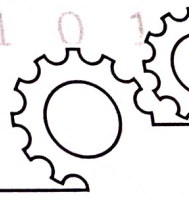
SIMATS
School of Engineering

Deep Learning

Computer Science and Engineering



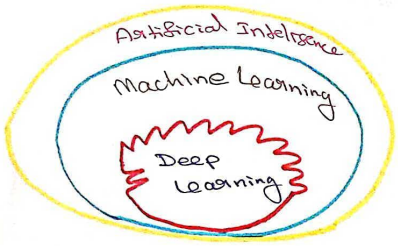
Saveetha Institute of Medical And Technical Sciences,Chennai.

INDEX

INTRODUCTION

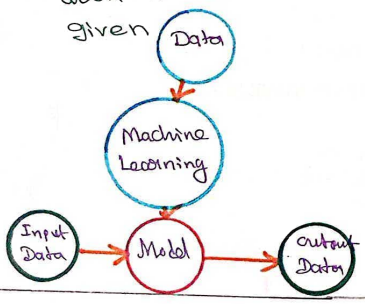
Machine Learning:-

* Machine learning is a subfield of Artificial Intelligence (AI).

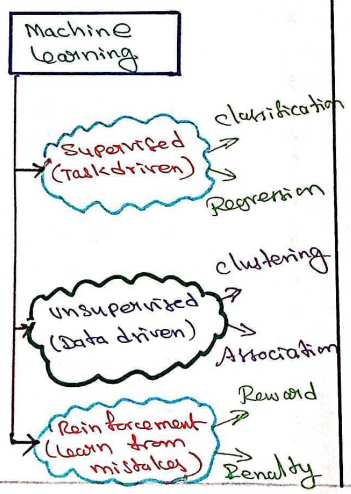


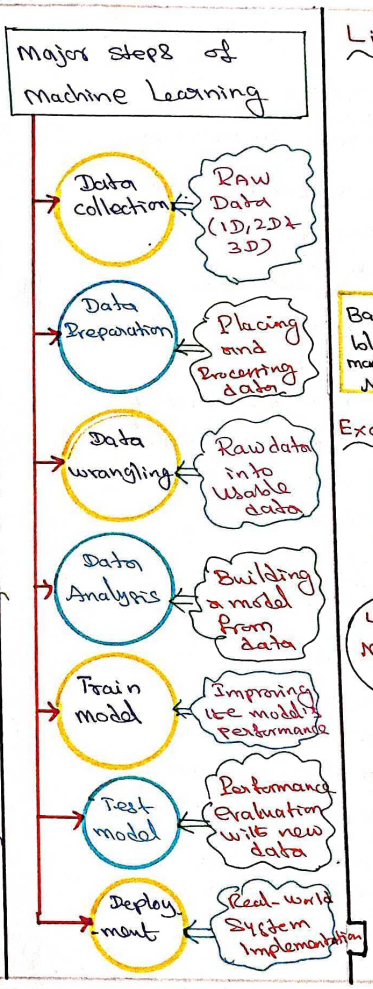
Machine Learning

- Data Assisted modelling scheme
- Learns from training Data
- Predicts result when a new data is given

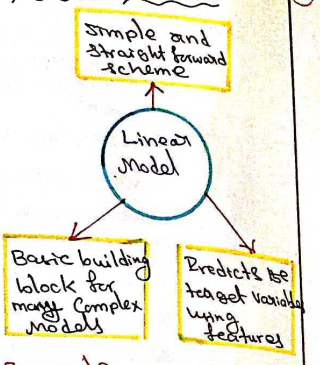


Types:-

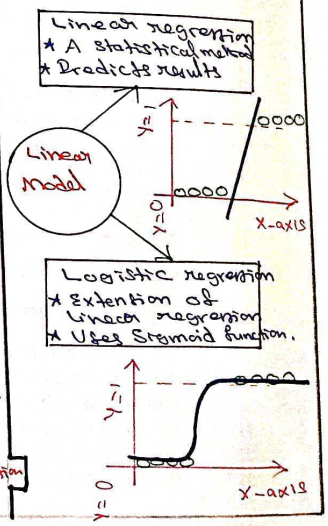




Linear Models:



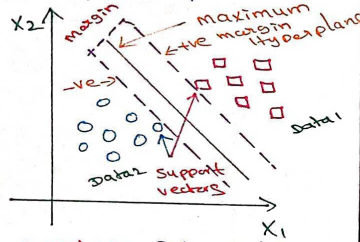
Example:



INTRODUCTION to Neural Network

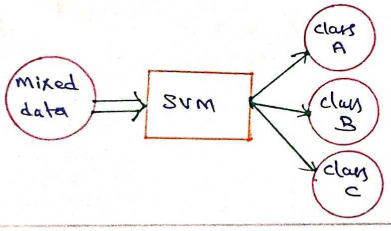
Support Vector Machine (SVM)

* A popular supervised learning algorithm
* Used for classification and regression problems.
* Creates a best line or decision boundary for data seperation



* Support vectors: Datapoints close to hyperplane
* Hyper plane: Decision space between set of objects
* Margin: Gap between two lines (i.e: +ve and -ve).

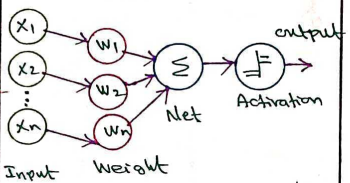
Classification:



Perceptron:

Perceptron is a building block of an Artificial Neural Network (ANN).

Components:



If: X_i → input
W_i → weights
b → bias.

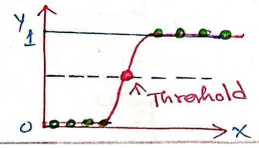
Then:

$$\sum X_i * W_i = X_1 * W_1 + X_2 * W_2 \cdots + X_n * W_n$$

output $Y = f\left(\sum X_i * W_i + b\right)$.

