Machine Learning
Midterm Practice Problems

Some of these sample problems had been used in past exams and are provided for practice, in addition to the homework problems which you should also review. A typical exam would have around 5 questions worth a total of 100 points. The exam is closed book, no electronics. A single 8“x11” sheet of paper with typed or handwritten notes on both sides is allowed.

Table of Contents

1. Math and Probability Basics

Q1.1 Definitions

[a] Give the definition of an orthogonal matrix.

A is orthogonal $\Leftrightarrow$ A is a square matrix ($A \in \mathbb{R}^{n \times n}$)

and $A^{-1} = A^T$

[b] Give the definition of an eigenvector and eigenvalue.

$A \in \mathbb{R}^{n \times n}$ $\quad \alpha \in \mathbb{R}^n$

if $\exists k \in \mathbb{R}$ s.t. $A\alpha = k\alpha$, $\quad$ k is an eigenvalue of A

α is an eigenvector of A.

[c] How is the probability density function different from the cumulative probability distribution?

For cont. variable, let p be the pdf. F be the cpd.

$$\int_0^x p(t)\,dt = F(x)$$

$\int_{-\infty}^{\infty} p(t) = 1$, $\quad p(t)$ can may > 1 for some t, but $0 \le F(t) \le 1, \forall t$

[d] What is a 'singular' matrix?

For square matrix $A \in \mathbb{R}^{n \times n}$.

A is singular $\Leftrightarrow$ rank(A) < n

$\Leftrightarrow \neg \exists A^{-1}$, st. $A \cdot A^{-1} = I$

$\Leftrightarrow \det(A) = 0$

[e] Give the definition of Baye's Rule.

let A, B be 2 events

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Q1.2 Covariance Matrix

Recall that in the derivation of normal equations in class, we used the fact that the data covariance marix, i.e. a matrix whose element in the *k, j* position is the covariance between the k-th and j-th elements of the random input vector, is given b

$$\sum_{i=1}^{m} x^{(i)} x^{(i)T} = X^T X$$

where $x^{(i)}$ is the ith $nx1$ input vector, m is the number of input vectors in the dataset, and X is the mxn design matrix. Show that the above equality is true. Show all your steps.

$X \in R^{m \times n}$

$\therefore X^T X \in R^{n \times n}$.

$$X^T X_{(ij)} = \sum_{k=1}^{m} X^T_{ik} X_{kj}$$

$$= \sum_{k=1}^{m} X_i^{(k)} X_j^{(k)}$$

$$\left(\sum_{k=1}^{m} x^{(k)} x^{(k)T} \right)_{(ij)} = \sum_{k=1}^{m} \left(x^{(k)} x^{(k)T} \right)_{ij}$$

$$= \sum_{k=1}^{m} x_i^{(k)} x_j^{(k)}$$

$$= X^T X_{(ij)}$$

□.

Q1.3 Matrix Norm

The trace of a square matrix $A \in \mathbb{R}^{nxn}$ is defined as the sum of diagonal entries, or

$$\mathrm{tr}A = \sum_{i=1}^{n} A_{ii}.$$

Prove the following fact

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} = \sqrt{\mathrm{tr}(A^T A)}.$$

where $\|A\|_F$ is the matrix Frobenius norm. Show all your steps.

$\sqrt{tr(A^TA)}$

$= \sqrt{\sum_{i=1}^{n} (A^TA)_{ii}}$

$= \sqrt{\sum_{i=1}^{n}\sum_{k=1}^{m} (A^T_{ik}A_{ki})}$

$= \sqrt{\sum_{i=1}^{n}\sum_{k=1}^{m} A_{ki}^2}$

$= \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$

$= \|A\|_F$

Q1.4 Flu Virus Test

After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a flu virus, and that the test is 99% accurate (i.e., the probability of testing positive given that you have the virus is 0.99, as is the probability of testing negative given that you don't have the disease). The good news is that this is a rare virus, striking only one in 10,000 people (10^{-4}). What are the chances that you actually have the disease? Show your calculations as well as giving the final result. *Hint: use Baye's Rule.*

$$P(\text{disease}\mid\text{positive}) = \frac{P(\text{positive}\mid\text{disease})\,P(\text{disease})}{P(\text{pos}\mid\text{dis})\cdot P(\text{dis}) + P(\text{pos}\mid\text{no dis})\cdot P(\text{no dis})}$$

