6.867: Exercises (Week 5)

Sept 29, 2016

Contents

1 Colonel Vector

Consider the kernel

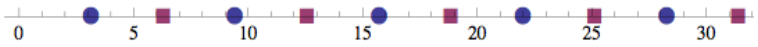
$$K(\underline{x}, \underline{z}) = \underline{x} \cdot \underline{z} + 4\,(\underline{x} \cdot \underline{z})^2$$

where the vectors $\underline{x}$ and $\underline{z}$ are 2-dimensional. This kernel is equal to an inner product $\phi(\underline{x}) \cdot \phi(\underline{z})$ for some definition of ϕ. What is the function ϕ?

2 Kernels of truth

Consider the data set, where the input is a one-dimensional real:

$$\{(\pi, -1), (2\pi, +1), (3\pi, -1), (4\pi, +1), (5\pi, -1), (6\pi, +1), (7\pi, -1), (8\pi, +1), (9\pi, -1), (10\pi, +1)\} \ .$$



For each of the kernels, indicate whether it can separate the data exactly.

1. ___ $K(x, y) = (xy)^2$
2. ___ $K(x, y) = (xy)^{20}$
3. ___ $K(x, y) = (xy + 1)^2$
4. ___ $K(x, y) = (xy + 100)^{20}$
5. ___ $K(x, y) = e^{-10(x-y)^2}$
6. ___ $K(x, y) = e^{-0.1(x-y)^2}$
7. ___ $K(x, y) = \cos(x)\cos(y)$
8. ___ $K(x, y) = \sin(x)\sin(y)$

3 SVM with 3 points

Consider a simple classification problem (of the kind that you could only encounter in an exam). The training data consist of only three labeled points

$$(x_1 = -1, y_1 = +1), (x_2 = 0, y_2 = -1), (x_3 = +1, y_3 = +1)$$

which we will try to separate with a linear classifier through origin in the feature space. In other words, our discriminant function is of the form $\underline{\theta} \cdot \underline{\phi}(x)$. The corresponding primal and dual

estimation problems are given by

$$\textbf{Primal:}\text{Minimize} \quad \frac{1}{2}\|\underline{\theta}\|^2 + C\sum_{i=1}^{3}\xi_i \tag{3.1}$$

$$\text{subject to} \quad y_i(\underline{\theta}\cdot\underline{\phi}(x_i)) \geqslant 1-\xi_i, \quad \xi_i \geqslant 0, \quad i=1,2,3 \tag{3.2}$$

$$\tag{3.3}$$

$$\textbf{Dual:}\text{Maximize} \quad \sum_{i=1}^{3}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{3}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{3.4}$$

$$\text{subject to} \quad 0 \leqslant \alpha_i \leqslant C, \quad i=1,2,3 \tag{3.5}$$

(a) We decided to solve the problem in the dual using kernel $K(x,x') = 1+|xx'|$ where $|\cdot|$ is the absolute value. What is the feature mapping $\underline{\phi}(x)$ corresponding to this kernel?

(b) Using this kernel (feature mapping), are the three training examples linearly separable through origin in the feature space?

(c) If we decrease the slack penalty C, the solution might not satisfy the margin constraint for $(x_2=0, y_2=-1)$ without a positive slack $\xi_2 > 0$. What does this mean in terms of α_2?

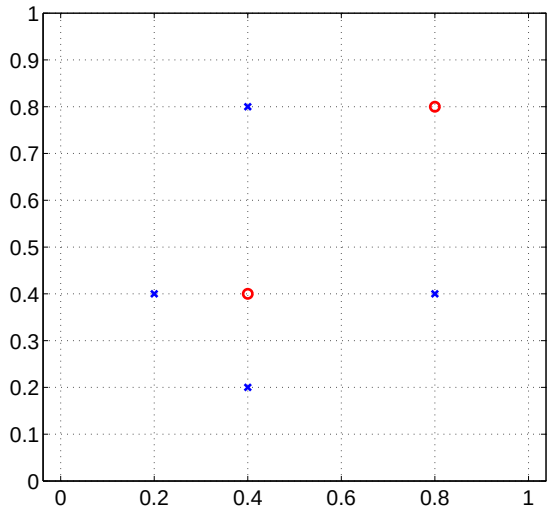
(d) Assume $K(x,x') = 1+|xx'|$ and the three point training set. Express the value of the discriminant function in the dual form for $x_2 = 0$. If we set $C < 1$, do we necessarily get a positive slack ($\xi_2 > 0$) for this example $(x_2=0, y_2=-1)$? Briefly justify your answer.

