Introduction

Amir H. Payberah
payberah@kth.se
30/10/2018

Course Information

Course Objective

- This course has a system-based focus
- Learn the theory of machine learning and deep learning
- Learn the practical aspects of building machine learning and deep learning algorithms using data parallel programming platforms, such as Spark and TensorFlow

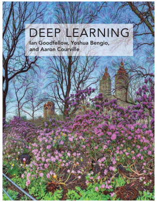
Topics of Study

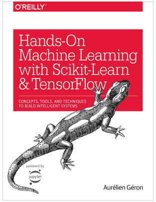
- Part 1: large scale machine learning
    - Spark ML
    - Linear regression and logistic regression
    - Decision tree and ensemble models

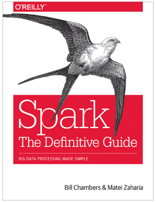
- Part 2: large scale deep learning
    - TensorFlow
    - Deep feedforward networks
    - Convolutional neural networks (CNNs)
    - Recurrent neural networks (RNNs)
    - Autoencoders and Restricted Boltzmann machines (RBMs)

The Course Material

- Deep learning, I. Goodfellow et al., Cambridge: MIT press, 2016
- Hands-on machine learning with Scikit-Learn and TensorFlow, A. Geron, O'Reilly Media, 2017
- Spark - The Definitive Guide, M. Zaharia et al., O'Reilly Media, 2018.







The Course Grading

- Two lab assignments: 30%
- One final project: 20%
- Eight review questions: 20%
- The final exam: 30%

The Labs and Project

- Self-selected groups of two
- Labs
  - Include Scala/Python programming
  - Lab1: Regression using Spark ML
  - Lab2: Deep neural network and CNN using Tensorflow
- Project
  - Selection of a large dataset and method
  - RNNs, Autoencoders, or RBMs
  - Demonstrated as a demo and short report

The Course Web Page

https://id2223kth.github.io

The Course Overview

Sheepdog or Mop

Chihuahua or Muffin

Barn Owl or Apple

Raw Chicken or Donald Trump

Artificial Intelligence Challenge

- Artificial intelligence (AI) can solve problems that can be described by a list of formal mathematical rules.
- The challenge is to solve the tasks that are hard for people to describe formally.
- Let computers to learn from experience.

History of AI

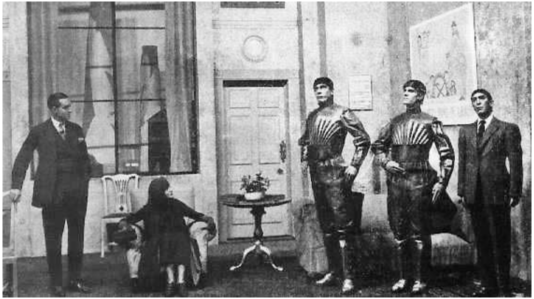
Greek Myths

- Hephaestus, the god of blacksmith, created a metal automaton, called Talos.

[the left figure: http://mythologian.net/hephaestus-the-blacksmith-of-gods]
[the right figure: http://elderscrolls.wikia.com/wiki/Talos]

1920: Rossum's Universal Robots (R.U.R.)

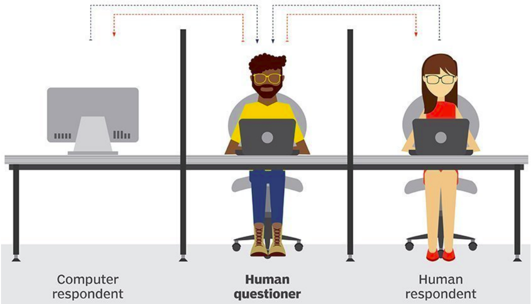
- A science fiction play by Karel Čapek, in 1920.
- A factory that creates artificial people named robots.



[https://dev.to/lschultebraucks/a-short-history-of-artificial-intelligence-7hm]

1950: Turing Test

- In 1950, Turing introduced the Turing test.
- An attempt to define machine intelligence.



[https://searchenterpriseai.techtarget.com/definition/Turing-test]

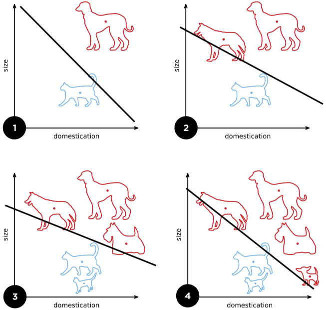
1956: The Dartmouth Workshop

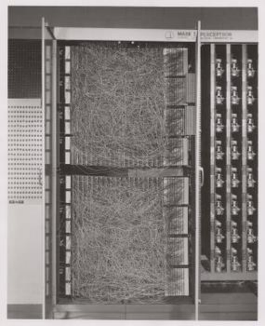
- Probably the first workshop of AI.
- Researchers from CMU, MIT, IBM met together and founded the AI research.