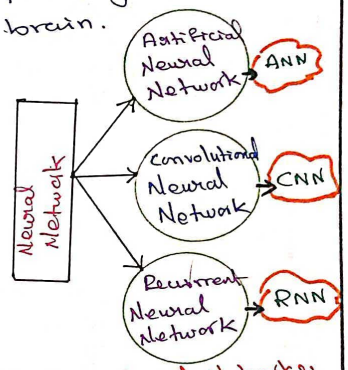
Logistic Regression:

* It predicts dependent variable from independent variable
* Provides probabilistic values between 0 and 1.
* Used in classification task

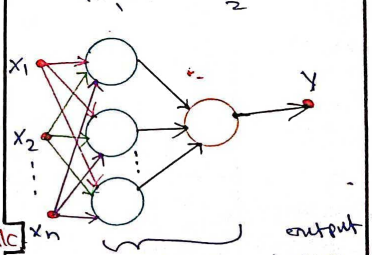


Neural Network:

It mimics the information processing by human brain.



Shallow Neural Networks:



* It has one (or) two hidden layers.

Training a Neural Network

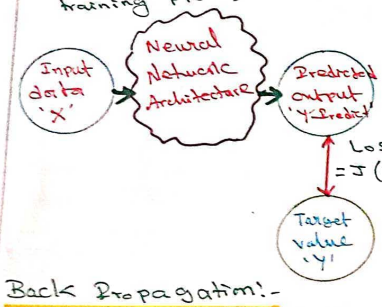
Neural Network training steps

- Construct the required neural network architecture
- Randomly initialize weights between 0 and 1.
- Implement forward Propagation algorithm
- Apply cost function to optimize parameters
- Compute error using back propagation algorithm
- Implement gradient checking for comparison
- Minimize the cost function to get better result

LEARNING ALGORITHMS

Loss function:

* Loss function compares the target and predicted values
* A prime value during the training process



Back Propagation:-

* It uses chain rule to compute gradient of the loss function
* It computes values in one layer at a time
* It generalizes the computation in delta rule.

Need:

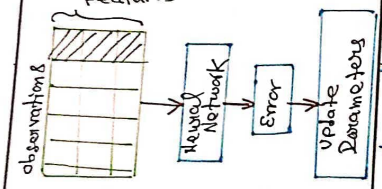
* Fast, Simple and easy
* No parameters to adjust
* Flexible and doesnot need prior knowledge about network
* Works well for different architectures

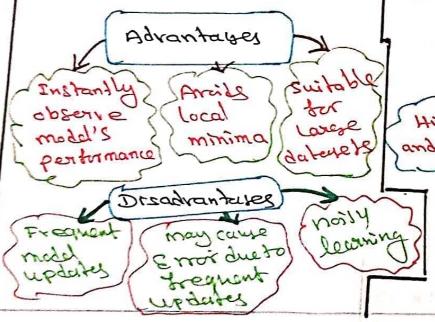
Stochastic Gradient Descent:

* It is an optimization algorithm
* It adjusts parameters to minimize particular function to local minima.

Features:

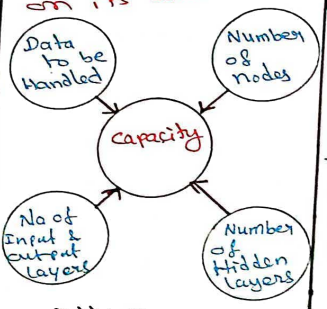
* Single observation for updation
* more variations in cost function
* more computation time





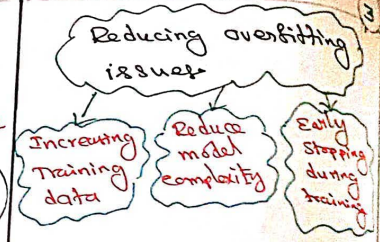
Capacity:-

* Capacity of an algorithm depends on its architecture



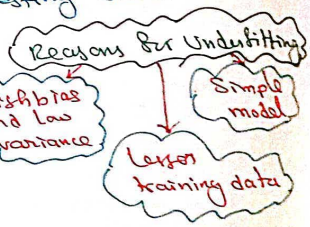
Overfitting:

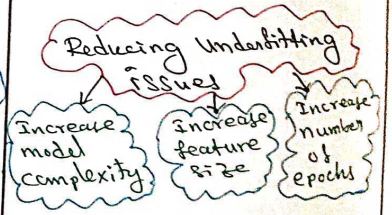
* Model learns the detail and noise in training data
* Happens in nonparametric and nonlinear models.



Underfitting:

* Underfitting is defined as the network performs well on training data but performs poorly on testing data.





Hyperparameter:

* Parameters, which are explicitly defined by the user to control the learning process.
* It also known as configuration

Hyperparameter tuning Steps

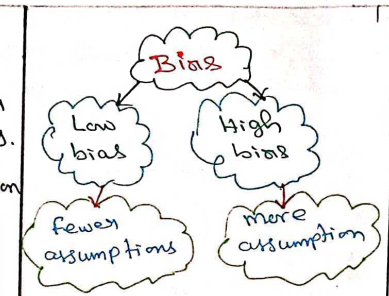
- Step 1: Model Selection
- Step 2: Review of parameters
- Step 3: Right search method
- Step 4: Cross validation & verification
- Step 5: Model Result verification

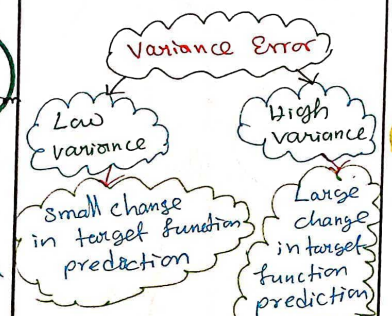
Validation Set:

* It is a part of the data used to verify the traning process
* The performance is assessed using Loss and Accuracy.

