Convolutional Neural Networks

Deep Learning + Tensor Flow

- Deep learning uses neural networks with hierarchy of layers b/w input & output.

- Shallow learning → Algo are given some knowledge like which features are important
  e.g., SVM, KNN, Decision trees, Bayes, Perceptron

- Deep learning → Khud raw data sey complex patterns nikalta hy.
  e.g., deep Convolutional networks, embeddings, deep autoencondes.

  - Architectures → CDNN
    RNN (LSTM)
    embeddings (Word2Vec)

- Convolution deep networks (convNets) (inspired by animals visual system)
  - Helps in object recognition.

- LeNet (orignally created for hand written characters recognization)

  - When lette given, LeNet divides it in categories, Probability to every category.
  - Sum of probabilities is always 1.
  - Category having most probabilty is chosen as output.

  ↓ reached through

  - Convolution (applies filter, imp features like edges, shapes)
  - Non-Linearity (ReLU) → ek activation function.
  - Pooling / Sub Sampling → image choti ki jati hy
  - classification.

[illegible]

Input $\rightarrow$ $\underset{\text{ReLU}}{\text{Convolution}}$ $\rightarrow$ Pooling $\rightarrow$ $\underset{\text{ReLU}}{\overset{}{\text{Convolution}}}$ + $\rightarrow$ Pooling $\rightarrow$ Classification.

- Input of LeNet is an image. Image is in matrix of pixels.
- Grey scale images have one Channel only (0 – black, 2~~5~~55 – White)

- Convolution (used in filtering of images)
  - Modify spatial frequency or extra/unwanted information filtered.
  - Central pixel found by – 4 or 8 connectivity.
  - Based on kernels used, different features can be extracted.

- Kernels (known as filters or feature detectors)
  $\Downarrow$
  Values learned during training phase.

  more kernels used => more feature extracted

•– Non-linearity – ReLU
- Rectified Linear Unit (ReLU) is applied by convolution
- • yeh function mein say har value ko zero kr daata hy.
- • $f(n) = \max(0, n)$

– Pooling (reduce size of data but keep imp features)
Can be done by computing (Max, Avg, Sum etc).

$$\begin{array}{|c|c|c|c|}\hline 4 & 3 & 7 & 4 \\ \hline 2 & 4 & 6 & 3 \\ \hline 2 & 3 & 1 & 4 \\ \hline 5 & 6 & 8 & 2 \\ \hline \end{array} \xrightarrow[\text{pooling.}]{\text{max}} \begin{array}{|c|c|}\hline 4 & 7 \\ \hline 6 & 8 \\ \hline \end{array}$$

- Classification. (Carried out in fully Connected layer)
  - A multi-layer ~~prep~~ perceptron is used.
    other Classifier like SVM can also be used.

- LeNet Overview:
  - Initilize Kernel & weights with random values.
  - Put image from training data to network.
    - Data pass through Convolution + pooling + RELU ---
    - We get output as output vectors probabilities.
  - Calculate total error at output layer.
  - Use back propagation.

- Training Phase output → A ConvNet which has optimized weights and kernel parameters.
  - For new unlabelled image → • put image in trained ConvNet
    - Image goes to output layer,
    - ~~label which gives more~~ probability, assign that to image.

- TensorFlow → (Use to make neural networks and train)
  - Tensors → main data elements (A multi-dimensional ~~array~~)

A vector = 1-D
Matrix = 2-D Tensor.

  - Tensor flows inside data graph & do complex numerical calculations.
  - Data flow are $\boxed{\text{nodes}}$, edges.
    - nodes ↓ ~~are~~ operations (addition, Convolution)
    - edges → receives tensors + outputs them.

[illegible]