Unit 3
Classification

Basic Concept, Decision Tree

Rupak Raj Ghimire

Objective

- Basic Concept of Classification

Classification

- Given a collection of records (training set)
  - Each record is by characterized by a tuple (x,y), where x is the attribute set and y is the class label
    - x: attribute, predictor, independent variable, input
    - y: class, response, dependent variable, output
- Task:
  - Learn a model that maps each attribute set x into one of the predefined class labels y

Classification Task

Task	Attribute Set, x	Class Label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from x-rays or MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

Classification Model

- A classification model is an abstract representation of the relationship between the attribute set and the class label

- More formally, we can express it mathematically as a target function f that takes as input the attribute set *x* and produces an output corresponding to the predicted class label.

- The model is said to classify an instance (x, y) correctly if $f(x) = y$

Example

- Classifying vertebrates into mammals, reptiles, birds, fishes, and amphibians

Vertebrate Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Example

- The attribute set includes characteristics of the vertebrate such as its body temperature, skin cover, and ability to fly.

- The data set can also be used for a binary classification task such as mammal classification, by grouping the reptiles, birds, fishes, and amphibians into a single category called non-mammals

Example

- Consider the problem of predicting whether a loan borrower will repay the loan or default on the loan payments

Example

- The attribute set includes personal information of the borrower such as marital status and annual income, while the class label indicates whether the borrower had defaulted on the loan payments

ID	Home Owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125000	No
2	No	Married	100000	No
3	No	Single	70000	No
4	Yes	Married	120000	No
5	No	Divorced	95000	Yes
6	No	Single	60000	No
7	Yes	Divorced	220000	No
8	No	Single	85000	Yes
9	No	Married	75000	No
10	No	Single	90000	Yes

Classification Model

- Classification Model
  - Predictive Model
    - Used to classify the previously unlabeled instances
    - A good classification model must provide accurate predictions with a fast response time
  - Descriptive Model
    - Used to identify the characteristics that distinguish instances from different classes
    - This is particularly useful for critical applications, such as medical diagnosis, where it is insufficient to have a model that makes a prediction without justifying how it reaches such a decision

Classification Model

- Take Example of vertebrate dataset
  - Predictive
    - Whole dataset can be used to predict the class label of the following vertebrate

Vertebrate Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

  - Descriptive
    - it can be used as a descriptive model to help determine characteristics that define a vertebrate as a mamma

General Framework for Building Classification Model

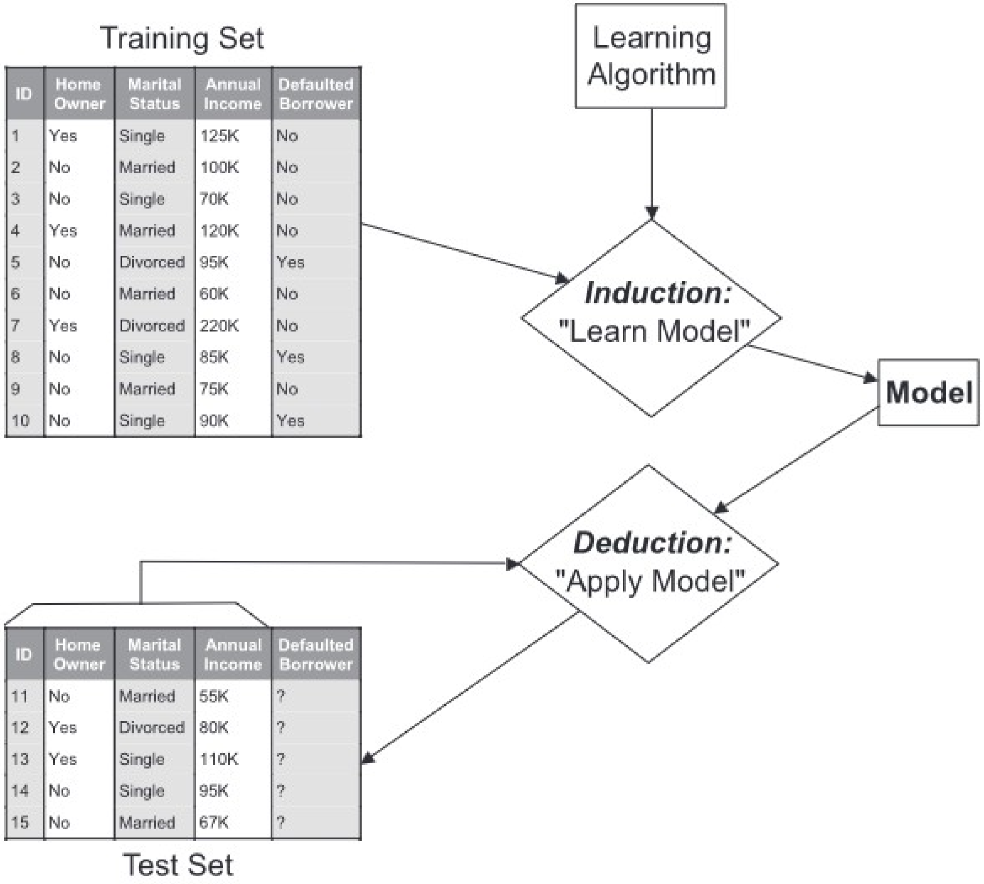
- Classifier
  - Classification is the task of assigning labels to unlabeled data instances and a classifier is used to perform such a task. A classifier is typically described in terms of a model as illustrated in the previous section
- Training Set
  - The model is created using a given a set of instances, known as the training set, which contains attribute values as well as class labels for each instance
- Learning Algorithm
- Induction
- Deduction

