Unit 1. Introduction to machine learning

Estimated time

01:15

Overview

This unit recaps the main topics in Module I, AI overview and provides a deeper view into complex subjects, such as:

- Machine learning
- Machine learning algorithms
- Neural networks
- Deep learning

IBM Training

Unit objectives

- Explain what is machine learning.
- Describe what is meant by statistical model and algorithm.
- Describe data and data types.
- Describe machine learning types and approaches (Supervised, Unsupervised and Reinforcement).
- List different machine learning algorithms.
- Explain what neural networks and deep learning are, and why they are important in today's AI field.
- Describe machine learning components.
- List the steps in the process to build machine learning applications.
- Explain what domain adaptation is and its applications.

Figure 1-1. Unit objectives

1.1. What is machine learning?

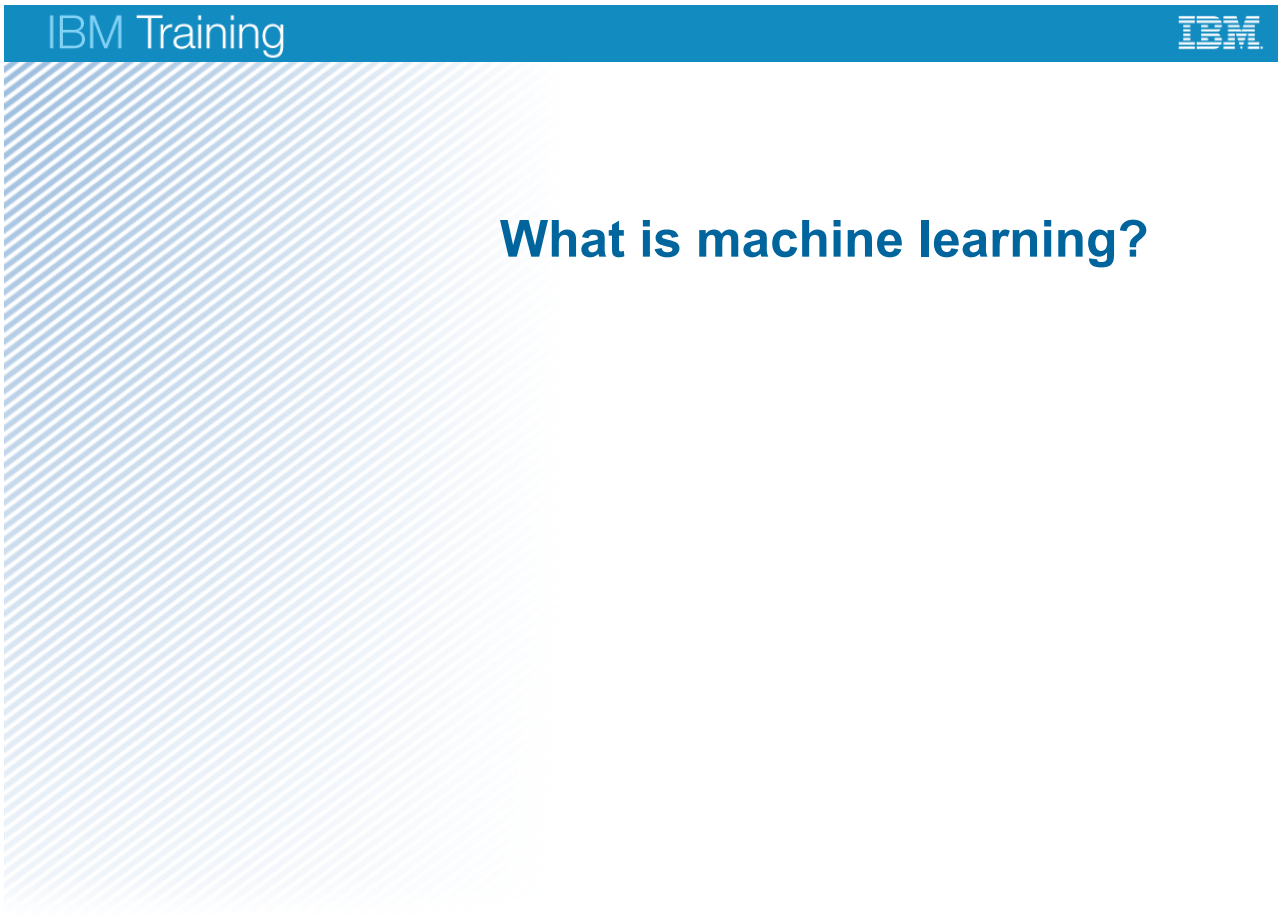


Figure 1-2. What is machine learning?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Figure 1-3. Topics

IBM Training

Machine learning

- In 1959, the term “machine learning” was first introduced by Arthur Samuel. He defined it as the “*field of study that gives computers the ability to learn without being explicitly programmed*”.
- The learning process improves the machine ***model*** over time by using training data.
- The evolved model is used to make future predictions.

Figure 1-4. Machine learning

Arthur Samuel, former IBM engineer and a professor at Stanford, was one of the pioneers in the field of computer gaming and artificial intelligence. He was the first one to introduce the term “machine learning”. Machine learning is a field of artificial intelligence. It uses statistical methods to give computer the ability to "learn" from data, without being explicitly programmed.

If a computer program can improve how it performs certain tasks based on past experiences, then it has learned. This differs from performing the task always the same way because it has been programmed to do so.

The learning process improves the so-called “model” over time by using different data points (training data). The evolved model is used to make future predictions.

References:

https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_1
https://link.springer.com/chapter/10.1007/978-94-009-0279-4_9

IBM Training

What is a statistical model

- A model in a computer is a mathematical function that represents a relationship or mapping between a set of inputs and a set of outputs.

$$f(x)=x^2$$

$$\text{Violent crime incidents per day} = \text{Average Temperature} \times 2$$

- New data “X” can predict the output “Y”.

$$Y = b0 \times X + b1$$

Introduction to machine learning

Figure 1-5. What is a statistical model

The representation of a model in the computer is in the form of a mathematical function. It is a relationship or mapping between a set of inputs and a set of outputs. For example, f(x)=x^2.

Assume that a system is fed with data indicating that the rates of violent crime are higher when the weather is warmer and more pleasant, even rising sharply during warmer-than-typical winter days. Then, this model can predict the crime rate for this year compared to last year’s rates based on the weather forecast.

Returning to the mathematical representation of the model that can predict crime rate based on temperature, we might propose the following mathematical model:

Violent crime incidents per day = Average Temperature × 2

This is an oversimplified example to explain that machine learning refers to a set of techniques for estimating functions (for example, predicting crime incidents) that is based on data sets (pairs of the day’s average temperature and the associated number of crime incidents). These models can be used for predictions of future data.

1.2. Machine learning algorithms

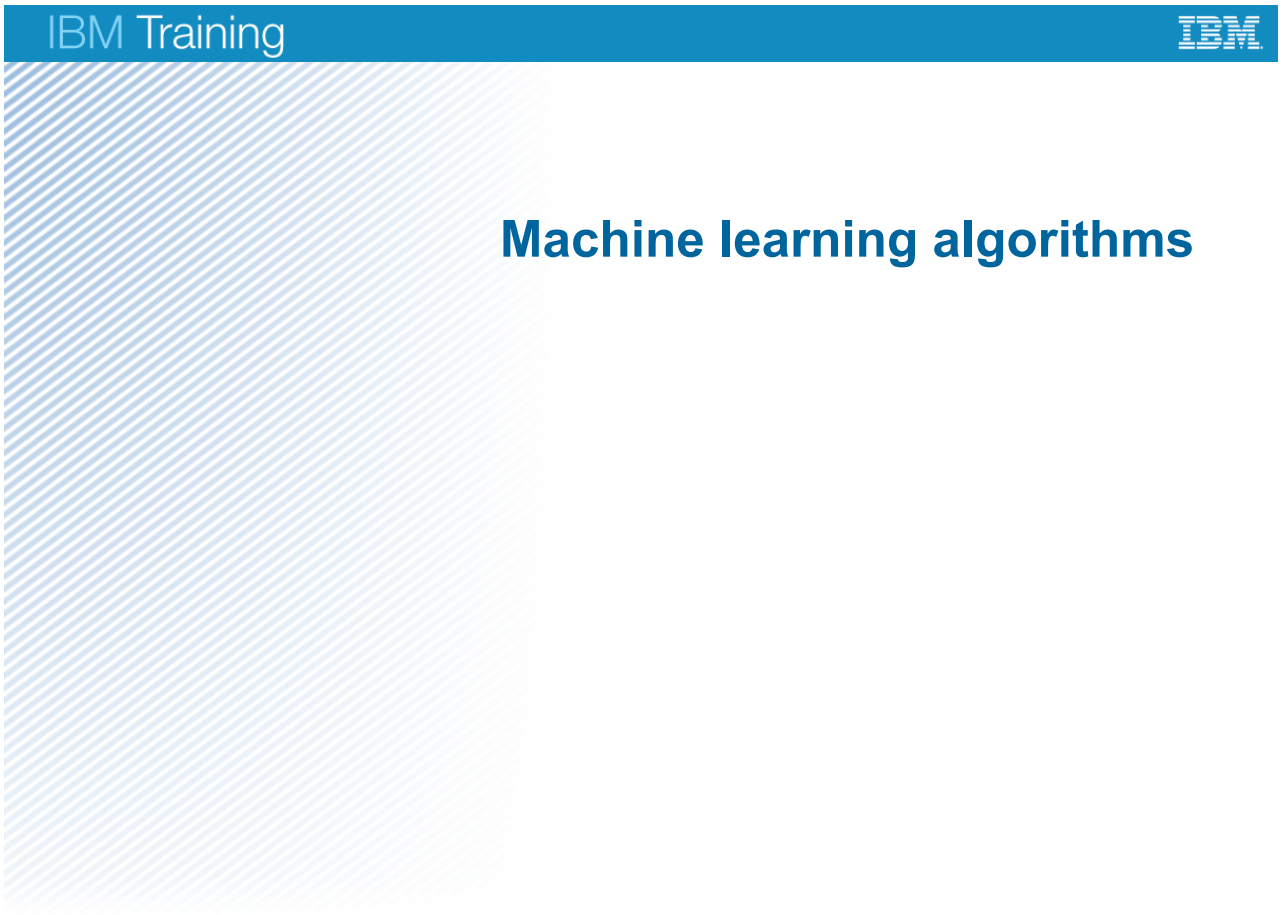


Figure 1-6. Machine learning algorithms

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Figure 1-7. Topics

IBM Training

Machine learning algorithms

- The machine learning algorithm is a technique through which the system extracts useful patterns from historical data. These patterns can be applied to new data.
- The objective is to have the system learn a specific input/output transformation.
- The data quality is critical to the accuracy of the machine learning results.

