$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$

Page No.
Date

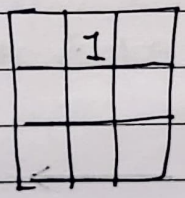
Unit 2 :- CNN

i) Alexnet
ii) Resnet
iii) LeNet
iv) VGGNet
v) RNN
vi) Transfer learning

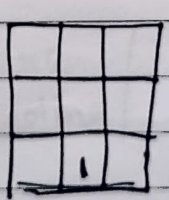
original matrix * ~~fi~~filter → new grid
grid

feature map.

9 * loopy patter detector =



6 * -11- =



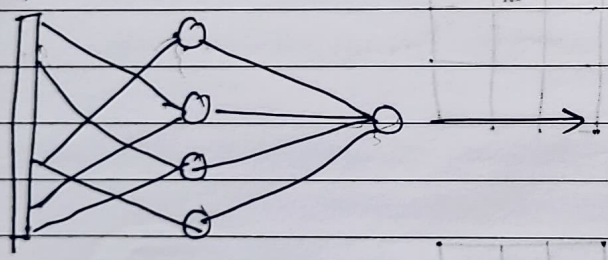
Relu → makes model non-linear.

Pooling layer → Reduce dimension of i/p image.

Benefits → i) Reduce. dimensions & comp..
ii) Reduce overfitting
iii) model is tolerant towards variations & distortions

i/p image → Convolutional + ReLU → Pooling → Convolutional + ReLU → Pooling

Feature extraction.



classification

Page No.
Date

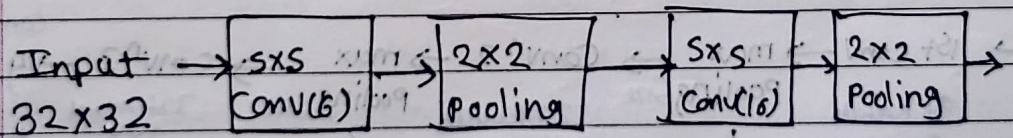
Convolution → • Connection sparcity reduces overfitting
• Conv + pooling gives location invarient feature detetion
• Parameter sharing

Relu → • Introduces non-linearity.
• Speeds up training, faster to compute.

Pooling → • Reduces dimension & computation
• Reduces overfitting
• Makes the model tolerant towards small distortion & variations

* Alex Net & LeNet

* LeNet



Page No.
Date

* Padding

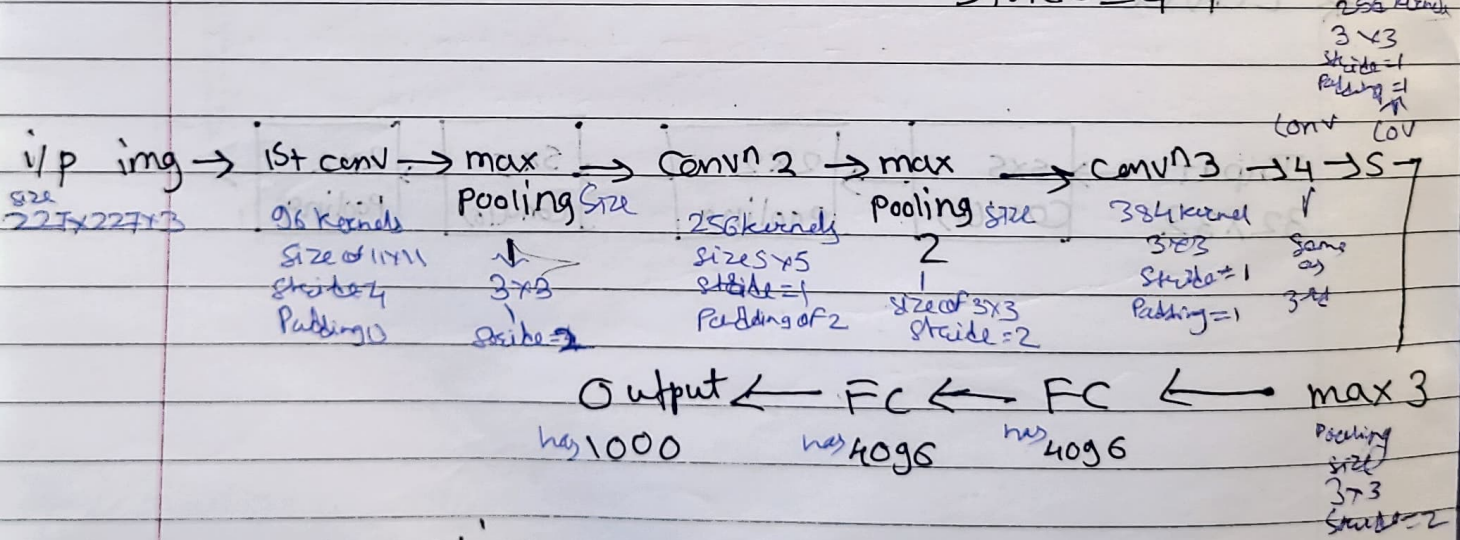
$(n+2p-f)+1 \rightarrow$ output size.

