Unit – 4
DEEP LEARNING - CNN

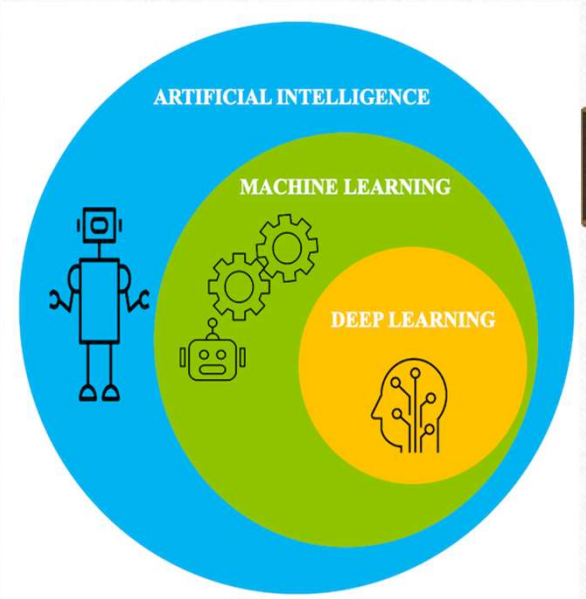
B. Keerthana
Assistant Professor
CSE department
GVPCE (A)

CONTENTS

- Concepts of deep learning,
- Introduction of Convolutional Networks,
- Components of CNN architecture,
- RELU layer.
- Properties, architectures and applications of CNN.

What is deep learning?

- Deep learning is a subdomain of machine learning.
- With accelerated computational power and large data sets, deep learning algorithms are able to self-learn hidden patterns within data to make predictions.
- Machine learning allows a system to learn and improve from experience automatically.
- Deep learning is an application of machine learning that uses complex algorithms and deep neural nets to train a model.



Deep learning and human brain

- Deep learning was inspired by the structure of a human brain.
- Deep learning architecture contains a computational unit that allows modeling of nonlinear functions called ***perceptron***.
- Generally, how a "neuron" in a human brain transmits electrical pulses throughout our nervous system, the perceptron receives a list of input signals and transforms them into output signals.

Deep learning and human brain (cont..)

- The perceptron aims to understand data representation by stacking together many layers, where each layer is responsible for understanding some part of the input.
- Each layer of perceptron is responsible for interpreting a specific pattern within the data.
- A network of these perceptron mimics how neurons in the brain form a network, so the architecture is called neural networks (or artificial neural networks).

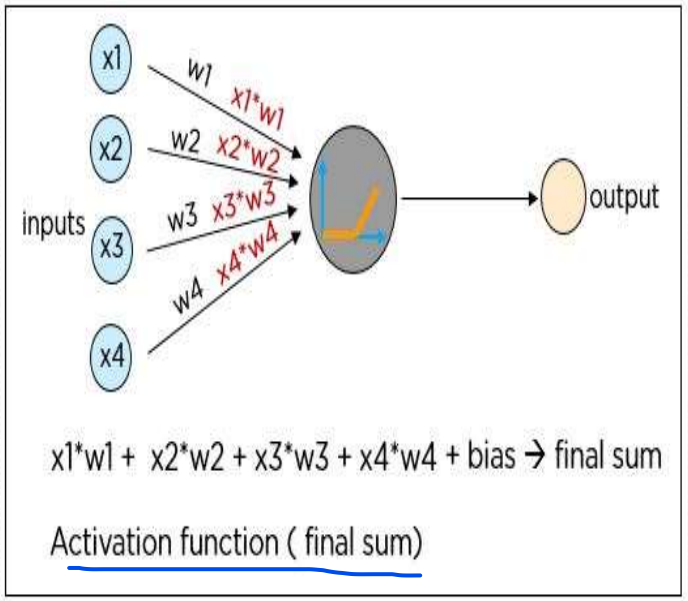
What are Neural Networks?

- A neural network is a system modeled on the human brain, consisting of an input layer, multiple hidden layers, and an output layer.
- Data is fed as input to the neurons. The information is transferred to the next layer using appropriate weights and biases.
- The output is the final value predicted by the artificial neuron.

What are Neural Networks?

Each neuron in a neural network performs the following operations:

- The product of each input and the weight of the channel it is passed over is found
- The sum of the weighted products is computed, which is called the weighted sum
- A bias value of the neuron is added to the weighted sum
- The final sum is then subjected to a particular function known as the ***activation function.***



Activation function?

- Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it.
- The purpose of the activation function is to **introduce non-linearity** into the output of a neuron.
- In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output. This process is known as *back-propagation*.
- Activation functions make the back-propagation possible since the ***gradients*** are supplied along with the error to update the weights and biases.

A gradient is just a derivative

Variants of Activation function?

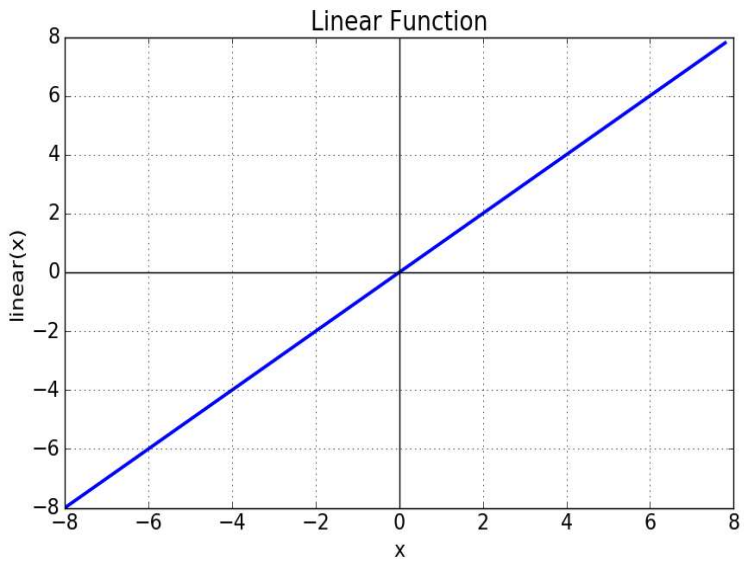
1) Linear Function :-

Equation : Linear function has the equation similar to as of a straight line i.e.

y = ax.

The range is [–infinity to +infinity]

It is a Linear activation function is used at just one place i.e. output layer.

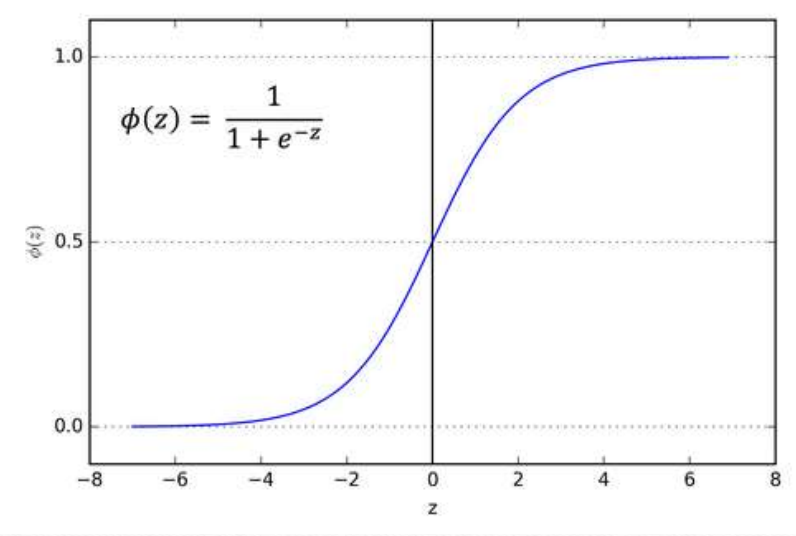


Variants of Activation function?

2) Sigmoid Function :- It is a function which is plotted as **'S'** shaped graph.

Equation : $A = 1/(1 + e^{-x})$, The range is [0 to 1]

It is a non linear function Notice that X values lies between -1 to 1, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.



Variants of Activation function?

