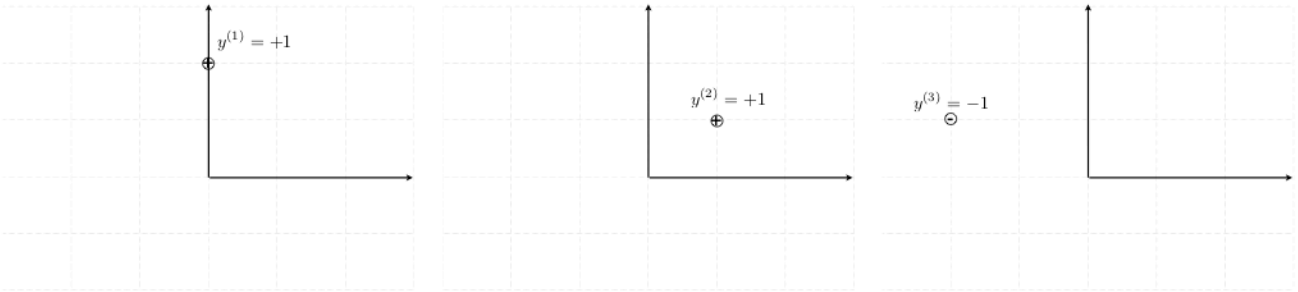
6.036: Midterm Review Problems Fall 2018

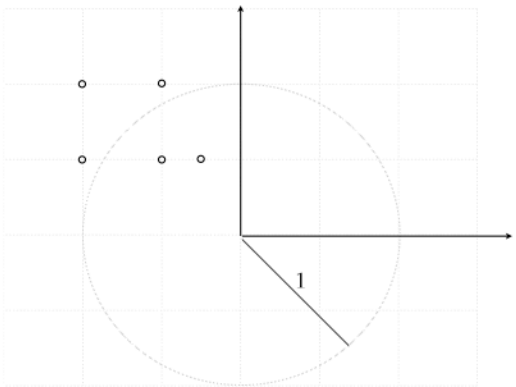
Spring 2013

(1.1) The perceptron algorithm remains surprisingly popular for a 50+ year old (algorithm). Hard to find a simpler algorithm for training linear or non-linear classifiers. Worth knowing this algorithm by heart, yes?

(a) **(6 points)** Let $\theta = 0$ (vector) initially. We run the perceptron algorithm to train $y = \text{sign}(\theta \cdot x)$, where $x \in \mathcal{R}^2$, using the three labeled 2-dimensional points in the figure below. In each figure, approximately draw both θ and the decision boundary *after* updating the parameters (if needed) based on the corresponding point.



(b) **(6 points)** We were wondering what would happen if we normalized all the input examples. In other words, instead of running the algorithm using x, we would run it with feature vectors $\phi(x) = x/\|x\|$. Let's explore this with linear classifiers that include offset, i.e., we use $y = \text{sign}(\theta \cdot x + \theta_0)$ or $y = \text{sign}(\theta \cdot \phi(x) + \theta_0)$ after feature mapping. In the figure below, label *all the points* such that 1) the perceptron algorithm wouldn't converge if they are given as original points, 2) the algorithm would converge if run with $\phi(x) = x/\|x\|$.



Spring 2013

(3.1) We are faced with a content filtering problem where the idea is to rank new songs by trying to predict how they might be rated by a particular user. Each song x is represented by a feature vector $\phi(x)$ whose coordinates capture specific acoustical properties. The ratings are binary valued $y \in \{0, 1\}$ ("need earplugs" or "more like this"). Given n already rated songs, we decided to use regularized linear regression to predict the binary ratings. The training criterion is

$$J(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{n}\sum_{t=1}^{n}(y^{(t)} - \theta \cdot \phi(x^{(t)}))^2/2 \tag{3}$$

(a) **(8 points)** Let $\hat{\theta}$ be the optimal setting of the parameters with respect to the above criterion, which of the following conditions must be true (check all that apply)

( ) $\lambda\,\hat{\theta} - \frac{1}{n}\sum_{t=1}^{n}(y^{(t)} - \hat{\theta} \cdot \phi(x^{(t)}))\phi(x^{(t)}) = 0$

( ) $J(\hat{\theta}) \geq J(\theta)$, for all $\theta \in \mathcal{R}^d$

( ) If we increase λ, the resulting $\|\hat{\theta}\|$ will decrease

( ) If we add features to $\phi(x)$ (whatever they may be), the resulting squared training error will NOT increase

(b) **(3 points)** Once we have the estimated parameters $\hat{\theta}$, we must decide how to predict ratings for new songs. Note that the possible rating values are 0 or 1. When do we choose rating $y = 1$ for a new song x? Please write the corresponding expression.

(c) **(3 points)** If we change λ, we obtain different $\hat{\theta}$, and therefore different rating predictions according to your rule above. What will happen to your predicted ratings when we increase the regularization parameter λ?

