Machine Learning

Session 6

24th March 2018

Rama Krishna Bhupathi
ramakris@gmail.com

Agenda

- Activation Functions
- Gradient Descent
- Logistic Regression

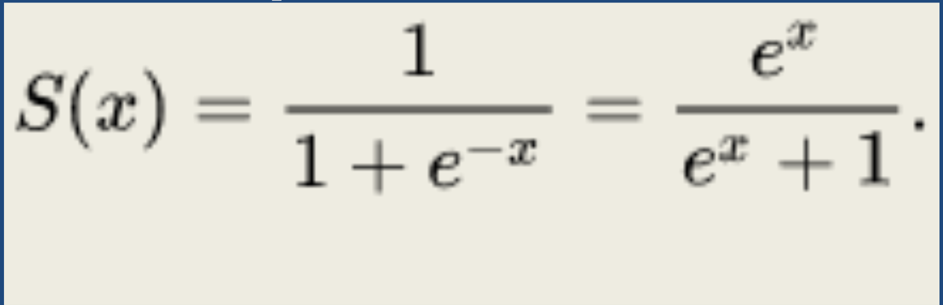
Activations Functions

- Sigmoid
- Softmax

Sigmoid… what is it ?

A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve. Often, sigmoid function refers to the special case of the logistic function shown in the next slide and defined by the formula.

Sigmoid function is represented by:



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

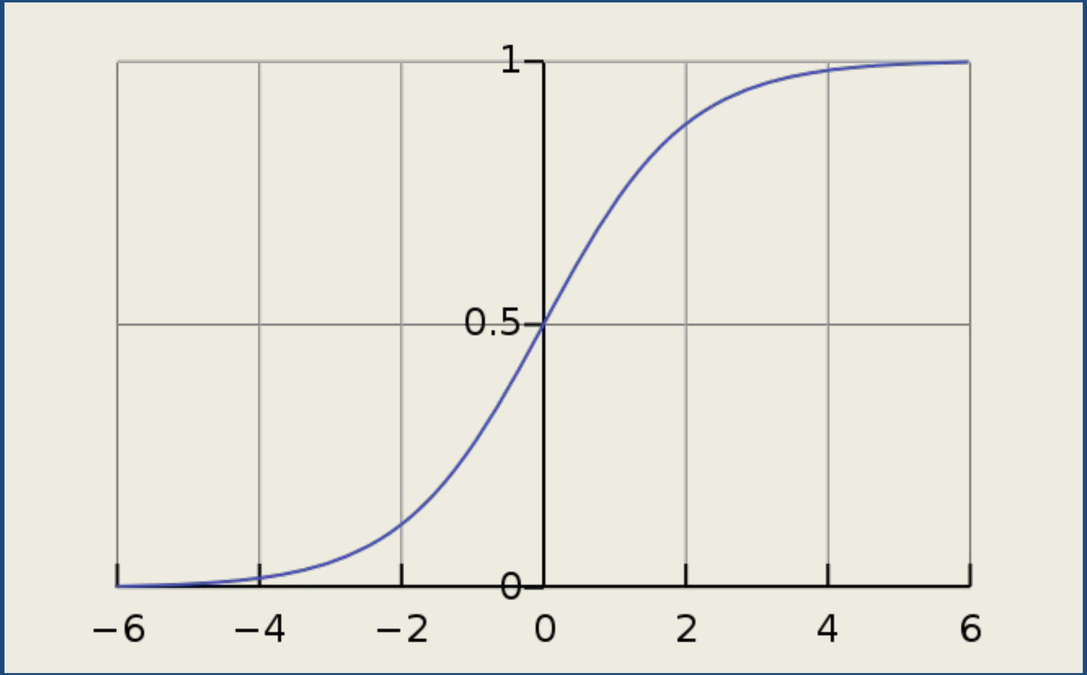
The inverse function of Sigmoid is called logit function.

