Different types of layers in Convolutional Neural Network

Seminar Under the guidance of

Prof. Amit sethi

Outline

Introduction to Neural Network

Convolutional Neural Network

Introduction to Residual Network

Introduction to Neural Network

Why Neural Network?

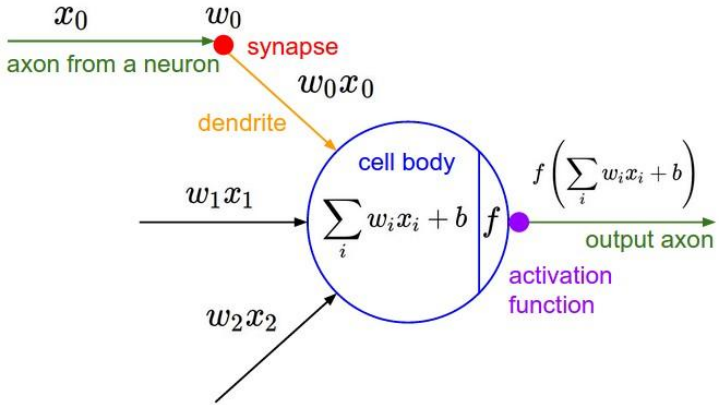
How Neural Network work?

How Neural Network learn?

Motivation-

- The input signals comes from the dendrites and received by each input signal and produces the output signals along its axon.
- The output eventually branches out from the axon and further connects to dendrites of the other neurons via synapses.

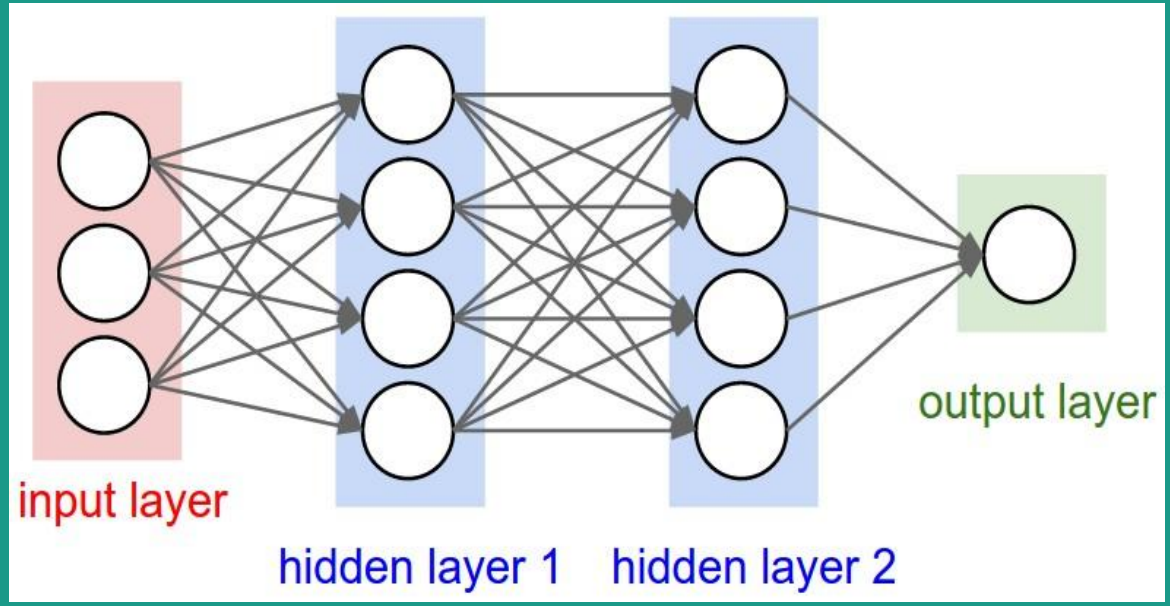
- The signals that moves along the axons interacts with the other neuron



source:http://cs231n.github.io/convolutional-networks

- There is a certain threshold based on that the neuron fires and sends a spike along the axon. This firing rate is modeled by an activation function.So basically,dot product is performed by each neuron and a bias is added to introduce the nonlinearity.

Why so many layers?



source:http://cs231n.github.io/convolutional-networks

The images contains hierarchical structure like eyes in the face and edges in the eyes,so layers are important to make sense for different data domains.
For any good recognition system depth is one important parameter.

