CNN – Convolutional Neural Networks

* CNN is a deep neural network. Convolutional is the name of the process.
* Single neuron focuses on one part of the network.
* Neurons closed to input takes care of the linear part eg: lines in an image
* The more we go to the higher layer the image formation will be there
* When we see images they are coloured images, but animals may or may not see colour, coloured imaged is called channelled images and black and white – grey-scale.
* NN can create 1000s of channels. Then it may detect certain shapes.
* When cnn sees an images, it sees it in a vertical manner. It finds the pattern in vertical manner so there are 1000s layers stacked on each other.
* A nn that use convolution in place of general matrix multiplication. i.e cnn does not does matrix multiplication.

Convolutional

* It is a mathematical operation on two functions (f and g) that produces a third function (f\*g) that expresses how the shape of one is modified by the other.
* The term convolution refers to both the result and the process of computing it.
* The convolution operation is typically denoted by an asterisk (\*).
* F(x) \*g(x)
* S(t) = (x\*w)(t)

2D convolution

* Stride: how many blocks are we shifting at a time, normally we shift it one time.
* Vertical Stride: when we shift the block downwards
* This is not matrix multiplication.
* Useful when we are dealing with images.
* Use cases:
  + Pattern recognition
  + Object detection
  + Click an image of a flower, phone can recognise the flower and will show similar flowers.
  + Image processing – satellite images
  + Medical images – pattern recognition – ct scan of brain
* Neurons closer to the image focuses on a particular region of the image.
* Lower layer misses many features and keeps only the ones they think are important.
* Some neurons may think some features are important that other may not.
  + Eg: one person may notice what people were wearing and other might not, so here clothes become an important feature to that person so their neurons work that way, while other person may not notice it.
* Neurons that fire together, wire together.
* Why to use:
  + Convolution leverages 3 important ideas that can help improve a ML system
    - Sparse interactions
    - Parameter sharing
    - Equivarient representations
* REFER IAN GOODFELLOW BOOK
* Sparse: one neuron is connected to some other neurons of the first layer
* Dense: one neuron of 2nd layer is connected to all neurons of the first layer
* Connections: Sparse <<<<< Dense
* Receptive Field: Sparse <<<<< Dense
* Parameter Sharing: Dense <<<< Sparse
* Using sparse connection is very efficient (less resources and time) than dense.

Architecture

* Cnn consists of visible and hidden layer:
  + Input layer – not a layer
  + Output layer
  + Hidden layer – as user only sees input and output layer (visible layer)
* Middle layers are hidden layers because their input and output are masked by the activation function and final convolution
* It performs a product of the kernel with input matrix. This product id <A,B> and activation is ReLu
* As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer.
* Kernel modifies the image.
* Feature map == o/p after convolution
* Feature map is different in terms of values, time, channels, size
  + i/p 3 channel image and o/p 10 channel image
* this is followed by other layers such as pooling layers, fully connected layers, and normalization layers.
  + After pooling image is smaller than feature image
  + Pooling is not mandatory
  + Pooling = summarization
* In cnn, the input is a tensor with a shape: (number of inputs) \* (input height) \* (input width) \* (input channel).
  + Normally images are 4d images. Index in 4d = 4
  + In case of MNIST, shape 🡪 (70,000 x 28 x 28 x 1)
* After passing through a convolution layer the image becomes abstracted to a feature map, also called an activation map, with the shape: (number of inputs) x (feature map height) x (feature map width) x (feature map channels).
  + Number of inputs will not change
  + Feature map channels is not equal to total channels
* Convolution layers involve the input and pass its result to the next layer, this is similar to the response of a neuron in the visual cortex to a specific stimulus
* Each convolution neuron pre-processes data only for receptive field.
* Receptive filed are defined portion of space or spatial construct containing units that provide input to a set of
* Kernel size will decide the receptive field of this neuron.

FILTERS/KERNELS

* They are also called feature detectors.
* We should be careful with what filters are we applying cause then features will be lost.
* We don’t have control over the value of filters.
* We don’t initialize a filter while training our deep learning network.
* Eg: to an original grey scale image after applying vertical filter we only get vertical lines of the image.
* It is not advised to have large filter as some features can go undetected, thus we take small filters.
* Image is the matrix and filter/kernel is also a matrix.
* Right sobel filter – size 3x3
* While performing stride – priority horizontal shift (column stride).
* With original matrix, apply filter and stride to get feature map.

Important formula for exam

* Output height = (input height – filter height / row stride) + 1
* Output width = (input width – filter width/ column stride) +1
* Input matrix = 13 x 8
* Filter = 3 x3
* Output with feature map = 11 x 6
* Let stride value = 1
* Apply formula and check

Shapes with 3 filter and 2d image

* We have 3 filters RGB and image is grey scale. So while performing convolution of blue with image, it gives us a feature map of blue, similarly with R and B.
* 720 x 960 x 3 (i/p) and 3 x 3 x3 (filter) – so the last one should be same 720 x 960 x 5 and

3 x 3 x5.

* 720 x 960 x 3 and n channels of 3 x 3 x 3 🡪 718 x 958 x N
* Image can change with layers.

Image

* 198 in image as stride is 1 thus subtract 2 according to formula
* Number of kernels become number of channels.
* 1000 x 198 x 198 x 32 🡪 1000 never changes images no.
* 64 x 3 x3 x 32 – 3 x 3 ke 32 channel wale 64 kernels hai, for this we need GPU
* For cnn, we need predicted o/p instead of matrix.
* After certain iterations we need to come back to the nn and that too without loosing any features. To get that predicted output 🡪 number
* FC Layers 🡪 the stage of converting the cnn to 2d layer then o/p layer will produce the images with their classes.
* FC 🡪 fully connected layer 🡪 abhi tak cnn me sparse connections ho rahe the now we o back to the conventional nn thus this is fully connected.
* 196 x 196 x64 will be collapsed to a single image this 2D o/p aega.