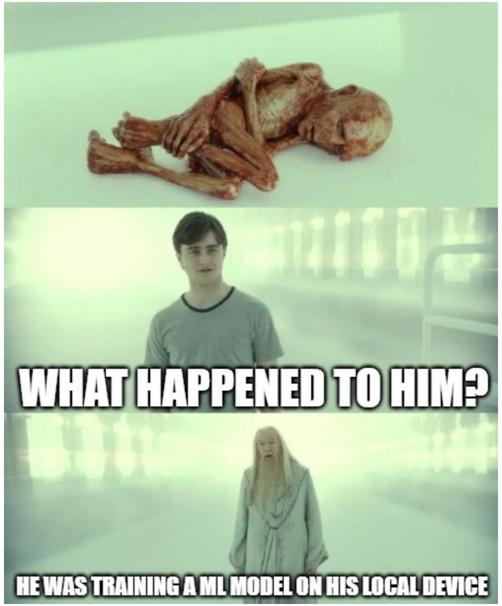
Data Augmentation, and Auto-Encoders

B.Tech. Data Science, NMIMS

By,

Bilal Hungund, Data Scientist, Halliburton



Convolution operation

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

6x6 image

$\times$

Filter (Weights)

-1	1	-1
2	3	2
1	1	1

3x3 Filter

$\Downarrow$ Output without padding

$n - f + 1$

$= 6 - 3 + 1$

$= 4$

Output Size

4	4	4	4
4	..	..	..
:	:	:	:
:	:	:	:

4x4

Stride = 1 (step size)

With Padding

Padding

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	1	1	1

$\times$

filter

1	-1
1	-1

$\Downarrow$

4x4

$4 \times 4 \xrightarrow{\text{padding}} 6 \times 6$

Pooling

25	48	11	58
192	10	20	110
38	0	9	31
50	8	23	47

Stride = 2
(Recommended for Pooling)

25	48
192	10

11	58
20	110

Pooling ⟹

38	0
50	8

9	31
23	47

Max Pooling

192	110
50	47

69	50
22	28

Average Pooling

Backpropagation in CNN

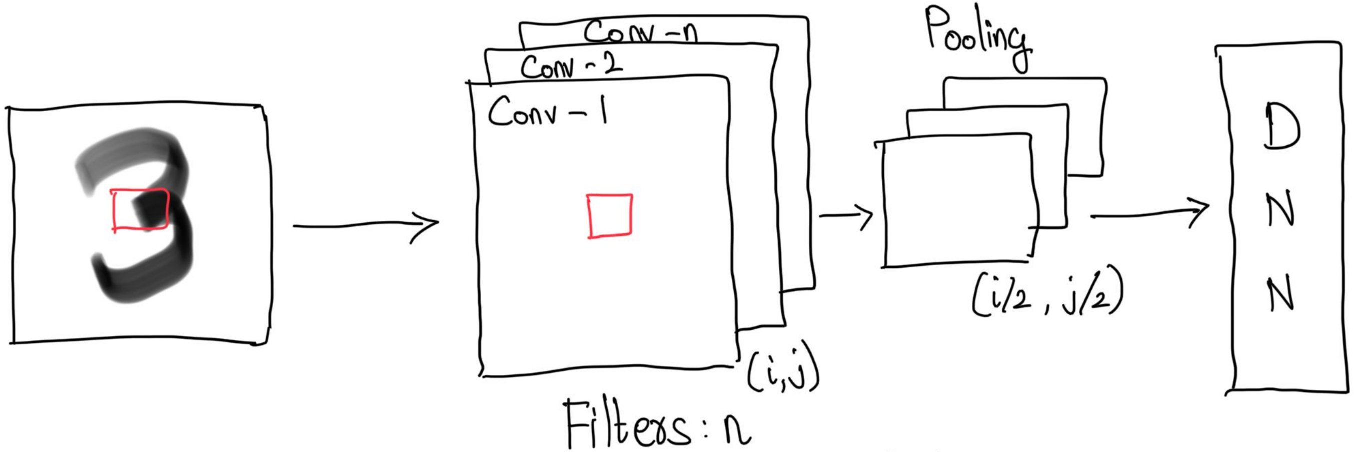
https://github.com/bilalProgTech/btech-nmims/blob/e0aead64d036c2dc2862d7036175b6eea5d1f66b/NN-DL/Class-Presentations/Back-propagation-CNN.pdf

Convolution Neural Network (CNN) for Classification

tf.keras.layers.Conv2D

tf.keras.activations.*

tf.keras.layers.MaxPool2D



1) Convolution: Filters to generate feature maps

2) Non-linearity: often relu

3) Backpropagation

4) Pooling: Downsampling feature maps

Image Augmentation

→ Simple and powerful tool to help you avoid overfitting

→ If data and its scope is limited then chance of potential future prediction is also limited

→ Example : You have a dataset of cats but in testing set you have a cat lying down. Thus making difficult for model to recognize

Solution : Rotating the images

Tensorflow Image Data Generator

→ rotation_range [0 – 180°]

→ width_shift_range
height_shift_range } shifting [0 – 1]
portion of shifting

→ shear_range [0 – 1]

→ Zoom_range [0 – 1]

→ horizontal_flip [True , False]

(rescale , fill_mode)

Auto-Encoders

→ It is a special type of neural network architectures in which the output is same as the input.

→ It is trained in an unsupervised manner in order to learn extremely low representations of an input data.

→ These low level features are deformed back to its actual data.

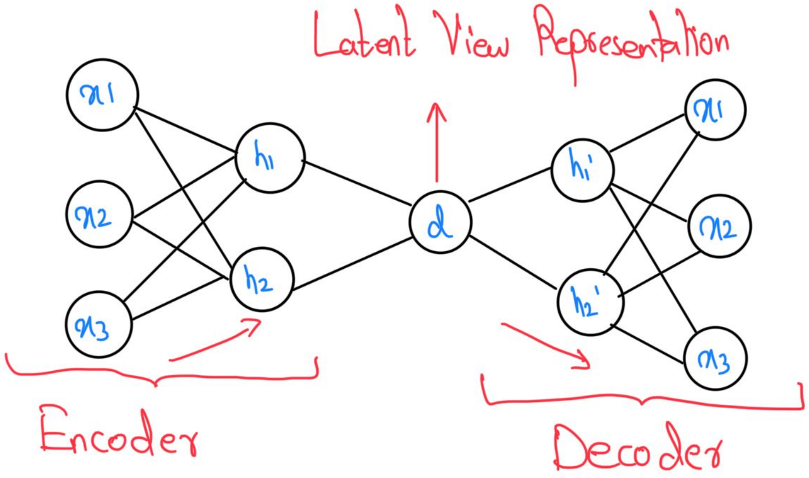
→ It is a regression task where the network is asked to predict its input (model identity function).

Auto-Encoders architecture

Encoding architecture: It comprises series of layers with decreasing number of nodes and ultimately reduces to a latent view representation

Latent View Representation: It represents the lowest level space in which the inputs are reduced and information preserved.

Decoding architecture: It is the mirror image of the encoder but in which number of nodes in every layer increases and ultimately outputs the similar (almost) input.



Encoders $z = f(Wx + b)$

Decoders $\hat{x} = f'(W'z + b)$

Loss $\mathcal{L}(x, \hat{x}) = |x - \hat{x}|$

Use cases

- Image Reconstruction
- Image Enhancement
- Image Compression
- Image Denoising
- Feature Extraction
- Binary Classification

References

- https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
- https://medium.com/analytics-vidhya/mathematical-prerequisites-for-understanding-autoencoders-and-variational-autoencoders-vaes-8f854025390e