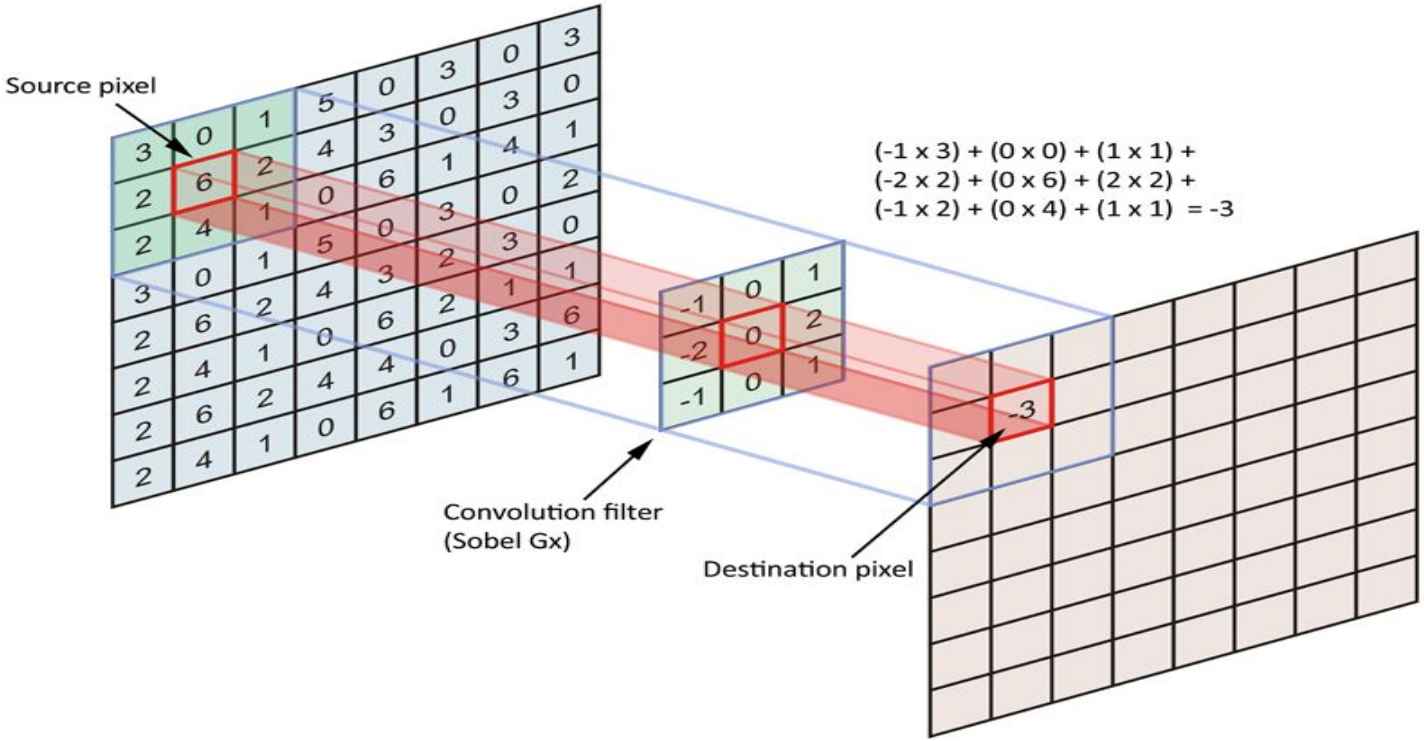
Why Convolutional Neural Network?

- The fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth.It recognizes an object by looking at its low level features first.

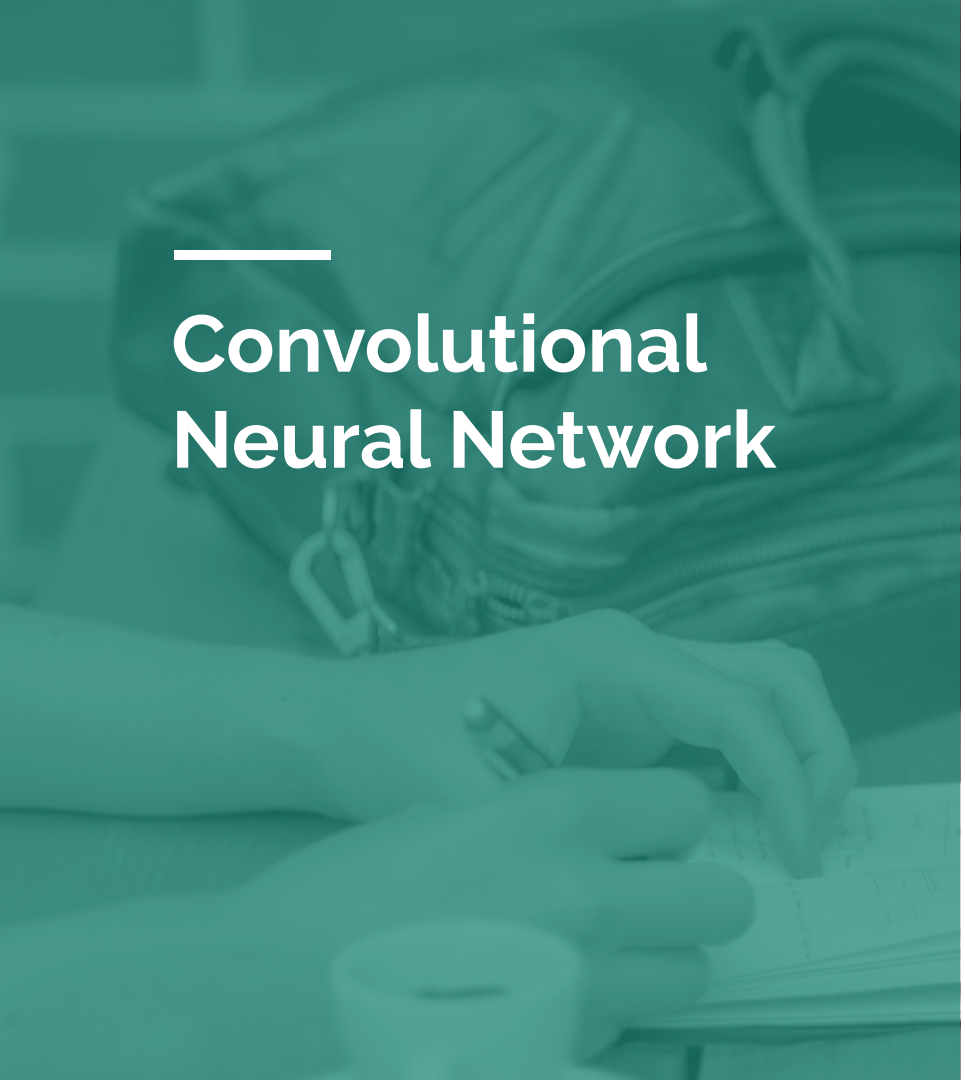
- During the convolution process,the network learns about each individual parts of an object with the help of the shared filters applied on small regions and finally gives advantage of reducing the actual parameter size.