Sigmoid…

Sigmoid translates values in the range (-∞ to +∞) to 0 to 1

It is used in

Binary Classification



What is the sigmoid When x=0?

Do you remember a Similar graph we learnt before?

Calculating Sigmoid

```
import python
def sigmoid(x):
  return 1/(1 + math.exp(-x))

sigmoid(0)
0.5
```

Derivative of Sigmoid

$$S'(z)=S(z) * (1-S(z))$$

Softmax

Softmax Regression (synonyms: Multinomial Logistic, Maximum Entropy Classifier, or just Multi-class Logistic Regression) is a generalization of logistic regression that we can use for multi-class classification (under the assumption that the classes are mutually exclusive).

Gradient Descent

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.

If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Let us apply gradient descent algorithm to find a local minimum of the function f(x)=x^4−3x^3+2, with derivative f'(x)=4x^3−9x^2.

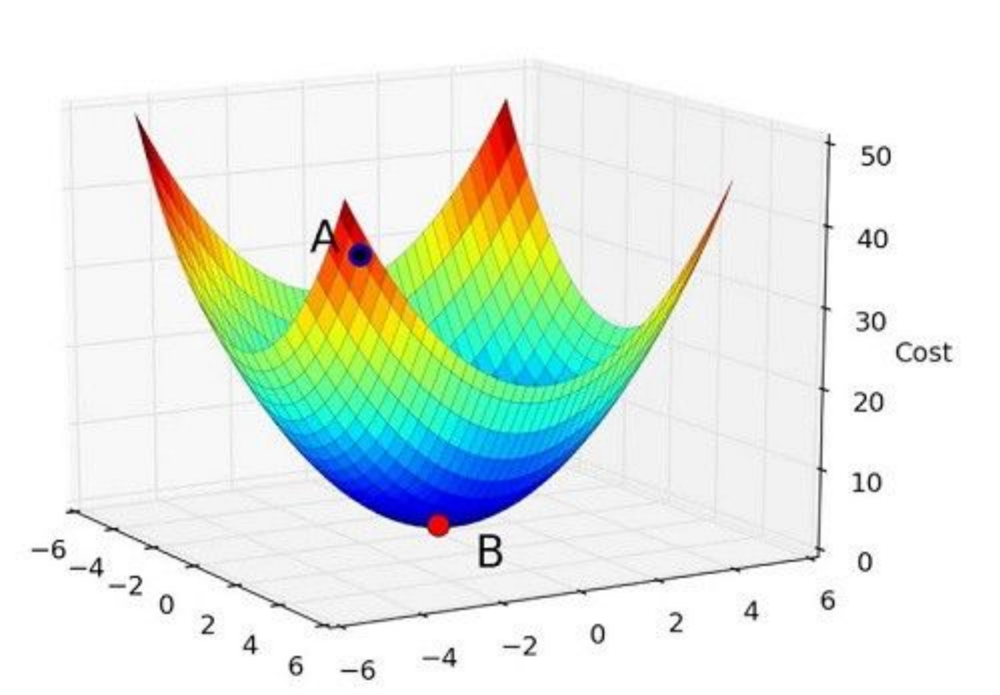
Rudimentary Implementation of Gradient Descent.