4 Slacking off

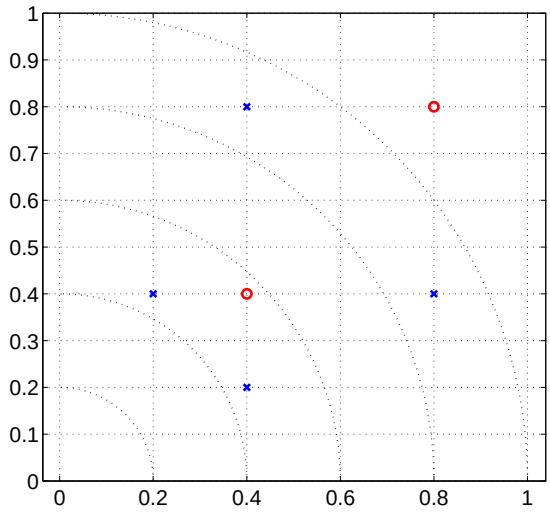
Consider training an SVM with slack variables, but with no bias variable. The kernel used is $K(\underline{x},\underline{z})$; it has the property that for any two points $\underline{x}_i$ and $\underline{x}_j$ in the training set, $-1 < K(\underline{x}_i,\underline{x}_j) < 1$. $K(\underline{x}_i,\underline{x}_i) < 1$ as well. There are n points in the training set. Show that if the slack-variable constant C is chosen such that $C < \frac{1}{n-1}$, then all dual variables α_i are non-zero (i.e., all points in the training set become support vectors).

5 Almost a linear kernel

1)



2)



1) Points that should be separable with a normalized linear kernel. 2) feature space with the original points overlaid with their original coordinate values.

A student in a machine learning course claimed that the points in part 1) above can be separated with "almost a linear kernel". Hard to believe, we responded, since the points are clearly not linearly separable. But the student insisted. The "almost a linear kernel" they had in mind was the following normalized kernel:

$$K_{norm}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|}$$

(a) What are the feature vectors corresponding to this kernel?

(b) Using figure 2 (right), graphically map the points to their new feature representation using the figure as the feature space.

(c) Draw the resulting maximum margin decision boundary in the feature space. Use the same figure 2 (right). The student was right, the points are separable!

(d) Does the value of the discriminant function corresponding to your solution change if we scale any point, i.e., evaluate it at $s\,\mathbf{x}$ instead of $\mathbf{x}$ for some $s > 0$? **(Y/N)**

(e) Draw the decision boundary in the original input space resulting from the normalized linear kernel. Use Figure 1 (left).

6 Kernel decision boundaries

The figure belowplots SVM decision boundaries resulting from using different kernels and/or different slack penalties. The methods used to generate the plots are listed below but (the absent minded) professor forgot to label them. Please assign the plots to the right method. Oh, we also forgot to list one of the methods.

Primal Method with slack penalties:

$$\min \frac{1}{2}\|\underline{\theta}\|^2 + C\sum_{t=1}^{n}\xi_t \text{ s.t. } \xi_t \geqslant 0,\ \ y_t(\underline{\theta}^T\underline{x}_t + \theta_0) - 1 + \xi_t \geqslant 0,\ \ t = 1,\ldots,n$$

Dual Method:

$$\max \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\underline{x}_i,\underline{x}_j)\alpha_i \geqslant 0,\ i = 1,\ldots,n,\ \ \sum_{i=1}^{n}\alpha_i y_i = 0$$

(a) Primal method where $C = 0.1$.

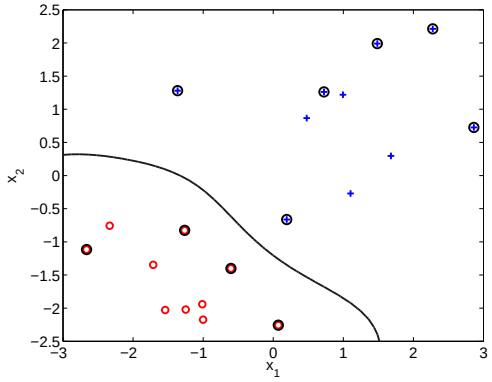
(b) Primal method where $C = 1$.

