1. General Concepts (1/2)

True or False

For the true/False answers, give a one sentence explanation of each answer; answers without explanation will not be given any points.

a) Suppose we use polynomial features for linear regression, then the hypothesis is linear in the original features [T/F]

Answer: false, it is linear in the new polynomial features

b) Maximum likelihood can be used to derive a closed-form solution to logistic regression [T/F]

Answer: false, it can be used to derive cost, but no closed form solution exists

c) The gradient descent update for logistic regression is identical to linear regression [T/F]

Answer: false, they look similar but the hypothesis is different

d) Changing the prior in Linear Discriminant Analysis changes the direction of the decision hyperplane [T/F]

Answer: false, changes only the position, not the direction of the decision hyperplane. The hyperplane will move further from the more likely class

e) One example of a discriminative classification model is logistic regression.

Answer: true, others are neural network and SVM

2. General Concepts (2/2)

Short answer questions

Answer the following questions in brief one to two sentence answers.

a) For a training dataset $\mathrm{D} = \{x_i, y_i\}$ where x_i are the inputs and y_i are the output, explain the difference between discriminative and generative classification models.

Answer: A generative model learns $p(x, y)$ which can *generate* examples $\{x, y\}$ and evaluate $p(y)$, $p(y|x)$ and $p(x|y)$, while a discriminative model only learns $p(y|x)$ or learns the decision boundary directly without modeling the output probability (eg. in the SVM).

b) Give one example of a generative model.

Answer: linear discriminant analysis (LDA)

c) What is cross-validation?

Answer: When we split training data into training and extra validation set; learn model parameters on the training set, test and tune hyper-parameters on the validation set. We can do this multiple times, taking a different portion of original training data for the validation set each time (known as N-fold cross-validation).

d) How can we use cross-validation to prevent overfitting? Explain the procedure using the setup of (d).

Answer: Train several models using different values of λ on the training set, test them on the validation set, and pick best model. This will reduce overfitting compared to tuning λ on training data.

3. Error Metrics

a) Give one example each of error metrics that can be used to evaluate: classification, regression, clustering.

Answer: accuracy, squared error, distance to cluster centers.

b) Which are the correct definitions of precision and recall? Here 'actual positives' are examples labeled positive (by humans), and 'predicted positives' are examples for which the algorithm predicts a positive label.

1. $precision = \frac{true\ positives}{predicted\ positives}$
2. $precision = \frac{true\ positives}{actual\ positives}$
3. $recall = \frac{predicted\ positives}{actual\ positives}$
4. $recall = \frac{true\ positives}{actual\ positives}$

Answer: 1 and 4

4. Bayesian Methods

Alice has a dataset of m points with n-dimensional inputs and scalar outputs. She has trained several regularized linear regression models using regularization parameters $\lambda = e^0, e^{-1}, e^{-2}, e^{-3}$.

a) Which parameter will lead to highest bias? To highest variance?

Answer: since λ weights the regularizer, smallest λ will lead to highest variance and largest λ will lead to highest bias.

b) Alice then decides to use a Bayesian approach to control the complexity of the model. What is the Bayesian equivalent to changing λ?

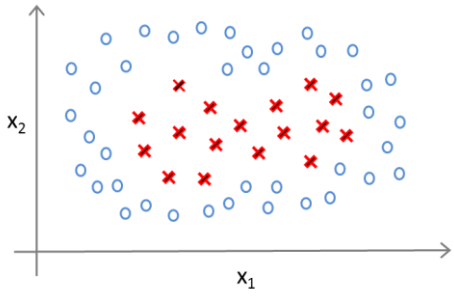
Answer: changing the prior on the parameters of the model

c) Which Bayesian model should she use? Explain what makes the model Bayesian.

