Machine Learning Course Outline

Last Modified Jan 14, 2020

# Goals

The goals of this course are to understand the mathematical underpinnings of modern machine learning systems, explore the kinds of problems machine learning is best suited to solve and to implement simple machine learning models. Students may choose to use any language they wish, however JavaScript or Python are encouraged.

*The schedule is designed as a guideline for parents and students to follow along. Some students may move faster and some may move slower. In the instance that the student is not understanding the concepts, logic, or math, we will be sure to notify the parent.*

Age recommendation: 12+

Minimum math requirement: Understands functions

# Tentative Schedule

#### Session 0 (skip if the student is comfortable with the chosen language)

* Introduce data types (if applicable)
* Introduce variables
* Introduce loops
* Introduce functions

#### Session 1 (skip if the student can complete the checkpoint)

* Reinforce functions (this **must** be concrete before continuing)
* Introduce arrays and collections
* Checkpoint: implement a function which returns the sum of all elements in an array
* Introduce mathematical notation for functions and loops (f of x, ellipses, sigma)

#### Session 2

* Give a demonstration of a machine learning application in action
* Review mathematical notation for functions and loops
* Introduce the artificial neuron, show how it compares to a human neuron
* Review array operations

#### Session 3

* Introduce the perceptron (a simple network)
* Introduce vectors (as arrays)
* Introduce the dot product operation, show how it can be used to compute the perceptron
* Checkpoint: Implement a function which takes two vectors and returns their dot product

Session 4

* Review the dot product function, reinforce how it can be used to compute the perceptron
* Implement a two input perceptron which approximates either the AND or OR functions
* Introduce linear separability, and demonstrate how this model **can’t** approximate XOR
* Introduce the idea of an optimizer, and the necessity of an error function

#### Session 5

* Introduce MSE (mean squared error) between two vectors
* Checkpoint: Implement a function which takes two vectors and returns their MSE
* Introduce the random optimizer
* Implement the random optimizer

#### Session 6

* Review the MSE function
* Introduce the idea of a genetic algorithm
* Introduce the selection, mutation, and recombination operators
* Checkpoint: Implement two of the three genetic operators

#### Session 7

* Review the genetic operators
* Implement an optimizer using a genetic algorithm
* Checkpoint: Train the network to a reasonable degree of accuracy
* Review linear separability and introduce activation functions

#### Session 8

* Review activation functions
* Checkpoint: Implement RelU and tanh
* Introduce hidden layers and multilayer perceptrons
* Implement a hidden layer with RelU activation

#### Session 9

* Introduce the idea of a matrix
* Show how a matrix can be used to simplify the weight calculation
* Implement weights as matrices
* Introduce the map operation

#### Session 10

* Review matrices
* Review the map operation
* Review activation function
* Checkpoint: use the map operation to apply the activation to each layer in the network

# Congratulations, you made it!

You’ve built a machine learning model that can approximate boolean functions from scratch, and that’s no small feat! You should now have a solid understanding of simple machine learning models. From here you can branch out into a few different fields:

* Image classification (computer vision, character recognition, etc…)
  + Learn about deep convolutional neural networks
  + Learn about convolutional kernels, pooling, and other techniques
  + Learn the TensorFlow library
  + Example: <https://github.com/shmishtopher/pneumonia-CNN>
    - Detect pneumonia in x-ray images
  + Example: [https://goldenrodivoryplanes.christopherschmitt.repl.co](https://goldenrodivoryplanes.christopherschmitt.repl.co/)
    - Classify handwritten digits
* Machine learning in games (AI flappy bird, AI snake, etc…)
  + Learn to build networks that can play games
  + Learn about dropout layers and other overfitting solutions
  + Learn simulation techniques
  + Example: <https://trinket.io/python/1b160f7edb>
    - Watch a perceptron learn to play perfect flappy bird
* Regression problems (profit maximizer, emotion analyzer, etc...)
  + Learn about graph models
  + Learn more advanced dataset labeling techniques
  + Learn about data sourcing and web scraping