Day 4: Overfitting and Generalization

Summer STEM: Machine Learning

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

July 16, 2020

Outline

1. Review
2. Non-linear Optimization
3. Logistic Regression
4. Lab: Diagnosing Breast Cancer
5. Multiclass Classificaiton

Review

- Machine learning pipeline:
  - Process Data
  - Train on training data
  - Test on testing data
- Is it possible have a high accuracy for the training data and a low accuracy for the testing data? What should we do?

Review

- Imagine you are preparing for the SATs and you come across a book full of practice questions you did not understand how to solve any of the problems. However, you memorized all of the answers.
- What do you think will happen if you try to solve practice questions in a different book.
- Why are you studying actual problem solving techniques instead of just memorizing solutions from practice questions?
- Assuming you have an eidetic memory will memorizing solutions from practice questions be a good strategy?

Review

- $J(w) = \frac{1}{N}\|Y - Xw\|^2 + \lambda\|w\|^2$
- $w = [10000, 20000, 30000, 10000]$ does this look good?

Outline

Motivation

- Cannot rely on closed form solutions
  - Computation efficiency: operations like inverting a matrix is not efficient
  - For more complex problems such as neural networks, a closed-form solution is not always available
- Need an optimization technique to find an optimal solution
  - Machine learning practitioners use **gradient**-based methods

Gradient Descent Algorithm

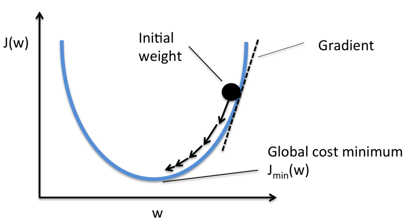
- Update Rule

  $Repeat\{$

  $$\mathbf{w}_{new} = \mathbf{w} - \alpha \nabla J(\mathbf{w})$$

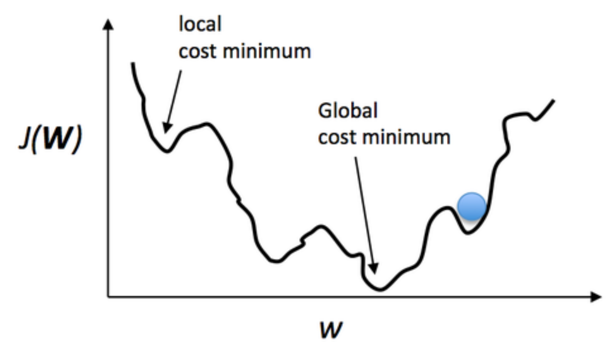
  $\}$

  α is the learning rate

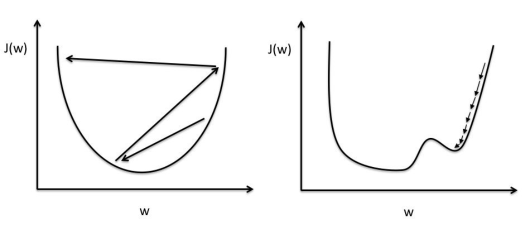


General Loss Function Contours

- Most loss function contours are not perfectly parabolic
- Our goal is to find a solution that is very close to global minimum by the right choice of hyper-parameters

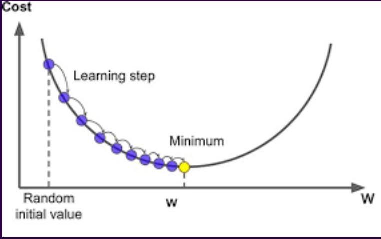


Understanding Learning Rate



Large learning rate: Overshooting.

Small learning rate: Many iterations until convergence and trapping in local minima.



