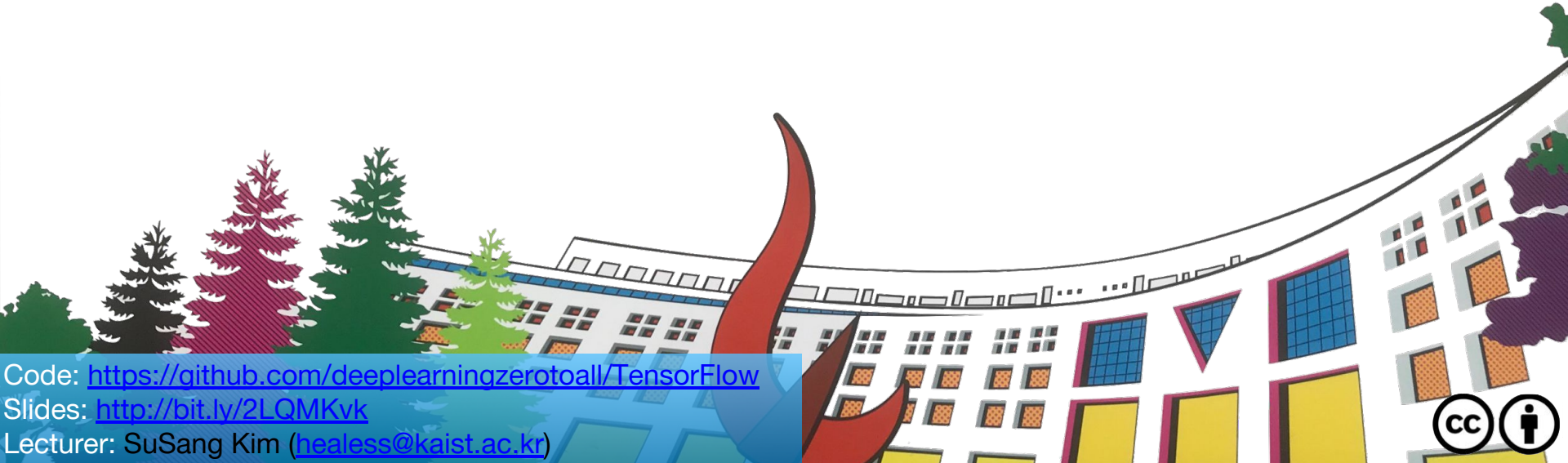
ML/DL for Everyone Season2

with TensorFlow

Lab 05-2 Logistic Regression

cost function & optimizer



Code: https://github.com/deeplearningzerotoall/TensorFlow
Slides: http://bit.ly/2LQMKvk
Lecturer: SuSang Kim (healess@kaist.ac.kr)

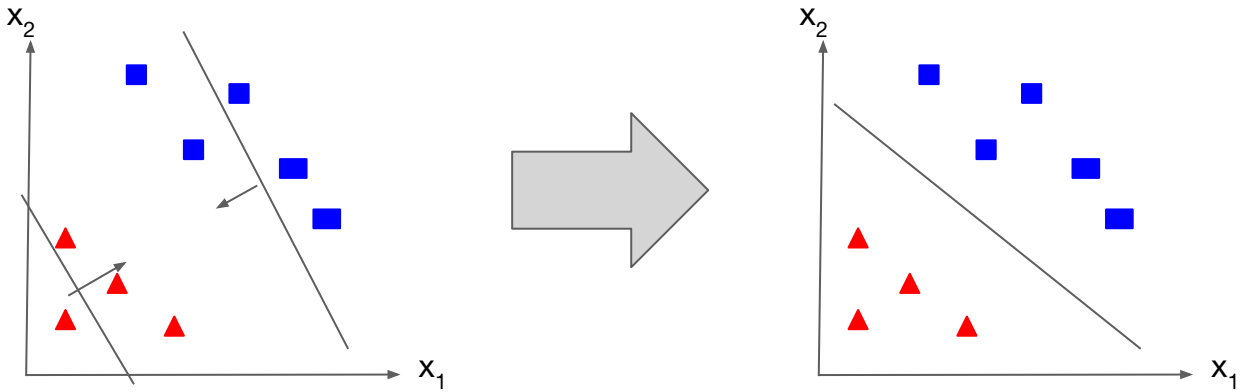
Logistic Regression

- What is Logistic Regression?
  - Classification
  - Logistic vs Linear
- How to solve?
  - Hypothesis Representation
  - Sigmoid/Logistic Function
  - Decision Boundary
  - Cost Function
  - Optimizer (Gradient Descent)
- Codes (Eager Execution)
- Summary

Andrew Ng's ML Class http://www.holehouse.org/mlclass/

Cost Function

the cost function to fit the parameters(θ)



