

Unsupervised learning:

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabeled data.

Ex: K meaning clustering

Scalable:

the idea is to be able to take one piece of code and then throw any number of computers at it to make it fast.

Trackable:

Easily handled.

Discrete Probability:

A discrete probability distribution is made up of discrete variable. Specifically, if a random variable is discrete, then it will have a discrete probability distribution.

Ex:

Game 1: Roll a die. If you roll a six, you win a prize. (discrete)

Game 2: Guess the weight of the man. If you guess within 10 pounds, you win a prize.
(continuous)

Here we use discrete probability because we have to predict random pixel output (raw pixel).

CNN:

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see $h \times w \times d$ (h = Height, w = Width, d = Dimension). Eg., An image of $6 \times 6 \times 3$ array of matrix of RGB (3 refers to RGB values) and an image of $4 \times 4 \times 1$ array of matrix of grayscale image.

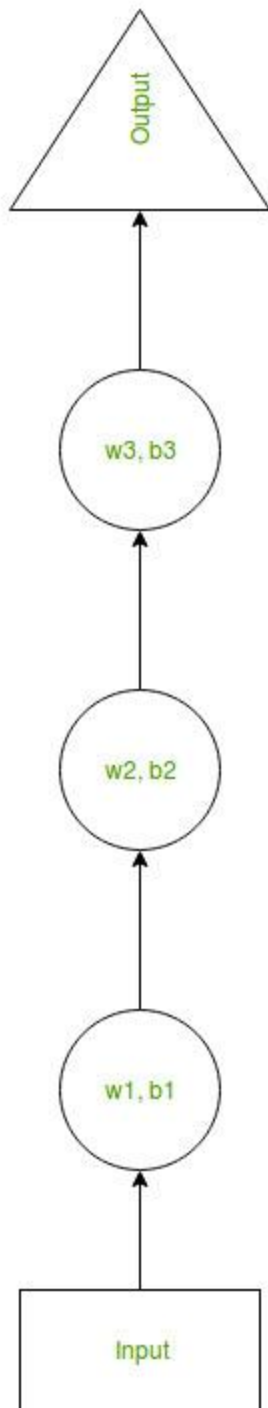
LINK: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

RNN:

Recurrent Neural Network(RNN) are a type of Neural Network where the **output from previous step are fed as input to the current step**. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is **Hidden state**, which remembers some information about a sequence.

Ex:

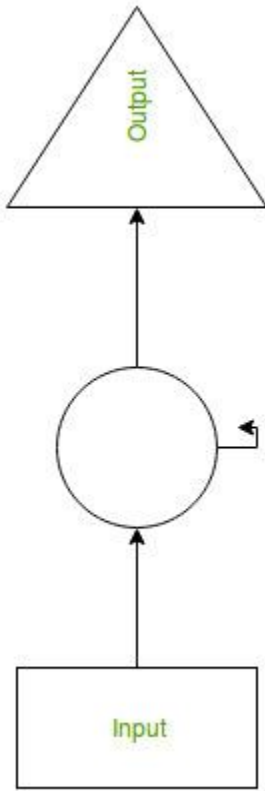
Suppose there is a deeper network with one input layer, three hidden layers and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w_1, b_1) , (w_2, b_2) for second hidden layer and (w_3, b_3) for third hidden layer. This means that each of these layers are independent of each other, i.e. they do not memorize the previous outputs.



Now the RNN will do the following:

- RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

- Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.



Link: <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>