Bias and Variance:

Machine learning Errors
- Reducible
  - Bias → Actual v Predicted
  - Variance → Difference between Expected and random variable.
- Irreducible





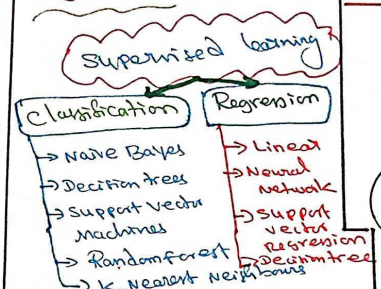
Bayesian Statistics:

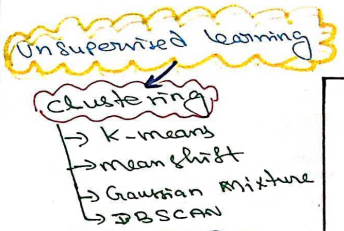
* It involves conditional probability

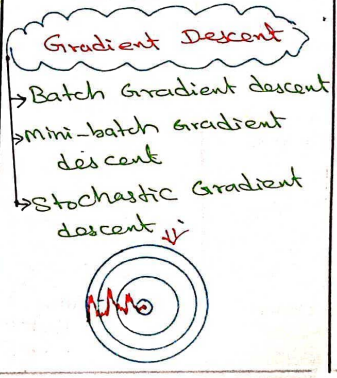
Bayesian statistics
→ Daye's Rule
→ Event A, Event B

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

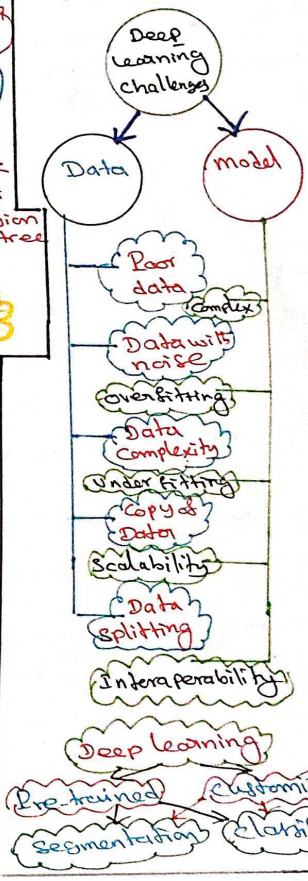
Supervised learning Algorithms







Challenges in Deep Learning:



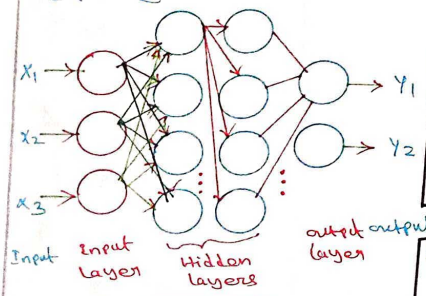
DEEP NETWORK

Deep Feed-Forward Network:

Feed-Forward Architecture

- Single input layer
- One or many Hidden layers
- Single output layer

* Fully connected multilayer feed-forward neural network topology.



Performance
- cost function
- Loss function

Advantages

- Performance Boosting
- Multi-Network capability
- Solves Complex tasks
- Handle Non-linear data
- Provides better decision
- Works with noisy data

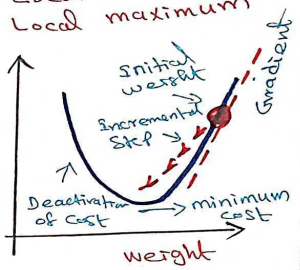
Gradient Based Learning:

* It is an optimization algorithm
* works based on Cost function

Gradient based learning
- Batch
- Mini Batch
- Stochastic

Gradient (or) Steepest Descent:

* Local minimum
* Local maximum



Merits of Batch Gradient descent

- Less noise
- Stable convergence
- computationally efficient

Merits of Mini-batch Gradient descent

- Easily fits with allocated memory
- Computationally efficient
- Stable convergence

Stochastic Gradient - Merits

- memory allocation is easy
- Faster response
- Works effectively with large data

Challenges

- finding local minimum for non-convex problem
- vanishing gradients
- Exploding gradient.

Hidden Units:-

Restricted Boltzmann machine parts

- Visible units
- Hidden Units
- Weights
- Hidden bias units
- Visible bias units

Input units → Hidden units → output units

Hidden layers

Hidden neurons

Optimization supported Selection

Architecture Design:

Deep learning Architecture

Supervised learning

Unsupervised learning

Convolutional Neural Network

Recurrent Neural Network

Self organizing maps

Long-short term memory

Auto encoders

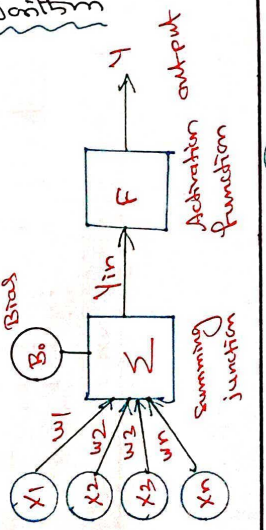
Gated Recurrent unit

Restricted Boltzman machines

Deep Networks

- Unsupervised Pretrained Network
- Convolutional Neural Network
- Recurrent Neural Network
- Recursive Neural Network

Back-propagation Algorithm



Steps in Back-propagation

- Receive inputs from pre Connected pabs
- Model inputs using weights
- Calculate output of each neuron
- Calculate Error in outputs

Regularization of Deep-learning

* Procedure to avoid overfitting

Types

- L2 & L1 Regularization
- Dropout
- Data augmentation
- Early stopping

Types

- Static Back propagation
- Recurrent Back-propagation

Regularization

- Ridge (L2) regularization
- Lasso (L1) Regularization

Differentiation algorithms :-

Input → Chain-Rule of differentiation → output

- modified scheme to get Life improvement of Neural scheme
- Modifies over/under fitted data

* It helps in avoiding over/under fit

Optimization for training and testing Approaches:

* Optimization for training Deep models

- Optimization Algorithms
  - Performance improvement in Deep learning
    - Change learning Rate
    - Change weights

- Deep learning optimizers
  - Conventional approaches
  - Heuristic methods
  - Meta-heuristic Methods
  - Hybrid approach

Types of Deep learning optimizers

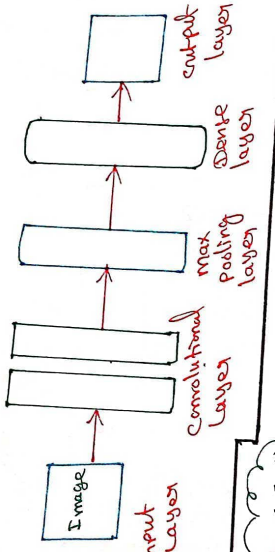
- Gradient descent
- Stochastic gradient descent
- Stochastic Gradient descent with momentum
- mini-batch gradient descent
- Adagrad
- RMS Prop
- Ada Delta
- Adam

Optimizers → Deep learning Architecture (Input → output) → Verification

Convolutional Networks:

- Deep Learning
  - Convolutional Neural Network (CNN)
    - Computer vision

CNN Architecture:



- Input Layer: Raw input with width, height, depth
- Convolutional layers:
  - Apply kernals to input data
  - Provides feature maps
- Activation layer: Adds nonlinearity to the network
- Pooling layer: To reduce size of data volume
- Flattening: Converting feature maps into 1D vector
- Fully Connected layer: It provides classification or regression result.

