SCC361 Week 1

Lecture 1 {

Learning Outcomes {
Understand AI Concepts, Applications and Trends
Understand Machine Learning terms
Train Machine Learning Models for specific tasks
Learn and implement simple AI based systems
Learn how to evaluate the performance of AI systems
}

}

Introduction to AI and ML {

AI Overview

Definition

Goals of AI

AI and Society

NLP {

Text that combines meaning with letters

Examples:
- Web search engine
- Text classification
- Sorting algorithms
- Spam filtering
- Machine translation
- Question answering
- Recommender Systems

Speech Technologies:
- Siri, Alexa, Cortana, Google Assistant
- Automatic Speech Recognition
- Dialogue Systems

}

Explainability = Demonstrate how an AI came to a solution
Uncertainty/Robustness = How confident is the AI with its decision

History of AI {

Early Days

1943: McCulloch & Pitts
Boolean Circuit Model of Brain

1950: Turing
Computing Machinery and Intelligence

Excitement

1950s: Early AI Programs
Samuel's checkers program
Newell & Simon's Logic Theorist
Gelernter's Geometry Engine

1956: Darthmouth Meeting
Artificial Intelligence adopted

1965: Robinson
Complete algorithm for logical reasoning

Knowledge-based approaches

1969-79: Early development

1980-88: Expert systems
Industry Booms

1988-93: Expert systems industry bursts
AI Winter

Statistical Approaches + Subfield Expertise

Resurgence of probability, focus on uncertain
General increase in technical depth
Agents and machine learning systems
AI Spring?

Excitement (Now)

Big data, big compute, deep neural networks
Some re-unification of subfields
AI used in many industries

}

The Thinking Machine {

What is AI? {

Approach 1: Thinking like a human
Approach 2: Acting like a human
Mimicing human behaviour
Approach 3: Thinking rationally
Approach 4: Acting rationally

```
      Human     Rational
     +------------------------------------
  Thinking |Systems that  | Systems that
     |think like humans | think rationally
     |-------------------|-----------------
  Acting  |Systems that  | Systems that
     |act like humans | act rationally
     +------------------------------------
```

What is an Agent? {

Definition = Something that acts within an environment

Acts intelligently if:
- Action is appropriate
- Flexible to change
- Learns from experience
- Makes appropriate choices

Computational Agent: Agent whose decisions and actions can be explained in terms of computation

Rational Agent: Acts to achieve best outcome
}

AI studies the synthesis and analysis of computational agents that act intelligently
}
}

Goals of AI {

Scientific Goal = Understanding principles of intelligent behaviour
Engineering Goal = Concerned with constructing intelligent agents

Business Benefits:
- Workflow automation
- Enhance creative tasks
- Increased accuracy
- Better predictions

Social Benefits:
- Healthcare
- Smart cities, transport, security
- Forecasts and predictions
- Agriculture
- Overall Lifestyle

}

Risks and Challenges {

Safety and Security
Trust and Social Manipulation
Explainable AI

Possible Job Loss

}

Ethical Concerns {

Accountability
Accuracy, Bias, Privacy and Inequality
AI learns from data from humans, which may encode prejudice
Technological Social Responsibility

}

Machine Learning {

AI systems were mostly rule-based
Machine Learning drives AI

Machine Learning = Field of study that gives computers the ability to learn, without being explicitly programmed

Declarative Knowledge = Accumulation of old facts
Imperative Knowledge = Deduce new facts from old facts

Supervised Learning = Tell computer how to learn
- Classification = Relationships and Categories
- Regression = Predicting
Mapping input to an output

Unsupervised Learning =
- Clustering = Inherent Grouping within data
- Association =
}
}

361 Week 2

Lecture 1 {

Definition of AI {

Field that studies the synthesis and analysis of computational agents that act rationally

Definition of Computational Agent = Agent whose decisions and actions can be explained in terms of computation

Definition of Agent = Something that acts within an environment
}

Goals {

Scientific vs Engineering Goals

Uses general scientific approach

Focuses on empirical systems not applications

}

Benefits {

Business and Social

Business
Process Automation
Enhance creative tasks
Increased accuracy
Better predictions and improved decisions

Social
Healthcare
Smart cities
Forecasts and predictions
Agriculture
Overall Lifestyle
}

Risks {

Safety and Security
Trust and Social Manipulation
Explainability
Job Loss
Accountability
Accuracy, bias, privacy and inequality
Human biases and prejudices
Technological Social Responsibility

