**Neural Network**

Neural Network is a part of machine learning known as deep learning. It takes its name as it takes inspiration from a human brain in its working. It is complicated network of neuron which are arranged in three levels, an input layer, an output layer, and a hidden layer. These layers also called as node layer are connected to each other by associating them with their respective weights and activation. Just like their biological counterparts these artificial neurons will also fire only when they are above a certain threshold and activation of one neuron will bring about activation in the next neuron and so on. This complicated network of neuron where each neuron is associated with its respected weight and activation is what makes a neural network such as powerful tool to predict outputs.

**Working**

Let’s take a very simple network of one input, output, and hidden layer. Now let’s associate each neuron in these layers with a weight and threshold(activation). Weight is basically telling us how important a neuron is in the network. All these inputs are multiplied with their respective weights and then summed; this output is then passed through an activation function. There are many types of activation function, the choice of it is dependent on the types of networks we want to build. Now if the output is above a determined threshold the neuron gets activated and become an input for the next neuron, in this manner one layer’s output become input for the next layer thus all the layers are interconnected to each other by these weights and activations and hence form the neural network. A final output is obtained which forms our output layer.

We use a dataset to train our model, this is the learning phase where the network makes errors and correct itself after every iteration by changing these weights. But the problem is, to make this learning effective we need a test score that we can provide the model telling it how wrong it is in its prediction. This is where the loss function comes in which is the root mean square error of the actual and predicted value. Most model minimize the loss function by gradient method and update weights via the back propagation. Since the whole of neural network is changing this weight to best fit the model, we need better optimizing algorithm to achieve this task.

**PINNs**

PINNs are an improvement on standard neural network, these are neural networks which are informed by the knowledge of physics laws followed by the dataset, described by the partial differential equation. Using this knowledge act as a regulating agent in the network which leads to better outcomes and prevent over fitting especially in cases where there is a shortage of data. By using these governing equations, we can inform our loss function to minimize it further. PINNs are essentially a neural network that follows the rule of partial differential equation.

Using this knowledge of partial differential equation, we can limit the space of outcome which helps in decreasing randomness in the network. It is the intersection of both, data driven machine learning and informed with physics laws which helps in regulation of the model to minimize the loss function due to which a high maintain level of consistency.

Most of the physical world is governed by these partial differential equations, for example fluid motion equation is a partial differential equation. With the help of this equation in our network we can reject all the flow which cannot exist hence decreasing the sample space of our output which thus increase accuracy and decrease computation power used.

Using these partial equations, we can inform our loss function given these are the equations we want our model to learn. But then why not feed our model these equations in the first place but we can’t do that since most often these equations are differential equations which are hard to solve to get a result.

<https://github.com/jdtoscano94/Learning-Python-Physics-Informed-Machine-Learning-PINNs-DeepONets/tree/main/PINNs>

https://www.youtube.com/watch?v=AXXnSzmpyoI&t=811s