Lecture Notes on Introduction to Artificial Neural Networks and Deep Learning (FYTN14/EXTQ40/NTF005F)

Mattias Ohlsson and Patrik Edén
Computational Biology and Biological Physics
Department of Astronomy and Theoretical Physics
Lund University

Fall 2019

Chapter 1

Preface

This course is intended to give an introduction to artificial neural networks and deep learning. Artificial neural networks (ANN) represents a technology that has connections to many disciplines such as neuroscience, computer science, mathematics, statistics and physics. This course will provide you with enough knowledge to use neural networks and deep learning in practical applications.

We will study ANNs from both a theoretical and a computational/practical point of view. The course material is divided into different parts:

1. Introduction
2. Feed-forward neural networks
3. Ensemble machines
4. CNN, Autoencoder and GAN
5. Recurrent neural networks
6. Self-organizing neural networks

There are two computer exercises in this course. The exercises are application oriented, meaning that we will focus on the methods/application and not so much on the programming part. The amount of programming required is therefore limited.

These notes are meant as a complement to the lectures. The lectures will also use material from the "Deep Learning Book" (DLB) found at www.deeplearningbook.org

Chapter 2

Introduction

Read the introduction chapter of the “Deep Learning Book” (DLB)[1].

2.1 Introduction to machine learning

From chapter 5 in the DLB book, read the following sections:

- “Intro”
- 5.1 Learning Algorithms
- 5.2 Capacity, Overfitting and Underfitting (pages 108-114)
- 5.11.1 The Curse of Dimensionality

One of the most important features of artificial neural networks (ANN) is the ability to learn from data. The DLB talks about three ingredients, the experience E, the task T and the performance measure P.

- The task. There are many tasks in machine learning, such as *classification, regression, sequence prediction, imputation of missing values* etc, where classification problems are very popular.
- The performance measure. We need to evaluate the performance of the machine learning algorithm. The performance measure is often specific to the task. One also can have different performance measures during training and when the model is actually used. During training one often uses an error measure instead of a performance measure.

[1] www.deeplearningbook.org

- The experience. To simplify we can say that the experience often comes as a dataset i.e. a collection of examples.

Learning can broadly be divided into two categories, **unsupervised** and **supervised** learning. During learning we have access to data (experience). The data is usually a set of examples or data points. Sometimes each data point comes with a label (eg. an image of a handwritten digit can have label "five"). Sometimes data comes without labels and we simply just have the numerical values for each data point. For supervised learning we have access to both data points and labels, but for unsupervised learning we only have the data points.

There are other learning process, such as semi-supervised learning, multi-instance learning, one-shot learning, reinforcement learning. In this course we will only look at supervised and unsupervised learning.

See also separate powerpoint presentation "Introduction to machine learning".

2.2 What is an artificial neural network?

Artificial neural networks (ANN) is a collection name for many algorithms that (almost) all have in common that they are inspired from the construction and functioning of the human brain. Haykin[2] defines an artificial neural network as:

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two aspects:

1. *Knowledge is acquired by the network from its environment through a learning process.*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*

Note that an ANN is by no means a simulation of the whole or some part of the brain. We continue this introduction with a small description of the biological neuron.

2.3 The biological neuron

Our brain consists of about 10^{11} neurons. We can say that they are the elementary computational nodes of our brain. There are many different types of neurons, where a typical

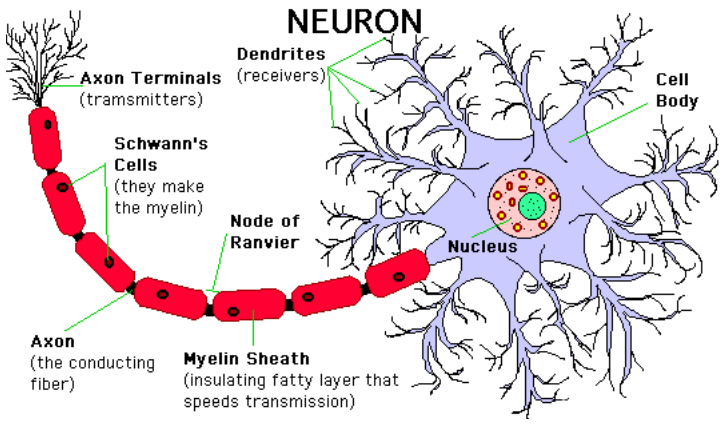

Figure 2.1: Illustration of a *pyramidal* cortical neuron.

