LECTURE 5-2

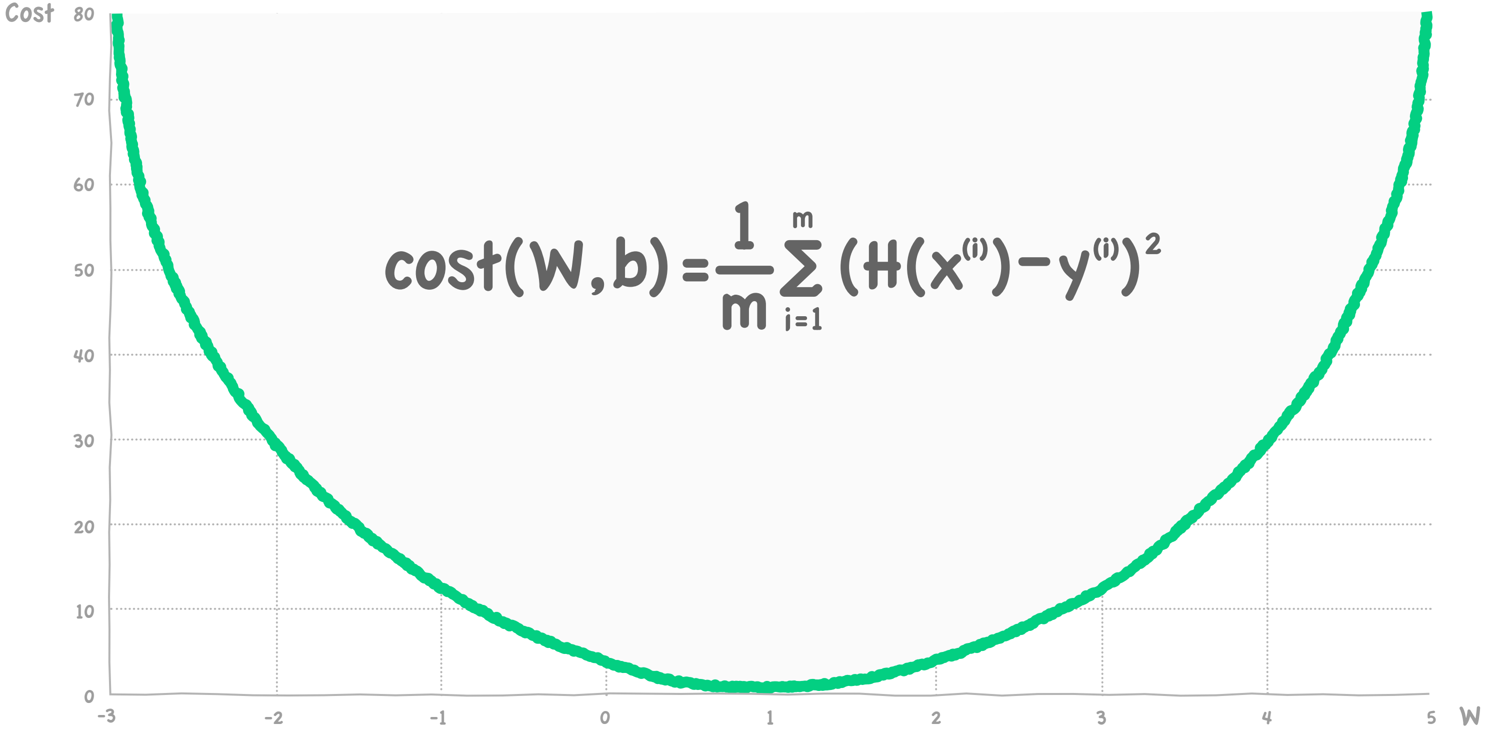
LOGISTIC (REGRESSION) CLASSIFICATION : COST FUNCTION & GRADIENT DESCENT

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

NAVER | Clova

Cost

When, $H(x) = Wx + b$



$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}\left(H(x^{(i)}) - y^{(i)}\right)^2$$

