Machine learning. Convolutional Neural network (convnets)

Pooran Singh Negi

University Of Denver

August 14, 2018

Outline

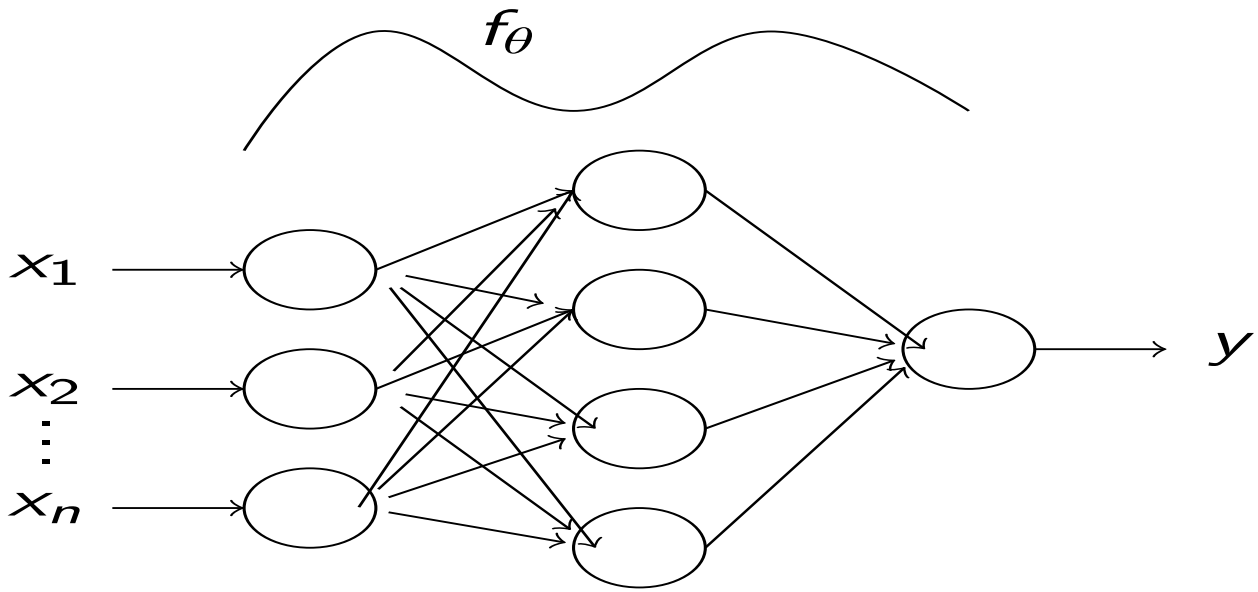
- Convolutional neural network
- Regularization of neural network

Last time: What is Neural Network and deep learning

Neural Network: Biological Neuron inspired, mathematical model. Inspiration is beautiful if you belive in connectionism.

Deep learning: Set of techinque for training deep neural network.