Spring 2014

(3.1) We can obtain a non-linear classifier by mapping each example x to a non-linear feature vector $\phi(x)$. Consider a simple classification problem in two dimensions shown in Figure 2. The points $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$ are labeled $y^{(1)} = 1$, $y^{(2)} = 1$, and $y^{(3)} = -1$.

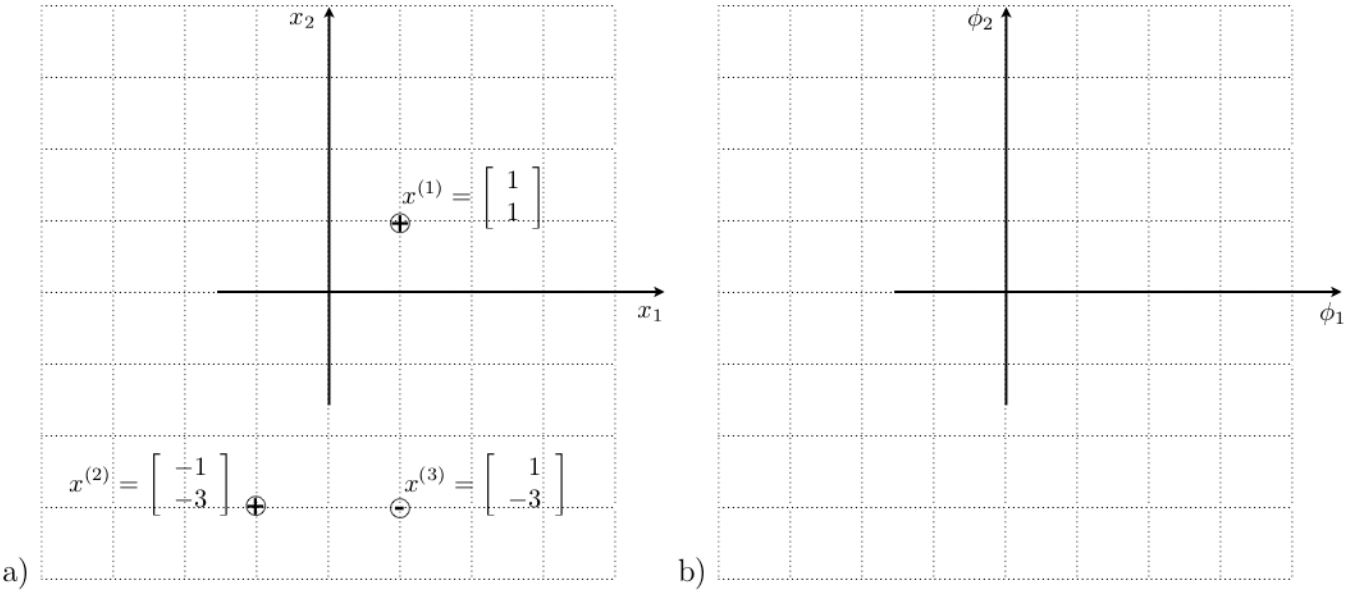


Figure 2: a) A labeled training set with three points. The labels are shown inside the circles. b) Examples in feature coordinates (to be mapped).

(a) **(2 points)** Are the points in Figure 2 linearly separable (Y/N) **(** **)**

(b) **(2 points)** Are the points in Figure 2 linearly separable through origin (Y/N) **(** **)**

(c) **(2 points)** Consider a two dimensional feature mapping $\phi(x) = [x_1, x_2 x_1]^T$ where x_1 and x_2 are the coordinates of x. Map the three training examples in Figure 2a) to their feature coordinates in Figure 2b).

(d) We will solve the classification problem in the feature space. In other words, we will find $\theta = [\theta_1, \theta_2]^T$ that minimizes

$$\frac{1}{2}\|\theta\|^2 \text{ subject to } y^{(i)}\theta \cdot \phi(x^{(i)}) \geq 1, \quad i = 1, 2, 3 \tag{4}$$

We denote the solution by $\hat{\theta}$.

(i) **(6 points)** Draw the resulting $\hat{\theta}$ (orientation is fine), the corresponding decision boundary, and the margin boundaries in the feature coordinates in Figure 2b). Circle the support vectors.

(ii) **(2 points)** What is the value of the resulting margin? **(** **)**

(iii) **(3 points)** What is the value of $\|\hat{\theta}\|$? **( )**

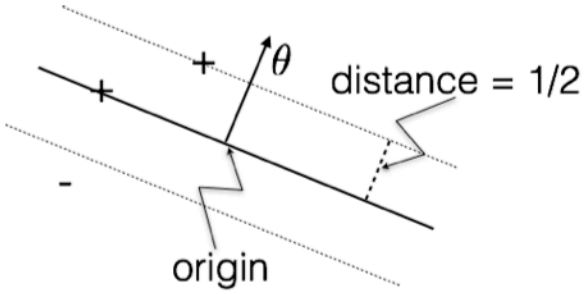
(e) **(4 points)** Draw the resulting decision boundary $\{x : \hat{\theta} \cdot \phi(x) = 0\}$ in the original x-coordinates in Figure 2a). The boundary divides the space in two or more areas. Mark each area based on how the points there would be classified. Show any calculations below.