Figure 1-8. Machine learning algorithms

To estimate the function that represents the model, an appropriate learning algorithm must be used. In this context, the learning algorithm represents the technique through which the system extracts useful patterns from the input historical data. These patterns can be applied to new data in new situations. The objective is to have the system learn a specific input/output transformation and to make future predictions for a new data point. Finding the appropriate algorithms to solve complex problems in various domains and knowing how and when to apply them is an important skill that machine learning engineers should acquire. Because the machine learning algorithms depend on data, understanding and acquiring data with high quality is crucial for accurate results.

IBM Training

Machine learning approaches

1) Supervised learning: Train by using labeled data, and learn and predict new labels for unseen input data.

- Classification is the task of predicting a discrete class label, such as “black, white, or gray” and “tumor or not tumor”.
- Regression is the task of predicting a continuous quantity, such as “weight”, “probability”, and “cost”.

Figure 1-9. Machine learning approaches

Supervised learning is one of the main categories of machine learning. In supervised machine learning, input data (also known as training examples) comes with a label, and the goal of learning is to predict the label for new, unforeseen examples. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

In practice, the problems that are solved by using supervised learning are grouped into either regression or classification problems.

Classification is the task of predicting a discrete class label, such as “black, white, or gray” and “tumor or not tumor”.

Regression is the task of predicting a continuous quantity, such as “weight”, “probability” and “cost”.

IBM Training

Machine learning approaches (cont.)

2) **Unsupervised learning:** Detect patterns and relationships between data without using labeled data.

- **Clustering algorithms:** Discover how to split the data set into a number of groups such that the data points in the same groups are more similar to each other compared to data points in other groups.

Figure 1-10. Machine learning approaches (cont.)

Unsupervised learning is a machine learning type that learns from data that has not been labeled. The goal of unsupervised learning is to detect patterns in the data. One of the most popular types of unsupervised learning is clustering algorithms.

Clustering algorithms are algorithms that discover how to split the data set into a number of groups such that the data points in the same groups are more similar to each other compared to data points in other groups.

IBM Training

Machine learning approaches (cont.)

3) **Semi-supervised learning:**

- A machine learning technique that falls between supervised and unsupervised learning.
- It includes some labeled data with a large amount of unlabeled data.
- Here is an example that uses pseudo-labeling:
  a. Use labeled data to train a model.
  b. Use the model to predict labels for the unlabeled data.
  c. Use the labeled data and the newly generated labeled data to create a new model.

Figure 1-11. Machine learning approaches (cont.)

Many real practical problems fall into this category of machine learning where you have little labeled data and the rest of the data is unlabeled.

Labeling data is an expensive or time-consuming process. In addition, it mandates having domain experts to label data accurately. Think about labeling skin diseases images that must be labeled by a domain expert. Also, too much labeling data might introduce human biases into the model.

In semi-supervised learning, you try to get the best out of your unlabeled data. There are different techniques to achieve this task. For example, you can use pseudo-labeling, which aims to give approximate labels to unlabeled data. Pseudo-labeling works as follows:

1. Use labeled data to train a model.
2. Use the model to predict labels for the unlabeled data.
3. Use the labeled data and the newly generated labeled data to create a model.

References:

http://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf

https://www.analyticsvidhya.com/blog/2017/09/pseudo-labelling-semi-supervised-learning-technique/

IBM Training

Machine learning approaches (cont.)

4) Reinforcement learning

- Reinforcement learning uses trial and error (a rewarding approach).
- The algorithm discovers an association between the goal and the sequence of events that leads to a successful outcome.
- Example reinforcement learning applications:
  - Robotics: A robot that must find its way.
  - Self-driving cars.

Figure 1-12. Machine learning approaches (cont.)

Reinforcement learning is a goal-oriented learning that is based on interaction with the environment. As the system performs certain actions, it finds out more about the world. Reinforcement learns through trial and error (a rewarding approach).

The algorithm discovers an association between the goal and the sequence of events that leads to a successful outcome.

Example reinforcement learning problems:

- Robotics: A robot that must find its way.
- Self-driving cars.

IBM Training

Machine learning algorithms

Understanding your problem and the different types of ML algorithms helps in selecting the best algorithm.

Here are some machine learning algorithms:

- Naïve Bayes classification (supervised classification – probabilistic)
- Linear regression (supervised regression)
- Logistic regression (supervised classification)
- Support vector machine (SVM) (supervised linear or non-linear classification)
- Decision tree (supervised non-linear classification)
- K-means clustering (unsupervised learning)

Figure 1-13. Machine learning algorithms

In the following slides, we explore different machine learning algorithms. We describe the most prominent algorithms. Each algorithm belongs to a category of learning. We explore supervised and unsupervised algorithms, regression and classification algorithms, and linear and non-linear classification.

IBM Training

Naïve Bayes classification

- Naïve Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.
  - For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.
  - Features: Color, roundness, and diameter.
  - Assumption: Each of these features contributes independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

Figure 1-14. Naïve Bayes classification

Naïve Bayes classifiers is a powerful and simple supervised machine learning algorithm. It assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.

Features: Color, roundness, and diameter.

A Naïve Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

Naïve Bayes classification (cont.)

Example: Use Naïve Bayes to predict whether the Red, Round shaped, 10 cm diameter label is an apple or not.

Sample No	Color	Shape	Diameter	Is Apple?
1	Red	Round	>= 10 CM	Yes
2	Red	Round	>= 10 CM	No
3	Red	Round	>= 10 CM	Yes
4	Yellow	Round	>= 10 CM	No
5	Yellow	Round	< 10 CM	Yes
6	Yellow	Cylinder	< 10 CM	No
7	Yellow	Cylinder	< 10 CM	Yes
8	Yellow	Cylinder	>= 10 CM	No
9	Red	Cylinder	< 10 CM	No
10	Red	Round	< 10 CM	Yes

Figure 1-15. Naïve Bayes classification (cont.)

Imagine that you have the data set that is shown in the table in this slide. The column with title “Is Apple?” represents the label of the data. Our objective is to make a new prediction for an unknown object. The unknown object has the following features:

- Color: Red
- Shape: Round
- Diameter: 10 cm

Note 1: Sometimes the terminology “parameters” or “variables” is used to describe the “features”.

Note 2: “Annotated data” or “labeled data” refer to the same terminology.

IBM Training

Naïve Bayes classification (cont.)

To do a classification, you must perform the following steps:

1. Define two classes (C_Y and C_N) that correspond to Apple = Yes and Apple = No.
2. Compute the probability for C_Y as **x:** $p(C_Y \mid \mathbf{x})$:
   p(Apple = Yes | Colour = Red, Shape = round, Diameter => 10 cm)
3. Compute the probability for C_N as **x:** $p(C_N \mid \mathbf{x})$:
   p(Apple = No | Colour = Red, Shape = round, Diameter => 10 cm)
4. Discover which conditional probability is larger:
   If $p(C\mathbf{Y} \mid \mathbf{x}) > p(C\mathbf{N} \mid \mathbf{x})$, then it is an apple.

Figure 1-16. Naïve Bayes classification (cont.)

Your algorithm basically depends on calculating two probability values:

- **Class probabilities**: The probabilities of having each class in the training data set.
- **Conditional probabilities**: The probabilities of each input feature giving a specific class value.

The process for solving this problem is as follows:

1. Define two classes C**Y** and C**N** that correspond to Apple = Yes and Apple = No.
2. Compute the probability for C**Y** as **x:** $p(C\mathbf{Y} \mid \mathbf{x})$: *p(Apple = Yes | Colour = Red, Shape = round, Diameter => 10 cm)*
3. Compute the probability for C**N** as **x:** $p(C\mathbf{N} \mid \mathbf{x})$: *p(Apple = No | Colour = Red, Shape = round, Diameter => 10 cm)*
4. Discover which conditional probability is larger: If $p(C\mathbf{Y} \mid \mathbf{x}) > p(C\mathbf{N} \mid \mathbf{x})$, then it is an apple.

IBM Training

Naïve Bayes classification (cont.)

Naïve Bayes model: $p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$

5. Compute $p(\mathbf{x}|C\mathbf{Y})$ = p(Colour = Red, Shape = round, Diameter =>10 cm | Apple = Yes).

   Naïve Bayes assumes that the features of the input data (the apple parameters) are independent.

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{D} p(x_i|C_k)$$

Introduction to machine learning

Figure 1-17. Naïve Bayes classification (cont.)

The Naïve Bayes formula is given by this model. Our target is to compute the formula to reach $p(C\mathbf{K}|\mathbf{x})$, where K is any class (C**Y** or C**N**).

5. Compute the conditional probability of having each feature given that the class is C**Y**: $p(\mathbf{x}|C\mathbf{Y})$ = p(Colour = Red, Shape = round, Diameter =>10 cm | Apple = Yes).

Because Naïve Bayes assumes that the features of the input data (the object features) are independent, to get the $p(\mathbf{x}|C\mathbf{Y})$ value, we calculate the conditional probability of each feature at a time with the class CY, and then multiply all the values.

IBM Training

Naïve Bayes classification (cont.)

Thus, we can rewrite $p(\mathbf{x}| C\mathbf{Y})$ as:

= p(Colour = Red | Apple = Yes) X p(Shape = round | Apple = Yes) X

p(Diameter => 10 cm | Apple = Yes)

Same for $p(\mathbf{x}| C\mathbf{N})$:

= p(Color = Red | Apple = No) X p(Shape = round | Apple = No) X

p(Diameter => 10 cm | Apple = No)

Figure 1-18. Naïve Bayes classification (cont.)

Thus, we can rewrite $p(\mathbf{x}| C\mathbf{Y})$ as:

= p(Colour = Red | Apple = Yes) X p(Shape = round | Apple = Yes) X p(Diameter => 10 cm | Apple = Yes)

We apply the same rule for $p(\mathbf{x}| C\mathbf{N})$ by multiplying the conditional probabilities of reach input feature given C**N**:

= p(Color = Red | Apple = No) X p(Shape = round | Apple = No) X p(Diameter => 10 cm | Apple = No)

Naïve Bayes classification (cont.)

6. Calculate each conditional probability:

p(Colour = Red | Apple = Yes) = 3/5 (Out of five apples, three of them were red.)
p(Colour = Red | Apple = No) = 2/5
p(Shape = Round | Apple = Yes) = 4/5
p(Shape = Round | *Apple = No) = 2/5*
p(Diameter = > 10 cm | Apple = Yes) = 2/5
p(Diameter = > 10 cm | Apple = No) = 3/5

Color	Shape	Diameter	Is Apple?
Red	Round	>= 10 CM	Yes
Red	Round	>= 10 CM	No
Red	Round	>= 10 CM	Yes
Yellow	Round	>= 10 CM	No
Yellow	Round	< 10 CM	Yes
Yellow	Cylinder	< 10 CM	No
Yellow	Cylinder	< 10 CM	Yes
Yellow	Cylinder	>= 10 CM	No
Red	Cylinder	< 10 CM	No
Red	Round	< 10 CM	Yes

Figure 1-19. Naïve Bayes classification (cont.)