When we apply filter in infos in corner is lost. To overcome this we use zero padding

f → filter size.

* AlexNet

- winner of 2012 Imagenet competition
- Consists of 8 layer
- 5 are convolution layer
- 3 fully connected layers
- I/p size → 227 x 227 x 3
- 1st convolution layer → 96 filters, 11x11 stride=4



- max pooling → 3x3 with paddin 2

~~form~~

$$\text{o/p size} = \left(\frac{n+2p-f}{s}\right) + 1$$

- Relu is applied after each convolutional & fully connected layer except last.
- Droupout is applied before 1st & 2nd fully connected layer.

* Transfer Learning

Page No.
Date

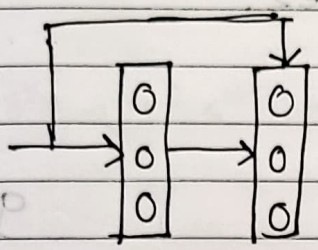
* Res-Net

- Deep networks are hard to train because of vanishing gradient problem
- ~~An~~ on Image Net dataset evaluate residual nets with depth of 152 layers - 8 deeper than VGG Net but still having lower complexity.
- An ensemble of thiese residual nets achieve 3.57% error on the ImageNet test set. This result won 1st place on ILSVRC 2015 classificatio task.
- Resnet 50 is a variant of Resnet model which has 48 conv. layers along with 1 max pool & 1 average pool layer..
- It has 3.8×10^9 floating point operations

{ vanishing gradient problem - As more layers using certain activation functions are added to nueral networks, the graiendts of the loss function approches zero, making the network hard to train }

- Resedual networks use the concept of skip connection.

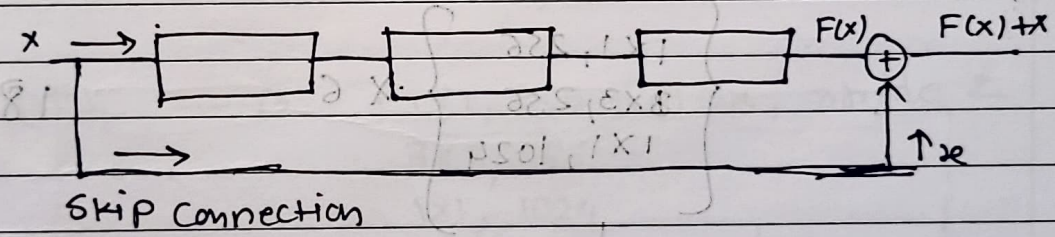
* skip connection



original i/p is added to the o/p of convolutional block.

Since, some of the connections are skipped, the value of the loss function gradients wont become zero.