so called pyramidal cell is shown in figure 2.1. This neuron consists of several parts. In a simple description they are:

- The cell body: Here is the cell nucleus located.
- Dendrites: Tree like constructions are connected to the cell body. They are called dendrites and can be viewed as receivers of signals from other neurons.
- Axon: Extending from the cell body is a long nerve fiber called the axon. The axon can be viewed as the transmitter of generated signals from the cell body.
- Synaptic terminals (synapses): The axon eventually branches off and forms a tree of transmitter ends called synaptic junctions or synapses. The connections to other receivers are dendrites or the cell bodies themselves. The strength of the synapses can vary and the connections can be both inhibitory and excitatory.

The fact that

- we have about 10^{11} neurons in our brain
- there are about 10^3 connections per neuron giving a total of about 10^{14} synapses in our brain
- signals travel up to 120 m/s in humans

makes our brain is a highly complex, non-linear and parallel information-processing system.

[2]Simon Haykin, “Neural Networks: A Comprehensive Foundation”, (1998)

2.4 Artificial neural networks

2.4.1 Models of a neuron

Figure 2.2 shows the basic element of an ANN, the artificial neuron. The output of the

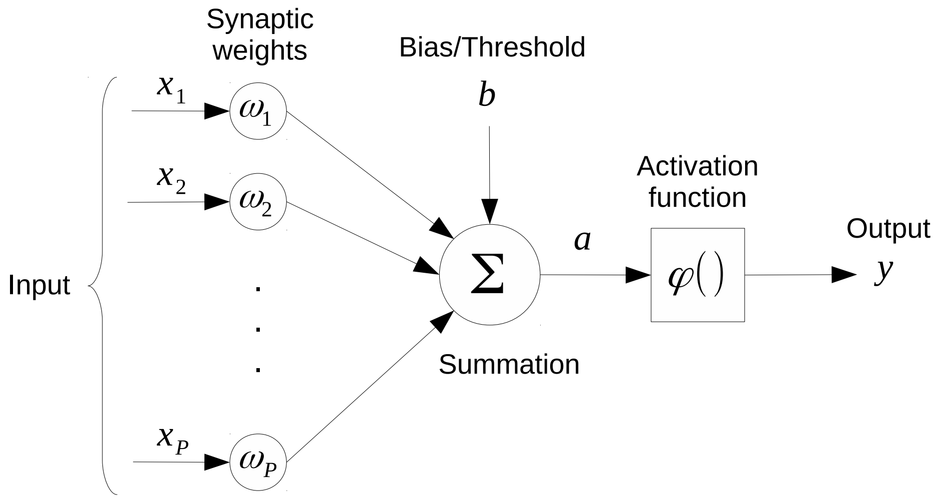

Figure 2.2: The basic element of the ANN, the neuron

neuron is calculated as

$$a = \sum_{k=1}^{P} \omega_k x_k + b$$
$$y = \varphi(a)$$

So what neurons do is a weighted summation of its incoming signals and then outputs a new signal. The weights ω_k can be both positive and negative. The function $\varphi(x)$ have many names, e.g. transfer function, activation function or node function. In these notes the name activation function will be used. Below is a list of a few activation functions and in figure 2.3 they are plotted.