$$= \frac{10^{-4}\cdot 0.99}{10^{-4}\cdot 0.99 + 0.01\cdot 0.9999}$$

$$\approx 0.98\%.$$

2. Gradient Descent

Q2.1 Gradient Descent for a Cost Function

Suppose we have a cost function

$$J(\theta) = \frac{1}{m}(\sum_{i=1}^{m}(x_i^T\theta + by_i)) + \frac{1}{2}\theta^T A\theta,$$

where $\theta \in \mathbb{R}^n$ is the parameter vector, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, $\{x_i, y_i\}$ are m training data points, $A \in \mathbb{R}^{nxn}$ is a symmetric matrix, and $b \in \mathbb{R}$. We want to find parameters θ using gradient descent.

a) [3 points] Give the pseudo code for the gradient decent algorithm for a **generic** cost function $J(\theta)$ (not the specific one above).

while θ not converge:

$\theta_{t+1} = \theta_t - \eta \frac{\partial J(\theta)}{\partial \theta_t}$, where η is the learning rate.

b) [3 points] For the specific function above, what is the vector of partial gradients of the cost function, i.e. the vector with the jth element equal to $\frac{\partial}{\partial \theta_j}J(\theta)$?

$$\frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m}\sum_{i=1}^{m} x_i + A\theta.$$

$$\therefore \frac{\partial}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m} x_{i(j)} + \sum_{i=1}^{n} A_{ji}\theta_i$$

c) [3 points] What is the design matrix? Describe its entries and give its dimensions.

The design matrix is all of the datapoints. X_{ij} represents the j^{th} dimension of i^{th} observation

$$X \in \mathbb{R}^{m \times n}$$

d) [3 points] Re-write the expression for the gradient without using the summation notation $\sum$. *Hint: use the design matrix X.*

let $\alpha = [1, 1, \cdots 1] \in \mathbb{R}^m$.

$$\frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m}\alpha X + A\theta.$$

e) [3 points] Suppose we run gradient descent for two iterations. Give the expression for θ after two updates, with step size $\alpha = 1$ and initial value of $\theta = \mathbf{0}$ (vector of zeros).

$$\theta_1 = -\alpha \frac{\partial}{\partial \theta} J(0) = -\frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\theta_2 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1) = -\frac{1}{m}\sum_{i=1}^{m} x_i - \frac{1}{m}\sum_{i=1}^{m} x_i - A\theta_1$$

$$= -\frac{1}{m}\left(2\sum_{i=1}^{m} x_i + A\sum_{i=1}^{m} x_i\right)$$

f) [3 points] How do we know when the algorithm has converged?

We can use the L2 difference between θ.

when $\|\theta_{t+1} - \theta_t\|_2 \leq t$, it converges (maybe $t = 0.01$)

g) [3 points] Give the closed-form solution for θ. You do not need to prove it is the minimum of the cost.

$$\frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m}\sum_{i=1}^{m} x_i + A\theta = 0$$

$$\therefore A\theta = -\frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\theta = -\frac{1}{m}\sum_{i=1}^{m} A^{-1} x_i$$

3. Regression and Classification

Q3.1 Linear Regression: Online Art Auction

Imagine you work for an online art auctioneer. You would like to estimate the price y that a piece of art will sell for in an auction (in dollars), based on the following features:

x_1= type of art (out of 15 types such as 1:painting, 2:sculpture, etc.),
x_2= artist popularity (rank out of 100 artists),
x_3= estimated value in dollars,
x_4= days in the auction,
x_5= previously owned (binary),
x_6= is abstract (binary),
…etc.

For example, a feature vector for the i^{th} item could be $x^{(i)} = [1, 28, 1700, 5, 1, 0, \dots]$. You have collected data points from previous auction sales, $(x^{(i)}, y^{(i)}),\ i = 1, \dots, m$.

a. [3 points] You decide to use a linear regression model, $y = \sum_{j=0}^{n} \theta_j x_j$. In what circumstances should you use gradient descent vs normal equations to fit the parameters?

For a mse loss func, let X be the design matrix. If X^TX is singular we can not use normal equation, then we should use gd. If $(X^TX)^{-1}$ is easy to calculate. then we should use the normal equation.

b. [3 points] Suppose you decide to use gradient descent. How can you tell if it is converging?

We can check the delta difference of θ by a setting a threshold t.

