Week 1

Quiz 1

1. A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T, as measured by P, improves with experience E. Suppose we feed a learning algorithm a lot of historical weather data, and have it learn to predict weather. What would be a reasonable choice for P?

Answer: The probability of it correctly predicting a future date's weather.

2. The amount of rain that falls in a day is usually measured in either millimeters (mm) or inches. Suppose you use a learning algorithm to predict how much rain will fall tomorrow. Would you treat this as a classification or a regression problem?

Answer: Regression

3. Suppose you are working on stock market prediction. You would like to predict whether or not a certain company will declare bankruptcy within the next 7 days (by training on data of similar companies that had previously been at risk of bankruptcy). Would you treat this as a classification or a regression problem?

Answer: Classification

4. Some of the problems below are best addressed using a supervised learning algorithm, and the others with an unsupervised learning algorithm. Which of the following would you apply supervised learning to? (Select all that apply.) In each case, assume some appropriate dataset is available for your algorithm to learn from.

Answer: Given genetic (DNA) data from a person, predict the odds of him/her developing diabetes over the next 10 years.

5. Which of these is a reasonable definition of machine learning?

Answer: Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.

Quiz 2

1. Consider the problem of predicting how well a student does in her second year of college/university, given how well she did in her first year. Specifically, let x be equal to the number of "A" grades (including A-. A and A+ grades) that a student receives in their first year of college (freshmen year). We would like to predict the value of y, which we define as the number of "A" grades they get in their second year (sophomore year).

x	y
3	2
1	2

x	y
0	1
4	3

Here each row is one training example. Recall that in linear regression, our hypothesis is hθ(x)=θ0+θ1x, and we use mmm to denote the number of training examples.

For the training set given above (note that this training set may also be referenced in other questions in this quiz), what is the value of mmm? In the box below, please enter your answer (which should be a number between 0 and 10).

Answer: m = 4

2. Consider the following training set of m=4 training examples:

x	y
1	0.5
2	1
4	2
0	0

Consider the linear regression model hθ(x)=θ0+θ1x that you would expect to obtain upon running gradient descent on this model? (Linear regression will be able to fit this data perfectly.)

Answer: θ0=0,θ1=0.5

3. Suppose we set θ0=−1,θ1=0.5. What is hθ(4)?

Answer: 1

4. Let fff be some function so that f(θ0,θ1) outputs a number. For this problem, f is some arbitrary/unknown smooth function (not necessarily the cost function of linear regression, so f may have local optima). Suppose we use gradient descent to try to minimize f(θ0,θ1) as a function of θ0 and θ1. Which of the following statements are true? (Check all that apply.)

Answer: If the learning rate is too small, then gradient descent may take a very long time to converge.

If θ0 and θ1 are initialized at a local minimum, then one iteration will not change their values.

5. Suppose that for some linear regression problem (say, predicting housing prices as in the lecture), we have some training set, and for our training set we managed to find some θ0, θ1 such that J(θ0,θ1)=0J(θ0,θ1)=0. Which of the statements below must then be true? (Check all that apply.)

Answer: Our training set can be fit perfectly by a straight line, i.e., all of our training examples lie perfectly on some straight line.

Quiz 3

1. A = [1 ; -4 ; -2 ; 1], B = [0 ; 3 ; 5 ; 8], A + B = ?

Answer: [1 ; -1 ; 3 ; 9]

2. Let x = [5 ; 5 ; 2 ; 7], 2 * x = ?

Answer: [10 ; 10 ; 4 ; 14]

3. Let u be a 3-dimensional vector, where specifically u = [8 ; 1 ; 4]. What is the transpose of u?

Answer: [8 1 4]

4. Let u and v be 3-dimensional vectors, where specifically u = [4 ; -4 ; -3], v = [4 ; 2 ; 4]. What is the result of the transpose of u times v?

Answer: -4

5. Let A and B be 3x3 (square) matrices. Which of the following must necessarily hold true? Check all that apply.

Answer: A+B=B+A ; If A is the 3x3 identity matrix, then A * B = B * A $A*B=B*A$

Week 2

Highlights

- Tricks for improving gradient descend

1. Feature scaling and mean normalization
2. Scrutinize the learning rate graph (Cost function and number of iterations):

- If J(theta) is decreasing, normal
- If J(theta) is increasing, use smaller alpha
- If J(theta) is increasing and decreasing alternatively, use smaller alpha

Quiz: Linear Regression with Multiple Variables

1.