}

Features in Machine Learning {

Machine Learning models dependent on data
Machine Learning algorithms designed to understand numbers

Feature = Individual measurable property or characteristic of a phenomenon being observed
A set of features represent the information you can draw from data

Label = Tag you wish to assign to a set of features

Feature Vector = Numeric | Symbolic characteristics called features of an object in a mathematical analysable way

Training:

Data -> Features and Labels -> Machine Learning Algorithm -> Predictive Model

New:

Data -> Features Vector -> Predictive Model -> Expected Label
}

Feature Extraction {

Definition = Set of methods that map input features to output features

Any technique that transform raw data into features that can be used as input to a learning algorithm

Process of transforming raw data into numerical features that can be processed while preserves the information in the original data set

Raw Signal - > Preprocessing -> Features Extracted -> Classifier -> Output

Learning algorithms prefer numeric data
Real World data is often non-numeric
Real World data mostly unstructured
Increase learning accuracy by extracting most significant features

Approaches {

Manual:
Manually identifying and describing relevant features
Requires knowledge

Automated:
Uses specialised algorithms with no human intervention
More efficient

}

Feature Vector = Collection of features and their labels | New encoding of image

Feature Extraction from Images {

Detects and represents the interesting parts of an image as a compact feature vector
Critical step in image processing and computer vision

}

Good Feature {

+ Repeatable
Should be detectable at same location in different images despite changes in viewpoint and illumination

+ Saliency
Descriptive
Same points in different images should have similar features

+ Compactness
Affect speed of matching
Fewer and smaller features are best

}
}
}

Lecture 2 {

Image = Array of Pixels

Pixel Values can be extracted as feature vector representation for images

Real World <=> Computations

Real World : Continuous Space
-Infinity to Infinity
Theory

Computations : Discrete Space
Grid
Practice

Computer Vision Theory:
View Image as Function
u: Z x Z -> R = pixel rows and columns -> intensity values

View Video as Function
u: Z x Z x Z -> R = pixel rows and columns, frame -> intensity values

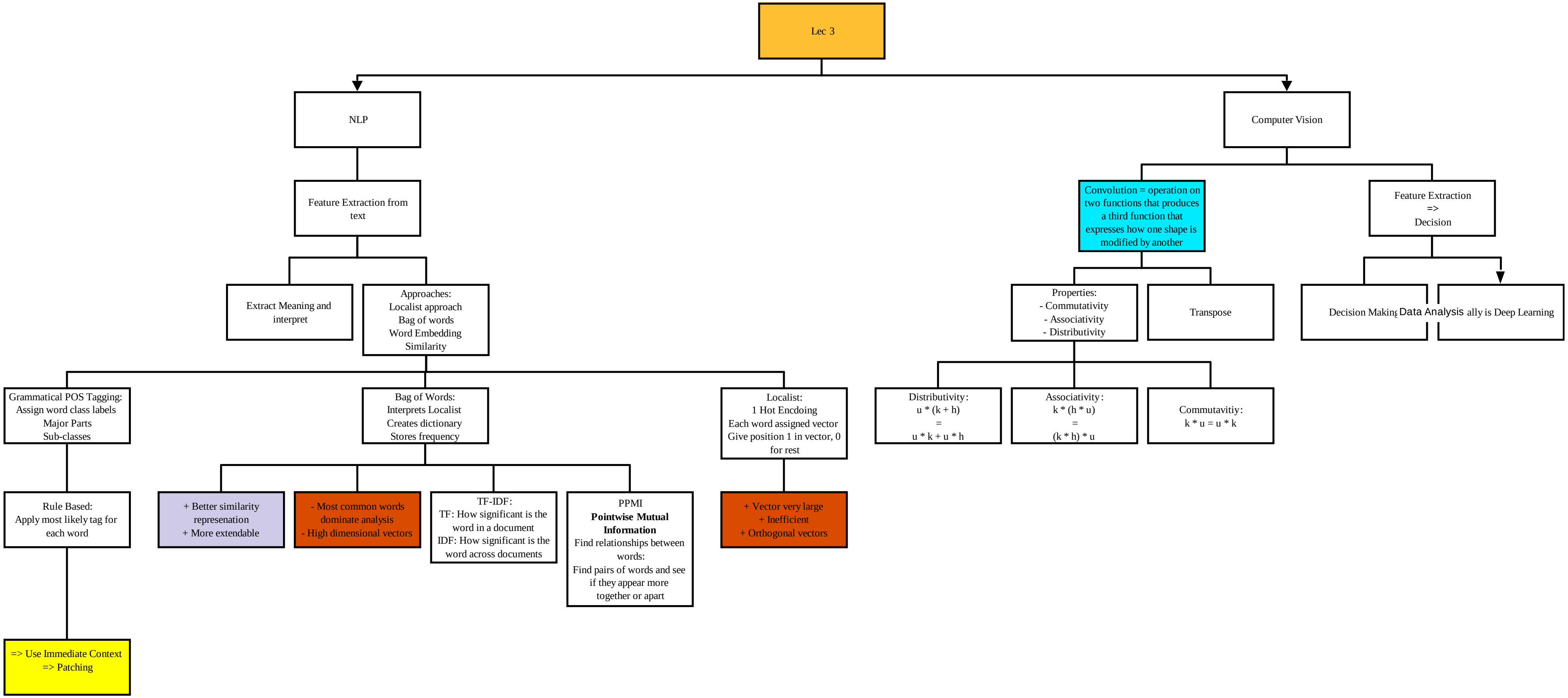
mesh(im(:,:,1))