Spring 2016

Problem 1

(1.1) **(2 points)** Consider a set of linearly separable examples $(x^{(i)}, y^{(i)})$, $i = 1, \ldots, n$, $x^{(i)} \in \mathcal{R}^d$. If we modify the input vectors by eliminating the first coordinate (e.g., setting it to zero for all $x^{(i)}$), are the resulting set of examples guaranteed to be linearly separable? (Y/N)( )

(1.2) **(2 points)** Can the perceptron algorithm be viewed as a stochastic gradient descent algorithm, just applied to minimize the zero one loss? (Y/N) ( )



Decision boundary together with the margin boundaries

(1.3) **(6 points)** SVM is an offline algorithm for estimating a linear separator. If we omit the offset parameter, SVM finds θ that minimizes the objective function

$$\left[\frac{1}{n}\sum_{i=1}^{n} \text{Loss}_h(y^{(i)}\theta \cdot x^{(i)})\right] + \frac{\lambda}{2}\|\theta\|^2 \tag{1}$$

Suppose we set $\lambda = 1$ and $n = 3$ as in the figure above. What is the value of the objective function based on the figure?

Based on the figure, is there a solution which has lower loss and smaller $\|\theta\|$?

(1.4) **(2 points)** Pegasos is an on-line stochastic gradient descent (SGD) method for optimizing the SVM objective. Which one of the following update rules is the correct Pegasos update in response to a training example (x, y)? Mark only the correct one or leave blank if all are incorrect.

$$(\quad) \qquad \hat{\theta} = \theta + \eta(yx - \lambda\theta) \tag{2}$$

$$(\quad) \qquad \hat{\theta} = \theta + \eta yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!] \tag{3}$$

$$(\quad) \qquad \hat{\theta} = \theta + \eta\big(yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!] - \lambda\theta\big) \tag{4}$$

$$(\quad) \qquad \hat{\theta} = \theta + \eta\big(yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!]/n - \lambda\theta\big) \tag{5}$$

Spring 2016

Problem 2 Given training samples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, ridge regression seeks to predict each response $y^{(i)}$ with a linear model $\theta \cdot x^{(i)}$ while encouraging θ to have a small norm. We omit the offset parameter for simplicity. Specifically, θ is estimated by minimizing

$$\left[\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \theta \cdot x^{(i)})^2/2\right] + \frac{\lambda}{2}\|\theta\|^2, \tag{6}$$

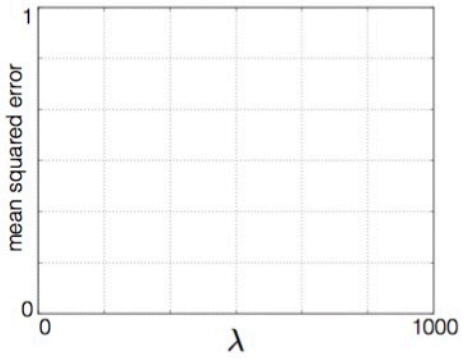
where $\lambda \geq 0$ is the regularization parameter, typically chosen in advance.

(2.1) **(2 points)** What is the solution $\hat{\theta}$ that minimizes Eq(6) if $\lambda \to \infty$?

(2.2) **(2 points)** If we assume that $(1/n)\sum_{i=1}^{n}(y^{(i)})^2/2 = 1$, sketch in the figure how the squared training error

$$\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{\theta}(\lambda) \cdot x^{(i)})^2/2 \tag{7}$$

behaves as we vary λ. Here $\hat{\theta}(\lambda)$ is the solution to Eq(6) with the chosen λ.



Spring 2016

Problem 4

(4.1) **(4 points)** Figure 1 shows the 'two-corners' dataset. It is clear that the two classes (marked with '+' and '▽') are not linearly separable.

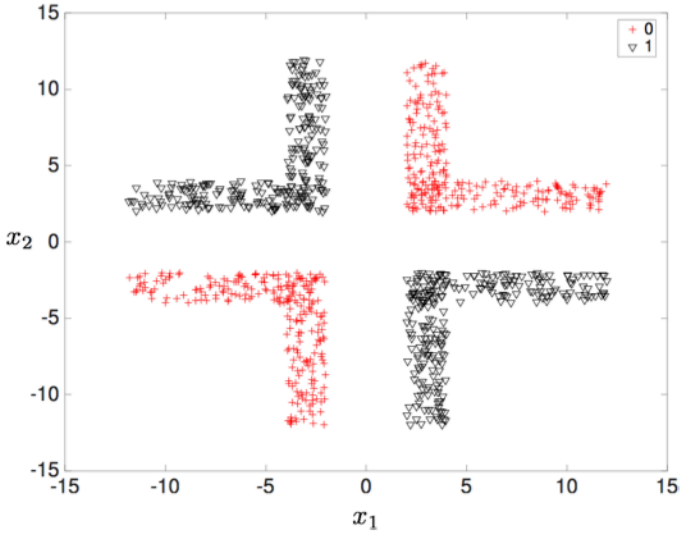


Figure 1: Two-corners dataset

The points labeled '+' live in the first and third quadrants, whereas the '▽' points live in the second and fourth quadrants. Perhaps we can make them separable if we introduce non-linear coordinates. We list a few possible choices below. Mark all the choices that would make this data linearly separable

