IntroML: Lecture 1 - Introduction to Machine Learning

Kyle Swanson

January 8, 2018

1 Introduction

Most of this lecture is a PowerPoint presentation available here: `https://github.com/swansonk14/IntroML/tree/master/Lectures`

2 Perceptron Limitations

Perceptrons generally perform very well on data sets which they are able to classify. However, there are many data sets which perceptrons are unable to classify, including some very simple data sets.

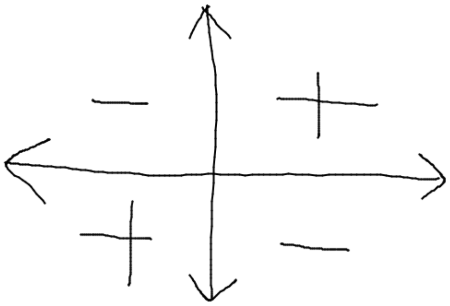
For example, perceptrons are unable to classify data points belonging to the XOR function $\oplus$. As a reminder, the XOR function is a boolean function which is 1 if exactly one of the inputs is 1.

$$0 \oplus 0 = 0$$
$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$
$$1 \oplus 1 = 1$$

We can extend the idea of the XOR function to 2-dimensional data points by performing an XOR operation on the sign of the two coordinates, which looks like this:

$$+ \oplus + = +$$
$$- \oplus + = -$$
$$+ \oplus - = -$$
$$- \oplus - = +$$

and produces a plot like this:



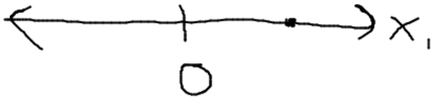
As simple as this function might be, perceptrons are unable to classify it, as we'll see tomorrow. However, we'll later learn a way to modify perceptrons so that they are able to classify more functions, including the XOR function.

3 Feature Representation

In order to apply a machine learning algorithm to data, we first need to extract features from the data. Features are represented as d-dimensional vectors, such as the following:

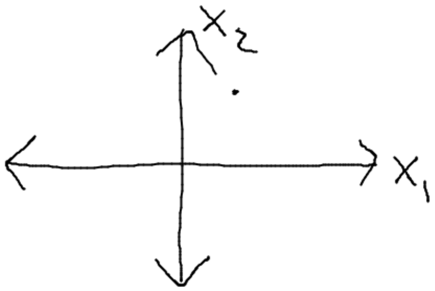
$$\mathbb{R}^1$$

$$x = \begin{bmatrix} 2 \end{bmatrix}$$



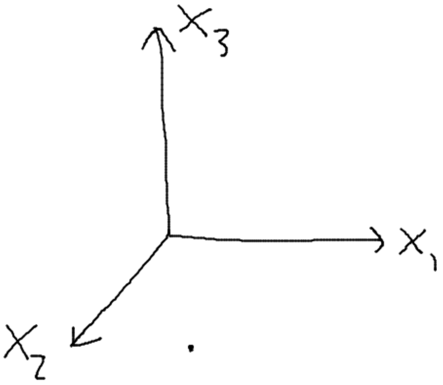
$$\mathbb{R}^2$$

$$x = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$



$$\mathbb{R}^3$$

$$x = \begin{bmatrix} 2 \\ 3 \\ -5 \end{bmatrix}$$



$$\vdots$$

$$\mathbb{R}^d$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

3.1 Example

3.1.1 Data and Labels

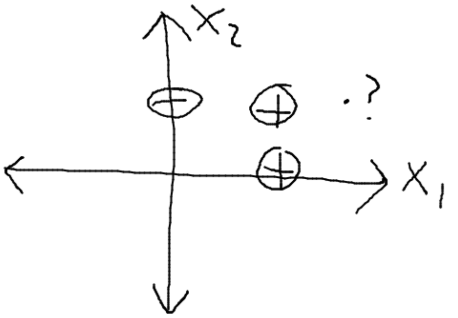
Data	Labels
$r^{(1)}$ = "The movie was good."	$y^{(1)} = +1$
$r^{(2)}$ = "Bad. Don't see."	$y^{(2)} = +1$
$r^{(3)}$ = "Good mostly. Some bad parts."	$y^{(3)} = +1$

3.1.2 Features

We will create feature vectors based on how many times the words "good" and "bad" appear in the text.

$x_1 = \#$ of times the word "good" appears
$x_2 = \#$ of times the word "bad" appears

$$x^{(1)} = \begin{bmatrix}1\\0\end{bmatrix}$$
$$x^{(2)} = \begin{bmatrix}0\\1\end{bmatrix}$$
$$x^{(3)} = \begin{bmatrix}1\\1\end{bmatrix}$$



Now that we have created feature vectors, our goal is to build a classifier which can predict whether new points, such as the ? point, are + or −.

4 Classifiers

A *classifier* is a function which maps d-dimensional vectors to labels:

$$h : \mathbb{R}^d \mapsto \{-1, +1\}$$

Essentially, a classifier predicts the label of a feature vector.
Below are some classifiers and the prediction for the unknown point ?.