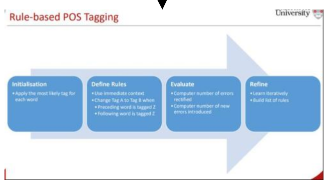
Edge Detection: Curve that follows a path of rapid change in image intensity
Used to identify edges in an image
Canny filter is best known => Gradients

// TODO:

CW = Create Edge Detector

Fully zoned out for the rest sorry bro
}


Lec 3
NLP
Computer Vision
Feature Extraction from text
Extract Meaning and interpret
Approaches: Localist approach Bag of words Word Embedding Similarity
Grammatical POS Tagging: Assign word class labels Major Parts Sub-classes
Rule Based: Apply most likely tag for each word
=> Use Immediate Context => Patching
Bag of Words: Interprets Localist Creates dictionary Stores frequency
+ Better similarity represenation + More extendable
- Most common words dominate analysis - High dimensional vectors
TF-IDF: TF: How significant is the word in a document IDF: How significant is the word across documents
PPMI Pointwise Mutual Information Find relationships between words: Find pairs of words and see if they appear more together or apart
Localist: 1 Hot Encdoing Each word assigned vector Give position 1 in vector, 0 for rest
+ Vector very large + Inefficient + Orthogonal vectors
Convolution = operation on two functions that produces a third function that expresses how one shape is modified by another
Properties: - Commutativity - Associativity - Distributivity
Transpose
Distributivity: u * (k + h) = u * k + u * h
Associativity: k * (h * u) = (k * h) * u
Commutavitiy: k * u = u * k
Feature Extraction => Decision
Decision Makin
Data Analysis
ally is Deep Learning

- 361 4
  - 

  - How to setup data for ML
    - Split into Training and Testing Data
      - 1. Training Process 2. Prediction 3. Evaluation
      - Also sometimes has validation data (Data tested during testing process)
        - Has no effect on model
  - Clustering and Classification
    - Supervised vs Unsupervised
    - Classification
      - Assigning categories to groups
        - Aim is to reduce dissimilarity in a cluster
        - Maps Feature Vector to Label
        - K Nearest Neighbour = Classify by closest Neighbours
    - Clustering
      - Identifying meaningful patterns/clusters/ groups in observed data points
        - Hierarchical vs K-Means
          - K-Means = Optimisation Problem

            Aim to minimise the within-cluster sum of squares (Variance)
            - Choose K by:
              - Prior knowledge of data space
              - Use elbow method
              - Run hierarchical clustering on subset of data
            - K = Amount of Categories
            - - Efficient
              - Number of Clusters can influence "wrong" results
              - Non deterministic
            - ```
              K = Choose
              while (true) {
                  For 1 to K: Assign point to closest
                  centroid

                  For 1 to K: Compute new centroid by avg
                  points in each cluster

                  If Centroids don't change
                      Stop
              }
              ```
          - Hierarchical = 1: Initialisation 2. Merge iteratively 3. Result
            - - Deterministic
              - Inefficient
              - Multilevel Representation
              - Flexible
            - Cluster Created by Splitting Data on Different Levels
            - Linkage = Distance

