Convolution Neural Network

Have you ever thought how facial recognition works on social media?,How object detection is done in self driving cars?,How object detection is done in OCR?,How image classification is done in google photos for all of the pictures you have?.

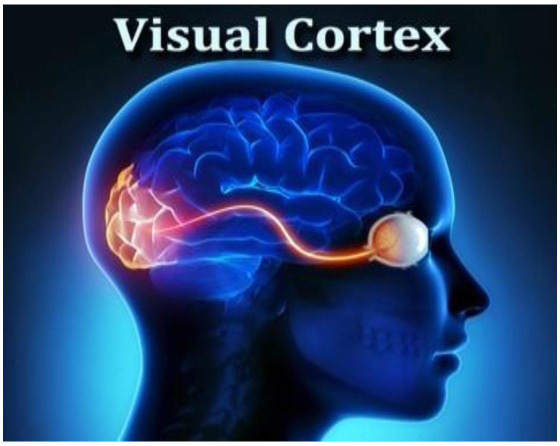
Its all possible , thanks to **Convolution neural network[CNN]**. In this assignment we will learn about essential components that make up a typical Convolution-Neural-Network.This will lay the foundation for you to learn Advanced CNN Architectures which are currently being used by Tech-Giant Companies.

For a quick recap of Neural Networks , Follow my Neural Network Series uploaded on my github-profile.

1. How Image Recognition is done in Human Brain?

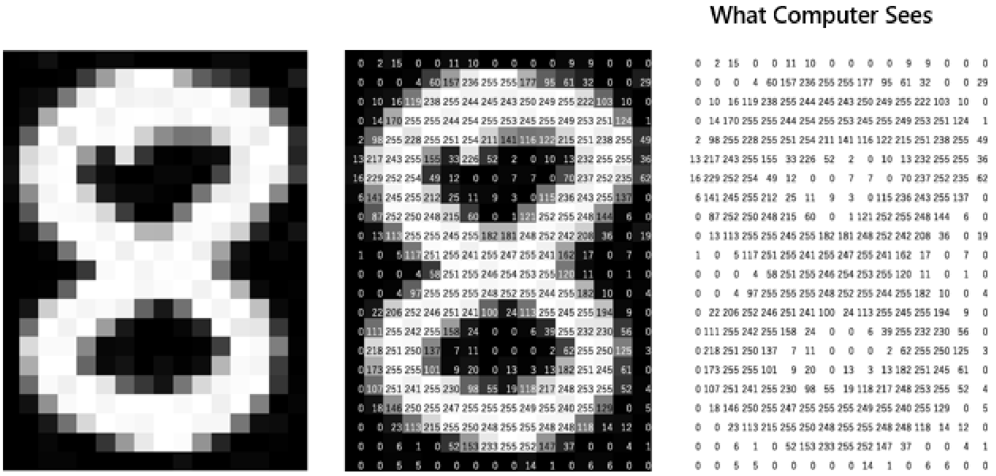
Human Brain consist of Cerebral Cortext which contains **Visual Cortex**.It’s responsible for seeing images/visuals and extracting information from images.

So whenever you see a Cat/Dog through eyes.It reaches to visual cortex via neurons.Visual cortex contains multiple layers like [v1,v2,v3…vn].Each of this is responsible to extract the information from images.Eg– v1:Responsible for finding edges of images[Ear/Nose Edge of cat],v2:Extracting information from face and so-on.Based on this we are then able to recognize that given image is cat/dog.Same happens in **CNN.**



2. How Computer ‘see’ Images?

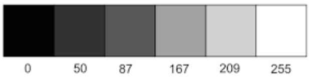
For Computer,its all about numbers.Every image can represented in 2D-or-3D

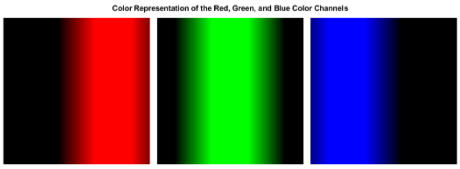


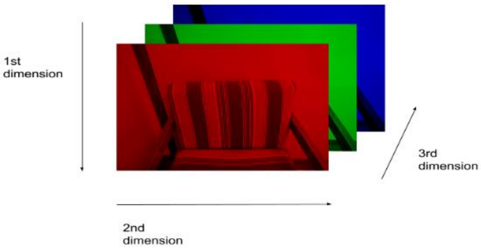
0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

Images can be Gray-Scale or Color(RGB).

Above image is Gray Scale.Each pixel value in Gray-Scale Image[2 Dimension] ranges from 0-255.Whereas each pixel value in Color-image[3Dimension] is combination of 3channels (RED,Green,Blue)