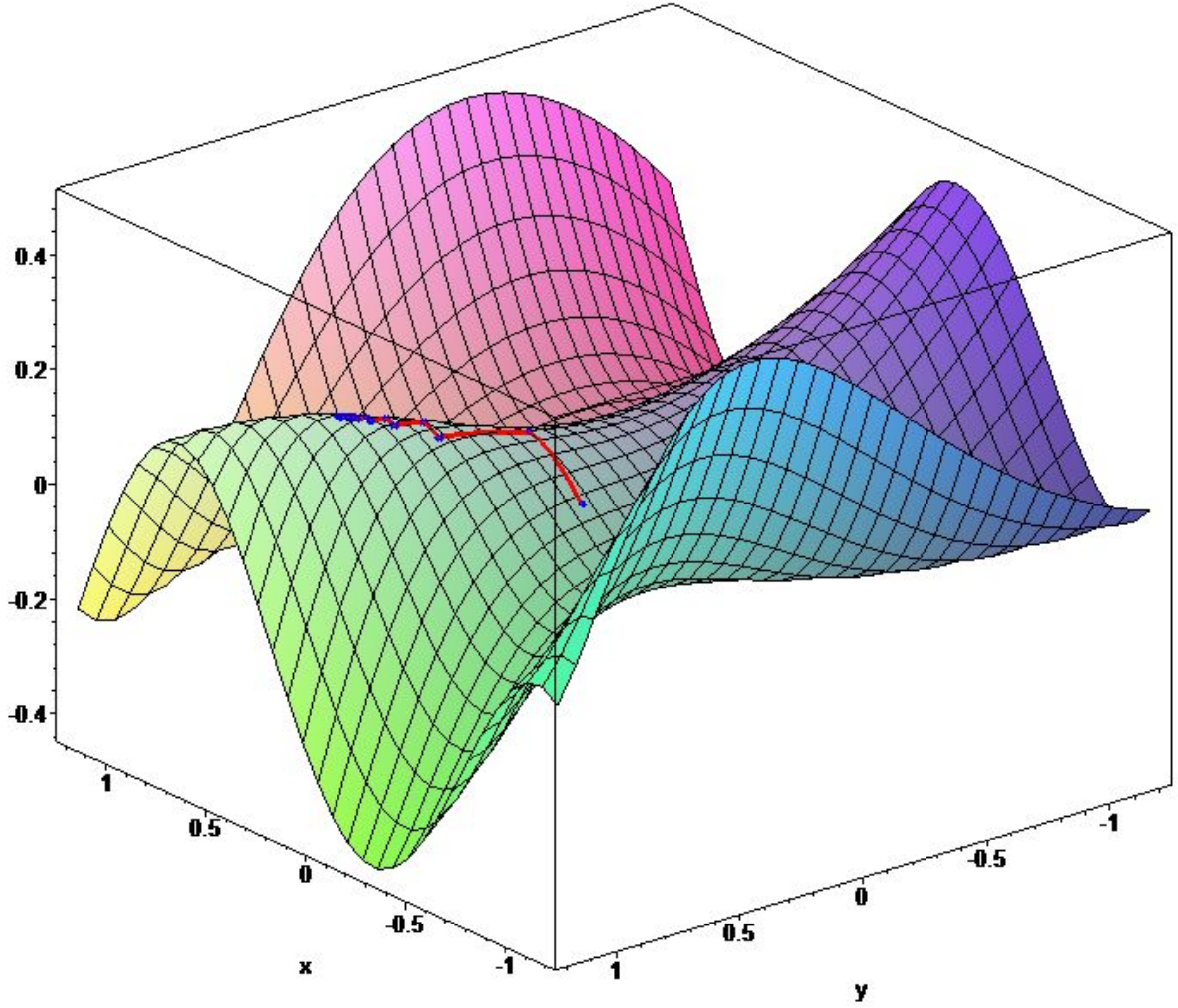
```
cur_x = 1 # The algorithm starts at x=6
gamma = 0.01 # step size multiplier
precision = 0.00001
previous_step_size = 1/precision; # some large value

df = lambda x: 4 * x**3 - 9 * x**2

while previous_step_size > precision:
        prev_x = cur_x
        cur_x += -gamma * df(prev_x)
        previous_step_size = abs(cur_x - prev_x)

print("The local minimum occurs at %f" % cur_x)
```


A
B
Cost
0
10
20
30
40
50
−6
−4
−2
0
2
4
6
−6
−4
−2
0
2
4
6


0.4
0.2
0
-0.2
-0.4
1
0.5
0
-0.5
-1
x
1
0.5
0
-0.5
-1
y

Logistic Regression.