1. Linear, $\varphi(x) = x$

2. Threshold function

$$\varphi(x) = \begin{cases} 0 & if \quad x < 0; \\ 1 & if \quad x \geq 0; \end{cases}$$

3. Logistic function
$$\varphi(x) = \frac{1}{1+e^{-x}}$$

4. Hyperbolic tangent, $\varphi(x) = \tanh(x)$

5. Rectifier function
$$\varphi(x) = \begin{cases} 0 & if \quad x < 0; \\ x & if \quad x \geq 0; \end{cases}$$

6. Softplus, $\varphi(x) = \log(1 + e^{x})$

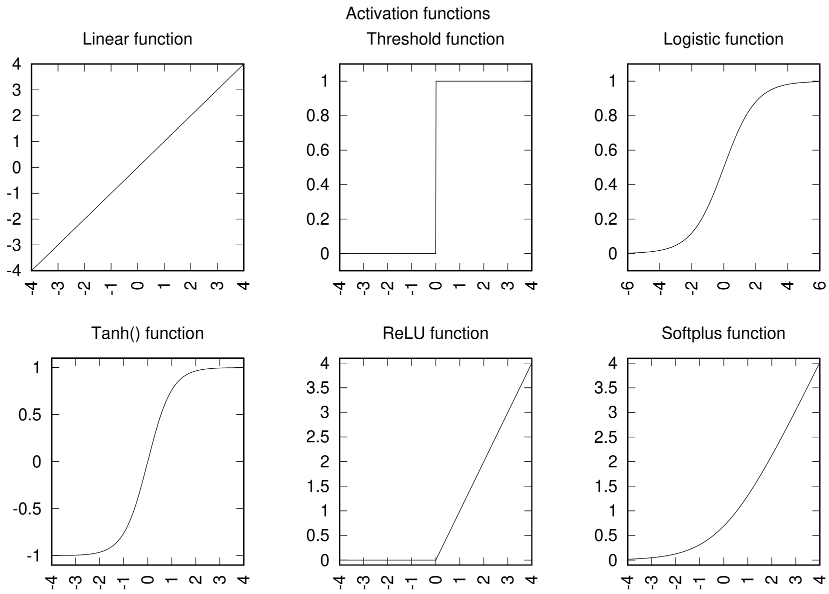

Figure 2.3: Examples of activation functions.

The derivatives of activation functions are often needed. For example, for the $tanh()$ we have
$$\varphi(x) = \tanh(x) \Rightarrow \varphi'(x) = 1 - \tanh(x)^2 = 1 - \varphi(x)^2$$

2.4.2 Network architectures

How do we connect all these neurons with each other? There are two major groups:

1. Feed-forward architectures

2. Feed-back architectures

Feed-forward networks

Figure 2.4 shows three examples of feed-forward networks.

