

PPT-One $\Rightarrow$ fairly rubbish

PPT-Two $\Rightarrow$ Types of Dataset :-

1D Dataset $\rightarrow$ Tabular Data, csv files, MAT files.

2D Dataset $\rightarrow$ RGB Images.

3D Dataset $\rightarrow$ MRI scans, Hyperspectral Images.

PPT-Three $\Rightarrow$

* ML vs DL :- Machine learning works on human designed and extracted features.

Deep learning works on extracting features and then optimizing weights for the end goal.

* Why Deep learning $\Rightarrow$ Manually designed features are overspecified, incomplete and time taking to design.

learned features are fast to adapt and learn.

Provides universal learning framework for all kind data challenges.

large amount of data is required.

* Precision = % of correct positives, Recall = % of correct positives captured from all positives

Accuracy = % of correct predictions.

* SVMs $\Rightarrow$ Hyperplane = Decision boundary built around for feature classification.

Support Vectors = Data point closest to the hyperplane, decides the orientation and placement of the hyperplane.

Margin = Distance between plane and nearest data point.

Cutler, Kernel, regularization parameter

Quadratic function problem $\Rightarrow$ minimize $\frac{1}{2} ||w||^2$
subject to $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$ } Separable data

Non Separable Data $\Rightarrow$ minimize $\frac{1}{2} ||w||^2 + C \sum_{i=1}^n \max(0, 1 - y_i(w \cdot x_i + b))$
classification mistakes

* CNNs $\Rightarrow$ Neural Network and Image
Dimensions $\downarrow$
Local relations $\downarrow$

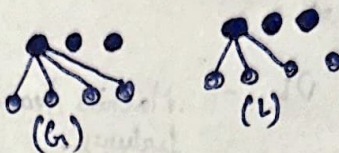
CNN $\Rightarrow$ Conv layer
Non-Linear layer
Pooling layer
Fully connected Layer
Classification layer

output
Input $\xrightarrow{\text{kernel}} \text{padding}$

$$O = \left[\frac{I - K + 2 \times P}{S} \right] + 1$$

$\hookrightarrow$ Stride

Global vs Local connectivity $\Rightarrow$
(Weight sharing)



$E \Rightarrow$

Input Image $[32 \times 32 \times 3] \xrightarrow{\text{Conv layer}} [32 \times 32 \times 3] \xrightarrow{1 \text{ filter}} [5 \times 5 \times 3] \xrightarrow{(K \text{ filters})} [28 \times 28 \times K]$

output of Conv 1

$[28 \times 28 \times K] \xrightarrow{\text{Conv layer}} [28 \times 28 \times K] \xrightarrow{1 \text{ filter}} [5 \times 5 \times K] \xrightarrow{(K_2 \text{ filters})} [24 \times 24 \times K_2]$

Out of Conv 2

Activation fcn $\Rightarrow \sigma(n) = \frac{1}{1 + e^{-n}}$ ReLU = $\max(0, n)$
 $\tanh(n)$ Leaky ReLU $\Rightarrow \max(0.1n, n)$

PPT-4 $\Rightarrow$

famous CNNs $\Rightarrow$ LeNet

AlexNet, ZFNet (7x7 & stride 2)

VGGNet $\Rightarrow$ Deeper Neural Network but smaller kernel size

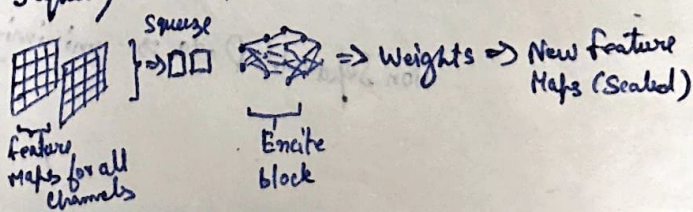
NIN $\Rightarrow$ MLP Conv Layer.

GoogleNet $\Rightarrow$ Introduction of Inception module

ResNet $\Rightarrow$ Residual Mapping, Batch Norm. after Conv.

Inception V4 $\Rightarrow$ ResNet + Inception.

SENET $\Rightarrow$ Squeeze and Excite Block.