Let us see how to calculate these conditional probabilities. For example, to calculate *p*(Colour = Red | Apple = Yes), you are asking, “What is the probability for having a red color object given that we know that it is an apple”.

You browse the table to see how many “is Apple?” has a “yes” label. You see that the occurrence is five times.

Now, from the table, how many of these five occurrences are when you have a color = red? You find that there are three occurrences for red color. Therefore, *p*(Colour = Red | Apple = Yes) = 3/5.

Repeat these steps for the rest of the features.

IBM Training

Naïve Bayes classification (cont.)

- *p*(Color = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X *p*(Diameter = > 10 cm | Apple = Yes)

  = (3/5) x (4/5) x (2/5) = 0.192

- *p*(Color = Red | Apple = No) X *p*(Shape = round | Apple = No) X *p*(Diameter = > 10 cm | Apple = No)

  = (2/5) x (2/5) x (3/5) = 0.096

- *p*(Apple = Yes) = 5/10
- *p*(Apple = No) = 5/10

Figure 1-20. Naïve Bayes classification (cont.)

Now, we have all the values that we need. As mentioned in step 5, we multiply the conditional probabilities as follows:

p(Color = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X *p*(Diameter = > 10 cm | Apple = Yes)

= (3/5) x (4/5) x (2/5) = 0.192

p(Color = Red | Apple = No)*p*(Shape = round | Apple = No)*p*(Diameter = > 10 cm | Apple = No)

= (2/5) x (2/5) x (3/5) = 0.096

p(Apple = Yes) = 5/10

p(Apple = No) = 5/10

IBM Training

Naïve Bayes classification (cont.)

Compare p(C**Y** | **x**) to p(CN | **x**):

$$\text{If } \frac{p(C_Y|\mathbf{x})}{p(C_N|\mathbf{x})} > 1 \therefore \mathbf{x} \in C_Y \text{ , else } \mathbf{x} \in C_N$$

$$\frac{p(C_Y|\mathbf{x})}{p(C_N|\mathbf{x})} = \frac{p(\mathbf{x}|C_Y)p(C_Y)}{p(\mathbf{x}|C_N)p(C_N)} = \frac{0.192 \times 0.5}{0.096 \times 0.5} = 2$$

Therefore, the verdict is that it is an apple.

Figure 1-21. Naïve Bayes classification (cont.)

Finally, we compare the values of *p*(C**Y** |**x**) versus *p(*C**N** |**x**). By substituting the values that were calculated in the previous steps, we discover that *p*(C**Y** |**x**) > *p(*C**N** |**x**), which means that the object is an apple.

IBM Training

Linear regression

- Linear regression is a linear equation that combines a specific set of input values (X) and an outcome (Y) that is the predicted output for that set of input values. As such, both the input and output values are numeric.
- The target variable is a continuous value.

Examples for applications:

- Analyze the marketing effectiveness, pricing, and promotions on the sales of a product.
- Forecast sales by analyzing the monthly company's sales for the past few years.
- Predict house prices with an increase in the sizes of houses.
- Calculate causal relationships between parameters in biological systems.

Introduction to machine learning

Figure 1-22. Linear regression

Regression algorithms are one of the key algorithms that are used in machine learning. Regression algorithms help analysts to model relationships between input variables X and the output label Y for the training data points. This algorithm targets supervised regression problems, that is, the target variable is a continuous value.

In simple linear regression, we establish a relationship between the target variable and input variables by fitting a line that is known as the regression line.

There are different applications that benefit from linear regression:

- Analyze the marketing effectiveness, pricing, and promotions on the sales of a product.
- Forecast sales by analyzing the monthly company's sales for the past few years.
- Predict house prices with an increase in the sizes of houses.
- Calculate causal relationships between parameters in biological systems.

Linear regression (cont.)

- Example: Assume that we are studying the real state market.
- Objective: Predict the price of a house given its size by using previous data.

Size	Price
30	30,000
70	40,000
90	55,000
110	60,000
130	80,000
150	90,000
180	95,000
190	110,000

Figure 1-23. Linear regression (cont.)

Assume that we are studying the real state market and our objective is to predict the price of a house given its size by using previous data. The label in this case is the price column.

Linear regression (cont.)

Plot this data as a graph

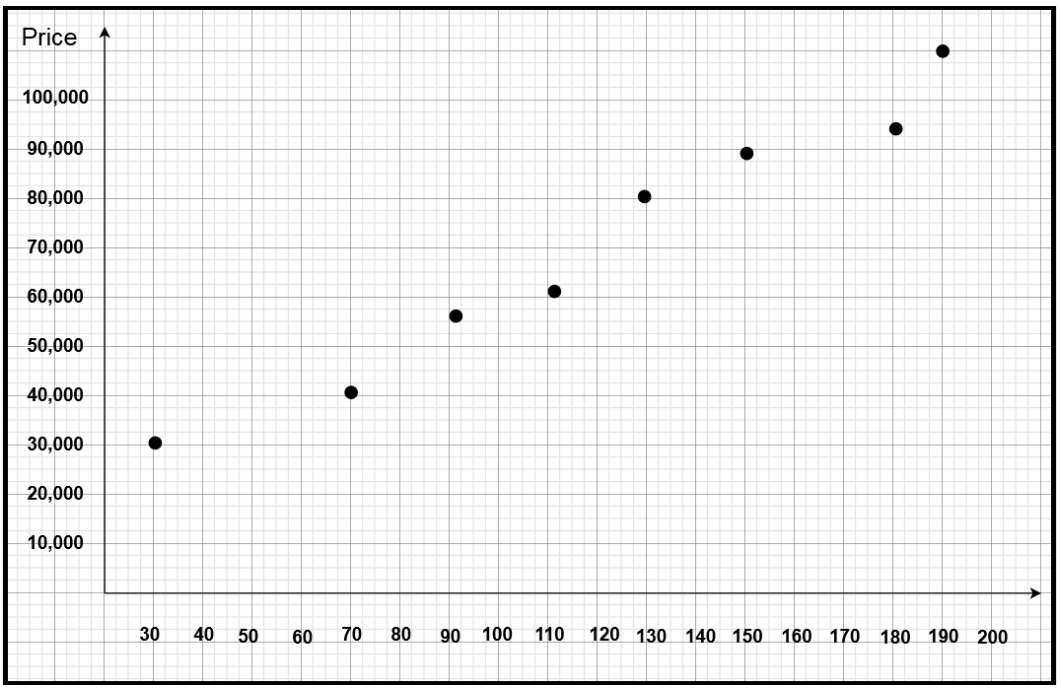


Figure 1-24. Linear regression (cont.)

After plotting the points on the graph, they seem to be forming a line.

IBM Training

Linear regression (cont.)

- Can you guess what is the best estimate for a price of a 140-meter square house?
- Which one is correct?

A. $60,000
B. $95,000
C. $85,000

Size	Price
30	30,000
70	40,000
90	55,000
110	60,000
130	80,000
150	90,000
180	95,000
190	110,000

Introduction to machine learning

Figure 1-25. Linear regression (cont.)

You want to find the price value of a 140-meter square house. Which of the following choices is correct?

1. $60,000
2. $95,000
3. $85,000

Linear regression (cont.)

- **Target**: A line that is within a “proper” distance from all points.
- **Error**: The aggregated distance between data points and the assumed line.
- **Solution**: Calculate the error iteratively until you reach the most accurate line with a minimum error value (that is, the minimum distance between the line and all points).

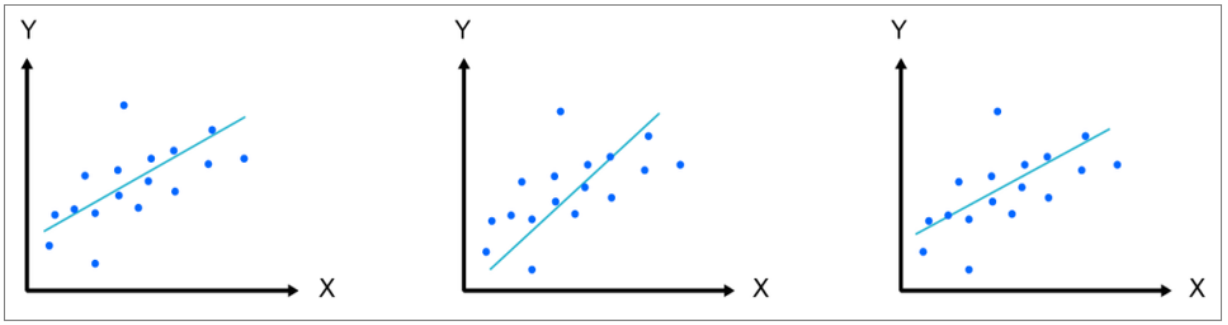


Figure 1-26. Linear regression (cont.)

To answer the question “What is the price for a 140-meter square house?”, we need to draw the line that best fits most of the data points.

How we can find the line that best fits all the data points? We can draw many lines, so which one is the best line?

The best line should have the minimal error value. The error refers to the aggregated distance between data points and the assumed line. Calculate the error iteratively until you reach the most accurate line with a minimum error value.

Linear regression (cont.)

- After the learning process, you get the most accurate line, the bias, and the slope to draw your line.
- Here is our linear regression model representation for this problem:

$$h(p) = p0 + p1 * X1$$

or

$$\text{Price} = 30{,}000 + 392 * \text{Size}$$

$$\text{Price} = 30{,}000 + 392 * 140$$

$$= 85{,}000$$

Figure 1-27. Linear regression (cont.)

After the learning process, you get the most accurate line, the bias, and the slope to draw your line.

p_0 is the bias. It is also called the intercept because it determines where the line intercepts the y axis.

p_1 is the slope because it defines the slope of the line or how x correlates with a y value before adding the bias.

If you have the optimum value of p_0 and p_1, you can draw the line that best represents the data.

IBM Training

Linear regression (cont.)

- Squared error function → $J(\mathsf{P}) = \frac{1}{2m}\sum_{i=1}^{m}(h_{\mathsf{P}}(x^{(i)}) - y^{(i)})^2$
  - m is the number of samples.
  - $h_{\mathsf{P}}(x^{(i)})$ is the predicted value for data point i.
  - $y^{(i)}$ is the actual value for data point i.

Target: Choose P values to minimize errors.

- Stochastic Gradient descent algorithm:

$$\mathbf{P}_j := \mathbf{P}_j - \alpha \left(h_{\mathbf{P}}(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

j is the feature number.