Answer: mean = (7921+5184+8836+4761)/4 = 6675.5

deviation = 8836-4761 = 4075

normalized $x_2^{(4)}$ = (4761-6675.5) / 4075 = -0.50

2. You run gradient descent for 15 iterations with α=0.3 and compute J(θ) after each iteration. You find that the value of J(θ) decreases quickly then levels off. Based on this, which of the following conclusions seems most plausible?

Answer: α=0.3 is an effective choice of learning rate.

3. Suppose you have m = 28 training examples with n = 4 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is theta = (X^T X)^{-1} X^T yθ. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

Answer: X is 28×5, y is 28×1, θ is 5×1

4. Suppose you have a dataset with m = 1000000 examples and n = 200000 features for each example. You want to use multivariate linear regression to fit the parameters θ to our data. Should you prefer gradient descent or the normal equation?

Answer: Gradient descent, since (X^T X)^{-1} will be very slow to compute in the normal equation.

5. Which of the following are reasons for using feature scaling?

Answer: It speeds up gradient descent by making it require fewer iterations to get to a good solution.

Quiz: Octave/Matlab Tutorial

1. Suppose I first execute the following Octave/Matlab commands:

A = [1 2; 3 4; 5 6];

B = [1 2 3; 4 5 6];

Which of the following are then valid commands? Check all that apply. (Hint: A' denotes the transpose of A.)

Answer:

i) C = A * B: As A is of dimension 3*2, B is of dimension 2*3, A*B works

ii) C = B' + A: As B transpose is of dimension 3*2, A is of dimension 3*2, B'+A works

2.

Answer:

i) B = A(:, 1:2);

ii) B = A(1:4, 1:2);

3. Let A be a 10x10 matrix andxx be a 10-element vector. Your friend wants to compute the product AxAx and writes the following code:

```
v = zeros(10, 1);
for i = 1:10
  for j = 1:10
    v(i) = v(i) + A(i, j) * x(j);
  end
end
```

How would you vectorize this code to run without any for loops? Check all that apply.

Answer:

i) v = A * x;

ii) v = x' * A;

4. Say you have two column vectors v and w, each with 7 elements (i.e., they have dimensions 7x1). Consider the following code:

```
z = 0;
for i = 1:7
  z = z + v(i) * w(i)
end
```

Which of the following vectorizations correctly compute z? Check all that apply.

Answer:

i) z = sum (v .* w);

ii) z = v' * w;

5. In Octave/Matlab, many functions work on single numbers, vectors, and matrices. For example, the sin function when applied to a matrix will return a new matrix with the sin of each element. But you have to be careful, as certain functions have different behavior. Suppose you have an 7x7 matrix X. You want to compute the log of every element, the square of every element, add 1 to every element, and divide every element by 4. You will store the results in four matrices, A, B, C, D. One way to do so is the following code:

```
for i = 1:7
  for j = 1:7
    A(i, j) = log(X(i, j));
    B(i, j) = X(i, j) ^ 2;
    C(i, j) = X(i, j) + 1;
    D(i, j) = X(i, j) / 4;
  end
end
```

Which of the following correctly compute A, B, C, D? Check all that apply

Answer:

i) B = X.^2

ii) C = X+1

iii) D = X/4

Week 3

Highlights:

1. An advanced optimization algorithm (faster than gradient descent) is using `fminunc`

Without Regularization

```
function [jval,gradient] = costFunction(theta)
  jval =                 % code to compute J(theta)
  gradient = zeros(2,1) % initialize a size for gradient
  gradient(1) =          % code to compute gradient1
  gradient(2) =          % code to compute gradient2

options = optimset('GradObj', 'on', 'MaxIter', 100);
initialTheta = zeros(2,1);
[optTheta, functionVal, exitFlag] = fminunc(@costFunction, initialTheta,
options);
```

With Regularization

```
function [jval,gradient] = costFunction(theta)
  jval =                                     % code to compute J(theta)
  gradient = zeros(2,1)                      % initialize a size for gradient
  gradient(1) =                              % code to compute gradient1
  gradient(2) =    +(lambda/m)*theta1        % code to compute gradient2
```

Quiz1: Logstic Regression

1.Suppose that you have trained a logistic regression classifier, and it outputs on a new example x a prediction $h_{\theta}(x) = 0.7$. This means (check all that apply):

Answer:

i) Our estimate for P(y=1|x;θ) is 0.7.

ii) Our estimate for P(y=0|x;θ) is 0.3.

