Deep Learning Intro

Karel Stryczek

CWRU Quants

November 17, 2025

Overview + Goals

- Gain a broad understanding of what is happening in the training loops
- Develop a general vocabulary to discuss higher level topics

Neural Network Training Loop

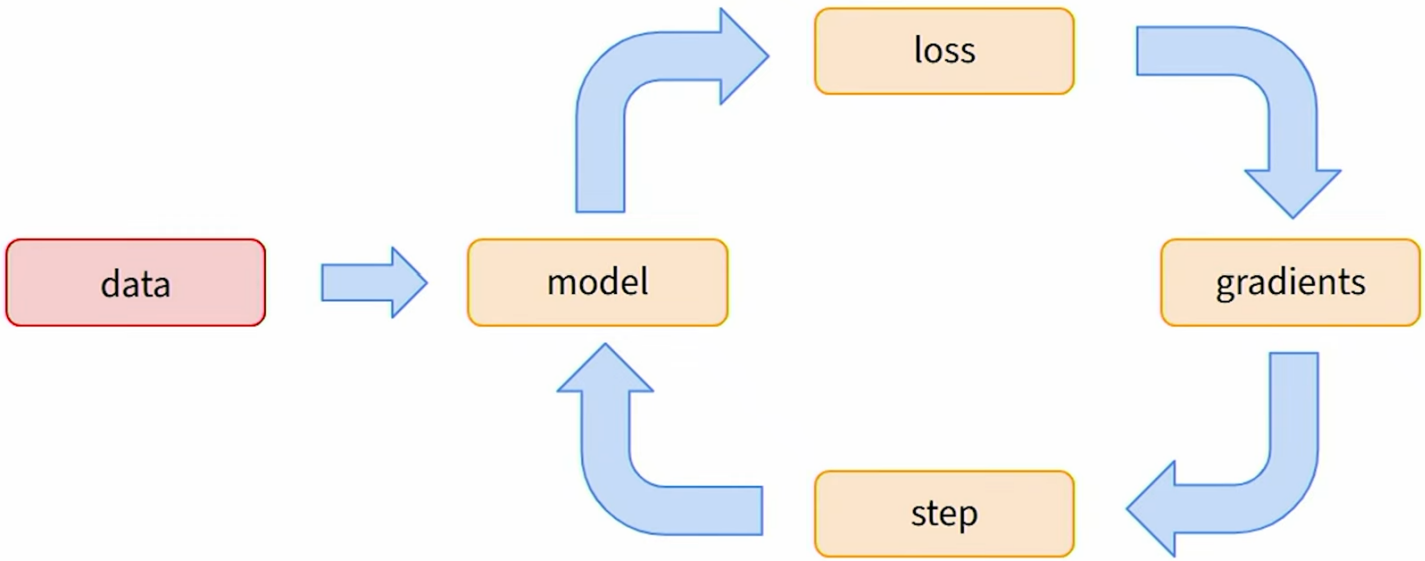


Figure: NN Training Loop (from Jane Street Tech Talk)

Initialize Your Model

- Define **layers, nodes, activation function**.
- Initialize **weights** and **bias** (randomly).