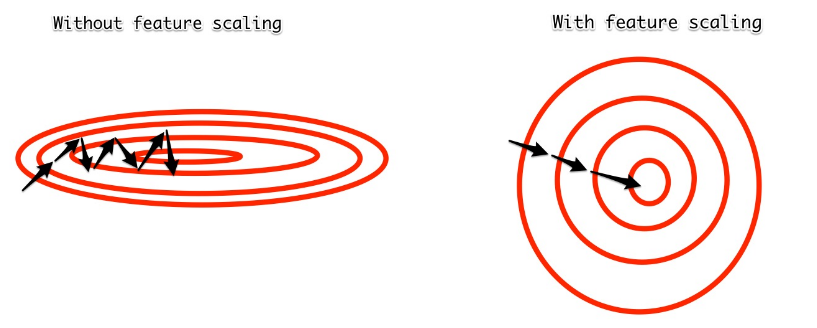
Correct learning rate

Some Animations

- Demonstrate gradient descent animation

Importance of Feature Normalization (Optional)

- Helps improve the performance of gradient based optimization



Some Gradient Based Algorithms

- Gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent
- Gradient descent with momentum
- RMSprop
- Adam optimization algorithm

We have many frameworks that help us use these techniques in a single line of code (Eg: TensorFlow, PyTorch, Caffe, etc).

Outline

Classification Vs. Regression

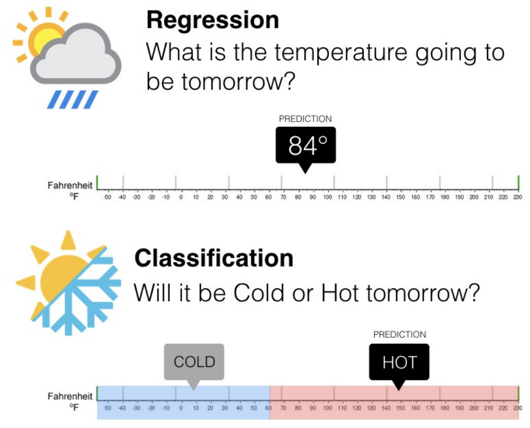
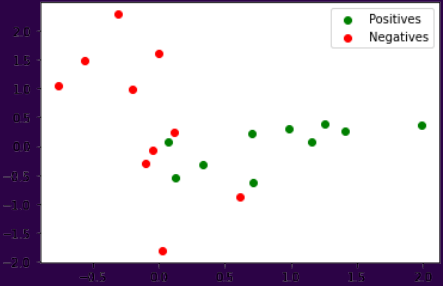


Figure: https://www.pinterest.com/pin/672232681855858622/?lp=true

Classification

Given the dataset (x_i, y_i) for $i = 1, 2, \ldots, N$, find a function $f(x)$ (model) so that it can predict the label $\hat{y}$ for some input x, even if it is not in the dataset, i.e. $\hat{y} = f(x)$.

- Positive : $y = 1$
- Negative : $y = 0$



Classification via regression

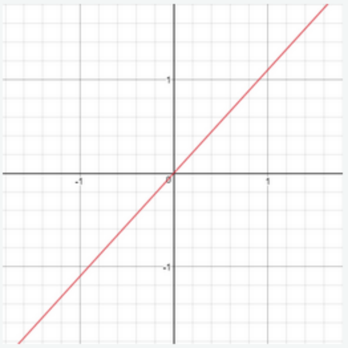
- Proposal: train a model to fit the data with linear regression (potentially with polynomial features)!

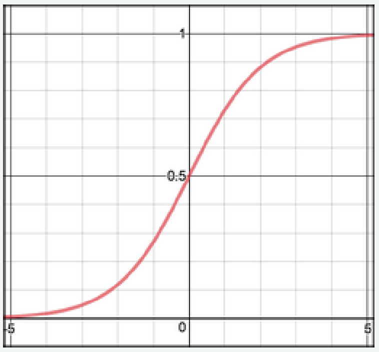
Classification via regression

- Proposal: train a model to fit the data with linear regression (potentially with polynomial features)!
- What could be the problem?