- The learned filters can be used again for more abstract features detection so for complex images and their recognition ConvNet is

Convolution process



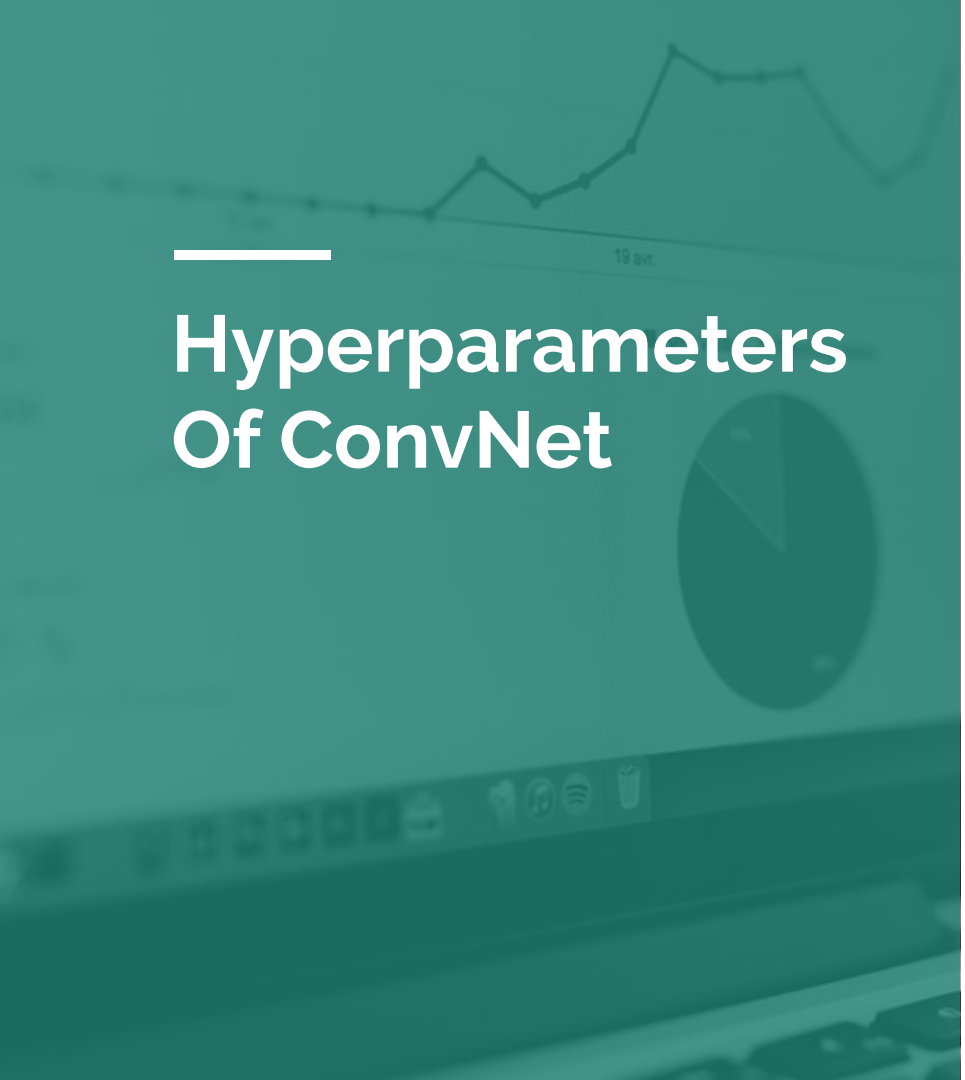
source:https://pyblog.xyz/convolutional-neural-network-binary-image-classification



Convolutional Neural Network

CONTENTS

- Hyperparameters
- Layers
- Architecture
- Dropout
- Batch normalization



Hyperparameters Of ConvNet

- Learning rate
- Number of epochs
- Batch size
- Activation function
- Number of hidden layers and units
- Weight initialization
- Dropout

1.Hyperparameters in ConvNet

There are so many parameters in any network so for configuring them all, some hyperparameters will be used.

– Learning rate-To update the weight in the optimization part where loss function is calculated learning rate is used.

– Number of epochs-The training set passes through neural network and the no of times the whole training set passes through is the number of epochs. According to the gap between the test error and the training error the number of epochs are increased or decreased.

– Batch size-ConvNet is observed sensitive to the batch size,so mini batch is preferred in the processed learning process.

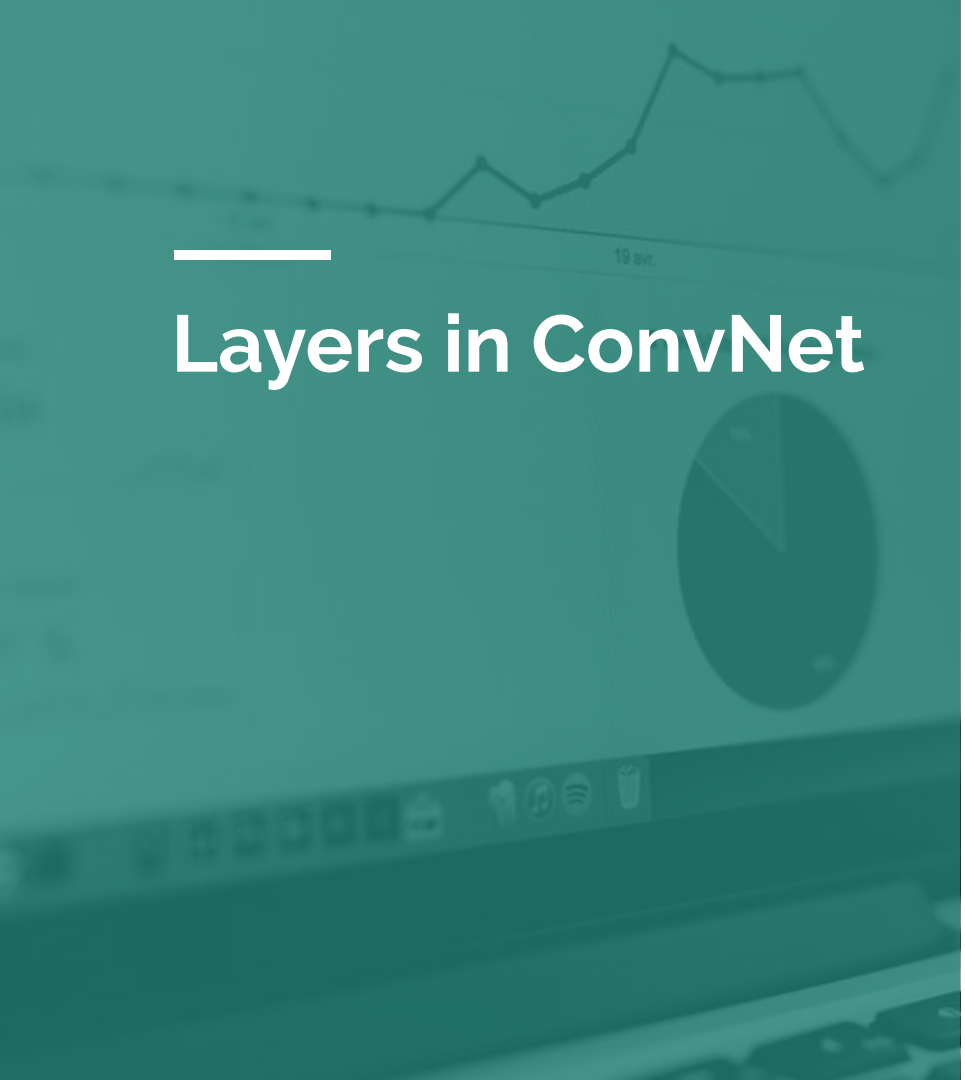
– Batch size-ConvNet is observed sensitive to the batch size,so mini batch is preferred in the processed learning process.