If $\|\theta_{new} - \theta\|_2^2 \leq t$, we can call it converges.

c. [3 points] Suppose you're monitoring convergence and find it is slow. Name two things you can try to speed it up (be specific).

① learning rate, we can increase it to increase the learning step.

②. convergence threshold t. We can use a larger t to detect convergence.

d. [3 points] You want to add new features to improve your predictor. You consider adding total minutes spent in the auction. Is this a good idea? Why or why not?

Not a good idea. ① may overlap the dimension 'days in the auction' so that it won't perform good.

②. the minutes spent may not have linear relationship of price. We can have art very popular but in median price, ort very valuable but easy for auction.

③ If we want to make predict in the day, we can not get information of

this attribute in time. If changes too quickly.

$\sigma = \sqrt{\frac{1}{\beta}}$.

$\beta^{-1} = \sigma^2$

e. [3 points] Your boss does not know how much to trust your prediction $y^{test} = \$15{,}000$ for a certain watercolor painting. She asks you to estimate the probability of the painting selling for more than $\$20{,}000$. Give the equation for this probability, using a linear regression model that assumes the outputs have Gaussian noise with variance β^{-1}.

$$y_{true} \sim N(y^{test}, \beta^{-1})$$

$$\therefore P(y_{true} > 20000) = \int_{20000}^{+\infty} \sqrt{\frac{\beta}{2\pi}}\, e^{-\frac{\beta(x-15000)^2}{2}}\, dx$$

f. [2 points] What is the probability the painting will sell for more than $15,000?

According to the gaussian assumption, 50%.

g. [3 points] Suppose you now want to estimate the probability that a piece of art does not get any bids, $p(no\_bids|x)$, based on historic data. What sort of features and machine learning method should you use?

It's a classification problem. We should use logistic regression for it.

every $X_{(i)}$ is good feature, we can use some basis function to generate more features if need.

For example. X_1^2, X_2^2 ... X_1X_2, X_1X_3 ...

Q3.2 Softmax Classifier

Typically, we use the *softmax* function to model the probability of one of several classes given the input. Instead, consider what would happen if a binary classifier with output labels $j \in \{0,1\}$ used the softmax function to model the probability of the binary label $y = j$ given the input:

$$P(y = j|x) = \frac{e^{w_j{}^T x}}{\sum_{j=0,1} e^{w_j{}^T x}}$$

Where w_j is the parameter vector of the jth class.

a) [5 points] Show that the solution to this problem can be expressed as a logistic regression classifier with parameter w, and give the expression for w.

$$P(y=1|x) = \frac{e^{w_1{}^T x}}{e^{w_0{}^T x} + e^{w_1{}^T x}} = \frac{1}{1 + e^{(w_1{}^T - w_0{}^T)x}}$$

$$= \sigma((w_1{}^T - w_0{}^T)x)$$

$$\therefore w = w_1 - w_0$$

b) [5 points] Show that the posterior $P(y = j|x)$ is invariant if a constant vector b is added to both weight vectors w_j.

$$\forall b,\ P(y=j|x,b) = \frac{e^{(w_j+b)^T x}}{\sum_{i=0}^{1} e^{(w_i+b)^T x}} = \frac{e^{w_j{}^T x} \cdot e^{b^T x}}{\sum_{i=0}^{1} e^{w_i{}^T x} \cdot e^{b^T x}} = \frac{e^{w_j{}^T x}}{\sum_{i=0}^{1} e^{w_i{}^T x}}$$

$$= P(y=j|x,0)$$

So, $P(y=j|x)$ is invariant.

c) [5 points] (b) implies that the solution w_j is not unique. We can guarantee a unique solution by making the objective function regularized, e.g. using the squared norm regularizer. Write down the objective function and say whether you should minimize or maximize it.

Loss (cross entropy) y: true label

$$= -\left(y \log \frac{e^{w_1{}^T x}}{e^{w_0{}^T x} + e^{w_1{}^T x}} + (1-y) \log \frac{e^{w_0{}^T x}}{e^{w_0{}^T x} + e^{w_1{}^T x}}\right) + \frac{\lambda}{2}\left(\|w_0\|_2^2 + \|w_1\|_2^2\right)$$

we should minimize it

4. Overfitting and Regularization

Q4.1 Bias-Variance and λ

Alice has a binary classification dataset of m points with n-dimensional inputs.

a) [3 points] She has trained several regularized logistic regression models using regularization parameters $\lambda = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. She computed the cross-validation (CV) and training errors for each value of λ, shown in the table below, but the rows are out of order. Fill in the correct values of λ for each row.

