

ML tutorials - UMIC

Author: Mithil Vakde

Brief Intro

What is computation?

Every single* computation done by computers today can be thought of as a solution to this problem:

Given an input X , you want an output Y

To do so, design an operator \hat{O} that acts on X and gives the output Y

In other words,

Given X and Y , design \hat{O} such that $\hat{O}(X) = Y$

How? Simple, \hat{O} is nothing but the program you write in python/assembly/binary/etc.

Wtf? Explain more

Try finding any counterexample. You'll see that you can always reduce the computation to a version of the above problem. Spend some time thinking, seriously.

The only variations you should be able to come up with are:

- Given X and \hat{O} , find Y
- Given Y and \hat{O} , find X
- Combine/chain any of the above 3 problems
- Modify any of the above with partial information on X , Y and/or \hat{O}

Hint please

The method computers use to implement this problem is by converting it to Boolean or Fuzzy logic (or a mix of the two).

Boolean logic:	Fuzzy logic:
$X \in \{0, 1\}^n$	$X \in [0, 1]^n$
$Y \in \{0, 1\}^m$	$Y \in [0, 1]^m$
$\hat{O} : \{0, 1\}^n \rightarrow \{0, 1\}^m$	$\hat{O} : [0, 1]^n \rightarrow [0, 1]^m$

Okay what is Machine Learning?

Very simple. Given your input X and output Y :

- Normal programming is manually designing \hat{O}

- Machine learning is automatically designing \hat{O}

Formally, machine learning is the solution \hat{M} to this problem:

Given X' and Y' , find \hat{O}' by designing \hat{M} such that $\hat{M}(X, Y) = \hat{O}'$

If you think about it, this is still the same as the original definition of computation.

- $\{X', Y'\}$ together become your input (= X)
 - \hat{O}' is your output (= Y)
 - \hat{M} is your operator (= \hat{O})
-

Prerequisites to this training module:

Assumed minimum CS prerequisites: Familiarity with Python, Numpy and Pandas

Other helpful python modules to know beforehand (not a prerequisite) : Scikit Image, OS

Assumed minimum Math prerequisites: Basics of statistics, calculus and linear algebra. (You would know all of this anyway from school)

We encourage you to refer material apart from the stuff given below as well to expand your understanding. We have tried to curate a list of resources we found would utilize your time in the best way.

This assignment, in particular, has a lot more material for the allocated time in the proposed timeline. Finish off the things as fast as you can, to explore more concepts included here. Try to be ahead of the given timeline here to cover extra material given in this guide (It is highly encouraged and recommended to do so)

Details of material in this training module:

(You will not have to complete all of these. What all you have to complete will be given below)

1. Machine Learning crash course, Google

- You are required to do around 10.5 hours of content from this course.
- This must be done in 2 days at max. (Brutal, but doable with enough focus)
- An advantage of the course is that a lot of important info mentioned in the videos is rehashed (better) in the subsequent text. Useful for referencing later on.
- Just before starting the Neural networks part, watch the **3Blue1Brown** Series given below.
- Do everything in this course diligently until Embedding. Skip everything after (*and including*) Embedding
- Note that the “Regularization: Sparsity” module is optional (but recommended)
- This might feel overwhelming, but we would like to point out that understanding is infinitely more important than mugging up stuff. For example: Even if you forget exactly what **prediction bias** or **bucketing** means, it's alright as long as you know that the average of predictions should be near the average of labels in the dataset and why you can't measure this average the same way when using logistic regression.

- [Link to the course](#)

2. **Neural Networks series by 3Blue1Brown, YouTube**

- Consists of 4 episodes
- Gives an extremely good intuitive (although basic) understanding of Neural Networks
- A total of ~64 minutes of (brilliant) content
- Worth watching more than once to clear up doubts later on
- Your objective is to understand the intuition. Don't try to memorise the math. As long as you're able to understand everything Grant is saying, that is enough. However, make sure that you understand everything in these videos.
- Mantra: Pause and Ponder
- [Link to the videos](#)

3. **Udacity Course: "Intro to Tensorflow for Deep Learning"**

- Highly recommended course
- Complete lessons 3, 4 and 5. This amounts to about 6 hours worth of material.
- You do not have to do all the lessons. Ignore everything from lesson 7 onwards.
- You are expected to do this in 1 day at max.
- [Link to the course](#)

4. **Tensorflow Core tutorials:**

- Good tutorials for understanding TF modules
- To be done once a basic understanding of ML is achieved
- Look at left side column on what to learn
- There are two types of tutorials, basic and advanced. We recommend doing the basic tutorials.
- [Link to the tutorials](#)

5. **Keras Guide**

- After TF core tutorials
- Look at left side column for what to learn
- Ignore RNNs
- [Link to the Guide](#)

Timeline:

Day 1:

Machine Learning Crash Course: Complete at least 6+ hours worth of material (Essentially, complete as much required so that you can finish the course along with the YouTube playlist tomorrow)

Day 2:

Machine Learning Crash Course: Finish the rest of the course

3B1B's Neural Networks: Complete this playlist before starting the Neural Networks part of the crash course

Day 3:

Udacity's Course: Complete Lessons 3,4,5 (Lessons 1 & 2 are somewhat of a recap of what you've learnt in the past 2 days. Please go through them first if time permits)

Ahead of the timeline:

1. Complete Lesson 7 of the Udacity course (We assume you've done Lesson 1 and Lesson 2 before reaching here).
2. Now you will be given a small assignment to do. **(This assignment is not related in any way to the actual project you will be doing after the training week.)** There are two projects given below: Intermediate and Advanced. You can choose any **ONE** assignment to do from below. You will receive the problem statement and the dataset for this when you ask for it from your mentor. But note that this will **NOT** be given unless you have completed all the above material or you know it from before. This project might require slightly more knowledge than what the above material provides. For those reasons, these can be helpful:
 1. TF core tutorials (the basic tutorials)
 2. Keras Guide (ignore RNNs, Transfer Learning)

Intermediate Assignment:

A database of pictures will be given. Your task will be to create a program that can separate the images into 2 classes based on one attribute. (Dataset and complete problem statement will be provided by your mentors when asked)

Advanced Assignment:

A database of pictures will be given. Your task will be to create a program that can separate the images into multiple classes based on 3 attributes. (Dataset and complete problem statement will be provided by your mentors when asked)