2. Suppose you have the following training set, and fit a logistic regression classifier $h_\theta(x)=g(\theta_0+\theta_1x_1+\theta_2x_2)$.

Answer:

i) Adding polynomial features could increase how well we can fit the training data.

ii) At the optimal value of θ (e.g., found by fminunc), we will have $J(\theta) \geq 0$. (As a linear decision boundary could not perfectly fit the dataset)

3.

4. Which of the following statements are true? Check all that apply.

A) The cost function $J(\theta)$ for logistic regression trained with $m \geq 1$ examples is always greater than or equal to zero.

B) For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum). This is the reason we prefer more advanced optimization algorithms such as fminunc (conjugate gradient/BFGS/L-BFGS/etc).

C) Since we train one classifier when there are two classes, we train two classifiers when there are three classes (and we do one-vs-all classification).

D) The one-vs-all technique allows you to use logistic regression for problems in which each $y^{(i)}$ comes from a fixed, discrete set of values.

Answer:

A) Correct.

B) Wrong. The reason of choosing advanced algorithms is because they don't need to choose alpha

C) Wrong. We need to train k classifier when there are k classes (k>1).

D) Correct.

5. Suppose you train a logistic classifier $h_\theta(x)=g(\theta_0+\theta_1x_1+\theta_2x_2)$. Suppose $\theta_0=6, \theta_1=-1, \theta_2=0$. Which of the following figures represents the decision boundary found by your classifier?

Answer:

Quiz2: Regularization

1. You are training a classification model with logistic

regression. Which of the following statements are true? Check

all that apply.

Answer:

A) Adding many new features to the model helps prevent overfitting on the training set.

> Wrong. This will lead to overfitting

B) Adding a new feature to the model always results in equal or better performance on the training set.

> Correct.

C) Introducing regularization to the model always results in equal or better performance on the training set.

> Wrong. Can lead to underfitting

D) Introducing regularization to the model always results in equal or better performance on examples not in the training set.

> Wrong. Underfitting will lead to worse performance on examples not in the training set.

2.

Answer: As when lambda = 1, we add the regularization term which will penalize when theta is big. Thus, when lambda = 1, theta will be relatively smaller than without regularization.

3. Which of the following statements about regularization are

true? Check all that apply.

Answer:

A) Consider a classification problem. Adding regularization may cause your classifier to incorrectly classify some training examples (which it had correctly classified when not using regularization, i.e. when λ=0).

> Correct.

B) Because logistic regression outputs values $0 \leq h_\theta(x) \leq 1$, its range of output values can only be "shrunk" slightly by regularization anyway, so regularization is generally not helpful for it.

> Wrong.

C) Using a very large value of λ cannot hurt the performance of your hypothesis; the only reason we do not set λ to be too large is to avoid numerical problems.

> Wrong. Very large λ can lead to underfitting problem.

D) Using too large a value of λ can cause your hypothesis to overfit the data; this can be avoided by reducing λ.

> Wrong. Very large λ leads to underfitting problem.

4.

Answer:

5.

Answer:

Week 4

Quiz 1: Neural Networks Representation

1.

Explanation:

A) Incorrect. The summation of the activation function is not necessary to be one.

B) Correct. Sigmoid function will always give value in (0,1).

C) Correct. Some logical functions like AND, OR can be represented by two layers only, while some logical functions like XNOR needs to be represented by multiple layers.

D) Incorrect. XOR can not be represented by two layers.

2.

Explanation:

3.

4.

Answer:

a2 = sigmoid (Theta1 * x);

5.

Week5

Highlights:

1. Matrix to Vector & Vector to Matrix

- When we are using the vectorized implementation, like `X * Theta'`, that is most of the cases, we will need the Matrix form of both Theta and D
- When we are using advanced optimization algorithms, like `fminunc`, we need to pass in the parameter as a vector That is

```
initialTheta = [Theta1(:);Theta2(:);Theta3(:)]                % get matrix
into a vector
... = fminunc[@costFunction, initialTheta, ...]

function[J, grad] = costFunction(ThetaVector)
Theta1 = reshape(ThetaVector(rowStart:rowEnd), row, column)  % get vector
into a matrix
...
grad = [grad1(:);...]                                         % get matrix
into a vector again
```

2. Gradient Checking

- This is to ensure that the backpropogation is working properly.
- This is done using numerical method:

```
epsilon = 1e-4;
for i = 1:n,
  thetaPlus = theta;
  thetaPlus(i) += epsilon;
  thetaMinus = theta;
  thetaMinus(i) -= epsilon;
  gradApprox(i) = (J(thetaPlus) - J(thetaMinus))/(2*epsilon)
end;
```

- Be sure to turn off gradient checking before actual learning.