Train error	*CV error*	λ
80%	85%	10^{-1}
40%	45%	10^{-3}
70%	76%	10^{-2}
35%	50%	10^{-4}

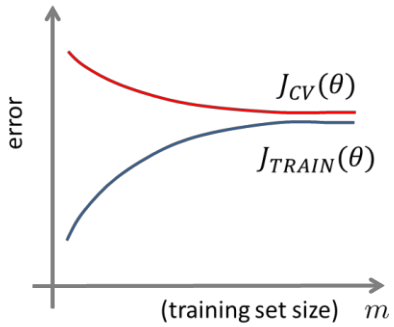
b) [3 points] Based on these results, which λ should she choose, and why?

$\lambda = 10^{-3}$ should be used. For the CV error is minimized. It is not overfitting nor underfitting.

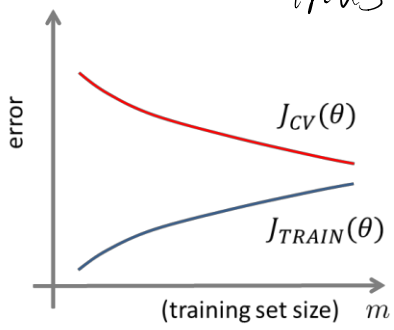
c) [3 points] Which of the four models will have the highest error due to variance? Why?

$\lambda = 10^{-4}$ will have highest error due to variance. Because we do not regularize, the overfitting will cause it's performance not stable, thus high variance.

d) [3 points] Alice also plotted learning curves for the models with $\lambda = 10^{-1}, 10^{-4}$. Match each plot with the correct value, and explain why it matches.



$\lambda = 10^{-1}$



$\lambda = 10^{-4}$

① J_{TRAIN} is lower at first if λ is smaller

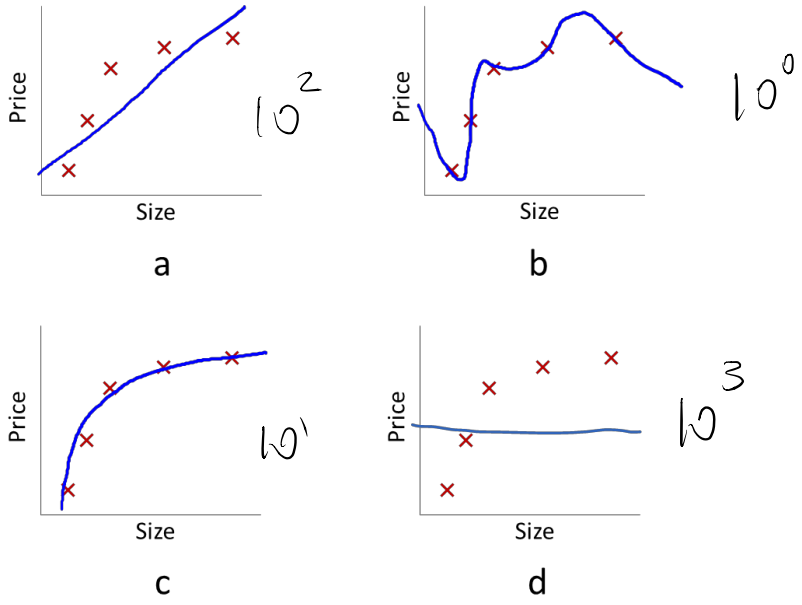
② The final convergence is better if λ is smaller (because of resolved).

Q4.2 Regularization for Linear Regression

Alice is trying to fit a linear regression model to predict house price based on size using polynomial features. Since her training dataset is very small, she is applying regularization. She fit several models by minimizing the cost function

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

for $\lambda = 10^0, 10^1, 10^2, 10^3$. The following are sketches of the resulting models.



a) [3 points] Which value of λ goes with each of the plots? (Write it next to the plot)

b) [3 points] Alice tries her model on a test set. Which model will have the highest error due to bias? d, because it predict the training set very hard.

c) [3 points] Which model will have the highest error due to variance? b. because it overfits to the training set, and will not perform consistently

d) [3 points] Which model, if any, will always have zero test error? c. because it seems this model fitts the data very well, not over nor under.