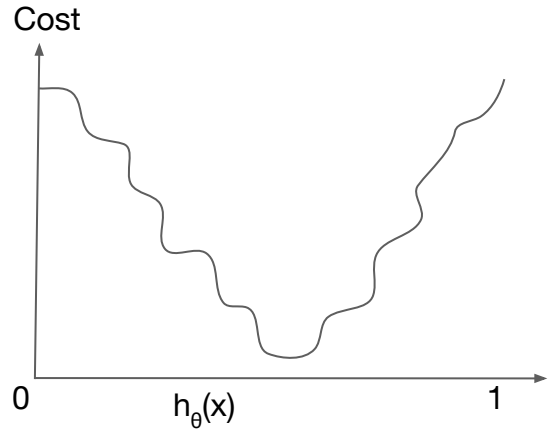
Given the training set how to we chose/fit θ? $h_\theta(x) = y$ then Cost = 0

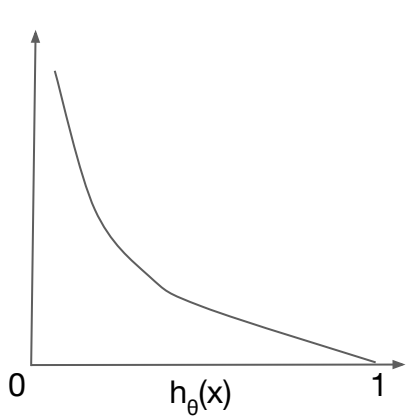
$$\text{cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1 - h_\theta(x))$$

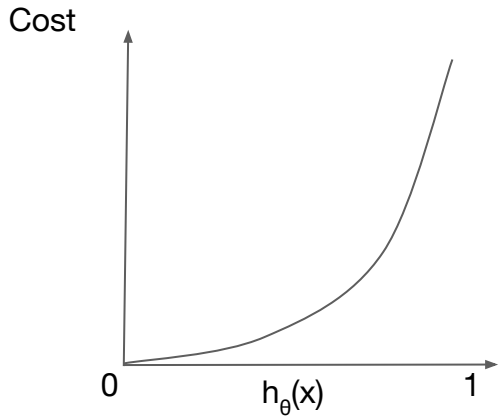
```
[Tensorflow Code]
def loss_fn(hypothesis, labels):
    cost = -tf.reduce_mean(labels * tf.log(hypothesis) + (1 - labels) * tf.log(1 - hypothesis))
    return cost
```

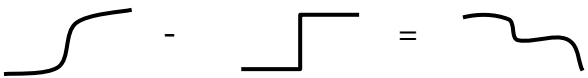
Cost Function

A convex logistic regression cost function









$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

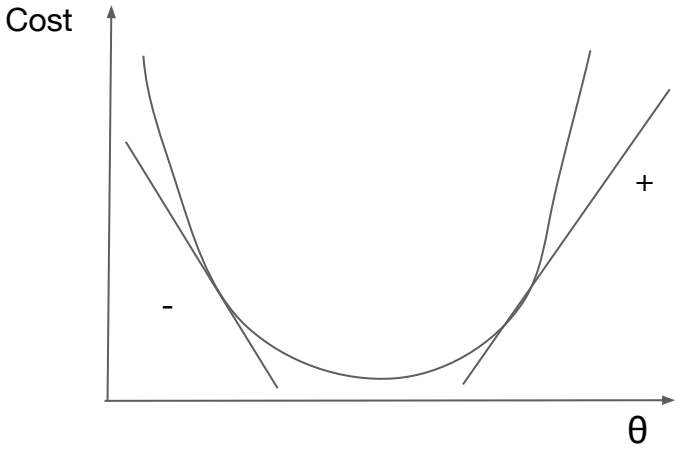
cost(h_θ (x),y) = -ylog(h_θ(x)) - (1-y)log(1- h_θ(x))

[Tensorflow Code]

```
cost = -tf.reduce_mean(labels * tf.log(hypothesis) + (1 - labels) * tf.log(1 - hypothesis))
```

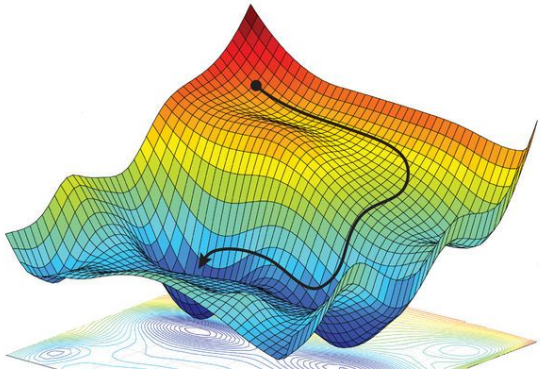
Optimization

How to minimize the cost function



$\text{cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1 - h_\theta(x))$

Repeat $\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \}$



```
[Tensorflow Code]
def grad(hypothesis, labels):
   with tf.GradientTape() as tape:
       loss_value = loss_fn(hypothesis, labels)
   return tape.gradient(loss_value, [W,b])
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.01)
optimizer.apply_gradients(grads_and_vars=zip(grads,[W,b]))
```