(c) Dual method where $K(\underline{x},\underline{x}') = \underline{x}^T\underline{x}' + (\underline{x}^T\underline{x}')^2$.

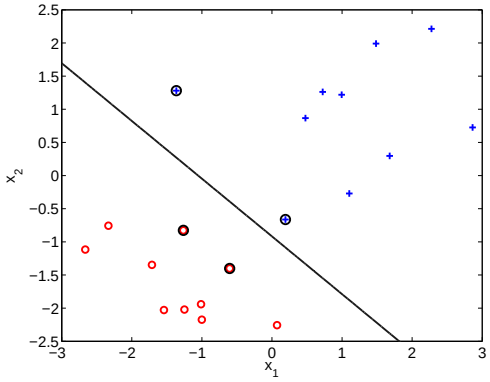
(d) Dual method where $K(\underline{x},\underline{x}') = \exp(-1/2\|\underline{x} - \underline{x}'\|^2)$.

(e) Dual method where $K(\underline{x},\underline{x}') = \exp(-\|\underline{x} - \underline{x}'\|^2)$.

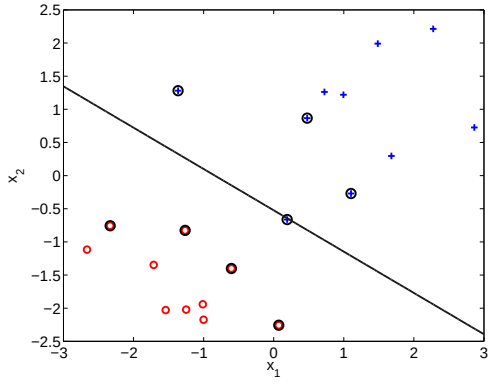
1)



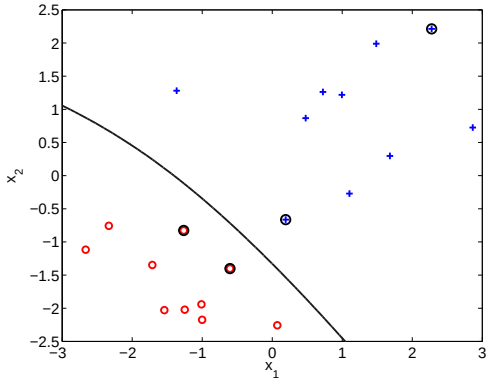
2)

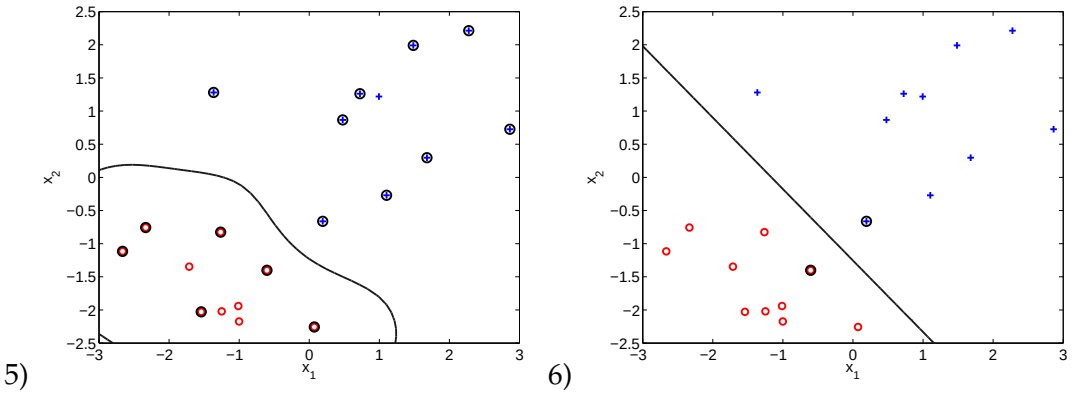


3)



4)





(f) Consider the linear SVM with slack penalties (primal method with slack penalties above):

Indicate which of the following statements hold as we *increase* the parameter C from any starting value. Use 'Y' for statements that *will necessarily hold*, 'N' if the statement is *never true*, and 'D' if the validity of the statement depends on the situation when C increases.