[https://twitter.com/lordsaicom/status/898139880441696257]

1958: Perceptron

- A supervised learning algorithm for binary classifiers.
- Implemented in custom-built hardware as the Mark 1 perceptron.

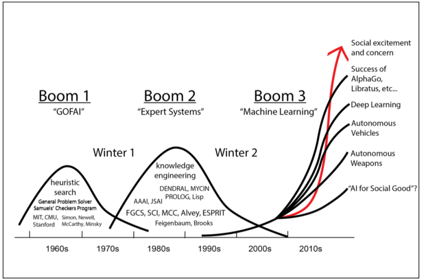




[https://en.wikipedia.org/wiki/Perceptron]

1974–1980: The First AI Winter

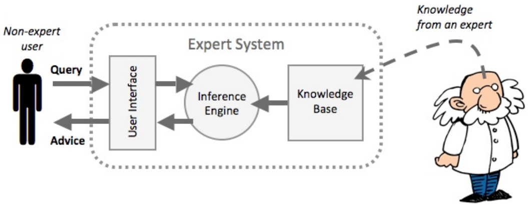
- The over optimistic settings, which were not occurred
- The problems:
    - Limited computer power
    - Lack of data
    - Intractability and the combinatorial explosion



[http://www.technologystories.org/ai-evolution]

1980's: Expert systems

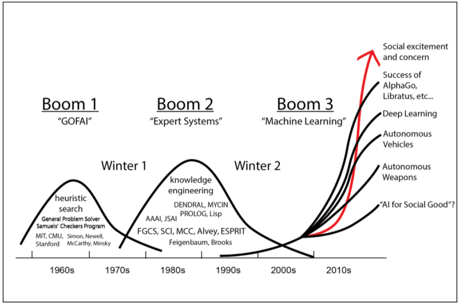
- The programs that solve problems in a specific domain.
- Two engines:
  - Knowledge engine: represents the facts and rules about a specific topic.
  - Inference engine: applies the facts and rules from the knowledge engine to new facts.



[https://www.igcseict.info/theory/7_2/expert]

1987–1993: The Second AI Winter

- After a series of financial setbacks.
- The fall of expert systems and hardware companies.



[http://www.technologystories.org/ai-evolution]

1997: IBM Deep Blue

- The first chess computer to beat a world chess champion Garry Kasparov.

[http://marksist.org/icerik/Tarihte-Bugun/1757/11-Mayis-1997-Deep-Blue-adli-bilgisayar]

2012: AlexNet - Image Recognition

- The ImageNet competition in image classification.
- The AlexNet Convolutional Neural Network (CNN) won the challenge by a large margin.

IMAGENET

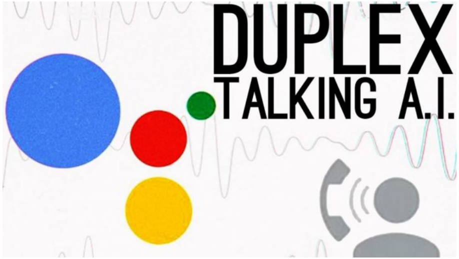
2016: DeepMind AlphaGo

- DeepMind AlphaGo won Lee Sedol, one of the best players at Go.
- In 2017, DeepMind published AlphaGo Zero.
    - The next generation of AlphaGo.
    - It learned Go by playing against itself.

[https://www.zdnet.com/article/google-alphago-caps-victory-by-winning-final-historic-go-match]

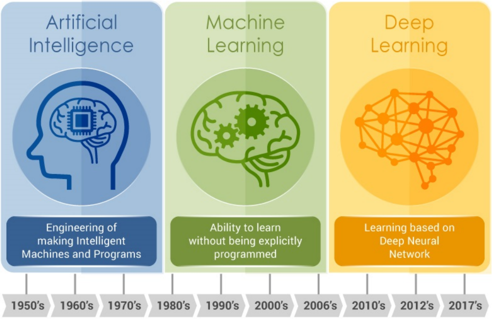
2018: Google Duplex

- An AI system for accomplishing real-world tasks over the phone.
- A Recurrent Neural Network (RNN) built using TensorFlow.