– Activation function-To introduce the non linearity in the model,activation function is used.

– Number of hidden layers and units-It is generally good adding more and more layers and making the network very deep but with the consideration of the improvement of the test error. It becomes computationally.

– Weight initialization-In account of preventing the dead neurons the weights should be initialized with any small number but with taking care of zero gradient so not too small.

– Dropout for regularization-It is the preferable technique to avoid the issue of overfitting in deep neural network.It randomly drops some of the units with respect to some defined probabilities further explained in the last section.



Layers in ConvNet

- Convolutional layer
- Pooling layer
- Fully connected layer

Convolutional Layer

- Overview
- Spatial Arrangement
- Parameter sharing
- Convolution process

Overview

- The core building block of any Convolutional Network that handles most of the computational weight is the Convolution Layer.

- The parameters of CONV layer consists of learnable filters . Every filter is small spatially along the height and the width, but they extends through the full depth of its input volume.

Spatial Arrangement

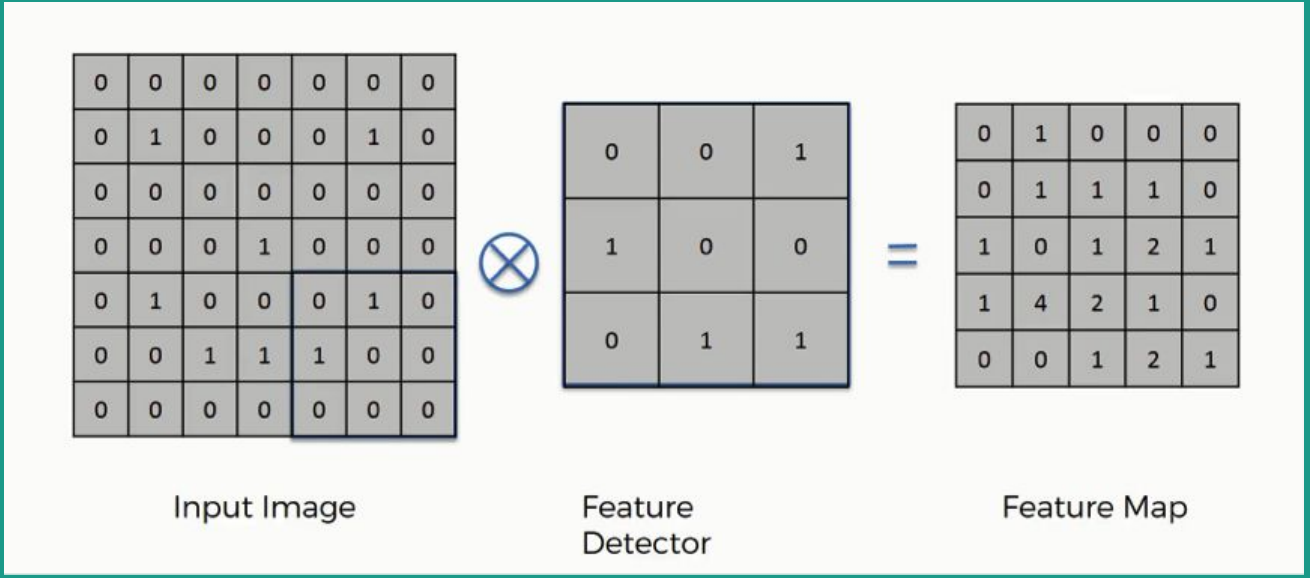
Regarding the number of the neurons in the output volume and their arrangement, there are three hyperparameters that controls the size of its output volume. The hyperparameters are depth, stride and zero-padding.

• Depth - Corresponds to the number of filters that is to be used, each learning to look for some variety in the input. Like in any raw input image that is taken by the first layer of the Conv layer ,while processing it,different neurons along the depth activates in presence of the different oriented edges, or blobs of the color.They are referred to a set of neurons looking at the same region in the input as depth column.

• Stride - To slide the filter, stride is required. When the stride is 1 then the filters moves by one pixel at a time.
Similarly when the stride is 2 then the filters jump 2 pixels at a time. It produces smaller output volumes spatially.

•Zero padding - allows to control the spatial size of the output volumes. It is convenient to pad the input volume with border of zeros.To make the size of input and output volume the same, zero padding can be done.

Convolutional process-



source:https://pyblog.xyz/convolutional-neural-network-binary-image-classification/

➔ During the forward pass, convolution obtained,each of the filter across the width and height of its input volume and dot product is computed between the entries of the given filter and the input at any particular position.

- As convolving the filter over the height and width of the input volume a 2-dimensional activation map is produced that gives responses of that filter at each and every spatial position.