( ) θ_0 will not increase

( ) $\|\underline{\hat{\theta}}\|$ increases

( ) $\|\underline{\hat{\theta}}\|$ will not decrease

( ) more points will be misclassified

( ) the geometric margin for the problem will not increase

7 1D Classification

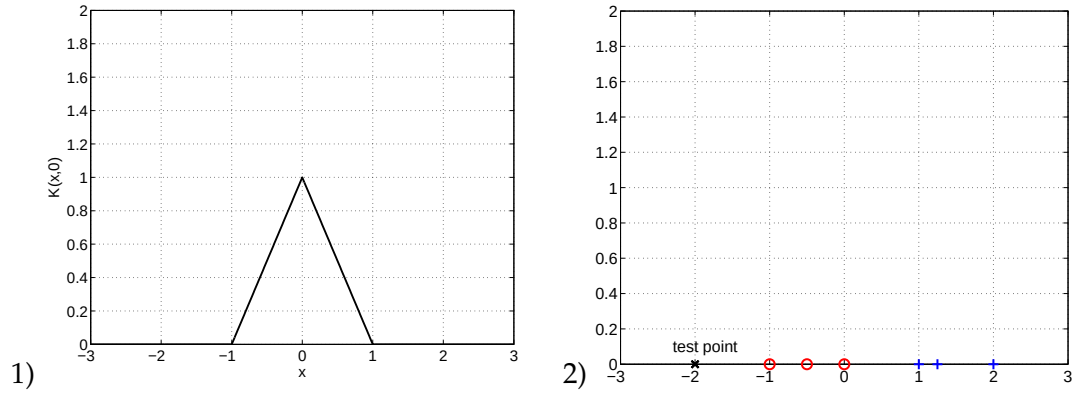


Figure 3. 1) Kernel $K(x, 0)$ for problem 3. 2) data for problems 3.b and 3.c.

Consider solving a 1-dimensional classification problem with SVMs and the kernel

$$K(x, x') = (1 - |x - x'|)^+ = \max\{0,\ 1 - |x - x'|\}$$

Figure 3.1) illustrates this kernel $K(x, 0)$ as a function of x. The feature "vectors" corresponding to this kernel are actually functions $\phi(x)(i)$ such that

$$K(x, x') = \int_{-\infty}^{\infty} \phi(x)(i)\phi(x')(i)di$$

(a) What is the form of $\phi(x)(i)$?

(b) What is the dual objective function for training SVMs (no slack) when we do not include the offset term θ_0 in the classifier? We maximize

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to ?

(c) What is the value of the discriminant function $\sum_{i \in n} \hat{\alpha}_i y_i K(x, x_i)$ on the test point in Figure 3.2)? Assume that $\hat{\alpha}_i$ are estimated on the basis of the training data in the figure without an offset parameter.

(d) Would the test point in Figure 3.2) become a support vector if it were included in the training set?

(e) We can improve the kernel function a bit by introducing a width parameter σ such that

$$K(x, x') = (1 - |x - x'|/\sigma)^+$$

What would be a reasonable method for choosing σ?

(f) Would your method solve the problem identified in 3.c? Briefly explain why or why not.

(g) It is sometimes useful to incorporate test inputs (if available) in some manner in training the classifier. How could you include the test points in selecting the kernel width parameter σ?

8 Radial basis kernel

This is a difficult question but it gets at an interesting and important point.

We can write the radial basis kernel in the following form:

$$K(x, x') = \exp\{-\frac{1}{2\sigma^2}\|x - x'\|^2\},$$

where σ is a width parameter specifying how quickly the kernel vanishes as the points move further away from each other. This kernel has some remarkable properties. Indeed, we can perfectly separate *any* finite set of *distinct* training points. Moreover, this result holds for any positive finite value of σ. While the kernel width does not affect whether we'll be able to perfectly separate the training points, it does affect generalization performance. We will try to understand both of these issues a bit better.

Let's proceed in stages. To make things easier we are going to prove a bit stronger result than we need to. In particular, we'll show that

$$\text{minimize } \frac{1}{2}\|\theta\|^2 \text{ subject to } y^i \theta \cdot \phi(x^i) = 1, \; i = 1, \ldots, n$$

has a solution regardless of how we set the ± 1 training labels y^i. You should convince yourself first that this is consistent with our goal. Here $\phi(x^i)$ is the feature vector (function actually) corresponding to the radial basis kernel. Our formulation here is a bit non-standard for two reasons. We try to find a solution where *all* the points are support vectors. This is not possible for all valid kernels but makes it easier to prove the result. We also omit the bias term since it is not needed for the result.