AI Generations

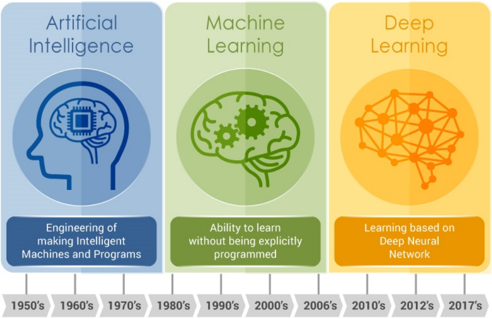
- Rule-based AI
- Machine learning
- Deep learning



[https://bit.ly/2woLEzs]

AI Generations - Rule-based AI

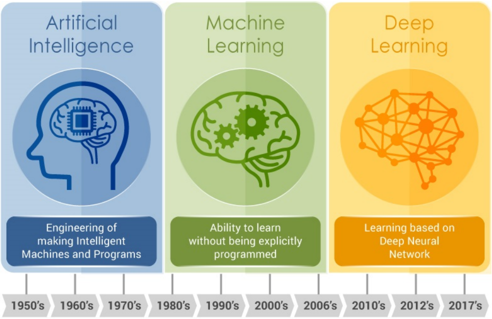
- Hard-code knowledge
- Computers reason using logical inference rules



[https://bit.ly/2woLEzs]

AI Generations - Machine Learning

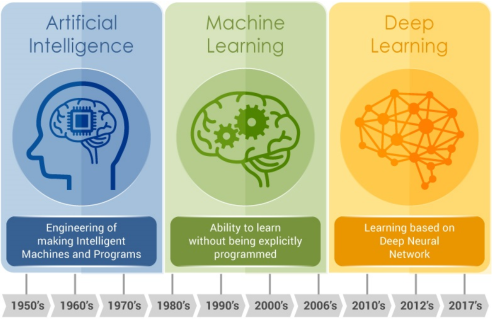
- If AI systems acquire their own knowledge
- Learn from data without being explicitly programmed



[https://bit.ly/2woLEzs]

AI Generations - Deep Learning

- For many tasks, it is difficult to know what features should be extracted
- Use machine learning to discover the mapping from representation to output



[https://bit.ly/2woLEzs]

Why Does Deep Learning Work Now?

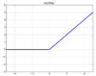
- Huge quantity of data
- Tremendous increase in computing power
- Better training algorithms

Data

GPUs

+

Weight Initialization



Non-Linearity

Machine Learning and Deep Learning

Learning Algorithms

- A ML algorithm is an algorithm that is able to learn from data.
- What is learning?
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (Tom M. Mitchell)

Learning Algorithms - Example 1

- A spam filter that can learn to flag spam given examples of spam emails and examples of regular emails.
- Task T: flag spam for new emails
- Experience E: the training data
- Performance measure P: the ratio of correctly classified emails

`[https://bit.ly/2oiplYM]`

Learning Algorithms - Example 2

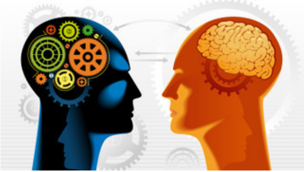
- Given dataset of prices of 500 houses, how can we learn to predict the prices of other houses, as a function of the size of their living areas?
- Task T: predict the price
- Experience E: the dataset of living areas and prices
- Performance measure P: the difference between the predicted price and the real price

[https://bit.ly/2MyiJUy]

Types of Machine Learning Algorithms

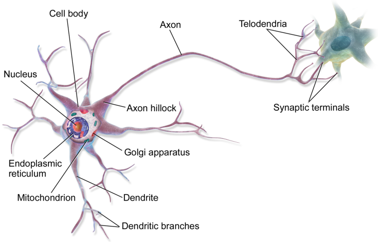
- Supervised learning
  - Input data is labeled, e.g., spam/not-spam or a stock price at a time.
  - Regression vs. classification

- Unsupervised learning
  - Input data is unlabeled.
  - Find hidden structures in data.



From Machine Learning to Deep Learning

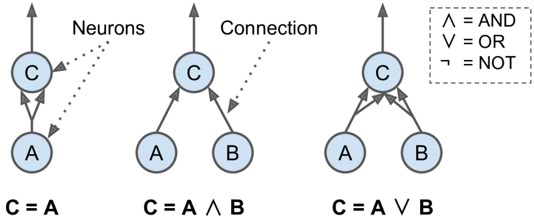
- Deep Learning (DL) is part of ML methods based on learning data representations.
- Mimic the neural networks of our brain.



[A. Geron, O'Reilly Media, 2017]

Artificial Neural Networks

- Artificial Neural Network (ANN) is inspired by biological neurons.
- One or more binary inputs and one binary output
- Activates its output when more than a certain number of its inputs are active.