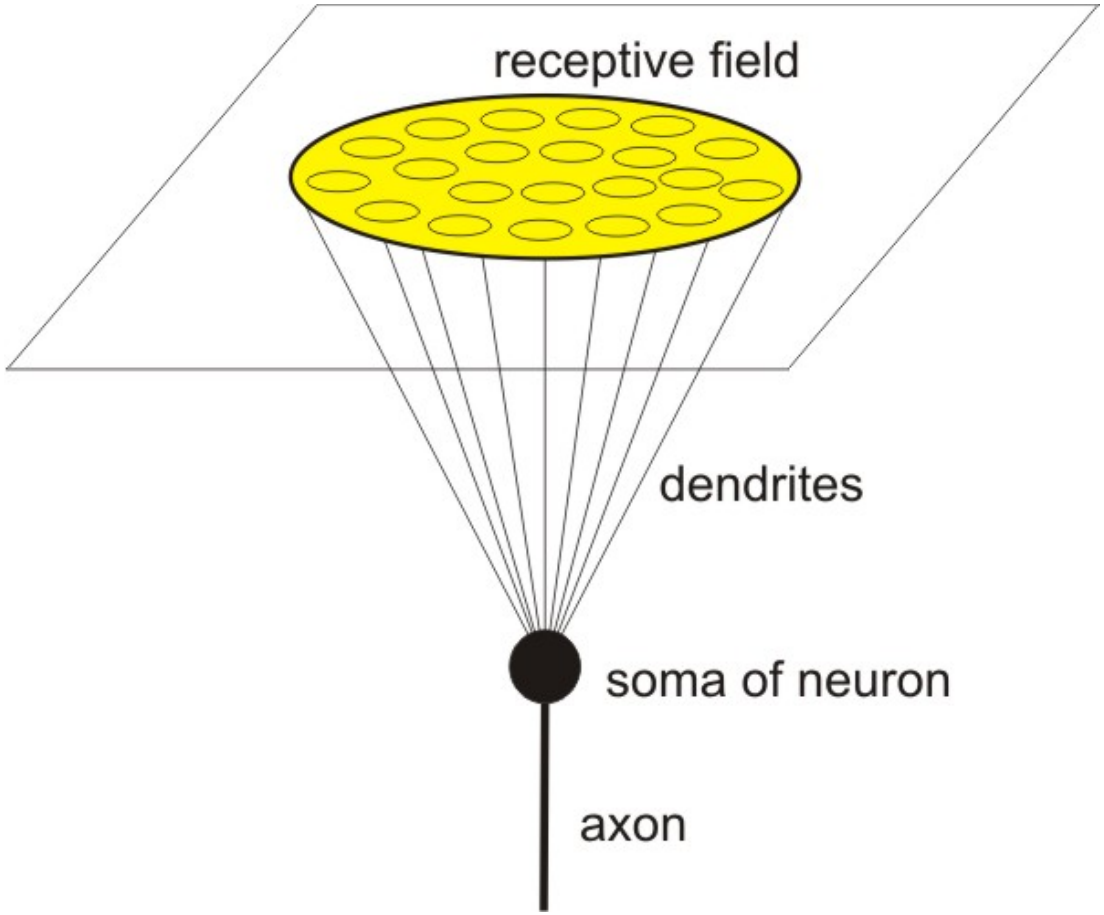
Feedforward Neural Networks(Fully connected layer)



End to end classifier

- Neural networks computes the function $y_i = f_\theta(x_i)$.
- Highly non linear end to end.
- Parameters θ can be learned via gradient descent by minimizing $\sum_{i=1}^{N}(y_i - f_\theta(x_i))^2$.

- Convolutional neural networks are widely used in vision applications.
- Convolutional neural network are inspired by visual processing in the brain.(Look into the research of D.H Hubel and T.N Wiesel) and Hubel and Wiesel Cat Experiment
- Simple visual neuron cell respond to lines at various orientation
- complex cell respond primarily to oriented edges and gratings, however it has a degree of spatial invariance and more receptive field.



neuron receptive field. Source: http://neuroclusterbrain.com/neuron_model.html

How to train machine to learn pattern in the images

f

```
[[ 79 100  70 ...  95  35 105]
 [ 61  95  65 ... 101  71  90]
 [ 78  79 104 ...  61  42  58]
 ...
 [ 41  32  24 ...  82  57  53]
 [ 26  34  44 ...  53  82  32]
 [ 20  22  46 ...  52  35  71]]
```

f[0,0]=79

f[1,2]=65

What machine see.

Source: https://unsplash.com/photos/rJOuDk5yX6U

Convolution emulates the function of visual neuron and its recpetive field.

- For 1-D signal convolution is

$$f[n] * g[n] = \sum_{i=-\infty}^{+\infty} f[i]g[n-i]$$

or for 2d signal(like an image)

$$f[m,n] * g[m,n] = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} f[i,j]g[n-i, m-j]$$

Think of previous operation as

- sliding a filter(kernel) f over the image g
- For each poistion of kernel, multiply overlapping values of f and g and put the sum of the product at the output location(image) where kernel is centered.