Color Representation of the Red, Green, and Blue Color Channels



3. Convolution Neural Network.

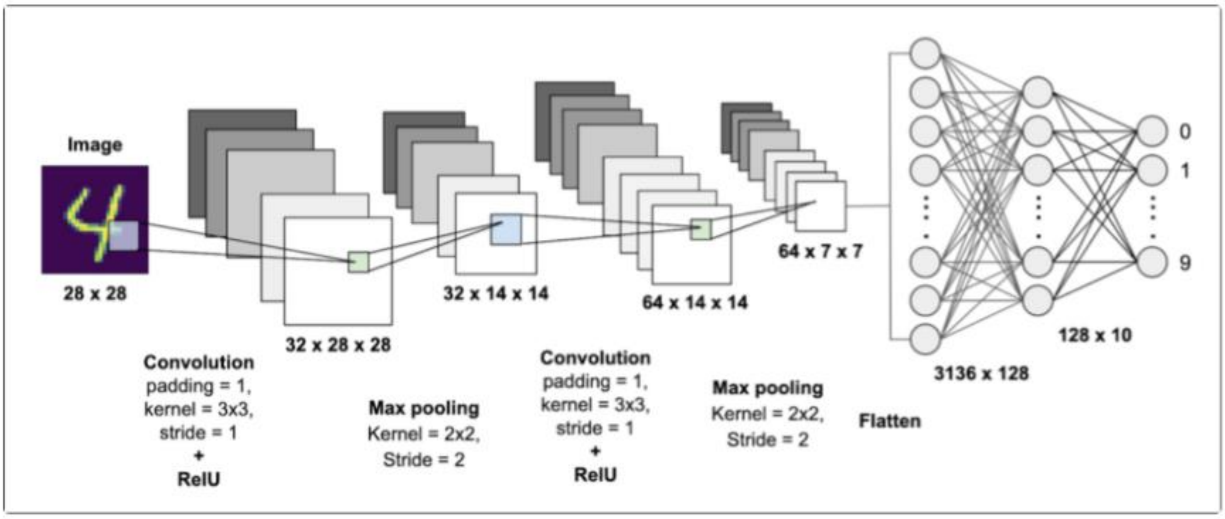
CNN is basically a Deep Learning algorithm that takes Images/Videos as its input and by process of learning able to differentiate one image from another.This can be achieved by changing parameters of the model. Following is the Detailed Architecture of CNN. Hmmmmm ,wait what are neural networks?,How models learns parameters?.Have a look in my Neural Network Series.

Seeing below architecture,You might get scared and confused .Don't worry,this assignment is made for beginners who are scared to learn about CNN.We will decode whole architecture by understanding the components of CNN.Which will make your fundamentals strong for Advanced CNN Models.

By Understanding below concepts will open your eyes about how fascinating neural network is.So without further ado,Brace yourself for the below concepts for understanding CNN.

1. **Convolution Layer–Convolution-Operation,Filters,Padding,Stride,Output Dimensions**
2. **Activation Functions–RELU,Softmax**
3. **Pooling Layer--Max/Max pooling,Average Pooling**
4. **Flattening Layer and Fully Connected Layer**

Gray-Scale Image is used to understant CNN,But same process is applied for color image as well



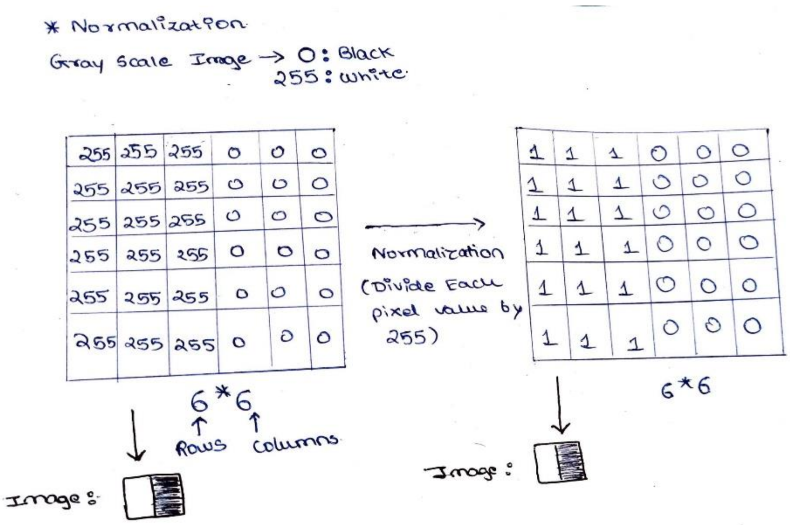
Representation of CNN Architecture

Normalization of Image inputs .

Why Image Normalization is important before training neural network ?

Data normalization ensures that each input parameter (pixel) is in same scale **(0-1).** This makes convergence faster while training the network.Data Normalization is done by dividing each pixel value by 255.

Image in Right represents how the Normalization is done on gray-scale image.In Gray-Scale:0 represents Black,255 represents white color.



* Normalization

Gray Scale Image → 0: Black
255: white

255	255	255	0	0	0
255	255	255	0	0	0
255	255	255	0	0	0
255	255	255	0	0	0
255	255	255	0	0	0
255	255	255	0	0	0

6*6

Normalization (Divide Each pixel value by 255)

1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0

6*6

1.Convolution Layer

As we know,Human brain consists visual cortex that has layers[v1,v2..vn]which is responsible to extract information from image.

Similarly we will create **Filters** in CNN.So these layers[v1,v2..vn] will act as filters in CNN.Eg:If we use Vertical Edge detection Filter then that filter is able to identify vertical edges in that image.Filter is a matrix of values called weights that are trained to detect specific features.