α is the learning rate.

Figure 1-28. Linear regression (cont.)

The squared error function J is represented by the difference between the predicted point and the actual points. It is calculated as follows:

$J(P) = (1/(2*m)) \Sigma (h_p(x^i) - yi)^2$

Where:

- i is the number of a sample or data point within the data set samples.
- $h_p(x^i)$ is the predicted value for data point i.
- y^i is the actual value for data point i.
- m is the count of data set samples or data points.

We can use an optimization technique that is called **stochastic gradient descent**. The algorithm evaluates and updates the weights on every iteration to minimize the model error. The technique works iteratively. In each iteration, the training instance is exposed to the model once. The model makes a prediction and the corresponding error is calculated. The model is updated to reduce the error for the next prediction. The process continues to adjust the model weights to reach the smallest error.

Here we use the gradient descent algorithm to iteratively get the values of p0 and p1 (the intercept and slope of the line are also called weights) by the following algorithm:

$Pj := Pj – \alpha (h_p(x^i) - yi) x_j^i$

Where:

j is the feature number.

α is the learning rate.

Linear regression (cont.)

- In higher dimensions where we have more than one input (X), the line is called a plane or a hyper-plane.
- The equation can be generalized from simple linear regression to multiple linear regression as follows:

  Y(X)=p0+p1*X1+p2*X2+...+pn*Xn

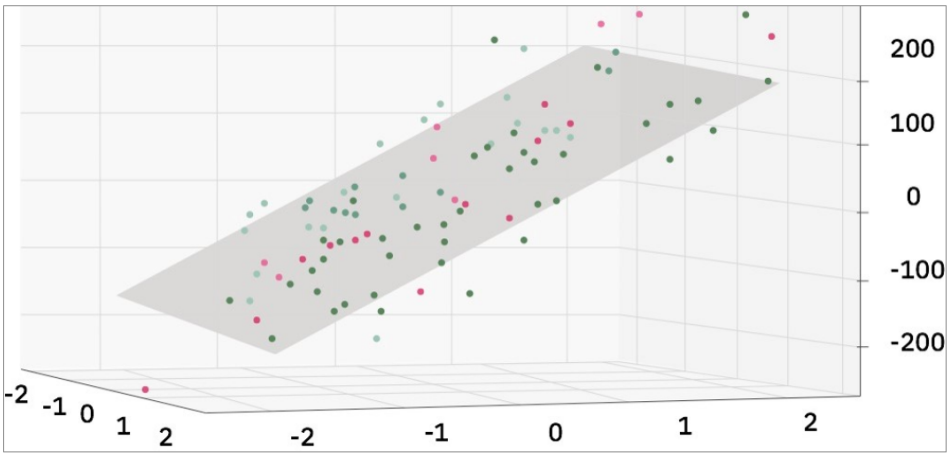


Figure 1-29. Linear regression (cont.)

With more features, you do not have a line; instead, you have a plane. In higher dimensions where we have more than one input (X), the line is called a plane or a hyper-plane.

The equation can be generalized from simple linear regression to multiple linear regression as follows:

Y(X)=p0+p1*X1+p2*X2+...+pn*Xn

IBM Training

Logistic regression

- Supervised classification algorithm.
- **Target**: A dependent variable (Y) is a discrete category or a class (not a continuous variable as in linear regression).

  Example: Class1 = Cancer, Class2 = No Cancer

Figure 1-30. Logistic regression

Logistic regression is a supervised classification algorithm. It is different from linear regression where the dependent or output variable is a category or class. The target is a discrete category or a class (not a continuous variable as in linear regression), for example, Class1 = cancer, Class2 = No Cancer.

IBM Training

Logistic regression (cont.)

- Logistic regression is named for the function that is used at the core of the algorithm.
- The logistic function (sigmoid function) is an S-shaped curve for data discrimination across multiple classes. It can take any real value 0 – 1.

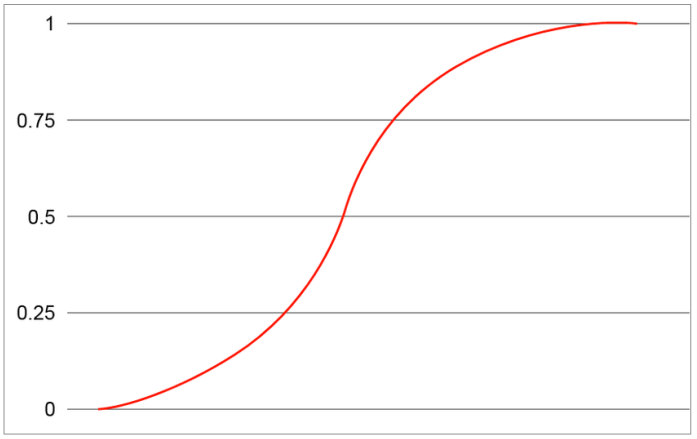


Figure 1-31. Logistic regression (cont.)

Logistic regression is named for the function that is used at the core of the algorithm, which is the logistic function. The logistic function is also known as the sigmoid function. It is an S-shaped curve (as shown in the figure) for data segregation across multiple classes that can take any real value 0 - 1.

IBM Training

Logistic regression (cont.)

- The sigmoid function squeezes the input value between [0,1].
- Logistic regression equation:

$$Y = \exp(p0+p1X)/(1+\exp(p0+p1X))$$

$$\mathbf{h}(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Figure 1-32. Logistic regression (cont.)

During the learning process, the system tries to generate a model (estimate a set of parameters p0, p1, …) that can best predict the probability that Y will fall in class A or B given the input X. The sigmoid function squeezes the input value between [0,1], so if the output is 0.77 it is closer to 1, and the predicted class is 1.

IBM Training

Logistic regression (cont.)

- Example: Assume that the estimated values of p's for a certain model that predicts the gender from a person's height are p0= -120 and p1 = 0.5.
- Class 0 represents female and class 1 represents male.
- To compute the prediction, use:
  Y = exp(-120+0.5X)/(1+exp(-120+0.5X))
  Y = 0.00004539

  P(male|height=150) is 0 in this case.

Figure 1-33. Logistic regression (cont.)

Example: Assume that the estimated values of p's for a certain model that predicts the gender from a person's height are p0= -120 and p1=0.5.

Assume that you have two classes where class 0 represents female and class 1 represents male.

Y = exp(-120+0.5X)/(1+exp(-120+0.5X))

Y = 0.00004539

P(male|height=150) is 0 in this case.

IBM Training

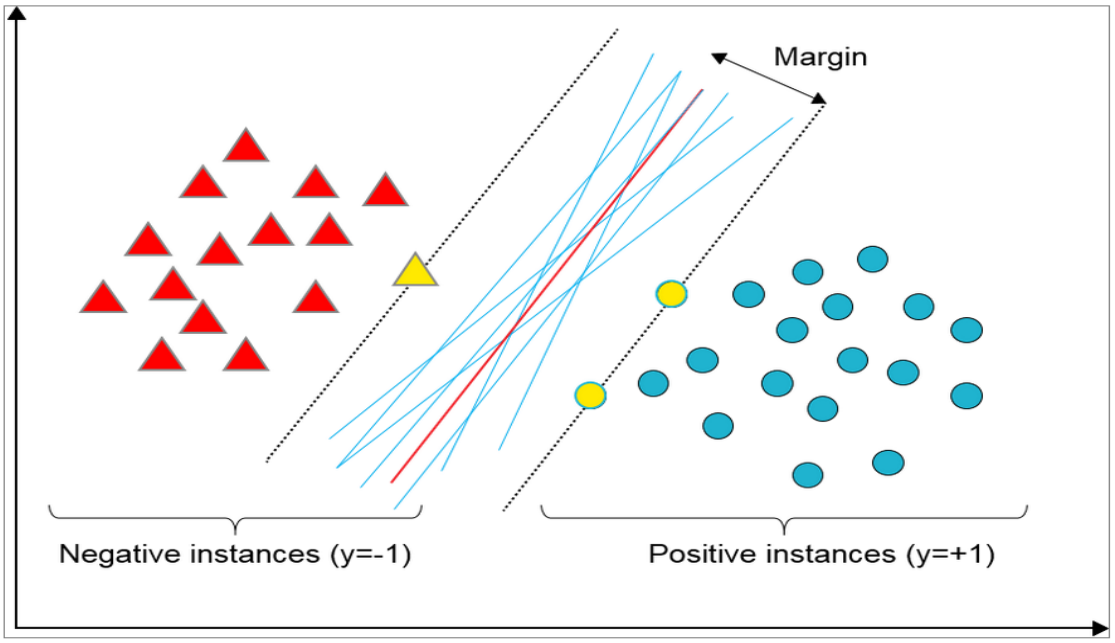
Support vector machine

- The goal is to find a separating hyperplane between positive and negative examples of input data.
- SVM is also called a “large Margin Classifier”.
- The SVM algorithm seeks the hyperplane with the largest margin, that is, the largest distance to the nearest sample points.

Figure 1-34. Support vector machine

SVM is a supervised learning model that can be a linear or non-linear classifier. SVM is also called a “large Margin Classifier” because the algorithm seeks the hyperplane with the largest margin, that is, the largest distance to the nearest sample points.

Support vector machine (cont.)



Highlighting the hyperplane with maximum margin with respect to the support vectors

Figure 1-35. Support vector machine (cont.)

Assume that a data set lies in a two-dimensional space and that the hyperplane will be a one-dimensional line.

Although many lines (in light blue) do separate all instances correctly, there is only one optimal hyperplane (red line) that maximizes the distance to the closest points (in yellow).

IBM Training

Decision tree

- A supervised learning algorithm that uses a tree structure to model decisions.
- It resembles a flow-chart or if-else cases.
- An example for applications is general business decision-making like predicting customers' willingness to purchase a given product in a given setting, for example, online versus a physical store.