Note if we don't flip filter f and directly multiply and sum overlapping value then it is called correlation. let say $\circ$ denotes correlation operation than $(f \circ g)[n] = \sum_{i=-N}^{i=N} f[i]\, g[n+i]$. similarly for 2D

Notice the difference of $+$ and $-$ in g in correlation and convolution definition

Although it is called Convolution in deep learning literature and library. Most of them implement correlation. It doesn't matter if network learns correlation or convolution weight. Their filter are 'just flipped version of each other'

let do this correlation in 1D

$g = [40, 50, 10, 20, 30]$ $f = [1/3, 1/3, 1/3]$

then output $f * g$ is $[30, 33.333, 26.66, 20, 16.66]$ $= f * g$

What is this filter f doing? See the Notebook demo

$1/3 \; 1/3 \; 1/3 \rightarrow f$

$? \; 40, 50, 10 \rightarrow g$

$= 0 \times \frac{1}{3} + 40 \times \frac{1}{3} + 50 \times \frac{1}{3}$

$1/3 \; 1/3 \; 1/3$ - padding

$40 \; 50 \; 10$

$= \frac{1}{3}40 + \frac{1}{3}50 + \frac{1}{3}10 = 33.33$

$()(0) = f * g(0)$

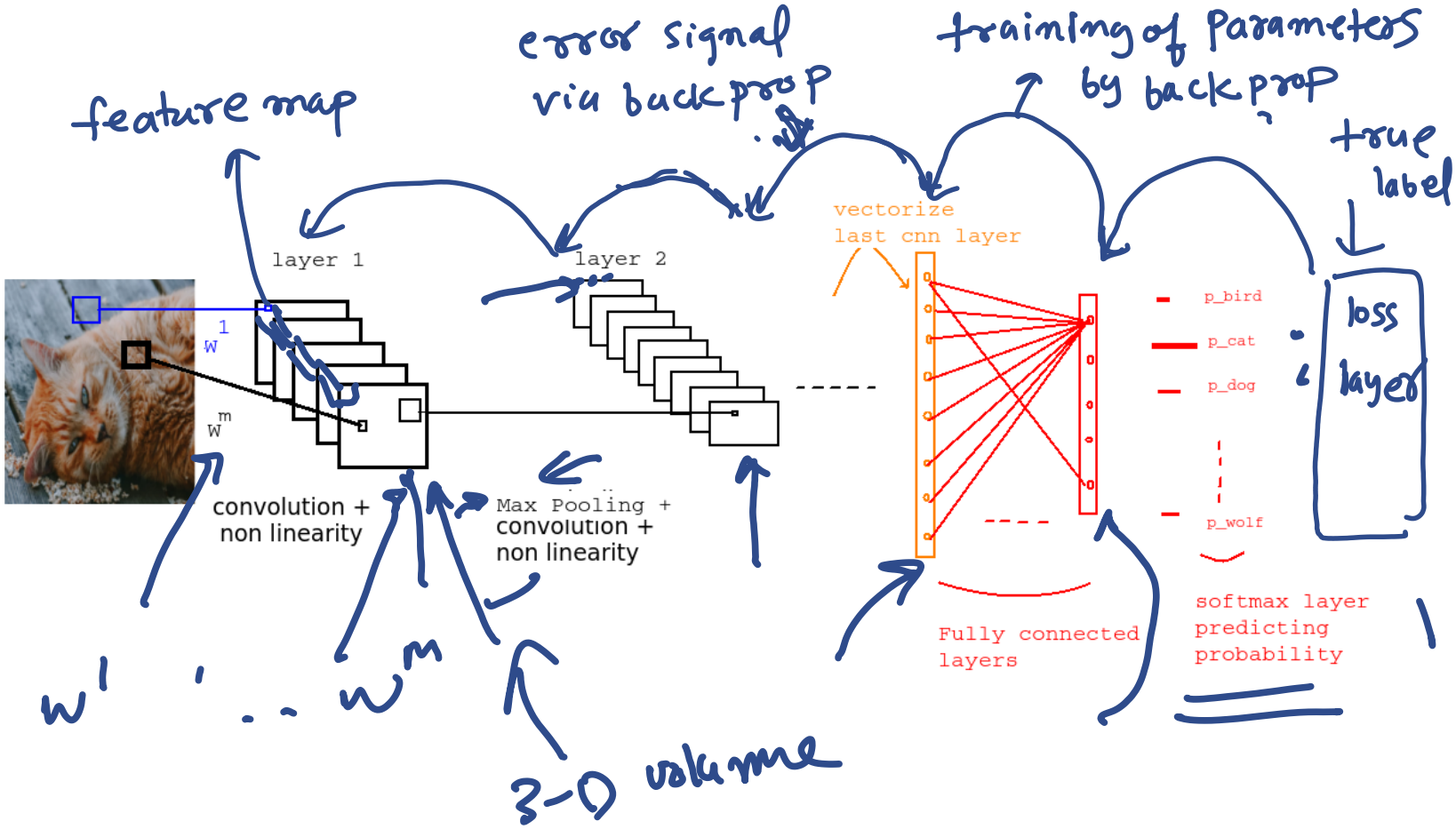
$= g \times f(1)$

- A Visual neuron i can be parameterized by a 2,3 or n-dim filter. A 2-d 3×3 filter is $W^i = \begin{bmatrix} w^i_{11}, w^i_{12}, w^i_{13} \\ w^i_{21}, w^i_{22}, w^i_{23} \\ w^i_{31}, w^i_{32}, w^i_{33} \end{bmatrix}$

weight

kernel filter

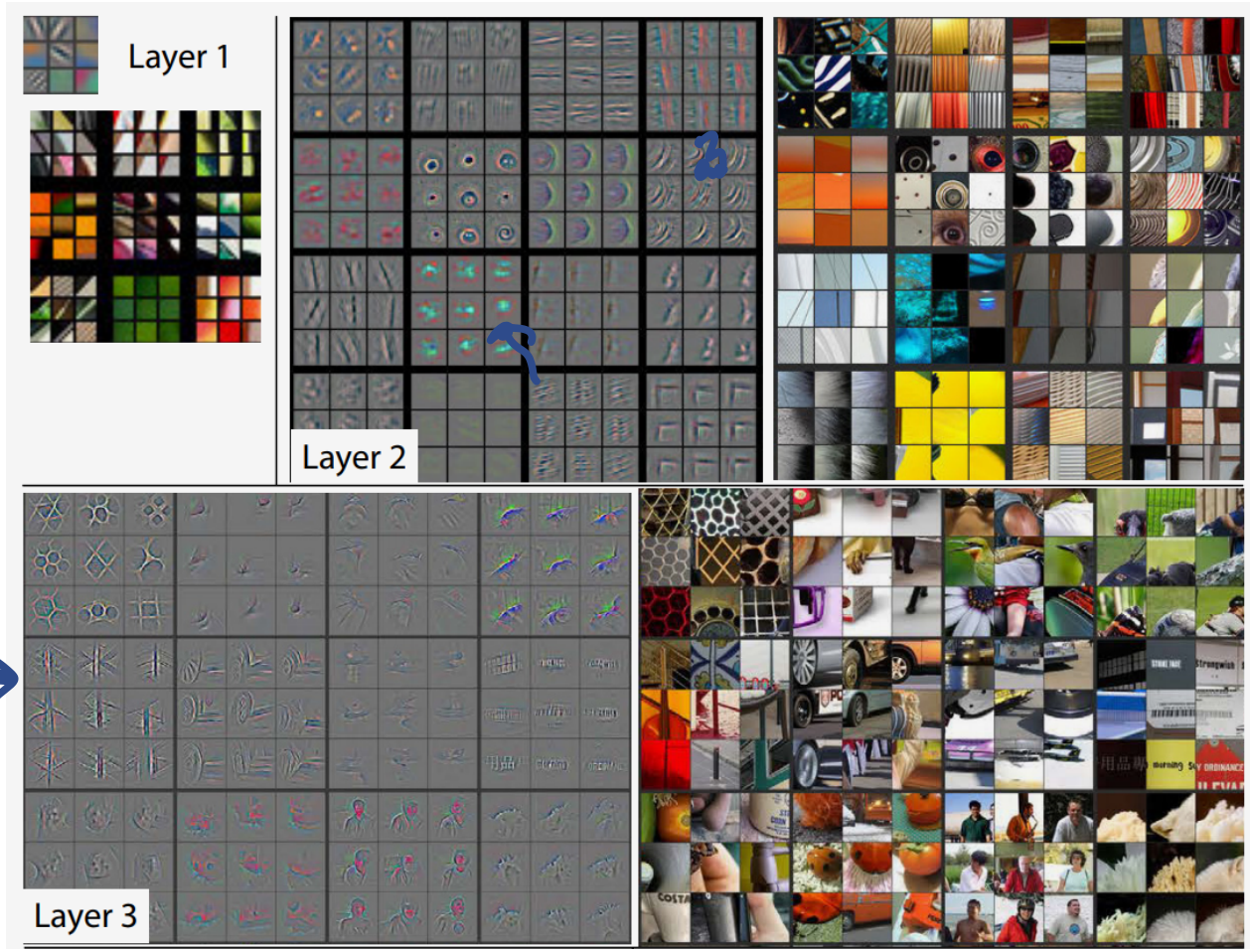
CNN Architecture



Vanilla CNN architecture for classification you will see in deep learning papers.