Sigmoid Function

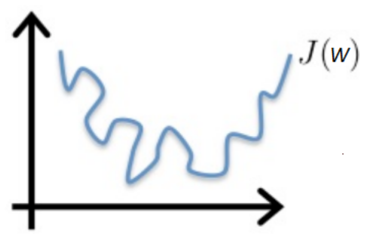
- Recall from linear regression $z = w_0 + w_1 x$
- By applying the sigmoid function to z, we enforce $0 \leq \hat{y} \leq 1$
  - $\hat{y} = sigmoid(z) = \frac{1}{1+e^{-z}}$

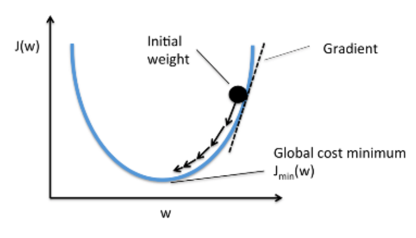




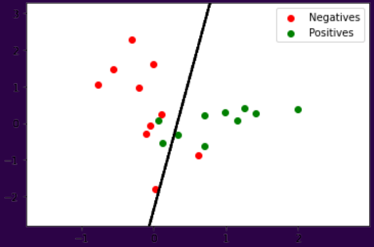
Classification Loss Function

- Cannot use the same cost function that we used for linear regression
  - MSE of a logistic function has many local minima
- Use $\frac{1}{N}\sum_{i=1}^{N}\Big[-y log(\hat{y}) - (1-y) log(1-\hat{y})\Big]$
  - This loss function is called binary cross entropy loss
  - This loss function has only one minimum





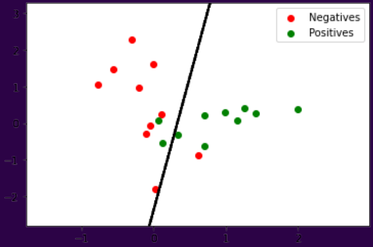
Decision Boundary



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}}$$

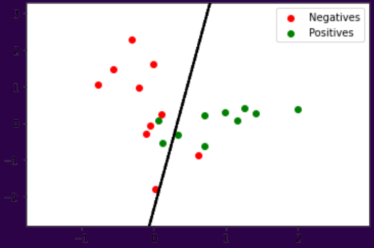
Decision Boundary



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}}$$

- What is the accuracy in this example ?



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}} = \frac{17}{20} = 0.85 = 85\%$$

Classifier

- How to deal with uncertainty ?

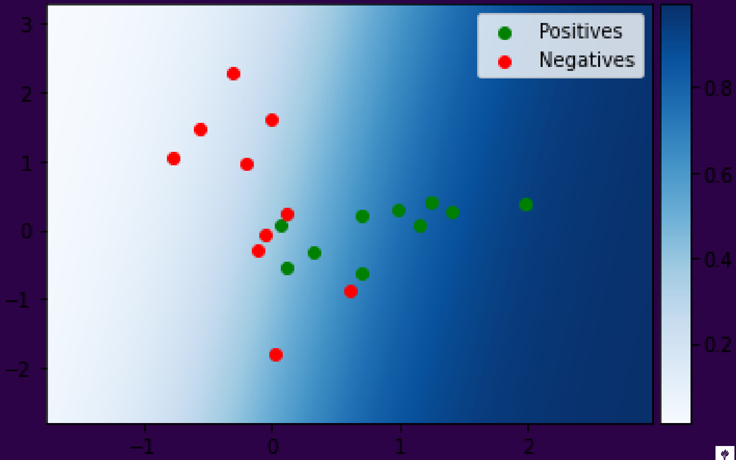
Classifier

- How to deal with uncertainty ?
  - $\hat{y} = f(x)$ should be between 0 and 1.

Classifier

- How to deal with uncertainty ?
  - $\hat{y} = f(x)$ should be between 0 and 1.
- If $\hat{y}$ is close to 0, the data is probably negative
- If $\hat{y}$ is close to 1, the data is probably positive
- If $\hat{y}$ is around 0.5, we are not sure.