5. Maximum Likelihood Principle

Q5.1 ML for Probabilistic Linear Regression

Recall that probabilistic linear regression defines the likelihood of observing outputs $t^{(i)} \in \mathbb{R}$ given inputs $x^{(i)} \in \mathbb{R}^p$, where $i = 1, \dots, m$ and m is the number of samples in the dataset, as

$$p(t_1, \dots, t_m | x_1 \dots, x_m, \theta, \beta) = \prod_{i=1}^{m} N(t^{(i)} | h(x^{(i)}), \beta^{-1})$$

where $h(x)$ is the linear regression hypothesis, θ, β are parameters and $N(x|\mu, \sigma^2)$ is the normal (Gaussian) probability density with mean μ and variance σ^2. Here $\beta = \sigma^{-2}$ is the inverse variance of the Gaussian noise that we assume is added to the data.

(a) [8 points] Find β_{ML}, the maximum likelihood solution for β. *Hint: maximize log likelihood with respect to only β.*

$$\log p = \log \prod_{i=1}^{m} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(h(x^{(i)}) - t^{(i)})^2}$$

$$= \sum_{i=1}^{m} \frac{1}{2} \log \frac{\beta}{2\pi} - \frac{\beta}{2}(h(x^{(i)}) - t^{(i)})^2$$

$$= \frac{m}{2} \log \beta - \frac{m}{2} \log 2\pi - \frac{\beta}{2} \sum_i (hx - t)^2$$

$$\frac{\partial}{\partial \beta} \log p = \frac{m}{2\beta} - \frac{1}{2} \sum_{i=1}^{m} (h(x^{(i)}) - t^{(i)})^2 = 0$$

$$\therefore \beta_{ML} = \frac{m}{\sum_{i=1}^{m} (h(x^{(i)}) - t^{(i)})^2}$$

(b) [2 points] What is the interpretation of the solution β_{ML}? Explain in one sentence.

$\beta_{ML} = \frac{1}{\sigma^2}$ $\therefore \frac{1}{\beta_{ML}} = \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - t^{(i)})^2$ represents the variance.

or we can see β as 'precision', for the closer $h(x^{(i)})$ and $t^{(i)}$ get, the higher β_{ML} will be.

Q5.2 ML for Linear Regression with Multivariate Outputs

Consider a probabilistic linear regression model for a multivariate p-dimensional target variable $t = [t_1, \dots, t_p]^T$ that has a Gaussian distribution over t of the form

$$p(t|W, \Sigma) = N(t|y(\phi(x), W), \Sigma)$$

where $\phi(x)$ is a basis function representation of the input, and

$$y(x, W) = W^T \phi(x)$$

W is a matrix of parameters and Σ is the covariance parameter matrix. We are given a training dataset of input basis vectors x_n and corresponding target vectors t_n, $n = 1, \dots, N$. Show that the Maximum Likelihood solution W_{ML} for the parameter matrix W has the property that each column is given by

$$w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$$

where Φ is the design matrix. You do not need to provide a solution for Σ. Show all your steps..

Hint: the p-dimensional **multivariate normal distribution** is given by

$$N(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Hint: you may also find some of the matrix differentiation rules in the appendix helpful.

$$\log P(D|W, \Sigma) = \log \prod_{i=1}^{m} N(t | y(\phi(x_i), W), \Sigma)$$

$$= \log \prod_{i=1}^{m} \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(t - W^T\phi(x_i))\Sigma^{-1}(t - W^T\phi(x_i))}$$

$$= \sum_{i=1}^{m} \left(-\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(t - W^T\phi(x_i))^T \Sigma^{-1} (t - W^T\phi(x_i))\right)$$

$$\frac{\partial \log P}{\partial W} = \sum_i \left[\frac{\partial \log P}{\partial (t - W^T\phi(x_i))} \cdot \frac{\partial t - W^T\phi(x_i)}{\partial W}\right]$$

$$= \sum_{i=1}^{m} -\phi(x_i) \cdot -\left[t - W^T\phi(x_i)\right]^T \Sigma^{-1}$$

$$= 0$$

$$\therefore \sum_{i=1}^{m} \phi(x_i)^T t = \sum_{i=1}^{m} \phi(x_i)^T \phi(x_i) w$$

using design matrix,

$$\phi^T t = \phi^T \phi w$$

$$\therefore w = (\phi^T \phi)^{-1} \phi^T t$$

Q5.3 ML for Poisson Regression

This problem asks you to derive the maximum likelihood solution for a Poisson hypothesis.