(a) ( ) $[x_1, x_2, x_1 x_2]$

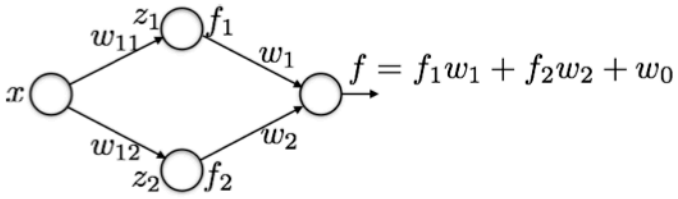
(b) ( ) $[x_1^2, x_2^2, \frac{x_1+x_2}{2}]$

(c) ( ) $[x_1, x_2, \tanh(x_1 + x_2)]$

(d) ( ) $[x_1 + x_2, x_1 x_2, 1]$

Spring 2016

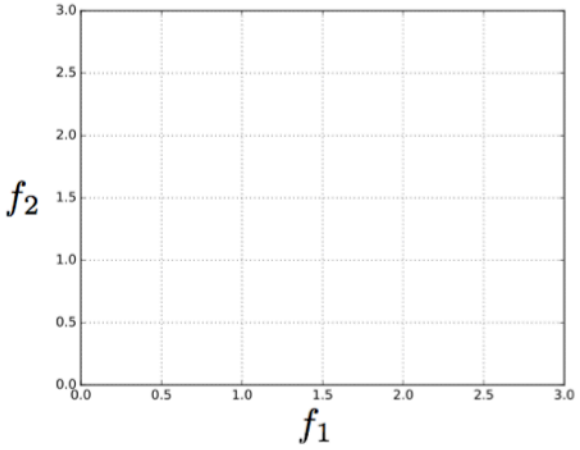
Problem 5 Consider a simple feed-forward neural network model that takes $x \in \mathbb{R}$ as an input and has two ReLU hidden units.



$$\begin{array}{ll} z_1 & = x\, w_{11} + w_{01} \\ z_2 & = x\, w_{12} + w_{02} \end{array}, \quad \begin{bmatrix} w_{11} & w_{01} \\ w_{12} & w_{02} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -2 \end{bmatrix} \tag{10}$$

(5.1) **(2 points)** When is the output of the first hidden unit, i.e., f_1, exactly zero as a function of $x \in \mathbb{R}$?

(5.2) **(6 points)** Given the parameters in Eq(10), sketch in the figure below how examples $x \in [-2, 2]$ map to the 2-dimensional feature coordinates (f_1, f_2). In other words, map the interval $[-2, 2]$ in the x-space to the 2-dimensional space of hidden unit activations.



(5.3) **(2 points)** Are the training examples $(x = -1, y = -1)$, $(x = 1, y = 1)$, and $(x = 2, y = -1)$ linearly separable in the (f_1, f_2) coordinates? (Y/N) ( )

(5.4) **(4 points)** Suppose $w_1 = 1$, $w_2 = 1$, and $w_0 = 0$. We use $\text{Loss}_h(yf) = \max\{0, 1 - yf\}$ to measure the loss given the network output f and true label y. The parameters will potentially change if we perform a gradient descent step in response to (x, y) where $y = -1$. Provide the range of values of x such that w_{02} *decreases* after the update: $x \in$ ( )

(5.5) **(3 points)** The architecture and training details are sometimes quite relevant to how well a particular neural network model works. Suppose we use ReLU activation functions for all the hidden units (one layer). Briefly explain what will happen if we initialized all the weights to zero, and all the offset parameters to -1, and ran the stochastic gradient descent method to learn the parameters

(5.6) **(6 points)** We experimented with three different architecture/training combinations:

( ) m hidden units, no regularization

( ) $2m$ hidden units, no regularization

( ) $2m$ hidden units, dropout regularization

Student who carried out the experiments observed that

Model A: training error 0.2, validation error 0.35

Model B: training error 0.25, validation error 0.3

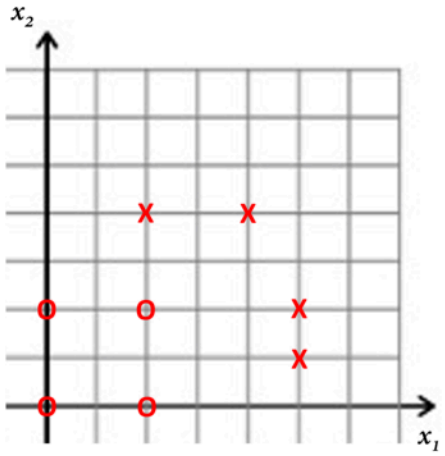
Model C: training error 0.1, validation error 0.4

Please assign the models A,B, and C to their best fitting setups above.