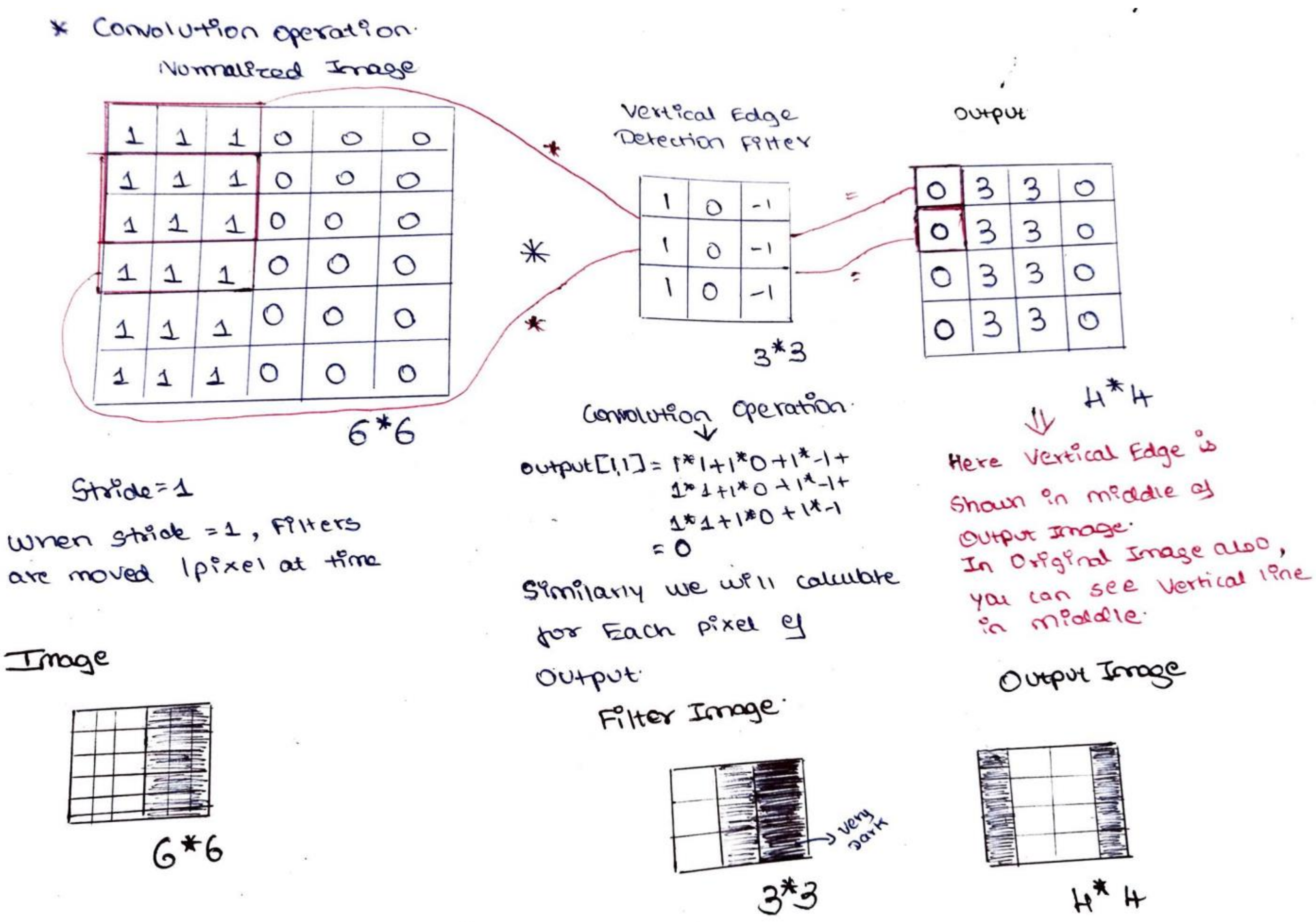
Convolution Operation

Filter moves over each part of the image to check if feature that is meant to detect is present or not.Filter carries out Convolution operation[Element wise product and sum between two matrices].

Strides

Stride is number of pixels shifts over the input matrix.When stride is 1,we move filter to 1 pixel at a time,When stride is 2,we move filter to 2 pixels at a time.

Understanding in better way by using our normalized image.



* Convolution operation.

Normalized Image

1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0

6*6

*

Vertical Edge Detection Filter

1	0	-1
1	0	-1
1	0	-1

3*3

=

Output

0	3	3	0
0	3	3	0
0	3	3	0
0	3	3	0

4*4

Stride=1

When stride =1, Filters are moved 1pixel at time

Convolution operation.

output[1,1] = 1*1+1*0+1*-1+
1*1+1*0+1*-1+
1*1+1*0+1*-1
=0

Similarly we will calulate for Each pixel of Output

Here Vertical Edge is Shown in middle of Output Image.
In Original Image also, you can see Vertical line in middle.

Image

6*6

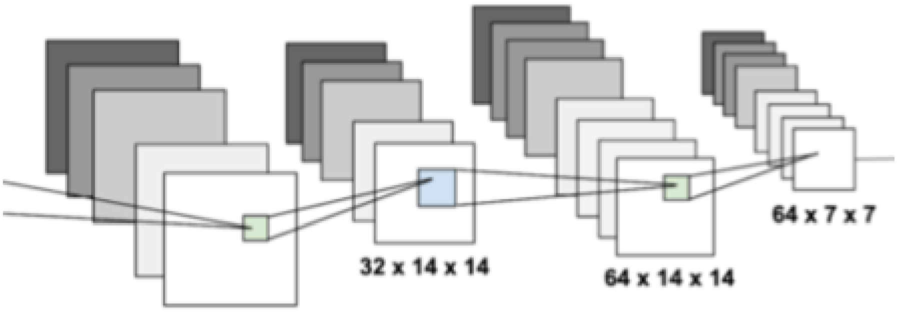
Filter Image.

3*3

Output Image

4*4

Multiple filters are used to extract information from image by doing Convolution operation.Therefore there are multiple channels in each layers.



1^{st} Dimension in each Layer is Number of Filters

Calculation of Convolution Operation when no padding and stride=1.