              Single = 1 to 1

              Complete = Furthest

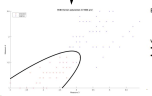
              Average = Average
              ```

361 5

Support Vector Machines

Separates out data

AIM: To automatically determine the correct label yi given the data xi



Each item X_i in the set S is a pair, including data and a label:

$$X_1 = (x_1, y_1), X_2 = (x_2, y_2), \dots, X_n = (x_n, y_n)$$

where
- Each x_i is a p-dimensional real vector
- Each y_i is a corresponding label for x_i. For this, we will use

$$y_i = \begin{cases} 1 & if\ X_i \in S_1 \\ -1 & if\ X_i \in S_2 \end{cases}$$

Creates Polynomial to separate data

Helps with Decision Making

Building SVM

Problem statement

Each item X_i in the set S is a pair, including data and a label:

$$X_1 = (x_1, y_1), X_2 = (x_2, y_2), \dots, X_n = (x_n, y_n)$$

where
- Each x_i is a p-dimensional real vector
- Each y_i is a corresponding label for x_i. For this, we will use

$$y_i = \begin{cases} 1 & if\ X_i \in S_1 \\ -1 & if\ X_i \in S_2 \end{cases}$$

Find maximum margin hyperplane

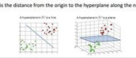
Building SVM

Hyperplanes

General equation for a hyperplane:

$$w^{\top}x - b = 0, \qquad where\ x = (x_1, x_2, \dots, x_n)$$

where
- w is the normal vector to the hyperplane
- $\frac{b}{\|w\|}$ is the distance from the origin to the hyperplane along the normal



Minimise $||w||$! Maximise distance

Kernels

Linear: $k(x_i, x_j) = x_i \cdot x_j$

Polynomial: $k(x_i, x_j) = (x_i \cdot x_j)^p$

Radial Basis Function: $k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$

Sigmoid Function: $k(x_i, x_j) = \tanh(\kappa_1 + \kappa_2 x_i \cdot x_j)$

Kernels

Multiclass SVM

Combine two classes to form a single class, and keep on doing this

AIM: Create Binary Problem

Measures of Success and Error Measures

$$Accuracy = \frac{\#\{\text{Correct Predictions}\}}{|S_x|}$$

$\#\{\text{Correct Predictions}\} = 4$

$|S_x| = 5$

$Accuracy = \frac{4}{5} = 0.8 = 80\%$

$$Accuracy = \|l - \hat{l}\|_2 = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$$

Actual Centre $l = (x, y)$ — Predicted Centre $\hat{l} = (\hat{x}, \hat{y})$

Confusion Matrices

TP = + Right
TN = - Right
FP = + Wrong
FN = - Wrong

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

True Class

Predicted Class	Positive	Negative
Positive	TP	FP
Negative	FN	TN

We consider
Sensitivity
Specificity
Precision
Negative Predictive Value
Miss rate
Fall out
False discovery rate
False omission rate
F1 score

Sensitivity: Probability of positive outcome if truly positive

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{|True\ Positives|}$$

Specificity: Probability of negative outcome if truly negative

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{|True\ Negatives|}$$

Precision: Proportion of truly positive outcomes to positive predictions

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{|Predicted\ Positives|}$$

NPV: Proportion of truly negative outcomes to negative predictions

$$NPV = \frac{TN}{TN + FN} = \frac{TN}{|Predicted\ Negatives|}$$

F1 Score: Harmonic mean of precision and recall

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2\frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

Categorical labels:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

True Class

Predicted Class	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Artificial Neural Networks

Scale to perform well in difficult spaces

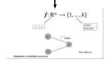
Simple Decisions ==> Refine using Deep Learning

As Data increases so does performance

AI => ML => ANN => DL

Mathematical Models inspired by Biological Neural Networks

Each neuron receives input signal and propagates to connected neuron through synapses



Define Hypothesis Function:

Takes input data and uses weights to make decision about data

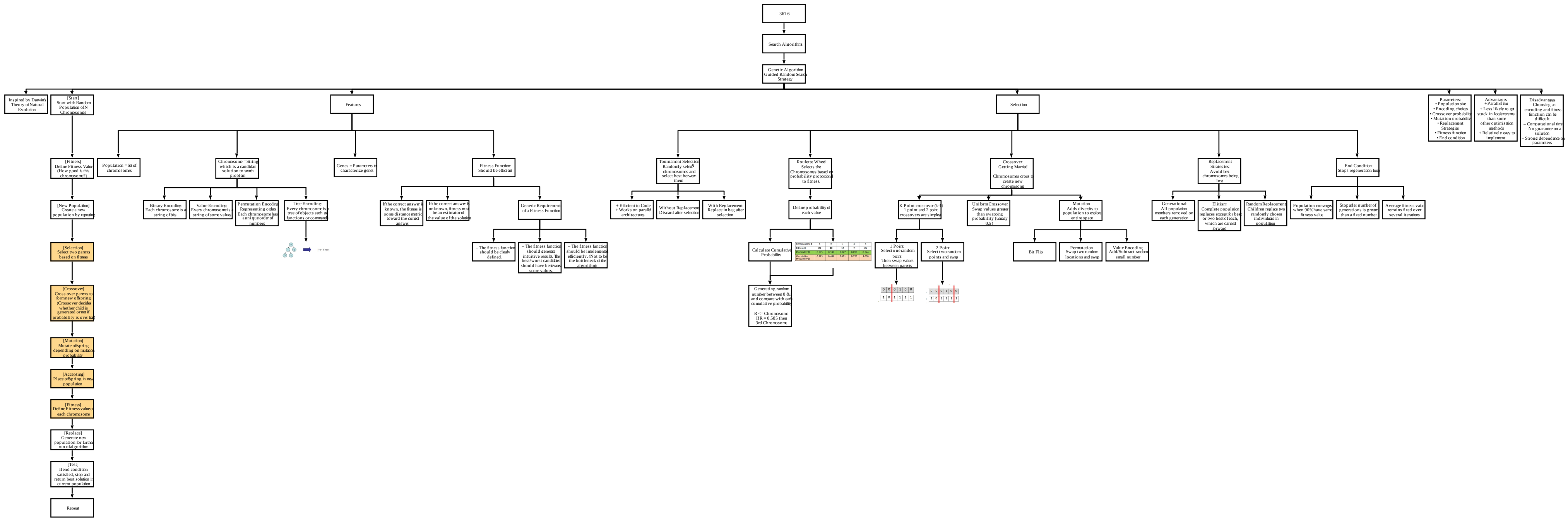
Activation Function: Takes output and maps onto decision

- Logistic Sigmoid Function
- Rectified Linear Unit
- Binary Step Function

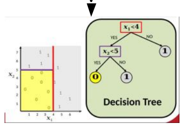
Input neurons at one end, output neuron at other

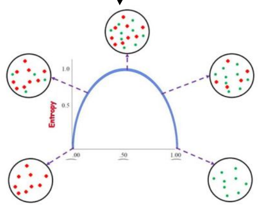
Training the Network:

- Learn how to adjust weights
- Aim: Learns by choosing different weights to minimise the loss/errors
- Loss Function: Errors determined by some loss function that compares it output with true output
- Solution Technique: + Weights adjusted using technique called gradient descent
+ Algorithm for minimizing loss when training neural network


