Workshop 6

Learning From Big Data

Artificial Neural Network, Keras and Tensorflow

*Please do the exercises or tasks without "Optional" mark at first. After that, if you still have some time, please try the tasks with "Optional".

- It takes considerable time to learn TensorFlow, as it's a big software library and a plenty of APIs. Comparably, Keras is more user friendly and easy to master its basics.

- Therefore, we emphasize on using Keras in this Lab.

Part 1: Working with Keras

In Session 3, we have learned how to perform a classification by using machine learning model. Alternatively, in this exercises you are going to employ an Artificial Neural Networks (ANN) for a classification .

We will implement our Deep Learning models with the Keras library.

- The goal of this exercises is to create a demographic segmentation model to tell the bank which of their customers are at highest risk of leaving. So, we will have a binary outcome, "true" (leaving) and "false" (stay). This is a classification problem, given a large amount of records of the customers.

- A notebook, called ann.ipynb, is given in …/Lab6, demonstrating how to build an ANN to solve this problem.
- Instead of my lecture to explain this implementation, I have put comments or text, including figures and web-links, in each cell of the notebook to interpret the codes, associated with the Keras APIs syntax.

Actions:

- Study this sample one cell by one cell, following the Python codes and comments.
- Run each cell in sequence while making sure that you understand each statement in *Python* and its output.
- Try to answer my questions, presented in the notebook.

Part 2: Working with TensorFlow (Optional)

Task 1: Running with TensorFlow

TensorFlow is a powerful open source software library for numerical computation, particularly well suited and fine-tuned for large-scale Machine Learning (and Deep Learning with Artificial Neural Networks). Let's have the following exercises:

Task 1: TensorFlow basics

There is a basic tutorial page on TensorFlow's web site, as below.
https://www.tensorflow.org/get_started/get_started

1). Open it and study its content
2). Pay attentions to the concept of **"Tensor"**, and how to build and run a **graph**, as well as tf.train API.

Task 2: Coding with TensorFlow

1) Open the notebook, running_with_tensorflow.ipynb, given in …/Lab_8/Lab8_tasks, have a practice with the codes, titled "Creating and Running Graph", which was lectured.

2) Review Section 2 "Implementing *Gradient Descent* with TensorFlow" of Chapter 14 in the lecture slides and then, study the codes in the notebook, following the titles:

   - Using the Normal Equation
   - Using Batch Gradient Descent
   - Feeding data to the training algorithm

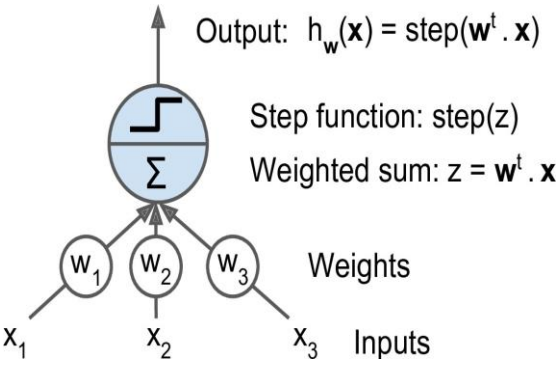
3) (**Optional**) TensorFlow is a big software, that takes some time to learn and practice. Here is its comprehensive tutorial site: https://www.tensorflow.org/tutorials/, you may have a look at it.

Task 2: Building Artificial Neural Networks with TensorFlow

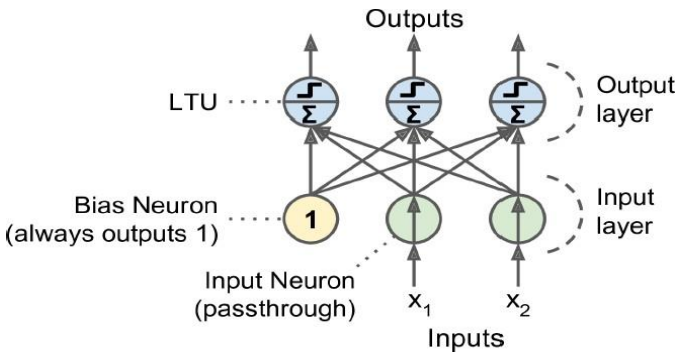
As lectured, ANNs are at the very core of Deep Learning. They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks.