1. Introduce Lagrange multipliers for the constraints similarly to finding the SVM solution (see also the tutorial on Lagrange multipliers that has been posted) and show the form that the solution $\hat{\theta}$ has to take. You can assume that θ and $\phi(x^i)$ are finite vectors for the purposes of these calculations. Note that the Lagrange multipliers here are no longer constrained to be positive. Since you are trying to satisfy equality constraints, the Lagrange multipliers can take any real value.

   We are after $\hat{\theta}$ as a function of the Lagrange multipliers. (this should not involve lengthy calculations).

2. Put the resulting solution back into the classification (margin) constraints and express the result in terms of a linear combination of the radial basis kernels.

3. Indicate briefly how we can use the following Michelli theorem to show that any n by n RBF kernel matrix $K_{ij} = K(x^i, x^j)$ for $i, j = 1, \ldots, n$ is invertible.

   Theorem: *If $\rho(t)$ is monotonic function in $t \in [0, \infty)$, then the matrix $\rho_{ij} = \rho(\|x^i - x^j\|)$ is invertible for any distinct set of points x^i, $i = 1, \ldots, n$.*

4. Based on the above results put together the argument to show that we can indeed find a solution where all the points are support vectors.

5. Of course, the fact that we can in principle separate any set of training examples does not mean that our classifier does well (on the contrary). So, why do we use the radial basis kernel? The reason has to do with margin that we can attain by varying σ. Note that the effect of varying σ on the margin is not simple rescaling of the feature vectors. Indeed, for the radial basis kernel we have

   $$\phi(x) \cdot \phi(x) = K(x, x) = 1$$

   Let's begin by setting σ to a very small positive value. What is the margin that we attain in response to any n distinct training points?

6. Provide a 1-dimensional example to show how the margin can be larger than the answer to part 5. You are free to set σ and the points so as to highlight how they might "contribute to each other's margin".

9 Using only positive training examples

One evening we thought we had come up with a great machine learning approach to predicting movie ratings. The idea was to base the predictions solely on positive training examples, movies we already know we like ($y = +1$), and simply ignore (as far as the training is concerned) all

the negative examples ($y = -1$). Assume movies are represented by vectors $\underline{x}_1, \ldots, \underline{x}_m$, where $\underline{x}_j \in \mathcal{R}^d$. We created these vectors from movie descriptions (automatically, of course).

Our primal SVM optimization problem, written only for positive examples without offset, is given by

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 \ \text{ subject to } \underline{\theta} \cdot \underline{x}_j \geqslant 1, \ \ j \in J_+ \tag{9.1}$$

where $J_+ \subset \{1, \ldots, m\}$ indexes our positive training examples (movies we already know we like).

1. What would the solution $\hat{\underline{\theta}}$ be if we included an offset parameter θ_0, i.e., changed the constraints to be $\underline{\theta} \cdot \underline{x}_j + \theta_0 \geqslant 1$?

2. Assume we can find the solution $\hat{\underline{\theta}}$ to the problem described in Eq.(9.1). What is the value of $\min_{j \in J_+}(\hat{\underline{\theta}} \cdot \underline{x}_j)$?

3. Suppose again that the solution $\hat{\underline{\theta}}$ to Eq.(9.1) exists. Based on this $\hat{\underline{\theta}}$, we predict labels for movies $\underline{x}$ (new and training examples) according to

$$\hat{y} = \begin{cases} 1, & \text{if } (\hat{\underline{\theta}} \cdot \underline{x}) \geqslant \min_{j \in J_+}(\hat{\underline{\theta}} \cdot \underline{x}_j) - \epsilon \\ -1, & \text{otherwise} \end{cases}$$

   for some small $\epsilon > 0$. Would this decision rule ensure that all the training movies, positive and negative, are classified correctly? Briefly justify your answer.