General Framework for Building Classification Model

- Learning Algorithm
  - The systematic approach for learning a classification model given a training set is known as a learning algorithm.
- Induction
  - The process of using a learning algorithm to build a classification model from the training data is known as induction
- Deduction
  - This process of applying a classification model on unseen test instances to predict their class labels is known as deduction

General Framework for Building Classification Model

- General Framework
- The process of classification involves two steps:
  - applying a learning algorithm to training data to learn a model, and
  - applying the model to assign labels to unlabeled instances



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
11	No	Married	55K	?
12	Yes	Divorced	80K	?
13	Yes	Single	110K	?
14	No	Single	95K	?
15	No	Married	67K	?

Classification Techniques

- Base Classifiers
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
  - Neural Networks, Deep Neural Nets
- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

Performance Measurement

- The performance of a model (classifier) can be evaluated by comparing the predicted labels against the true labels of instances
- This information can be summarized in a table called a **confusion matrix**

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Confusion Matrix

- The table depicts the confusion matrix for a binary classification problem

- Each entry f_{ij} denotes the number of instances from class *i* predicted to be of class *j*
- For Example
  - F01 is the number of instances from class 0 incorrectly predicted as class 1

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Confusion Matrix

- The number of correct predictions made by the model is (f11 + f00)
- The number of incorrect predictions is (f10 + f01)

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Model Accuracy

- Although a confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information into a single number makes it more convenient to compare the relative performance of different models.
- This can be done using an evaluation metric such as **accuracy**

Model Accuracy

- **Accuracy**

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

For binary classification problems, the accuracy of a model is given by

$$\text{Accuracy} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Model Accuracy

- **Error Rate**

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

  - The learning algorithms of most classification techniques are designed to learn models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set.