- After that the network learns filters that activate when they notices some kind of visual features like a blotch of any color on starting layer, or an edge of orientation or wheel-like patterns on the other higher layers of the network.

Pooling layer

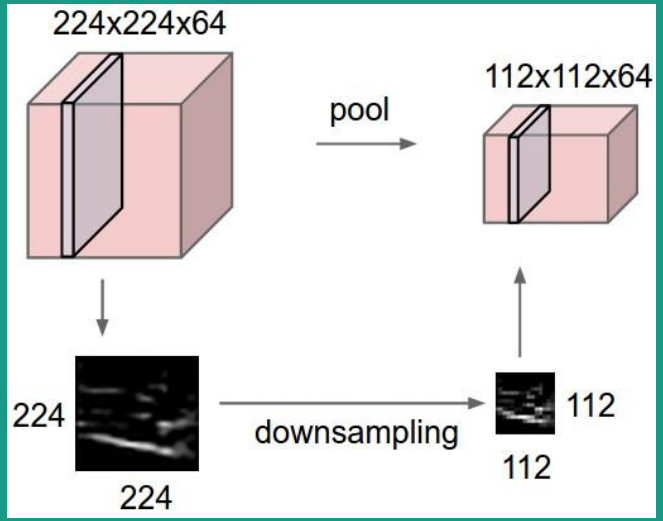
- Overview
- Operations
- Applications

Overview

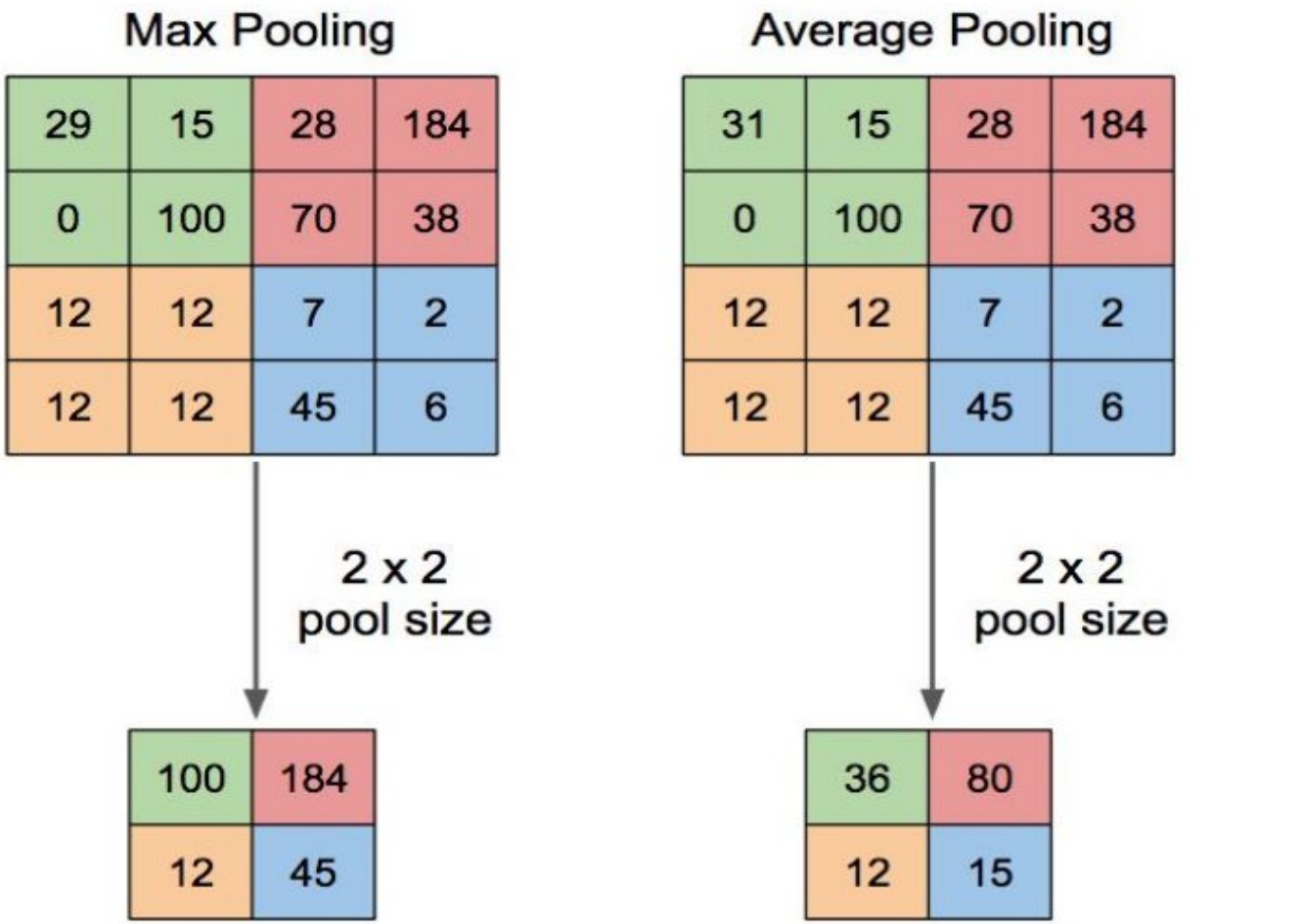
- Non linear downsampling is applied by pooling layer on activation maps.On every depth slice of input pooling layer processed independently and resizes it spatially.
- It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture.
- The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations.

Operations-

➔ The operations that can be performed in pooling layers are MAX pooling, MEAN pooling etc but MAX poolings generally preferred over the others.



source:http://cs231n.github.io/neural-networks-1



source:https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/

Backpropagation.

The backward pass for a max(x, y) operation has a simple interpretation as only routing the gradient to the input that had the highest value in the forward pass. Hence, during the forward pass of a pooling layer it is common to keep track of the index of the max activation so that gradient routing is efficient during backpropagation.