* Residual block



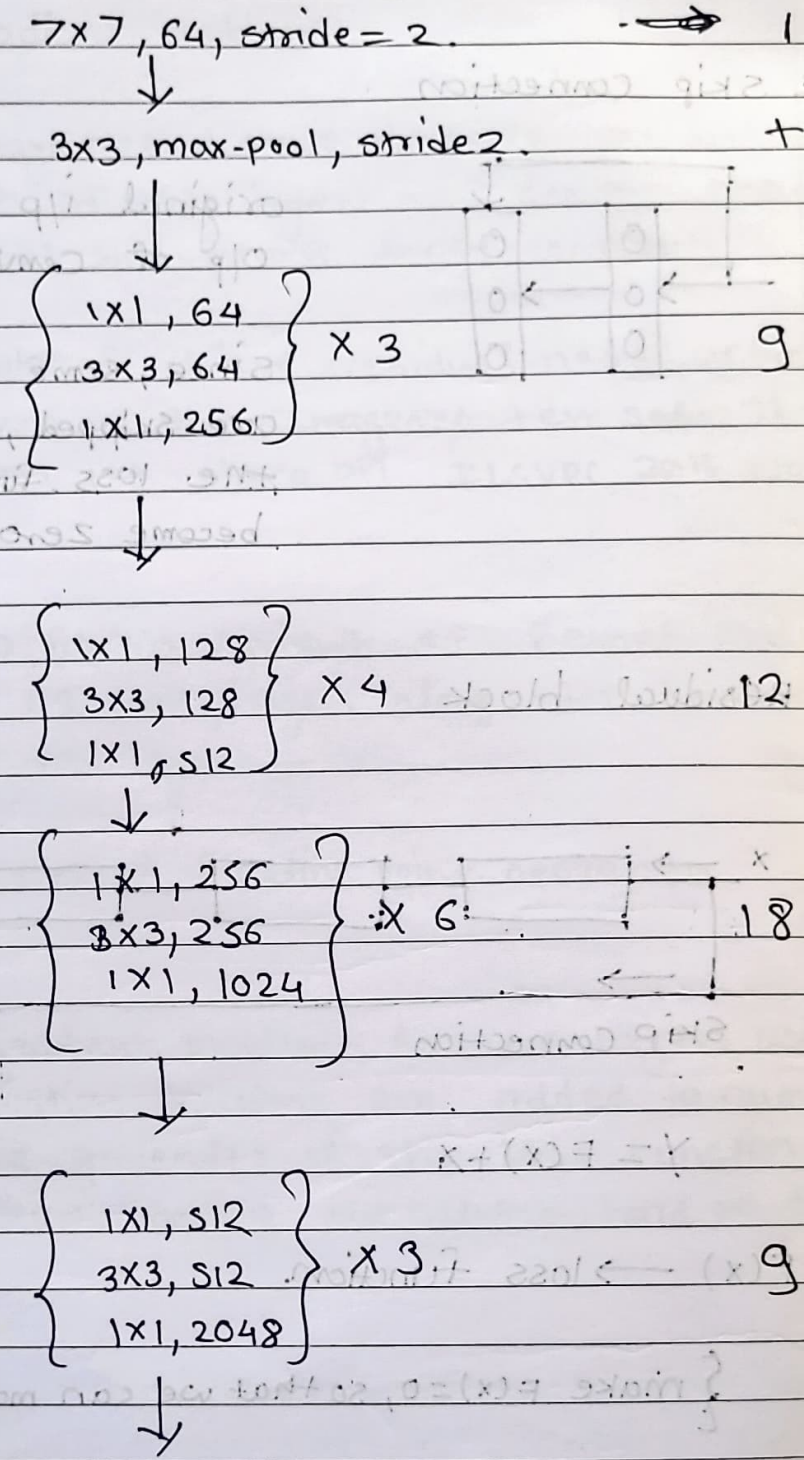
$Y = F(x) + x$

$F(x) \longrightarrow$ loss function.