3. Random Initialization of Theta

- Initialize to all zero might give the problem of **Symmetric breaking** redundant feature i.e. identical hidden units.

```
Theta1 = rand(row,col) * (2*INIT_EPSILON) - INIT_EPSILON;
```

4. The process for Neural Networks

- Randomly initialize weights
- Forward propogation to get $h_{theta}(x)$
- Compute cost function J(theta)
- Backward propogation to get partial derivatives (usually use a for-loop, loop through m examples)
- Compare the partial derivatives with the numerical method using gradient checking, then disable gradient checking
- Minimize J(theta) with gradient descent / advanced optimization algorithms

Quiz1: Neural Networks： Learning

1.

2.

Explanation:

Theta1 takes from row 1 to row 15(3*5), thus Theta2 will be from 16 to 39.(16 + 4*6 -1)

3.

Explanation:

$J(\theta+\epsilon) = 2*(1.01)^4 + 2 = 4.08120802$ $J(\theta-\epsilon) = 2*(0.99)^4 + 2 = 3.92119202$ $(J(\theta+\epsilon) - J(\theta-\epsilon))/ 2\epsilon = 8.0008$

4. Which of the following statements are true? Check all that apply.

Answer:

A) Using a large value of λ cannot hurt the performance of your neural network; the only reason we do not set λ to be too large is to avoid numerical problems.

> Wrong. Large λ will underfit the hypothesis as the regularization term is too significant.

B) Using gradient checking can help verify if one's implementation of backpropagation is bug-free.

> Correct.

C) If our neural network overfits the training set, one reasonable step to take is to increase the regularization parameter λ.

> Correct.

D) Gradient checking is useful if we are using gradient descent as our optimization algorithm. However, it serves little purpose if we are using one of the advanced optimization methods (such as in fminunc).

> Wrong. Gradient checking is useful to check if the backward propogation is bug-free, and both advanced optimization methods and gradient descent will use backward propogation.

5. Which of the following statements are true? Check all that apply.

Answer:

A) Suppose we are using gradient descent with learning rate α. For logistic regression and linear regression, $J(\theta)$ was a convex optimization problem and thus we did not want to choose a learning rate α that is too large. For a neural network however, $J(\Theta)$ may not be convex, and thus choosing a very large value of α can only speed up convergence.

> Wrong.

B) Suppose we have a correct implementation of backpropagation, and are training a neural network using gradient descent. Suppose we plot $J(\Theta)$ as a function of the number of iterations, and find that it is increasing rather than decreasing. One possible cause of this is that the learning rate α is too large.

> Correct.

C) Suppose that the parameter $\Theta(1)$ is a square matrix (meaning the number of rows equals the number of columns). If we replace Θ (1) with its transpose $(\Theta(1))^T$, then we have not changed the function that the network is computing.

> Wrong.

D) If we are training a neural network using gradient descent, one reasonable "debugging" step to make sure it is working is to plot $J(\Theta)$ as a function of the number of iterations, and make sure it is decreasing (or at least non-increasing) after each iteration.

> Correct.

Week 6

Highlights

1. Model Seclection - Evaluating Hypothesis

- Splitting the training set into a training set & validation set & a test set
- Learn theta from training set
- Get the cost using corresponding cost function from validation set / Or use the misclassfication matrix
- Then pick the hypothesis with the lowest cross validation error

Model Selection - Selecting regularization term

- Try lambda from 0 to 10 (0, 0.1, 0.4, 0.8,...)
- Get the cost using corresponding cost function from validation set
- Pick the one with the lowest cross validation error, then evaluate using the test set

2. Learning Curve conclusions:

- A model with high bias (Underfitting/J_{cv}&J_{train} is high): the accuracy will not increase as feeding more data to the model
- A model with high variance (Overfitting/J_{cv} is high J_{train} is low): the accuracy will increase as feeding more data to the model

3. Useful decisions:

- Getting more training examples: Fixes high variance
- Trying smaller sets of features: Fixes high variance
- Adding features: Fixes high bias
- Adding polynomial features: Fixes high bias
- Decreasing λ: Fixes high bias
- Increasing λ: Fixes high variance.