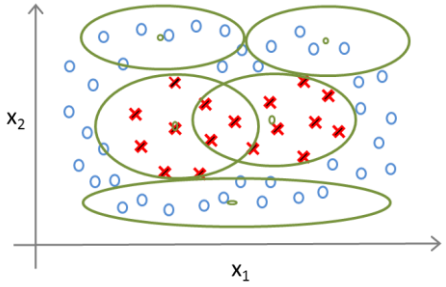
Answer: Bayesian linear regression, it puts a prior distribution over the linear parameters

5. SVMs and Kernels

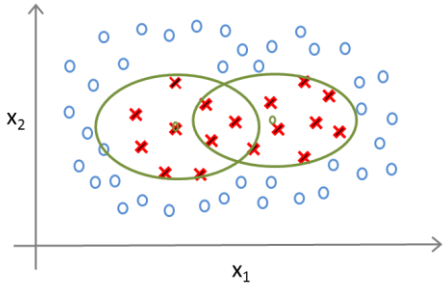
Consider the following dataset of 2-dimensional datapoints:



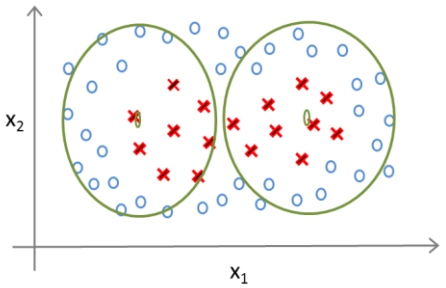
a) Which placement of Gaussian basis functions corresponds to a kernel feature representation for this dataset? Explain your answer in one sentence below.



(A)



(B)



(C)

(D) none of the above

Answer: (D); to compute a kernel representation, we place a separate Gaussian distribution centered at **each** data point.

b) How does increasing the variance of the Gaussian σ^2 affect the bias and variance of the resulting Gaussian Kernel SVM classifier? Explain.

Answer: it makes the boundary smoother, so the classifier has higher bias and lower variance

c) For $k(x_1, x_2)$ to be a valid kernel, there must be a feature basis function $\varphi(.)$ such that we can write $k(x_1, x_2) = \varphi(x_1)^T \varphi(x_2)$. Suppose $k_1(x_1, x_2)$ and $k_2(x_1, x_2)$ are valid kernels. Prove that the following is also a valid kernel:

$$k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2)$$

Answer:
$k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2) = \phi_1(x_1)^T \phi_1(x_2) + \phi_2(x_1)^T \phi_2(x_2) = \phi(x_1)^T \phi(x_2)$ where $\phi(x) = [\phi_1(x), \phi_2(x)]$

d) Both SVMs with Gaussian kernels and Neural Networks with at least one hidden layer can be used to learn non-linear decision boundaries, such as the boundary between positive and negative examples in the dataset above. Describe the main similarity and the main difference in how these two approaches achieve this.

Answer: The similarity is that both can be thought of as mapping the input features x into a new feature space: the SVM kernel maps x to a new feature vector $\phi(x)$, and the hidden layer of the neural network maps x to the activations of the last hidden layer, a. The difference is in how the mapping is done, where SVM uses training data as landmarks, but neural network learns the feature mapping through layer parameters.

e) Explain what slack variables are used for when training SVMs.

Answer: slack variables are assigned to solve a problem that is not linearly separable by adding 'slack' values to the constraints

6. Overfitting and Regularization

Q6.1 Bias-Variance and λ

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

a) 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%	
40%	45%	
70%	76%	
35%	50%	

Answer: $10^{-1}, 10^{-3}, 10^{-2}, 10^{-4}$

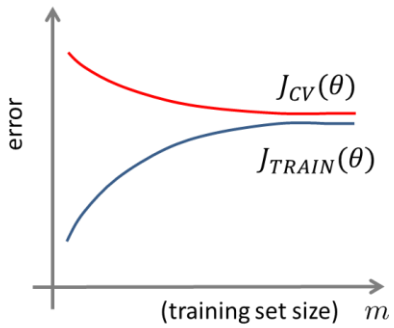
b) Based on these results, which λ should she choose, and why?

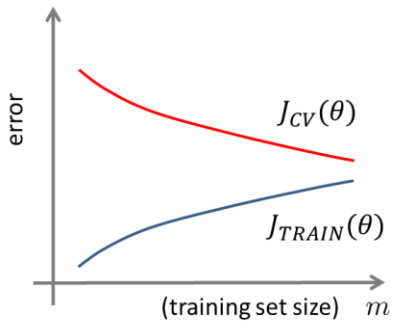
Answer: 10^{-3}, the one with lowest CV error has best generalization

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

Answer: 10^{-4}, it is has the least regularization and is the most complex

d) 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.