Figure 1-36. Decision tree

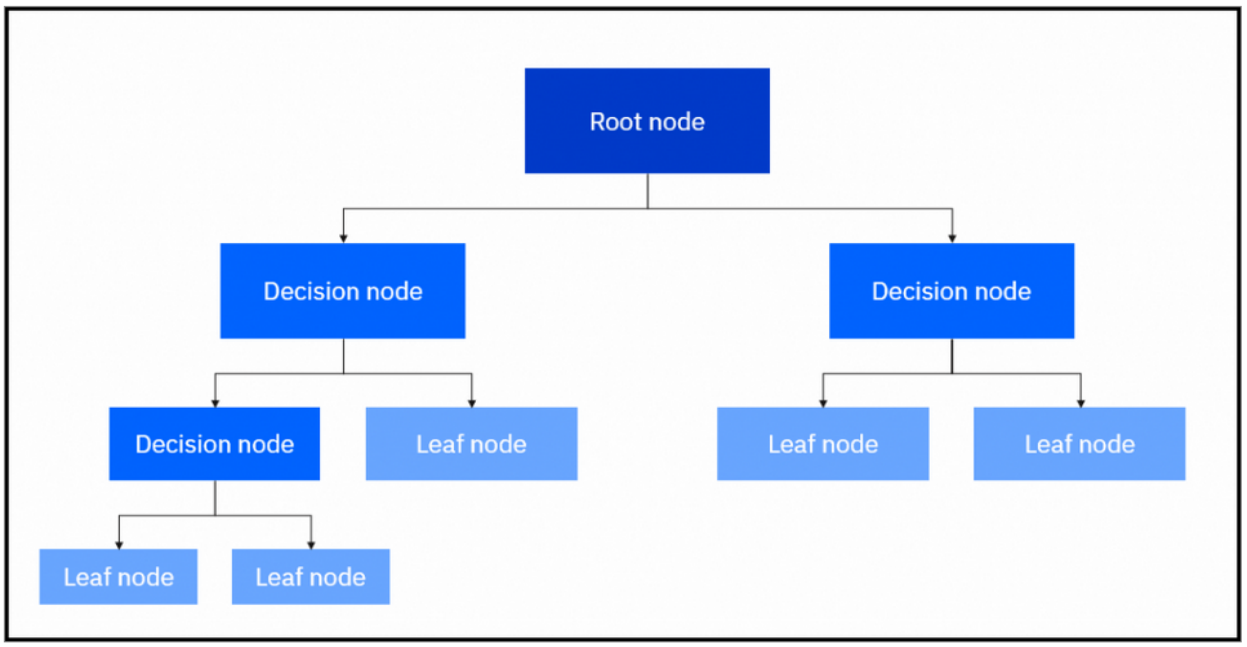
A decision tree is a popular supervised learning algorithm that can be used for classification and regression problems. Decision trees are a popular prediction method. Decision trees can explain why a specific prediction was made by traversing the tree.

There are different examples for applications that can use decision tree in business. For example, predicting customers' willingness to purchase a given product in a given setting, for example, online versus a physical store.

Note

In our scope, we focus on a classification tree.

Decision tree (cont.)



Graphical representation of decision tree machine learning algorithm

Figure 1-37. Decision tree (cont.)

A decision tree includes three main entities: root node, decision nodes, and leaves. The figure shows the graphical representation of these entities.

A decision tree builds the classification or regression model in the form of a tree structure. It resembles a flowchart, and is easy to interpret because it breaks down a data set into smaller and smaller subsets while building the associated decision tree.

Decision tree (cont.)

Play Tennis Example: Data

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 1-38. Decision tree (cont.)

The “Play Tennis” example is one of the most popular examples to explain decision trees.

In the data set, the label is represented by “PlayTennis”. The features are the rest of the columns: “Outlook”, “Temperature”, “Humidity”, and “Wind”. Our goal here is to predict, based on some weather conditions, whether a player can play tennis or not.

Reference:

http://jmvidal.cse.sc.edu/talks/decisiontrees/choosingbest.html?style=White

Decision tree (cont.)

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

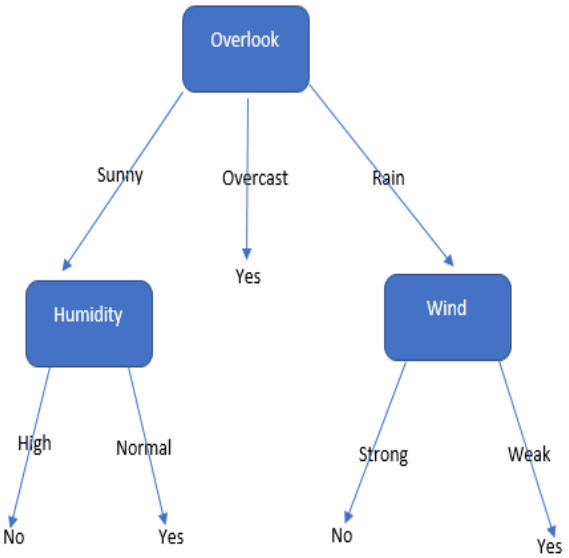


Figure 1-39. Decision tree (cont.)

Back to the example, the decision tree representation on the right side of the figure shows the following information:

- Each internal node tests an attribute.
- Each branch corresponds to an attribute value.
- Each leaf node assigns a classification.

Eventually, we want to make a classification of “if Play Tennis = {Yes, No}”.

Reference:

http://jmvidal.cse.sc.edu/talks/decisiontrees/choosingbest.html?style=White

IBM Training

Decision tree (cont.)

A decision tree is built by making decisions regarding the following items:

- Which feature to choose as the root node
- What conditions to use for splitting
- When to stop splitting

Figure 1-40. Decision tree (cont.)

The algorithm works by recursively splitting the data based on the value of a feature. After each split, the portion of the data becomes more homogeneous.

Now, the algorithm needs to decide:

1. Which feature to choose as the root node.
2. What conditions to use for splitting.
3. When to stop splitting.

IBM Training

Decision tree (cont.)

- Using entropy and information gain to construct a decision tree.
- **Entropy**: It is the measure of the amount of uncertainty and randomness in a set of data for the classification task.
- **Information gain:** It is used for ranking the attributes or features to split at given node in the tree.

  Information gain = (Entropy of distribution before the split)–(entropy of distribution after it)

Figure 1-41. Decision tree (cont.)

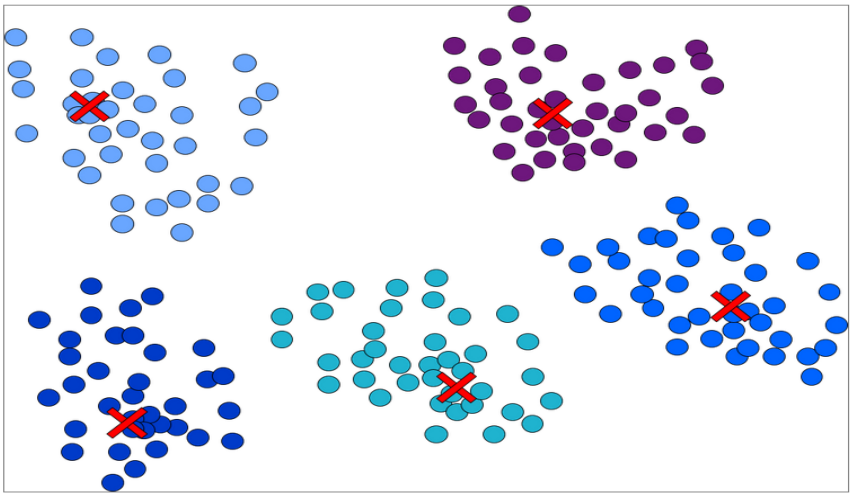
The Iterative Dichotomiser3 (ID3) algorithm works by using entropy and information gain to construct a decision tree. Entropy is the measure of the amount of uncertainty and randomness in a set of data for the classification task. Entropy is maximized when all points have equal probabilities. If entropy is minimal, it means that the attribute or feature appears close to one class and has a good discriminatory power for classification.

Entropy zero means that there is no randomness for this attribute.

Information gain is a metric that is used for ranking the attributes or features to split at given node in the tree. It defines how much information a feature provides about a class. The feature with the highest information gain is used for the first split.

K-mean clustering

- Unsupervised machine learning algorithm.
- It groups a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than those in other groups (other clusters).



Data partitioned into five clusters - Cluster centroids shown as crosses

Figure 1-42. K-mean clustering

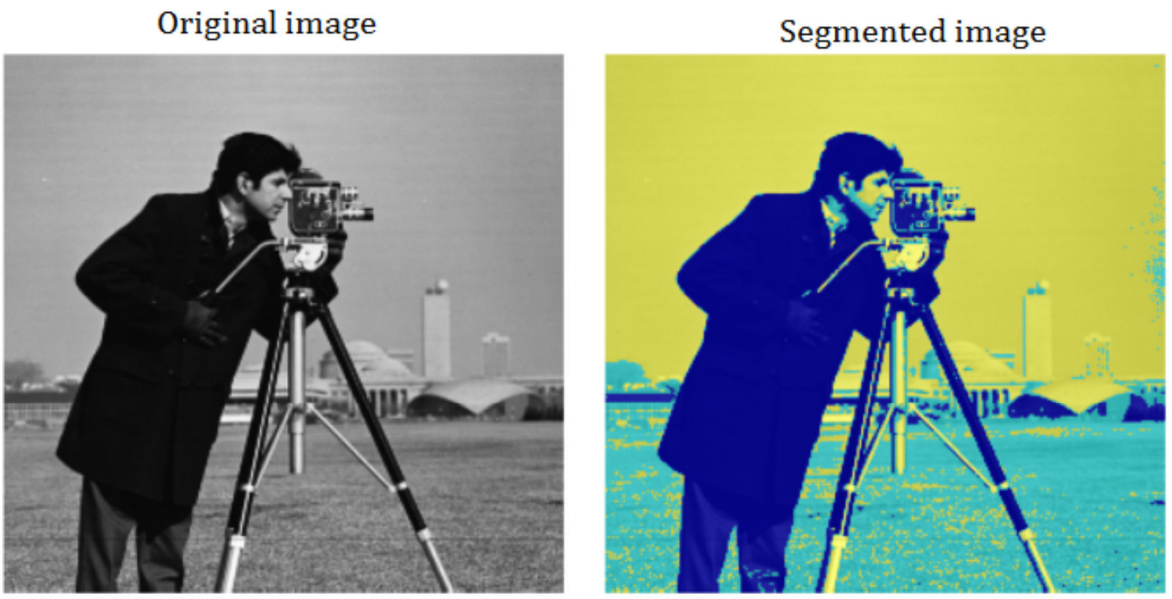
K-means clustering is an unsupervised machine learning technique. The main goal of the algorithm is to group the data observations into k clusters, where each observation belongs to the cluster with the nearest mean.

A cluster's center is the centroid. The figure shown plots of the partition of a data set into five clusters, with the cluster centroids shown as crosses.

IBM Training

K-means clustering (cont.)

Examples for applications include customer segmentation, image segmentation, and recommendation systems.



Introduction to machine learning

Figure 1-43. K-means clustering (cont.)

Examples of applications include:

- Customer segmentation: Imagine that you are the owner of electronics store. You want to understand preferences of your clients to expand your business. It is not possible to look at each client's purchase details to find a good marketing strategy. But, you can group the details into, for example, five groups based on their purchasing habits. Then, you start building your marketing strategy for each group.
- Image segmentation and compression: The process of partitioning a digital image into multiple segments (sets of pixels) to simplify and change the representation of an image into something that is more meaningful and easier to analyze. To achieve this task, we need a process that assigns a label to every pixel in an image such that pixels with the same label share certain features. The image in this slide is segmented and compressed into three regions by using k-means clustering. With smaller number of clusters, it provides more image compression but at the expense of less image quality.
- Recommendation systems: These systems help you find users with the same preferences to build better recommendation systems.

