00 Introduction

May 21, 2021

Information: *Some basics concepts of Machine Learning.*

Written by: *Zihao Xu*

Last update date:: *05.21.2020*

1 Definition of Artificial Intelligence, Machine Learning, Deep Learning

1.1 Artificial Intelligence:

- A.I. is a very **broad** field.
- Turing test:
  - 'A human tries to tell the difference between a human correspondent and a machine one. When they can't, according to Turing, the machine should be judged intelligent.'
- In Oxford Dictionary
  - 'the theory and development of computers systems **able to perform tasks that normally require human intelligence**, such as visual perception, speech recognition, decision-making, and translation between languages'

1.2 Machine Learning:

- A huge part in the concept of AI (Models, Algorithms, Theory) as well as useful tools for some AI applications (Computer Vision, Robotics).
- **Train an algorithm to reproduce answers from data.**
- Usually can be divided into **supervised learning** and **unsupervised learning**.
- Supervised Learning: To estimate a function between input and output given only **input-output examples**.
  - Examples: Regression, Classification, Filtering, …
- Unsupervised Learning: To **model the input data directly**, or in other words, understand the input data directly.
  - Examples: Clustering, Dimensionality Reduction, Density Estimation, Generative models, …
  - Motivation: Labeling data is always expensive (supervised) while gathering raw data is cheap (unsupervised).
  - Yann LeCun, Head of Facebook AI, 2016
    - 'AI systems today do not possess "common sense", which humans and animals acquire by observing the world, acting in it, and understanding the physical constraints

of it. Some of us see unsupervised learning as the key towards machines with common sense.'

1.3 Deep Learning:

1.3.1 What is Deep Learning?

- A particular successful **Machine Learning method based on deep sequences of neural networks**.
- Instead of handcrafting features in classical Machine Learning, Deep Learning let the deep model do all the feature engineering automatically.
- Generally speaking, deep models are **sequential transformations learned from data**. If you use deep neural networks to define sequential transformations from input to output, then you do Deep Learning.

1.3.2 Successful Applications of Deep Learning

- Any machine learning task:
  - Regression, Classification, Clustering, Dimensionality reduction, Density estimation, Filtering, …
- Natural language processing
- Reinforcement learning
  - self-driving cars, playing Atari games, AlphaGo
- Design of engineering systems

1.3.3 Why does Deep Learning work so well?

- The universal approximation theorem (Cybenko, 1989)
- Deep Learning can automatically learn **a hierarchy of representations of high-dimensional data**.
- Advanced learning algorithms (back-propagation, stochastic gradient descent, etc.)
- Advanced software (tensorflow, pytorch) and advanced hardware (GPU)

1.3.4 Problems of applying Deep Learning in engineering

- Many approaches are **black boxes** with no guarantees on convergence and performance.
- Deep Models would make errors if data are perturbed.

2 General Machine Learning problems

2.1 Supervised Learning

2.1.1 Problem Definition

- **Mathematical Representation**

$$y = f_{gt}(x)$$

  - x: Input of the function
    - Called features, attributes, covariates or variables
    - Can be numeric, categorical, discrete or nominal
    - Examples:

        - [GRE scores, GPA, major]
        - An image
        - A sentence
        - A d-dimentional vector of numbers
    - y: Output of the function
        - Called output, response, target, or label
        - Can be numeric, categorical, discrete or nominal
            - The problem is known as Regression when output is numeric
            - The problem is known as Classification when the output is categorical
    - f_{gt}: Ground truth of the function to be estimated
        - We know nothing about this function, such as its structure, type or basis functions.
        - It stands for the true relationships between the input and output.

- **Goal**:
    - Capture the **features** of f and make use of it, so that the ability to inferring corresponding y from x is obtained.

2.1.2 Machine Learning Approach

- **Mathematical Representation**

$$\hat{y} = \hat{f}_\theta(x)$$

    - x: Input of the function.
    - $\hat{y}$: Output of the estimated function. Usually called predictions.
    - $\hat{f}_\theta$: Estimations of the ground truth function. Consists of basic functions we're familiar with and can be tuned with the parameter θ.

- **Approach Explanation**:
    - Given a set of input-output pairs which is called **Training Set** and denoted by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, tune the parameter θ to achieve the "best" or at least a "good"answer, which means trying to make the prediction/s $\hat{y}$ as close as possible to the ground truth y
        - "Training set: A set of examples used for learning, which is to fit the parameters [i.e., weights] of the classifier."

- **Possible solution for function basis**:
    - Support vector machines
    - Radial basis functions
    - Gaussian mixture functions
    - Neural networks
        - Very high capacity order
        - Easy to train with modern computational tools
        - Shallow neural networks: Use one easy-to-train layer
        - Deep neural networks: Train a hierarchical stack of layers.
    - ...