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %

- Accuracy is misleading because model does not detect any class 1 example

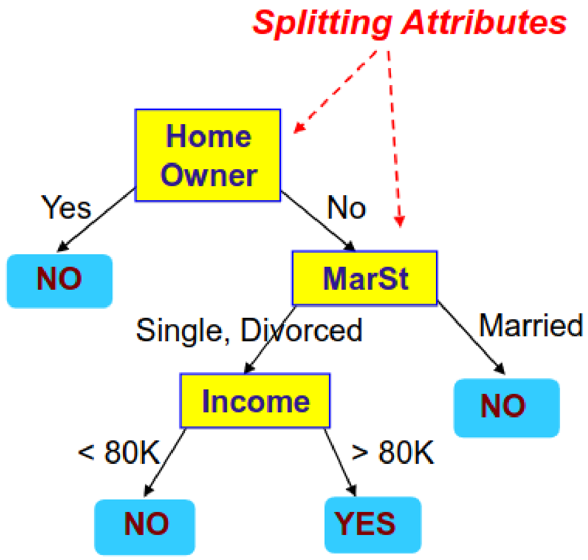
Decision Tree – Build tree

- Induction

categorical categorical continuous class

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

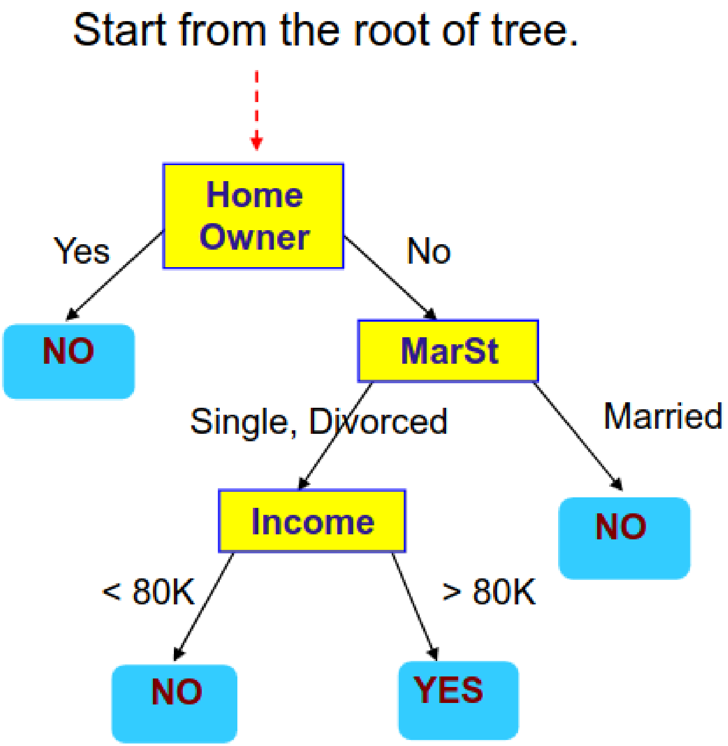
Training Data



Model: Decision Tree

Decision Tree – Apply Model

- Deduction



Test Data

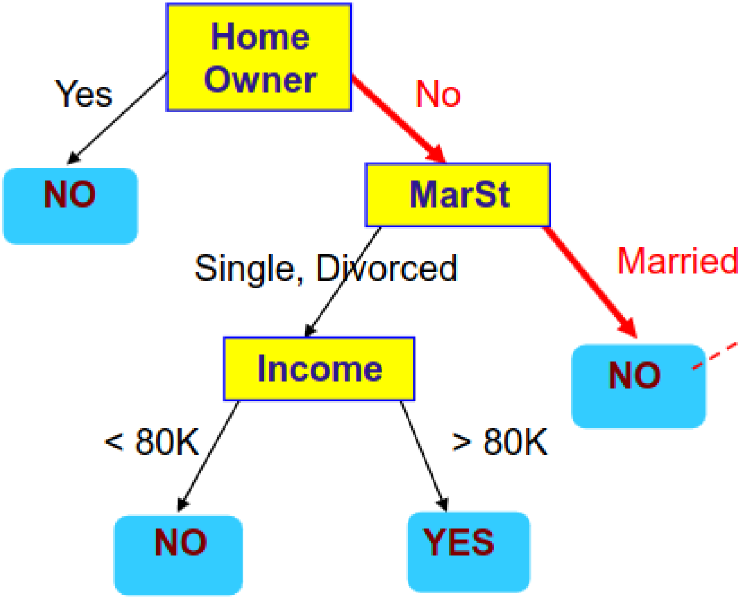
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Decision Tree – Apply Model

- Deduction

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

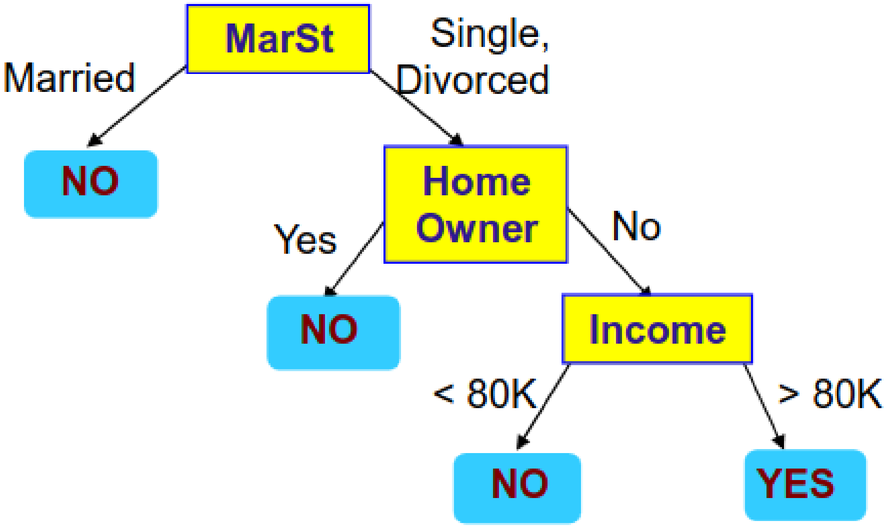


Assign Defaulted to "No"