References:

https://www.mathworks.com/help/examples/images/win64/SegmentGrayscaleImageUsingKMeansClusteringExample_02.png
https://www.mathworks.com/help/examples/images/win64/SegmentGrayscaleImageUsingKMeansClusteringExample_01.png

K-means clustering (cont.)

- Example: Given the following data points, use K-means clustering to partition data into two clusters.

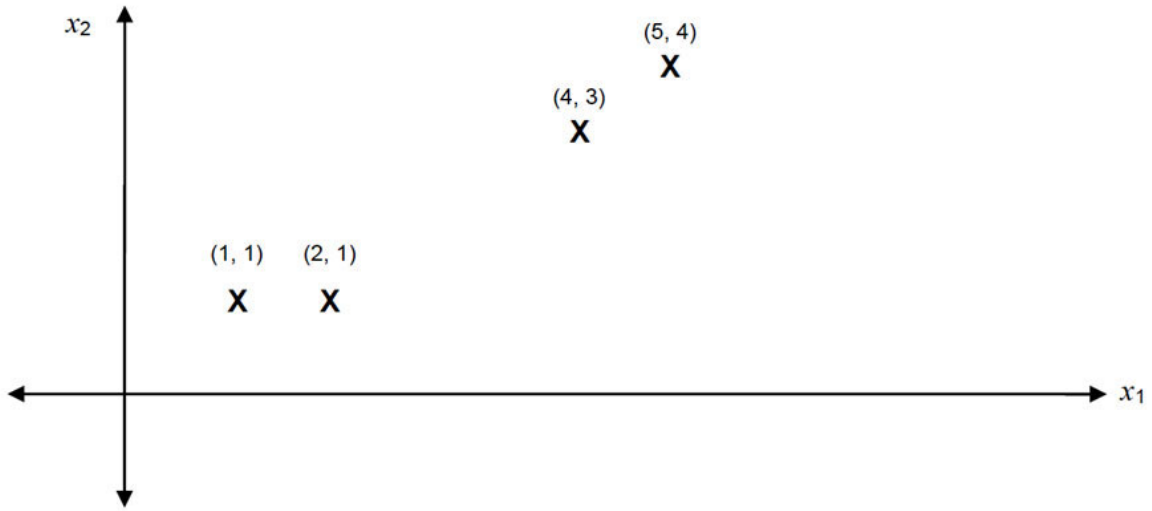


Figure 1-44. K-means clustering (cont.)

Assume that you have the data points that are show in the figure. Your goal is to cluster each data point into one of two groups. Thus, the cluster size is 2. C1 and C2 represent these two clusters.

K-means clustering (cont.)

- Set initial centroids are C1 (1,1) and C2 (2,1)

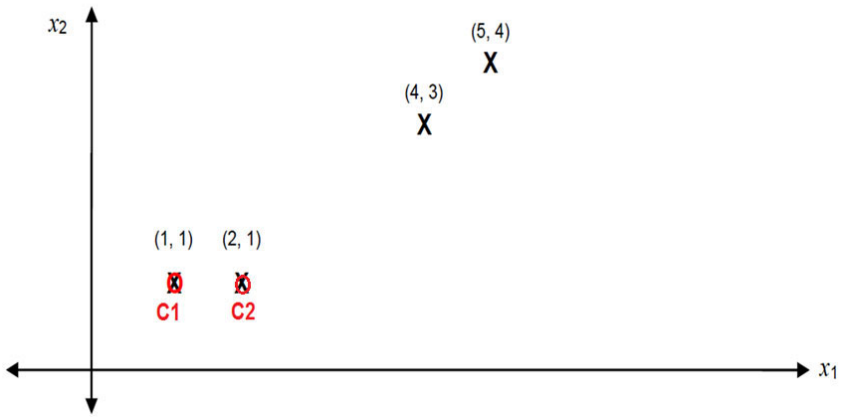


Figure 1-45. K-means clustering (cont.)

Assume initial centroids are C1, point (1,1) and C2, point (2,1)

IBM Training

K-means clustering (cont.)

Find a new centroid by using→ $C_{new} = \frac{1}{m} \times \sum_{i=1}^{m} (x^i)$

Iteration 1:

- Now, we calculate for each point to which center it belongs. The result depends on the distance between the center and the point (by using Euclidian distance):
  Point 1: (1, 1) → d11 = Yes d12 = No

  This means point1(2,2) belongs to C1 and not C2 because it is closer to C1.

  - Point 2: (2, 1) → d21 = No, d22 = Yes
  - Point 3: (4, 3) → d31 = No, d32 = Yes
  - Point 4: (5, 4) → d41 = No, d42 = Yes
- Now, we calculate the new centroid as follows:
  - C1 = (1, 1)
  - C2 = 1/3 ((2, 1) + (4, 3) + (5, 4)) = **(3.67, 2.67)**

Figure 1-46. K-means clustering (cont.)

To compute the centroid of a cluster, use an iterative process where each point is examined and you determine whether it belongs to a specific cluster. Then, you compute the new centroid by using the mean of all points.

K-means clustering (cont.)

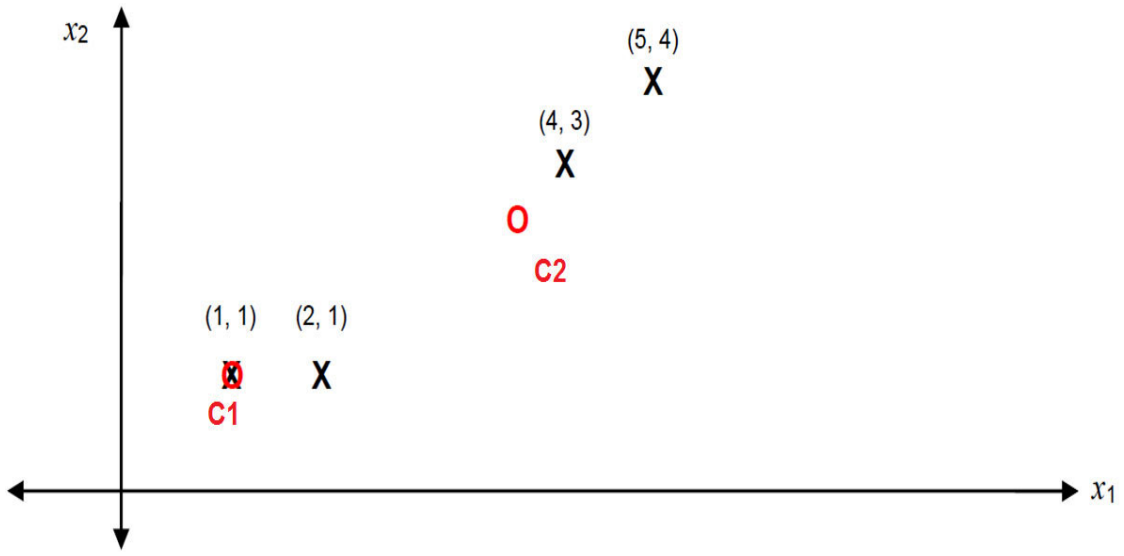


Figure 1-47. K-means clustering (cont.)

As you see, the new points in red are the new centroids. We apply another iteration to find a better centroid that represents each cluster.

K-means clustering (cont.)

Iteration 2:

- Point 1: (1, 1) → d11 = Yes, d12 = No
- Point 2: (2, 1) → d21 = Yes, d22 = No
- Point 3: (4, 3) → d31 = No, d32 = Yes
- Point 4: (5, 4) → d41 = No, d42 = Yes

Now, we calculate the new centroid as follows:

- C1 = 1/2 ((1, 1)+(2,1)) = (1.5,1)
- C2 = 1/2 ((4, 3) + (5, 4)) = **(4,5, 3.5)**

Figure 1-48. K-means clustering (cont.)

Now, we examine each point again against the centroid by using Euclidian distance and calculate the new centroids (C1 and C2).

K-means clustering (cont.)

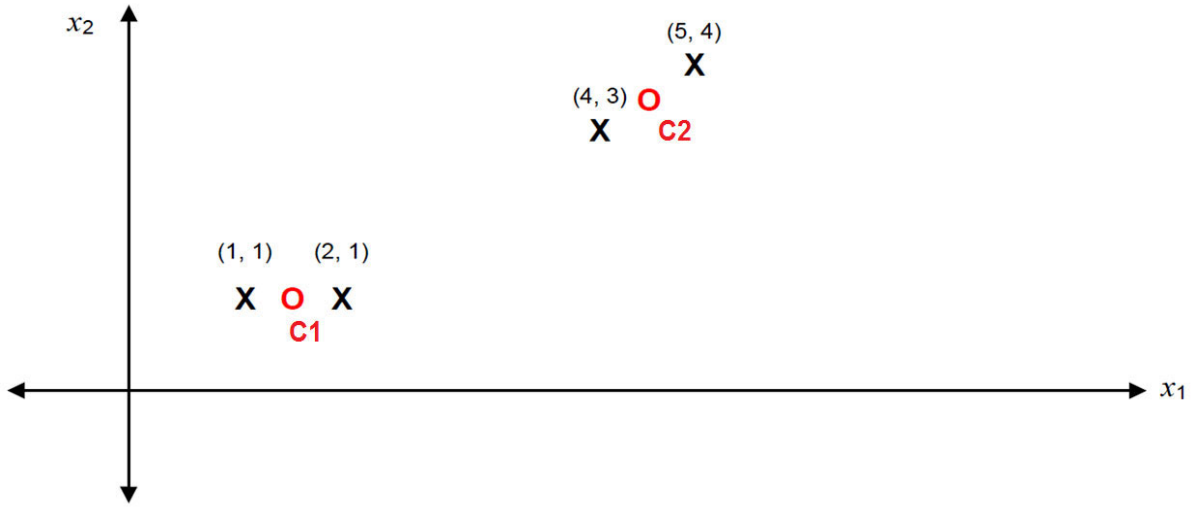


Figure 1-49. K-means clustering (cont.)

As you see, the new red centroids represent the centers of the two clusters. The algorithm stops when the centroids do not change or change slightly, or if a maximum number of iterations are defined.

1.3. What are neural networks?

IBM Training

What are neural networks?

Figure 1-50. What are neural networks?