361 6
Search Algorithms
Genetic Algorithm Guided Random Search Strategy
Inspired by Darwin's Theory of Natural Evolution
[Start] Start with Random Population of N Chromosomes
[Fitness] Define Fitness Value (How good is this chromosome?)
[New Population] Create a new population by repeating
[Selection] Select two parents based on fitness
[Crossover] Cross over parents to form new offspring (Crossover decides whether child is generated or not if probability is over half
[Mutation] Mutate offspring depending on mutation probability
[Accepting] Place offspring in new population
[Fitness] Define Fitness value of each chromosome
[Replace] Generate new population for further run of algorithm
[Test] If end condition satisfied, stop and return best solution in current population
Repeat
Features
Population = Set of chromosomes
Chromosome = String which is a candidate solution to search problem
Binary Encoding: Each chromosome is a string of bits
Value Encoding: Every chromosome is a string of some values
Permutation Encoding: Representing orders Each chromosome has a unique order of numbers
Tree Encoding: Every chromosome is a tree of objects such as functions or commands
Genes = Parameters to characterize genes
Fitness Function: Should be efficient
If the correct answer is known, the fitness is some distance metric toward the correct answer
If the correct answer is unknown, fitness must be an estimator of the value of the solution
Generic Requirements of a Fitness Function
– The fitness function should be clearly defined.
– The fitness function should generate intuitive results. The best/worst candidates should have best/worst score values.
– The fitness function should be implemented efficiently. (Not to be the bottleneck of the algorithm)
Selection
Tournament Selection: Randomly select S chromosomes and select best between them
+ Efficient to Code
+ Works on parallel architectures
Without Replacement: Discard after selection
With Replacement: Replace in bag after selection
Roulette Wheel: Selects the Chromosomes based on probability proportional to fitness.
Define probability of each value
Calculate Cumulative Probability
Generating random number between 0 & 1 and compare with each cumulative probability
R <= Chromosome
If R = 0.585 then 3rd Chromosome
Crossover: Getting Married
Chromosomes cross to create new chromosome
K Point crossover (k=1) 1 point and 2 point crossovers are simplest
1 Point: Select one random point Then swap values between parents
2 Point: Select two random points and swap
Uniform Crossover: Swap values greater than swapping probability (usually 0.5)
Mutation: Adds diversity to population to explore entire space
Bit Flip
Permutation: Swap two random locations and swap
Value Encoding: Add/Subtract random small number
Replacement Strategies: Avoid best chromosomes being lost
Generational: All population members removed on each generation
Elitism: Complete population replaces except for best or two best of each, which are carried forward
Random Replacement: Children replace two randomly chosen individuals in population
End Condition: Stops regeneration loop
Population converges when 90% have same fitness value
Stop after number of generations is greater than a fixed number
Average fitness value remains fixed over several iterations
Parameters:
• Population size
• Encoding choices
• Crossover probability
• Mutation probability
• Replacement Strategies
• Fitness function
• End condition
Advantages:
+ Parallelism
+ Less likely to get stuck in local extrema than some other optimisation methods
+ Relatively easy to implement
Disadvantages:
– Choosing an encoding and fitness function can be difficult
– Computational time
– No guarantee on a solution
– Strong dependence on parameters

361 7

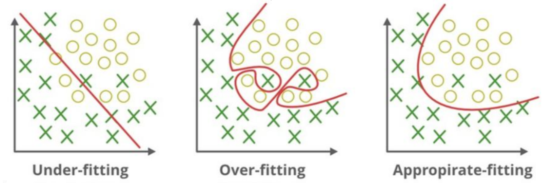
- Decision Trees: Contains test feature test conditions to separate samples
  - Linearly Separable Data: Data that can be separated by lines
    - Non Linear Data can be separated using Decision Trees
  - Classifying using trees
    - 

  - Consists of:
    - Root Node
    - Internal Node
    - Leaf Node (assigned class labels)
  - Exponentially Many Decision Trees that can be constructed from a set of features
  - Best Decision Trees:
    - Accurately separating trees
    - As small as possible
    - Heuristic Algorithms to find best decision tree
      - ID3 Algorithm: Find most important features
        - Determine best feature by reducing impurity
          - Impurity is positively correlated to uncertainty
          - Less Uncertainty = Better sample
        - Entropy: Set of samples, P_T = Pure True, P_F = Pure False
          - 

          - $E(S) = -P_T \log_2 P_T - P_F \log_2 P_F$

            $P_T = \frac{9}{16}$ $P_F = \frac{7}{16}$

            $E(S) = -\frac{9}{16}\log_2\frac{9}{16} - \frac{7}{16}\log_2\frac{7}{16} = 0.99$
          - Highest when Uncertainty is Greatest
          - Entropy is a metric telling us the uncertainty of the dataset
          - Information Gain: Expected reduction in Entropy

            Gain(S,A) = Entropy(Parent) - [Weighted Average] Entropy(Children)
            - Difference between Entropy
  - Summary
    - + Can apply to data from any distribution
      + Simple to understand
      + Handles numerical and categorical data
      + Fast
    - - Trees can be very non-robust (Small change in training => large change in tree)
      - Problem of learning optimal tree is NP-Complete
      - Prone to overfitting, especially when deep
- Overfitting: Means Memorising Not learning
  - 

  - Avoid Overfitting by:
    - Random Decision Forests: Collection of decision trees whose results aggregated into one final result
      - RDF by training on different samples of data:
        - RDF by trainin...

          Zoned out here
      - Random Subset of Features: Adding randomness to each decision tree by searching for the best feature among a random subset of features
        - Results in a Wide Diversity that generally results in a better classification model
    - Set Max Tree Depth: Increases Error