Example

categorical categorical continuous class

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

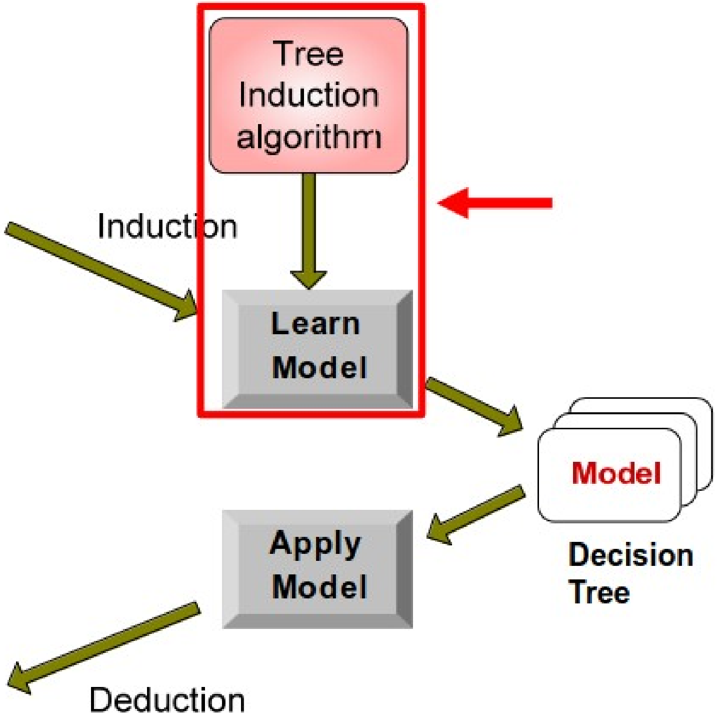


There could be more than one tree that fits the same data!

Modeling DT based Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Decision Tree

- Decision tree is a supervised machine learning algorithm used for classification task
- Root Node
  - The root node is where the tree starts.
  - It’s the big issue or decision you are addressing.
- Decision Node
  - The decision nodes represent a decision in your tree. They are possible avenues to "solve" your main problem

Decision Tree

- Leaf Node
  - The lead nodes represent possible outcomes of a decision.
- Branches
  - Branches are the arrows that connect each element in a decision tree.
  - Follow the branches to understand the risks and rewards of each decision.

Advantages of Decision Trees

- Compared to other classification algorithms, the concept is rather easy to understand.
- The decision tree can be visualized to help understanding or interpreting it.
- Can not only handle numeric, but also categorical data.

Disadvantages of Decision Tree

- Prone to overfitting, which means creating extremely complex trees that fail to properly generalize the data.
- Using only a simple decision tree is prone to variations; even small variations in the data can lead to a various different Decision Trees.
  - This can bee avoided by using ensembles of Decision Trees, which we will also look later.
- Depending on how the Decision Nodes are chosen, the data can be easily biased, which mean that certain classes dominate the Decision Tree.

Overfitting in Decision Tree algorithm

- The problem of overfitting is considered when the algorithm continues to go deeper and deeper to reduce the training-set error but results with an increased test-set error.
- So, accuracy of prediction for our model goes down.
- It generally happens when we build many branches due to outliers and irregularities in data.

Solution of Overfitting

- Pre-Pruning:
  - In pre-pruning, we stop the tree construction a bit early. We prefer not to split a node if its goodness measure is below a threshold value. But it is difficult to choose an appropriate stopping point.
- Post-Pruning:
  - In post-pruning, we go deeper and deeper in the tree to build a complete tree. If the tree shows the overfitting problem then pruning is done as a post-pruning step.
  - We use the cross-validation data to check the effect of our pruning. Using cross-validation data, we test whether expanding a node will result in improve or not. If it shows an improvement, then we can continue by expanding that node. But if it shows a reduction in accuracy then it should not be expanded. So, the node should be converted to a leaf node

Decision Tree Induction

- Many Algorithms:
- – Hunt’s Algorithm (one of the earliest)
- – CART
- – ID3, C4.5
- – SLIQ,SPRINT