IBM Training

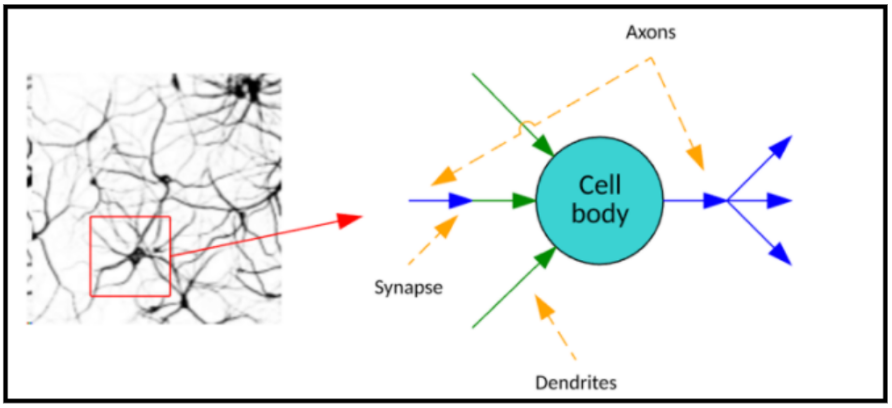
Topics

What is machine learning?
Machine learning algorithms
What are neural networks?
What is deep learning?
How to evaluate a machine learning model?

Figure 1-51. Topics

Neural networks

- Machine learning models that are inspired by the structure of the human brain.
- The human brain is estimated to have 100 billion neurons, and each neuron is connected to up to 10,000 other neurons.



Biological neural network

Figure 1-52. Neural networks

Neural networks represent an information-processing paradigm that is inspired by the human brain. In the brain, neurons are highly connected and communicate chemical signals through the synapses (a junction between two nerve cells) between the axons and dendrites. The human brain is estimated to have 100 billion neurons, with each neuron connected to up to 10,000 other neurons.

The figure shows a representation of a network of neurons in the brain.

IBM Training

Neural networks (cont.)

- Artificial neural networks are collections of interconnected “neurons” (called nodes) that work together to transform input data to output data.
- Each node applies a mathematical transformation to the data it receives; it then passes its result to the other nodes in its path.
- Examples for applications:
  - Object detection, tracking, and image and video analysis
  - Natural language processing (for example, machine translation)
  - Autonomous cars and robots

Figure 1-53. Neural networks (cont.)

Artificial neural networks communicate signals (numbers) through weights and activation functions that activate neurons. Using a training algorithm, these networks adjust those weights to solve a problem.

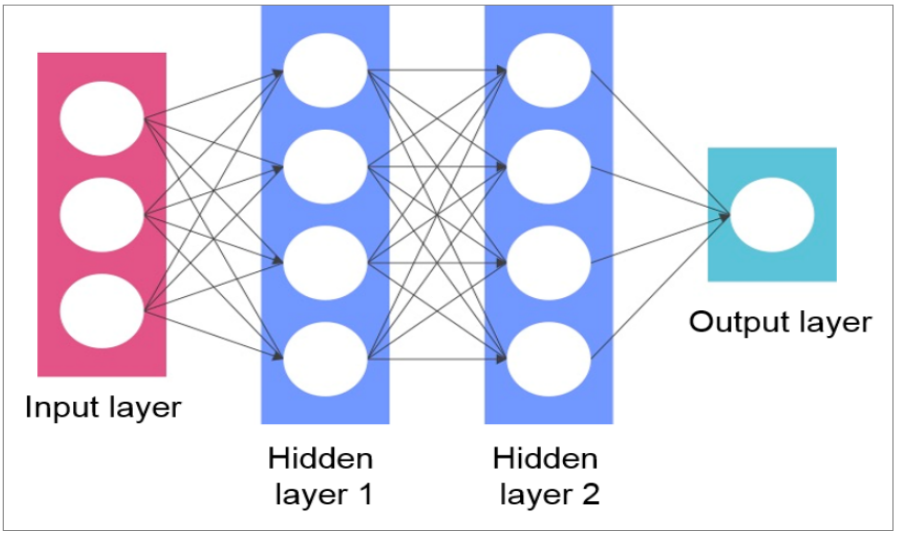
Each node applies a mathematical transformation to the data it receives; it then passes its result to the other nodes in its path. Each connection between nodes represents a different parameter to the model.

A neural network is useful for machine learning tasks that have too many features (millions). For example:

- Object detection, tracking, and image and video analysis by using a Convolutional Neural Network (CNN)
- Natural language processing tasks like speech recognition and machine translation by using a recurrent neural network (RNN)
- Autonomous cars and robots (more complex neural networks)

Neural networks (cont.)

- Three or more layers (an input layer, one or many hidden layers, and an output layer).
- Neural network models can adjust and learn as data changes.



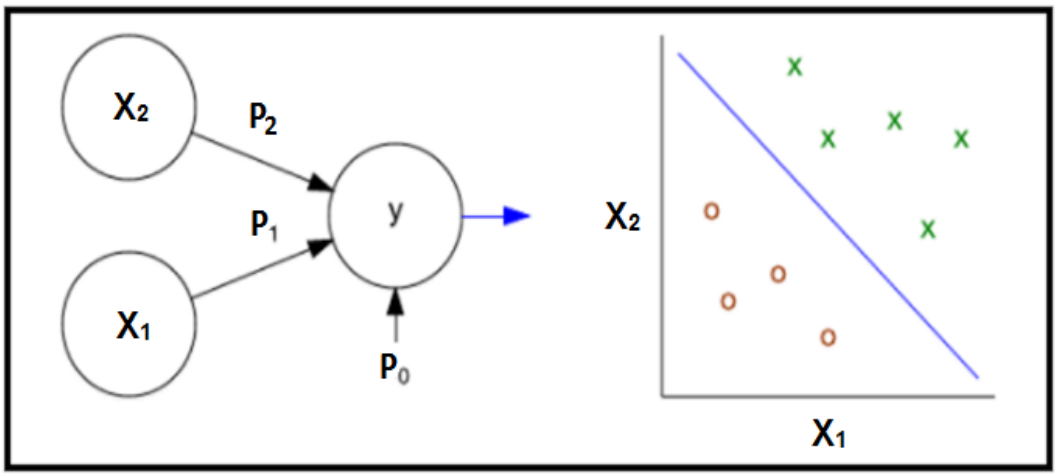
Multilayer neural network

Figure 1-54. Neural networks (cont.)

A neural network is composed of three or more layers: an input layer, one or many hidden layers, and an output layer. Data is imported through the input layer. Then, the data is modified in the hidden and output layers based on the weights that are applied to their nodes. The typical neural network can consist of thousands or even millions of simple processing nodes that are densely interconnected.

Perceptron

- A single neuron model and originator for the neural network.
- Similar to linear classification, where each input has weight.
- One bias.



Perceptron and linear classification

Figure 1-55. Perceptron

A perceptron is a single neuron model that was an originator for neural networks. It is similar to linear regression. Each neuron has its own bias and slope (weights). For example, assume that a neuron have two inputs (X_1 and X_2), so it requires three weights (P1, P_2 and P0). The figure in this slide shows a weight for each input and one for the bias.

IBM Training

Neural networks: Backpropagation

Backpropagation is an algorithm for training neural networks that has many layers. It works in two phases:

- **First phase:** The propagation of inputs through a neural network to the final layer (called feedforward).
- **Second phase:** The algorithm computes an error. An error value is then calculated by using the wanted output and the actual output for each output neuron in the network. The error value is propagated backward through the weights of the network (adjusting the weights) beginning with the output neurons through the hidden layer and to the input layer (as a function of the contribution of the error).

Figure 1-56. Neural networks: Backpropagation

Backpropagation is an algorithm for training neural networks that have many layers. It works in two phases:

- Propagation of inputs through a neural network to the final layer (called feedforward).
- The algorithm computes an error. An error value is then calculated by using the wanted output and the actual output for each output neuron in the network. The error value is propagated backward through the weights of the network (adjusting the weights) beginning with the output neurons through the hidden layer and to the input layer (as a function of the contribution of the error).

Backpropagation continues to be an important aspect of neural network learning. With faster and cheaper computing resources, it continues to be applied to larger and denser networks.

1.4. What is deep learning?

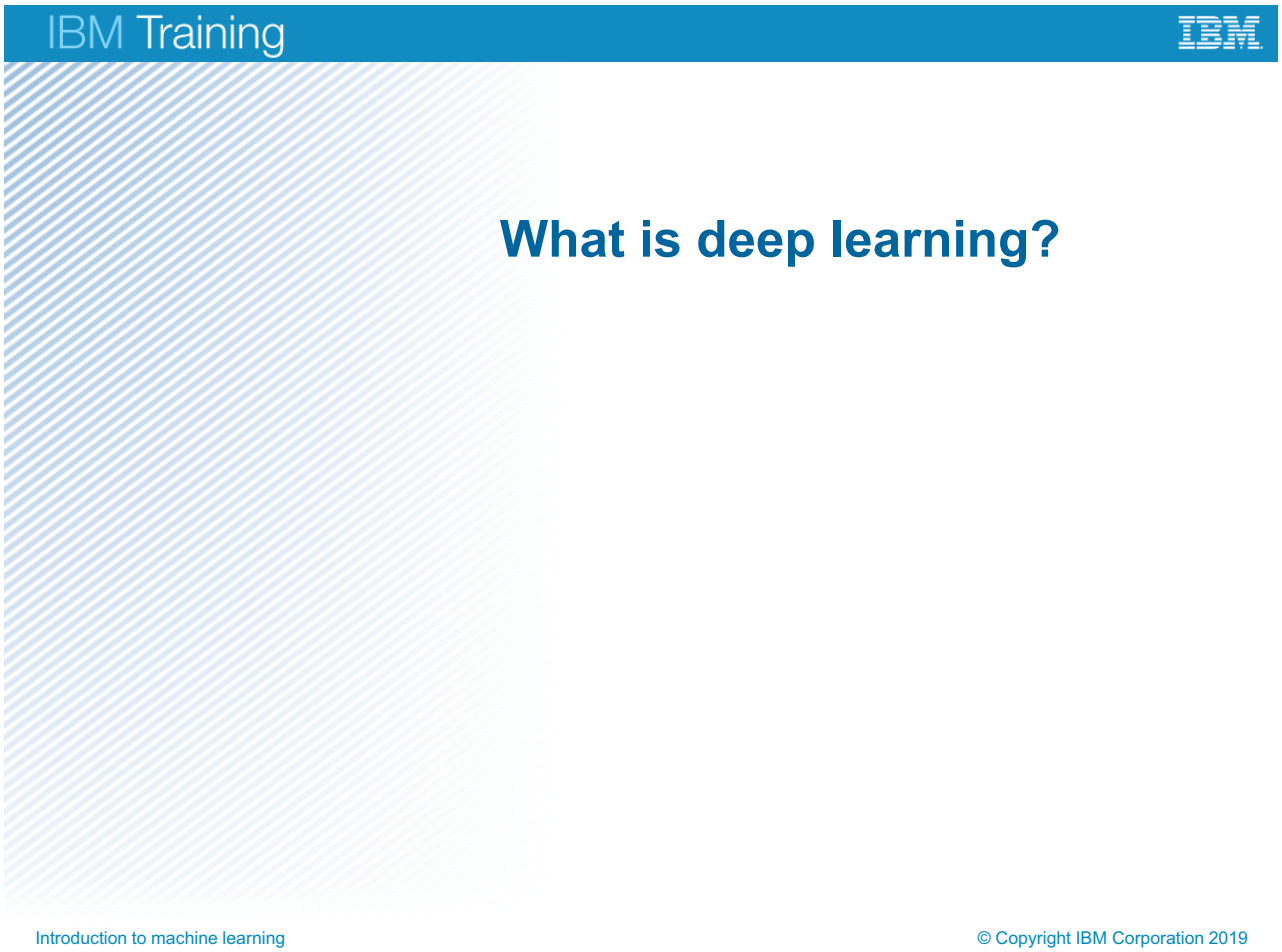


Figure 1-57. What is deep learning?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Figure 1-58. Topics

IBM Training

Deep learning

- Similar to a traditional neural network, but it has many more hidden layers.
- Deep learning has emerged now because of the following reasons:
  - Emergence of big data, which requires data processing scaling.
  - Improvement in processing power and the usage of GPUs to train neural networks.
  - Advancement in algorithms like the rectified linear unit (ReLU).