1. Perceptron

The **Perceptron** is one of the simplest ANN architectures. It is based on a slightly different artificial neuron called **a** linear threshold unit (**LTU**), as shown:



A **Perceptron** with two inputs and three outputs is represented as above. This Perceptron can classify instances simultaneously into three different binary classes, which makes it a multioutput classifier.



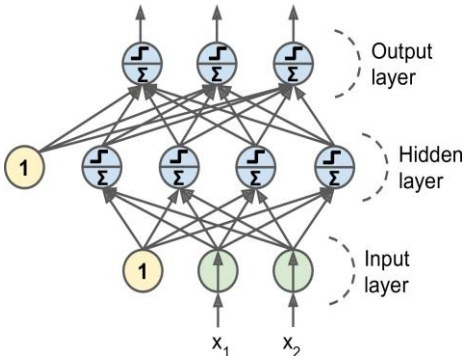
Actions:

1. Review Section 3 "Perceptron" of Chapter 15, make sure of understanding its definition or architecture and "perceptron learning role" at first.

2. Study the codes titled "Perceptron" in notebook artificial_neural_networks.ipynn, given in …LAB_8/LAB8_TASKS, and understand how to use it to classify different iris flowers.

2. Multi-Layer Perceptron (MLP)

In Section 4 "Multi-Layer Perceptron (MLP) and Backpropagation", a very important concept of "Deep Neurol Network" (DNN) was introduced as shown in



Actions:

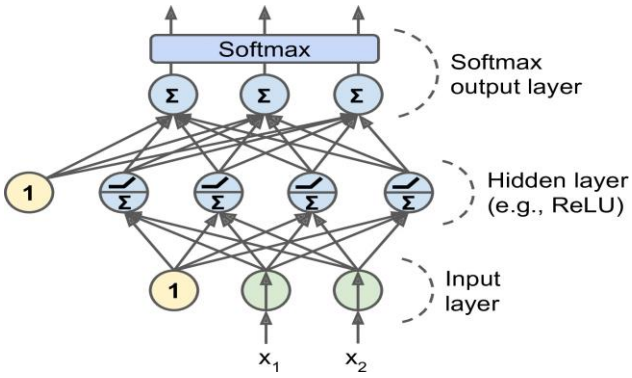
1. Review the slides in Section 4 "Multi-Layer Perceptron (MLP) and Backpropagation" and understand "backpropagation training algorithm", as well as four different "activation functions".

2. Simply look at the diagrams of the four activation functions, plotted by the codes in notebook, artificial_neural_networks.ipynn. You may change or modify some the values of parameters and see what happens in the diagrams.

3. MLP with ReLU and softmax functions for classification

An MLP is often used for classification, with each output corresponding to a different binary class (e.g., spam/ham, urgent/not-urgent, and so on). When the classes are exclusive (e.g., classes 0 through 9 for digit image classification), the output layer is typically modified by replacing the individual activation functions by a shared **softmax function.**

As lectured, a modern MLP (including ReLU and softmax) for classification is shown as below:



The output of each neuron corresponds to the estimated probability of the corresponding class. Note that the signal flows only in one direction (from the inputs to the outputs), so this architecture is an example of a feedforward neural network **(FNN).**

In order to build the model as above, the codes for implementing "Minibatch Gradient Descent to train it on the MNIST dataset" are given in notebook artificial_neural_networks.ipynb.

Actions:

- Study a part of the codes in the notebook, titled "FNN for MNIST", and see how to create and evaluate the model, how to train the model with "Minibatch Gradient Descent" by using MNIST dataset.