Quiz1: Advice for Applying Machine Learning

1.

Explanation:

> As test error and train error got flattened when m becomes larger, while the platform has larger value than expected.

2.

Explanation:

> The model has high variance. To reduce the variance, can choose to reduce number of features or feed more data (from learning curve).

3.

Explanation:

> The model has high bias (Underfitting). To reduce the bias, can choose to add number of features or decrease the regularization term.

4.

Explanation:

> A) Should not use test set in choosing regularization parameter.

> B) Correct.

> C) Correct.

> D) Should seperately train the train set, verify on the cross validation set and test set.

5. Which of the following statements are true? Check all that apply.

A) If a learning algorithm is suffering from high bias, only adding more training examples will not improve the test error significantly.

B) If a learning algorithm is suffering from high bias, adding more features is likely to improve the test error.

C) When debugging learning algorithms, it is useful to plot a learning curve to understand if there is a high bias or high variance problem.

D) We always prefer models with high variance (over those with high bias) as they will be able to better fit the training set.

Explanation:

> A) Correct, known from the learning curve (flattened).

> B) High variance -> overfitting -> adding features will not help

> C) Correct.

> D) Wrong.

Quiz2: Machine Learning System Design

1.

Explanation:

Accuracy = (true positives + true negatives) / (total examples) = (85+10) / (1000) = 0.095

2.

3.

Explanation: When the threshold has been decreased to 0.1,

> Recall = (true positives) / (true positives + false negatives), true positives are more, (true positives + false negatives) remains unchanged. Thus, recall increases.

> Precision = (true positives) / (true positives + false positives), true positives are more, (true positives + false positives) are more also. Thus, precision undetermined.

> Accuracy = (true positives + true negatives) / (total examples), true positives are more, (total examples) remains unchanged. Thus, accuracy increases.

4.

Explanation:

> When always predicting non-spam:

> Accuracy = (true positives + true negatives) / (total examples), (true positives + true negatives) is (0 + 99%), total example is 100%. Accuracy = 99%.

> Recall = (true positives) / (true positives + false negatives), true positive is 0, (true positives + false negatives) = actual positive = 1%, thus recall = 0.

> Precision = (true positives) / (true positives + false positives), true positive is 1%, (true positives + false positives) = predicted positive = 0%, thus precision = 0.

> When always predicting spam:

> Accuracy = (true positives + true negatives) / (total examples), (true positives + true negatives) is (1% + 0), total example is 100%. Accuracy = 1%.

> Recall = (true positives) / (true positives + false negatives), true positive is 1%, (true positives + false negatives) = actual positive = 1%, thus recall = 100%.

> Precision = (true positives) / (true positives + false positives), true positive is 1%, (true positives + false positives) = predicted positive = 100%, thus precision = 1%.

> A) Correct.

> B) Correct.

> C) Correct.

> D) Wrong.

5.

Explanation:

> A) When there are not enough useful features, feeding more data will not help.

> B) Not neccessary.

> C) Not always the case.

> D) Large dataset will not overfit the model.

> E) True.

Week 7

Highlights:

1. Popular Supervised Learning Algorithms

- Linear regression
- Logistic regression (simple classfication)
- Neural Networks (complex non-linear functions)
- Support Vector Machine: provides a large margin to seperate classes (complex non-linear functions)

> Question: What's the usage difference between NN and SVM?

2. SVM with Kernals

- **Kernals:** Can understand as Similarity function, and perticularly kernals go well with SVM not with other algorithms like logistic regression.

  Choices:

  - No kernal (linear kernal = logistic regression): when n is large, m is small
  - Guassian kernal: when n is small, m is large
  - Polynomial kernal
  - String kernal: when the input is text
  - ...