To Get output Dimension of convolution operation use Formula:

$$\boxed{n - f + 1}$$

n : No of Pixels in Row/column

f : Filter Size

For Example

In above image :

$n = 6$ [No of pixels in I/p image]

$f = 3$ [Filter size 3*3]

$\therefore\ 6-3+1 = 4$

$\therefore$ 4*4 Output Dimension.

Output formula changes when we use Padding and Stride.

Padding

When we perform convolution operation, Output Dimension of Convolution operation is Less then Input Dimension i-e we are Loosing Information

To avoid that Padding can be use.

Padding can be applied on I/p Image How to Calculate Padding value?

$\therefore$ Now we know, that our Output Dimension should same as I/p i-e 6

Let $n - f + 1 = 6$

$n = 6 + f - 1$

$n = 6+3-1$ [Filter Size = 3*3]

$\boxed{n = 8}$ i-e We will add 1 Row at each side & 1 column

i-e I/p Image Dimension → (8*8)

Two Types of Padding

① Zero Padding - Insert 0 in cells

② Neighbour Padding

Insert Nearest Neighbour value

* we are using Zero Padding

$$\begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix} * \begin{bmatrix} 1&0&-1 \\ 1&0&-1 \\ 1&0&-1 \end{bmatrix} = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&3&3&0&0 \\ 0&0&3&3&0&0 \\ 0&0&3&3&0&0 \\ 0&0&3&3&0&0 \\ 0&0&0&0&0&0 \end{bmatrix}$$

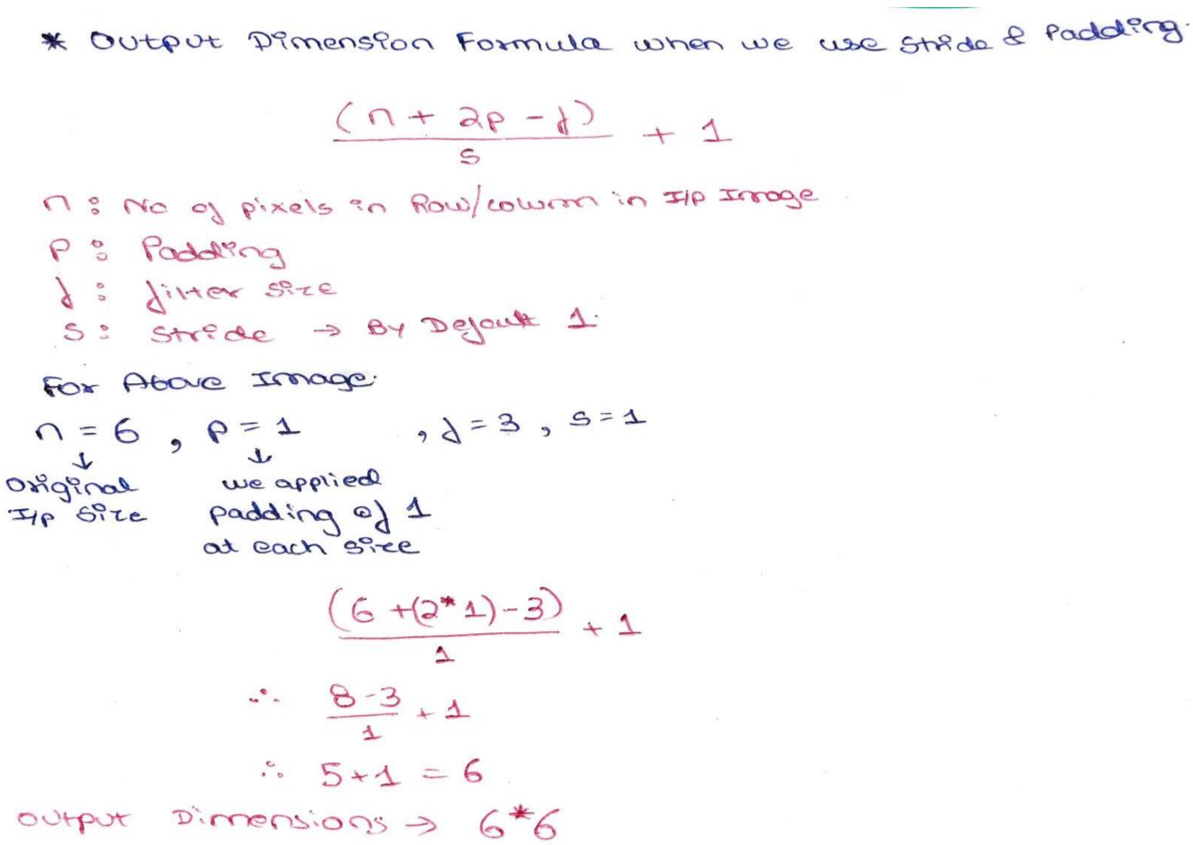
8*8

3*3

6*6

No Information LOSS

Calculation of Convolution Operation output when there is padding and stride.



* Output Dimension Formula when we use Stride & Padding.

$$\frac{(n + 2p - f)}{s} + 1$$

n : No of pixels in Row/column in I/p Image

p : Padding

f : filter size

s : stride → By Default 1.

