Week2 - Logistic Regression with Gradient Descent

Donnerstag, 16. Juli 2020 19:32

Logistic Regression:

Binary Classification problem.

Loss function (error):

Cost function:

Gradient descent:

Vectorizing Algorithms

Take out from programming assignment:

The first step to a classifier is commonly data preprocessing.

1. Check the dimensions and shapes of the datasets, including training dataset, test dataset, etc.
2. In case of images, reshape the datasets into vector of size (height * width * 3 , 1) may be helpful.
3. Standardize data.

After preprocessing of the datasets, the logistic regression itself is implemented by:

1. initialize parameters (w, b)
2. Forward propagation:
    a. sigmoid() or other transfer function
    b. activation A = sigmoid(w*X + b)
    c. cost function L
3. Back propagation:
    a. dw = dL/dw
    b. db = dL/db
4. Optimizing (w, b):
    a. for i in range(number of iterations)
        i. w = w - learning_rate * dw
        ii. b = b - learning_rate * db
5. Predict using optimized (w, b)
    a. if A = sigmoid(w*X + b) > 0.5:
        i. Y_predict = 1
    b. else:
        i. y_predic = 0
6. integrate all previous steps into a learning model

Note: This is the implementation of a logistic regression problem with gradient decent approach. The Forward / Back propagation involves Neural network thinking and will be looked into detail in the

further assignments.

Learning rate tuning is also an important part of designing a gradient descent model:

a. Large learning rate may not converge, it may also converge fast.
b. Small learning rate greatly decrease the speed of the learning process.