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Good Tutorial

http://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

Logistic Regression Vs. Linear Regression

Given data on time spent studying and exam scores, Linear Regression and logistic regression on the data can predict different things:

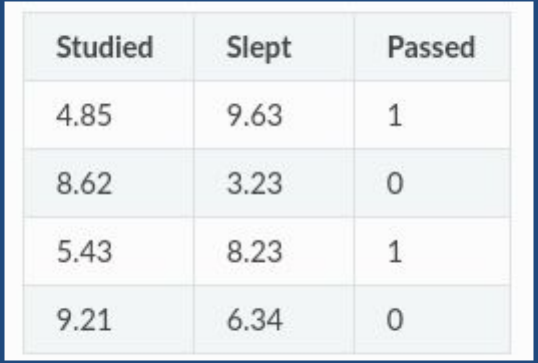
Linear Regression could help us predict the student's test score on a scale of 0 - 100. Linear regression predictions are continuous (numbers in a range).

Logistic Regression could help use predict whether the student passed or failed. Logistic regression predictions are discrete (only specific values or categories are allowed). We can also view probability scores underlying the model's classifications. Types of logistic regression:

Binary (Pass/Fail)
Multi (Cats, Dogs, Sheep)
Ordinal (Low, Medium, High)

Binary logistic regression

Say we’re given data on student exam results and our goal is to predict whether a student will pass or fail based on number of hours slept and hours spent studying. We have two features (hours slept, hours studied) and two classes: passed (1) and failed (0).



Studied	Slept	Passed
4.85	9.63	1
8.62	3.23	0
5.43	8.23	1
9.21	6.34	0

Use the data_classification.csv in the repo to build a logistic model as an exercise.