- Unlike typical image processing filters, weight W_{jk}^{i} are trainable by back propogation alogorithms.

Feature learned by various CNN filters



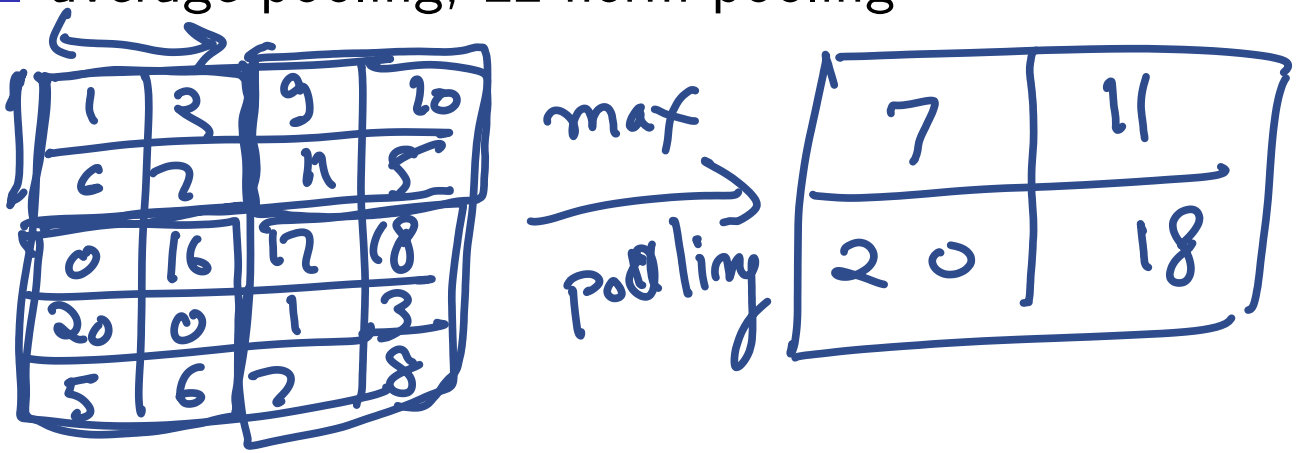
Features leared by various CNN layers.

Source Zeiler Fergus https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf

What is maxpooling pooling

Reduces spatial dimension, provides invariance to translation, rotation and scaling.
There are different types of pooling like