{make $F(x) = 0$, so that we can make $Y = x$}

Page No.
Date

* Resnet -50



Page No.
Date

1st layer → 7×7, 64 filters with stride 2

2nd layer → maxpool layer 3×3 with stride 2

3rd, 4th & 6th → 1×1, 64 filters
3×3, 64 filters
1×1, 256 filters

7th → 1×1, 128 filters, stride 2
3×3, 128 filters
1×1, 512 filters

8th, 9th, 10th → 1×1, 128
3×3, 128
1×1, 512

11th → 1×1, 256 filters, stride 2
3×3, 256
1×1, 1024

12th, 13th,
14th, 15th & 16th → 1×1, 256
3×3, 256
1×1, 1024

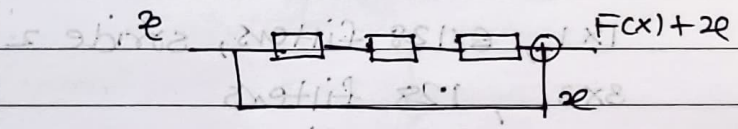
17th → 1×1, 512 with stride 2
3×3, 512
1×1, 2048

18th & 19th → 1×1, 512
3×3, 512 1×1, 2048

20th ⟶ Average pool with strides.

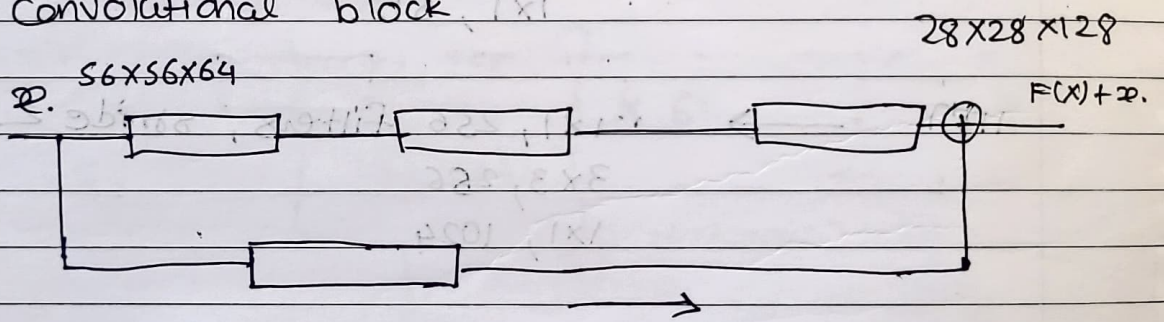
fully connected layer(fc),
1000

* Identity block



i/p size = o/p size

* Convolutional block



i/p size ≠ o/p size.

we can add x only if size of F(x) = x. but when x ≠ F(x) we add a convolution layer in shortcut path so that, F(x) = x

we use 1×1 convolution to make size of F(x) is equal to x.

$$\left(\frac{n+2p-f}{s}\right)+1$$

$$\left(\frac{56+2\times0-1}{2}\right)+1 \longrightarrow 28$$

* Hyper parameters

- Batch Normalization is used after each conv. layer.
- Xavier initialization with SGD+momentum is used.
- The learning rate is 0.1 & is divided by 10 as validation error becomes constant
- more over, batch size is 256 & weight decay is 1e-5
- There is no droupat in ResNet.

