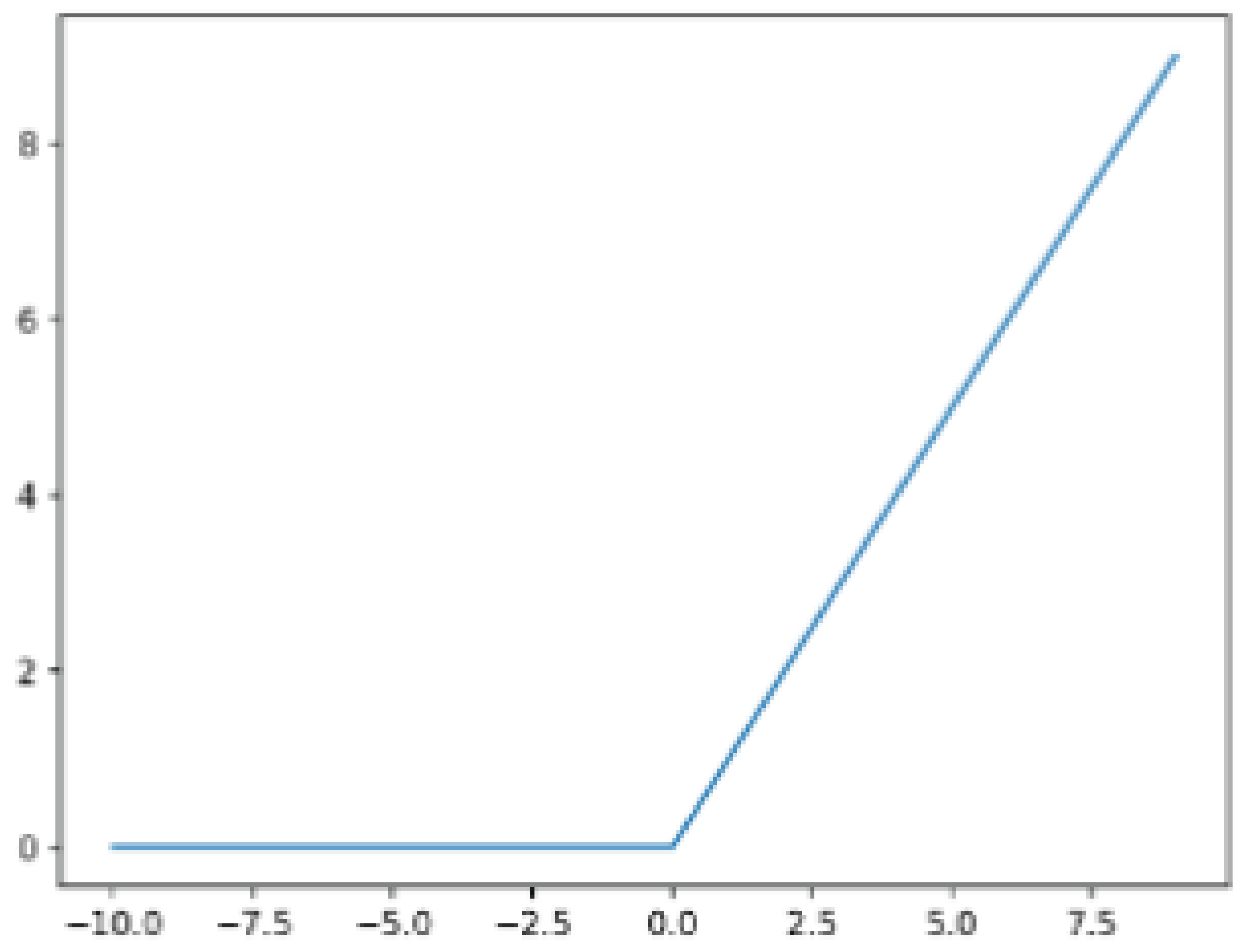


III Activation Functions

An activation function is a function applied at each neuron on the weights*input + bias to get the final output.

