Neural Networks and Fuzzy Logic For Tinkers

The course team
June 16, 2017

Birla Institute of Technology and Science, Pilani

Contents

1 Introduction

Machine learning is a powerful mathematical tool to transform data into useful knowledge. But under the hood machine learning is just clever linear algebra, essentially mathematics. Just like any other form of mathematics, it is no spectator's sports.

This document outlines suggested changes in the structure of the course, Neural Networks and Fuzzy Logic. The changes are planned so as to put the students behind the wheels and help them grow while tinkering with the course material.

2 Timeline

The following represents a rough timeline for the various suggested components.

2nd August	Pre-course survey.
7th August	Hands-on-coding session 0: Introduction to Python and Numpy.
14th August	Assignment 0 release: Introduction to Matrix operations and Fuzzy logic.
23rd August	Assignment 0 Submission.
28th August	Hand-on-coding session 1: Plotting with Matplotlib and k-means clustering.
2nd September	Assignment 1 release: Fuzzy Logic and Genetic Algorithms.
11th September	Assignment 1 Submission.
13th September	Hand-on-coding Session 2: Introduction to Scikit-Learn for data pre-processing and visualisation.
16th September	Assignment 2 release: RBF Networks, SVM and Multilayer perceptron.
25th September	Assignment 2 Submission.
30th September	Mid-Semester survey.
13th October	Hand-on-coding session 3: Hassle-free set-up of deep learning frameworks with DL-docker, Introduction to Pytorch.
14th October	Assignment 3 release: Exploring activation functions and variations of gradient descent, Introduction to high-level machine learning libraries.
23rd October	Assignment 3 Submission.
25th October	Hand-on-coding session 4: Scikit-Learn and Scikit-fuzzy by examples.
6th November	Hand-on-coding session 5: Pytorch by examples.
8th November	Open hours: Assignment 4 topic selection help.
8th November	Assignment 4 release: Guided project.
14th November	Assignment 4:Stage 0 evaluation.
21-23nd November	Assignment 4:Stage 1 evaluation.
25th November	End-Semester survey.

3 Details

3.1 Course Surveys

Three course surveys shall be conducted at the beginning, middle and end of the semester. The pre-course survey will help us assess the prior knowledge of students so that we could adapt to appropriate pace. Mid-semester survey would be crucial towards helping students where they face difficulty and correct our methods. We could further improve the course in future iterations based upon the end-semester survey.

3.2 Hands-on-coding session 0

The first session would provide an introduction to Python, Numpy and jupyter notebooks. This session could be modelled around these tutorials from cs231n.

3.3 Assignment 0

Assignment 0 would start with some practice problems on Numpy, so that students get comfortable with matrix operations. Watching this scipycon video tutorial could be suggested as an extension of our first hands-on coding session. The problems shall cover important concepts like fancy-indexing and masked index. 100 Numpy exercises is a great resource and we can have some 10-15 small problems similar to/based on the 1 and 2-star questions. And one three star question, personally I like Convoy's game of life a lot since it looks cool as well.

The later half of the assignment would deal with fuzzy set operations, the functions written in the first assignment shall be used for more implementation of the fuzzy logic algorithms in the next assignment.

3.4 Hands-on-coding session 1

We shall introduce Matplotlib and show different examples of loading and plotting data in python. This scipycon tutorial could be used as to model this talk. In the later half, we could walk through implementation of K-Means clustering using numpy and matplotlib.

3.5 Assignment 1

The first half of assignment 1 will be concerned with implementing a fuzzy c-means clustering and defuzzification algorithms. We would provide with most part for implementation of these algorithms and ask them to fill in specific parts so that the students could focus on the core concepts.

Formulating a problem for genetic algorithm could be tricky and shall be the centre of our focus. We shall provide with most of the helper functions required for implementation of the genetic algorithm and ask students to fill in the crucial components. This is a beautiful application of genetic algorithm, we would provide the code to run the simulation, students would have to write the fitness function and fine tune the algorithum so as to maximize the distance traveled. We could have kaggle like scoreboard, inspiring students to try out more variations in the algorithm. Harvard's CS50 does that for some of their assignments, and they are ones enjoyed the most by students as per their course survey.