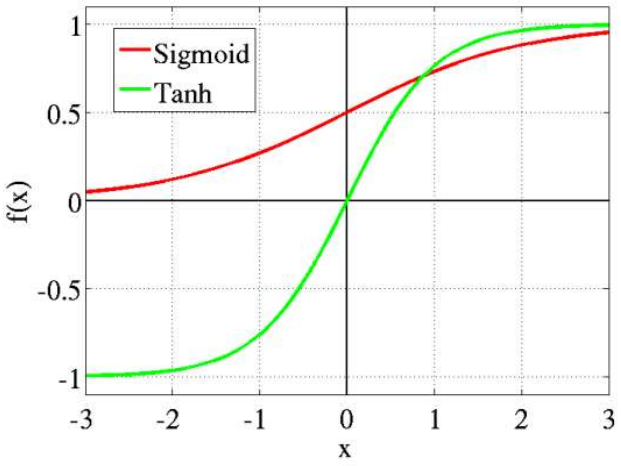
3) Tanh Function :- The function works almost better than sigmoid function. It's mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other. The range is [-1 to 1]

Equation :-

$$f(x) = \tanh(x) = 2/(1 + e^{-2x}) - 1$$

OR

$$\tanh(x) = 2 * \text{sigmoid}(2x) - 1$$



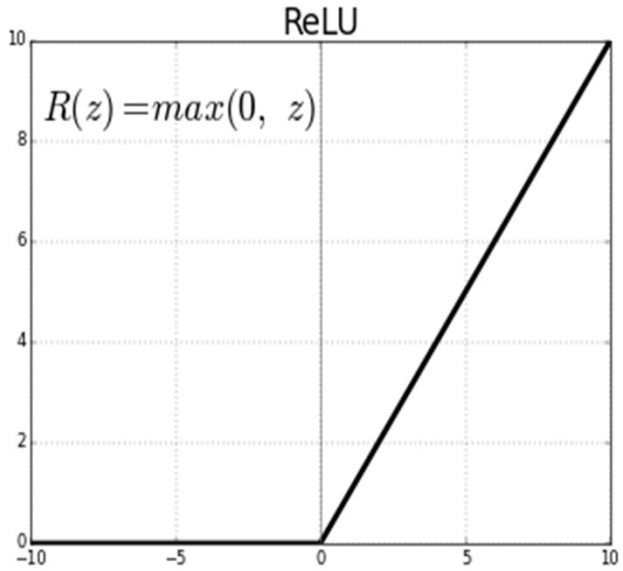
Variants of Activation function?

4) RELU :- Stands for *Rectified linear unit*. It is the most widely used activation function.

Equation :- ***A(x) = max(0,x)***. It gives an output x if x is positive and 0 otherwise.

Value Range :- [0, inf]

Nature :- non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.

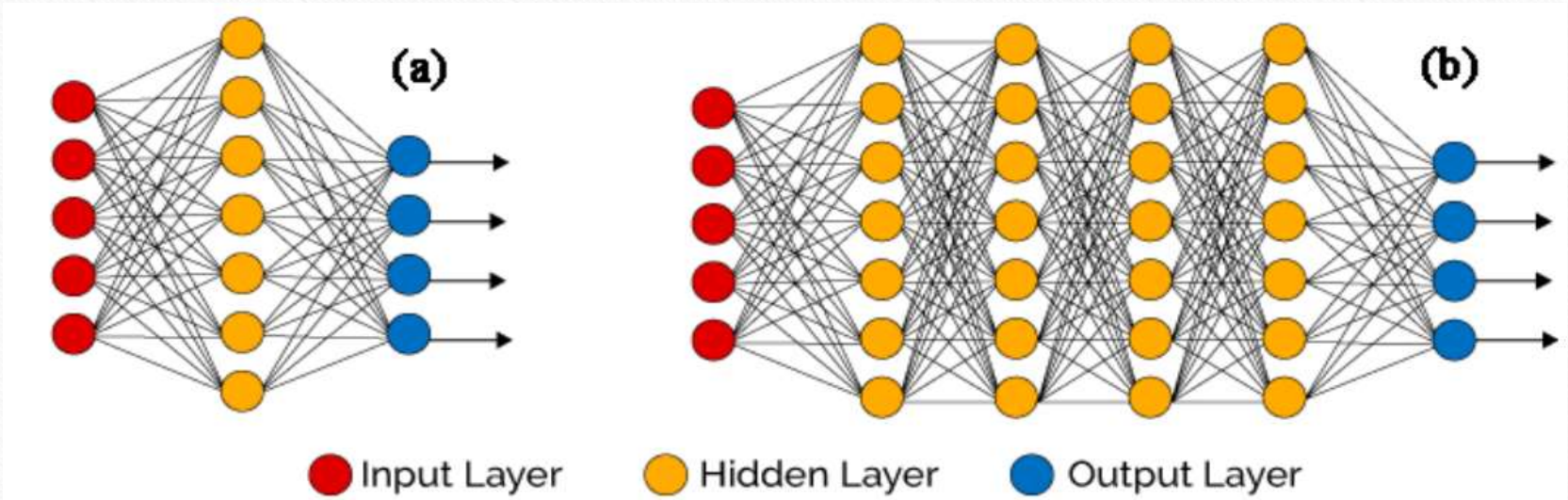


only 1 disadvantage -> if all the values of x are negative , then everything becomes 0

Gradient ?

- The network works to minimize an objective function, for example, the error incurred across all points in a data sample.
- At the output layer, the network must calculate the total error for all data points and take its derivative with respect to weights at that layer.
- **The derivative of error function with respect to weights is called the gradient of that layer.**
- The weights for that layer are then updated based on the gradient. This update can be the gradient itself or a factor of it. This factor is known as the ***learning rate.***

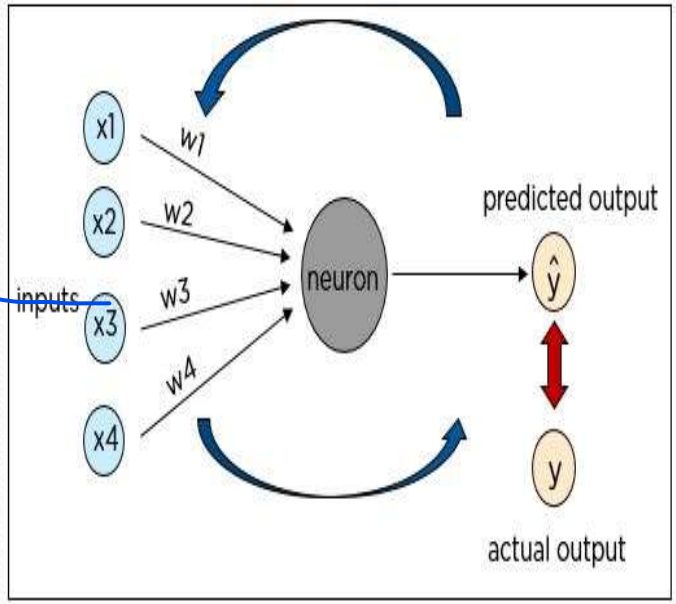
Structure of NN



(a) Simple neural network architecture; (b) Simple architecture of deep neural network (DNN)[49]

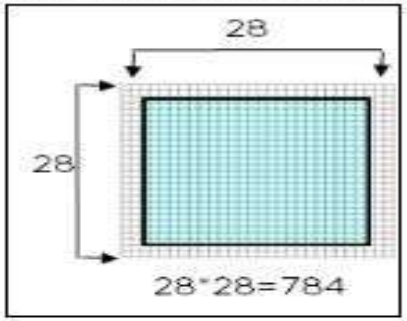
Cost Function

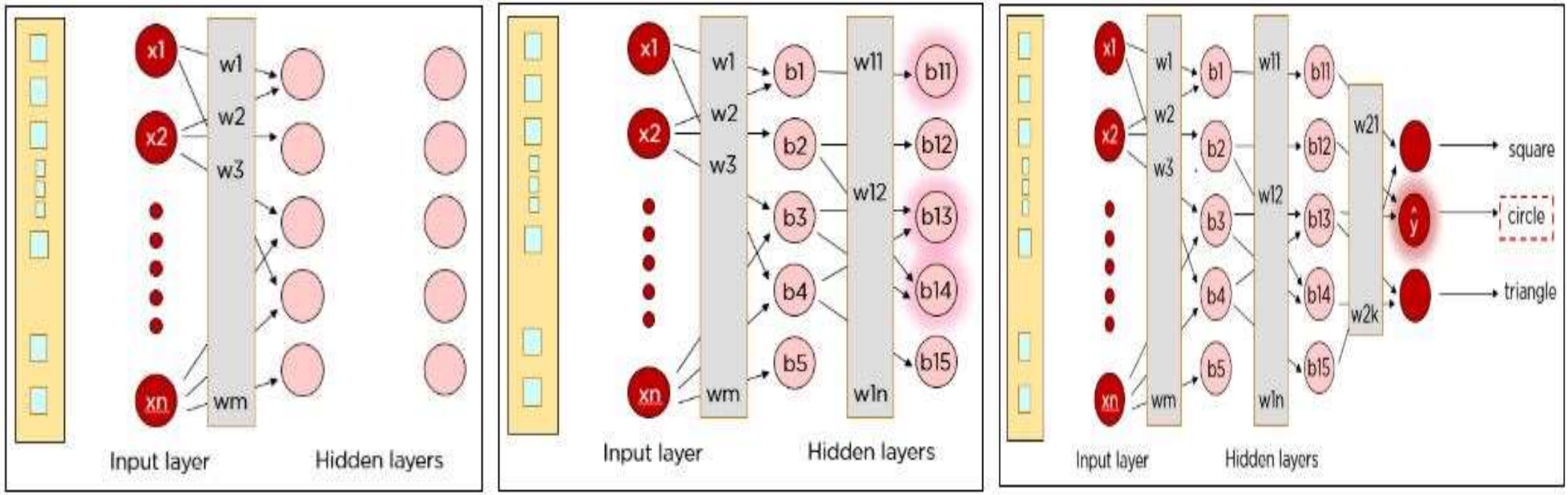
- The cost function is one of the significant components of a neural network.
- The ***cos***t value is the difference between the neural nets predicted output and the actual output from a set of labeled training data.
- The least-cost value is obtained by making adjustments to the weights and biases iteratively throughout the training process.