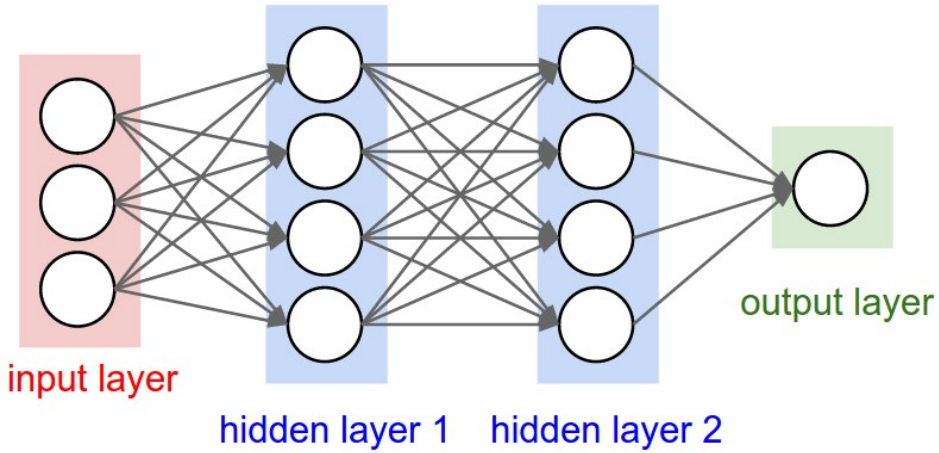
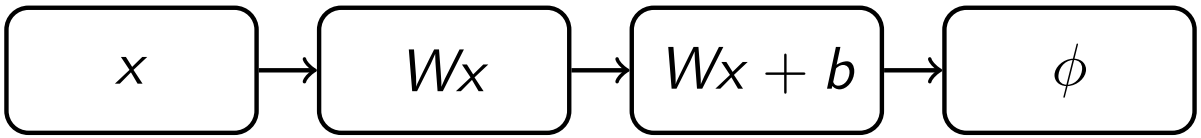


Figure: Neural Network Visualization



Define Loss Function + Finding Error

- Essentially a way to measure how *far* away your prediction is from target values.
- For our purposes MSE is quite natural, and this is where the "Physics Informed" portion comes in for us.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathcal{L} = MSE_{\text{data}} + MSE_{\text{PDE}}$$

Gradients + Updating Parameters

- To actually train the neural network, we must have a way to update the parameters of our network (weights and biases).
- We use the chain rule to find $\nabla_\theta L(\theta)$ where θ is our weights and biases, we go backwards from the last layer, calculating partial derivatives at each layer for each parameter, this is called **backpropagation**.
- Once we have collected all of these partial derivatives, use an optimization algorithm to find best values for each parameter (gradient descent shown below).
- In implementation, Keras uses **Gradient Tape** to keep track of gradients.

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$$

Completed Training Step

- Once we have completed the before process, this is **one training step**.
- Once the model has seen all of the training data once, this is one **epoch**.
- Can also think of epochs as going through all the batches, say we have 1000 samples and split them into batches of 100, our parameters would update once for each of the batches and after the model has seen all 1000 samples, this would be considered **one epoch**.

Final Thoughts

- Computationally quite expensive, CoLab solves this.
- Incorporating the PDE into the loss function.

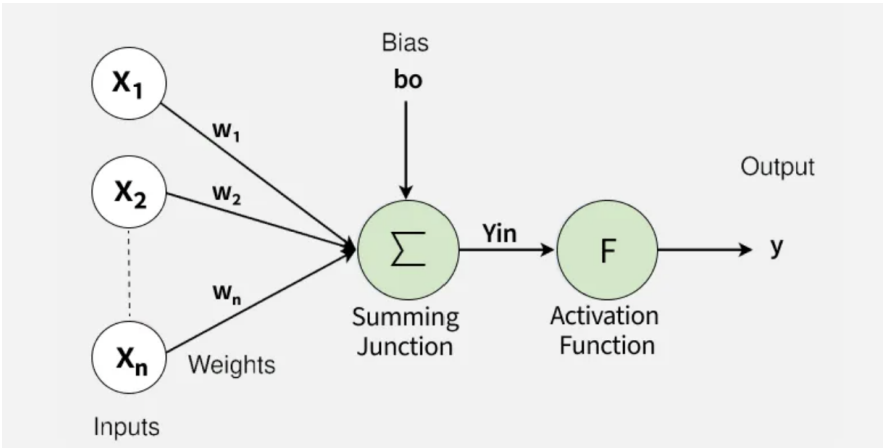


Figure: One Neuron Example