- **How to define the "best performance"?**

  - Measure the "distance" between ground truth y and estimation $\hat{y} = \hat{f}_\theta(x)$
  - The function to measure this "distance" is usually called **loss function** and is denoted by $l(\theta)$
  - Largely depends on specific tasks and the output type. The required performance should be guaranteed when the loss is minimized.
    * Regression: Mean Square Error
    * Classification: Cross-Entropy Loss
    * Localization: Intersection over Union
    * ...

- **How to determine the best value of θ?**
  - Select θ which minimizes the loss function
  - Mathematically speaking: $\theta^* = \underset{\theta}{\operatorname{argmin}}\{l(\theta)\}$
  - Finding this best parameter θ^* is completed with the help of modern optimization algorithms such as gradient descent and expectation maximization.

- **How to optimize the options set manually when building the structure of estimation function?**
  - Given another set of input-output pairs which is usually called **Validation Set**, test the performance of the learned model ($\hat{y} = \hat{f}_{\theta^*}(x)$) with different options and set the option to the one has the best performance.
    * "Validation set: A set of examples used to tune the parameters (i.e., architecture, not weights) of a classifier, for example to choose the number of hidden units in a neural network."

- **How to measure the final performance of the estimation?**
  - Given another set of input-output pairs which is usually called **Test Set**, check the performance either by watching the visualized results or calculate the loss function on this data set.
    * "Test Set: A set of examples used only to assess the performance (generalization) of a fully specified classifier."

- **Supervised Learning Approach in summary**
  1. Collect train set which contains input-output pairs
  2. Determine the function basis (model structure)
  3. Determine the loss function needs to be optimized
  4. Train the model by finding out the best θ which minimized the loss function
  5. Train the model under different hyper-parameters set manually and select the best hyper-parameters based on its performance on validation set
  6. Test such a well specific model on test set

2.2 Unsupervised Learning

2.2.1 Problem Definition

- In supervised learning, the objective is the mappings between input data x and output (or label) y. However, in unsupervised learning, the input data do not have a label while some patterns or features are supposed to be implicitly or explicitly included in the data set. Then

the objective is to get the ability of **finding out these patterns or features** without any prior information.

2.2.2 Machine Learning Approach

- In unsupervised learning, the **training set** is only a set of input values $\mathcal{D} = \{(x_i)\}_{i=1}^n$. Thus, in my understanding, common unsupervised learning approach do not share similar procedures (PCA, clustering, generating). It depends on what specific features or results we want to get even though sometimes the same optimization algorithms in supervised learning are used.

2.3 No Free Lunch Theorem

- All models are approximations
- All models make assumptions
- Assumptions are never perfect
- While a general-purpose alogorithm has average performance on all types of problem, a highly specialized algorithm can reach a incredibly high performance on a specific type of problem, sacrificing its performance on all the other type of problems.

3 A single layer Neural Network

- **General definition**

$$\hat{y} = B\sigma(Ax + b)$$

  - A: A matrix of multiplicative weights
  - b: A column vector of additive offsets
  - σ: A point-wise activation function
  - B: A matrix of multiplicative weights
- **Why does this structure work?**
  - Universal Approximation Theorem (Cybenko,1989)
    - Fix a continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ (activation function) and positive integers d, D. The function σ is not a polynomial if and only if, for every continuous function $f : \mathbb{R}^d \to \mathbb{R}^D$ (target function), every compact subset K of $\mathbb{R}^d$, and every $\epsilon > 0$ there exists a continuous function $f_\epsilon : \mathbb{R}^d \to \mathbb{R}^D$ (the layer output) with representation

$$f_\epsilon = W_2 \circ \sigma \circ W_1$$

      where W_2, W_1 are composable affine maps and $\circ$ denotes component-wise composition, such that the approximation bound

$$\sup_{x \in K} ||f(x) - f_\epsilon(x)|| < \epsilon$$

      holds for any ϵ arbitrarily small (distance from f to f_ϵ can be infinitely small).
  - The point-wise activation function provides the network with the ability to approximate non-linearities. Otherwise, the combination of linear transformation would also be a linear transformation.
- **Common activation functions**
  - Logistic sigmoid function:

    * $\sigma_i(z) = \frac{1}{1+e^{-z_i}}$
- Rectified linear unit (ReLU):
    * $\sigma_i(z) = \begin{cases} 0 & if\ z_i \leq 0 \\ z_i & if\ z_i > 0 \end{cases}$
- Leaky ReLU
    * $\sigma_i(z) = \begin{cases} \alpha z_i & if\ z_i \leq 0 \\ z_i & if\ z_i > 0 \end{cases}$
- Softmax
    * $\sigma_i(z) = \frac{e^{z_i}}{\Sigma_j e^{z_j}}$