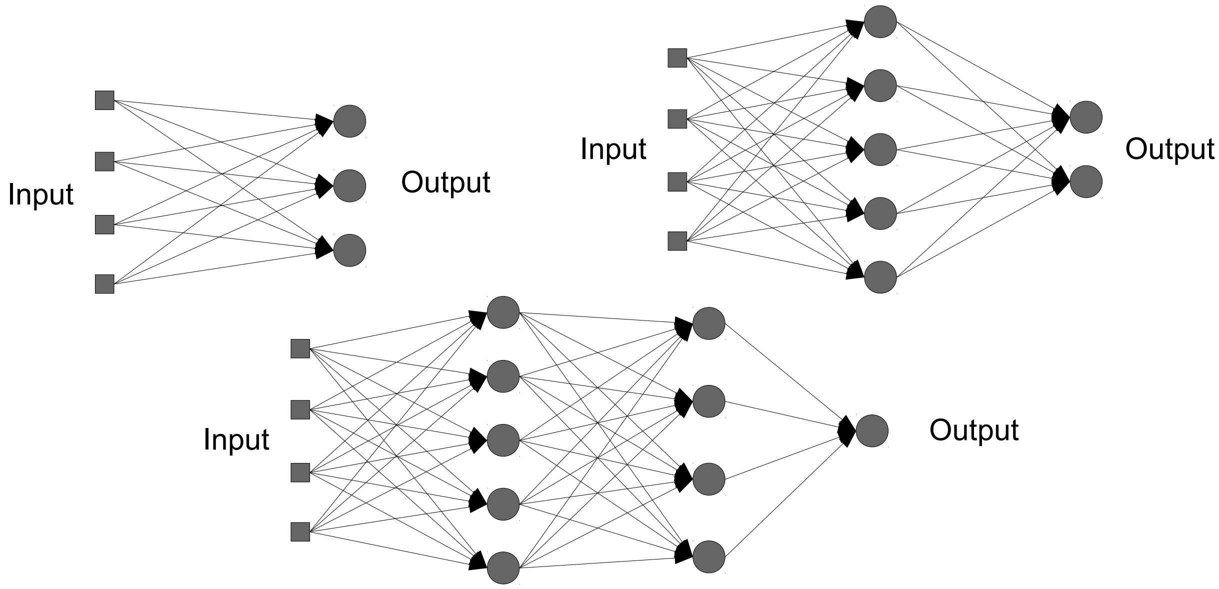

Figure 2.4: (Top left) A feed-forward network with only one layer of computational nodes. (Top right) A feed-forward network with two layers of computational nodes. (Bottom) Finally a feed-forward network with three layers of computational nodes.

- We say that it is a feed-forward type of networks since signals are entering from the left and then propagates through the network to the right.
- The feed-forward network can be fully connected or partially connected.
- There can be skip-layer weights e.g. weights connecting input neurons with output neurons.

Recurrent (feed-back) networks

Figure 2.5 shows two examples of networks with feed-back connections.

- Networks with feed-back connections are often used to model sequence data.
- Feed-back connections can occur "locally", meaning a specific node feeds into itself or "globally", meaning nodes can feed back into nodes several layers "earlier" in the computational graph.

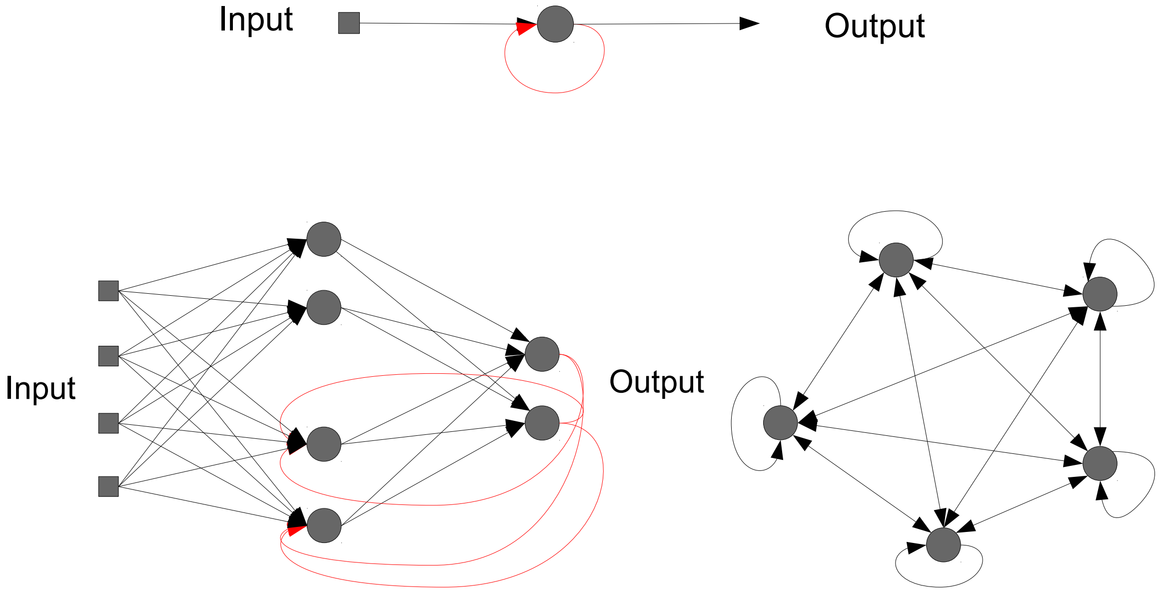

Figure 2.5: (Top) A single node with a feed-back connection (red). (Lower left) A neural network with mainly feed-forward and a few feed-back connections (red). The bottom two nodes in the first layer receives inputs from the two output nodes. (Lower right) A fully recurrent network with 5 nodes.