Installation and setup instructions for the course CS-456 Artificial Neural Networks

March 26, 2021

1 Overview

This document describes the setup of an coding environment for the miniprojects of the CS456 course. Following the instructions in this document will help you with installing all the requirements and keeping a clean coding environment. You are asked to set up an virtual `python 3` environment using `Conda`. You should always use this environment when working on the miniprojects of this course since it manages all the dependencies of your code, avoids conflicts with other installations on your machine and increases reproducibility/continuous integration of your code.

The following procedure works on Unix based operating systems (Linux/Mac). If you are a Windows user, we recommend to setup a Linux partition on your machine.

2 Setup and Installation

2.1 Installation or update of (Mini)conda

If you have not yet a `Conda` version installed (you can check by typing `$ conda` in a terminal), please install `Conda` or the lightweight version `miniconda` `https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html`. In case you already have `Conda` installed please update it to the latest version by running `$ conda update conda` in a terminal.

2.2 Creation of the conda environment for this course

- Create the environment. Run the following in a terminal:
  `$ conda create -n CS456 python=3.7`
- Activate the new environment:
  `$ conda activate CS456`
- For deactivating the environment use:
  `$ conda deactivate`

2.3 Installation of packages

Activate your CS456 environment. Then follow the below points.

- Update pip and install tensorflow and other packages:
  `$ pip install --upgrade pip`
- Install the packages we need for this course:
  `$ pip install tensorflow tensorflow-datasets keras`
  `$ pip install -U matplotlib`

2.4 Installing and handling of jupyter notebooks

Still in your environment, install `jupyter` and register the kernel in the environment:

- Installation:
  `$ pip install jupyter`
  `$ ipython kernel install --user --name CS456`

- To launch a `jupyter notebook` type the following in your terminal:
  `$ jupyter notebook &`

  Now a dashboard should open in a browser which allows you to create a new `jupyter notebook` or to open an existing one. Be sure to select the correct CS456 `python 3` kernel.

3 Testing setup with an XOR toy example

Test your CS456 environment and `tensorflow 2` setup with a small toy example. To do so you will implement an artificial neural network (ANN) that models an XOR gate `https://en.wikipedia.org/wiki/XOR_gate`. This exercise will allow you to get use to symbolic programming and the Keras frontend. Launch an empty `jupyter notebook` within the CS456 environment and follow the steps below.

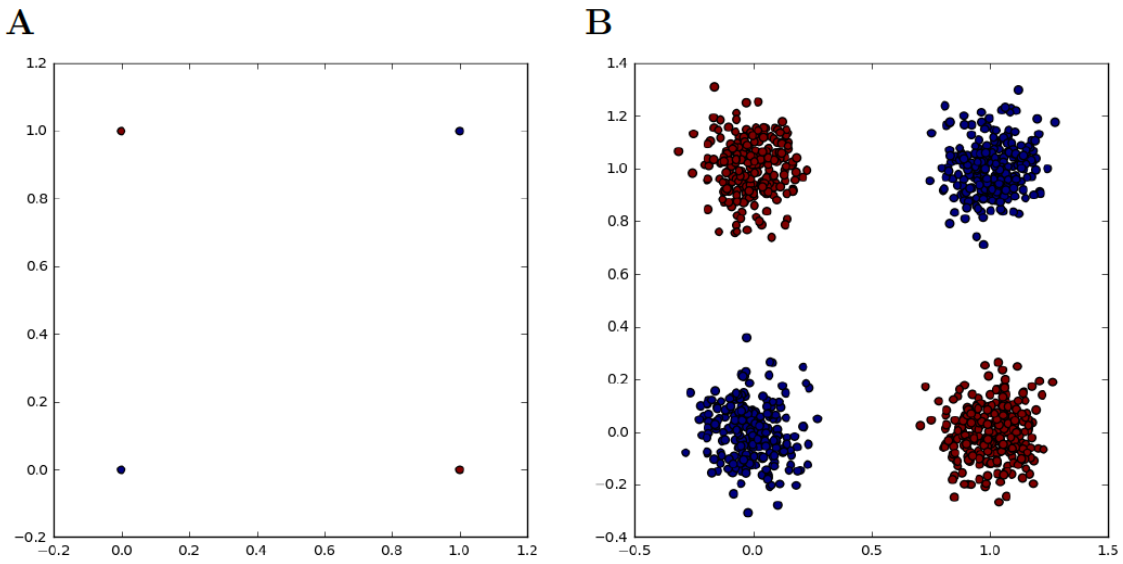

Figure 1: XOR classification task. Graphical visualization of the XOR problem. **A** XOR problem without noise $\sigma = 0$. **B** XOR problem with Gaussian noise $\sigma = 0.1$.

- Read and follow the `Keras` tutorial on the XOR problem: `https://goo.gl/pUKAYL`. The tutorial was written for `tensorflow 1` and `python 2`. Since we are using `tensorflow 2` we should import `keras` functions via `tensorflow 2`:
  `$ from tensorflow.keras.models import Sequential`
  `$ from tensorflow.keras.layers import Dense`
  You might also want to change the `print` prompt of `python 2` to the `print()` function of `python 3`
- Augment the XOR problem with noisy data as in Figure 1. Try several combinations of architecture, optimizer, and activation functions to achieve good performances. Report the network performances (loss and accuracy) in a table. Hint: For robustness of your parameter search, use split sets (training, validation, testing) and cross validation.
- Plot the learning curves (loss and accuracy) and visualize the data and network pre- dictions. Plot the separating surface. Hint: Uniformly sample the input space to determine the separating surface.