Spring 2017

Problem 1 Consider a labeled training set shown in the table and figure below:

Label	-1	-1	-1	-1	+1	+1	+1	+1
Coordinates	(0,0)	(2,0)	(0,2)	(2,2)	(5,1)	(5,2)	(2,4)	(4,4)
Perceptron mistakes	1	6	1	5	3	1	2	0



(1.1) **(4 points)** We initialize the parameters to all zero values and run the Perceptron algorithm through these points in a particular order until convergence. The number of mistakes made on each point are shown in the table. What is the resulting offset parameter θ_0?

(1.2) **(2 points)** The mistakes that the algorithm makes often depend on the order in which the points were considered. Could the point (4,4) labeled '+1' have been the first one considered? (Y/N) ( )

Suppose that we now find the linear separator that **maximizes** the margin instead of running the perceptron algorithm.

(1.3) **(4 points)** What are the parameters θ and θ_0 corresponding to the maximum margin separator?

(1.4) **(2 points)** What is the value of the margin attained?

(1.5) **(2 points)** What is the sum of Hinge losses evaluated on each example?

(1.6) **(3 points)** Suppose we modify the maximum margin solution a bit and divide both θ and θ_0 by 2. What is the sum of Hinge losses evaluated on each example for this new separator?

Spring 2017

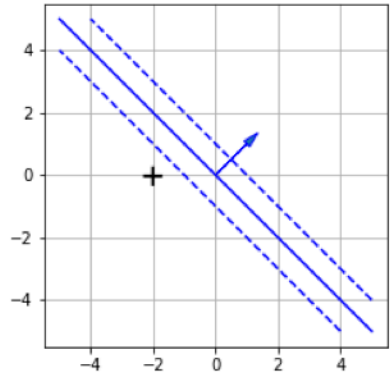
Problem 2 Stochastic gradient descent (SGD) is a simple but widely applicable optimization technique. For example, we can use it to train a Support Vector Machine. The objective function in this case is given by (here without offset)

$$J(\theta) = \left[\frac{1}{n}\sum_{i=1}^{n} \text{Loss}_h(y^{(i)}\theta \cdot x^{(i)})\right] + \frac{\lambda}{2}\|\theta\|^2 \tag{1}$$

where $\text{Loss}_h(z) = \max\{0, 1 - z\}$ is the Hinge loss.

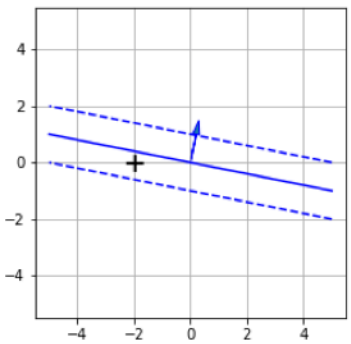
(2.1) **(4 points)** Let θ be the current parameters. Provide an expression for the stochastic gradient update. Your expression should involve only θ, points, labels, λ, and the learning rate η.

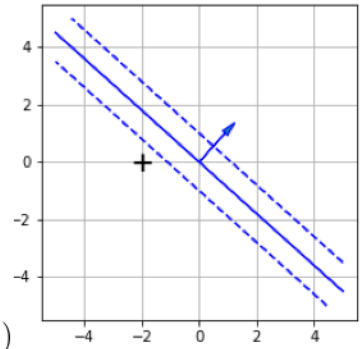
(2.2) **(6 points)** Consider the situation shown in this figure:

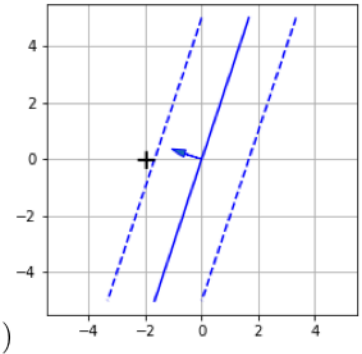


Associate the figures (A,B,C) below to a single SGD update made in response to the point labeled '+1' when we use

( ) small λ, small η, ( ) small λ, large η, ( ) large λ, large η.

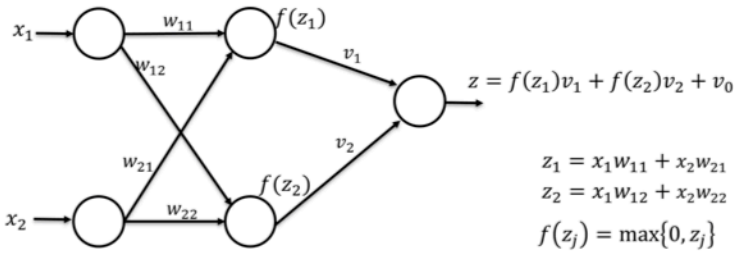
A) 

B) 