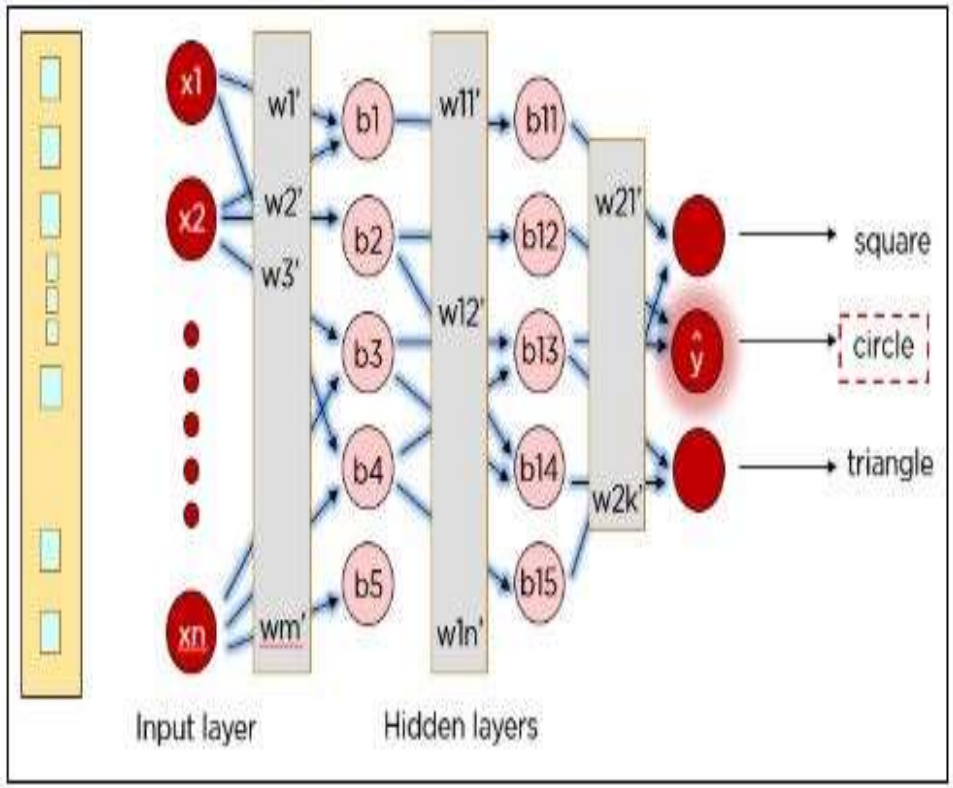


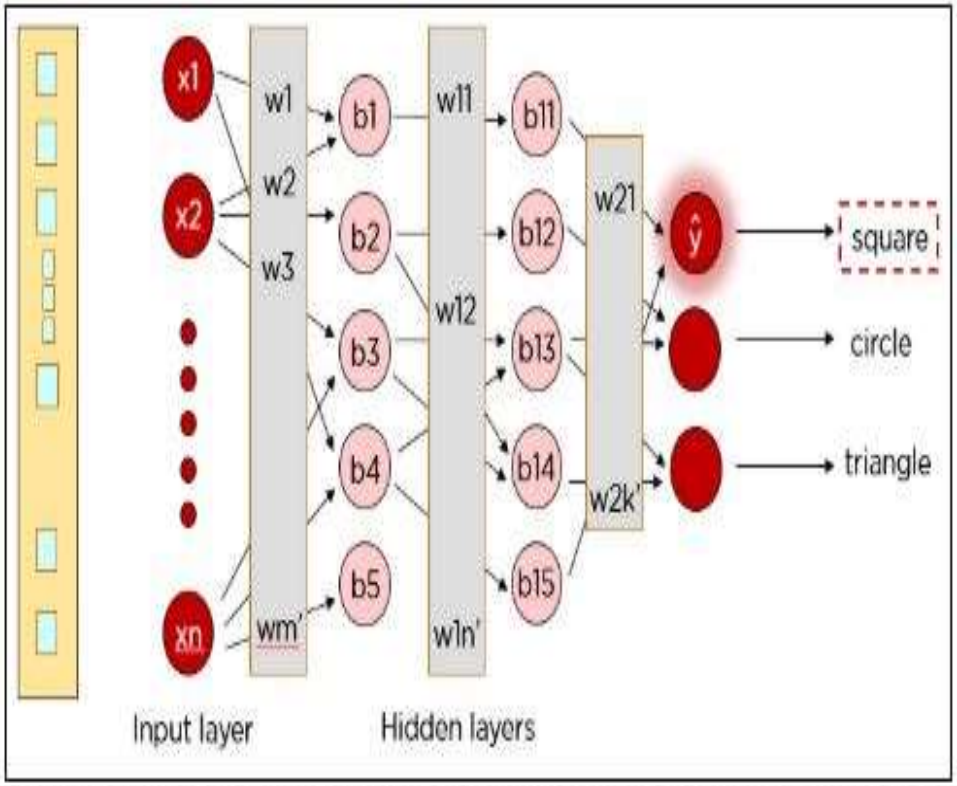
back propogation

Example


28
28
28*28=784


x1
x2
xn
w1
w2
w3
wm
Input layer
Hidden layers
b1
b2
b3
b4
b5
w11
w12
w1n
b11
b12
b13
b14
b15
w21
w2k
square
circle
triangle


x1
x2
xn
w1'
w2'
w3'
wm'
b1
b2
b3
b4
b5
w11'
w12'
w1n'
b11
b12
b13
b14
b15
w21'
w2k'
ŷ
square
circle
triangle
Input layer
Hidden layers


x1
x2
xn
w1
w2
w3
wm'
b1
b2
b3
b4
b5
w11
w12
w1n'
b11
b12
b13
b14
b15
w21
w2k'
ŷ
square
circle
triangle
Input layer
Hidden layers

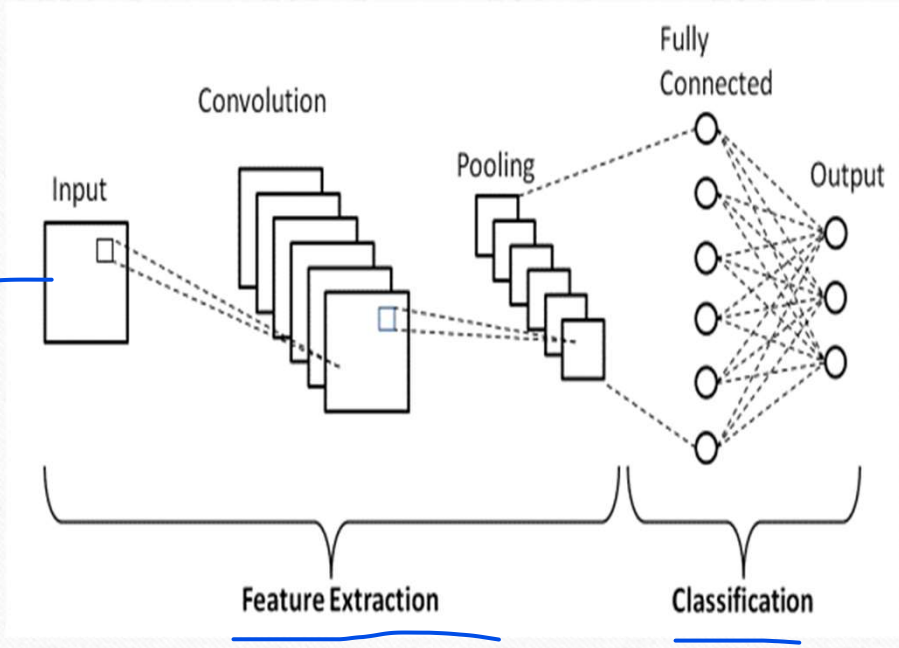
Introduction of Convolutional Networks

- CNN or ConvNet is a class of deep, feed-forward artificial neural networks, which was being mostly deployed in the field of analyzing/image recognition.
- The definition of convolution is a "*mathematical operation being applied on the two functions that give output in a form of a third function that shows how the shape of one function is being influenced, modified by the other function*".
- The applications are image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, NLP and financial time series.