(1) **ReLu:** The Rectified Linear Unit function is defined as:

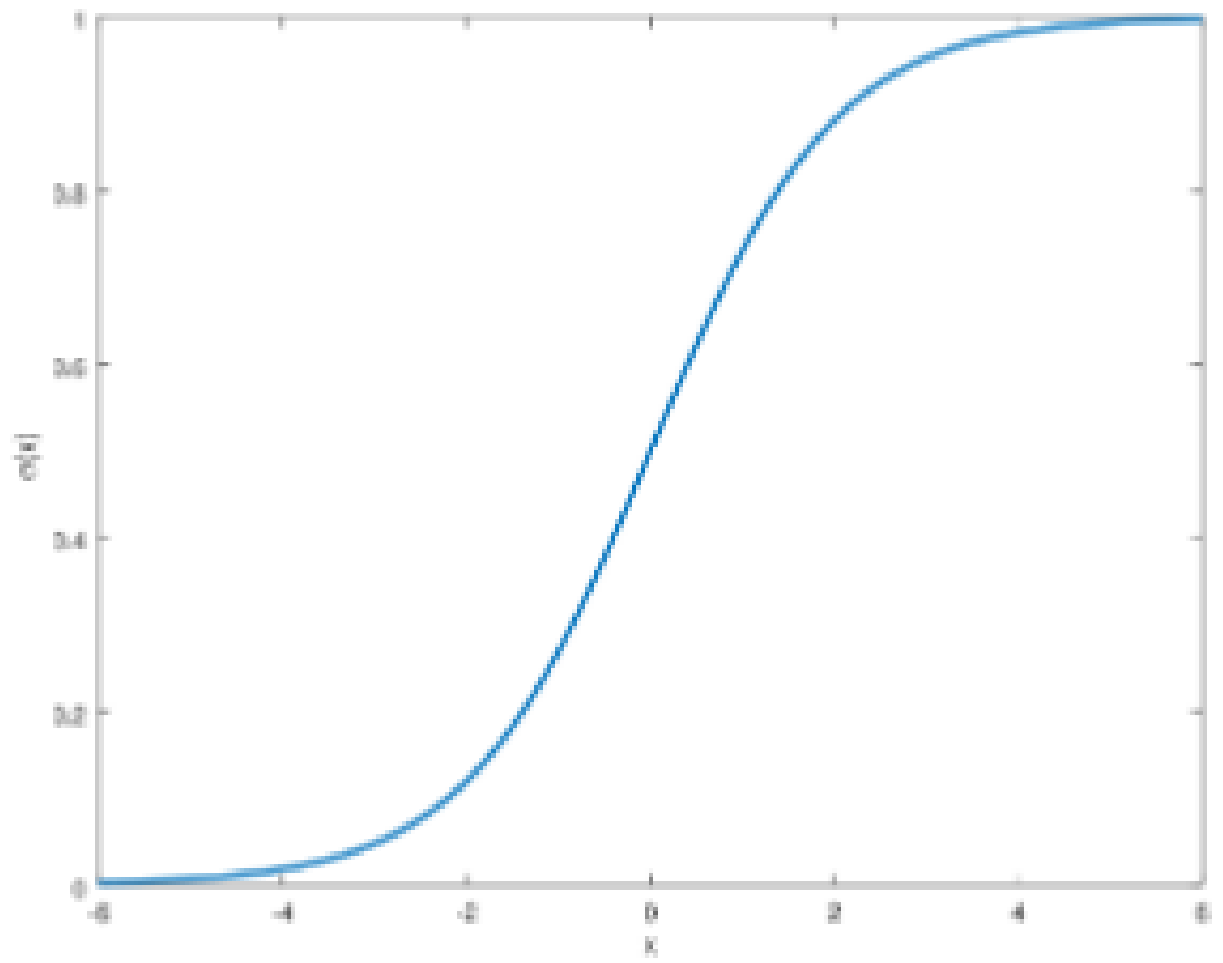
$$f(x) = \begin{Bmatrix} x, & \text{if } x > 0 \\ 0, & \text{if } x < 0 \end{Bmatrix}$$



(2) **Sigmoid:** This function is defined as :

$$f(x) = \frac{1}{1 + e^{-x}}$$

The graphical visualization of this function is given as :



Compilation of PyTorch functions

torch.rand(size, requires_grad)
Generates a tensor of random numbers of given size.

torch.relu(tensor), torch.sigmoid(tensor)
Apply relu/sigmoid activation on the tensor.

torch.mean(tensor)
Finds the mean of all values in the tensor.

tensor1.backward()
Starts back propogation.

tensor2.grad()
Find the gradient of tensor1 wrt tensor2.