Advantages-

- Its function is to progressively reduce the spatial size of the representation.
- To reduce the amount of parameters and computation in the network, and hence to also control overfitting.

Is it learnable?

NO

Introduces zero parameters since it computes a fixed function of the input

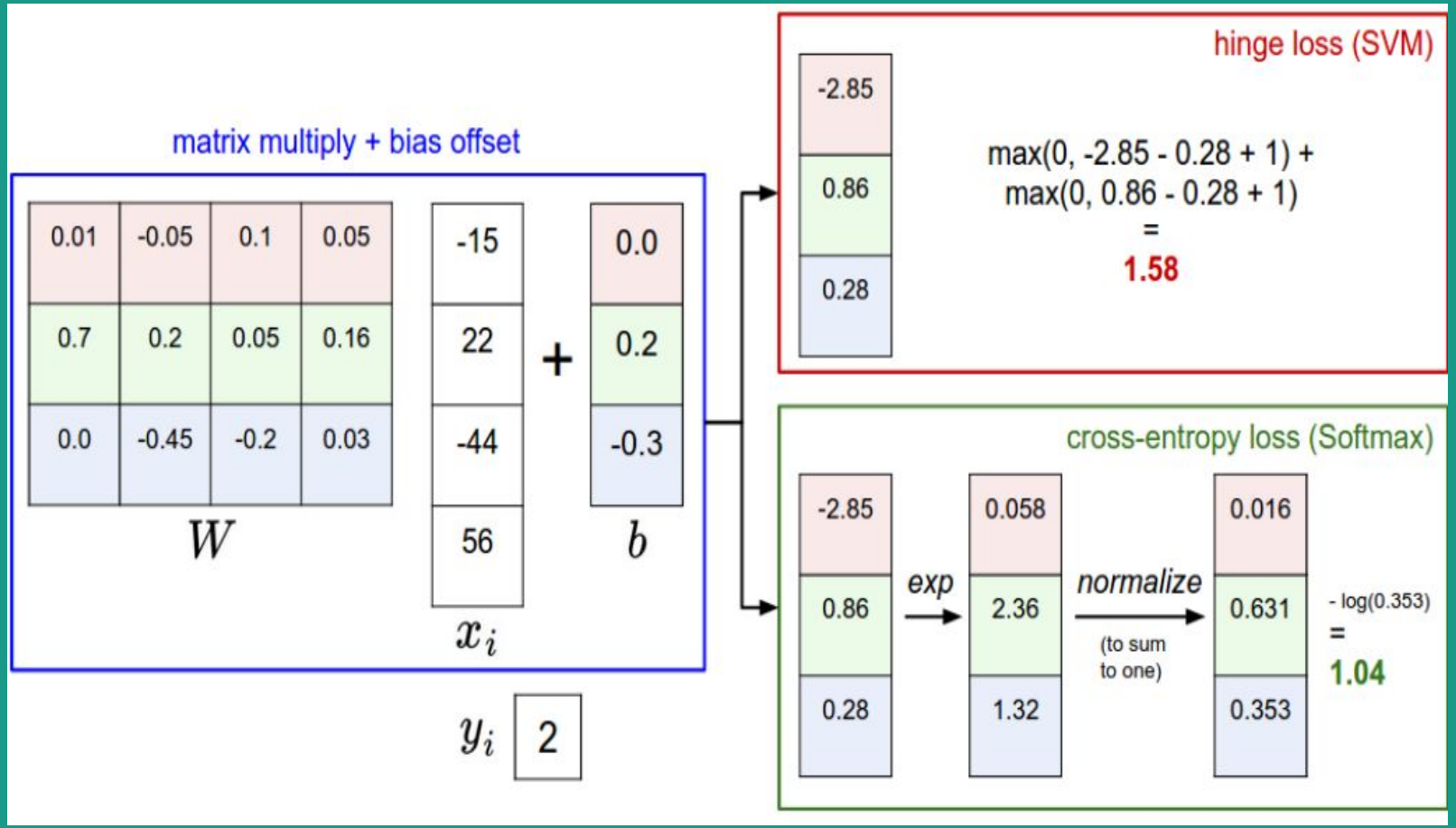
Fully Connected Layer

- Unlike regular neural networks, neurons have full connections to all the activations in its previous layers.
- The activation output in the fully connected layer can be obtained by matrix multiplication followed by a constant bias offset.
- The obtained common output is a vector that afterward passes through softmax functions to represent confidence of classification by using probability normalization.

Softmax-

- ➔ It has probabilistic interpretation and it gives more intuitive output as uses the normalized class probabilities.Softmax keeps the function mapping unchanged but the scores are interpreted as unnormalized log probabilities.