CNN's Basic Architecture

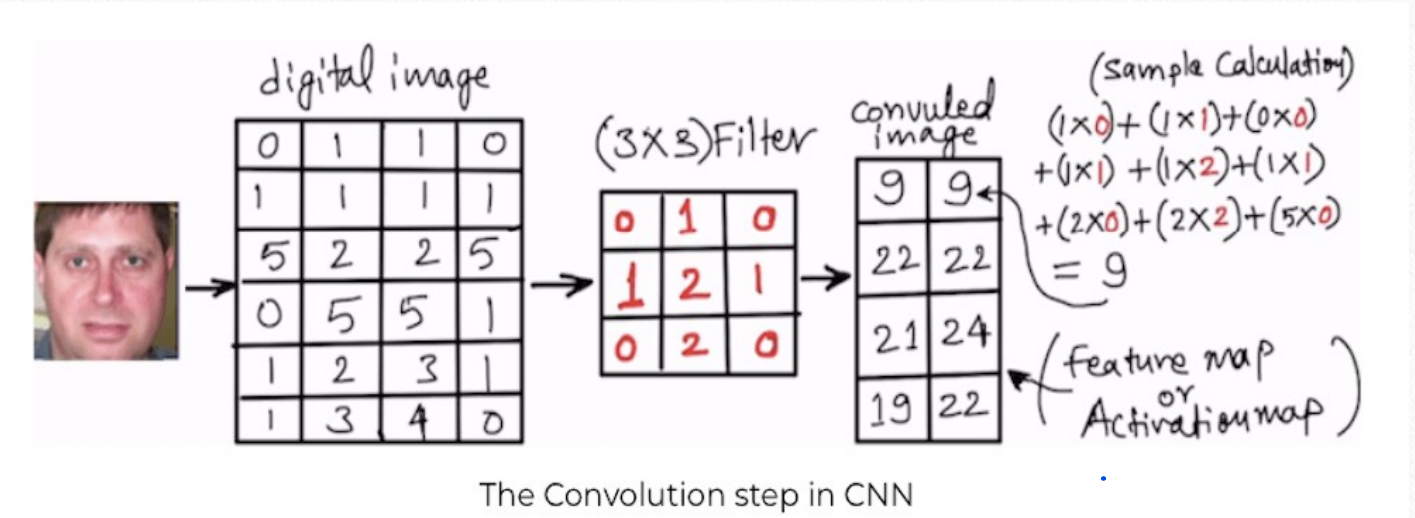
Key points:

- A convolution tool that separates and identifies the distinct features of an image for analysis in a process known as Feature Extraction.
- A fully connected layer that takes the output of the convolution process and predicts the image's class based on the features retrieved earlier.



Components of CNN architecture
1. Convolutional Layer

Convolutional Layer: used for extraction of the different features from the input images.



The Convolution step in CNN

1. Convolutional Layer (cont..)

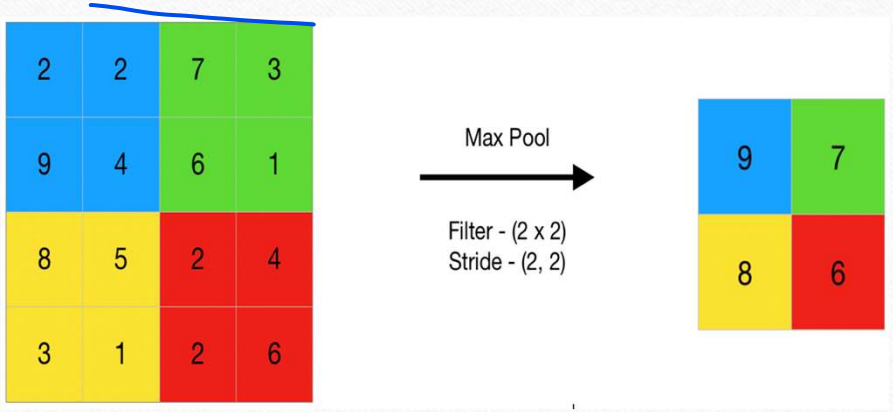
Feature map:The feature maps of a CNN capture the result of applying the filters to an input image. The size of the feature map is controlled by :

- ***Depth***: number of filters used for the convolution operation.
- ***Stride***: It is the number of pixels by which the filter matrix is slide over the input matrix.(stride =1 filter moves one pixel at a time)
- ***Zero-padding***: Refers to **the process of symmetrically adding zeroes to the input matrix**. It's a commonly used modification that allows the size of the input to be adjusted to our requirement.

Components of CNN architecture
2.Pooling Layer

Pooling Layer: This layer is responsible for the reduction of the size of the Convolved Feature. There are two types of pooling

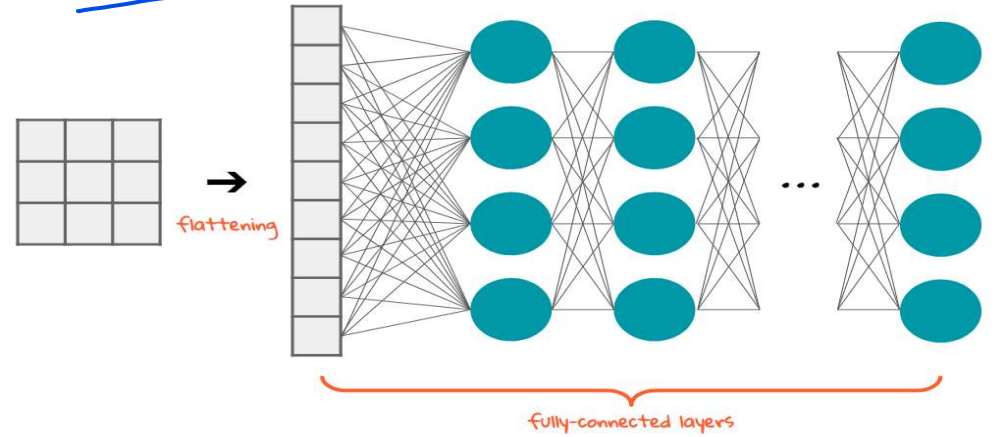
1. Max pooling

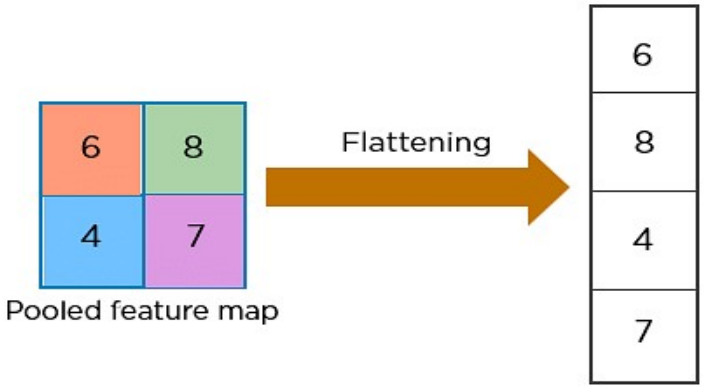


2. Average pooling

Components of CNN architecture
3.Flattening Layer

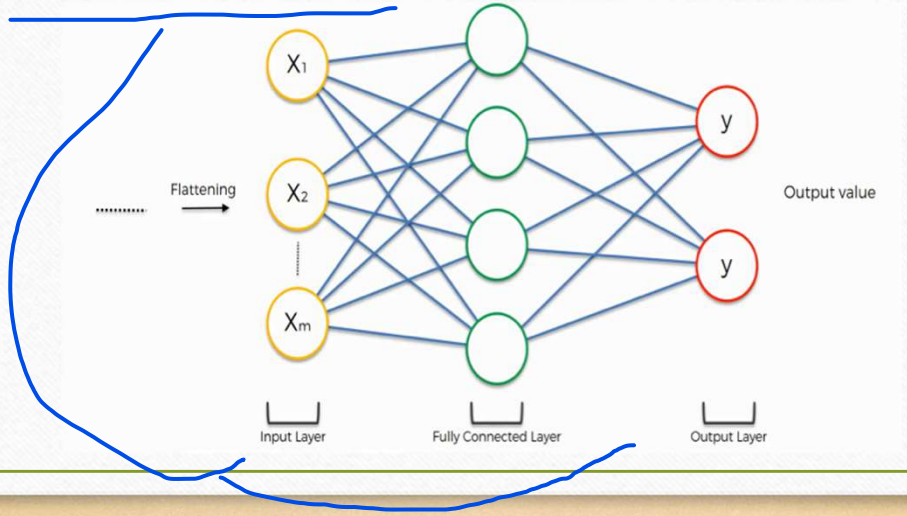
Flattening Layer: Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