  Prerequisite:

  - Kernals must satisfy Mercer's theorem

  Choicing conditions:

  - n >> m: logistic / SVM without a kernal
  - n is small, m is intermediate: SVM with Guassian kernal
  - n << m: logistic / SVM without a kernal
  - NN works for all cases, but slower

- Hypothesis: compute f ($f^{(i)}_m$=similarity($x^{(i)}$,$l^{(m)}$))

  Predict 1 if $theta^T$ * f >= 0

- Training: minimize the new cost function for SVM

- A change on the regularizated cost term allows SVM to run more efficiently on larger dataset.

- SVM Parameters: C(=1/lambda), sigmaSquare(larger value, smoother, underfitting, higher bias; lower value, sharper, overfitting, higher variance)

Quiz1

1.

Answer:

It would be reasonable to try decreasing C. It would also be reasonable to try increasing σ^2.

> This decision boundary overfits the trainning dataset. We need to increase the regularization term, that is to decrease C. And we need the kernal to be smoother, that is to increase σ^2

2.

Answer:

When σ^2 is decreased, the kernal looks less smoother.

3.

Answer:

SVM requires a more precise boundary.

4.

Answer:

> A) We have n = 10 (small), m = 5000 (intermediate), using SVM with Guassian kernal is reasonable.

> B) *As now the model is underfitting the kernal, thus decreasing the regulatization term will help.*

> *D) As now the model is underfitting the kernal, thus more features will help.*

5.

JbhInZzjhgwHnpET

Week 8

Highlights

1. Popular Unsupervised Learning algorithms:

- Clutering algorithms
  - K-means algorithms (can be used for both seperated dataset and non-seperated dataset)
- Dimensionality Reduction
  - Principle component analysis (different with linear regression, PCA is to minimize the projection error between x1 and x2, linear regression is to minimize the vertical error from x to y)

2. PCA process

- Data preprocessing (feature scaling/mean normalization): make each feature has exactly zero mean & make the range of different features roughly the same
- Compute the covariance matrix
- Compute the eigenvectors of covariance matrix, and take the first k columns from the U matrix (into a $U_{reduced}$)
- z = $U_{reduced}$' * x
- Choosing k by the smallest value smaller than or equal to the pre-determined threshold

3. PCA Application

- Data compression: save memory & speed up algorithms
- Visualization(usually then k = 2 or 3)
- Warning: PCA cannot prevent from overfitting

Quiz1: Unsupervised Learning

1.

2.

Explanation:

> distance from [-1;2] to [1;2] = 2^2+0 = 4

> distance from [-1;2] to [-3;0] = 2^2+2^2 = 8

> distance from [-1;2] to [4;2] = 5^2+0 = 25

> Thus, this training example will be assigned to crentroid 1.

3.

4.

5.

Quiz2: Principle Component analysis

1.

Answer:

2.

3.

4.

5.

Explanation:

> C) PCA is a technique used for unsupervised learning

Week 9

Highlights

1. Anomaly detection algorithm (Unsupervised learning)

- Make use of the Gaussian distribution if you suspect that the dataset comes from a Gaussian distribution, compute the Gaussian distribution parameters for each feature
- Treat all parameters as independent events, get p(x) from the multiply combination of all features' Gaussian distribution

2. To evaluate an anomaly detection algorithm

- Assume we have labeled data (evaluate in a supervised way)
- Fit the model on the training set
- Evaluate on the cross validation set/test set according to the F1 score
- Pick the epsilon that gives the best F1 score

3. Anomaly detection algorithm VS supervised learning algorithm

- Anomaly detection algorithm: Small number of positive examples and a large number of negative examples; Supervised learning algorithm: Large number of positive and negative examples
- Anomaly detection algorithm: many types of anomalies (a totally new type of anomaly is possible); Supervised learning algorithm: future positive examples are likely to be the same type that the training data have

4. Applications of Anomaly detection algorithm

- Fraud detection
- Manufacturing (aircraft quality)
- Monitor machines in a data center

5. Multivariate Gaussian Distribution

- P(x) ~ Multivariate Gaussian(mu, sigma) where sigma is the covariance matrix, such that we will be able to capture an ellipse shape of cluster (towards any direction)

6. Recommender System

- Content Based Algorithm (we know the content of the product)
  - Some features defining type of product & some parameters (theta) defining each customer
  - To learn theta: find the theta that gives the smallest cost function J (a seperate linear regression for each user) using gradient descent
- Collaborative Filtering Algorithm

Type 1:

```
- Randomly initiate certain theta
- Compute features according the initialized theta using cost function and
gradient descent
```

```
- Using the features, compute a better set of theta using cost function
and gradient descent
- Going back and forth (Only we have large set of rated movies)
```