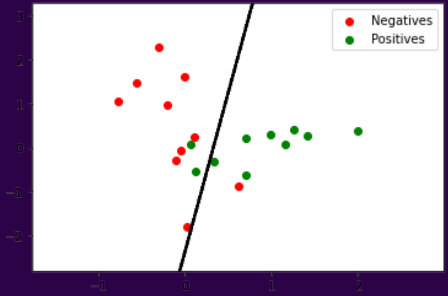
Classifier



Types of Errors in Classification

- Correct predictions:
  - True Positive (TP) : Predict $\hat{y} = 1$ when $y = 1$
  - True Negative (TN) : Predict $\hat{y} = 0$ when $y = 0$
- Two types of errors:
  - False Positive/ False Alarm (FP): $\hat{y} = 1$ when $y = 0$
  - False Negative/ Missed Detection (FN): $\hat{y} = 0$ when $y = 1$

Example



- How many True Positive (TP) are there ?
- How many True Negative (TN) are there ?
- How many False Positive (FP) are there ?
- How many False Negative (FN) are there ?

Performance metrics for a classifier

- Accuracy of a classifier:
  - (TP + TF)/(TP+FP+TN+FN) (percentage of correct classification)
- Why accuracy alone is not a good measure for assessing the model

Performance metrics for a classifier

- Accuracy of a classifier:
  - (TP + TF)/(TP+FP+TN+FN) (percentage of correct classification)
- Why accuracy alone is not a good measure for assessing the model
  - There might be an overwhelming proportion of one class over another (unbalanced classes)
  - Example: A rare disease occurs 1 in ten thousand people
  - A test that classifies everyone as free of the disease can achieve 99.999% accuracy when tested with people drawn randomly from the entire population

Other metrics

Some other metrics

- Sensitivity/Recall/TPR = TP/(TP+FN) (How many positives are detected among all positive?)
- Precision = TP/(TP+FP) (How many detected positives are actually positive?)
- Specificity/TNR = TN/(TN+FP) (How many negatives are detected among all negatives?)

Exercise: think of tasks for which sensitivity, precision, or specificity is a better metric.

Outline

Lab: Diagnosing Breast Cancer

- We're going to use the breast cancer dataset to predict whether the patients' scans show a malignant tumour or a benign tumour.
- Let's try to find the best linear classifier using logistic regression.

Outline

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
  - One-hot / 1-of-K vectors, ex : 4 Class
  - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
  - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
  - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
  - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
  - One-hot / 1-of-K vectors, ex : 4 Class
  - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
  - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
  - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
  - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$
- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(W^T \phi(\mathbf{x}))$
- Shape of $W^T \phi(\mathbf{x})$: $(K, 1) = (K, D) \times (D, 1)$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{z_k}}{\sum_j e^{z_j}}$

Multiclass Classificaiton

- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(\mathbf{z})$ with $\mathbf{z} = W^T\phi(\mathbf{x})$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{\mathbf{z}_k}}{\sum_j e^{\mathbf{z}_j}}$
- Softmax example: If $\mathbf{z} = \begin{bmatrix} -1 \\ 2 \\ 1 \\ -4 \end{bmatrix}$ then,

$$\text{softmax}(z) = \begin{bmatrix} \frac{e^{-1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{2}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{-4}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \end{bmatrix} \approx \begin{bmatrix} 0.035 \\ 0.704 \\ 0.259 \\ 0.002 \end{bmatrix}$$

Cross-entropy

- Multiple outputs: $\hat{\mathbf{y}}_\mathbf{i} = \text{softmax}(W^T \phi(\mathbf{x}_i))$
- Cross-Entropy: $J(W) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik})$
- Example : $\text{K} = 4$

$$\text{If, } \mathbf{y}_i = [0, 0, 1, 0] \text{ then, } \sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik}) = log(\hat{\mathbf{y}}_{i3})$$