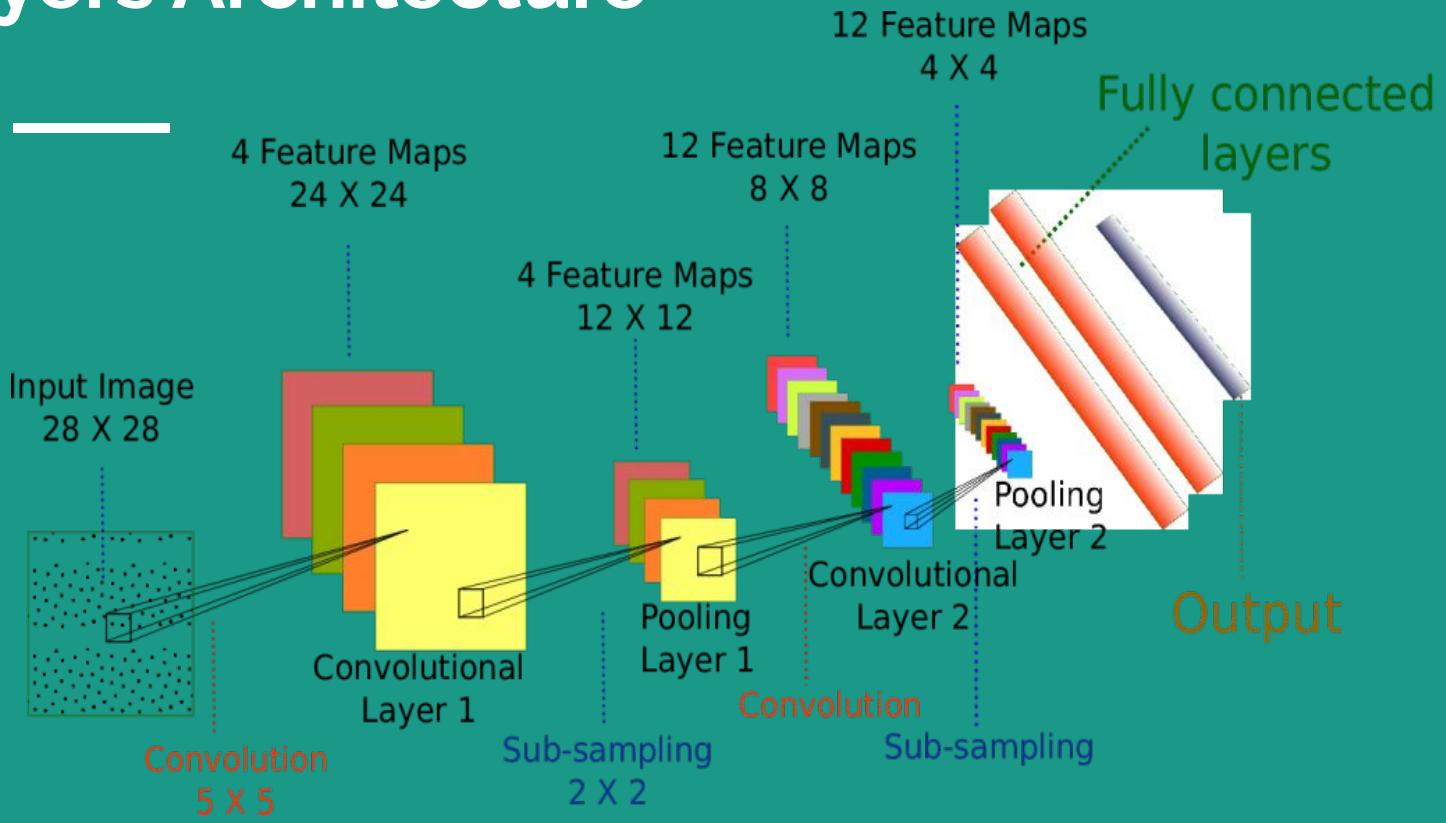
- ➔ It is calculated for each of the class and scores are replaced by hinge loss and that too with a cross entropy.



source:https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd

- ➔ **The SVM does not care about the details of the individual scores.**
- ➔ **The SVM is satisfied once the margins are satisfied and it does not micromanage the exact scores beyond this constraint.**
- ➔ **Compared to the Softmax classifier, the SVM is a more *local* objective.**

Layers Architecture



source:https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/

CNN Architecture

- Layer patterns.
- Layer sizing patterns.

Layer patterns-

The most common ConvNet architecture follows the pattern:

INPUT -> [[CONV -> RELU]*N -> POOL]*M -> [FC -> RELU]*K -> FC

- * Indicates repetition

Eg.
INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL]*3 -> [FC -> RELU]*2 -> FC
Two CONV layers stacked before every POOL layer.

Layer sizing patterns-

Why use stride of 1 in CONV?

Why use padding?

Stride 1 allows us to leave all spatial downsampling to the POOL layers, with the CONV layers only transforming the input volume depth-wise.

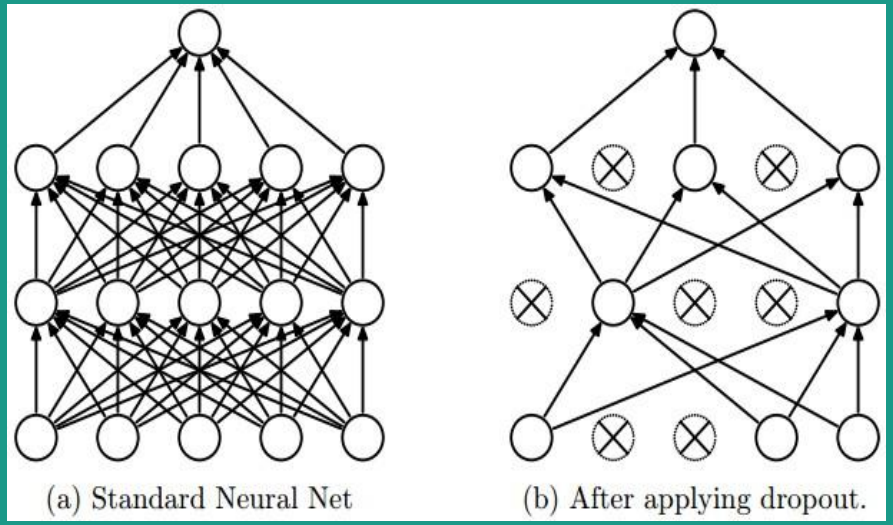
If the CONV layers were to not zero-pad the inputs and only perform valid convolutions, then the size of the volumes would reduce by a small amount after each CONV, and the information at the borders would be “washed away” too quickly.

Dropout

It temporary removes the units from the network and choice of those units are completely random.It removes the units from all its incoming and outgoing connections.

The choice of dropping the units in its simplest case is when each unit is retained with some fixed probability p which is independent from other units.The value of p can be chosen 0.5 as optimal for wider range of tasks and the networks.

Dropout samples the neural network into a thinned network. The final output of the thinned network contains all the units that survived in the dropout.



source:https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd

Dropout samples the neural network into a thinned network. The final output of the thinned network contains all the units that survived in the dropout.

Batch Normalization

A part of model architecture is normalized and also for each training mini batch normalization is performed.

High learning rates are obtained after batch normalization and also it is less careful about the parameter initialization.