a) [3 points] Given a training set $\{x_i, y_i\}$, Poisson regression models the probability of observing an output given an input as $p(y_i|\lambda) = \frac{1}{y_i!}\lambda^{y_i} e^{-\lambda}$ where $\lambda = e^{\theta^T x_i}$ for some parameter vector θ. Derive the cost function $J(\theta)$ corresponding to maximizing the log likelihood for a **single training example**.

$$J(\theta) = -\log P(y_i | x_i, \theta)$$

$$= \log(y_i)! - y_i \log\lambda + \lambda$$

$$= \log(y_i)! - y_i \theta^T x_i + e^{\theta^T x_i}$$

(we should minimize it, cause it's cost).

b) [3 points] The cost function in (b) has no closed form solution, so we must use an iterative method. Show the stochastic gradient descent update for this cost function.

$$\frac{\partial J(\theta)}{\partial \theta} = -y_i x_i + x_i e^{\theta^T x_i} \quad (\text{for given } i)$$

Using SGD, let us assume our batch has size m, the learning rate is η

$$\theta_{new} = \theta - \eta \frac{\partial J(\theta)}{\partial \theta}$$

$$= \theta - \eta \sum_{i=1}^{m} (x_i e^{\theta^T x_i} - y_i x_i)$$

6. Unsupervised Learning

Q6.1 Principle Component Analysis

a) PCA assumes a specific relationship between the unobserved latent coordinates z and the observed data points x. Express this relationship as an equation. Clearly identify and name the parameters which are learned.

b) Name one objective function which could be minimized to learn the parameters of PCA.

$f(w) = \sum_x \left| \left| x - \text{proj}_w(x) \right| \right|_2^2$ is the sum of square diffrence between x and it's projection on w.

c) For a dataset of arbitrary points $x^{(1)}, \ldots, x^{(m)}$, specify the steps of the PCA algorithm.

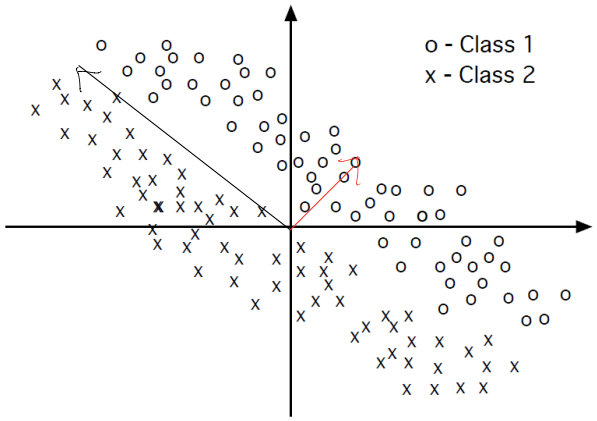
① Let X be the design matrix.

② SVD: $X = U \Sigma V^T$

③ if we want k main component.
$V_k = V[:, :k]$
$X' = \Sigma V_k$

④ if want to rebuild.
$X^* = \Sigma V_k V_k^T$

d) Suppose you are given 2D feature vectors for a classification task which are distributed according to the figure below. You apply PCA to the entire dataset. On the figure, draw all the PCA components.



e) In (d) above, could you use PCA components directly to classify the data (without training a classifier)? Explain.

In general, we can not because PCA only provides the principle component of our data, without any information of class label. However, because of the particular feature of (d), we can do classification on PCA directly. We just use the projection on the second principle component as our criteria, and set 0 as a threshold. If the projection >0, it's Class1 and otherwise it's class0.

Q6.2 Gaussian Mixture Models

a) Describe in words the two main steps of the Expectation Maximization algorithm used to solve Gaussian Mixture Models. E: Find the expectation of latent variable z for each datapoint

M: Maximize the likelihood of parameters μ, Σ, π given z.

