Perceptrons and Environment Setup

CMSC 389A: Lecture 2

Sujith Vishwajith

University of Maryland

Agenda

1. Logistic Regression Recap
2. Perceptrons
3. Processing Features
4. Example
5. Environment Setup
6. Announcements

Logistic Regression Recap

Logistic Regression Recap

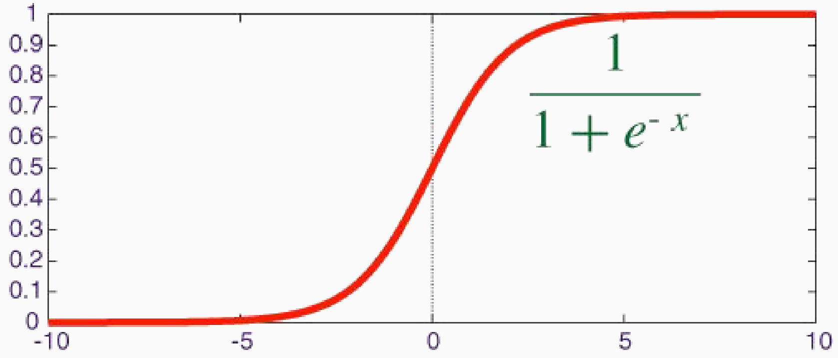
Tries to find a linear separator between classes.

Models the probability that an example belongs to a certain class.

$$P\,(Y = 1 \mid data)$$

Want to predict a probability that either ***Y=1*** or ***Y=0*** for binary classification.

Utilizes the logistic (sigmoid) function.



Recap (cont.)

Features are represented as $[x_0, x_1, \dots, x_{|F|}]$ where **$|X|$ = *# of Features***.

Weights represented as a list $[w_0, w_1, \dots, w_{|B|}]$ where **$|B| = |X|$.**

Additional parameter ***b*** is the bias.

Our weights and bias are estimated from our training data.

Therefore probability of belonging to default class is:

$$P\,(Y=1 \mid x) = \sigma\,(w^T x + b).$$

Recap (cont.)

Update using SGD over every training example .

True label = $\bar{y}$, Features = x, Bias = b, Weights = w, *Learning Rate* = α

Prediction = $\hat{y} = \sigma(w^T x + b)$

Error = $e = \bar{y} - \hat{y}$

Updates bias:

$$b \leftarrow b + \alpha * e * \hat{y} * (1 - \hat{y})$$

Update every weight:

$$w_i \leftarrow w_i + \alpha * e * \hat{y} * (1 - \hat{y}) * x_i$$

Perceptrons

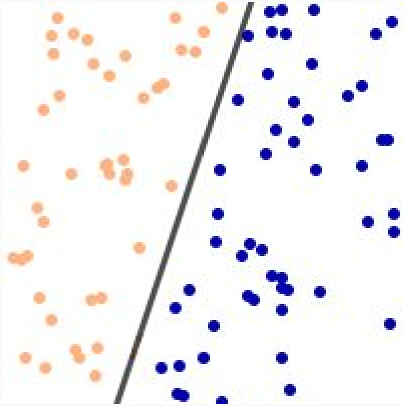
Perceptron Overview

Invented in 1957 by Frank Rosenblatt.

Similar to logistic regression as it is a linear classifier.