3.6 Hands-on-coding session 2

Data pre-processing is an crucial step to make any learning algorithum work and hence we shall cover vector representations for different kind of data and

normalization schemes. Since, students will be fairly comfortable with Numpy and Matplotlib by this time we could easily introduce them to Scikit-learn. PCA and t-SNE have become industry standards in data visualisation. Though it might not be possible to explain the mathamatics behind those in an introductory class, we can provide them as a tool, while conveying the underlying intution.

3.7 Assignment 2

RBF networks were part of the course last year and are an effective way towards understanding how neural networks work as a fitting algorithm. Though RBF nets are not as popular in present days, SVMs with RBF kernels which are a specific case of RBF nets beat multi-layer perceptron on many benchmark problems. Hence, we shall start the assignment with implementation of SVM with RBF kernel and then generalise it to RBF nets. Later, we would show how we could keep most of the code written for RBF nets intact and make slight variations to create multi-layer perceptrons.

We shall keep one common problem throughout the assignment and show the comparative analysis of the three algorithms. We could have a scoreboard similar to the one in the previous assignment.

3.8 Hands-on-coding session 3

We all have faced trouble setting up our deep learning environment with GPU but never the less getting it done is important. For a newcomer, this experience could be overwhelming, but Docker provides and effective and simple solution to the problem. DL-docker from Floydhub includes all the popular deep learning packages caffe, torch, theano, tensorflow, pytorch, keras and lasagne which students might want to explore during their final assignment. Since, many of these libraries require manual build even for those students who want to run the ecosystem on a CPU would also find it helpful.

Though tensorflow is the most popular deep learning library it is not the easiest one to use. Pytorch from Facebook AI Research has been gaining popularity at an unprecedented rate. Dynamic graphs in pytorch makes its simple to debug and extremely user-friendly. Essentially, writing code in pytorch is just like writing code in numpy. In this session, we shall provide a gentle introduction to pytorch with maybe a hello world problem like MNIST.

3.9 Assignment 3

Activation functions and decent algorithm are possibly the most important meta parameter in a neural network. More often than not people fail at making the right choice. The first half of the assignment would deal with implementing various activation functions and variations of gradient descent in numpy (building over the code in assignment 2) and analysing it's results.

In the later half, we shall have a modelling problem which students would have to code in pytorch. Again, we could have a scoreboard. This exercise would be instrumental towards the final assignment.

3.10 Hands-on-coding session 4

Scikit-Learn and Scikit-fuzzy might be crucial for implementation of the final assignment for students working on algorithms apart from neural networks. As most of our exercises in the assignments will be written in numpy, a session where we explore more examples in the above mentioned libraries could be very helpful.

3.11 Hands-on-coding session 5

The sixth and final hands-on coding session would also be on similar lines as that of the previous one except it would focus on applying pytorch on real world problems.

3.12 Assignment 4

The final assignment will be special since it will carry twice the weightage of other assignments. During the mid-semester survey, we shall also ask students about the algorithms they would want to work with in final assignment and their previous experience in machine learning. For this assignment students would have to form teams of 3, we would possibly have about 60 teams based upon registrations in the previous semester. We could select about 10 classic papers covering topics from the course, we could increase or decrease the number of papers on a certain topic based on the survey.

We shall try to select the papers such that they introduce some new concepts or ideas. For example, Geoff Hinton and Raslan Salakhutdinov's 'Reducing the dimensionality of data with neural networks'. Or for someone more experienced something like AlexNet or ResNet. The very day we allot the topics we could have an open hour, where we could change the topic for those unhappy with the assigned topic.

The first stage of evaluation would test the understanding of the paper. Since we have restricted ourselves to only 10 papers. We could have an MCQ test which would have to be taken individually. The key benefits of this would be that we get to test understanding of each individual student in a quantitative way. While the second stage would involve submission of code and Viva. Since multiple teams would be working on the same paper, we could set up an evaluation script along with scoreboard for each problem. This would eliminate the risk of plagiarism to a certain extent and incentivise students to innovate.

This way we could have a very objective evaluation of projects adding both transparency and fairness. It sure does all a bit of restriction and affects select few who have something very specific in mind. But selecting a topic from 10

classic papers should not be a that difficult. Moreover, seems a reasonable choice in favour of the benefit of the majority.