For Above Image

n = 6 , p = 1 , f = 3 , s = 1

↓ Original I/p size

↓ we applied padding of 1 at each side

$$\frac{(6+(2*1)-3)}{1} + 1$$

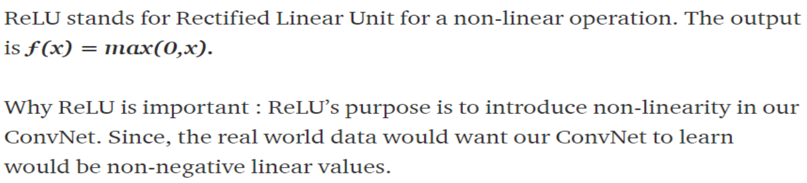
$$\therefore \frac{8-3}{1} + 1$$

$$\therefore 5+1 = 6$$

output Dimensions → 6*6

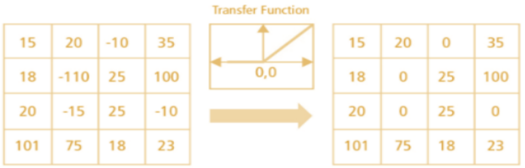
2.RELU Activation-Function :

RELU Activation function is applied on output of ConvolutionOperation performed on image.**Hmmm wait what is RELU Activation**?.If you are not clear about **RELU** and why activation function are use ?.Please view my [Neural Network Series.]()



ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0,x)$.

Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.



15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

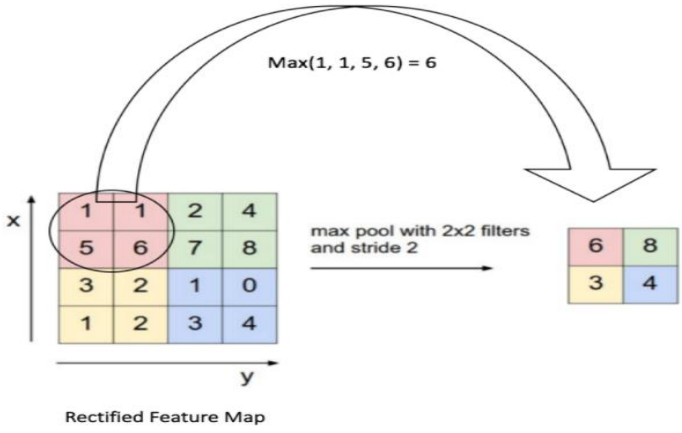
15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

3.Max-Pooling Layer :

It does following:

1. **Location Invariant**:Identifies all faces of cat in one image,If there are many cats in one image.
2. Reduces the size of convolution-operation-output while preserving the information

There are 3 types of Pooling–Min[Use_Min_value],Max[Use_Max_value],Average[Use_Avg_value]

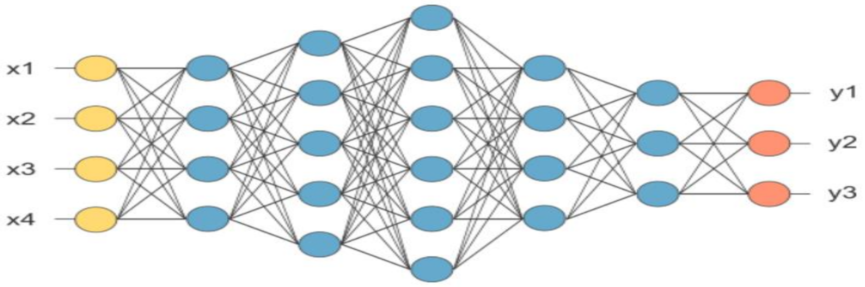


1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

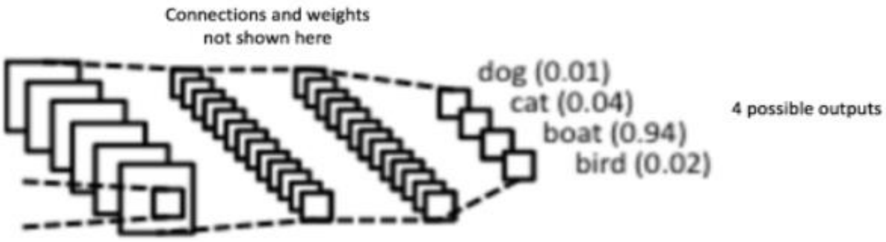
4.Flatten Layer and Fully Connected Layer

Output of Final convolution and pooling layer is transformed into Vector[1D_Array].And connected to one or more fully connected layers[Dense_Layers]



5.Last layer activation function.

If it is a multiclass classification task,then softmax function is used or else sigmoid function is used.Visit my Neural-Network-Series



4.Putting All together

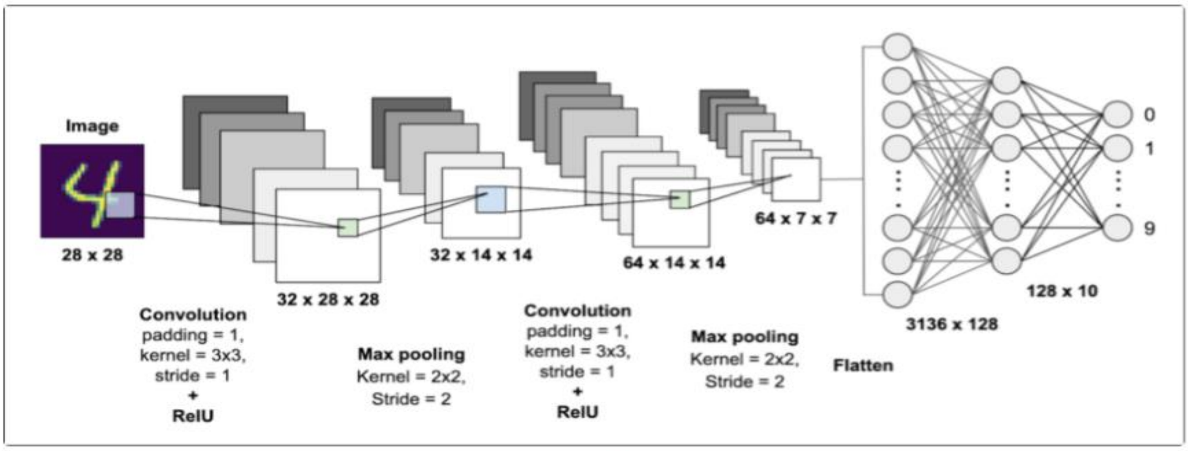
We train neural networks by repeating the learning process for the model then identify whether our model has found the pattern of the image input and able to differentiate the image input. After each learning process is done, the model will reevaluate it’s parameter on each layer to perform better on the next iteration. This action is called back-propagation which helps model to perform better each time it is trained.

