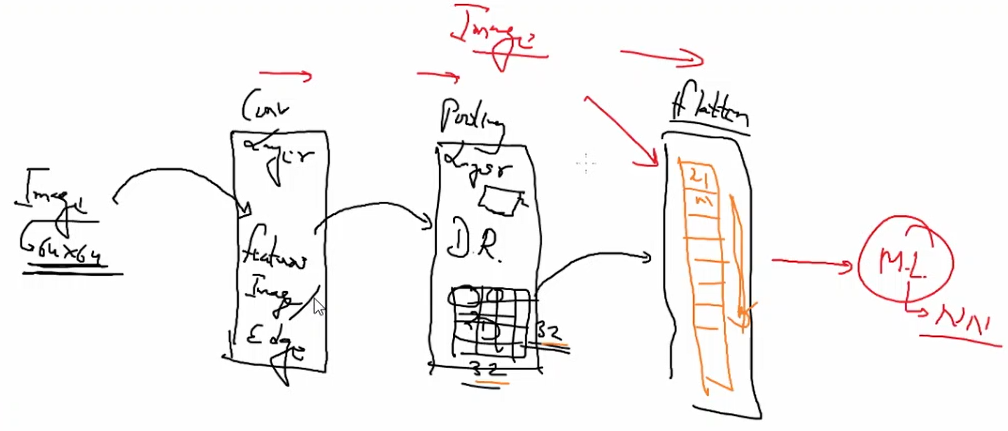
Session-178 CNN

* Computer is designed for working with numbers & mathematics.
* For training our model we need numbers.
* So first of all we have to convert images/ videos/ strings into numbers.
  + We have some libraries like numpy and cv2, it will covert binary file of image video into array (Numbers).
  + We can use cv2.imread for converting image to array.
  + In normal HD image is 1024\*1024 pixels = 1048576 pixels.
    - So we have to get weight of this much features.
  + 1 complete image = 1 record.
  + If you use normal approach we have to use huge number of amounts to train this type of data.
  + So if you see we have 1048576 dimensions.
    - Our moto is to reduce this dimension.
    - We also say image is 2D, 3D but this is with respect to color of image.
    - Gray, and all that.
  + But if you reduce dimension then you may lose some data, so it will affect accuracy.
    - We have to also check with respect this.
* All of this thing we can achieve with convolution neural network (CNN).
  + When you something with human eye, there we also loose many information.
  + Most probably we only focus on the main objects, like persons.
  + So we may use data but we have to make sure losing data will not effect much to our final y.
  + For learning anything we have to detect edges.
  + FL is also about detecting edges.
* For detecting edges we use concept called filter in CNN model.
  + Suppose we have group of pixels, and we are taking 3\*3.
  + It has same color so it is look like there is no edge here.
  + Lets move box by 1 column.
    - Here we also don’t have different colors, there is little bit change in shade but not very much changes.
    - Lets move by one more column.
  + In programming world this moving is known as **stride**.
    - Now we have sudden color change in the box. So we can know here we have edge.
    - By this process we can detect edges and edge finally gives feature of object.
    - Smaller box requires more time (more stride) to detect the edge.
    - Suppose we have 1000 pixels and we are using 3x3 box & 1 stride so we have to move 998 times.
  + In market standard filter is 3x3 box.
    - In bigger box we may miss some small edges.
    - Here filter is not image part, eg we are using 3x3 filter then it is matrix of it and we multiply this filter value with image part value (image 3x3 matrix).
    - If sudden change in output numbers, then we can say there is an edge.
  + This method of detecting edges is known as convolution.
    - Filter is also known as kernel.
  + Convolution (convolve layer) only meant for detecting edges they cannot help you for reducting dimension.
* If you have colorful image then it will take more time.
  + So most of time before processing we convert image into black and white.
  + Converting image we don’t loose data, because we just working with edges here.
* For reducing dimension we use pool layer.
  + Here let’s take 2x2 matrix, if all 4 pixels value is same then instead of 4, we can use 1 pixel.
  + For this we have to use pool layer.
* First we use convolve layer then we use pool layer, we keep on using this layer multiple times.
  + Now after doing all we have 2D data but in NN we require 1D input.
  + So we have to convert 2D into 1D.



* + - We can also use multiple time this setup before using flatten for converting 1D.
  + Here we use term called fully connected (FC) layer (instead of NN layer).

Session 179 CNN

* What is convolve.
  + Convolve layer auto create filter for you.
  + We have passed 32 filters in last lecture.
  + So total 32 different filter, we hit and try to check which one is good for our input set.
* we can use **scipy** library, for doing some and try with convolution.
  + They auto come with one default image.
  + We can also create filter manually using numpy array.
    - f = numpy.array([
    - [1,2,1],
    - [0,0,0],
    - [-1,-2,-1]
    - ])
  + If you focus after convolve layer size of img increase by some pixels.
    - We are using filter of 3x3, so for last column it will automatically add to more column for mathematical calculation.
    - We can also resist this thing.
    - This extra space is known as padding.
  + Normally convolve, pooling and flattening layer work together and combinelly known as feature learning.

Session 180 CNN

* Complete image is one single record, but for accuracy we need lots of record, so we need lots of images.
* We can download image dataset from ImageNet website.
* The standard image size is 64x64.
  + But you can use any size of image.
  + Small size faster change.
* We have used cat-dog dataset.
  + Here we have 8000 training image & 2000 testing image.
  + Out output is eighter cat or dog so we have to use **sigmoid** function in last layer.
  + Optimizer is adam and loss function is binary\_crossentropy, metrics is accuracy.
* Here we have small number of images but in real world we have lots of images.
  + We can also generate multiple image from single image.
  + Scaling, rotating and other methods.
  + We use this thing very much so in keras we have function for data generator.
  + Here we are using function called ImageDataGenerator.
* Now we have to run our NN.
  + In fit function we can directly pass testing dataset.
* If you want to load single image you can use image module from keras.preprrocessing.
  + We have created one more folder and there we have add some random image.
* In image world instead of predicting we want object detection.
  + But all the function are same.

Session 181 CV

* Instead of CNN we are going to use some traditional approach for real time face recognition.
  + We are going to use algorithm called LBPH (local binary patterns histogram).
  + Pip install opencv-contrib-python
    - This library provides LBPH algorithm.
* Using harcascade and opencv we will capture the live image.
  + LBPH train image.
  + And we can write any function to work in our last phase.
  + Eg open linkedin profile.

Session 185 –