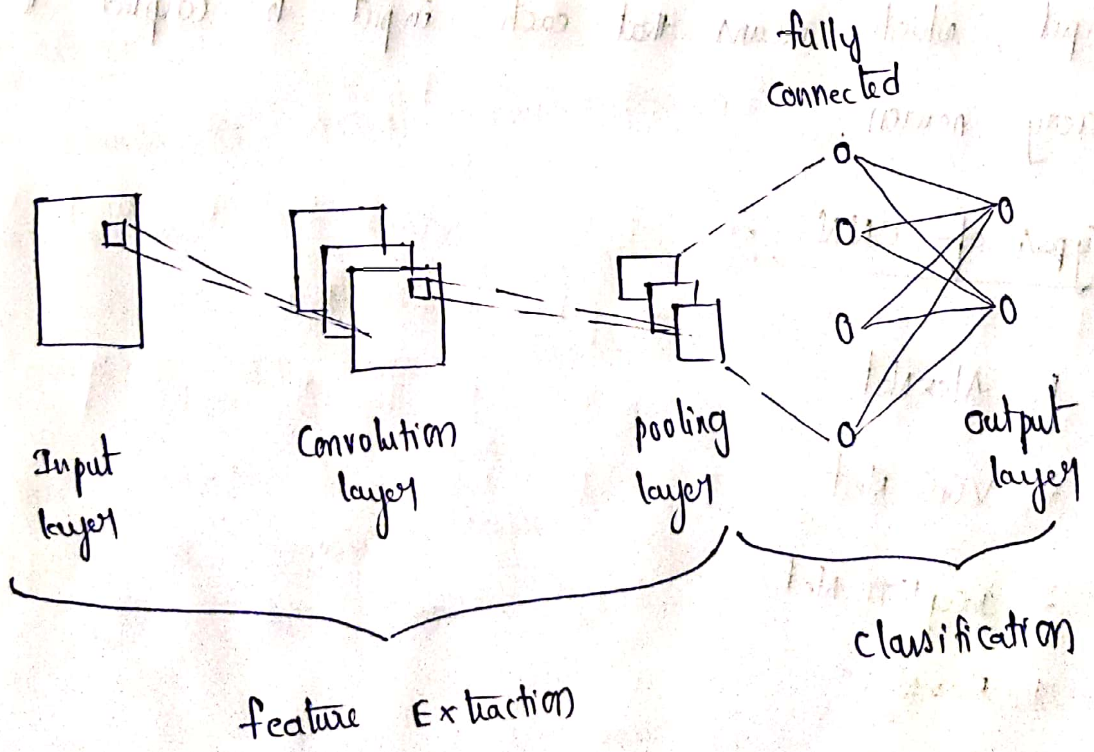
CNN - Convolutional Neural Networks

- CNN are a type of multi-layer neural network that is meant to discern visual patterns from pixel images.
- In CNN convolution is referred to as the mathematical function.
- CNN cannot function without convolutional layers.
- A convolutional neural network is made up of numerous layers, such as
  1. pooling layers
  2. convolution layers
  3. fully connected layers
  4. Input output layers.

CNN Architecture



Convolutional Layer :

- They are the foundation of CNN, and they are in charge of executing convolution operations.
- The kernel is the component in this layer that performs the convolution operation (matrix).

pooling Layer :

- This layer is in charge of reducing dimensionality.
- pooling layer is divided into 2 types
  1. Maximum pooling
  2. Average pooling

Fully Connected layer :

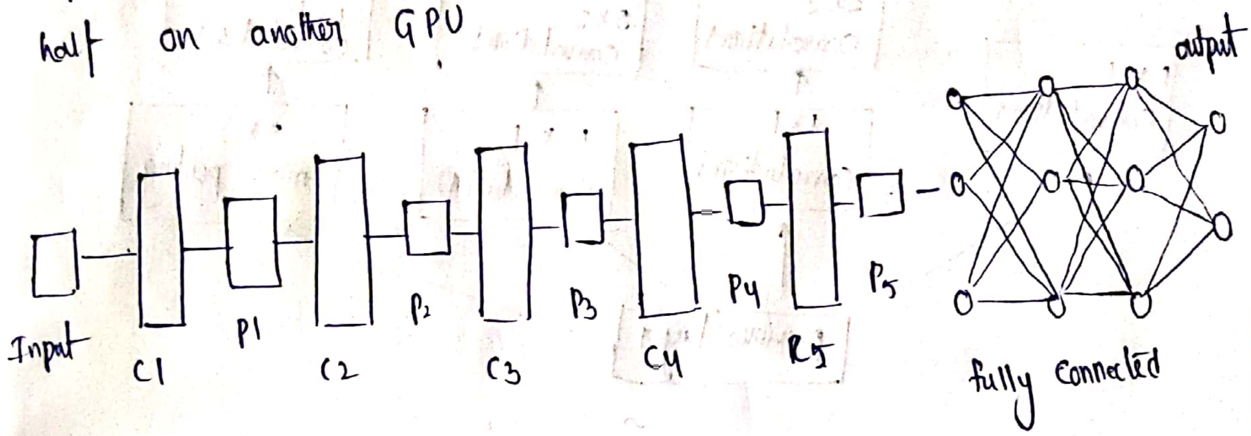
- The fully connected layer (FC) works with a flattened input, which means that each input is coupled to every neuron.