In Each epoch,the model is able to distinguish between dominating and low-level features in images and classify them using the **Softmax-Classification** technique.

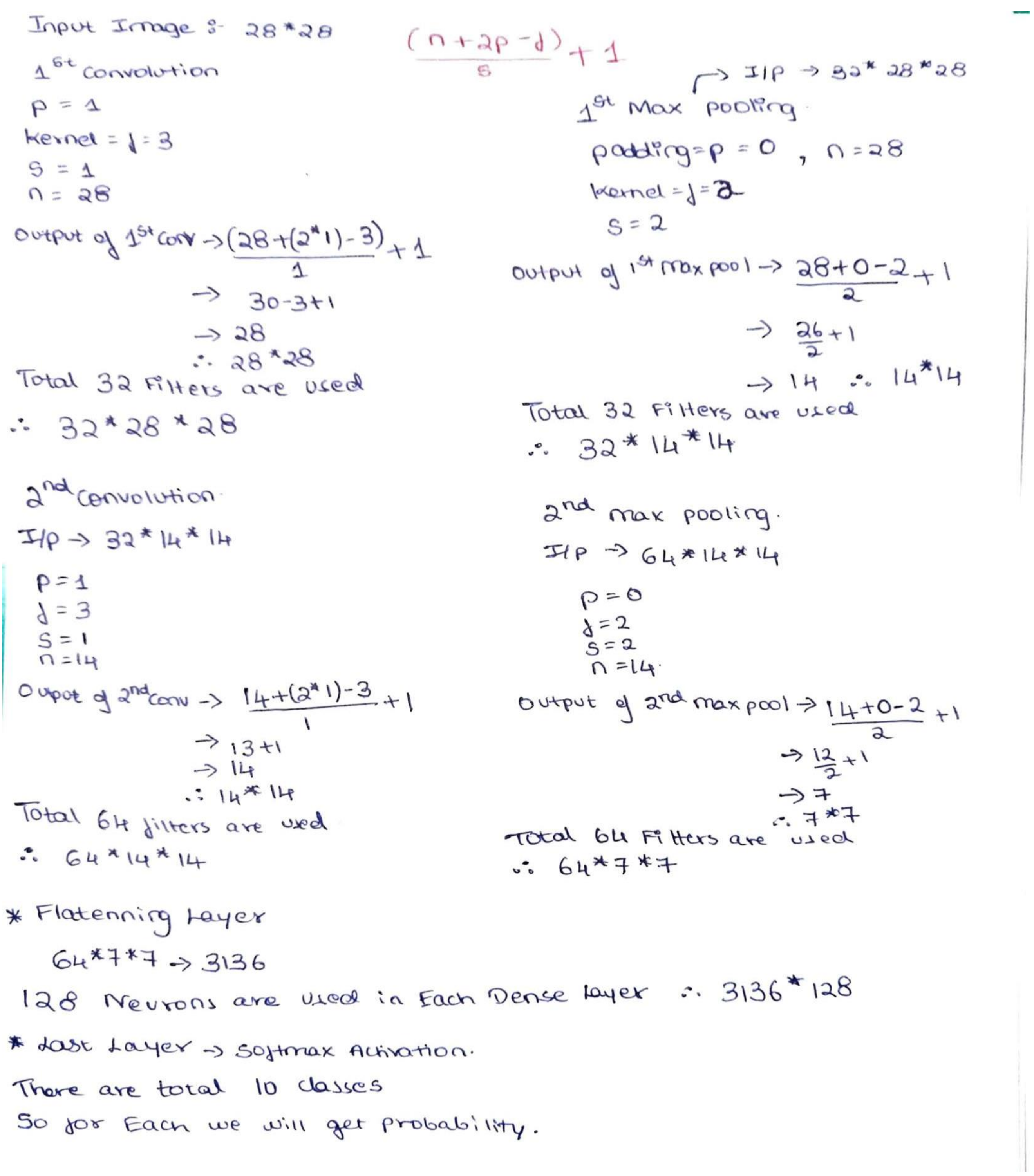
Training a network is a process of finding kernels in convolution layers and weights in fully-connected layers which minimize differences between output predictions and given ground truth labels on a training dataset.

5.Calculating Output Dimensions of CNN for ever layer.

