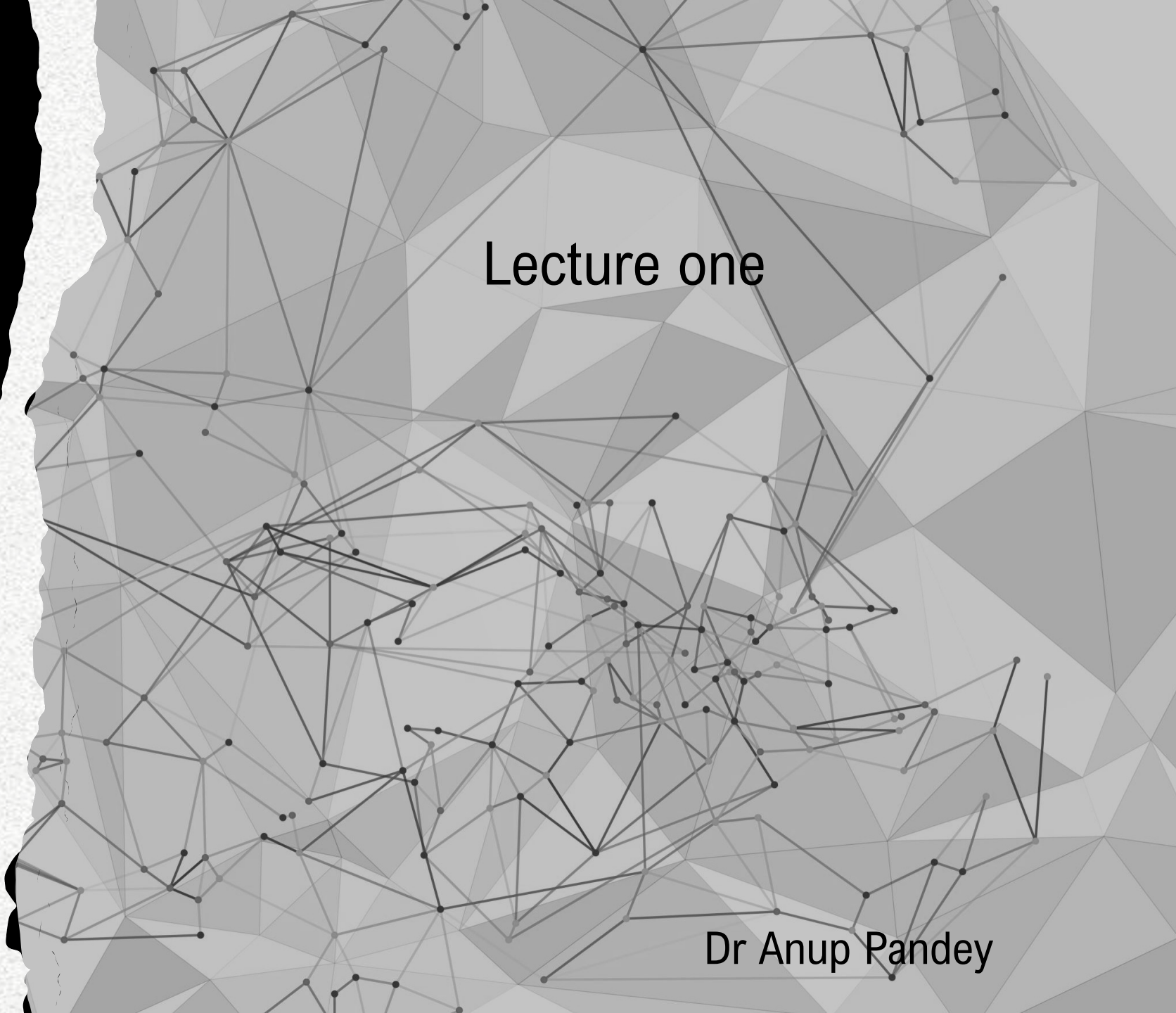
Neural Networks Lab

Lecture 1



Lecture one

Dr Anup Pandey

Neural Networks in Python

There are two ways to create a neural network in Python:

From Scratch – this can be a good learning exercise, as it will teach you how neural networks work from the ground up

Using a Neural Network Library – packages like Keras and TensorFlow simplify the building of neural networks by abstracting away the low-level code.
If you're already familiar with how neural networks work, this is the fastest and easiest way to create one.

Neural Networks in Python

No matter which method you choose, working with a neural network to make a prediction is essentially the same:

Prepare Data set(Text, number , Image, Audio file, Video files etc..)

Import the libraries. For example **import NumPy as np**

Define Neural Networks (there are sequential and Functional Networks)

Add weights and bias (if applicable) to input features. These are learnable parameters, meaning that they can be adjusted during training.