- max pooling
- average pooling, L2 norm pooling



maxpooling calculation

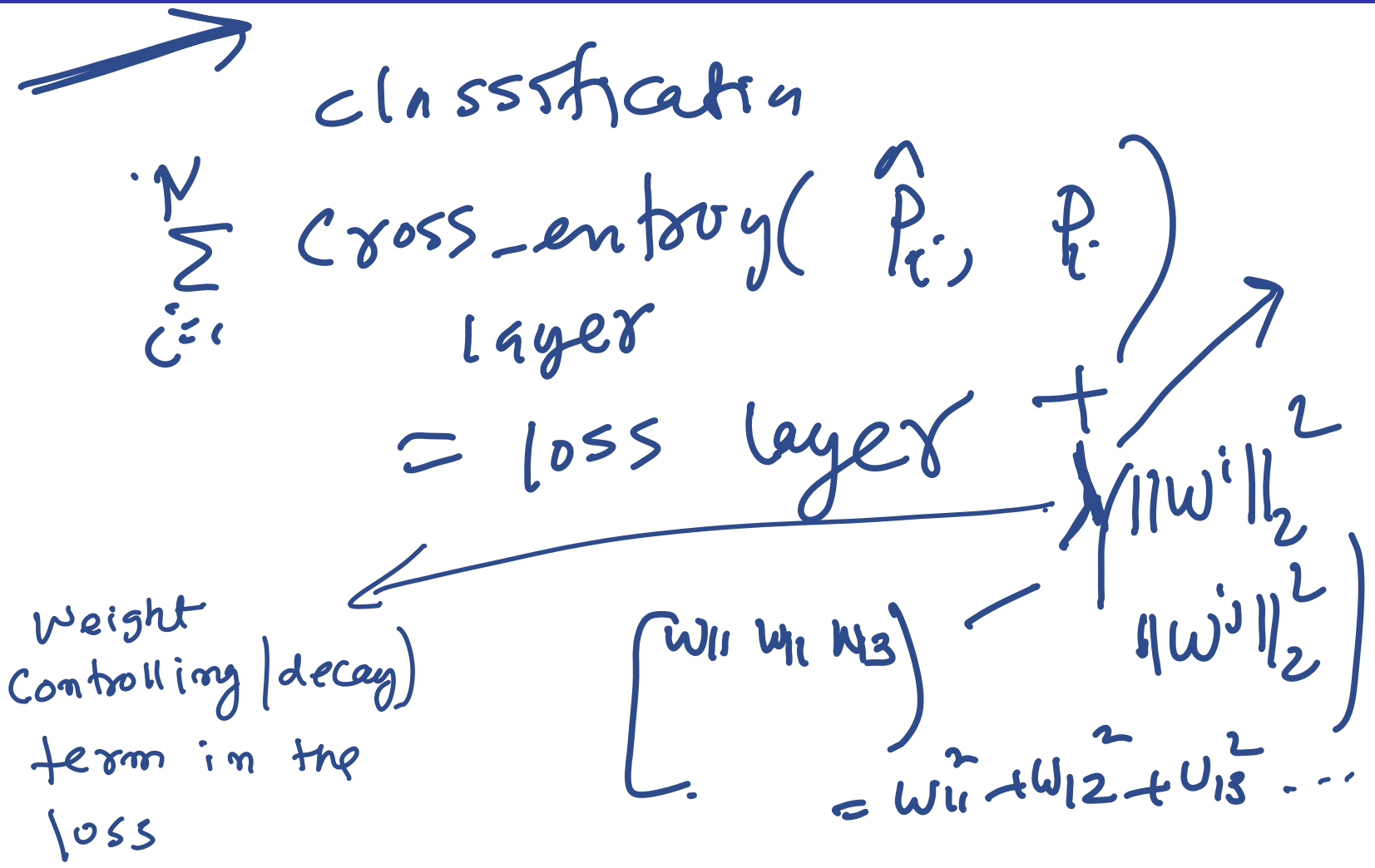
In designinig CNN we need to take care of following hyper parameters

- Number of output feature map from cnn layer(called depth). Same as number of filters used in the layer
- Stride: Slide step size of filters. with stride 1 we move filter by 1 pixel. with stride 2 we move filter by 2 etc. Clearly when we move by higher stride we get smaller volume.