PPT-5 $\Rightarrow$ Adversarial Attacks

White Box + Black Box

$\downarrow$
Access to model & weights

$\downarrow$
No access to model & weights



Input Image



Perturbation



Adversarial (new Input)

$\Rightarrow$ Model misclassifies this

Poisoning + Evasion Attack

* Generating Adversarial $\Rightarrow$

$$FGSM: \quad \eta \leftarrow \eta + \text{sgn} \left(\frac{\partial L(\eta, y^*)}{\partial \eta} \right) \times \epsilon$$

$\downarrow$ $\rightarrow$ loss $\rightarrow$ constant
 $\rightarrow$ derivative across features

$$f(\eta) = \eta \begin{cases} 1, & \eta > 0 \\ 0, & \eta = 0 \\ -1, & \eta < 0 \end{cases}$$

Iterative Gradient Sign Method :- Take multiple small steps and create a high confidence misclassified adversarial.

- (i) Create an adversarial
- (ii) Check direction of loss
- (iii) re-iterate on that desired direction (clipped values)

Least Likely :- This method is a targeted adversarial attack technique where the goal is to modify an input image that is classified as the class considered least probable.

Semantic Attack :- Negatives of image or individual pixel.

Deep Fool :- Deepfool treats the classifier as an linear classifier. At each step it finds smallest perturbation required to move the input across the nearest class boundary. updation with least minimal perturbation.

Universal Adv :-

- $r = 0$
- $\eta_i + r$ is misclassified skip else $\Rightarrow$ (amplifier)
- enforce $\|r\|$ is $\leq \epsilon \Rightarrow \left[r = \epsilon \cdot \frac{r}{\|r\|} \right]$
- update at each do for all.
- Stop at when reached fooling rate.

* Defending Attack :- Adversarial Training.
Reject Adv.
Robust Arch.
Image Pre-processing.

PPT-6 $\Rightarrow$ Regularization & Transfer Learning

* Regularization Techniques $\Rightarrow$ • L_1 , L_2 , $(L_1 + L_2)$ Regularization

$$\sum_i \|w_i\| \quad \sum_i w_i^2 \quad \sum_i B w_i^2 + \|w_i\|$$

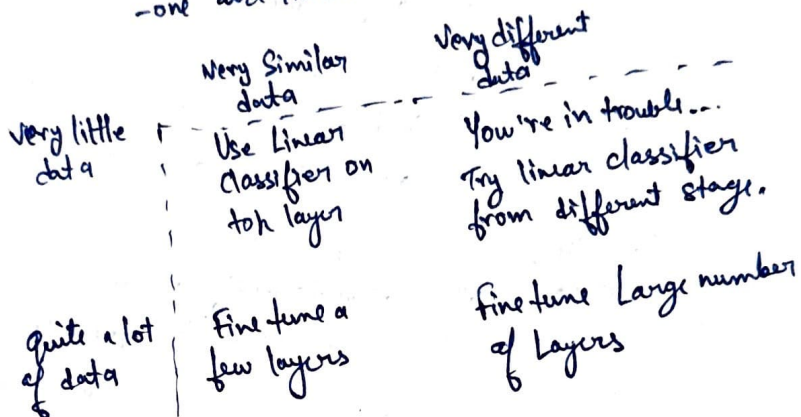
• Dropout

• Batch Normalization $\Rightarrow$ • Act as a form of regularization
• Improves gradient flow through network

• Data Augmentation • High learning rates.

× can't be used in RNNs.
× Depends on size of mini-batch

* Transfer Learning $\Rightarrow$ Using a very Model and using that model as backbone and trains or uses new classifier layer.



PPT-10 $\Rightarrow$ NLP - Natural Language Processing

Data Pre-Processing

Tokenization
Stop words Removal
Stemming
Lemmatizing

Modelling Techniques

Bag of Words
TF-IDF
Word Embeddings
Sentiment Analysis

• Tokenization $\Rightarrow$ Splitting text into smaller units called tokens. Tokens are usually words, sentences. Used for searching and other Modelling.

• Stemming $\Rightarrow$ Process of reducing words to its root form. By removing prefix suffix.

• Lemmatization $\Rightarrow$ Converting it to root form with grammatical accuracy.

• Bag of Words $\Rightarrow$ Vector representation of text on the basis of word count from a list of words in document.

• TF-IDF $\Rightarrow$

Term freq = Count of word in each doc

IDF = $\log \left(\frac{N}{N_0} \right)$ total doc
↳ no. of documents in doc