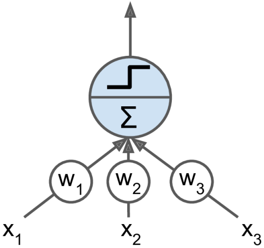


[A. Geron, O'Reilly Media, 2017]

The Linear Threshold Unit (LTU)

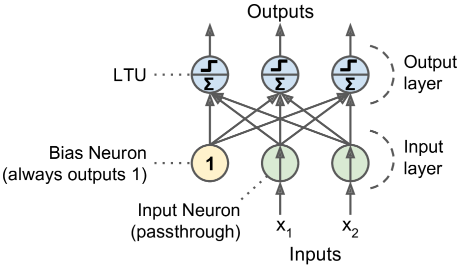
- Inputs of a LTU are numbers (not binary).
- Each input connection is associated with a weight.
- Computes a weighted sum of its inputs and applies a step function to that sum.

- $z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \mathbf{w}^\intercal \mathbf{x}$
- $\hat{y} = \mathtt{step}(z) = \mathtt{step}(\mathbf{w}^\intercal \mathbf{x})$



The Perceptron

- The perceptron is a single layer of LTUs.
- The input neurons output whatever input they are fed.
- A bias neuron, which just outputs 1 all the time.

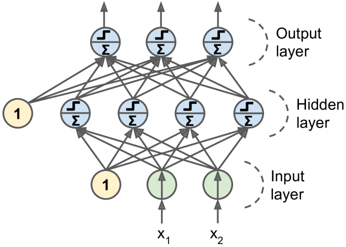


Deep Learning Models

- Deep Neural Network (DNN)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Autoencoders

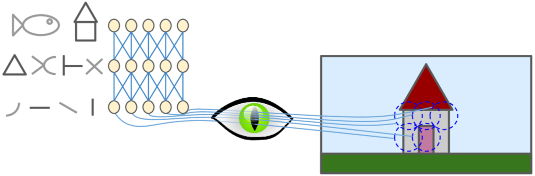
Deep Neural Networks

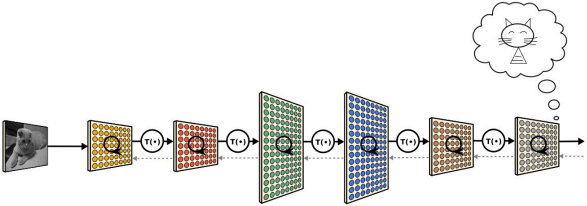
- Multi-Layer Perceptron (MLP)
  - One input layer
  - One or more layers of LTUs (hidden layers)
  - One final layer of LTUs (output layer)
- Deep Neural Network (DNN) is an ANN with two or more hidden layers.
- Backpropagation training algorithm



Convolutional Neural Networks

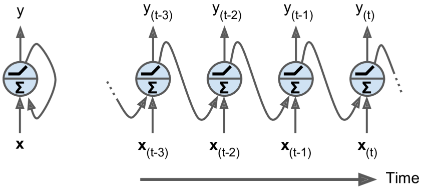
- Many neurons in the visual cortex react only to a limited region of the visual field.
- The higher-level neurons are based on the outputs of neighboring lower-level neurons





Recurrent Neural Networks

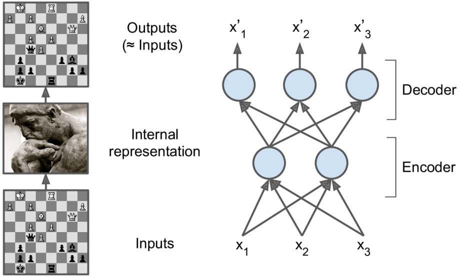
- The output depends on the input and the previous computations.



- Analyze time series data, e.g., stock market, and autonomous driving systems
- Work on sequences of arbitrary lengths, rather than on fixed-sized inputs


Autoencoders

- Learn efficient representations of the input data, without any supervision.
  - With a lower dimensionality than the input data
- Generative model: generate new data that looks very similar to the training data.
- Preserve as much information as possible



[A. Geron, O'Reilly Media, 2017]

Linear Algebra Review

Vector

- A vector is an array of numbers.
- Notation:
  - Denoted by **bold** lowercase letters, e.g., $\mathbf{x}$.
  - x_i denotes the ith entry.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Matrix and Tensor

- A matrix is a 2-D array of numbers.
- A tensor is an array with more than two axes.
- Notation:
  - Denoted by **bold** uppercase letters, e.g., $\mathbf{A}$.
  - $\mathtt{a}_{\mathtt{ij}}$ denotes the entry in $\mathtt{i}$th row and $\mathtt{j}$th column.
  - If $\mathbf{A}$ is $\mathtt{m} \times \mathtt{n}$, it has $\mathtt{m}$ rows and $\mathtt{n}$ columns.