It can also reduce the need of the dropout so acts as a regularizer.

Any issues?

Yes

Vanishing gradient problem

Vanishing gradient problem

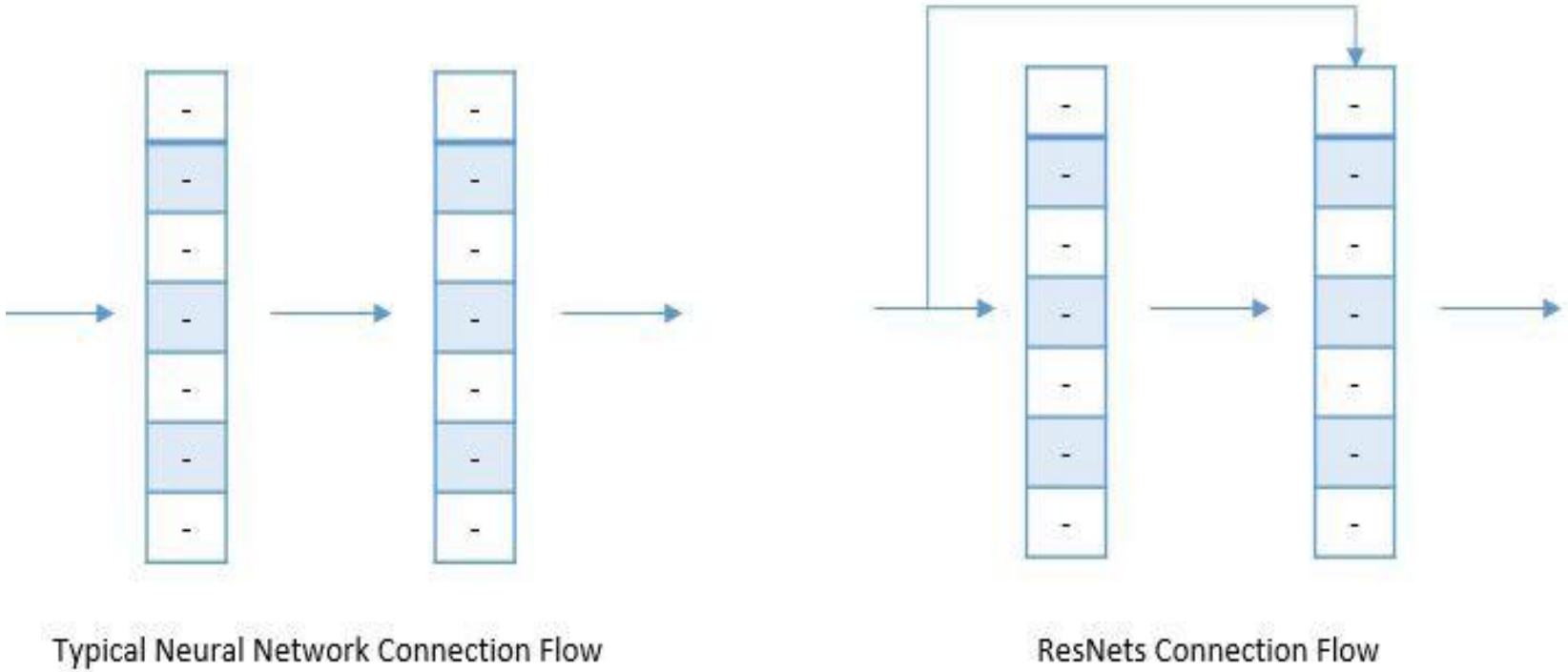
The method involves the weight updation proportional to the partial derivative of an error function respect to its current weight on each of the iteration of the training.

In some cases the gradient can be vanishingly very small, so it effectively prevents the weight from changing the values.

This problem can completely stop the neural network from its further training in its worst case.

What is the solution ?

RESIDUAL NETWORK

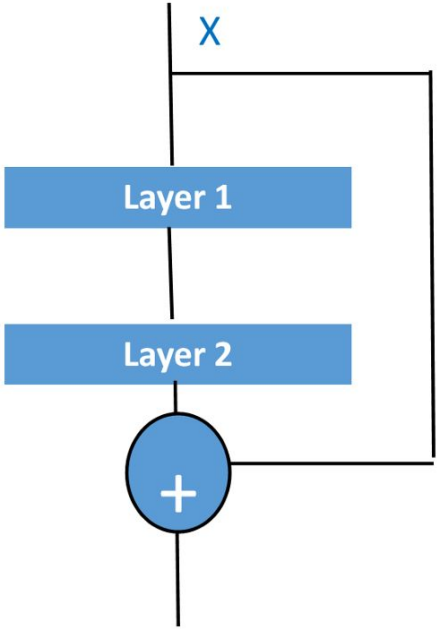


source:https://towardsdatascience.com/residual-networks-resnets-cb474c7c834a

ResNet

In the residual connections,the output of previous layer is connected to the output of the new layer.

Assume network of any n layers, then in this residual setup the output is not only passed from initial layer to the next layer but also adding up the output of the initial layer to the output of the second layer.



Each layer - f(x)

In any standard network y = f (x)

In a residual network y = f (x) + x

source:https://towardsdatascience.com/residual-networks-resnets-cb474c7c834a

Overall, ResNets breaks down a deep neural network intothe small chunks of network which were connected through shortcut connections to form the bigger network.

References

[1] http://cs231n.github.io/neural-networks-1

[2] http://cs231n.github.io/convolutional-networks

[3] http://cs231n.github.io/transfer-learning

[4] http://www.cs.toronto.edu/ rsalakhu/papers/srivastava14a.pdf

[5]https://towardsdatascience.com/residual-networks-resnets-cb474c7c834a

[6]https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd

[7] https://new.idcbest.com/dsj/dsjdx/11925.html

[8] https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/

[9] http://cs231n.github.io/neural-networks-2/

Thank you

Swati meena,183070009
Department of Electrical Engineering
IIT BOMBAY