Decoding Every Dimension of CNN Architecture shown below[Of Which you were scared in the beginning]



Representation of CNN Architecture



Input Image :- 28*28

1^{st} Convolution

p = 1

kernel = f = 3

S = 1

n = 28

Output of 1^{st} conv → $\frac{(28+(2*1)-3)}{1}+1$

→ 30-3+1

→ 28

∴ 28*28

Total 32 Filters are used

∴ 32*28*28

$\frac{(n+2p-f)}{s}+1$

→ I/P → 32*28*28

1^{st} Max pooling

padding = p = 0 , n = 28

kernel = f = 2

S = 2

Output of 1^{st} max pool → $\frac{28+0-2}{2}+1$

→ $\frac{26}{2}+1$

→ 14 ∴ 14*14

Total 32 Filters are used

∴ 32*14*14

2^{nd} Convolution

I/P → 32*14*14

p = 1

f = 3

S = 1

n = 14

Ouput of 2^{nd} conv → $\frac{14+(2*1)-3}{1}+1$

→ 13+1

→ 14

∴ 14*14

Total 64 filters are used

∴ 64*14*14

2^{nd} max pooling.

I/P → 64*14*14

p = 0

f = 2

s = 2

n = 14.

Output of 2^{nd} max pool → $\frac{14+0-2}{2}+1$

→ $\frac{12}{2}+1$

→ 7

∴ 7*7

Total 64 Filters are used

∴ 64*7*7

* Flatenning Layer

64*7*7 → 3136

128 Neurons are used in Each Dense layer ∴ 3136*128

* Last Layer → Softmax Activation.

There are total 10 classes

So for Each we will get probability.

6.Practical Implementation of CNN

View my notebook. Click here

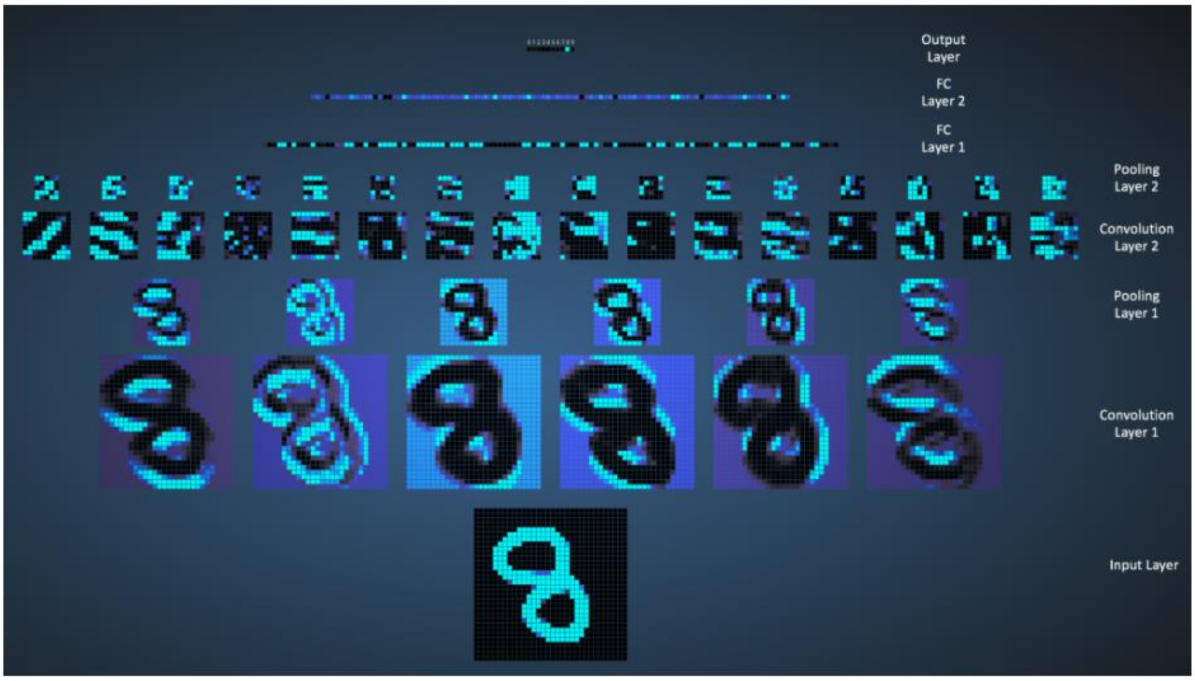
7.Transfer learning

Now if your fundamentals are clear for CNN.You can start learning Advanced CNN Architectures.Neural network models take time to train,so to reduce the time we can reuse the model weights from pretrained models [AlexNet,VGG,Inception,ResNet]

For TransferLearningModels Click here . I have uploaded all architecture intuitions