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{bmatrix}$$

Matrix Addition and Subtraction

- The matrices must have the same dimensions.

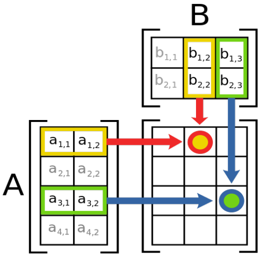
$$\mathbf{A} = \begin{bmatrix} \mathtt{a} & \mathtt{b} \\ \mathtt{c} & \mathtt{d} \end{bmatrix} + \begin{bmatrix} \mathtt{e} & \mathtt{f} \\ \mathtt{g} & \mathtt{h} \end{bmatrix} = \begin{bmatrix} \mathtt{a} + \mathtt{e} & \mathtt{b} + \mathtt{f} \\ \mathtt{c} + \mathtt{g} & \mathtt{d} + \mathtt{h} \end{bmatrix}$$

Matrix Product

- The matrix product of matrices **A** and **B** is a third matrix **C**, where $\mathbf{C} = \mathbf{AB}$.
- If **A** is of shape $\mathtt{m} \times \mathtt{n}$ and **B** is of shape $\mathtt{n} \times \mathtt{p}$, then **C** is of shape $\mathtt{m} \times \mathtt{p}$.

$$\mathtt{c}_{\mathtt{ij}} = \sum_{\mathtt{k}} \mathtt{a}_{\mathtt{ik}} \mathtt{b}_{\mathtt{kj}}$$

- Properties
  - Associative: $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$
  - Not commutative: $\mathbf{AB} \neq \mathbf{BA}$



[https://en.wikipedia.org/wiki/Matrix_multiplication]

Matrix Transpose

- Swap the rows and columns of a matrix.

$$\mathbf{A} = \begin{bmatrix} \mathtt{a} & \mathtt{b} \\ \mathtt{c} & \mathtt{d} \\ \mathtt{e} & \mathtt{f} \end{bmatrix} \Rightarrow \mathbf{A}^\intercal = \begin{bmatrix} \mathtt{a} & \mathtt{c} & \mathtt{e} \\ \mathtt{b} & \mathtt{d} & \mathtt{f} \end{bmatrix}$$

- Properties
  - $\mathbf{A}_{\mathtt{ij}} = \mathbf{A}^\intercal_{\mathtt{ji}}$
  - If $\mathbf{A}$ is $\mathtt{m} \times \mathtt{n}$, then $\mathbf{A}^\intercal$ is $\mathtt{n} \times \mathtt{m}$
  - $(\mathbf{A} + \mathbf{B})^\intercal = \mathbf{A}^\intercal + \mathbf{B}^\intercal$
  - $(\mathbf{AB})^\intercal = \mathbf{B}^\intercal\mathbf{A}^\intercal$

Inverse of a Matrix

- If $\mathbf{A}$ is a square matrix, its inverse is called $\mathbf{A}^{-1}$.

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

- Where $\mathbf{I}$, the identity matrix, is a diagonal matrix with all 1's on the diagonal.

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

L^p Norm for Vectors

- We can measure the size of vectors using a norm function.
- Norms are functions mapping vectors to non-negative values.
- L^1 norm

$$||\mathbf{x}||_1 = \sum_i |x_i|$$

- L^2 norm

$$||\mathbf{x}||_2 = (\sum_i |x_i|^2)^{\frac{1}{2}} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

- L^p norm

$$||\mathbf{x}||_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$$

Probability Review

Random Variables

- Random variable: a variable that can take on different values randomly.
- Random variables may be discrete or continuous.
    - Discrete random variable: finite or countably infinite number of states
    - Continuous random variable: real value
- Notation:
    - Denoted by an upper case letter, e.g., $\mathtt{X}$
    - Values of a random variable $\mathtt{X}$ are denoted by lower case letters, e.g., $\mathtt{x}$ and $\mathtt{y}$.

Probability Distributions

- Probability distribution: how likely a random variable is to take on each of its possible states.
    - E.g., the random variable `X` denotes the outcome of a coin toss.
    - The probability distribution of `X` would take the value 0.5 for `X` = `head`, and 0.5 for `Y` = `tail` (assuming the coin is fair).

- The way we describe probability distributions depends on whether the variables are discrete or continuous.

Discrete Variables

- Probability mass function (PMF): the probability distribution of a discrete random variable $\mathtt{X}$.

