NOTES

Neural Networks.

Multilayer Perceptron

↓

1989 Convolutional Neural Network.

↓

Rename neural network as

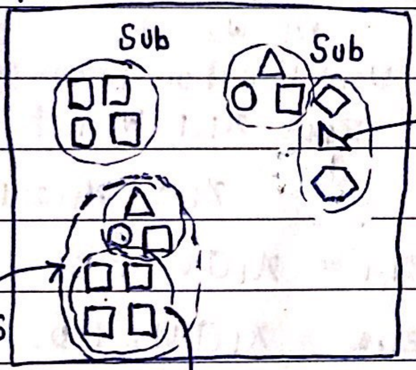
2010 Deep learning - A form of machine learning where model has multiple layers.

↓

2013 CNN + Graphic processor unit + Image net (A data set of million of images)

$\underbrace{\text{Graphic processor unit}}_{\text{Computational Platform.}}$

Conseptual CNN



Atomic elements.

Motifs

Sub motifs (repeated combinations of atomic elements)

CNN assuming above image as input → Classification.

Stack of feature maps one for each → ○ ← motif filters

sub motif shape

Stackof feature maps one for each atomic shape. → ○ ← Sub motif filters (Mini Stacks of atomic filters)

Convolution - Moving each filter to every location of input image

Image above → ○ ← Atomic shape filters

NOTES

Feature maps - Degree of correlation /match between the filters and input image. If the filter is present at a certian location of the image there is a high correlation / amplitude / match. in feature map.

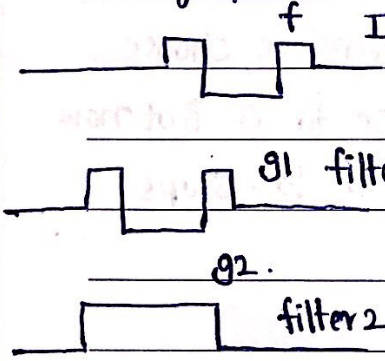
When it comes to motif and sub motif filters each filter in the mmi stack of atomic filters (for sub motif) or sub motif filters (for motif) will be convoled with its respective feature maps from previous convolution. And if these motifs or sub motifs are present in the image these feature maps will have strong correlation / amplitude nearby in space.

Convolution?? Sliding 2 signals one over another.

$$f*g[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] \quad \text{(Discreate)}$$

When it comes to filter x image convolution.

1D. graph. (equals to autocorrelation)



f Input signal (image)

g1 filter 1

g2. filter 2.

f * g1 - Since there is a match. convolution (ഉയരും) will be highest when f and g1 superimposed.

f * g2 - Since there is no match convolution will not be highest at any point.

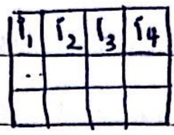
(equals to cross correlation)

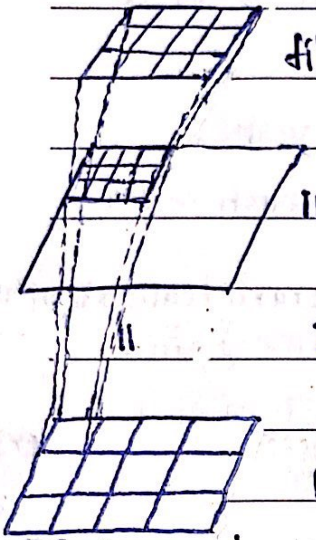
f * g1 - This is a graph starting from a certian value increase to its highest value (area of g1 graph neglecting negative just summing up) then gradualy decreasing to it startup value.

Scanned with CamScanner

2D convolution (In images).

filter



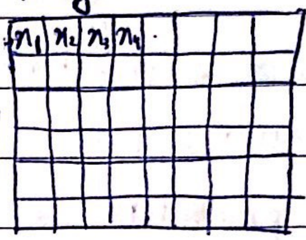


filter (each pixel has a value)

x (pixel values x image pixel).

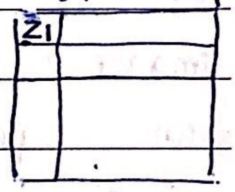
image $\left[z_1 = x_1 \cdot i_1 + x_2 \cdot i_2 + x_3 \cdot i_3 + \cdots \right]$

image.



Σ (those multiplications are summed)

Feature map



Move the filter through every position of image. and get the feature map.

Feature map / heat map.

When there is a perfect with the filter correlation value on feature map will be highest (auto correlation. highest value)

Feature map is a map with correlation values (high and low)

These filters may correspond to varies features of the image. (Ex:- different shapes, edges, ..)