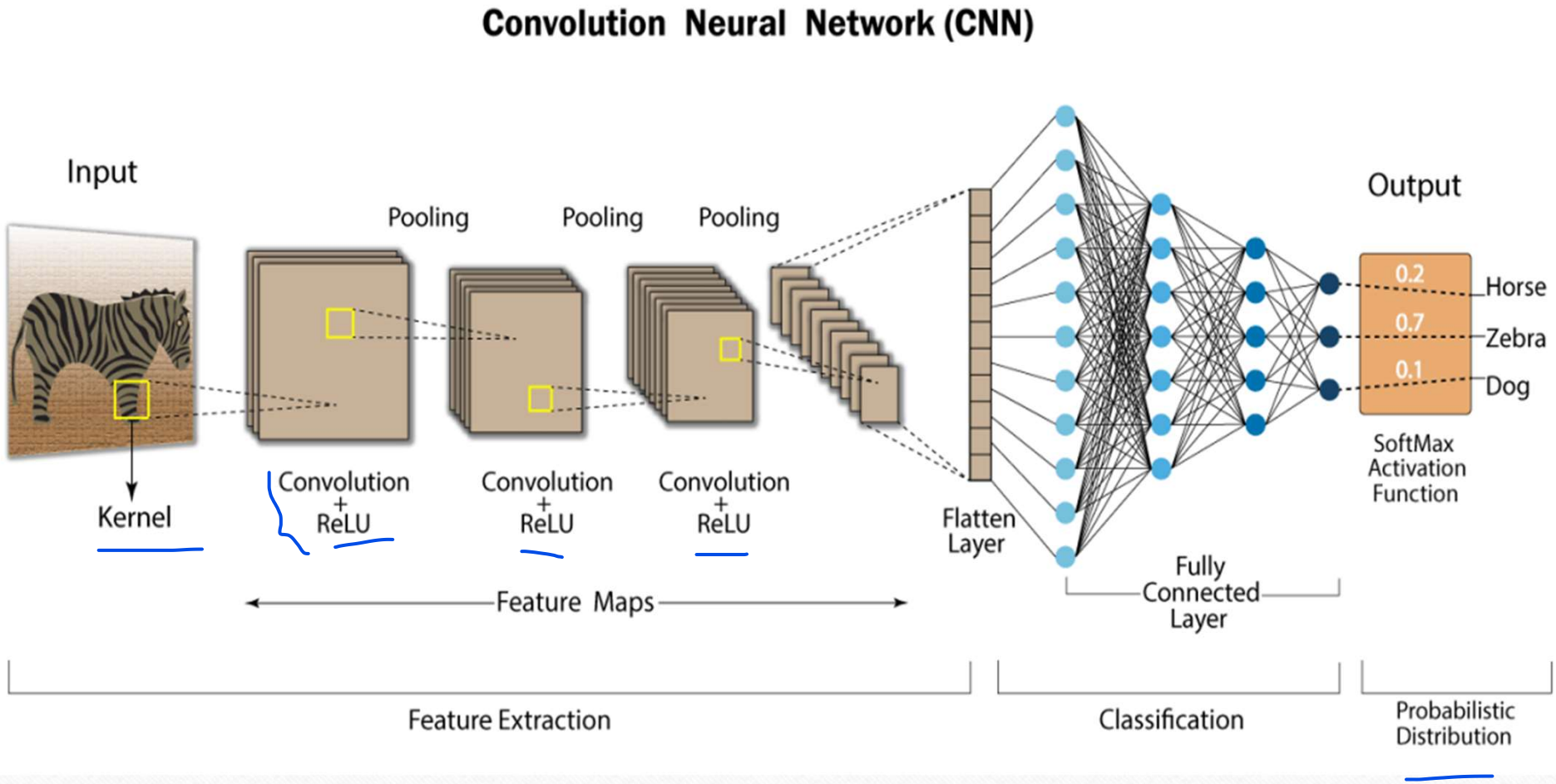




Components of CNN architecture
4.Fully connected Layer

- Fully Connected (FC) layer : The Fully Connected (FC) layer comprises the weights and biases together with the neurons and is used to connect the neurons between two separate layers.




Convolution Neural Network (CNN)
Input
Pooling
Pooling
Pooling
Output
0.2
Horse
0.7
Zebra
0.1
Dog
Kernel
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Flatten Layer
SoftMax Activation Function
Feature Maps
Fully Connected Layer
Feature Extraction
Classification
Probabilistic Distribution

ReLU Layer

- A **Rectified Linear Unit (ReLU)** is a non-linear activation function that performs on multi layer neural network (ex: f(x)=max(0,x) where x is input).
- In this layer we remove every negative value from the filtered image and replace it with zero.
- This function only activates when the node input is above a certain quantity. So, when the input is below zero the output is zero
- The purpose of applying the rectifier function is to increase the non-linearity in our images. This function causes dying ReLU problem and the solution is Leaky ReLU

if complete input <0 ----> then output is 0

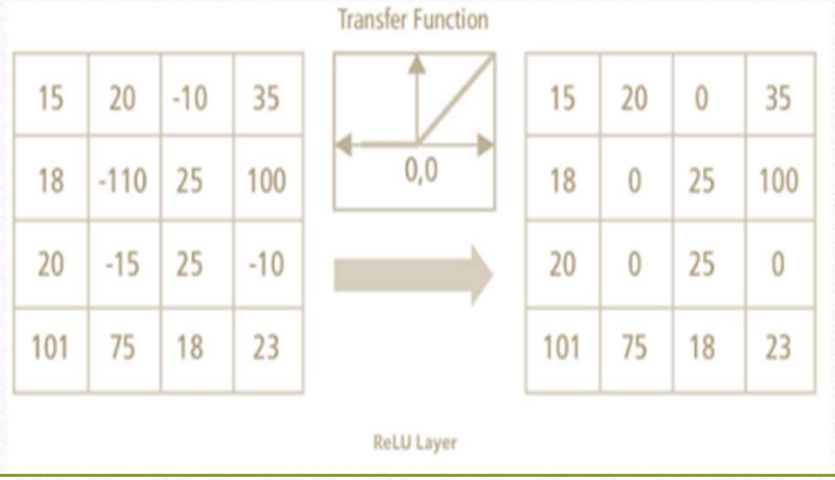
ReLU Layer (cont..)

- Performs element wise operations
- It has an output that is a rectified feature map

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases}$$

where x = an input value

x	f(x)=x	F(X)
-110	f(-110)=0	0
-15	f(-15)=0	0
15	f(15)=15	15
25	f(25)=25	25



Leaky ReLU

$$R(z) = \begin{cases} z & z > 0 \\ \alpha z & z <= 0 \end{cases}$$

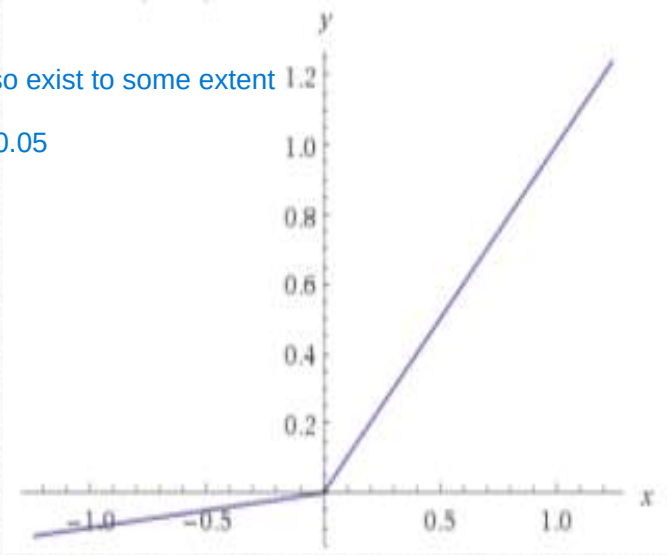
Leaky ReLU: It is most common and effective method to reduce a dying ReLU. (α=0.01)

negative values can also exist to some extent

eg: -5 -> alpha*-5=-0.05

What is a dying ReLU?

The **dying ReLU** refers to the problem when ReLU neurons become inactive and only output 0 for any input. So, once a neuron gets negative input, it will always output zero and is unlikely for it to recover. It will become inactive forever.



Properties of CNN

Weight sharing:

- A property of CNN is to share the same weight vectors across the receptive fields in particular layer and it also reduces overfitting.
- It is known that a filter is applied to all the receptive fields in a layer to produce a feature map.

Translation Invariance:

- This property allows CNN to recognize the identified object even when it is moved to any different place.