Cost Function

$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

$$H(x) = W+b$$

$$H(x) = \frac{1}{1+e^{-W^{T}X}}$$

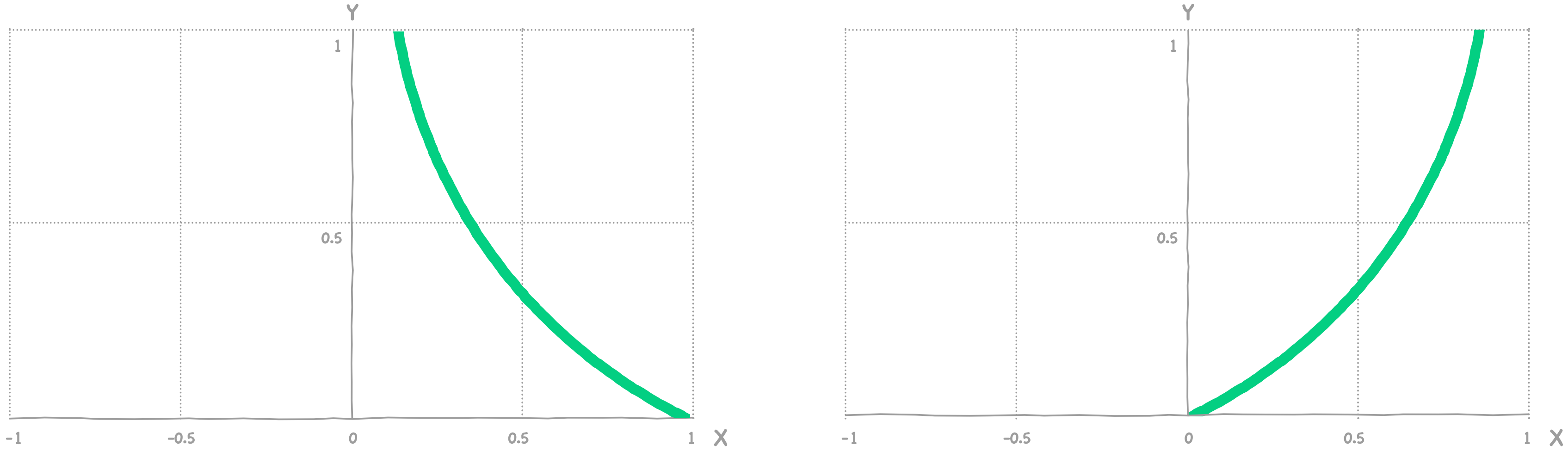
New Cost Function for Logistic

$$\mathrm{cost}(W) = \frac{1}{m}\sum c(H(x),y)$$

$$c(H(x),y) = \begin{cases} -\log(H(x)) & : y=1 \\ -\log(1-H(x)) & : y=0 \end{cases}$$

Understanding Cost Function

$$c(H(x),y) = \begin{cases} -\log(H(x)) & :y=1 \\ -\log(1-H(x)) & :y=0 \end{cases}$$



(A) -log(x)

(B) -log(1-x)

Cost Function

$$cost(W) = \frac{1}{m} \sum c(H(x), y) \qquad c(H(x), y) = \begin{cases} -\log(H(x)) & : y = 1 \\ -\log(1 - H(x)) & : y = 0 \end{cases}$$

$$C(H(x), y) = -y\log(H(x)) - (1-y)\log(1 - H(x))$$

$$= -(y\log(H(x) + (1-y)\log(1 - H(x)))$$

Minimize Cost

Gradient Descent Algorithm

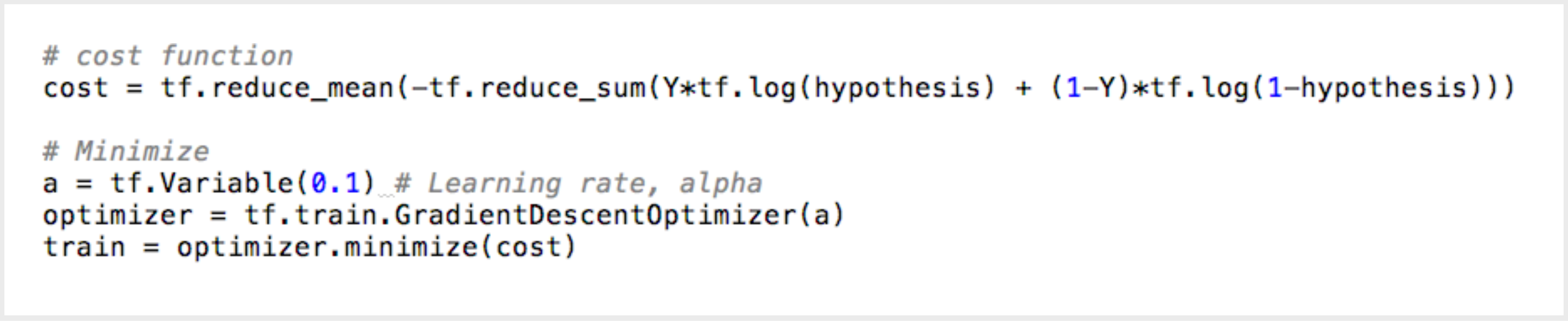
$$cost(W) = -\frac{1}{m}\sum y\log(H(x))+(1-y)\log(1-H(x))$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

Gradient Descent Algorithm

$$cost(W) = -\frac{1}{m}\sum y\log(H(x))+(1-y)\log(1-H(x))$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$



```
# cost function
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(hypothesis) + (1-Y)*tf.log(1-hypothesis)))

# Minimize
a = tf.Variable(0.1) # Learning rate, alpha
optimizer = tf.train.GradientDescentOptimizer(a)
train = optimizer.minimize(cost)
```

NEXT LECTURE

MULTINOMIAL CLASSIFICATION (SOFTMAX)