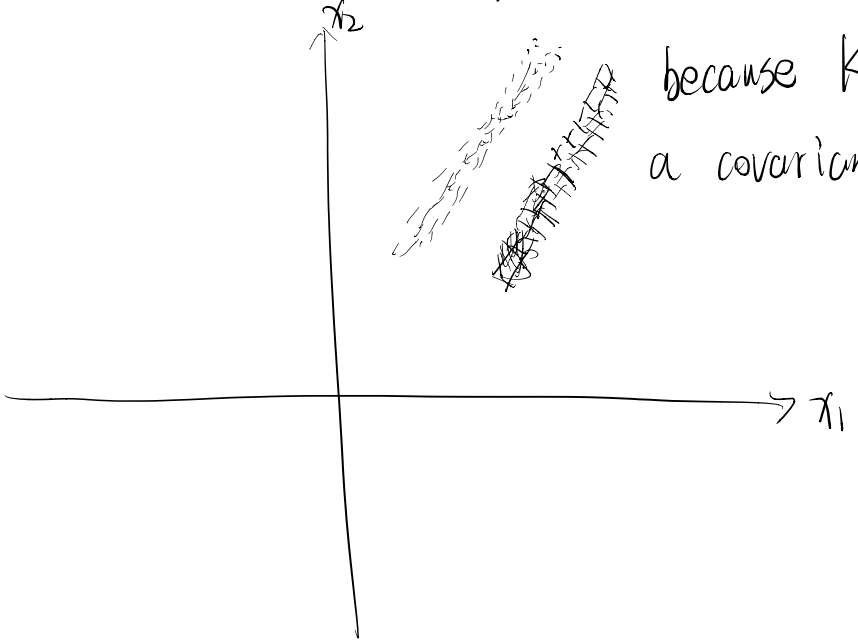
b) True or False: In the case of fully observed data, i.e. when all latent variables are observed, EM reduces to Maximum Likelihood.

True.

c) True or False: Since the EM algorithm guarantees that the value of its objective function will increase after each iteration, it is guaranteed to eventually reach the global maximum.

False.

d) Sketch a dataset on which K-Means would work poorly but a Gaussian Mixture Model with the same number of clusters would do well. Describe why K-Means wouldn't work well.

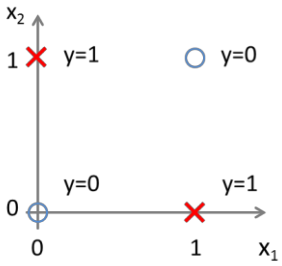


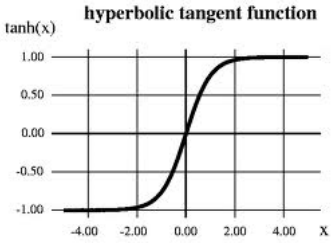
because K-Means don't have a covariance Matrix.

7. Neural Networks

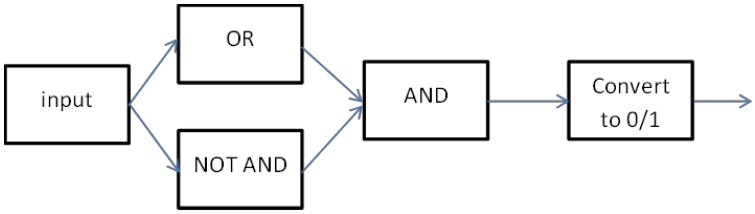
Q7.1 Neural Network for XOR

Design a neural network to solve the XOR problem, i.e. the network should output 1 if only one of the two binary input variables is 1, and 0 otherwise (see left figure). Use the hyperbolic tangent, or *tanh,* activation function in all nodes (right figure), which ranges in [-1, +1].