$\lambda =$ $\lambda =$

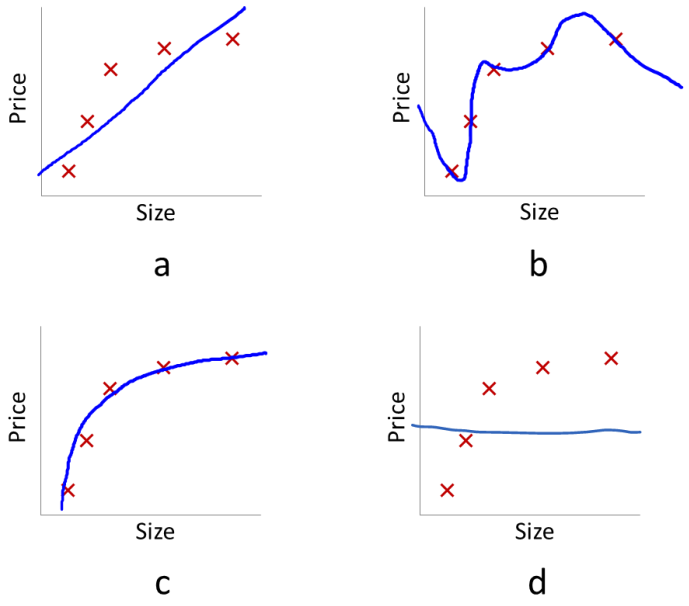
Answer: Left: 10^{-1} because it has high bias, more data doesn't help; Right: 10^{-4} because it has high variance, more data may help.

Q6.2 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)

Answer: $a: 10^2$ $b: 10^0$ $c: 10^1$ $d: 10^3$

b) [3 points] Alice tries her model on a test set. Which model will have the highest error due to bias?

Answer: d

c) [3 points] Which model will have the highest error due to variance?

Answer: b

d) [3 points] Which model, if any, will always have zero test error?

Answer: none

7. Maximum Likelihood Principle

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) Find β_{ML}, the maximum likelihood solution for β. *Hint: maximize log likelihood with respect to only β.*

Answer: maximize log likelihood w.r.t. β. The likelihood function is:

$$\ln p(\boldsymbol{t}|\boldsymbol{x}, \theta, \beta) = -\frac{\beta}{2}\sum_{i=1}^{m}\left(h(x^{(i)}) - t^{(i)}\right)^2 + \frac{m}{2}\ln\beta - \frac{m}{2}\ln(2\pi)$$

Take partial derivative w.r.t. β and set it to 0, then solve for w.r.t. β

$$-\frac{1}{2}\sum_{i=1}^{m}\left(h(x^{(i)}) - t^{(i)}\right)^2 + \frac{m}{2}\frac{1}{\beta} = 0$$

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

To get β_{ML} we just need to take the inverse of the right-hand side.

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

Answer: It is the inverse of the average squared error of the prediction.

8. Unsupervised Learning

Q8.1 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.

Answer: $z_k = {u^{(k)}}^T \bar{x}$ where $\bar{x}$ is a normalized data point and $u^{(k)}, k = 1, \dots, K$ are the principle component vectors, which are the parameters that need to be learned.

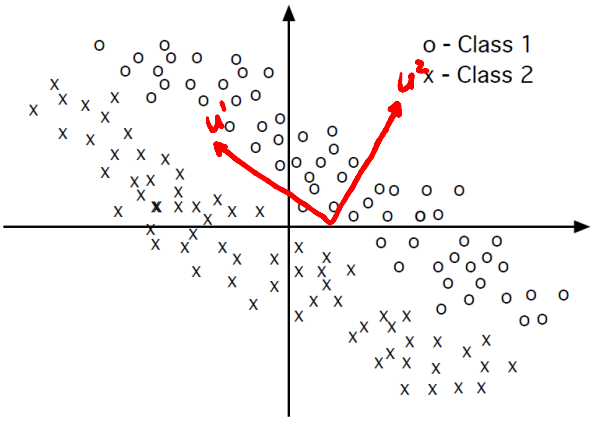
b) Name one objective function which could be minimized to learn the parameters of PCA.

Answer: Reconstruction error. Another objective is maximizing the total variance of projected points.

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

Answer: First, normalize the points to have zero mean and unit standard deviation in each coordinate. Then, compute the covariance matrix of the data, and decompose it with Singular Value Decomposition to obtain its eigenvectors/values. Finally, project each point onto the top k eigenvectors to obtain the lower-dimensional points. The eigenvalues can be used to determine how many components to keep in order to preserve a certain percentage of the total variance.

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.