Gives a label to In — Classifier — $C(T_n, W) \leftarrow T_n = f(N_n ; \alpha_1, \alpha_2 \cdots)$

$\uparrow$ layer 2 feature maps

W - Classifier parameter

$\alpha_1, \alpha_2, \cdots \rightarrow O \leftarrow N_n = f(M_n ; \psi_1, \psi_2, \ldots)$

layer 3 filters

$\uparrow$

(1st layer feature map stack of nth input) $M_n = f(I_n ; \phi_1, \phi_2, \cdots \phi_K) \rightarrow O \leftarrow \psi_1, \psi_2 \cdots$

$\uparrow$ K feature maps. — layer 2 filters

I_n - Input image $\rightarrow O \leftarrow \phi_1, \phi_2, \cdots \phi_K$

layer 1 filters

Even though we treat here I_n as a singal image in real they have colors.

So I_n will be a stack of input images. One image per a fundamental color (red, green, blue). layer 1 filters will also be mini stacks.

One mini stack will have 3 filters one for each color.

NOTES

How to get these filter and parameter values?

AVGL should be min. For massive data set · SGD.

Classifie predicted value

$$\frac{1}{N}\sum L(C_n, y_n) = E(\phi, \psi, \alpha, W)$$

True label

When AVGL min $\hat{\phi}, \hat{\psi}, \hat{\alpha}, \hat{W}$ will be our final parameters $\nabla AVGL = \nabla E$ (multidimentional gradient).

$\theta_{t+1} \leftarrow \theta_t - \alpha \nabla E_{\theta}(\theta_t)$ Gradient is taken with respect to each parameter we have to learn.

(the parameter we have to learn)

Why send In through several layers? Why not get the convolution with layer 3 filters directly?

If so we will have to learn layer 3 filters independently But this way we can share knowledge and use data more effectively.

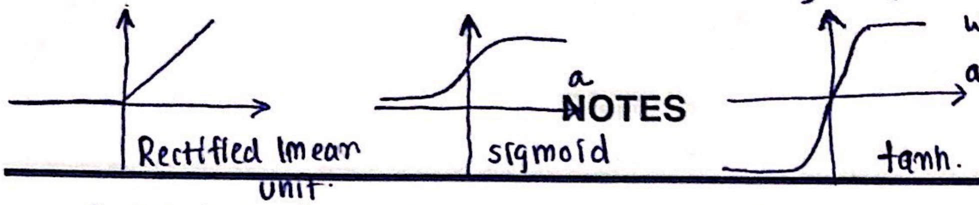
Since layer (3 filters) motifs ~~are~~ manifested sub motifs (layer 2 filters) learn in one motif will provide informations on other motifs too. (Same sub motifs are repeated in other motifs.)

Or we can even use previously input data paramaters for later input data low level parameters.

Fundamental elements of CNN

1. Convolution layers. - filter size - (3x3), (5,5) Cannot be too big. If its too big cannot capture small features of.
   - filter stride - How many squares the filter move at one time.
   - filter (feature) number - What feature we are looking for in the image.
   - When filter stride increase → feature map size reduces. (less no of pixels).
     ∴ Reduce computational load.

Scanned with CamScanner



NOTES

when a is very high or very low there won't be a significant change in output

2. Activation function - We use non linear activation function.

z_i is the value of a pixel in feature map.

z_i is sent through a non linear activation function (NLAF) before inputing the next layer.

Then the value of of pixels in feature map will be functions of that NLAF.

Why? Increase functional capacity of neural network.

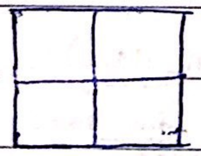
~~(Let the output be O_m)~~

3. Pooling - After convolution and after activation.

Reduce the input size ~~of~~ passed on to next layer. ∴ computation complexity reduces.

Combats overfitting.

Pooling filter encourage translational invariance.



Has a window size (2x2), (3x3)...

Stride.

Chooses the max (or avg) value of the output (O_m) within that window size and move a step of stride.

Image $\xrightarrow{\text{Convolution}}$ Feature map layer ① $\xrightarrow[\text{Convolution}]{\text{Activation + Pooling +}}$ Feature maps layer ②

↓ Activation + pooling.

If the last output of the CNN is $m \times m$ size N pooled maps.

← Vectorization ← $m^2 N \times 1$ input to the full connected layer ← Logistic regression or MLP.

(All the input points are multiplied by weighs and summed)

NOTES

CNN are used. for – medical image analysis.

To segmant out perticular features of an image (TSA screening airport)

Create automatic 3D surface meshes.

In playing games. like "GO" through image. analysis.

We can use transfer learning. for parameters.

(Because of hierarchical structure)

We can use parameters that we have learnt previously in a imagenet by transferng them to the medical image analysis. We only have to learn the parameters at the top (classifier parameters) of the network which are directly connected to the medical image. Why are these low level filters can be transfered? (boild up from gabor func)

Because these are very similer neuro receptive fields of mammels.

To turn on a signal visual cortax of that mammel what should be the shape of the light that is shined to mammel's eyes. Fundamental shape visual cortax can recognize.