Architectures of CNN

1. LeNet
2. AlexNet
3. ZFNet
4. GoogLeNet
5. VGGNet
6. ResNet
7. DenseNet

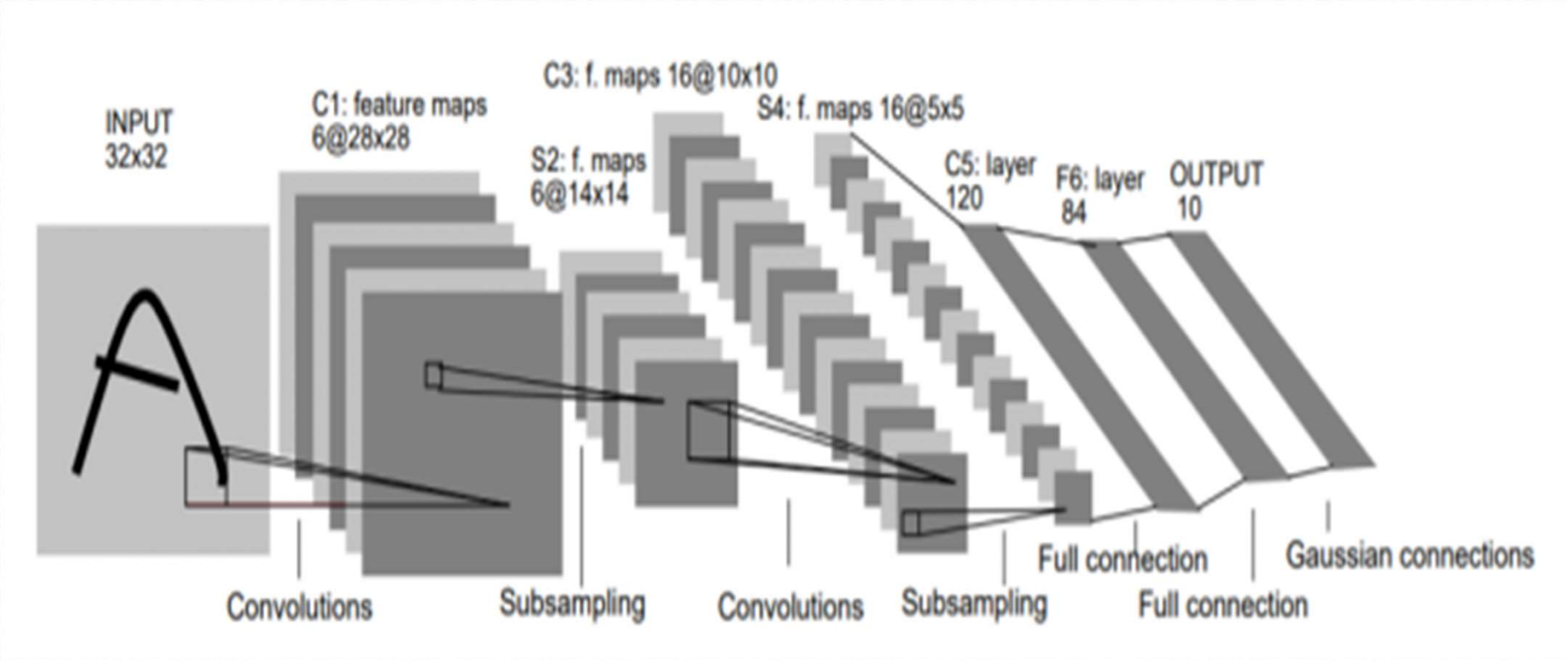
1. LeNet

- LeNet is the first CNN architecture. It was developed in 1998 LeCun and Christopher for handwritten digit recognition problems.
- The model has five convolution layers followed by two fully connected layers.
- LeNet could not train well due to the vanishing gradients problem.

derivatives are almost near to 0

- To solve this issue, a shortcut connection layer known as max-pooling is used between convolutional layers to reduce the spatial size of images which helps prevent overfitting and allows CNNs to train more effectively.

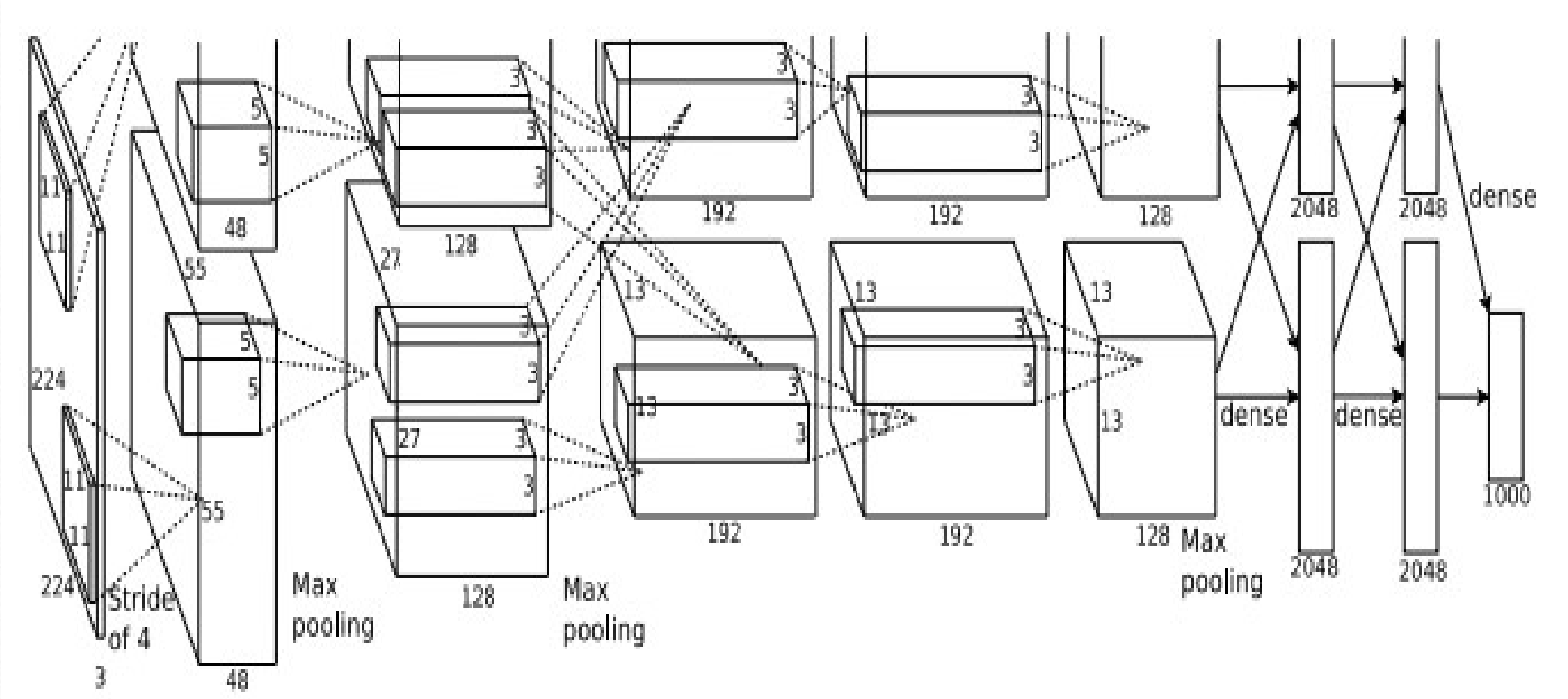
1. LeNet


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

2. AlexNet

- It was developed by Alex and Hinton.
- AlexNet network had a very similar architecture to LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other.
- AlexNet was the first large-scale CNN and was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012.
- The activation function used in all layers is Relu. The activation function used in the output layer is SoftMax.

2. AlexNet


224
224
3
11
11
Stride
of 4
48
55
5
5
48
Max
pooling
27
27
128
128
Max
pooling
13
13
13
13
192
192
192
192
128
128
Max
pooling
2048
2048
2048
2048
dense
dense
dense
1000

3. ZFNet

- ZF Net was developed by Matthew Zeiler and Rob Fergus. It was the ILSVRC 2013 winner.
- The network has relatively fewer parameters than AlexNet, but still outperforms it on ILSVRC 2012 classification task by achieving top accuracy with only 1000 images per class
- ZF Net CNN architecture consists of a total of seven layers: Convolutional layer, max-pooling layer (downscaling), concatenation layer, convolutional layer with linear activation function, and stride one, dropout for regularization purposes applied before the fully connected output

3. ZFNet

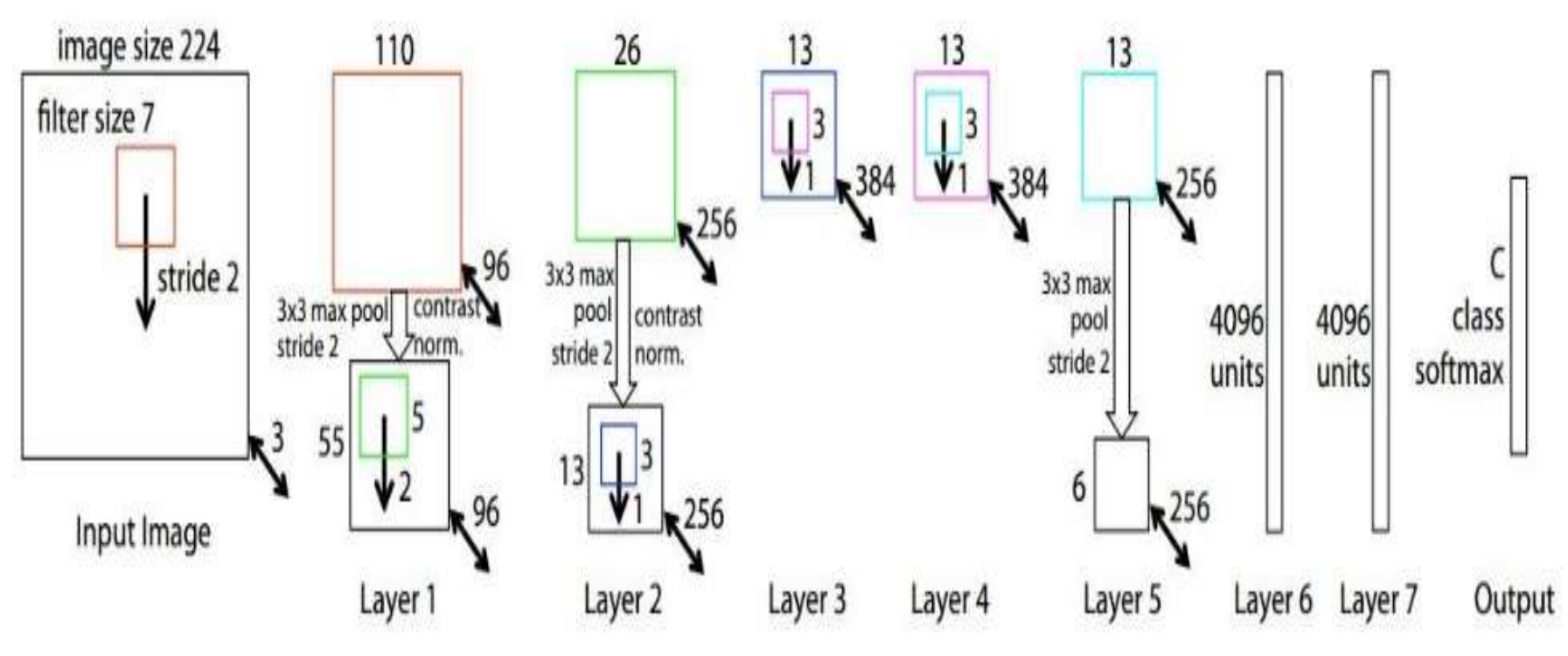
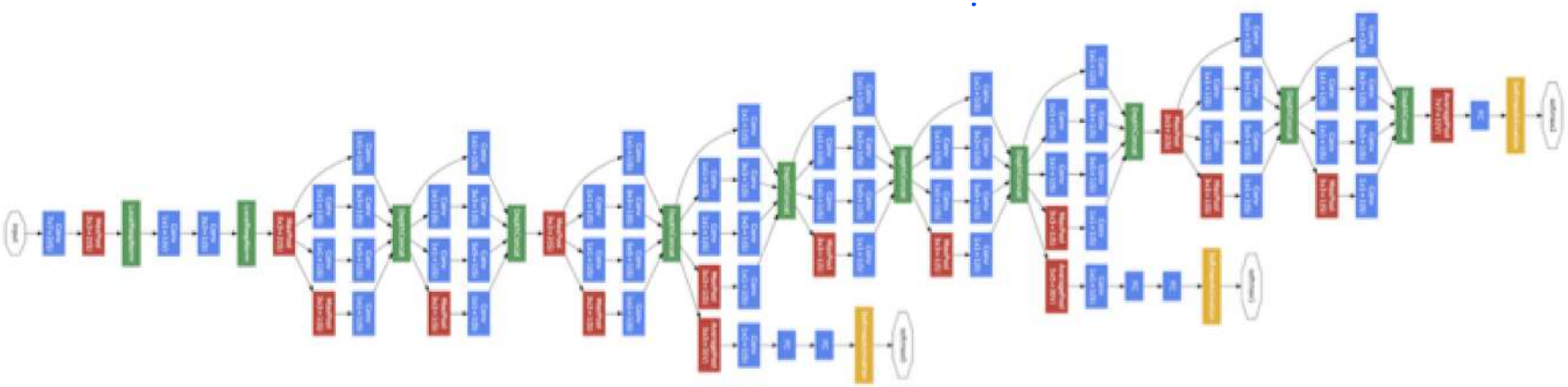