https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy

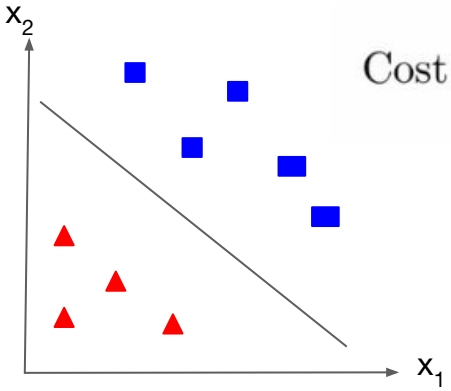
Code(Eager)

```
import tensorflow.contrib.eager as tfe
tf.enable_eager_execution()
dataset = tf.data.Dataset.from_tensor_slices((x_train, y_train)).batch(len(x_train))
W = tf.Variable(tf.zeros([2,1]), name='weight')
b = tf.Variable(tf.zeros([1]), name='bias')
def logistic_regression(features):
   hypothesis  = tf.div(1., 1. + tf.exp(tf.matmul(features, W) + b))
   return hypothesis
def loss_fn(hypothesis, labels):
   cost = -tf.reduce_mean(labels * tf.log(loss_fn(hypothesis) + (1 - labels) * tf.log(1 - hypothesis))
   return cost
def grad(hypothesis, features, labels):
   with tf.GradientTape() as tape:
       loss_value = loss_fn(hypothesis,labels)
   return tape.gradient(loss_value, [W,b])
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.01)
for step in range(EPOCHS):
   for features, labels  in tfe.Iterator(dataset):
       grads = grad(logistic_regression(features), features, labels)
       optimizer.apply_gradients(grads_and_vars=zip(grads,[W,b]))
       if step % 100 == 0:
              print("Iter: {}, Loss: {:.4f}".format(step, loss_fn(logistic_regression(features) ,labels)))

def accuracy_fn(hypothesis, labels):
   predicted = tf.cast(hypothesis > 0.5, dtype=tf.float32)
   accuracy = tf.reduce_mean(tf.cast(tf.equal(predicted, labels), dtype=tf.int32))
   return accuracy
test_acc = accuracy_fn(logistic_regression(x_test),y_test)
```

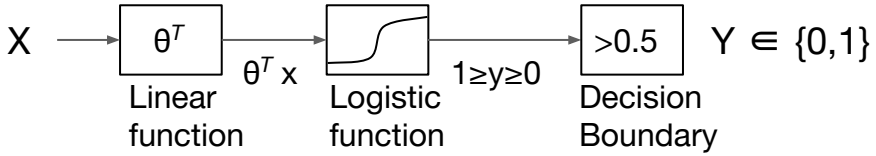
```
x_train = [[1., 2.],
          [2., 3.],
          [3., 1.],
          [4., 3.],
          [5., 3.],
          [6., 2.]]
y_train = [[0.],
          [0.],
          [0.],
          [1.],
          [1.],
          [1.]]

x_test = [[5.,2.]]
y_test = [[1.]]
```

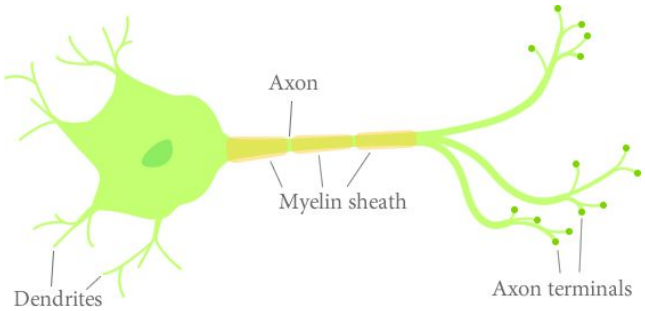
https://github.com/deeplearningzerotoall/TensorFlow/blob/master/lab-05-1-logistic_regression-eager.ipynb

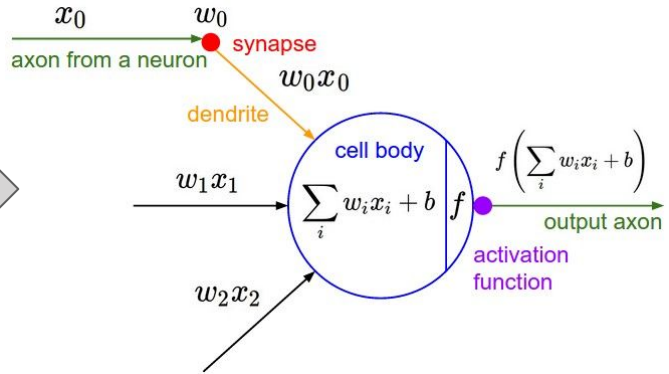
Summary



$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$







https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/neural_networks.html

What's Next?

Softmax classification