Deep Learning Lecture 1: Introduction

Abdelrahman Khaled

Machine Learning Research Cluster
German University in Cairo

March 27, 2019

Outline

On Research and the Cluster

Who are we?

We are a small group of volunteers that are enthusiastic about machine learning, its applications and its future. We are trying to create a community within the GUC (supervised by Dr.Aisha Al-Safty) that can work hard with us and eventually carry the community for many years to come.

What do we do exactly?

We are trying to mentor everyone who wants to study ML. Whether it's for a personal project of theirs, or if they want to join the community.

On Research and the Cluster

What can you do if you join?

We hope that the members who want to join can become any of the following four types of members:

Mentor/Instructor: This member should be ready to prepare material to present and for the students to learn from.

Researcher: This member should be able to create interesting projects and write papers for them.

Competitive: This member should try to enter ML competitions and try to achieve better results each time.

Management: This member should be responsible for non-academic related tasks. (Reserving labs, communications, social, etc.)

Neural Networks in Action

Neural Networks have been used for many things including, but not limited to:

- Character Recognition
  Through convolutional neural networks (sometimes through classical neural networks) the model can learn to distinguish between different characters of a language and, in turn, understand how to "read" or "write".
- Stock Market Prediction
  From data available from the stock market's enormous backlog of past stocks, try to predict the trend of future stock prices.
- Playing Games With Large State Spaces
  It seems like a very trivial task, but learning to traverse an environment is one of the most difficult problems in AI, and neural networks are our best method so far.

Course Outline

This course will consist of 3 main parts:

1. Neural Networks From Scratch
2. Deep Networks Using Keras and/or Tensorflow
3. Convolutional Neural Networks

There is a lot more ground to cover when it comes to neural nets, but for the sake of time, and our lack of knowledge we will only discuss these.

If you want to learn more, consider reading about the following when the course is over:

- Recurrent Neural Network (RNN)
- Long/Short Term Memory (LSTM)

If time allows it, we will try to cover a little bit of RNN.

How To Benefit From This Course

This course leans more towards the theoretical part of ML, however we will still be doing many practical demonstrations and will discuss and use many ML tools.

You can use any programming language you want at first, but eventually we will all be using Python. I will be using processing only for the first few sessions.

How To Benefit From This Course

This course leans more towards the theoretical part of ML, however we will still be doing many practical demonstrations and will discuss and use many ML tools.

You can use any programming language you want at first, but eventually we will all be using Python. I will be using processing only for the first few sessions.

- Read the slides as soon as you can, and check the references for more sources you can learn from.
- Try to implement everything you learn within a week of learning it.
- Follow the practical demonstrations that your instructor will be doing.

How To Benefit From This Course

This course leans more towards the theoretical part of ML, however we will still be doing many practical demonstrations and will discuss and use many ML tools.

You can use any programming language you want at first, but eventually we will all be using Python. I will be using processing only for the first few sessions.

- Read the slides as soon as you can, and check the references for more sources you can learn from.
- Try to implement everything you learn within a week of learning it.
- Follow the practical demonstrations that your instructor will be doing.
- Ask! During the lecture and on the Facebook group too.
- If you can, answer other people's questions.

Recall: Linear and Logistic Regression

- Logistic regression
  - A binary classification algorithm.
  - Outputs a real number (probability) signifying what class the input belongs to.
  - The probability is discretized using a threshold function.
  - Multiple logistic regressors can be used to classify more than two classes.
- Linear Regression
  - A regression algorithm.
  - Fits a line onto known data.
  - It's predictions for unknown data relies on the best fit line.

Linearly Separable Classification Problem

- A problem where the two classes of data can be separated by a line in 2-D, a plane in 3-D or a hyper-plane in n-D.
- Examples of linearly separable classification problems are the Boolean AND & OR functions.

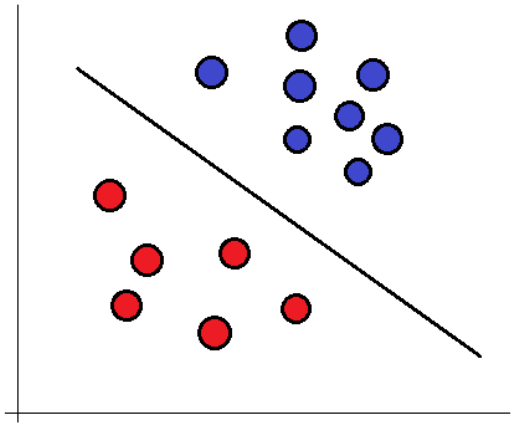

Figure: Linearly separable data. Each axis represents a feature.

The Perceptron

In machine learning, the perceptron is a mathematical model made to approximate the function of a single neuron in the human brain.

- A neuron is either active (firing) or inactive.
- Is this enough to solve any linearly separable problem?