- Notation: denoted by a lowercase $\mathtt{p}$.
  - E.g., $\mathtt{p}(\mathtt{x}) = 1$ indicates that $\mathtt{X} = \mathtt{x}$ is certain
  - E.g., $\mathtt{p}(\mathtt{x}) = 0$ indicates that $\mathtt{X} = \mathtt{x}$ is impossible

- Properties:
  - The domain $\mathtt{D}$ of $\mathtt{p}$ must be the set of all possible states of $\mathtt{X}$
  - $\forall \mathtt{x} \in \mathtt{D}(\mathtt{X}), 0 \leq \mathtt{p}(\mathtt{x}) \leq 1$
  - $\sum_{\mathtt{x} \in \mathtt{D}(\mathtt{X})} \mathtt{p}(\mathtt{x}) = 1$

Independence

- Two random variables $\mathtt{X}$ and $\mathtt{Y}$ are independent, if their probability distribution can be expressed as their products.

$$\forall \mathtt{x} \in \mathtt{D}(\mathtt{X}), \mathtt{y} \in \mathtt{D}(\mathtt{Y}), \mathtt{p}(\mathtt{X} = \mathtt{x}, \mathtt{Y} = \mathtt{y}) = \mathtt{p}(\mathtt{X} = \mathtt{x})\mathtt{p}(\mathtt{Y} = \mathtt{y})$$

- E.g., if a coin is tossed and a single 6-sided die is rolled, then the probability of landing on the head side of the coin and rolling a 3 on the die is:

$$\mathtt{p}(\mathtt{X} = \mathtt{head}, \mathtt{Y} = 3) = \mathtt{p}(\mathtt{X} = \mathtt{head})\mathtt{p}(\mathtt{Y} = 3) = \frac{1}{2} \times \frac{1}{6} = \frac{1}{12}$$

Conditional Probability

- Conditional probability: the probability of an event given that another event has occurred.

$$\mathtt{p}(\mathtt{Y} = \mathtt{y} \mid \mathtt{X} = \mathtt{x}) = \frac{\mathtt{p}(\mathtt{Y} = \mathtt{y}, \mathtt{X} = \mathtt{x})}{\mathtt{p}(\mathtt{X} = \mathtt{x})}$$

- E.g., if 60% of the class passed both labs and 80% of the class passed the first labs, then what percent of those who passed the first lab also passed the second lab?
  - E.g., $\mathtt{X}$ and $\mathtt{Y}$ random variables for the first and the second labs, respectively.

$$\mathtt{p}(\mathtt{Y} = \mathtt{lab2} \mid \mathtt{X} = \mathtt{lab1}) = \frac{\mathtt{p}(\mathtt{Y} = \mathtt{lab2}, \mathtt{X} = \mathtt{lab1})}{\mathtt{p}(\mathtt{X} = \mathtt{lab1})} = \frac{0.6}{0.8} = \frac{3}{4}$$

Expectation

- The expected value of a random variable X with respect to a probability distribution $p(X)$ is the average value that X takes on when it is drawn from $p(X)$.

$$E_{x \sim p}[X] = \sum_{x} p(x)x$$

- E.g., If $X : \{1, 2, 3\}$, and $p(X = 1) = 0.3$, $p(X = 2) = 0.5$, $p(X = 3) = 0.2$
  - $E[X] = 0.3 \times 1 + 0.5 \times 2 + 0.2 \times 3 = 1.9$

Variance and Standard Deviation

- The variance gives a measure of how much the values of a random variable $\mathtt{X}$ vary as we sample it from its probability distribution $\mathtt{p(X)}$.

$$\mathtt{Var(X)} = \mathtt{E}[(\mathtt{X} - \mathtt{E[X]})^2]$$

$$\mathtt{Var(X)} = \sum_{\mathtt{x}} \mathtt{p(x)}(\mathtt{x} - \mathtt{E[X]})^2$$

- E.g., If $\mathtt{X} : \{1, 2, 3\}$, and $\mathtt{p}(\mathtt{X} = 1) = 0.3$, $\mathtt{p}(\mathtt{X} = 2) = 0.5$, $\mathtt{p}(\mathtt{X} = 3) = 0.2$
  - $\mathtt{E[X]} = 0.3 \times 1 + 0.5 \times 2 + 0.2 \times 3 = 1.9$
  - $\mathtt{Var(X)} = 0.3(1 - 1.9)^2 + 0.5(2 - 1.9)^2 + 0.2(3 - 1.9)^2 = 0.49$

- The standard deviation, shown by σ, is the square root of the variance.