1. Weights = input parameters that influence output
2. Bias = an extra threshold value added to the output

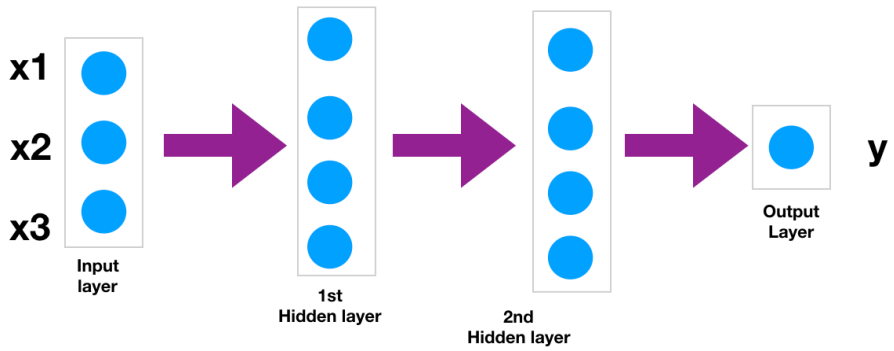
Train the network against known, good data in order to find the correct values for the weights and biases.

Test the Network against a set of test data to see how it performs.

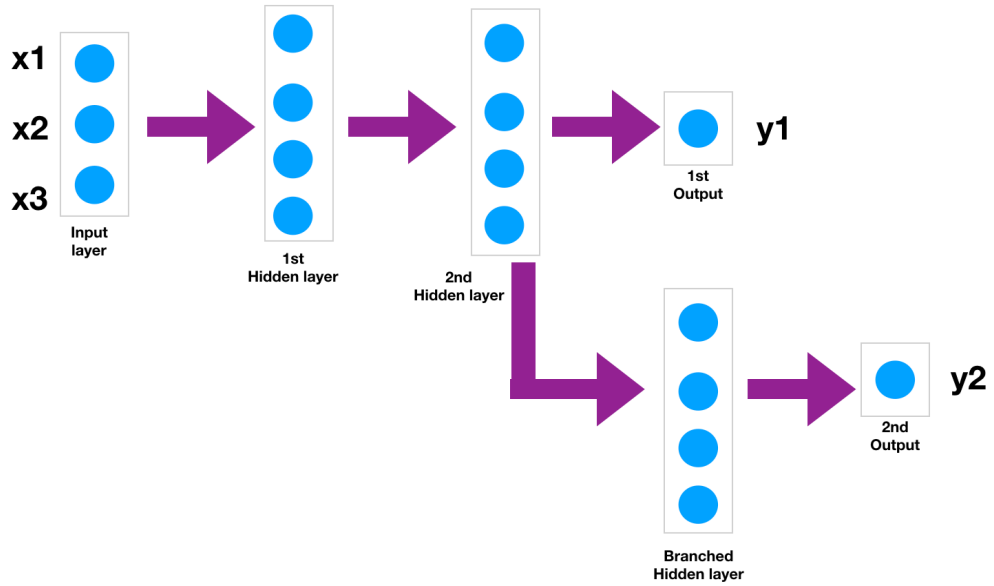
Fit the model with hyperparameters (parameters whose values are used to control the learning process), calculate accuracy, and make a prediction.

Neural Networks in Python

Sequential



Functional



Please refer to these two models

https://www.tensorflow.org/guide/keras/sequential_model

https://www.tensorflow.org/guide/keras/functional

Neural Networks in Python

Task1

Reading

Please refer NumPY which is very important to implement NN

https://numpy.org/doc/stable/reference/generated/numpy.dot.html

Please refer scikit library
https://scikit-learn.org/stable/

Please refer Keras and try to understand how to implement API
https://keras.io/api/models/

Lab

Neural Networks in Python

Task2

Students should attempt to answer these questions. The solutions will be discussed during the practical sessions.

Question 1

Hand “wire” (manually configure) a perceptron so that it computes the Boolean NAND (NOT AND) of its two inputs.

Question 2

Using the perceptron training rule, train a perceptron (with 2 inputs and one output) to learn the AND function. Assume the initial weights of your perceptron are –0.5 and 0.5, the threshold is 1.5 and the learning rate is 0.9. Make sure you train the threshold.

Question 3

Consider a perceptron with 5 inputs and one output, where all of the inputs and the output are either 0 or 1.

1. Hand “wire” this perceptron (specify the weights and the bias) so that it computes the “majority” function for the 5 inputs.That is, it should output 1 if at least half of the inputs are 1 (in our case 3), and 0 otherwise.
2. The same problem, but computing the “minority” function for the 5 inputs. The output is 1 if not more than half of the inputs are 1 (in our case 0, 1 or 2 inputs high), otherwise 0.
3. Generalize for n inputs, n an odd number.

Lab

Neural Networks in Python

Task 3

Please download the jupyter notebook and understand basic implementation of Perceptron

https://files.coventry.aula.education/67f53523c6167a60e990f971e1ce2d09perceptron_or_xor_and.ipynb

The file is in the same folder