Types of CNN :

1. AlexNet
2. VGG Net
3. Inception Net
4. Res Net

1. Alex Net :

- Alex Net has 8 layers with learnable parameters
- this model have 5 Convolution layers and 3 fully conneted layers.
- AlexNet allows for multi - GPU training by putting half of the model's neurons on one GPU and the other half on another GPU



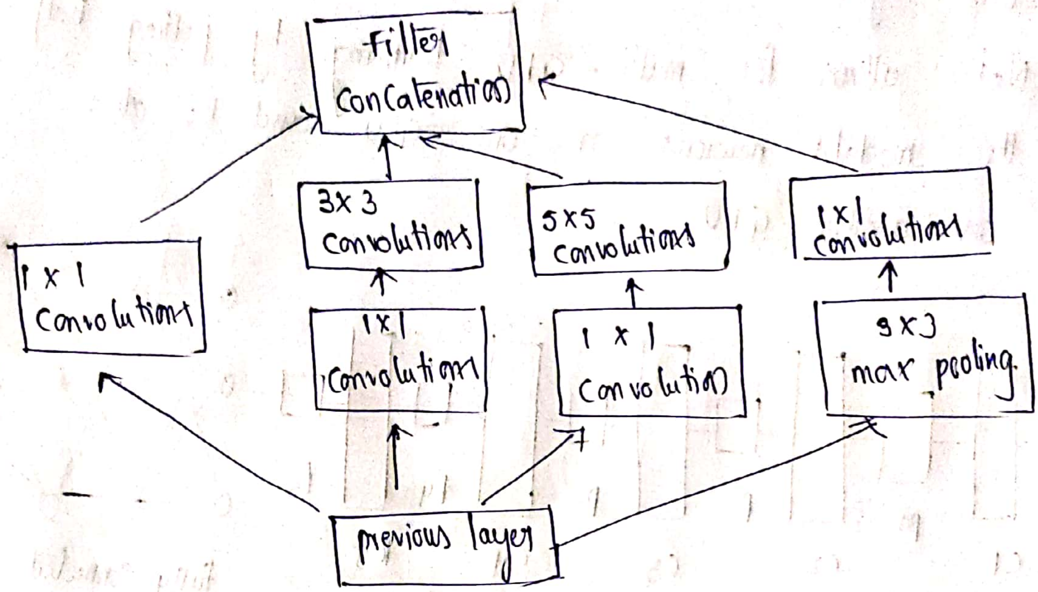
c - Convolution layers
p - pooling layers.

2. VGG Net :

- it stands for visual geometric graph
- VGG 16 has 16 layers,
  - i) 13 convolution layers
  - ii) 3 fully Connected layers
- VGG 19 has 19 layers
  - i) 16 convolution layers
  - ii) 3 fully Connected layers.
- The VGG architecture is the basis of ground - breaking object recognition models.

3. Inception Net :

* Inception Net also known as GoogleNet.
* Inception Net uses the Inception module



4. ResNet :

* ResNet has 34 layers
  - i) 31 Convolution layers
  - ii) 3 fully connected layers

=> ResNet is a powerful backbone model that is used very frequently in many computer vision tasks.

-> ResNet uses skip connection to add the output from an earlier layer to a later layer.

