Chapter 1

Theoretical Background

DEEP LEARNING, *a class of Machine Learning (ML) algorithms based on neural networks (NNs)*, has revolutionized the way we tackle a problem from a ML perspective and is one if not the most important factor for recent ML achievements. Solving complex tasks such image classification or language translation, that for years have bedevilled traditional ML algorithms, constitutes the signature of Deep Learning (DL). Admittedly, *the advent of a deep convolutional neural network (CNN)*, the `AlexNet` (**alexnet**) on September 30 of 2012, signified the "modern birthday" of this field. On this day, `AlexNet` not only won the `ImageNet` (**Deng_2009**) Large Scale Visual Recognition Challenge (ILSVRC), but dominated it, achieving a top-5 accuracy of 85%, surpassing the runner-up which achieved a top-5 accuracy of 75%. `AlexNet` showed that NNs are not merely a pipe-dream, but they can be applied in real-world problems. It is worth to notice that ideas of NNs trace back to 1943, but it was until recently that these ideas got materialized. The reason for this recent breakthrough of DL (and ML) is twofold. First, the availability of large datasets—the era of big data—such as `ImagNet`. Second, the increase in computational power, mainly of GPUs for DL, accelerating the training of deep NNs and traditional ML algorithms.

1.1 Machine Learning Preliminaries

Since DL is a subfield of ML, it is necessary to familiarize with the later before diving into the former. In this section, the minimum theoretical background and jargon of ML is presented. Machine Learning can be defined as *"the science and (art) of programming computers so they can learn from the data"* (**ml**). A more technical definition is the following:

DEFINITION 1.1 (Machine learning, **mitchell**). *A computer program is said to learn from experience **E** with respect to some class of tasks **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**.*

For instance, a computer program that classifies emails into spam and non-spam (the task **T**), can improve its accuracy, i.e. the percentage of correctly classified emails (the performance **P**), through examples of spam and non-spam emails (the experience **E**). But in order to take advantage of the experience (aka *data*), it must be written in such a way that *adapts to the patterns in the data*. Certainly, a *traditional spam filter can not learn from experience*, since the latter does not affect the classification rules of the former and as such, its performance. For a traditional spam filter to adapt to new patterns and perform better, it must change its hard-wired rules, but by then it will be a different program. In contrast, a *ML-based filter can adapt to new patterns, simply because it has been programmed to do so*. In other words, *in traditional programming we write rules for solving **T** whereas **in ML we write rules***

to learn the rules *for solving* ***T***. This subtle but essential difference is what gives ML algorithms the ability to take advantage of the data.

1.1.1 Learning paradigms

Depending on the type of experience they are allowed to have during their *training phase*, ML approaches are divided into three main *learning paradigms*: ***unsupervised learning***, ***supervised learning*** *and* ***reinforcement learning***. The following definitions are not by any means formal, but merely serve as an intuitive description of the different paradigms.

Definition 1.2 (Unsupervised learning). *The experience comes in the form $\mathcal{D}_{train} = \{\boldsymbol{x}_i\}$, where $\boldsymbol{x}_i \sim p(\boldsymbol{x})$ is the input of the i-th training instance. In this paradigm we are interested in learning useful properties of the underlying structure captured by $p(\boldsymbol{x})$.*

For example, suppose we are interested in generating images that look like Picasso paintings. In this case, the input is just the pixel values, i.e. $\boldsymbol{x} \in \mathbb{R}^{W \times H \times 3}$. The latter follow a distribution $p(\boldsymbol{x})$, so all we have to do is to train an unsupervised learning algorithm with many Picasso paintings to get a *model*, that is $\hat{p}(\boldsymbol{x})$. Assuming the estimation of the original distribution is good enough, new realistically looking paintings (with respect to original Picasso paintings) can be "drawn" by just sampling from $\hat{p}(\boldsymbol{x})$. In the ML parlance, inputs are also called *features, predictors or descriptors*.

Definition 1.3 (Supervised learning). *The experience comes in the form $\mathcal{D}_{train} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, where $(\boldsymbol{x}_i, \boldsymbol{y}_i) \sim p(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{y}_i$ is the output aka label of the i-th training instance. In this paradigm we are usually interested in learning a function $f \colon X \to Y$.*

This paradigm comes mainly under two flavors: *regression and classification.* In regression the interest is in predicting a continuous value given an input, i.e. $y \in \mathbb{R}$, such as a molecular property given a mathematical representation of a molecule. In classification, the interest is to predict in which of k classes an input belongs to, i.e. $y \in \{1, \ldots, k\}$, such as predicting the breed of a dog image given the raw pixel values of the image. The term "supervised" is coined due to the "human supervision" the algorithm receives during its training phase, through the presence of the correct answer (the label) in the experience. In a sense, in this paradigm we "teach" the learning algorithm aka *learner*. It should be emphasized that the label is not constrained to be single-valued, but can also be multi-valued. In this case, one talks about *multi-label regression or classification* (**Read_2009**).

A more exotic form of supervised learning is *conditional generative modelling*, where the interest is in estimating $p(\boldsymbol{x} \mid \boldsymbol{y})$. For example, one may want to build a model that generates images of a specific category or a *model that designs molecules/materials with tailored properties*. This is one approach of how ML can solve the *inverse design problem*.

Index