- Advantages of CNN
  - Detects patterns in 1D, 2D & 3D data
  - Robust performance
  - Can handle large data

SEQUENCE MODELLING

Recurrent and Recursive nets:—

Recursive Neural Network (features)

- It implements same set of weights
- The nodes may be traversed
- Trained using reverse mode of automation (Automatic differentiation)

Recursive Neural network (RvNNs)

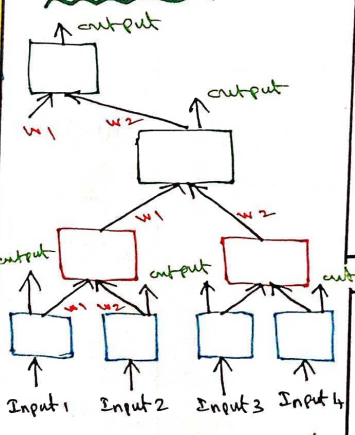
Generalized form of Recurrent Neural Network (RNNs)

Tree like structure helps to learn better on the data

RvNNs

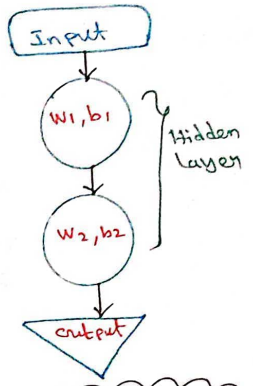
- Child node and parents
- Child and parent has weight matrix
- Similar children share same weights

General structure:



Processed or Raw input for model development

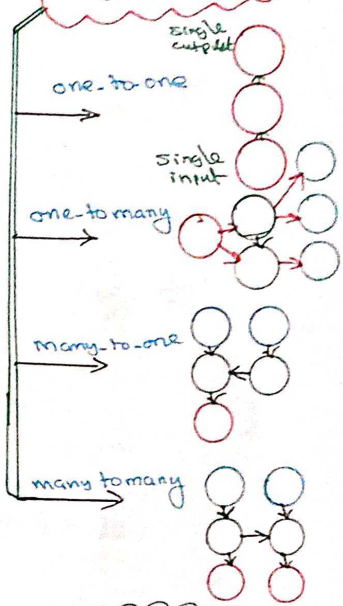
Recurrent Neural Network:



Training of RNN

- Providing single-time step of input
- Calculate current state using Previous state
- Update input to next state
- Repeat based on time step
- Calculate output when step-time completes
- Compare output and update.

Types of RNN



Merits of RNN

- Remembers Complete information
- Can be used with CNN

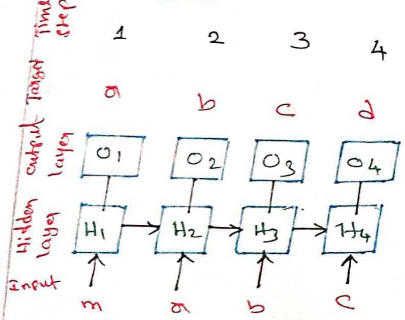
Dis advantages of RNN

- Gradient vanishing & Exploding
- Training is Complex
- Will not work for long sequences

Applications of RNN

- Language modelling & text generation
- speech recognition
- machine translation
- Face detection

Deep Recurrent Networks:

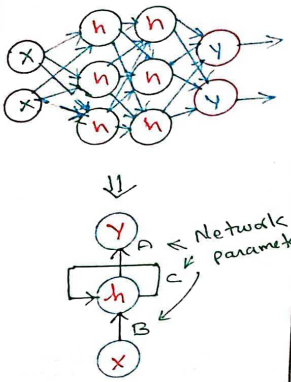


Sequence models:-

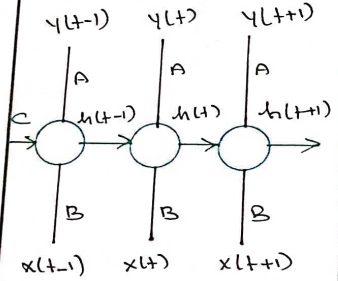
- Sequency models
- Text sentences
- Time-series data
- Discrete sequence data

RNN implementation methodology:-

- RNN Construction
- Save the output of a particular layer
- Feed back the saved information to the input layer
- Predict the output



Fully connected RNN:

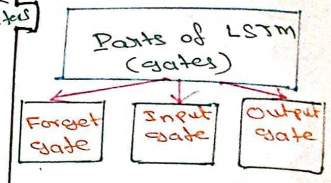


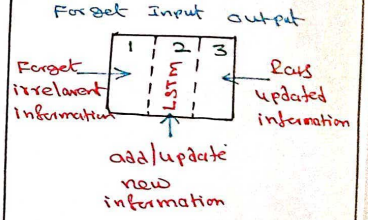
$h(t) = f_c(h(t-1), x(t))$

$h(t)$ → new state; $x(t)$ → input
$h(t-1)$ → old state; f_c → function

Long Short-Term Memory

- Advance form of RNN to handle vanishing gradient problem
- It is a sequential neural network, which allows information to persist





Forget gate:

$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f)$$

Input gate:

$$i_t = \sigma(X_t * U_i + H_{t-1} * W_i)$$

output gate:

$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o)$$

LSTM Applications

- Language modeling
- Machine translation
- Handwriting recognition
- Image captioning
- Image generation
- Question answering
- Video-to-text conversion
- Music modeling
- Speech Synthesis
- Protein structure prediction

Performance Metrics:

* The quality of the developed model is confirmed using performance metrics.
* classification task:
  - Confusion matrix
  - Accuracy; • Precision,
  - Recall; • F-score
  - Area under the curve (AUC)

Confusion Matrix

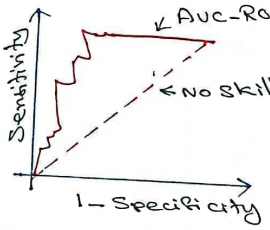
* Common procedure to measure the merit of model
* Table which relates Actual and Predicted values.