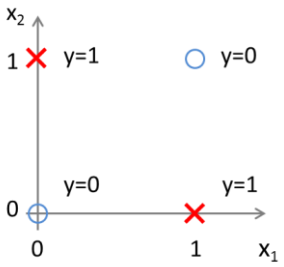
e) In (d) above, could you use PCA components directly to classify the data (without training a classifier)? Explain.

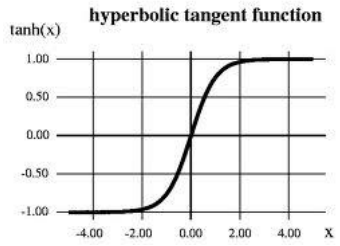
Answer: yes, we can project the points onto the second eigenvector, u^2, and then threshold the one-dimensional points at 0 to assign the class label. Specifically, the classifier will assign labels $sign((u^2)^T x)$ where +1 corresponds to class 1.

9. Neural Networks

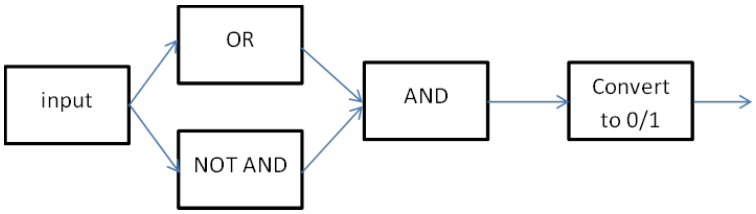
Q9.1 Neural Network for XOR

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].



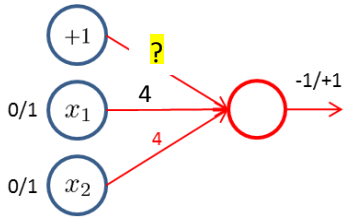


Note that (A XOR B) can be expressed as (A OR B) AND NOT(A AND B)), as illustrated below:

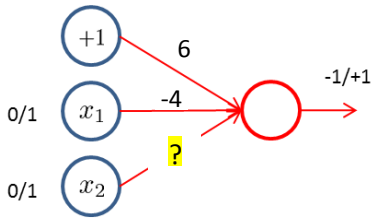


In the diagrams below, we filled in most of the tanh units' parameters. Fill in the remaining parameters, keeping in mind that tanh outputs +1/-1, not 0/1. Note that we need to appropriately change the second layer (the AND node) to take +1/-1 as inputs. Also, we must add an extra last layer to convert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

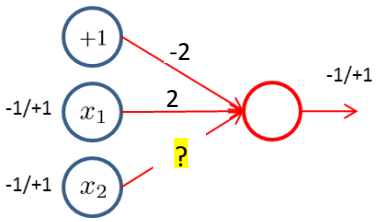
x_1 OR x_2



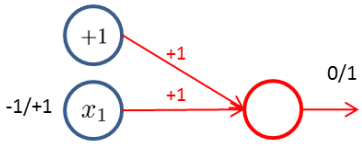
NOT(x_1 AND x_2)



x_1 AND x_2



convert to 0/1



Answer: -2, -4, +2

Q9.2 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.

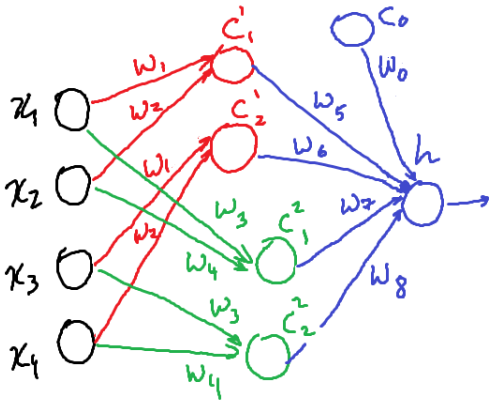
Answer: see lecture notes for examples.

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

Answer: see lecture notes for examples.

Q9.3 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?



Answer: 9 parameters, see w's in the plot. The c's are the hidden convolutional units, with each channel sharing parameters. Note that the last layer should also have a "dummy" unit set to 1 and thus an extra parameter.

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

Answer: Backpropagation in time. The steps for computing the gradient for 1 training sequence are as follows. First, we unroll the network by replicating it once for each time step in our training sequence. Then we use regular backprop on the unrolled network to find the gradient. See lecture notes for more detail.