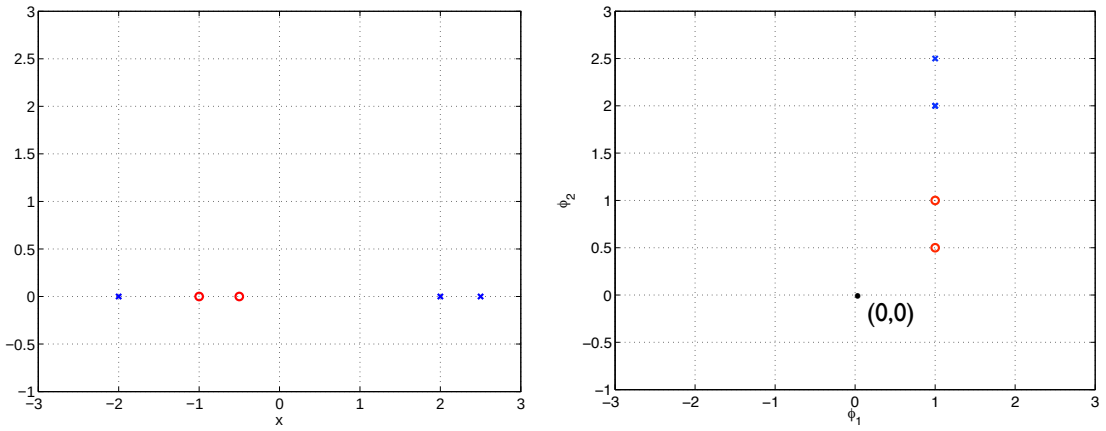


Figure 1: Movie data. Original space (left). Feature space (right).

4. The problem might sometimes get a little challenging. Figure 1 (left) shows the movie data, positive (small blue 'x') and negative (red 'o') examples, when movies are represented by real numbers x_j. Briefly describe why we cannot solve Eq.(9.1) in this case.

5. We will apply the algorithm described in Eq.(9.1) with a feature mapping, i.e., we replace one dimensional x with $\underline{\phi}(x) = [1, |x|]^T$. In Figure 1 (right), we have plotted the movie data, positive ('x') and negative ('o'), in the feature coordinates ϕ_1 and ϕ_2. Sketch the solution $\hat{\underline{\theta}}$ in the feature space by drawing $\hat{\underline{\theta}} \cdot \underline{\phi} = 0$ and $\hat{\underline{\theta}} \cdot \underline{\phi} - 1 = 0$ in the figure on the right.

6. Is $\hat{\underline{\theta}} \cdot \underline{\phi}(x) > 0$ at $x = -1$?

10 String theory

We are interested in doing regression, in which the input to our regressor will be strings of arbitrary length, and the output will be a real number. We plan to apply the extension of ordinary least-squares regression to use a kernel function.

Pat claims the following function is a kernel:

$$K(x, z) = \sum_{\beta \in \text{alphabet}} (\text{occurrences of } \beta \text{ in } x)(\text{occurrences of } \beta \text{ in } z)$$

where the alphabet is the set of Roman characters 'a' through 'z'. We will perform a kernelized regression, finding parameters α_i, so that the predictions are of the form:

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x^{(i)}, x).$$

Answer the following questions assuming the training data is:

x	y
"abalone"	10
"xyzygy"	1
"zigzag"	3

(a) What is the feature vector associated with Pat's kernel? Be sure to specify the dimension, d, of the vector.

(b) Determine an expression for $y(\text{"ziggy"})$ in terms of the α_i parameters.

(c) The vector α can be determined by solving a system of equations of the form $A\alpha = b$. Give the numerical values for matrix A and vector b.

11 Silly friends

(a) Pat sees your neural network implementation with sigmoidal activation and says there's a much simpler way! Just leave out sigmoids, and let $g(a) = a$. The derivatives are a lot less hassle and it runs faster.

What's wrong with Pat's approach?

(b) Chris comes in and says that your network is too complicated, but for a different reason. The sigmoids are confusing and basically the same answer would be computed if we used step functions instead.

What's wrong with Chris's approach?

(c) Jody sees that you are handling a multi-class problem with 4 classes by using 4 output values, where each target $y^{(i)}$ is actually a length-4 vector with three 0 values and a single 1 value, indicating which class the associated $x^{(i)}$ belongs to.

Jody suggests that you just encode the target $y^{(i)}$ values using integers 1, 2, 3, and 4.

What's wrong with Jody's approach?

12 Yes vs No

In two-class classification with a standard sigmoid logistic function, the negative log-likelihood error function is the cross entropy:

$$E(w) = -\sum_{n=1}^{N} (y^{(n)} \log h(x^{(n)}, w) + (1 - y^{(n)}) \log(1 - h(x^{(n)}, w)) \quad .$$

(a) Assuming input x and weight w are scalar, and that $N = 1$ (there is a single training example), what is $\partial E(w)/\partial w$?

(b) What would the neural network weight-update rule be?