Entropy

- Entropy is a measure of uncertainty or unpredictability
- Entropy is a measurement of a data set's impurity in the context of machine learning

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Entropy

- If we have a dataset of 10 observations belonging to two classes YES and NO. If 6 observations belong to the class, YES, and 4 observations belong to class NO, then entropy can be written as below.

$$E(S) = -(\, P_{yes}\log_2 P_{yes} + P_{no}\log_2 P_{no}\,)$$

- Pyes is the probability of choosing Yes and Pno is the probability of choosing a No. Here Pyes is 6/10 and Pno is 4/10.

$$E(S) = -\,(6/10 * \log_2 * 6/10 + 4/10 * \log_2 * 4/10) \quad \approx 0.971$$

Information Gain

- Information gain is the amount of knowledge acquired during a certain decision or action
- A feature's relevance to the categorization of the data increases with information gain
- Information gain is used to decide which feature to split on at each step in building the tree.

$$Information\ Gain = Entropy_{parent} - Entropy_{children}$$

Example – ID3 Algoritm

- Separate Document

Demo

- For ID3 follow the Lab Notebook

K-Nearest Neighbor Classifier

- Follow the lab notebook

Naive Bayes Classifier

- Follow the lab notebook

Artificial Neural Network

Neural Network

- Neural networks are inspired by attempts at modeling a neuron
- Very good for performing the binary classification problems
  - An object can either be of one class or not
- Simplest and least complex neural network
  - Perceptron

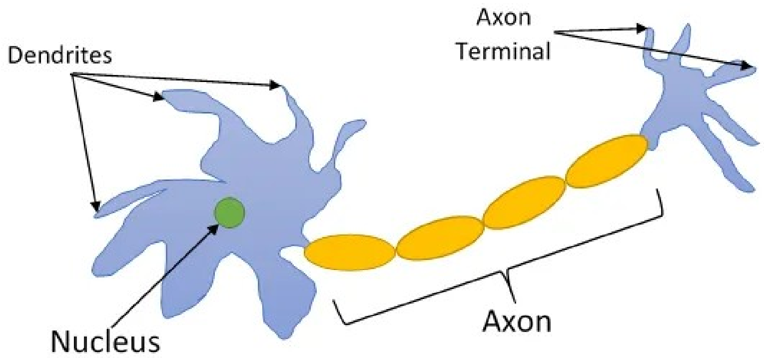
Perceptron

- Conceived in 1958, the Perceptron is one of the longest-lived machine learning algorithms that we are most likely to be aware of and have available in our modern data science and machine learning tool kits.

- Usability of the Perceptron
  - Solving the linearly separable problems
  - In fact the AI winter (stagnation in areas of the AI field)
  - This is only discovered after AI winter.
    - two major winters approximately 1974–1980 and 1987–2000
    - https://en.wikipedia.org/wiki/AI_winter

Perceptron

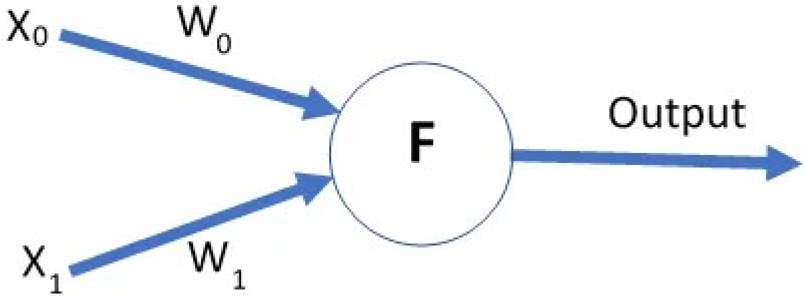
- Receives signals from its Dendrites
- Based on those inputs it decides to pass or not to pass the signal (based on the strength) via. Axon
- One Perceptron is connected to another to form a network



- These neurons form a complex connected network that is capable of solving various problem

Modeling Perceptron

- A Perceptron is a mathematical representation of a Neuron and would look something like this
  - The Perceptron receives a series of signals (the X values) and uses them inside a function (F) to decide what the Output is.
  - Because we want to tune the output, we need to give a weight to each of the input signals (W) to get the best output.
  - The value that is given to the function is simply the weights multiplied by the signal and summed all together.