Predicted \ Actual	1 (Yes)	0 (No)
1	True-positive (TP)	False-positive (FP)
0	False-Negative (FN)	True-Negative (TN)

* TP ⇒ Actual and Predicted data class = 1
* TN ⇒ Actual and Predicted data point = 0
* FP ⇒ Actual data = 0 and Predicted data = 1
* FN ⇒ Actual data = 1 and Predicted data = 0
* With the help of these measures, other metrics are computed.

* Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$
* Precision = $\frac{TP}{TP+FP}$
* Sensitivity (Recall) = $\frac{TP}{TP+FN}$
* Specificity = $\frac{TN}{TN+FP}$
* F1 score = $\frac{2*(Precision*Recall)}{Precision+Recall}$

Area Under Curve:



* These metrics are computed to confirm the model during classification task
* Confusion matrix values are considered to compute other metrics

Hyperparameters:-

* These parameters control the learning process
* These parameters control the model's output

Hyperparameter Optimization techniques

- Manual search
- Random search
- Grid search
- Halving
- Automated tuning
  - Bayesian method
  - Genetic Algorithm
- Artificial Neural Network tuning
- Bayes search

Examples of Hyperparameter

- Learning Rate
- Train-test data split
- Batch size
- Number of epochs
- Hidden layers and hidden neurons

Types of Hyperparameters

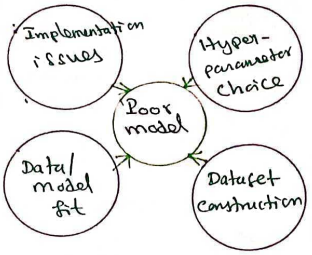
- Hyperparameters for Optimization
- Hyperparameters for specific models

Hyperparameter Optimization

- Learning Rate
  - Error value
  - weights updation
  - Cross-check
- Batch size
  - Data Split
  - Epochs
  - Validation and Stop

Debugging:

* when a developed model performs very poor, correction is required
* Model adjustment or replacement is needed



Steps to discover Error:

- Interpret the output of trained model
- Diagnosing the behavior of trained models
- Using synthetic data for decision making

Also check for:

* Data balancing
* Run time Error
* Choice of Optimizer
* Dropout rate
* Cross validation and Statistical Computation

Starting to develop a model

- Construct Simple scheme
- Consider known Optimizer
- Normalize inputs
- Use simple problem

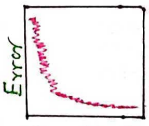
Evaluation of model

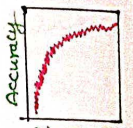
- Irreducible Error (Human performance)
- Avoidable bias (measure of underfitting)
- Variance (measure of overfitting)
- Validation set overfitting (Difference between test and validation error)

Debug Procedure for Neural Networks

- Start with Simple problem
- Implement & Debug
- Evaluate
- Meets requirement
- Improve model
- Tune Hyperparameters

Debugging the Learning Curve





* Stop the training and adjust the model when the validation error decreases compared to training Error

PROBABILISTIC MODELS PCA AND FACTOR ANALYSIS

DEFINITION :

Probabilistic principal components analysis (PCA) is a dimensionality reduction technique that analyzes data via lower dimensional latent space.

THE MODEL

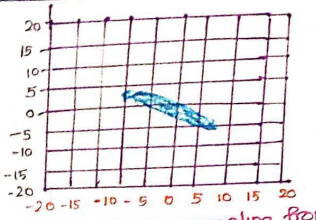
data set $X = \{x_n\}$,

$x_n | z_n \sim N(Wz_n, \sigma^2 I)$,

Probabilistic PCA generalises classical PCA,

$$x_n \sim N(0, WW^T + \sigma^2 I)$$

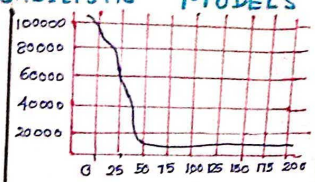
THE DATA

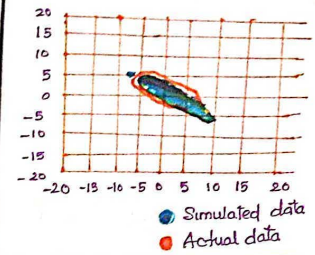


- Generate data by sampling from the joint prior distribution. Visualise dataset.

MAXIMUM A POSTERIORI INFERENCE

- Calculating the values of W and Z
- Maximise posterior density.
- Model to sample data for inferred values for W and Z.

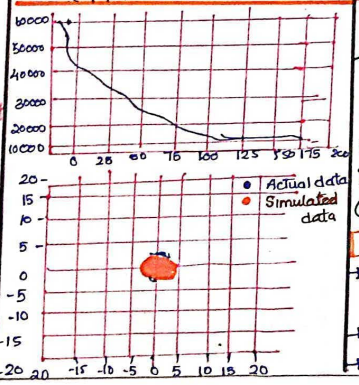




VARIATIONAL INFERENCE

MAXIMIZE EVIDENCE LOWER BOUND

$$E_q(w,z,\lambda)[\log p(w,z,x) - \log q(w,z;\lambda)]$$



FACTOR ANALYSIS

- finding latent variables in dataset.
- Common factor model
- hypothesizing that latent variables

$x_1 = c_{11} * k_1 + c_{12} * k_2 + d_1$

$x_2 = c_{21} * k_1 + c_{22} * k_2 + d_2$

$x_3 = c_{31} * k_1 + c_{32} * k_2 + d_3$

$x_4 = c_{41} * k_1 + c_{42} * k_2 + d_4$

Factor Analysis
- Apply Relations
- Estimate Components

PCA
- One fixed outcome
- Doesn't estimate Components

INDEPENDENT COMPOUND ANALYSIS

→ ICA is statistical and computational technique used in ML to separate multivariate signal into independent linear non-Gaussian components.