image size 224
filter size 7
stride 2
3
Input Image
110
96
3x3 max pool
stride 2
contrast
norm.
55
5
2
96
Layer 1
26
256
3x3 max
pool
stride 2
contrast
norm.
13
3
1
256
Layer 2
13
3
1
384
Layer 3
13
3
1
384
Layer 4
13
256
3x3 max
pool
stride 2
6
256
Layer 5
4096
units
Layer 6
4096
units
Layer 7
C
class
softmax
Output

4. GoogLeNet

- The winner of the ILSVRC 2014 competition was GoogLeNet.
- It achieved an error rate of 6.67% with its 22 layers, which was very close to the human level performance.
- The major idea was to cover a bigger area but to keep a fine resolution on the images.


Convolution
Pooling
Softmax
Other

5. VGGNet

- VGGNet is the CNN architecture that was developed by Karen Simonyan, Andrew Zisserman et al. at Oxford University.
- VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture.
- It is currently the most preferred choice in the community for extracting features from images.

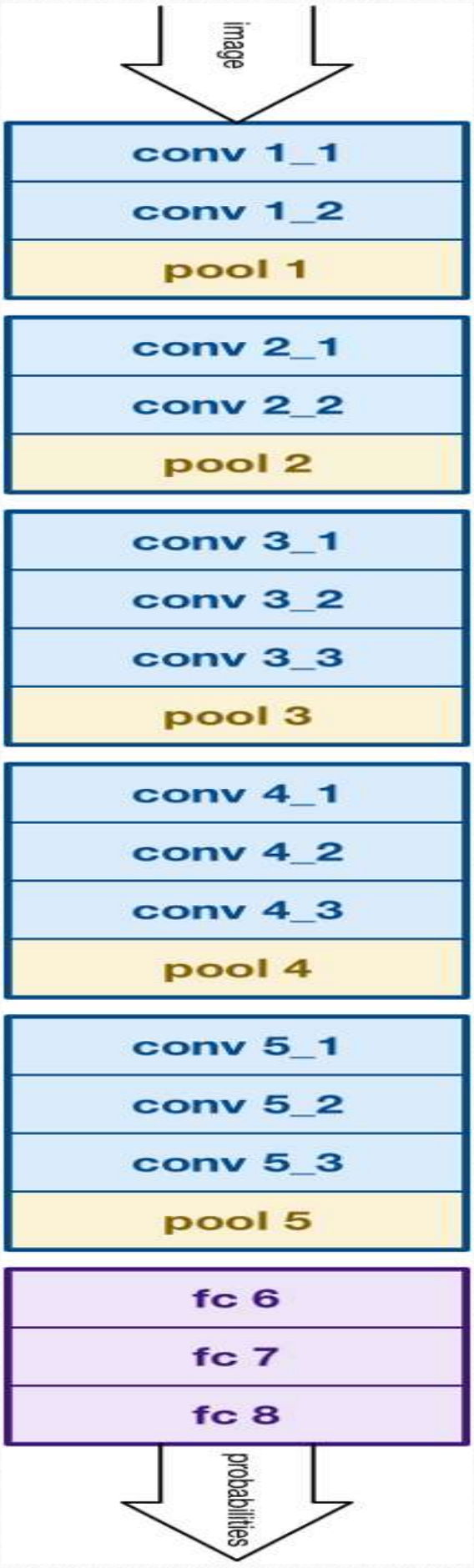
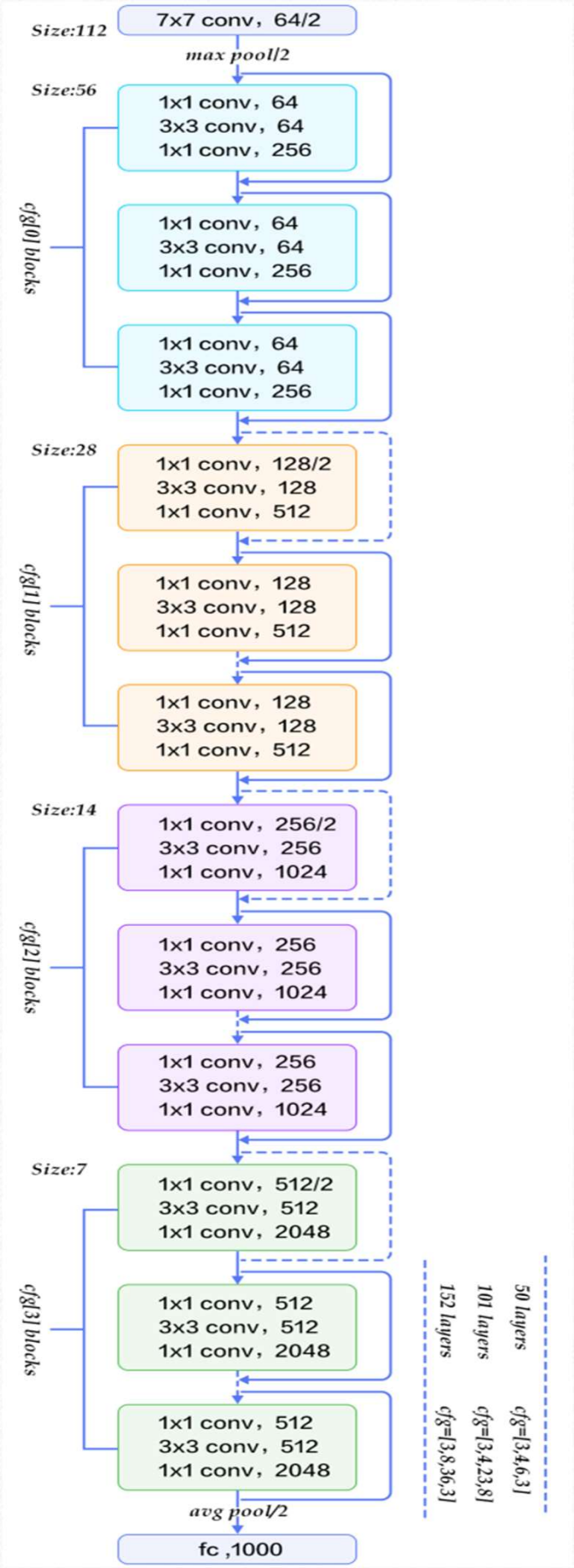


image
conv 1_1
conv 1_2
pool 1
conv 2_1
conv 2_2
pool 2
conv 3_1
conv 3_2
conv 3_3
pool 3
conv 4_1
conv 4_2
conv 4_3
pool 4
conv 5_1
conv 5_2
conv 5_3
pool 5
fc 6
fc 7
fc 8
probabilities