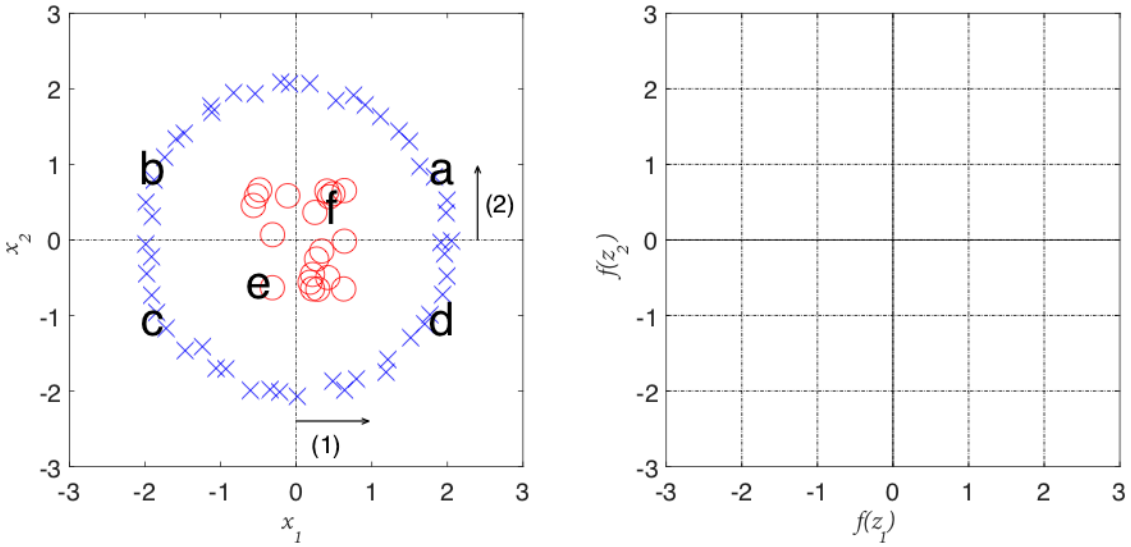
C) 

Spring 2017

Problem 4 Consider a 2-layer feed-forward neural network that takes in $x \in \mathbb{R}^2$ and has two ReLU hidden units. *Note that the hidden units have no offset parameters.*



(4.1) **(6 points)** The values of the weights in the hidden layer are set such that they result in the z_1 and z_2 "classifiers" as shown in the figure by the decision boundaries and the corresponding normal vectors marked as (1) and (2). Approximately sketch on the right how the input data is mapped to the 2-dimensional space of hidden unit activations $f(z_1)$ and $f(z_2)$. Only map points marked 'a' through 'f'. Keep the letter indicators.



(4.2) **(2 points)** If we keep these hidden layer parameters fixed but add and train additional hidden layers (applied after this layer) to further transform the data, could the resulting neural network solve this classification problem? (Y/N) ( )

(4.3) **(3 points)** Suppose we stick to the 2-layer architecture but add lots more ReLU hidden units, all of them without offset parameters. Would it be possible to train such a model to perfectly separate these points? (Y/N) ( )

(4.4) **(3 points)** Initialization of the parameters is often important when training large feed-forward neural networks. Which of the following statements is correct? Check T or F for each statement.

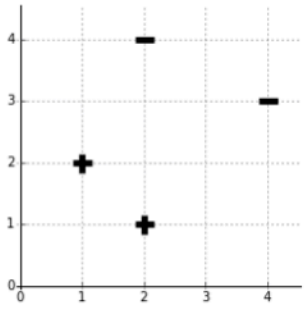
( ) If we use tanh or linear units and initialize all the weights to very small values, then the network initially behaves as if it were just a linear classifier

( ) If we randomly set all the weights to very large values, or don't scale them properly with the number of units in the layer below, then the tanh units would behave like sign units.

( ) A network with sign units cannot be effectively trained with stochastic gradient descent

Fall 2017

1. (8 points) Consider a dataset of labeled points below:



(a) Please select one of the following alternatives.

- ◯ The points are linearly separable through origin
- ◯ Linearly separable but not through the origin
- ◯ Only non-linearly separable

(b) Specify the coefficients of the following **decision boundary** that would suffice for separating the points in the figure above.

_____ x_1+ _____ x_2+ _____ x_1^2+ _____ x_2^2+ _____ $= 0$

(c) Consider a data set that is **not** linearly separable. Lenny Arsep suggests that applying a feature mapping $\phi(x_1, x_2) = [2x_1, x_1 + x_2]^T$ would make it linearly separately in the new coordinates. Is Lenny correct?