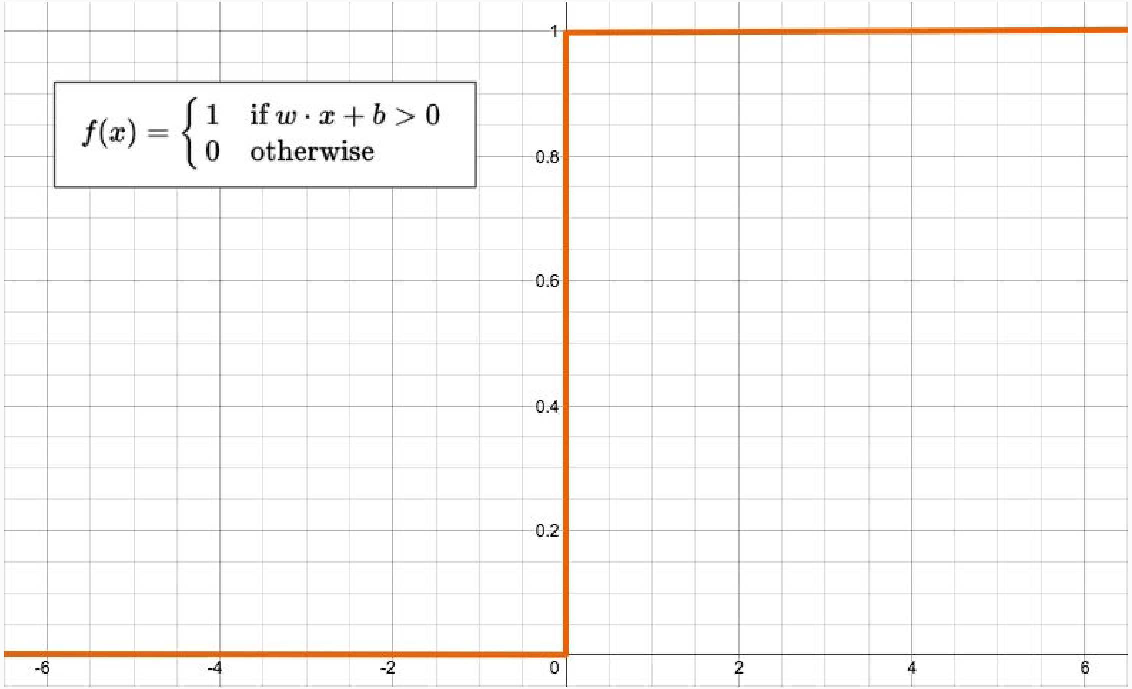
Simulates a neuron as a perceptron is either on (1) or off (0) based on a signal (features).

Uses an activation function to decide if on or off.



Activation Function

Known as the Heaviside step function.



Basics

Features are represented as $[x_0, x_1, \ldots, x_{|F|}]$ where $|X|$ = ***# of Features***.

Weights represented as a list $[w_0, w_1, \ldots, w_{|B|}]$ where $|B| = |X|$.

Additional parameter ***b*** is the bias.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Our data is classified as **Y=1** if:

$$f(w^T x + b) = 1$$

And **Y=0**:

$$f(w^T x + b) = 0$$

Training a Perceptron

Update over every training example .

True label = $\bar{y}$, Features = x, Bias = b, Weights = w, *Learning Rate* = α

Prediction = $\hat{y} = f\,(w^T x + b)$ $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

Error = $e = \bar{y} - \hat{y}$

Updates bias:

$$b \leftarrow b + \alpha * e$$

Update every weight:

$$w_i \leftarrow w_i + \alpha * e * x_i$$

Notice how no update happens if classified correctly.

Logistic Regression vs Perceptron

Perceptrons

Linear classifier.

Either 1 or 0 (no probability indication).

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Updates:

$b \leftarrow b + \alpha * e$

$w_i \leftarrow w_i + \alpha * e * x_i$

Logistic Regression

Linear classifier.

Indication of confidence (probability).

$$f(x) = \frac{1}{1 + e^{-x}}$$

Updates:

$b \leftarrow b + \alpha * e * \hat{y} * (1 - \hat{y})$

$w_i \leftarrow w_i + \alpha * e * x_i * \hat{y} * (1 - \hat{y})$

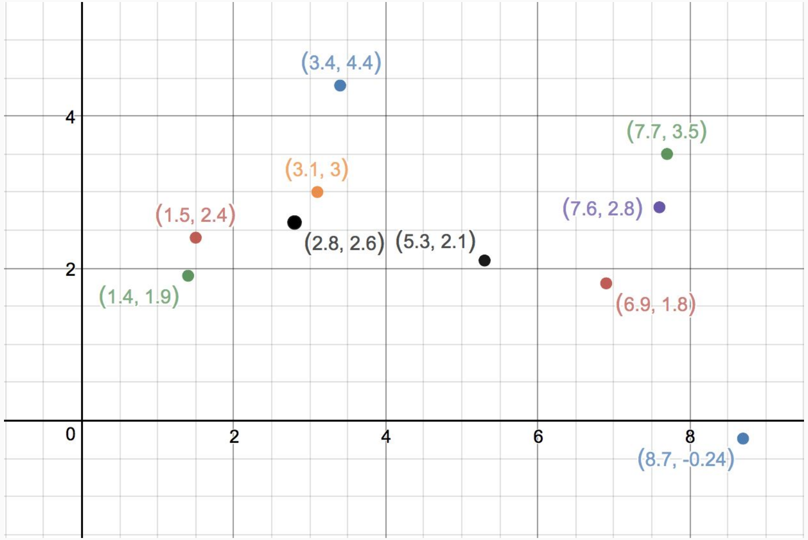
Observe how both models are incredibly similar and only differ in activation functions and weight updates.

Example

Example

Let's train a Logistic Regression model and a Perception over some sample data

X_1	X_2	Y
1.5	2.4	0
5.3	2.1	1
3.1	3.0	0
7.6	2.8	1



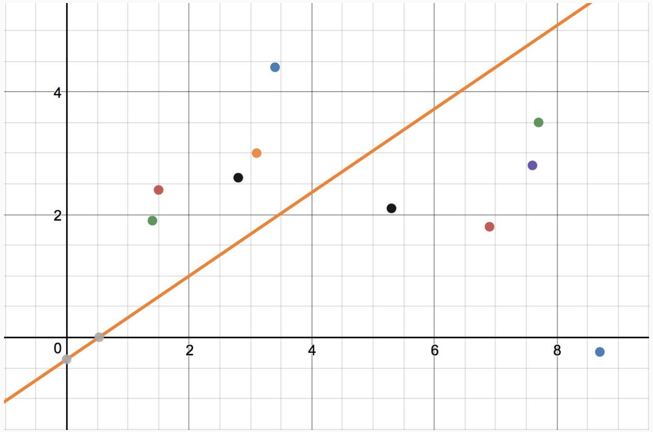
* This is a made up example found here

Results

Trained with α = 0.1 and for 10 epochs we get:

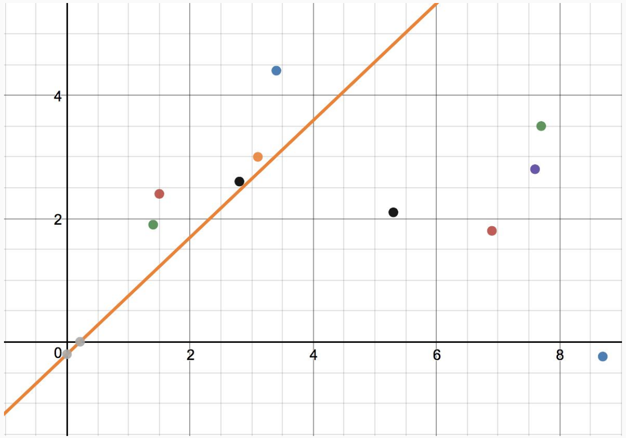
Logistic Regression:

Bias: -0.622 , $\mathbf{W_1}$: 1.172, $\mathbf{W_2}$: -1.724



Perceptron:

Bias: 0.1 , $\mathbf{W_1}$: 0.469, $\mathbf{W_2}$: -0.494



Processing Features

Continuous Features

Numeric values in $\mathbb{R}$ that range from -Inf to Inf

Examples:

- Width of flower petal in inches
- Number of times a word appears in a document

Important to rescale values between 0 (or -1) and 1. Why?

You can do this by computing: **x = (x - min x) / (max x - min x)**

Discrete Features

Finite range of integer numbers (0,1,2) or string values (male, female)

Examples:

- Boolean values: true or false
- Gender: male or female

Encode features into numbers (male -> 0, female -> 1)

Environment Setup

Let's go through the setup process for our environment to run our projects.

https://umd-cs-stics.gitbooks.io/cmsc389a-practical-deep-learning/content/course-information/environment-setup.html

Announcements

Announcements

Join Piazza for class questions and discussions.

Please complete weekly feedback.

Practical 1 is due **February 16th** at **11:59 p.m**.

Questions?