Covariance (1/2)

- The covariance gives some sense of how much two values are linearly related to each other.

$$\mathtt{Cov}(\mathtt{X},\mathtt{Y}) = \mathtt{E}[(\mathtt{X}-\mathtt{E}[\mathtt{X}])(\mathtt{Y}-\mathtt{E}[\mathtt{Y}])]$$

$$\mathtt{Cov}(\mathtt{X},\mathtt{Y}) = \sum\sum_{(\mathtt{x},\mathtt{y})} \mathtt{p}(\mathtt{x},\mathtt{y})(\mathtt{x}-\mathtt{E}[\mathtt{X}])(\mathtt{y}-\mathtt{E}[\mathtt{Y}])$$

Covariance (2/2)

			Y		
	p(X, Y)	1	2	3	p(X)
	1	1/4	1/4	0	1/2
X	2	0	1/4	1/4	1/2
	p(Y)	1/4	1/2	1/4	1

$$E[X] = \frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2} \qquad E[Y] = \frac{1}{4} \times 1 + \frac{1}{2} \times 2 + \frac{1}{4} \times 3 = 2$$

$$\mathrm{Cov}(X, Y) = \sum_{(x,y)}\sum p(x, y)(x - E[X])(y - E[Y])$$

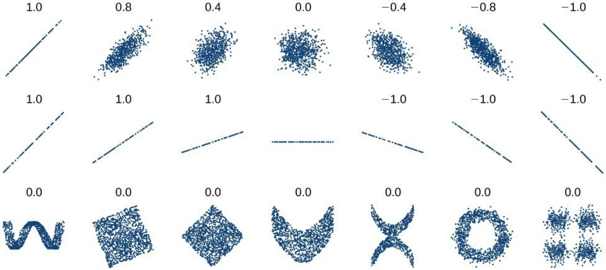
$$= \frac{1}{4}(1 - \frac{3}{2})(1 - 2) + \frac{1}{4}(1 - \frac{3}{2})(2 - 2) + 0(1 - \frac{3}{2})(3 - 2) +$$

$$= 0(2 - \frac{3}{2})(1 - 2) + \frac{1}{4}(2 - \frac{3}{2})(2 - 2) + \frac{1}{4}(2 - \frac{3}{2})(3 - 2) = \frac{1}{4}$$

Correlation Coefficient

- The Correlation coefficient is a quantity that measures the strength of the association (or dependence) between two random variables, e.g., $\mathtt{X}$ and $\mathtt{Y}$.

$$\rho(\mathtt{X},\mathtt{Y}) = \frac{\mathtt{Cov}(\mathtt{X},\mathtt{Y})}{\sigma(\mathtt{X})\sigma(\mathtt{Y})}$$



Probability and Likelihood (1/2)

- Let $\mathtt{X} : \{\mathtt{x}^{(1)}, \mathtt{x}^{(2)}, \cdots, \mathtt{x}^{(m)}\}$ be a discrete random variable drawn independently from a distribution probability $\mathtt{p}$ depending on a parameter θ.
  - For six tosses of a coin, $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$, $\mathtt{h}$: head, and $\mathtt{t}$: tail.
  - Suppose you have a coin with probability θ to land heads and $(1 - \theta)$ to land tails.

- $\mathtt{p}(\mathtt{X} \mid \theta = \frac{2}{3})$ is the probability of $\mathtt{X}$ given $\theta = \frac{2}{3}$.

- $\mathtt{p}(\mathtt{X} = \mathtt{h} \mid \theta)$ is the likelihood of θ given $\mathtt{X} = \mathtt{h}$.

- Likelihood ($\mathtt{L}$): a function of the parameters (θ) of a probability model, given specific observed data, e.g., $\mathtt{X} = \mathtt{h}$.

$$\mathtt{L}(\theta \mid \mathtt{X}) = \mathtt{p}(\mathtt{X} \mid \theta)$$

Probability and Likelihood (2/2)

- The likelihood differs from that of a probability.
- A probability $\mathtt{p}(\mathtt{X} \mid \theta)$ refers to the occurrence of future events.
- A likelihood $\mathtt{L}(\theta \mid \mathtt{X})$ refers to past events with known outcomes.

Maximum Likelihood Estimator

- If samples in $\mathtt{X}$ are independent we have:

$$\begin{aligned} \mathtt{L}(\theta \mid \mathtt{X}) = \mathtt{p}(\mathtt{X} \mid \theta) &= \mathtt{p}(\mathtt{x}^{(1)}, \mathtt{x}^{(2)}, \cdots, \mathtt{x}^{(\mathtt{m})} \mid \theta) \\ &= \mathtt{p}(\mathtt{x}^{(1)} \mid \theta)\mathtt{p}(\mathtt{x}^{(2)} \mid \theta) \cdots \mathtt{p}(\mathtt{x}^{(\mathtt{m})} \mid \theta) = \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta) \end{aligned}$$

- The maximum likelihood estimator (MLE): what is the most likely value of θ given the training set?