Dimentionality Reduction

——— ** imp

* this technique Reduce no. of input variables in a dataset
* Higher no. of features is harder to visualize the training set and to work in that
* features can be corelated and hence redundent In such case we need DR (Dimentionality Reduction)
* DR is a process of reducing no. of random variables under corelation by updating a set of principle variables
* DR can be divided into 2 types
    1. feature selection
    2. feature extraction

1. feature selection

* it is the process of selecting the subset of the relavant feature and leaving out the irrelavant feature present in the dataset to build a model of high accuracy
* it is the way of selecting optimal features from input dataset

Methods of feature selection

1. filter
2. Wrapper
3. Embedded

1. Filter method

* Data set filtered with only relevant features
2. 1. corelation
   2. chi square set
   3. ANOVA
   4. Information gate

2. Wrapper method

* filtered relevant feature using machine learning techinique.
  1. forward selection
  2. Backward selection
  3. Bidirectional elimination

3. Embedded

* It checks different training situation of ML model and ~~eva~~ Evaluate important of each feature
  1. Lasso
  2. Elastic net
  3. Rich reggression

2. Feature Extraction

→ it is the process of transforming the space containing many dimension into space with fewer dimension

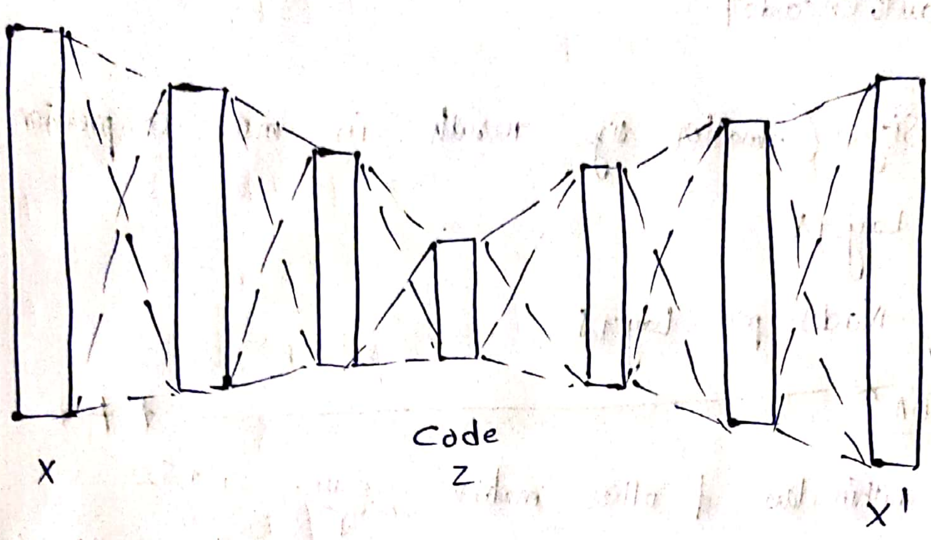
1. PCA
2. LDA (Linear ~~Dimenti~~ discriminate Analysis)
3. QDA

Applications

1. Speech recognization
2. Signal processing
3. Bio Informatic

Autoencoder

→ it is an unsupervised ANN that compresses the data to lower dimension and than reconstructs the input back



X - Input → encode

X' - output → Decode

Z - code / reconstruct

→ it finds the representation of data in a lower dimension by focusing more on the important features getting rid of noise and redundancy

→ Base on Encoder and Decoder task, encoder encodes the high big dimensional data to Low dimensional data and decoder takes the low dimensional data to reconstruct the high dimensional data

→ The mapping of higher to lower dimension can be lead a linear or non-linear depending upon the choice of activation function

Hyper parameter

- Hyperparameter that we need to set before training the autoencoder

1. Code Size (smaller size results in more compression)
2. No. of Layers
3. No. of Nodes per Layers

— * * problems

PCA → principle Component Analysis

→ * used to Reducing the Dimentionality / feature extension also

* Important point

* example : 100 variables are present (suppose) → it reduces the 100 variable into 50 variable

---

step 1 : Data set

step 2 : Computation of Mean of variable

step 3 : Computation of Covariance matrix.

step 4 : Eigen value, Eigen vector of Normalised Eigen vector

step 5 : New Dataset.

Problem 1 :

features	Eg 1	Eg 2	Eg 3	Eg 4
x	4	8	13	7
y	11	4	5	14

Solution:

Step 1 : Data set

No. of samples (N) = 4

No. of features (n) = 2

step 2 : computation of mean of variable

$$\bar{x} = \frac{4+8+13+7}{4}$$

$$\boxed{\bar{x} = 8}$$

$$\bar{y} = \frac{11+4+5+14}{4}$$

$$\boxed{\bar{y} = 8.5}$$

Step 3 : Covariance matrix (S)

ordered pair : (x,x) (x,y) (y,y) (y,x)

$$cov(x,x) = \frac{1}{N-1} \Sigma ((x_i - \bar{x})^2$$

$$= \frac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2 \right)$$

$$= \frac{1}{3}(42)$$

$$\boxed{cov(x,x) = 14}$$

$$cov(y,y) = \frac{1}{N-1} \Sigma (y_i - \bar{y})^2$$

$$= \frac{1}{3}\left[(11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2 \right]$$