13 Grady Ent

Grady Ent decides to train a single sigmoid unit using the following error function:

$$E(w) = 1/2 \sum_i (y(x^i, w) - y^{i*})^2 + 1/2\beta \sum_j w_j^2$$

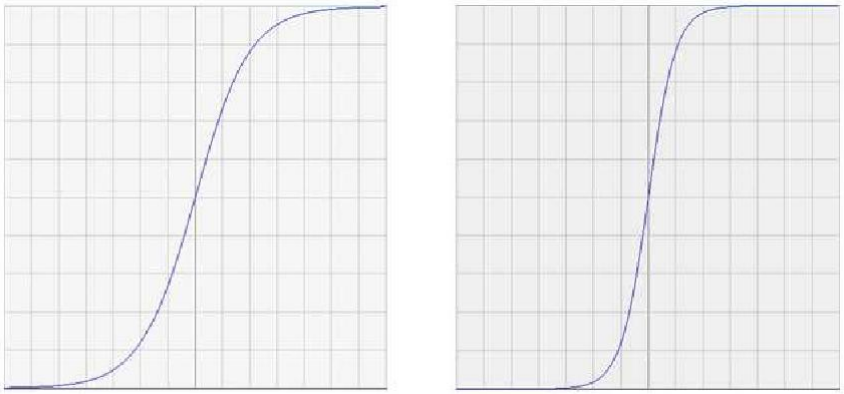
where $y(x^i, w) = s(x^i \cdot w)$ with $s(z) = 1/(1 + e^{-z})$ being our usual sigmoid function.

1. Write an expression for $\partial E/\partial w_j$. Your answer should be in terms of the training data.

   We're using y in the last line here as shorthand for $y(x^i, w)$, that is, the output of the network on input x^i.

2. What update should be made to weight w_j given a single training example $x, y^*)$. Your answer should be in terms of the training data.

3. Here are two graphs of the output of the sigmoid unit as a function of a single feature x. The unit has a weight for x and an offet. The two graphs are made using different values of the magnitude of the weight vector ($\|w\|^2 = \sum_j w_j^2$).



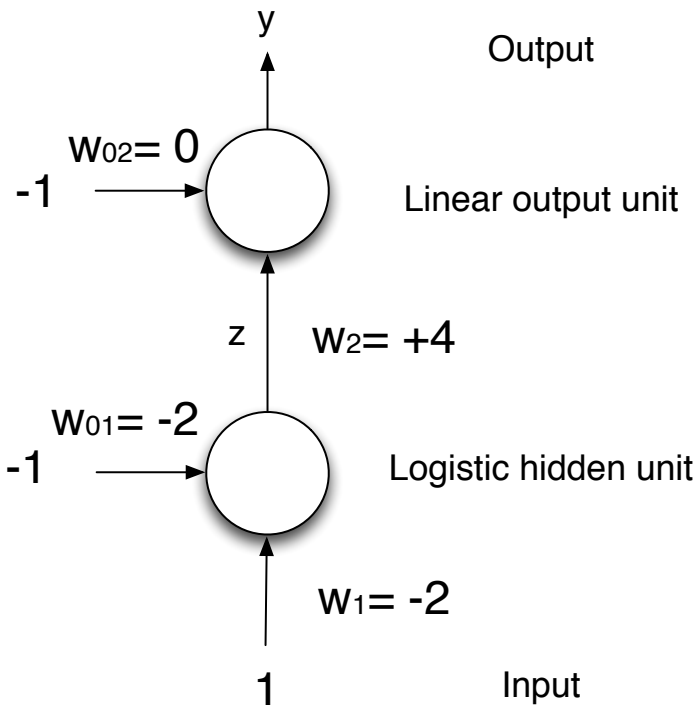
A B

   Which of the graphs is produced by the larger $\|w\|^2$? Explain.

4. Why might penalizing large $\|w\|^2$, as we could do above by choosing a positive β, be desirable?

14 Backpropagation

Here you see a very small neural network: it has one input unit, one hidden unit (logistic), and one output unit (linear).



Let's consider one training case. For that training case, the input value is 1 (as shown in the diagram), and the target output value $t = 1$. We're using the following loss function:

$$E = \frac{1}{2}(t - y)^2$$

Please supply numeric answers; the numbers in this question have been constructed in such a way that you don't need a calculator. Show your work in case of mis-calculation in earlier steps.