- ◯ True for all data sets
- ◯ True for at least one data set but not all
- ◯ False for all data sets

Fall 2017

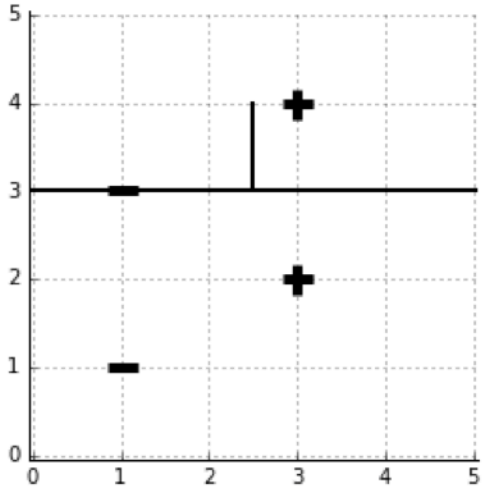
4. (12 points) Recall the definitions of hinge loss

$$L_h(z) = \begin{cases} 1 - z & \text{if } z < 1 \\ 0 & \text{otherwise} \end{cases}$$

and the SVM objective function:

$$J(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} L_h(y^{(i)}(\theta^T x^{(i)} + \theta_0)) + \lambda \|\theta\|^2$$

Use this data throughout the question:



(a) Let $\theta = (0, 1)$ and $\theta_0 = -3$.

i. Do these weights correspond to the separator in the figure? ◯ Yes ◯ No

ii. What is the hinge loss, $L_h(y^{(i)}(\theta^T x^{(i)} + \theta_0))$, for each point?

(1, 1) : _____ (1, 3) : _____ (3, 2) : _____ (3, 4) : _____

iii. If $\lambda = 1$, what is J?

Consider the following separators (note the orientation!):

1. $\theta^{(1)} = (0, 1)$ and $\theta_0 = -3$.
2. $\theta^{(2)} = (0.1, 0)$ and $\theta_0^{(2)} = -0.2$
3. $\theta^{(3)} = (100, 0)$ and $\theta_0^{(3)} = -200$

(b) What is the margin of each separator given the above training set? Write N/A if the value cannot be calculated.

(1) : _____ (2) : _____ (3) : _____

(c) Which of the parameter settings minimizes the value of the objective function for $\lambda = 0$

◯ (1) ◯ (2) ◯ (3)

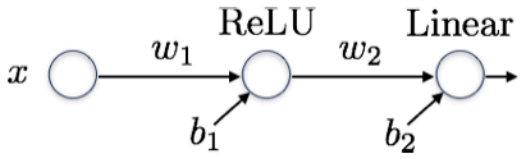
(d) Would setting $\lambda = 100$ change your answer to (c)? ◯ Yes ◯ No

Fall 2017

5. (18 points) Here is a very simple neural network with one hidden unit (ReLU) and one linear output unit. The weights and biases for this network are given by

$$w_1 = -2, b_1 = 3, w_2 = 3, b_2 = 0$$

Let f_1 and f_2 denote the activations of the hidden and output unit, respectively, and $z_1 = w_1 x + b_1$ and $z_2 = w_2 f_1 + b_2$ the pre-activation inputs to these units.



(a) What is the numerical value of the output unit, i.e., the value of f_2, when presented with an input $x = 1$?

(b) We assume that the loss is simply the squared loss, i.e., $\text{Loss}(f_2, y) = (f_2 - y)^2/2$. Give an expression for the gradient of the loss with respect to z_2, the input to the output unit (your final answer should no longer include derivative signs, and should be in terms of x, y, w_1, w_2, b_1, and b_2.)

(c) Under the same assumption of squared loss, give an expression for the gradient of the loss with respect to the input to the hidden unit, i.e., with respect to z_1 (your final answer should no longer include derivative signs, and should be in terms of x, y, w_1, w_2, b_1, and b_2.)

(d) Let input $x = 1$ and target output $y = 1$. We continue to use the squared loss $\text{Loss}(f_2, y) = (f_2 - y)^2/2$ as discussed above. Provide the numerical value of w_1 after a single SGD update with learning rate 0.5, under the assumption that (as for part (a)), $w_1 = -2, b_1 = 3, w_2 = 3, b_2 = 0$.

(e) Under what conditions will w_1 be unchanged by a back-propagation update step given training example (x, y)? (Answer generically, in terms of x, y, w_1, w_2, b_1, and b_2.)

Fall 2017

6. (10 points) A concert venue is trying to predict how many people will attend an event based on features such as the date of the concert, the genre of the band, etc. They will use this model to decide which bands to book, but they would like the predictions to be conservative: if their model under-estimates the attendance it is better than if it overestimates. We can express this preference in a loss function, where g is the predicted ("guessed") value and y is the true target value:

$$L(g, y) = \begin{cases} c_1(g-y)^2 & \text{if } g > y \\ c_2(g-y)^2 & \text{otherwise} \end{cases}$$

(a) For what values of c_1 and c_2 is this loss function equivalent to squared error?

(c) Consider a linear model in which $g = \theta^T x + \theta_0$. Derive a stochastic gradient descent update rule for θ and θ_0, with step size η. (If you are having trouble thinking about θ and x as vectors, start by figuring it out for the scalar case.)

Fall 2017

7. (10 points) We work for the investment firm Golden Sacks, and we are trying to build several predictive models about the stocks of companies.