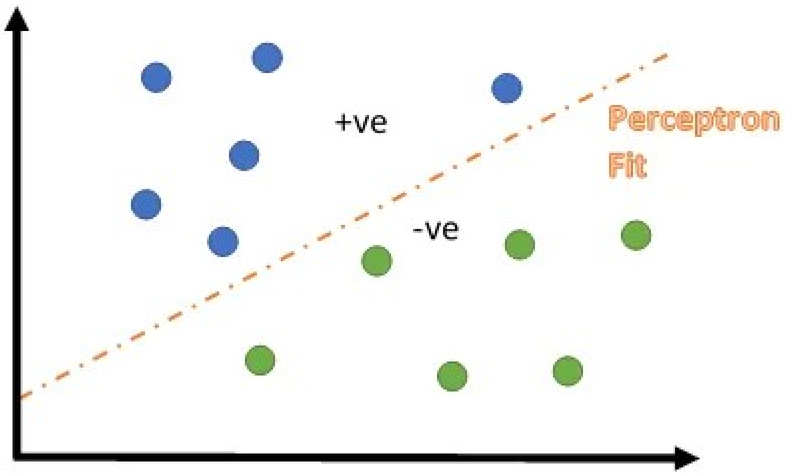


Bias

- Bias is something that can be added to Perceptron's to improve their behaviour
  - often seen by improvements in accuracy prediction
- If the input features are all zero then the Perceptron can only output a zero.
- Adding a bias enables this behavior to be different.

Example

- We have green and blue points
  - We want the Perceptron to be able to tell us which is which, based on where they are on the graph.
- To do this it draws a straight line and anything on one side belongs to one group and the ones on the other side the second group.
- The decision is based on the sign (+ve or -ve)
  Activation Function

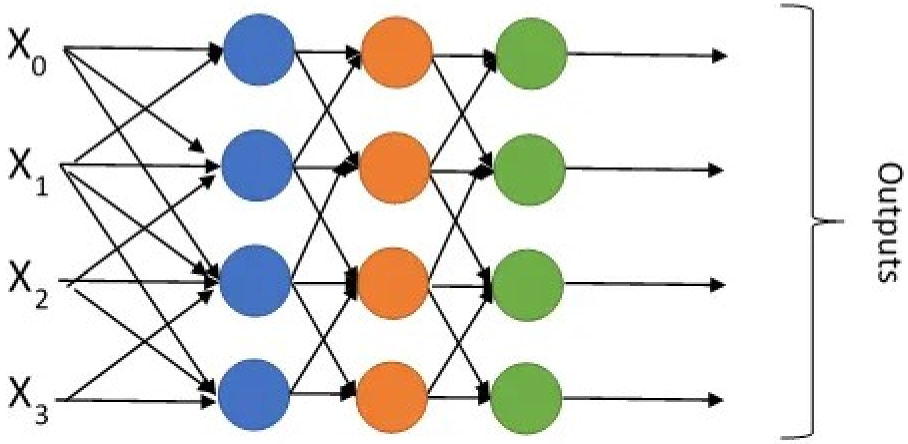


Issue of Being Linearly Separable

- A Perceptron can only separate the groups perfectly if it can draw a straight line.
  - If there are multiple groups or they are intermixed slightly then it cannot separate them.
- This is one of the problems the Perceptron has as it can only separate things into two groups and they must be well separated for it to work perfectly.

Multilayer Perceptron

- It is worth noting that a lot of the issues with a Perceptron can be solved by chaining them together.
  - Inputs feed into several Perceptron’s, the outputs of these feed into another layer of Perceptron’s and so on until we get a final layer that gives us our final output. This is called a Multilayer Perceptron



Understanding together

- The signals travel one way in the Perceptron and a single output is given
  - Feed Forward Algorithm
- A set of inputs we labeled X0, X1 etc.
- There can be as many inputs as you like to the Perceptron
- Each input has a weight (W0, W1 etc.)

Understanding together

- The weighted inputs are fed into an Activation Function (e.g. the sign function we were using)

- Adjusting the weights enables the Perceptron to adjust its answers if it gets the answers wrong by changing the weighting across the inputs

- This adjustment of weights is called Gradient Descent where we minimise the error on the function

Understanding together

- The activation function can be changed to alter the behaviour of the Perceptron’s outputs
- A bias can also be added to the Perceptron inputs to improve behaviour

Implementation of the Neural Network

- The coding Train
  - Neural Networks - The Nature of Code
  - https://www.youtube.com/playlist?list=PLRqwX-V7Uu6aCibgK1PTWWu9by6XFdCfh
  -
-

Thank you