CHAPTER 1: WHAT IS DEEP LEARNING?

Machine Learning

1. Symbolic AI: fake one
2. 3 things of Machine Learning: Input data/ Expected Output/ Way to measure how good is algorithm
3. Machine-learning models are all about finding appropriate representations for their input data (Ex: new coordinate system...)
4. **Learning**: in context of ML, is an automatic search process for better representations.
5. Operation can be coordinate changes, linear projections, translations, nonlinear operations...
6. ML doesn’t create new representation, they just merely search for predefinded set of operations:

**Hypothesis space**

Deep Learning

1. New take on learning representations from data that putss an emphasis on learning successive layers of increasingly meaningful representations.

2. **Deep:** successive layers of representations. (depth: how many layers)

3. Also called **layered representations learning** or **hierarchical representations learning**

4. **Shallow learning** just has 1 or 2 layers

5. **Neural network:** literal layers stacked on top of each other.

6. DL is mathematical framework for learning representations from data.

7. Deep network as a multistage information-distillation operation

8. Weights (number): specification of what layers do to their input datas. Or called parameters of a layers.

9. **Learning:** in this context, means finding a set of values for the weights of all layers in a network.

10. DNN can contain tens of millions of parameters!!! And modify one can affect others.

11. First observe, then control. To control the output of NN, need to measure how far from expected.

12. **=> Loss function (objective function)**: takes the outputs of NN and expected outputs, computes the **distance score** => how good of NN => Used as **Feedback signal** => adjust weights to lower **Loss score**.

13. **Optimizer**: is used in adjustment to lower loss score.

14. **Backpropagation** algorithm, the above process

15. Initial, weights are random number => imprement random transformations => high loss score

16. **Training loop**: repeats sufficient number of time (around 10 of thousands of examples).

17. Image classification/ speech recognition/ Handwriting transcription/ Improved machine translation/ Improved text-to-speech conversion/ Digital assistants/ Autonomous driving/ Improved ad targeting/ Search research/ Ability to answer natural-language questions/ Go playing

18. **Probabilistic modeling:** is the application of the principles of stat to data analysis. Best known algorithms is **Naïve Bayes**

19. **logistic regression** like hello world

20. **Kernel method** are a group of classification algorithms, best known is **Support vector machine (SVM).** It finds **decision boundaries.**

21. 2 steps: 1- Find high-dimensional representation (hyperplan)

2- Maximize the distance between the hyperplane and closest data points from each class. Or **maximizing the margin**.

22. **Kernel trick** when high-dimensional representation is computationally intractable. => compute the distance between pairs of points in that space using **kernel function** (faster and more efficiency) – now crafted by hand).

23. SVMs is good with small dataset, but hard to scale (shallow method) (no good for image classification). In order to do it, need to extract useful representations manually (**feature engineering).**

24. **Decision trees** (flowchart structure) (kernel methods) (input => output by branches).

25. **Random forest** algorithm introduce a robust, practical take on decision-tree => building large number of specialized decision trees and ensembling their outputs. Considered as **second best** algorithm for any shallow ML task.

26. **Gradient boosting machines** is better than RF (like). ML technique based on ensembling weak prediction models. It iteratively training new models that specialize in addressing the weak points of the previous models. => **the best** algorithm in **Kaggle.**

27. **Convolutional neural network (CNN) (convnets)** becomes go to algorithm for all computer vision task (perceptual tasks)

CHAPTER 2: MATHEMATICAL BUILDING BLOCK OF NN

**A FIRST LOOK AT NN**

1. **NIST (MNIST or National Institute of Standards and Technology):** like hello world i=of DL => verify if algorithm works as intended.

from keras.datasets import mnist

(training\_set), (test\_set) = mist.load\_data()

1. In classification: category = class, datapoints = samples, class w/ specific sample = label

Ex: Images will be encoded in Numpy arrays, label will be a array of digits (0 to 9) => handwriting problem

1. Workflow:
   1. Feed NN training data
   2. NN will learn to associate images and labels
   3. Use it on test data
2. The core building block of NN is the layer, a data-processing module like a filter for data
3. **Layers** extract representations out of the data fed into them

Combine simple layers => form of progressive **data distillation**

1. A DL model is like a sieve for data processing, made of succession of increasingly refinded data filter- the layers

from keras import models

from keras import layers

network = models.Sequential()

network.add(layers.Dense(512, activation='relu', input\_shape=(28 \* 28,)))

network.add(layers.Dense(10, activation='softmax'))

1. **2 Dense** layers (densely/fully connected), the 2nd layers is 10-way softmax layer => return an **array** of 10 probability scores (summing to 1). Each score => probability that current digit image belongs to.
2. To make NN ready for training: 3 more things: **loss function/optimizer/metric to monitor process**