APPLICATIONS

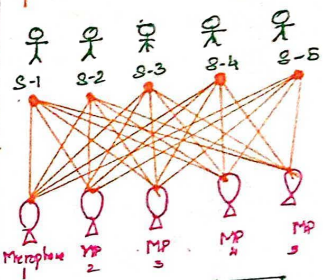
→ Signal processing, image analysis, data compression.
→ conjunction with other ML
→ Identify a set of basis functions.

Advantages of ICA

→ Ability to separate mixed signals
→ Non-parametric approach
→ Unsupervised learning
→ feature extraction

Disadvantages of ICA

→ Non-Gaussian assumption
→ Linear mixing assumption
→ Computationally expensive
→ Convergence issues.



PCA

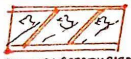
→ It reduces the dimensions to avoid problem of overfitting
→ deals with principal components

ICA

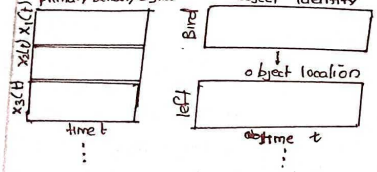
→ It decomposes mixed signal into its independent sources signal
→ Independent Components.

SLOW FEATURE ANALYSIS

SLOWNESS PRINCIPLE



high level representation object identity



DEF: An unsupervised learning algorithm for extracting slowly varying feature from a quickly varying i/p signal.

Illustration of slowness principle

* behaviourally relevant visual elements
* primary sensory signal.
* The difference in time scale leads to idea of slowness principle

OPTIMIZATION PROBLEM

$$Y_i(t) = g_i[x(t)](t)$$

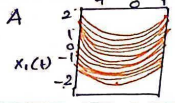
minimize

$$\Delta(y_i) = \langle y'^2 \rangle_t \quad (2)$$

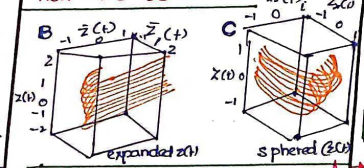
Avoid the trival constant solution — SFA seeks to 'maximise slowness — slowly varying output.

STEP BY-STEP ILLUSTRATION OF SFA ALGORITHM

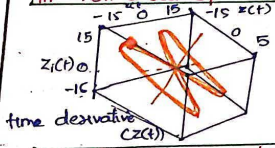
1. The non-linear problem transformed into linear functions



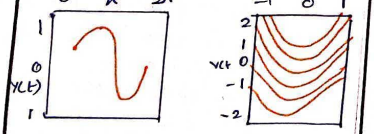
2. The expanded signal is normalised constraints



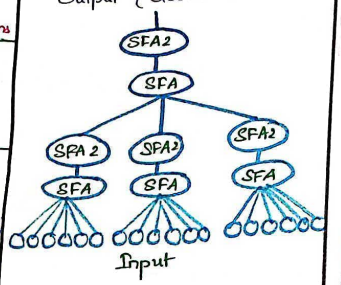
3. Temporal variation measured in normalised space.



4. The slowest-varying directions are extracted.



HIERARCHICAL SFA NETWORKS FOR HIGH-DIMENSIONAL DATA



SPARSE CODING

- data translated into a high-dimensional feature space.

ENCODE DATA

- learn sparse representation
- requires i/p data
- utilize unsupervised learning
- apply to any data
- find the representation

TWO CONSTRAINTS

- Create sparse representation as vector, data x (vector)
- encoder matrix, D
- data x, decoder matrix D

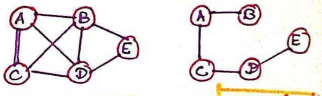
MATHEMATICAL SPARSE CODING

$$x \approx \sum_{i=1}^{n} h_i d_i = hD$$

- Sparse learning, D
- Sparse encoding, (D)
- Outer minimization, h_k, X_k D
- Inner minimization, L1.

SPARSE CODING NEURAL METHODS

- i/p neural networks
- frequent activation
- Avg activity ratio
- produce reliable neural code.



Applications of Sparse Coding

- Brain imaging
- Natural Language processing
- image processing
- Sound processing
- Available in several python modules
- Tool kit – Scikit learn
- Creating of dictionary functions

THE CHALLENGES OF UNSTRUCTURED MODELING

Goal : Scale ML to kinds of challenges needed to solve artificial intelligence.

* Understand high data-D with rich structure.

- Density Estimation
- Denoising
- Missing value imputation
- Sampling
- Memory
- Statistical Efficiency
- Runtime. the cost of inference

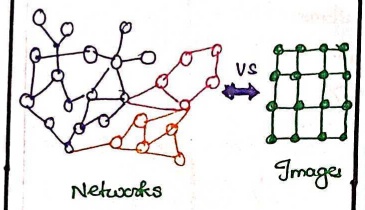
FORMAL FRAMEWORK FOR MODELLING

INTERACTIONS BETWEEN RANDOM VARIABLES

- Significant fewer parameters
- estimate reliably from less data
- Dramatically reduced computational cost & performing
- interferance in Model
- finds compact, efficient representations of complex data.

GRAPHS FOR MODEL STRUCTURE

- Describe relationships and interactions between pair of nodes.
- Think of nodes as users, edges as connections



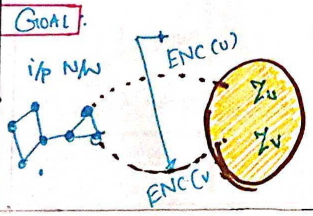
Text

GNN directly applied to graphs and provide an node-level, edge-level and graph-level prediction tasks

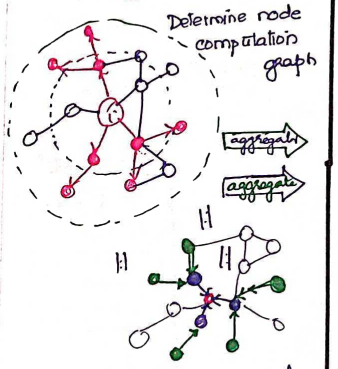
CNN's fail on graphs :

- takes a little sub-patch of image (a little rectangular part of image).
- apply functions.