(a) What is the output of the hidden unit for this input?

(b) What is the output of the output unit for this input?

(c) What is the loss, for this training case?

(d) What is the derivative of the loss with respect to w_2, for this training case?

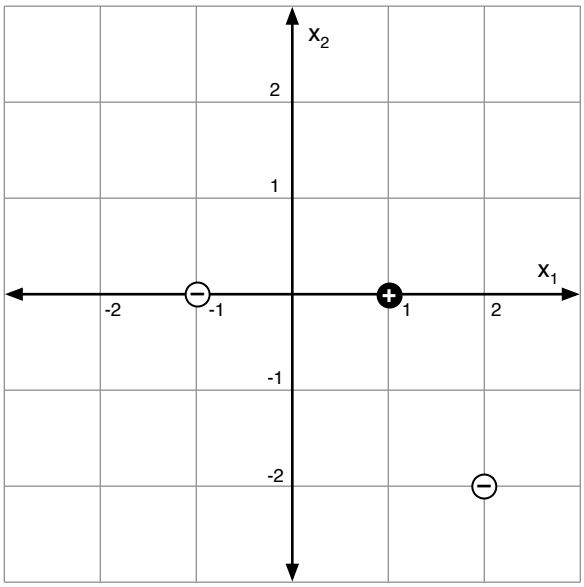
(e) What is the derivative of the loss with respect to w_1, for this training case?

(f) With sigmoidal activation, the derivative with respect to w_1 and w_2 are

$$\frac{\partial E}{\partial w_2} = (t - y)z, \text{ and } \frac{\partial E}{\partial w_1} = (t - y) \cdot w_2 \cdot z \cdot (1 - z) \cdot x.$$

Assume that we now use the rectified linear unit (ReLU) as our activation (or a *ramp* function). This means that $z = \max(0, w_1 x + w_{01})$. What is the derivative of the loss with respect to w_1 and w_2 *at differentiable points* with ReLU? Don't use numerical value for this question.

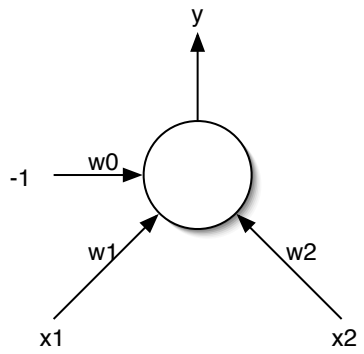
15 Neural Net



Data points are: Negative: (-1, 0) (2, -2) Positive: (1, 0)

Recall that for neural nets with sigmoidal output units, the negative class is represented by a desired output of 0 and the positive class by a desired output of 1. Hint: Some useful values of the sigmoid $s(z)$ are $s(-1) = 0.27$ and $s(1) = 0.73$.

Assume we have a single sigmoid unit:



Assume that the weights are $w_0 = 0, w_1 = 1, w_2 = 1$. What is the computed y value for each of the points on the diagram above?

(a) $x = (-1, 0)$

(b) $x = (2, -2)$

(c) $x = (1, 0)$

(d) What would be the change in w_2 as determined by backpropagation using a step size (η) of 1.0? Assume the squared loss function. Assume that the input is $x = (2, -2)$ and the initial weights are as specified above. Show the formula you are using as well as the numerical result.

1. $\Delta w_2 =$

16 Probable cause

You have a binary classification problem, but your training examples are only labeled with probabilities, so your data set consists of pairs $(x^{(i)}, p^{(i)})$, where $p^{(i)}$ is the probability that $x^{(i)}$ belongs to class 1.

You want to train a neural network **with a single unit** to predict these probabilities.

(a) What is a good choice for the activation function of your final output unit?

(b) You can think of the training label $p^{(i)}$ as specifying a true probability and the current output of your neural network as specifying an approximate probability $q^{(i)}$. You think a reasonable objective would be to minimize the KL divergence $KL(p \parallel q)$ between the distributions implicitly represented by the predicted and target outputs. So, the empirical risk would be

$$E = \sum_i -(p^{(i)} \log q^{(i)} + (1 - p^{(i)}) \log(1 - q^{(i)}))$$

If $q^{(i)} = f(w \cdot x^{(i)})$ where f is your activation function, what is the SGD (stochastic gradient descent) weight update rule when using the KL objective function above? For simplicity, assume that $x^{(i)}$ and w are both scalars and write f' for the derivative of f.