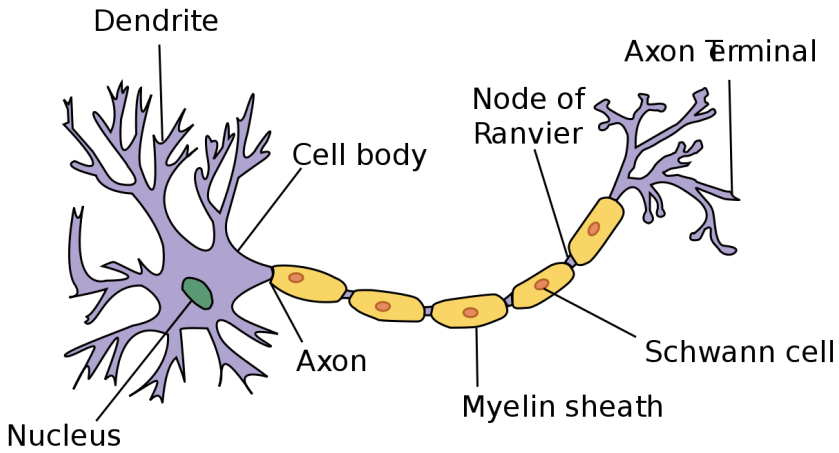

Figure: A human neuron (nerve cell).

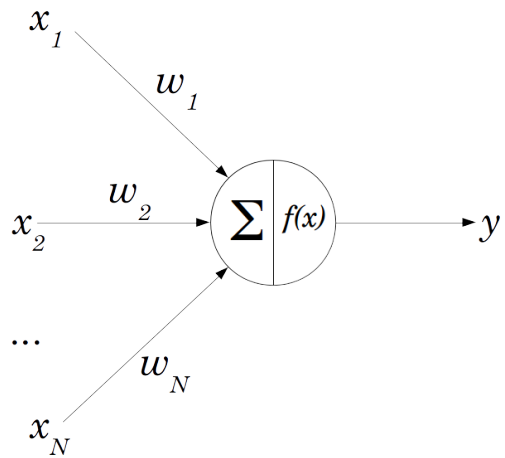

Figure: The mathematical neuron. *Image Source*

The Perceptron

The idea behind the perceptron is that some features are more important than others, therefore they need to carry more weight!

We will define a special number h that is the weighted sum of all the features of an instance of the dataset.

$$h = \sum_{i=1}^{N} w_i x_i$$

And we will define an activation value y which will be the result of discretizing the sum h according to a predefined threshold.

$$y = f(h) = \begin{cases} 1, & h >= \theta \\ -1, & h < \theta \end{cases}$$

Solving the OR Problem

The OR table can be transformed into a simple graph on the Cartesian plane.

x1	x2	output
0	0	0
0	1	1
1	0	1
1	1	1

Table: OR truth table

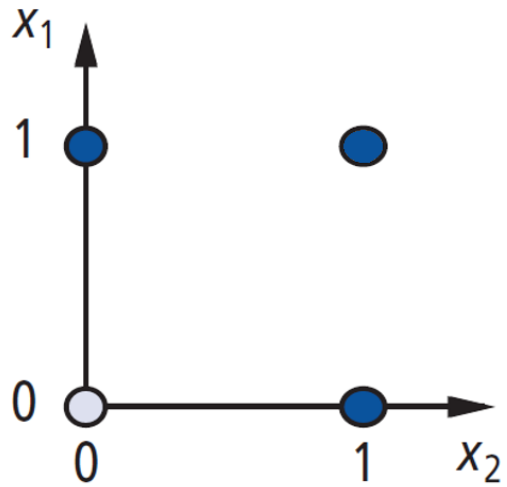

Figure: OR on the plane

Training the Perceptron

Training a perceptron means to tweak the weights such that for every iteration the weights get a little closer to what they should be.
We can define an error E such that the larger E is the farther a perceptron's prediction y is from its actual target value t.

$$E = t - y$$

We can use this error to shift the weights in the direction of the target.

$$\Delta w_i = (t - y)x_i$$

Note

We can also use $\Delta w_i = tx_i$ because it is proportional

Training the Perceptron

For a simple problem with two features, we can think of the weights as representing the separating line in 2D space. But to accurately represent a line, we need 3 numbers. That third number is called the bias.

Bias Weight

An extra weight connected to an input that is always one in order to give the line a point of reference other than the origin

---

Note

Obtaining the sum h is the same thing as computing the dot product between the two vectors w and x.

The Preceptron Algorithms

Input: num vector x
Output: num
Static : num θ, num vector w

if w *not initialized* **then**
  w = zeroes;
end
$h = w \cdot x$;
if $h < \theta$ **then**
  return -1;
else
  return 1;
end

Algorithm 1: Prediction

The Preceptron Algorithms

Input: num vector x, num t
Static: num vector w, num l

$y = \text{predict}(x)$;
$E = t - y$;
for $i = 0$; $i < w.length$; $i++$ **do**
  $w[i] = w[i] + l * (E * x[i])$
end

Algorithm 2: Training

Learning Rate

The variable l is called the learning rate and it dictates how fast (or slow) the perceptrons weights change.

Next Time

1. Perceptron Convergence.
2. Batch Training.
3. Multi-Layer Perceptron.

References

- Applications of Neural Networks. link
- Goodfellow, Bengio, Courville. *Deep Learning*. Massachusetts: MIT Press, 2017. **Chapter 6**
- Chapman, Hall. *Machine Learning: An Algorithmic Perspective*. CRC Press, 2014. **Chapter 4**
- Non-academic resources (YouTube)
  - 3 blue 1 brown. **Neural Networks**. link
  - The Coding Train. **10: Neural Networks - The Nature of Code**. link