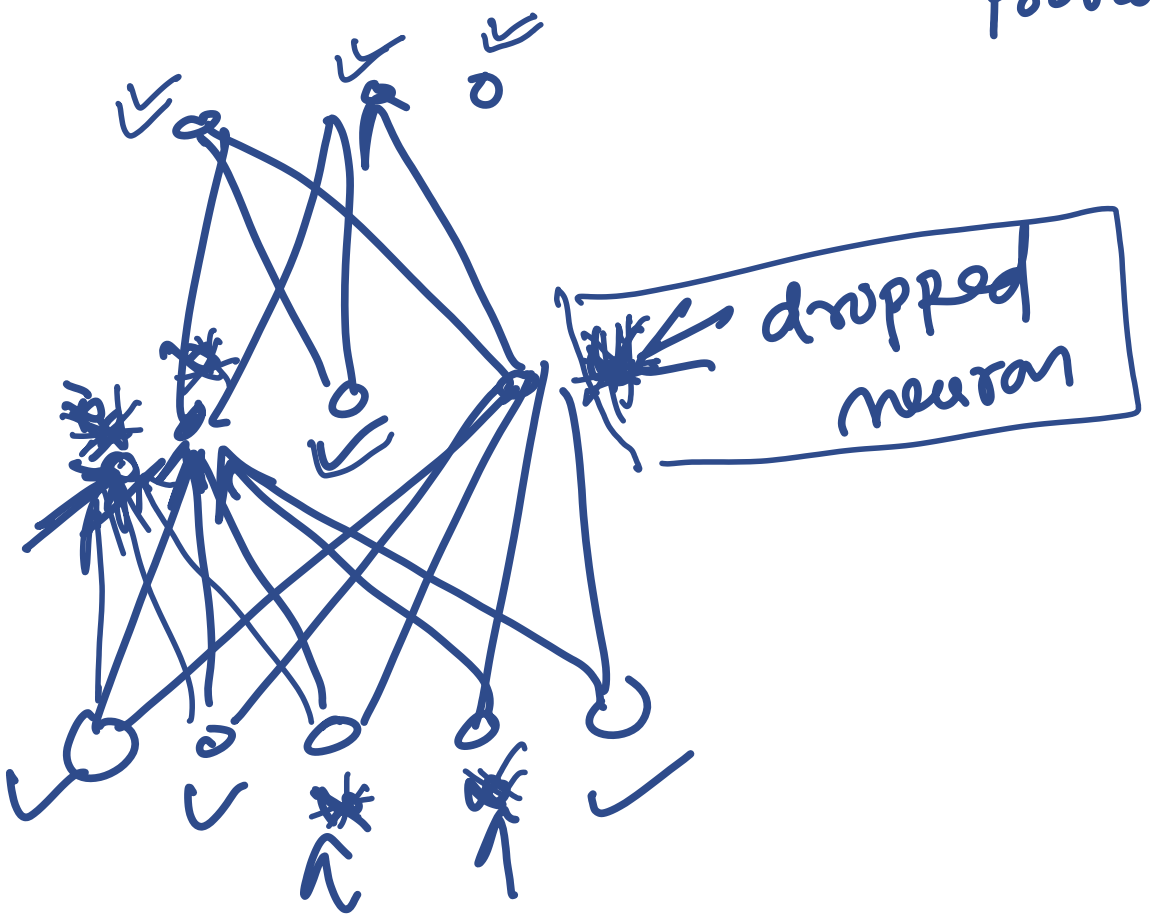
In designinig CNN we need to take care of following hyper parameters

- Padding. For region outside the filter, how should we treat the input(image). Some option are zero padding, same padding, symmetric
- pooling size. Should use 2x2 or 3x3 and type of pooling, max, or average or L_2 etc.

Regularizing neural network

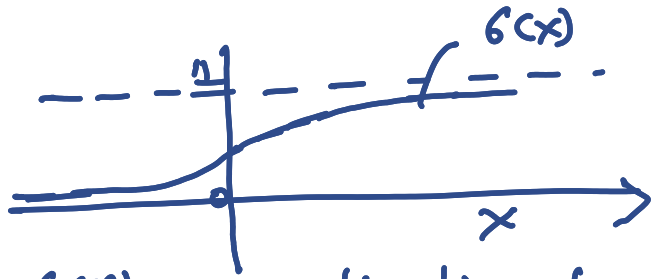


→ Dropout = you drop the neuron with probability p



Why Relu activation

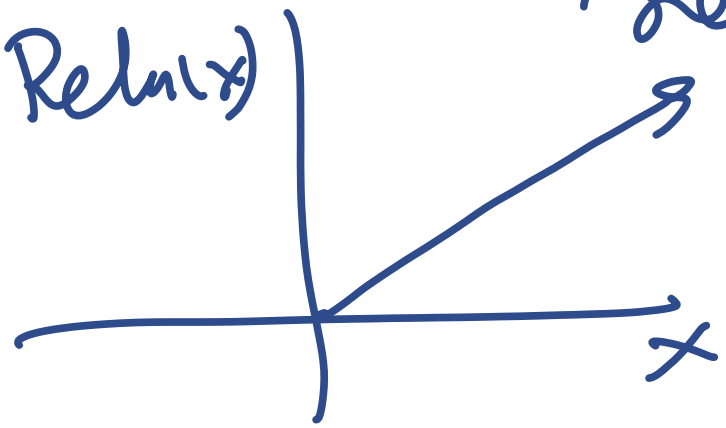
$\sigma(x)$ reach value close to 1 pretty soon and derivative becomes close to zero. If we use $\sigma(x)$ as activation function then their is high chance that no gradient (error signal) will flow between layers of deep neural network and network will not learn. this is called saturation problem in $\sigma(x)$



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Solution? use Relu activation.

$$relu(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$



$$relu'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

derivative

Thank you!

check the website later for more information about CNN. This is dynamically evolving field. You have to see the papers in vision and machine learning Conferences.