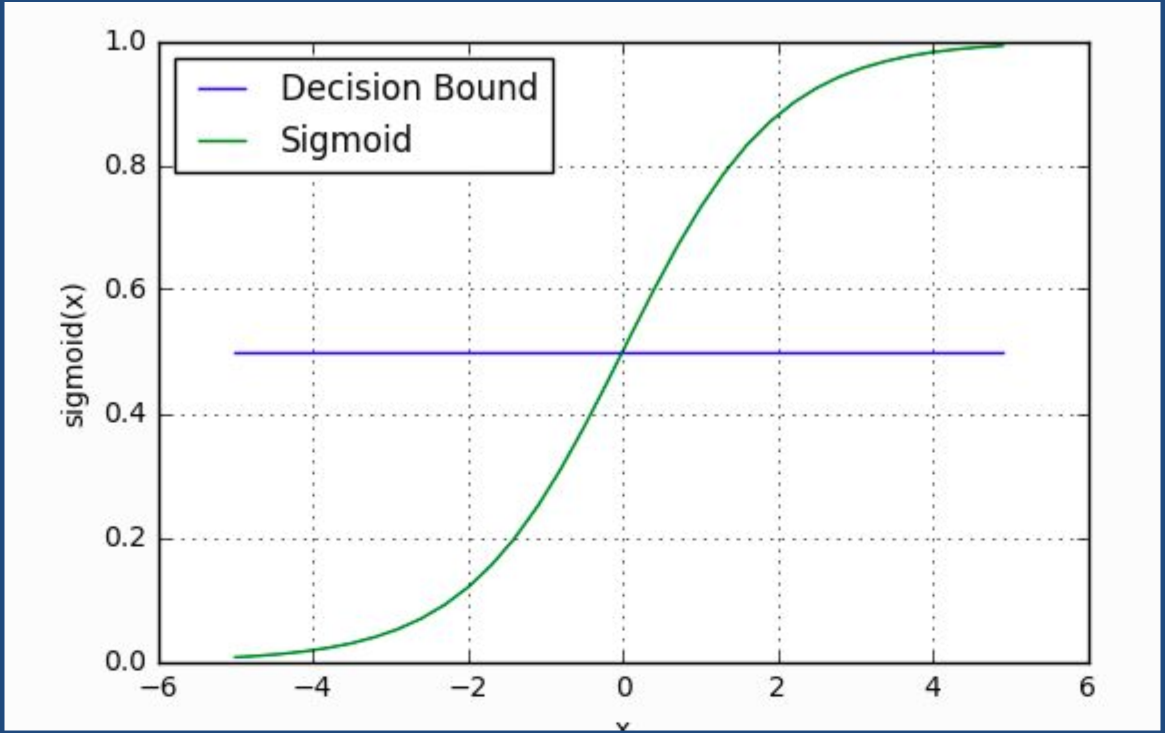
Decision Boundary

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

$P \geq 0.5, class=1$
$P < 0.5, class=0$

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.

Decision Boundary



2018© Bhupathi Consulting Inc

Prediction Function

A prediction function in logistic regression returns the probability of our observation being positive, True, or “Yes”. We call this class 1 and its notation is P(class=1). As the probability gets closer to 1, our model is more confident that the observation is in class 1.

So what is our equation ……Z=W0+W1*Studied+W2*Slept

Generic Version . $z = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$ $z = w^T x.$

OBJOBJ

We will transform the output using the sigmoid function to return a probability value between 0 and 1.

$$P(class = 1) = \frac{1}{1 + e^{-z}}$$

Cost Function

What is the cost function ?

Can we use the cost function RMSE?

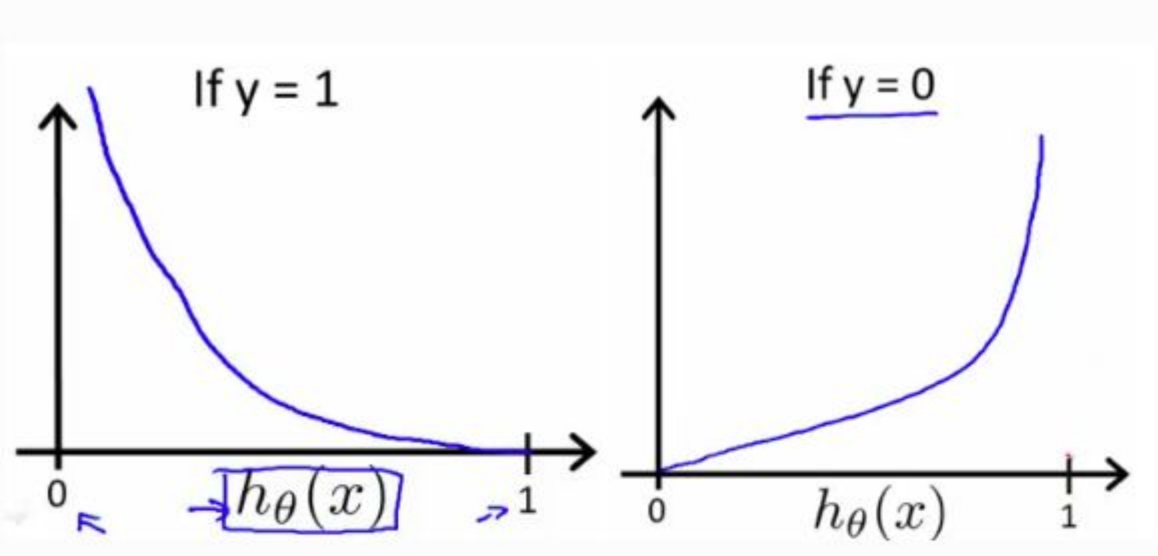
Why not?

Because our prediction is non-linear due to application of sigmoid

Loss function ...Cross Entropy

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))]$$

Y-axis
Is loss


If y = 1
$h_\theta(x)$
If y = 0
$h_\theta(x)$