GOAL:

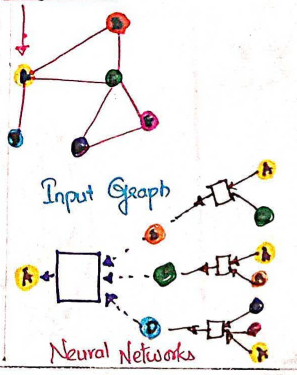


- Locality
- Aggregate information
- Stacking multiple layers



Propagate and Transform information

Target node



Training can be supervised and unsupervised.

* **Unsupervised Learning**
  - → Graph structure
  - → similar nodes
  - → Unsupervised loss function
* **Supervised learning**
  - → Train model
  - → anomalous node
* **Graph convolution**
* **Linear Layer**
* **Non-linear activation**

GraphSAGE idea

* simple neighborhood aggregation

$$h_v^k = \sigma\left(W_k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B_k h_v^{k-1}\right)$$

MEAN : Take a weighted avg of neighbors

POOL : Transform neighbour vectors, symmetric vector function

LSTM : Apply LSTM to reshuffled of Neighbours.

SAMPLING FROM GRAPHICAL MODELS

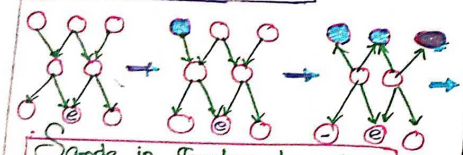
Importance of Sampling :
- → Transform the probabilistic inference
- → Generate random samples from Q
- → Estimate expected value

Properties :
- Convergence
- Unbiased
- Variance

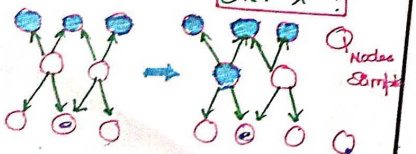
EFFECTIVE SAMPLE SIZE

$$P(x_i|e) = \sum_z g_{x_i}(z) P(z|e)$$

WEIGHTING SAMPLING



Sample in Topological order over X !

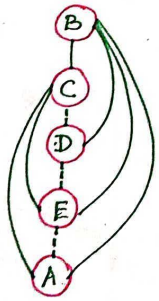


$P(X_i|Pa_i) P(X_i|Pa_i)$

PERFECT SAMPLING USING BUCKET ELIMINATION

ALGORITHM
- Run Bucket elimination
  $O = (X_N, \dots X_1)$
- Reverse ordering :
  $(X_1, \dots, X_N)$
- Probability
  $P(X_i | X_1, \dots X_{i-1})$

bucket B: $P(B|A)\ P(D|B,A)$
bucket C: $P(C|A)\ h^B(A,D,C,e)$ — $P(e|B,C)$
bucket D: $h^C(A,D,e)$
bucket E: $h^D(A,e)$
bucket A: $P(a)\ h^E(a)$ — $P(e)$



bucket
- B : $P(B|A)\ P(D|B,A)\ P(e|B,C)$
- C : $P(C|A)\ h^B(A,D,C,e)$
- D : $h^C(A,D,e)$
- E : $h^D(A,e)$
- A : $P(A)\ h^E(A)$

$Q(A) \propto P(A) \times h^E(A)$

Sample: $A = a \leftarrow Q(A)$

MINI BUCKETS : "LOCAL INFERENCE"

ELIMINATION
- → Space and Time Constraints
- → BE generates a function having scope size : 3

SAMPLING OUTPUT OF MBE

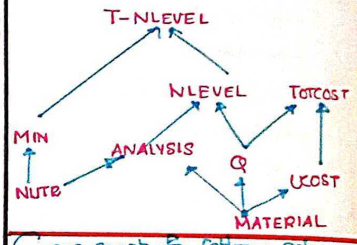
Bucket B :	P(e|B,C)	P(B|A) P(D|B,A)
Bucket C :	P(C|A)	$h^B(C,e)$
Bucket D :	$h^B(A,D)$	
Bucket E :	$h^C(A,e)$	
Bucket A :	$h^E(A)$. $h^D(A)$	

Sampling is same as in BE Sampling.

Gibbs SAMPLING
↳ Markov Chain Monte Carlo.

ADVANTAGES OF STRUCTURED MODELLING

SML → Semantic framework that deliberately represents.
→ Supports Structured modelling • Semantic framework.



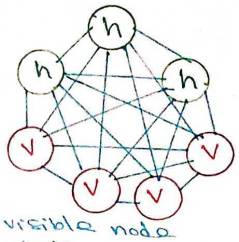
Genus graph for Rubrmix schema

Advantages :
1. Error detecting
2. Ability
3. Browsable definitional dependency
4. Automatic generation of Rdata
5. partial consistency
6. Complete independence between general structure
7. complete independence between models and solvers

DEEP GENERATIVE MODELS

Boltmann machines:-

* It is an unsupervised deep-learning method
* Every node of the network is interconnected
* It is a generative deep-learning scheme



v - visible node
h - hidden node

Types

- Restricted Boltzmann machine (RBM)
- Deep belief network (DBN)
- Deep Boltzmann machine (DBM)

* Global Energy "E" in a Boltzmann machine:

$$E = -\left(\sum_{i<j} W_{ij} S_i S_j + \sum_i \Theta_i S_i\right)$$

- W_{ij} → Connection strength of ij
- S_i → State of unit i
- Θ_i → Bias of unit i

Restricted Boltzmann machines (RBM):

* Hidden nodes cannot be connected to one another
* Visible nodes connected to one another

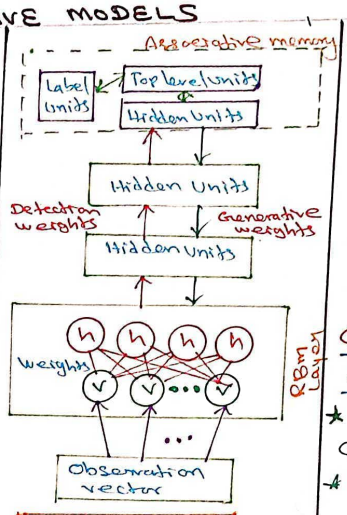
Applications

- Colloborative filtering
- Image and video processing
- Natural language Processing
- Bioinformatics
- Financial modeling
- Anomaly detection

Deep Belief Network:

- It is a stack of RBM
- Top two layers are undirected
- Connection between all lower layers are directed
- No intra layer
- It is a hybrid generative graphical model

* DBN are easy to implement using optimal hidden layers.