   (a) Companies are described to us in terms of 4 features. For each feature, describe a transformation that one should apply to convert it to features that could be concatenated to make a feature vector representing the company, or indicate that it should be omitted from the input representation.

      i. Market segment (one of "service", "natural resources", or "technology")

      ii. Number of countries in which it operates (1 – 50)

      iii. Total valuation (-1 billion to + 1 billion)

      iv. Company name

(b) Now, consider output representation. For each goal below, specify the number of output units you would use, as well as the activation and loss function.

i. Your goal is to predict whether or not the company will have an IPO in the next year.

α) Number of units _____

β) Activation function

γ) Loss function

ii. Your goal is to predict the net profit of the company next year.

α) Number of units _____

β) Activation function

γ) Loss function

iii. You have 100 favorite clients each of whom has certain preferences for the types of stocks they buy. You have to predict, for a given new stock x, which clients will like it and which will not.

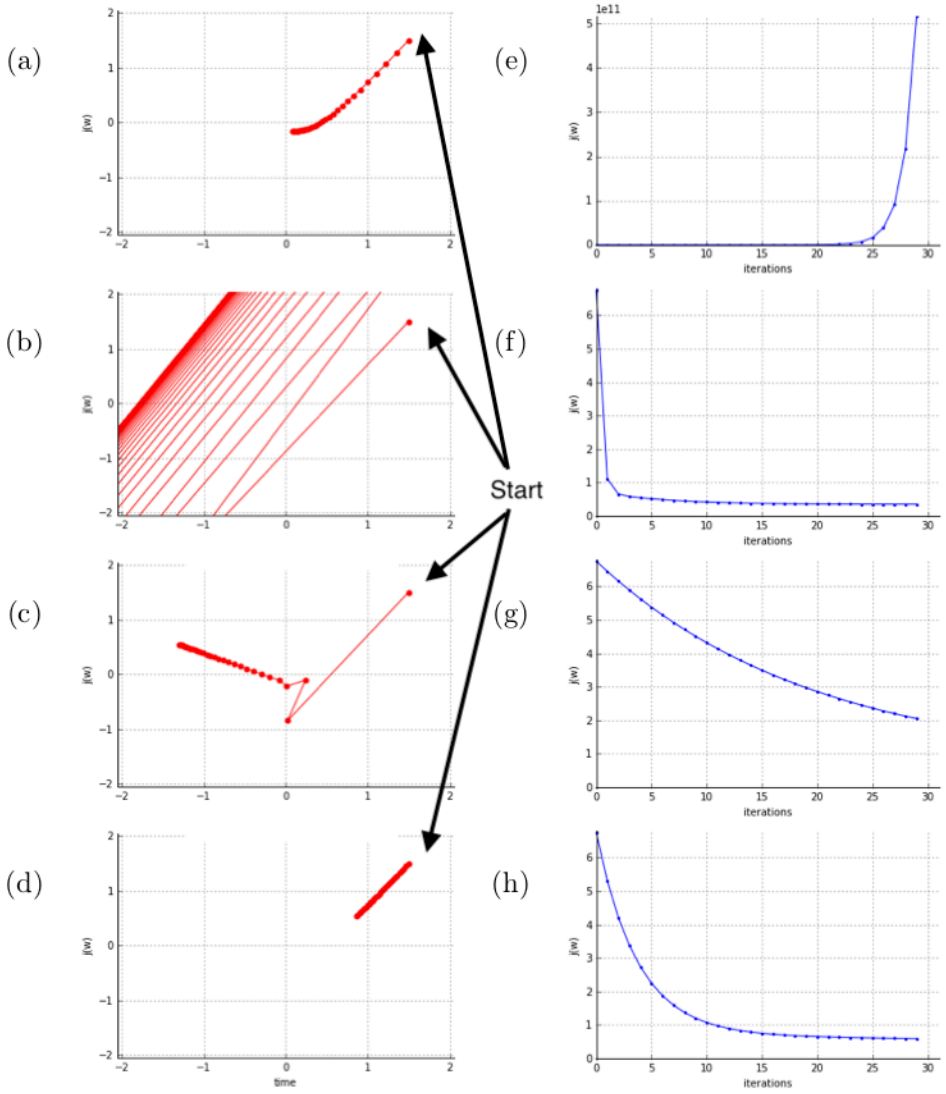
α) Number of units _____

β) Activation function

γ) Loss function

Fall 2017

9. (8 points) The following plots show the results of performing linear regression using gradient descent with fixed step size. The first column shows the trajectory of the weight vector as a function of iteration number; the second shows objective as a function of iteration number.



***Note that the weight vector is two dimensional. The column on the left shows that the starting value for all four plots is the same, w = [1.5, 1.5].**

For each step size, specify the corresponding weight trajectory and objective plot.

(a) 0.01: Weight trajectory _____ Objective _____

(b) 0.05: Weight trajectory _____ Objective _____

(c) 0.50: Weight trajectory _____ Objective _____

(d) 1.00: Weight trajectory _____ Objective _____