6) Adadelta and Rms prop

$$\eta' = \frac{\eta}{\sqrt{s_{dw} + \epsilon}}$$

$$s_{dw_t} = \beta s_{dw_{t-1}} + (1-\beta)\left(\frac{\partial L}{\partial w_{t-1}}\right)^2$$

Increase by smaller value.

7) Adam optimizer

Momentum + Rmsprop

$$w_t = w_{t-1} - \eta' v_{dw}$$

$$\eta' = \frac{\eta}{\sqrt{s_{dw} + \epsilon}}$$

$$v_{dw_t} = \beta * v_{dw_{t-1}} + (1-\beta)\frac{\partial L}{\partial w_{t-1}}$$

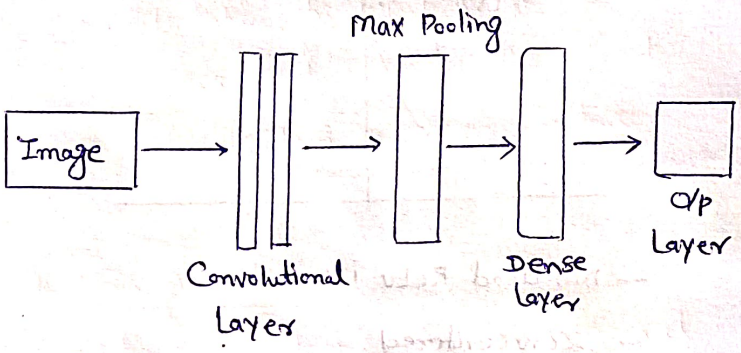
⊕: 1) Smoothening

2) Learning Rate becomes Adaptive

* CNN (Convolutional Neural Network)

↳ It is a deep Learning models designed to process data with a grid-like topology such as Images.

→ CNN Consist of multiple layers like Input Layer, Convolution Layer, Pooling Layers, Dense Layer.



* Convolution

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

*

1	0	-1
2	0	-2
1	0	-1

⇒ Verticle Edge filter

⇓

0	-4	-4	0
0	-4	-4	0
0	-4	-4	0
0	-4	-4	0

→ If Image size = n and filter size = f
then output size = n-f+1

→ This reduce the Image size, that lost some data.
↓
So We add Padding in the Image.

1	2	3
4	5	6
7	8	9

→

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

↳ Zero Padding

output size = $\underline{n + 2p - f + 1}$

- Activation Layer -
  - → Applied after the Convolution
  - → Activation Layer add non linearity to the network.
  - → It will apply an element wise activation function to the output of the Convolution Layer.
  - → Some Common Activation fn -
    - RELU : max (0, x)
    - Tanh, Leaky RELU

- Pooling Layer :-
  - → This Layer reduces the spatial size which makes Computation fast, reduces memory and also prevents overfitting.

→ Types :-

1) Max pooling : picks the max value

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2 ↓

6	8
3	4

2) Average pooling : Takes the Average of all Value.

→ only height and Width shrink, Channels stay Same.

Input : 28 (W) X 28 (h) X 32 (↳ Channels (After Applying Convolution))

maxPool (2x2) stride = 2 ↓

output : 14 X 14 X 32

Note:

Convolution output size -

H_1 : Input Height

W_1 : Input Width

C_1 : Input Channels

Convolution With -

F : Filter / Kernel size (FxF)

P : Padding

S : Stride

K : Number of filters

Output Height $(H_2) = \left\lfloor \frac{H_1 - F + 2P}{S} \right\rfloor + 1$

Output Width $(W_2) = \left\lfloor \frac{W_1 - F + 2P}{S} \right\rfloor + 1$

Output Channels $(C_2) = K$

Exa- Image - 6X6

Filter - 3X3

$P = 0$

$S = 2$

Output Height $= \left\lfloor \frac{6 - 3 + 0}{2} \right\rfloor + 1$

$= 1 + 1$

$= \underline{2}$

- Dropout Layer:-

→ This is a Regularization Layer, that prevents model to overfit.

→ This Layer Randomly turns off some Neurons. (e.g. - 30%)

→ ⊕ : 1) prevents overfitting

2) Focus Network to Learn more Robust features

- Flatten Layer:-

→ The Resulting feature maps are flattened into 1D vector after Convolution or pooling layers so they Can passed into a Completely linked layer for Categorization or regression.

- Fully Connected Layer (Dense Layer):-

→ It takes the input from the previous Layer and Computes the final classification or regression tasks.

* VGG Architecture:-

→ VGG 16 is deep Convolutional neural network designed for Image Classification tasks.

→ VGG 16 Consists 16 Layers, including 13 Convolutional Layers and 3 fully Connected Layers. These Layers are organized into blocks.

Block 1
- Conv, 3X3, 64 filters
- Conv 3X3, 64 filters
- max Pool 2X2

Block 2
- Conv 3X3, 128 filters
- Conv 3X3, 128 filters
- max Pool 2X2

Block 3

- Conv 3X3, 256 filters
- Conv 3X3, 256 filters
- Conv 3X3, 256 filters
- max Pool, 2X2

Block 4

- Conv 3X3, 512 filters
- Conv 3X3, 512 filters
- Conv 3X3, 512 filters
- maxPool 2X2

Block 5

- Conv 3X3, 512 filters
- Conv 3X3, 512 filters
- Conv 3X3, 512 filters
- max Pool 2X2

↓

flatten

↓

Dense Layer