Page No.
Date

* VGG 16

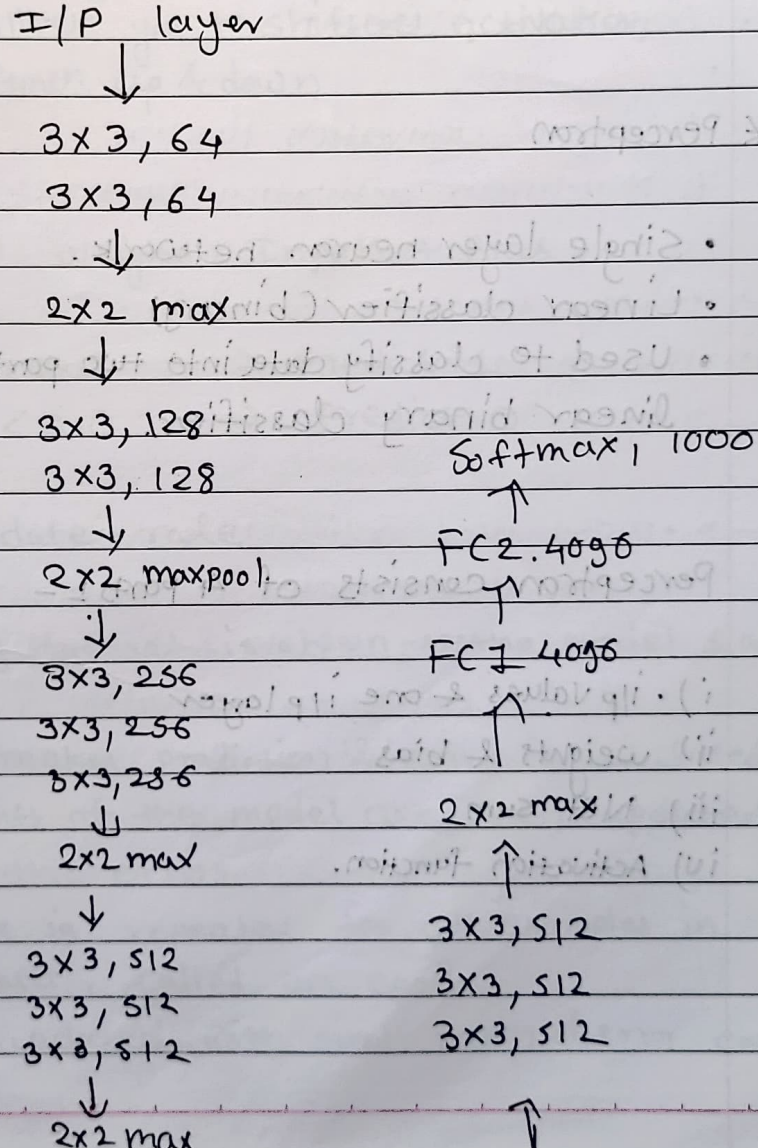
- Introduced by simonyan & zisserman in 2014.
- VGG Net was runner up of the Imagenet visual recognition (ILSVRC) classification in 2014.
- VGG stands for visual geometry group.
- VGG 16 has 16 conv. layers & VGG 19 has 19 conv. layers.
- The VGG architecture is the basis of ground braking object recognition models.
- VGG Net 16 achieves almost 92.7% top-5 test accuracy in ImageNet.
- It replaces large kernel size filters. with several 3x3 kernel sized filters.
- VGG Net-16 supports 16 layers & can classify images into 1000 object categories.

* Architecture

i) VGG 16

- 13 convolutional layers + 3 fully connected layers.
- Input size → 224 x 224
- VGG's conv. layers leverage the minimal receptive field. i.e. 3x3, the smallest possible size that still captures up/down & left/right.

- There are also ~~1x1~~ 1×1 conv. layers acting as linear transformation i/p.
- This followed by ReLU. ReLU reduces training time.
- All hidden layers in VGG uses ReLU.
- VGG doesnot usually leverages Local Response Normalization. (LRN) as it increases memory consumption & training time.
- Out of the 3 fully connected layers 2 have 4096 nuerons & last one has 1000.



* VGGNet Vs AlexNet
* VGGNet Vs ResNet

* Unit 1! - Deep learning basics.