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \arg\max_{\theta} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

Maximum Likelihood Estimator - Example

- Six tosses of a coin, with the following model:
  - Possible outcomes: h with probability of θ, and t with probability $(1-\theta)$.
  - Results of coin tosses are independent of one another.

- Data: $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$

- The likelihood is

$$\begin{aligned}
\mathtt{L}(\theta \mid \mathtt{X}) &= \mathtt{p}(\mathtt{X} \mid \theta) \\
&= \mathtt{p}(\mathtt{X}=\mathtt{h} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{h} \mid \theta)\mathtt{p}(\mathtt{X}=\mathtt{t} \mid \theta) \\
&= \theta(1-\theta)(1-\theta)(1-\theta)\theta(1-\theta) \\
&= \theta^2(1-\theta)^4
\end{aligned}$$

- $\hat{\theta}$ is the value of θ that maximizes the likelihood:

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \frac{2}{2+4}$$

Log-Likelihood

- The MLE product is prone to numerical underflow.

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \mathtt{L}(\theta \mid \mathtt{X}) = \arg\max_{\theta} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

- To overcome this problem we can use the logarithm of the likelihood.
  - It does not change its arg max, but transforms a product into a sum.

$$\hat{\theta}_{\mathtt{MLE}} = \arg\max_{\theta} \sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{log}\,\mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$$

Negative Log-Likelihood

- Likelihood: $\mathtt{L}(\theta \mid \mathtt{X}) = \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Log-Likelihood: $\mathtt{log}\mathtt{L}(\theta \mid \mathtt{X}) = \mathtt{log} \prod_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta) = \sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{log}\mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Negative Log-Likelihood: $-\mathtt{log}\mathtt{L}(\theta \mid \mathtt{X}) = -\sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{log}\mathtt{p}(\mathtt{x}^{(\mathtt{i})} \mid \theta)$
- Negative log-likelihood is also called the cross-entropy

Cross-Entropy

- Coss-entropy: quantify the difference (error) between two probability distributions.

- How close is the predicted distribution to the true distribution?

$$\mathtt{H}(\mathtt{p},\mathtt{q}) = -\sum_{\mathtt{x}} \mathtt{p}(\mathtt{x})\mathtt{log}(\mathtt{q}(\mathtt{x}))$$

- Where $\mathtt{p}$ is the true distribution, and $\mathtt{q}$ the predicted distribution.

Cross-Entropy - Example

- Six tosses of a coin: $\mathtt{X} : \{\mathtt{h}, \mathtt{t}, \mathtt{t}, \mathtt{t}, \mathtt{h}, \mathtt{t}\}$
- The true distribution $\mathtt{p}$: $\mathtt{p}(\mathtt{h}) = \frac{2}{6}$ and $\mathtt{p}(\mathtt{t}) = \frac{4}{6}$
- The predicted distribution $\mathtt{q}$: h with probability of θ, and t with probability $(1 - \theta)$.
- Likelihood: $\theta^2(1 - \theta)^4$
- Negative log likelihood: $-\mathtt{log}(\theta^2(1 - \theta)^4) = -2\mathtt{log}(\theta) - 4\mathtt{log}(1 - \theta)$
- Cross entropy: $\mathtt{H}(\mathtt{p}, \mathtt{q}) = -\sum_{\mathtt{x}} \mathtt{p}(\mathtt{x})\mathtt{log}(\mathtt{q}(\mathtt{x}))$
  $= -\mathtt{p}(\mathtt{h})\mathtt{log}(\mathtt{q}(\mathtt{h})) - \mathtt{p}(\mathtt{t})\mathtt{log}(\mathtt{q}(\mathtt{t})) = -\frac{2}{6}\mathtt{log}(\theta) - \frac{4}{6}\mathtt{log}(1 - \theta)$

Summary

Summary

- Logic-based AI, Machine Learning, Deep Learning

- Deep Learning models
  - Deep Feed Forward
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Autoencoders

- Linear algebra and probability
  - Random variables
  - Probability distribution
  - Likelihood
  - Negative log-likelihood and cross-entropy

References

- Ian Goodfellow et al., Deep Learning (Ch. 1, 2, 3)

Questions?

Acknowledgements

Some of the pictures were copied from the book Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurelien Geron, O'Reilly Media, 2017.