Type 2:

```
- Randomly initiate certain theta and feature values
- Minimize the cost function that contains both theta and feature using
gradient descent (that is to minimize these two simultaneously)
- Given a user, and a movie with certain feature value, predict rating.
```

Type 3:

```
- Given a product A with feature x<sub>a</sub>, find a product B with
feature x<sub>b</sub> that the distance from x<sub>a</sub> to
x<sub>b</sub> is small
- Predict product B as a similar product as product A
```

7. A prepocessing technique: Mean normalization

- User who has never rated any product in Recommender System
- According to the cost function, we will have 0 for all product for the user who hasn't rated any product.
- Using mean normalization, for each movie, all users' rating sum is tuned to be 0 (each entry - mean of the row)
- Apply the usual Collaborative Filtering algorithm
- Add the mean back to obtain the actual theta value

Quiz 1: Anomaly Detection

1.

Explanation:

> Number of types for transactions and diseases is countable and predictable. Thus should use a clustering algorithm

2.

Explanation:

> Increase the threshold so that we will have a large range for detecting anomalies.

3.

4.

Explanation:

> A) Correct.

> B) Usually we will have a skewed dataset for anomaly detection, thus accuracy may not be a good matrix to represent the model performance.

> C) Usually we have lots of normal training examples, and a few anomaly examples.

> D) Correct.

5.

Quiz 2:

1.

2.

3.

4.

5.

Week 10

Highlights

1. Techniques When Dealing with large dataset

- Gradient descent
  - Batch: compute all training examples at a time
  - Stochastic: first randomly reshuffle the dataset, then repeat compute the cost function for each training example, that is try to minimize the cost upon each iteration for each training example instead of computing all the examples at a time. Repeat the second step 1-10 times. **Note:** SGD may not give a global minimum finally and it's just meandering about the global minimum, to have a higher chance of getting closer to the global minimum, we can tune the learning rate smaller with every iteration.
  - Mini batch gradient descent (sometimes faster than SGD): use b (mini batch size) in each iteration (In-between), this can be sometimes faster than SGD as it can have a good vectorized implementation
- Online learning (when we are having a continuous stream of data coming in)
  - Can adapt to changing of user preferences (real-time)
  - Look at the data once at a time and then discard it, learn continuously
  - E.g. according to which link user clicked, show the products that users most likely to click on
- Map Reduce (run the ml problem in several machines)
  - Split the training set into several machines/several cores of a single machine (less network latency)
  - All the results will be sent to a master machine to combine them together

Quiz: Large Scale Machine Learning

1.

2.

3.

4.

Explanation:

> C) D) are using SGD, which already deceased the computational power required, thus we don't need to apply map reduce techniques to them again.

5.

Week 11

Highlights

1. Phone OCR Pineline (An inspiration on how to divide the work also)

- Text recognition
  - Can do a sliding window to a image patch
  - Do blend out the text area to a rectangle
- Character segmentation
  - Do a one-dimentional sliding window
- Character classification

2. Artificial Data Synthesis

- Before expanding training examples, should confirm have low-bias classifier

- Ask the question: how long does it take to get 10x the data?

  - Artificial Data Synthesis
  - manually label dataset
  - crowdsource (Amazon Mechanical Turk)

- Take characters in different fonts & put them into random background

- Take a existing image & do the artificial distortion to the image

2. Ceiling Analysis (give guidance on which part of the pipeline will improve the performance)

- Measure the accuracy on the overall
- Measure the accuracy on the overall system when the correct answer of certain (e.g. Text detection, from part 1 to the last part) is given by us
- Compare how much increased from the previous row to the next row, find the largest increased value, which is probably going to the most valuable component that you should be focus on and do improvement on

Quiz: Photo OCR

1.

Explanation:

> At scale 10*10, it requires ((1000-10)/2) * ((1000-10)/2) = 245025

> At scale 20*20, it requires ((1000-20)/2) * ((1000-20)/2) = 240100

> Thus, overall will be 245025 + 240100 = 485125

2.

Explanation:

> Time required for each labeller = 10,000/(4*60) = 41.67 hours

> Cost = 41.67 * 10 = 416.7 dollars

3.

4.

Explanation:

> Some of the artificial Synthesis ways will result in a non-car image.

5.