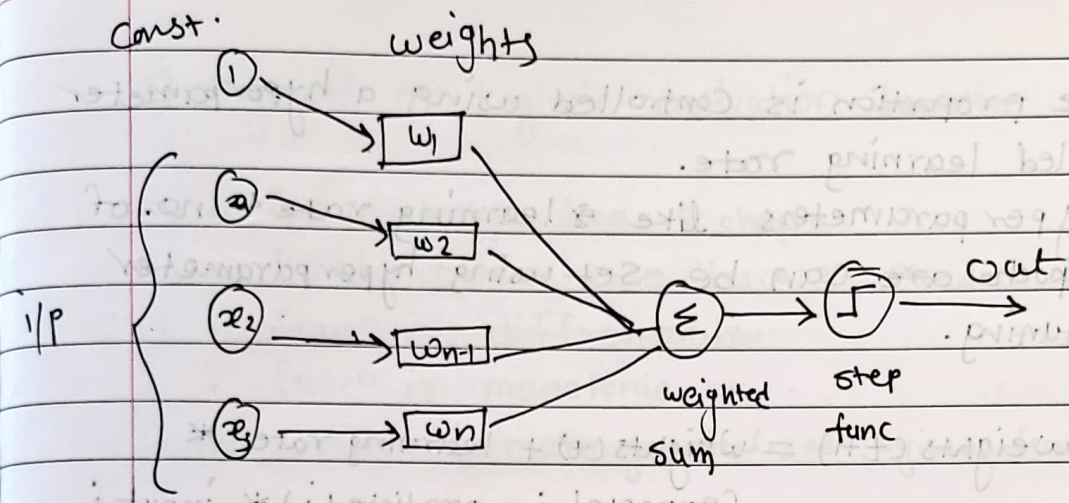
* To generalize if a mxm image is convolved with nxn kernel. The o/p image size is $(m-n)+1) \times (m-n)+1)$

* Perceptron

- Single layer neuron network.
- Linear classifier (binary)
- Used to classify data into two parts, hence known as linear binary classifier.

Perceptron consists of 4 parts :-

i) . i/p values & one i/p layer
ii) weights & bias
iii) Net sum
iv) Activation function.



weights → strength of particular node
bias → Allows you to shift activation fcnch up & down

Activation → weights x Inputs + Bias

if activation > 0.0 → predict 1
if activation < 0.0 → predict 0

* Perceptron update rule

- The training dataset is shown to the model 1 at a time.
- the model makes prediction & error is calculated.
- The weights of the model are then updated to reduce the errors.
- This process is repeated for all examples in training dataset, called an epoch.
- model is updated with small portion of error each batch.

Page No.
Date

- The proportion is controlled using a hyperparameter called learning rate.
- Hyperparameters like learning rate & no. of epochs are can be set using hyperparameter tuning.

$$weights(t+1) = weights(t) + learning\ rate * (expected\_i - predicted\_i) * input\ i$$

* Activation functions

i) Linear activation function
ii) Non-linear activation function.

i) → Linear function is a line or liner func^n. Therefore the o/p of func^n will not be confined between any ranges.

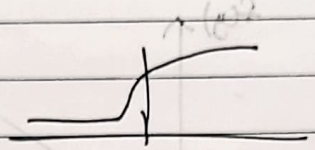
ii) → • most used activation functions
- easy for the model to generalize or adapt with variety of data & diffrentiate between o/p.
- The non-linear activation functions are mainly divided on the basis of range or curves

Page No.
Date

1) Sigmoid or Logistic activation function

- Curve looks like S shape
- Used where we have to predict probability
- funcn is diffrenciable
- funcn is monotonic.
- Softmax is more generalized logistic funcn.

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



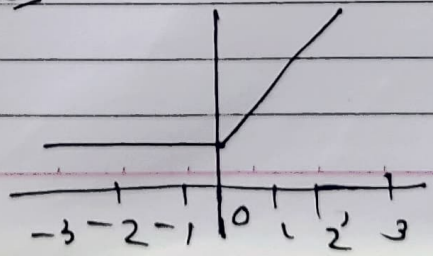
2) Tanh or hyperbolic tangent activation function.

- tanh is also like logistic sigmoid but better. The range of the tanh funcn is from (-1 to 1).
- The advantage is negative i/p will be ~~marked~~ mapped strongly ~~to~~ negative & zero i/p will be mapped near zero in tanh graph.
- The funcn is diffrentiable.
- The funcn

3) ReLU

- most used activation funcn monotonic
- Half rectified from bottom
- Range → [0 to ∞]

$$F(z) = \max(0, z)$$



Page No.
Date

4) Leaky ReLU

- The leak helps to increase the rang of ReLU, usually, value of $a \rightarrow 0.01$
- when a is not 0.01 then it is randomized ReLU
- Range $\rightarrow (-\infty, \infty)$
- monotonic in nature & thier derivatives too