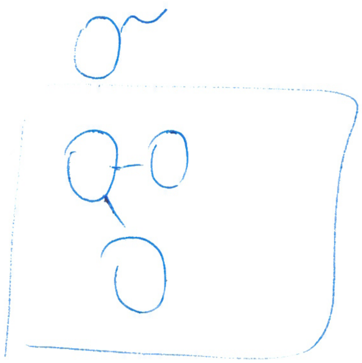
CS4811
January 13, 2016
Wednesday

brain

linear regression

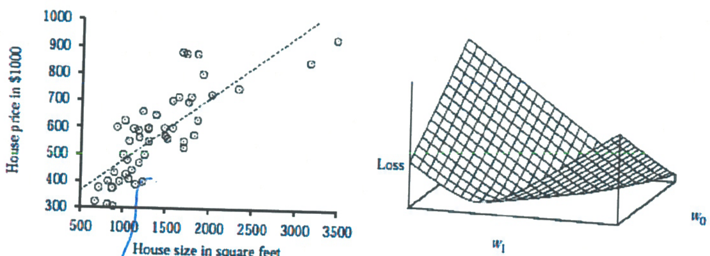
neuron

Math



Univariate linear regression problem

- A *univariate linear function* is a straight line with input x and output y.
- The problem is to "learn" a univariate linear function given a set of data points.
- Given that the formula of the line is $y = w_1 x + w_0$, what needs to be learned are the weights w_0, w_1.
- Each possible line is called a *hypothesis*: $h_{\vec{w}} = w_1 x + w_0$



(a) (b)

training examples
training instances

h hypothesis : your representation of what is learned
the formula

Learning is the process of searching for the "right" hypothesis given what we have.

$h_{\vec{w}}(x) = w_1 X + w_0$ formula of a line

Pick the hypothesis that minimizes "loss"

Loss = sum of squared errors.

$$\text{Loss}(h_{\vec{w}}(x)) = \underbrace{\sum_{j=1}^{N}}_{\text{for each data point}} L_2(y_j, h_{\vec{w}}(x_j))$$

N data points (training instances)

$$= \sum_{j=1}^{N} \left(y_j - h_{\vec{w}}(x_j)\right)^2$$

$$= \sum_{j=1}^{N} \left(y_j - (w_1 x_j + w_0)\right)^2 = \text{sum}$$

Find $\vec{w}^* = \operatorname{argmin}_{\vec{w}} \text{Loss}(h_{\vec{w}})$

computes the $\vec{w}$ vector that gives the minimum value for $\text{Loss}(h_{\vec{w}})$

$\min \text{Loss}(h_{\vec{w}})$

This sum is minimized when its partial derivatives with respect to w_0 and w_1 are zero.

$$\frac{d}{dw_0} \text{sum} = 0 \qquad \frac{d}{dw_1} \text{sum} = 0$$

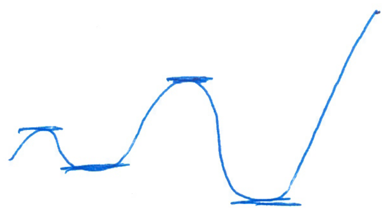
Has a unique solution

$$w_1 = \frac{N\left(\sum x_j y_j\right) - \left(\sum x_j\right)\left(\sum y_j\right)}{N\left(\sum x_j^2\right) - \left(\sum x_j\right)^2}$$

$$w_0 = \frac{\left(\sum y_i - w_1 \left(\sum x_j\right)\right)}{N}$$

$w_1 = 0.232$ ④

$w_0 = 246$



univariate . linear model

going to nonlinear models

no closed form solution

general optimization problem in a continuous weight space

use "gradient descent"

use a "hill climbing algorithm"

Algorithm:

$\vec{w} \leftarrow$ any points in the parameter space

loop until "convergence" do

for each w_i in $\vec{w}$ do

$$w_i \leftarrow w_i - \alpha \frac{d}{dw_i} \mathrm{Loss}(\vec{w})$$

⋮ skip math

weight update rules:

$$w_0 \leftarrow w_0 - \alpha \left(y - h_{\vec{w}}(x) \right)$$

$$w_1 \leftarrow w_1 - \alpha \left(y - h_{\vec{w}}(x) \right) * x$$

$$\begin{matrix} (x, y) \\ (x, y) \\ (x, y) \end{matrix} \xrightarrow[\text{input}]{} \boxed{\begin{matrix}\text{linear} \\ \text{regression} \\ \text{learner}\end{matrix}} \xrightarrow[\text{output}]{} \text{a weight vector that represents a line.}$$

N