Figure 1-59. Deep learning

Deep learning is a machine learning technique that uses neural networks to learn. Although deep learning is similar to a traditional neural network, it has many more hidden layers. The more complex the problem, the more hidden layers there are in the model.

Deep learning has emerged now because of the following reasons:

- The continuous increase in big data requires data processing scaling to analyze and use this data correctly.
- Improvement in processing power and the usage of GPUs to train neural networks.
- Advancement in algorithms like the rectified linear unit (ReLU) instead of the Sigmoid algorithm helps make gradient descent converge faster.

IBM Training

Deep learning (cont.)

Applications:

- Multilayer perceptron (MLP): Classification and regression, for example, a house price prediction.
- Convolutional neural network (CNN): For image processing like facial recognition.
- Recurrent neural network (RNN): For one-dimensional sequence input data. Like audio and languages.
- Hybrid neural network: Covering more complex neural networks, for example, autonomous cars.

Figure 1-60. Deep learning (cont.)

There are various types of neural networks. Each network is more suitable for a type of machine learning problem. Here is an overview for these networks and their applications:

- Multilayer perceptron (MLP): A class of feed-forward artificial neural networks (ANNs). It is useful in classification problems where inputs are assigned a class. It also works in regression problems for a real-valued quantity like a house price prediction.
- Convolutional neural network (CNN): Takes an input as an image. It is useful for image recognition problems like facial recognition.
- Recurrent neural network (RNN): Has a temporal nature where the input may be a function in time, such as audio files. It is also used for one-dimensional sequence data. It is suitable for inputs like audio and languages. It can be used in applications like speech recognition and machine translation.
- Hybrid neural network: Covers more complex neural networks, for example, autonomous cars that require processing images and work by using radar.

Reference:

https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/

1.5. How to evaluate a machine learning model?

IBM Training

How to evaluate a machine learning model?

Figure 1-61. How to evaluate a machine learning model?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Figure 1-62. Topics

IBM Training

Model evaluation

- **Overfitting** occurs when a machine learning model can fit the training set perfectly and fails with unseen future data.
  - **Reason:** Too many features are used or you are reusing training samples in testing.
  - **Solution**:
    - Fewer features
    - More data
    - Cross-validation

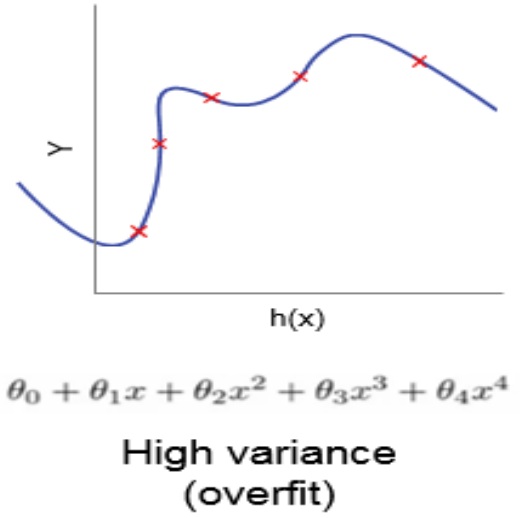


Figure 1-63. Model evaluation

After you have successfully trained your model, you need a methodology to follow to evaluate your machine learning model performance. A classic mistake is to use the same sample data that is used in training to test a model, which produces a false perfect score. This is called “overfitting” (also referred as “high variance”). The problem with overfitting is that your model fails at predicting future unseen data.

Another case that can cause overfitting is where you have unbalanced data. For example, assume that you are working on a data set for churn analysis. The customers who churned are actually 2% of your data set. Using this data set “as is” causes overfitting.

The objective of a good machine learning model is to generalize for any future data points. Overfitting also can occur if you are using too many features. Relatively, if the number of features is the same as or greater than the number of training samples, that can cause overfitting. One of the solutions to overcome overfitting is to increase the number of data set samples that is used for training compared to features. Another solution is to manually decrease the number of features, but that might result in removing useful information. Another solution is to perform model selection by using cross-validation.

References:

https://www.coursera.org/lecture/machine-learning/the-problem-of-overfitting-ACpTQ

https://en.oxforddictionaries.com/definition/overfitting

https://ai.stanford.edu/~ang/papers/cv-final.pdf

https://www.youtube.com/watch?v=OSd30QGMI88

IBM Training

Model evaluation (cont.)

- **Underfitting** occurs when a machine learning model cannot fit the training data or generalize to new data.
  - **Reason:** The model is using a simple estimator.
  - **Solution**: Add More features or use different estimator

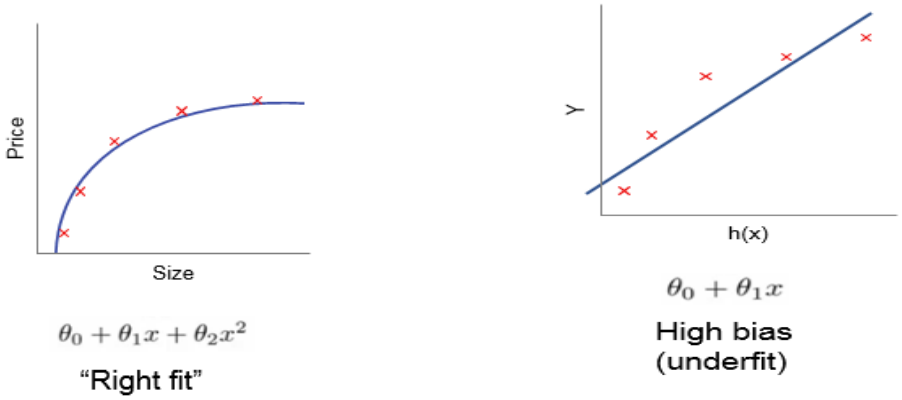


Figure 1-64. Model evaluation (cont.)

Underfitting (also referred to as "high bias") occurs when a machine learning model cannot fit the training data or generalize to new data.

A possible reason might be that the model is using a simple estimator. For example, you might be using a linear estimator, but what you actually need is a quadratic or higher degree polynomial estimator to develop your model like in "Right fit" graph.

Another reason might be that you are not using enough features, so your estimator fails to capture the structure of the data. A possible solution would be to add more features and try a different estimator.

There are other methods that are used to help resolve the overfitting and underfitting of your model such as regularization, but these methods are beyond the scope of this course.

References:

https://en.oxforddictionaries.com/definition/overfitting
https://www.youtube.com/watch?v=OSd30QGMI88

IBM Training

Model evaluation (cont.)

- **Cross-validation (CV)** is a process to evaluate a model by dividing the data set once or several times in training and testing.
- **Hold-out method:** Randomly splits the data set into a training set and test set.
- **K-fold cross validation:** Splits data into K subsamples where each subsample gets a chance to be the validation set, and K-1 is the training set.
- **Leave one out cross validation (LOO-CV):** Similar to K-fold except that one subsample that contains one data point is held out, and the rest of data is used for training.

Figure 1-65. Model evaluation (cont.)

It is common practice when applying a (supervised) machine learning task is to hold out part of the available data as a test set. There are different methods to achieve that task:

- Cross-validation (CV) is a process to evaluate a machine learning model by splitting a data set once or several times to train and test the model. The data set can be split into a training set to train the model and a validation set to pre-test the model. Select the model that has least error. Finally, there is a test set to evaluate the model. Thus, the data set can be split as 60% - 20% - 20% for training, validation, and testing sets.

  One criticism of this process is that splitting the data set into three parts reduces the number of samples that can be used for training the model.
- The hold-out method partitions the data set into a majority set for training and minority set for testing. The split of the training set to test set is 80% - 20% or 70% - 30%, with no fixed rule.

- K-fold cross validation randomly partitions data into K equal sized subsamples. For each iteration, one subsample is kept as validation set and the rest of the subsamples (K-1) are the training set. The iterations are repeated K times, where each subsample has one chance to be the validation set. The K results can then be averaged to produce a single model. The biggest advantage of K-fold is that all data is changed to be used for both training and validation. There is no strict rule for the number K, but it is commonly K=5 or K=10, which are 5-fold cross-validation or 10-fold cross-validation. For each subsample, you maintain approximately the same percentage of data of each target class as in the complete set, which is known as the Stratified K-fold method.
- Leave one out CV (LOO-CV) is similar to K-fold, but in this case each one sample data point is held out as a validation set, and the rest of data set is the training set. Comparing LOO-CV and K-fold, K-fold is faster and requires less computation, but in terms of accuracy, LOO-CV often has a high variance as an estimator.

References:

https://projecteuclid.org/download/pdfview_1/euclid.ssu/1268143839

http://scikit-learn.org/stable/modules/cross_validation.html

https://www.cs.cmu.edu/~schneide/tut5/node42.html

IBM Training

Unit summary

- Explain what is machine learning.
- Describe what is meant by statistical model and algorithm.
- Describe data and data types.
- Describe machine learning types and approaches (Supervised, Unsupervised and Reinforcement).
- List different machine learning algorithms.
- Explain what neural networks and deep learning are, and why they are important in today's AI field.
- Describe machine learning components.
- List the steps in the process to build machine learning applications.
- Explain what domain adaptation is and its applications.

Figure 1-66. Unit summary