$$= \frac{1}{3}(69)$$

$$\boxed{cov(y,y) = 23}$$

$$cov(x,y) = \frac{1}{N-1}\left(\Sigma((x_i - \bar{x}) * (y_i - \bar{y})\right)$$

$$= \frac{1}{3}\left((4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5) \right)$$

$$= \frac{-33}{3}$$

$$= -11$$

$$\boxed{cov(x,y) = cov(y,x) = -11}$$

Covariance matrix

$$S = \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix}$$

$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

<u>step 4:</u> Eigen value, Eigen vector, Normalized Eign vector

<u>i. Eigen value</u> :

$$\det(S - \lambda I) = 0$$

$n = 2 \quad n \times n =$

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\det\left(\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = 0$$

$$\det\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} = 0$$

$= 14(23) - 14\lambda - 23\lambda + \lambda^2 - 121$

$= \lambda^2 - 37\lambda + 322 - 121$

$= \lambda^2 - 37\lambda + 201$

$$(14-\lambda)(23-\lambda) - (11 \times 11) = 0$$

$$\lambda^2 - 37\lambda + 201 = 0$$

$$= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= \frac{37 \pm \sqrt{37^2 - 4(1)(201)}}{2}$$

$$= \frac{37 \pm \cancel{2\sqrt{4}}\sqrt{1369 - 804}}{2}$$

$\frac{1}{2}$

$$= \frac{37 \pm \sqrt{565}}{2}$$

$$= \frac{37 \pm 23.76}{2}$$

$$= \frac{37 + 23.76}{2} \quad (or) \quad \frac{37 - 23.76}{2}$$

$$= \frac{60.76}{2} \quad (or) \quad \frac{13.24}{2}$$

$\lambda_1 = 30.3849 \qquad \lambda_2 = 6.6151$

take always largest value $\rightarrow \lambda_j$

ii) Eigen vector

$$(S - \lambda I)\, U = 0 \qquad \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

$$\begin{bmatrix} 14 - \lambda & -11 \\ -11 & 23 - \lambda \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$(14 - \lambda) U_1 - 11 U_2 = 0$ —— ①

$-11 U_1 + (23 - \lambda) U_2 = 0$ —— ②

Eqn ① $\quad 14 U_1 - \lambda U_1 - 11 U_2 = 0$

Eqn ② $\quad -11 U_1 + 23 U_2 - \lambda U_2 = 0$

$11 U_2 = (14 - \lambda) U_1$

$\frac{U_1}{11} = \frac{U_2}{14 - \lambda}$ —— ①

$11 U_1 = (23 - \lambda) U_2$

$\frac{U_1}{23 - \lambda} = \frac{U_2}{11}$ —— ②

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda} = t$$

we know that $\boxed{t = 1}$ ⟶ Assume

$$\frac{u_1}{11} = 1$$

$$u_1 = 11$$

$$\frac{u_2}{14-\lambda} = \cancel{\frac{11}{11}}$$

$$\boxed{u_2 = 14-\lambda}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix} = \begin{bmatrix} 11 \\ 14 - 30.3849 \end{bmatrix} = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

3. Normalized Eigenvector

$$\begin{bmatrix} \frac{u_1}{\sqrt{(u_1^2)^2 + u_2^2}} \\ \frac{u_2}{\sqrt{u_1^2 + u_2^2}} \end{bmatrix} = \begin{bmatrix} \frac{11}{\sqrt{11^2 + (-16.384)^2}} \\ \frac{-16.384}{\sqrt{11^2 + (-16.384)^2}} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} = e$$

step 5 : New dataset :

	P_{11}	P_{12}	P_{13}	P_{14}
x	4	8	13	7
y	11	4	5	14

$$P_{11} = e_1^T \begin{bmatrix} x_1 - \bar{x} \\ y_1 - \bar{y} \end{bmatrix}$$

$$P_{11} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$$= -2.2296 - 2.07575$$

$$= -4.30535$$

Scanned with CamScanner

$$P_{12} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 0 \\ -4.5 \end{bmatrix}$$

$$= 3.73635$$

$$P_{13} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 5 \\ -3.5 \end{bmatrix}$$

$$= 2.787 + 2.90605$$

$$= 5.69305$$

$$P_{14} = \begin{bmatrix} 0.5574 & \cancel{-0} \; -0.8303 \end{bmatrix} \begin{bmatrix} -1 \\ 5.5 \end{bmatrix}$$

$$= -0.5574 - 4.56665$$

$$= -5.12405$$