6. ResNet

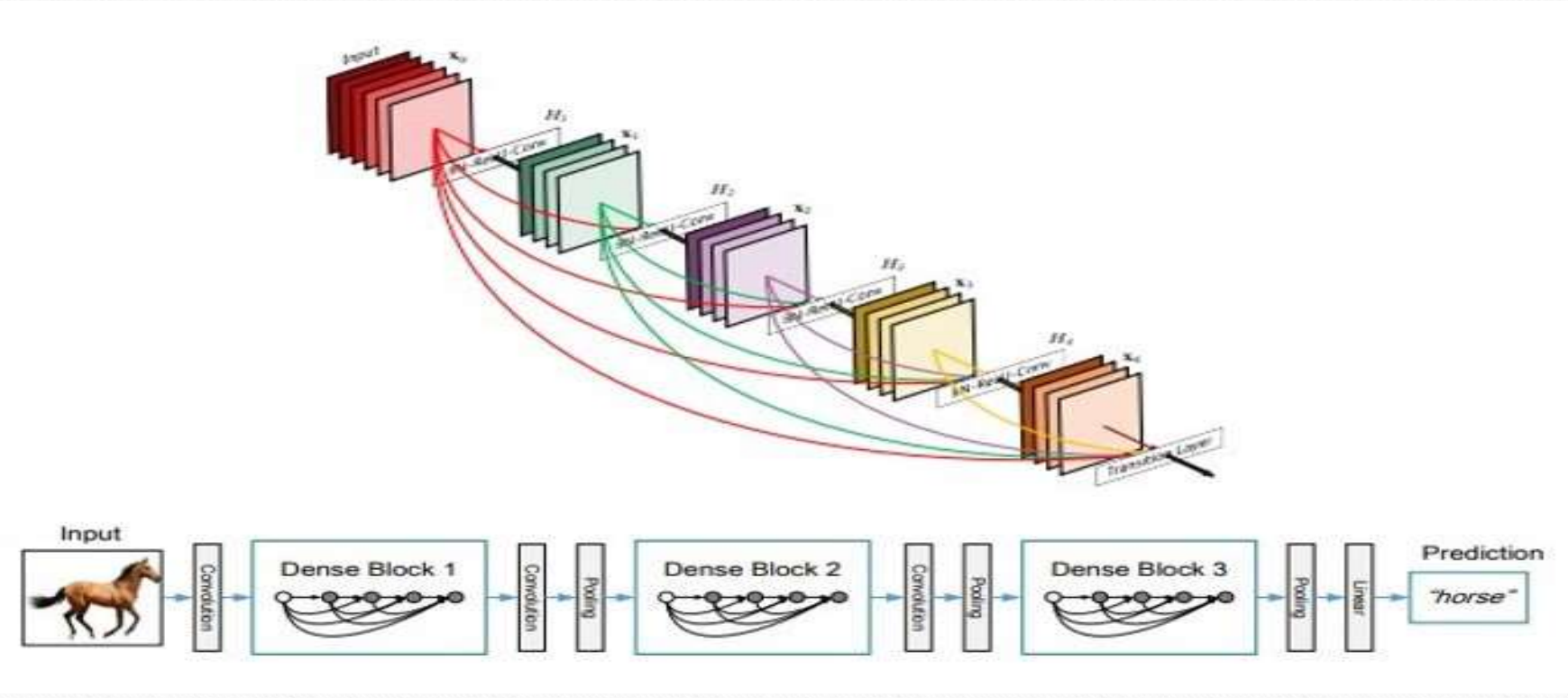
- At last, at the ILSVRC 2015, the so-called Residual Neural Network (ResNet) by Kaiming introduced anovel architecture with “skip connections”.
- Such skip connections are also known as **gated units or gated recurrent units** and have a strong similarity to recent successful elements applied in RNNs/
- In this technique they were able to train a NN with 152 layers while still having lower complexity than VGGNet.
- It achieves a top-5 error rate of 3.57% which beats human-level performance on this dataset.


Size:112
7x7 conv, 64/2
max pool/2
Size:56
1x1 conv, 64
3x3 conv, 64
1x1 conv, 256
cfg[0] blocks
Size:28
1x1 conv, 128/2
3x3 conv, 128
1x1 conv, 512
cfg[1] blocks
Size:14
1x1 conv, 256/2
3x3 conv, 256
1x1 conv, 1024
cfg[2] blocks
Size:7
1x1 conv, 512/2
3x3 conv, 512
1x1 conv, 2048
cfg[3] blocks
avg pool/2
fc ,1000
50 layers cfg=[3,4,6,3]
101 layers cfg=[3,4,23,8]
152 layers cfg=[3,8,36,3]

7. DenseNet

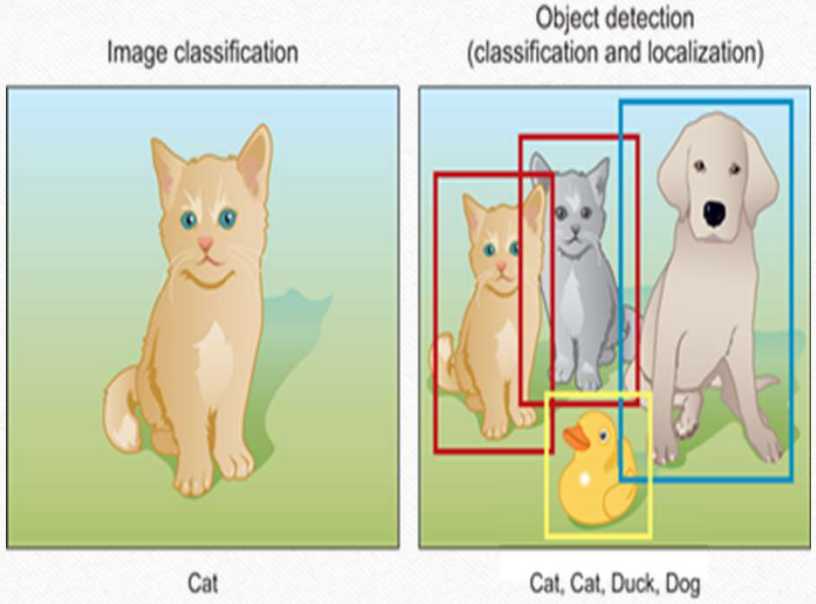
- Counter-intuitively, by connecting this way DenseNets ***require fewer parameters*** than an equivalent traditional CNN, as there is ***no need to learn redundant feature maps.***
- In fact, the number of parameters of ResNets are big because every layer has its weights to learn. Instead, ***DenseNets layers are very narrow*** (e.g. 12 filters), and they just ***add a small set of new feature-maps***.
- DenseNets make the first difference with ResNets is **DenseNets do not sum the output feature maps of the layer with the incoming feature maps but concatenate them**.

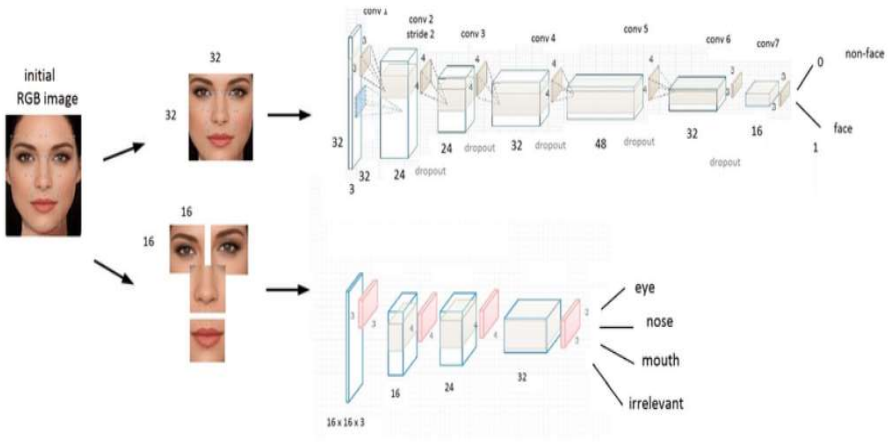
- DenseNets are divided into **DenseBlocks, where the dimensions of the feature maps remains constant within a block, but the number of filters changes between them**. These layers between them are called ***Transition Layers***

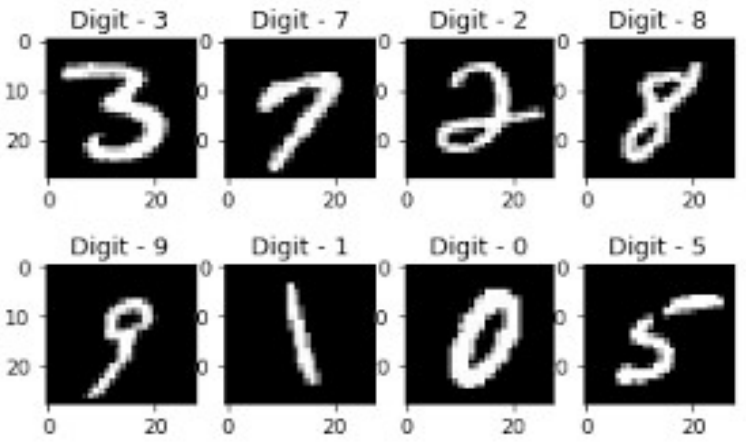


Applications of CNN

- Object detection
- Face recognition
- Scene labelling
- Optical character Recognition (OCR)
- Handwritten Digit Recognition








Office


Corridor


Rest space


Kitchen


Office


Discussion area

Thank You