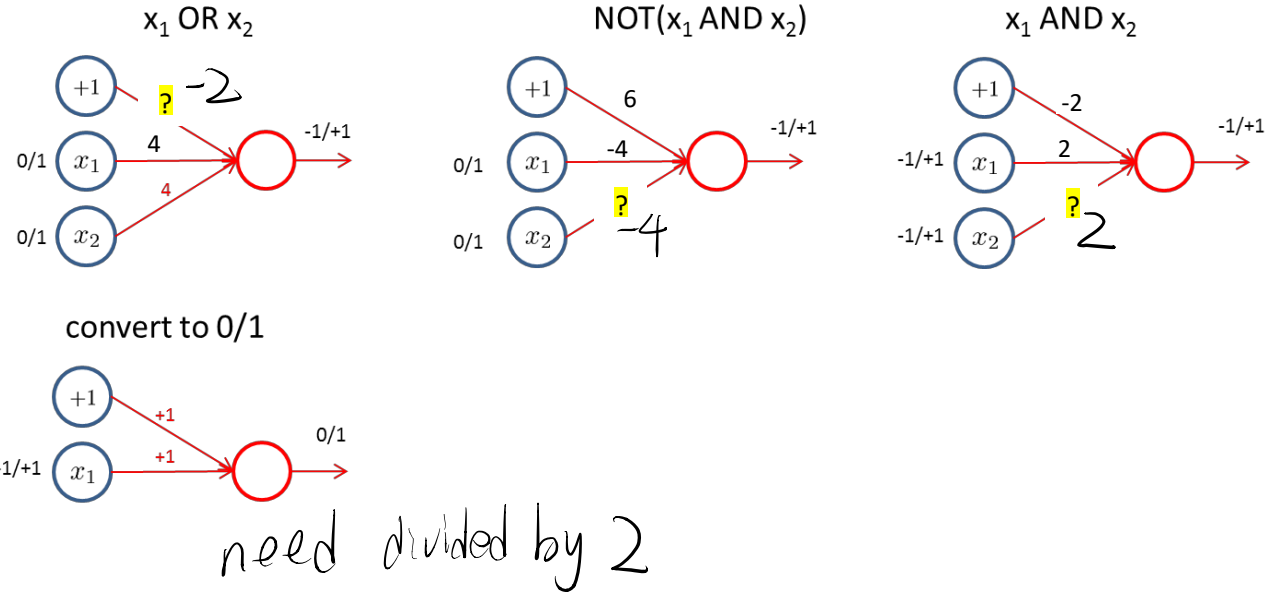




Note that (A XOR B) can be expressed as (A OR B) AND NOT(A AND B)), as illustrated below:



In the diagrams below, we filled in most of the tanh units' parameters. Fill in the remaining parameters, keeping in mind that tanh outputs +1/-1, not 0/1. Note that we need to appropriately change the second layer (the AND node) to take +1/-1 as inputs. Also, we must add an extra last layer to convert the final output from +1/-1 to 0/1. *Hint: assume tanh outputs* -1 *for any input* $x \leq -2$, $+1$ *for any input* $x \geq +2$, 0 *for* $x = 0$.



Q7.2 Computation Graph and Backpropagation

In class, we learned how to take a complex function that consists of multiple nested functions and represent it with a computation graph, which allows us to write down the forward and backward pass used to compute the function gradient.

a) Practice converting different functions $f_\theta(x) = f_k(f_{k-1}(\dots f_1(x))$ of input vector x parametrized by θ to their computation graphs.

$x \longrightarrow f_1(x) \longrightarrow f_2(f_1(x)) \cdots \longrightarrow f_k(f \cdots f_1(x))$

b) For the computation graphs obtained in (a), write down the forward pass and the backward pass equations.

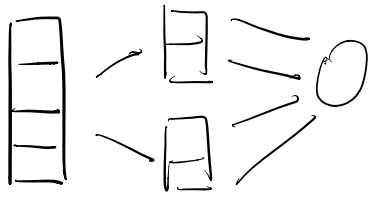
forward: $a_i = f_i(a_{i-1})$, $a_0 = x$, for $i \le k$

backward: $\frac{\partial f_\theta}{\partial a_i} = \frac{\partial f_\theta}{\partial a_{i+1}} \cdot f'_{i+1}(a_i)$, for $0 \le i \le k-1$

$\frac{\partial f_\theta}{\partial a_k} = f'_k(a_k)$

Q7.3 Neural Network Architectures

a) Draw a convolutional network with input $x \in R^4$, one hidden layer with 2x1 filters and 2 channels with stride 2, and a fully-connected output layer with one neuron. How many parameters does this network have?



W: 2×2+4

b: 2×2+1

total = 13

b) What algorithm is used for learning the parameters of a recurrent network? Name the algorithm and sketch out its main steps.

back propagate through time

for each output, $(O_1, O_2 \cdots O_n)$, let it be O_i.

repeat itimes iterate to update W in hidden layer, use chain rule to compute the gradient every time.

Appendix: Useful Formulas

Matrix Derivatives

For vectors x, y and matrix A,

$y = Ax$, then $\frac{\partial y}{\partial x} = A$

If $z = x^T Ax$, then $\frac{\partial z}{\partial x} = x^T(A + A^T)$. For the special case of a symmetric matrix A, $\frac{\partial z}{\partial x} = 2x^T A$.

Chain Rule: if z is a function of y, which is a function of A, then $\frac{\partial y}{\partial A} = \frac{\partial y}{\partial A}\frac{\partial z}{\partial y}$ (note the order).

Single-Dimension Normal Distribution

$$N(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Multivariate Normal Distribution

The p-dimensional multivariate normal distribution with mean μ and covariance matrix Σ is given by

$$N(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$