8.CNN-Visualization

https://www.cs.ryerson.ca/~aharley/vis/conv/



9.Assessments

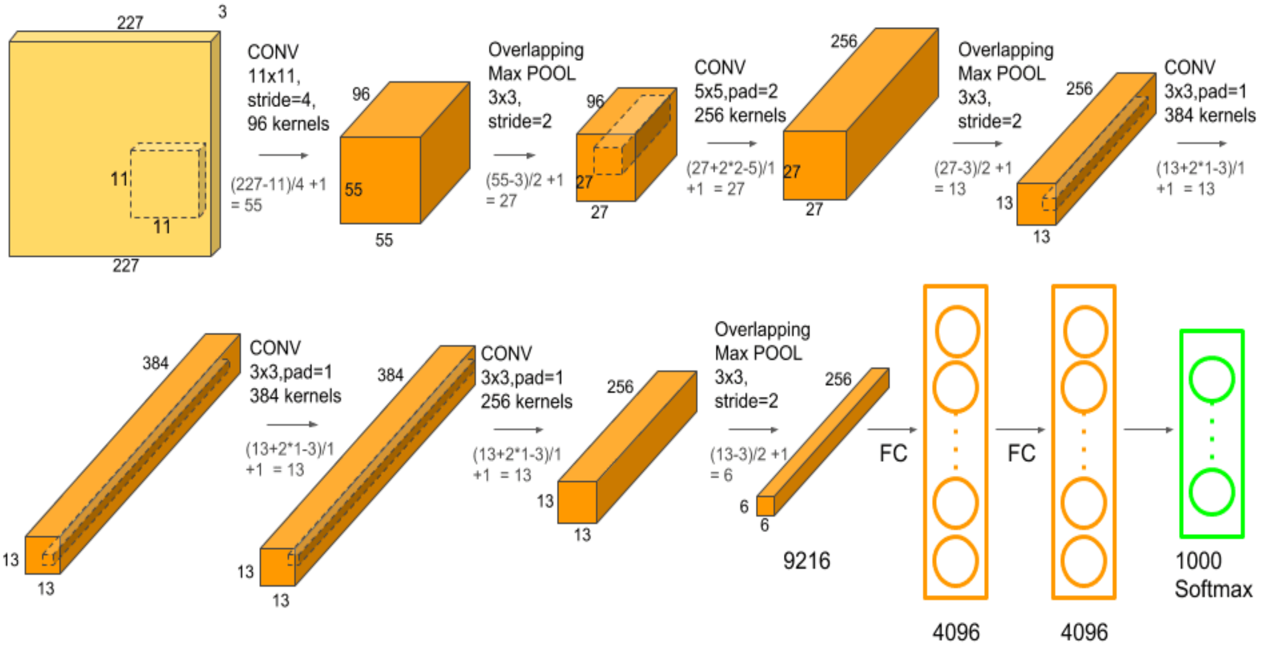
Question_1:Differentiate Gray-Scale Image&Color Image?

Question_2:Calculation of Output Dimensions after each Layer in CNN?

Question_3:Convolution Operation?

Question_4:Can you decode the complete Architecture of CNN now– Yes or No

Question_6:Can you Decode below AlexNet Architecture ?



10.Additional Questions

1.Which eLearning project/product/course you find most inspirational

Youtube is great learning tool.There are tons of videos in different topics,ranging from entertainment,education and many more.

Advantages:Free to use,Easy to use,Information Sharing,Recommendations based on your behaviour,Generation of Backlinks.

Disadvantages:Everything is public,Advertisements

2.Data Science-Curriculum

Part:1-Python-Important-Concepts

-List----Tuples----Sets----Dictionaries----Pandas----Numpy----MatplotLib----Seaborn----Lambda_Functions ,MapFunctions---FilterFunctions

Part:2-Statistics

-Population&Sample----RandomVariables(Discrete,Continous)----Distributions----Covariance----Pearson-Correlation-Coefficient----Spearman_Rank_Correlation----Zscore,IQR,Outliers

Part:3-Feature-Engineering

----Handling-Imbalanced-Dataset----Handling-Missing-Values----Handling-Outliers----CategoricalEncoding ----FeatureTransformation

Part:4-Feature-Selection

----BasicTechniques---Advanced Feature-Selection Techniques

Part:5-Exploratory-Data-Analysis

----Univariate,BiVariate,Multivariate---Histogram-Zscore---PDF&CDF---Hypothesis-Testing----Pvalue,Ttest,Anova-Test,Correlation

Part:6-Machine_Learning

---Bias&Variance----PerformanceMetrics-of-Regression&Classification----LinearRegression----Logistic Regression----DecisionTree----KNN----EnsembleTechiques----NaiveBayes----SVM----Cross-Validation-Techniques----Curse-of-Dimensionality----PCA----Clustering---Kmeans----Heirarchical----DBScan

Part:7-NLP

---TextPreprocessing-Level1(Tokenization,Stemming,Lemmatization)----Level2(BOW,TFIDF)----Level3 Word-Embeddings-Word2Vect----Glove----Machine-Learning-Usecases

Part:8-DeepLearining

---Neural-Network-Concepts----Activation-Functions----Weight-Initialization-Techniques---Epochs&Iterations---DeepLearning-Optimizers---ANN----CNN----RNN----LSTM----BI-DirectionalLSTM----Encoder-Decoder----Attention-Models----Transformers

Part:9-Project

CapstoneProject

To get more details about each and every topic see my Data Science repository on github.

11.Reference

https://github.com/LuckyRathod/DeepLearning/blob/main/DeepLearning-Complete_RoadMap/NeuralNetwork-Imp%20Concepts.txt

https://github.com/LuckyRathod/DataScience/blob/main/7.DeepLearning_Roadmap.txt

https://github.com/LuckyRathod/DeepLearning/tree/main/DeepLearning-Complete_RoadMap/2.CNN/3.Transfer%20Learning

https://github.com/LuckyRathod/DeepLearning/tree/main/DeepLearning-Complete_RoadMap

https://github.com/LuckyRathod/DeepLearning/tree/main/DeepLearning-Complete_RoadMap/2.CNN

https://www.kdnuggets.com/2020/06/introduction-convolutional-neural-networks.html

https://medium.com/analytics-vidhya/cnn-series-part-1-how-do-computers-see-images-32462a0b33ca

https://medium.com/analytics-vidhya/what-is-cnn-a-5-year-old-guide-to-convolutional-neural-network-c9d0338c3bf7

https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148#:~:text=Strides,with%20a%20stride%20of%202.