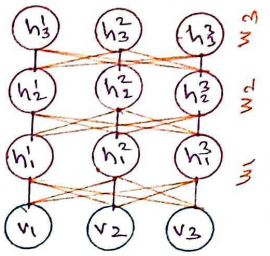


Applications

- Image generation
- Image classification
- Video recognition
- Motion Capture
- Speech processing

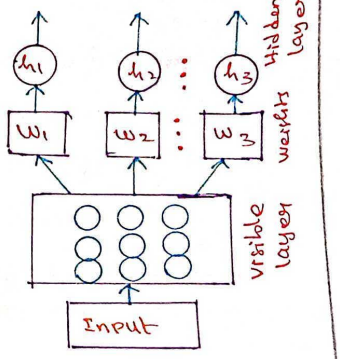
Deep Boltzmann Machine:

* Unsupervised models with Undirected Connections
* No intralayer Connections
* Learns using layer by layer pretraining

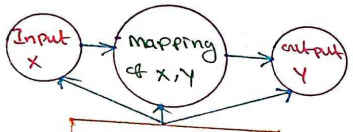


Convolutional Boltzmann Machine:

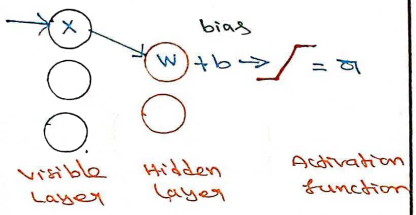
* Working is similar to convolutional neural network
* Handles very high Dimensional inputs
  - Large image file
  - Video processing



Boltmann Machine for Structured and sequential outputs:



* Model estimates a probabilistic distribution



activation 'f' ((W * X)+b) = σ

Directed Generative Networks

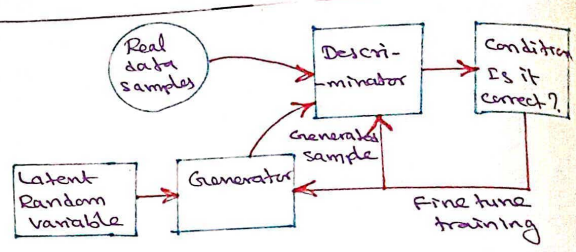
* It is a graphical model
* Generate new data instances
* It captures joint probability p(X,Y).

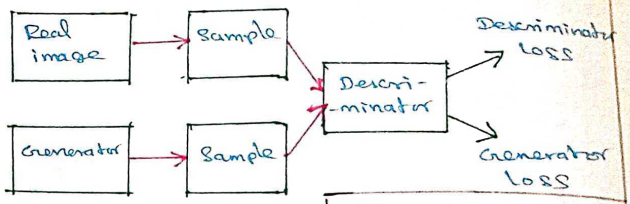
Directed Generative Networks

- Sigmoid belief nets
- Differentiable generator nets
- Variable autoencoders
- Generative adversal networks (GAN)
- Generative moment matching networks
- Convolutional generative nets
- Autoregressive nets
- Linear Autoregressive
- Neural Auto regressive
- Neural autoregressive density estimators.

Generative Adversarial network (GAN):

- Generative: Generates data interms of a probabilistic model
- Adversarial: Training of model is done in an adversarial setting
- Network: Uses a deep neural model for the training





Types

- Deep convolutional GAN
- Conditional GAN
- Least square GAN
- Dual video discriminator GAN
- SR GAN
- Info GAN
- Auxilery classifier GAN (It is same as conditional GAN)
- cycle GAN

Advantages

- Synthetic data generation
- High-quality results
- Unsupervised learning
- Versality

Disadvantages

- Training instability
- Computational cost
- overfitting
- Bias & fairness
- Interpretability

Evaluating Generative models:

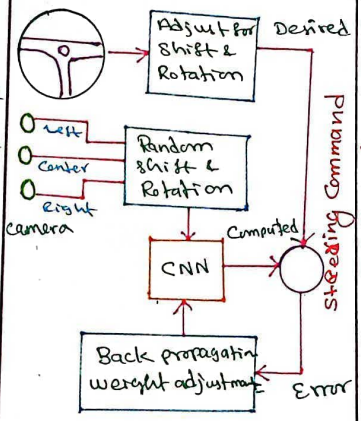
Evaluation
- Qualitative measures
- Quantitative measures

Common metrics
- Fidelity
- Diversity
- Authenticity
- Predictive performance
- Frechet inception distance
- Kernal inception distance
- Inception score
- Precision and Recall
- Perceptual path length

CASE STUDIES
* Knowledge discovery
* Predictive Analytics

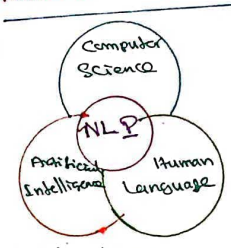
Examples:
- Self driving cars
- Natural language processing
- Virtual assistants
- Healthcare applications

Self Driving Car:
* It uses a Convolutional neural network
* The image captured from front-facing camera controls the whole system
* It provides a complete control based on the instruction by the CNN.



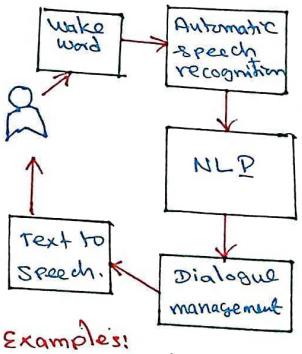
Training
- Data selection
- Augmentation
- simulation
- Evaluation
- on-road tests

Natural Language Processing



Applications:-
- Translation
- Automatic summarization
- Named entity recognition
- Speech recognition
- Relationship Extraction
- Topic segmentation

Virtual Assistants:



Examples:
* Amazon's Alexa
* Apple's Siri
* Google's Assistant

Health Care Applications:
* To diagnose the patient and to generate report
* To assist the doctor for decision making
* Precise treatment execution
* Care after treatment

Health care
- Early detection
- Improved decision making
- Treatment execution
- Expanded access to medical services
- Associated care
- Checking health through wearable
- Remote monitoring
- End of life care

Uses:
- Creation and maintenance of electronic health record
- Signal & Image based diagnosis
- Personalized medical treatments
- Response to patient queries