CS 189 Introduction to Machine Learning
Fall 2017 gs2

1 Backprop: *Adapted from Fall 2015 Final Exam*

Suppose we have a neural network that takes in $P = 2000$ input features, has $H = 500$ hidden units, and $N = 4$ outputs.

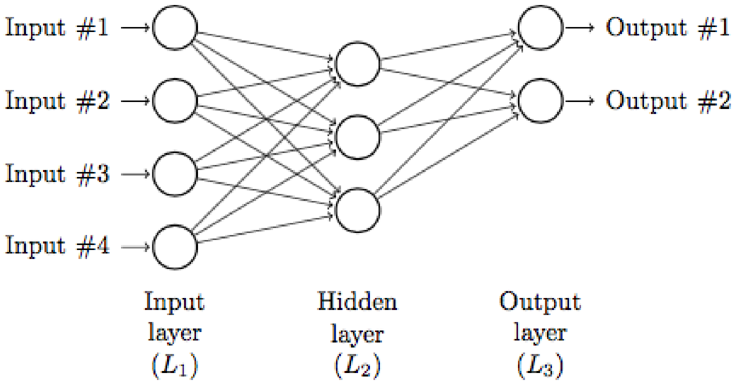


Figure 1: Network without all input and hidden layer nodes shown

$f(x)$ is the activation function in the hidden layer, and $g(x)$ is the activation function in the output layer. You may refer to their respective partial derivatives as $f'(x)$ and $g'(x)$. We will use the mean squared error loss function, which is $L(z) = \frac{1}{2}||z-y||^2$ where z is your estimate for label y.

Let's define some notation!
$x \in \mathscr{R}^P$ is a feature vector
$s_j^h = \sum_{i=1}^{P} V_{i,j} x_i$ are inputs to the hidden layer
$h_j = f(s_j^h)$
$s_k^o = \sum_{j=1}^{H} W_{j,k} h_j$ are inputs to the output layer
$z_k = g(s_k^o)$
$V \in \mathscr{R}^{H \times P}$ are weights between input layer and hidden layer
$W \in \mathscr{R}^{N \times H}$ are weights between hidden layer and output layer

(a) Suppose you want to include a bias term in both the input layer and hidden layer, how many weights will be trained? How does this change our weight matrices V and W?

Solution:

$$2001 \times 500 + 501 \times 4 = 1002504$$

Our weight matrices have different dimensions.

$$V \in \mathscr{R}^{H \times (P+1)} \text{ and } W \in \mathscr{R}^{N \times (H+1)}$$

(b) Derive the following terms:

$$\frac{\partial L}{\partial W_{j,k}} =$$
$$\frac{\partial L}{\partial V_{i,j}} =$$

Solution: Back propagation is just a nice term for calculating derivatives using the chain rule. **The key to using the chain rule in neural networks is to know which variables are related and which variables are independent of each other**. Since W_{jk} is the weight between neuron h_j in the hidden layer and neuron z_k in the output layer, it's important to see W_{jk} and z_k are related while W_{jk} and z_{k+1} are not. By the same reason,

$$\begin{aligned}\frac{\partial L}{\partial W_{j,k}} &= \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial s^o_k}\frac{\partial s^o_k}{\partial W_{j,k}} \\ &= (z_k - y_k)g'(s^o_k)h_j\end{aligned}$$

In the same way, V_{ij} is the weight between each neuron x_i in the input layer and each neuron h_j in the hidden layer and each h_j affects all the neurons in the output layer. Thus when deriving $\frac{\partial L}{\partial V_{ij}}$, we should apply the chain rule over all output neurons and over only a single neuron h_j in the hidden layer.

$$\begin{aligned}\frac{\partial L}{\partial V_{i,j}} &= \sum_{k=1}^{N} \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial s^o_k} \cdot \frac{\partial s^o_k}{\partial h_j} \cdot \frac{\partial h_j}{\partial s^h_j} \cdot \frac{\partial s^h_j}{\partial V_{i,j}} \\ &= \sum_{k=1}^{N} (z_k - y_k)g'(s^o_k)W_{j,k}f'(s^h_j)x_i\end{aligned}$$

(c) As you probably noticed in your homework, vectorizing our backpropagation can speed up computation by taking advantages of matrix operation libraries like numpy. Vectorize the partial derivatives from part (b) and derive the update equation for stochastic gradient descent with learning rate η_1 and η_2 for V and W. (Use $\otimes$ for elementwise multiplication)

Solution: Since $\frac{\partial L}{\partial z_k}$ and $\frac{\partial z_k}{\partial s^o_k}$ are associated with subscript k, we should apply element wise multiplication between them. Since $\frac{\partial L}{\partial s^o_k}$ and $\frac{\partial s^o_k}{\partial W_{j,k}}$ have different subscript j involved, we should apply matrix multiplication between them.

$$\frac{\partial L}{\partial W} = ((z-y) \otimes g'(S^o))h^\intercal$$

Using similar reason as above, we should apply matrix multiplication between $\frac{\partial L}{\partial s_k^o}$ and $\frac{\partial s_k^o}{\partial h_j}$, element-wise multiplication between $\frac{\partial L}{\partial h_j}$ and $\frac{\partial h_j}{\partial s_j^h}$, matrix multiplication between $\frac{\partial L}{\partial s_j^h}$ and $\frac{\partial s_j^h}{\partial V_{i,j}}$. Note that one should mention $\frac{\partial L}{\partial V} \in \mathbf{R}^{500\times 2001}$ in which is all but last row of W in the problem setup since bias term in hidden layer is independent to the values in the input layer.

$$\frac{\partial L}{\partial V} = (W^\intercal((z-y)\otimes g'(S^o))\otimes f'(S^h))x^\intercal$$

And the update equations is given by:

$$\begin{aligned}
V &= V - \eta_1 \frac{\partial L}{\partial V} \\
&= V - \eta_1 (W^\intercal((z-y)\otimes g'(S^o))\otimes f'(S^h))x^\intercal \\
W &= W - \eta_2 \frac{\partial L}{\partial W} \\
&= W - \eta_2((z-y)\otimes g'(S^o))h^\intercal
\end{aligned}$$

(d) The softmax function, or normalized exponential function, is a generalization of the logistic function to handle multiclass classification. The softmax function "squashes" a N-dimensional vector s of arbitrary real values to a K-dimensional vector $\sigma(s)$ of real values in the range [0, 1] that add up to 1. The i-th value corresponds to the probability that the output is class i. The function is given by the following:

$$\sigma(s_j) = \frac{e^{s_j}}{\sum_{k=1}^N e^{s_k}} \qquad \text{for } j = 1...N$$

The cross entropy error is commonly paired with softmax activation in the output layer and is defined as:

$$L(z) = \sum_{k=1}^N y_k \ln(z_k)$$

Now, let $g(x)$ be the softmax function and let our loss function be the cross entropy loss. Calculate the partial derivative of L with respect to $W_{i,j}$.

Solution: Notice, that our solution from part (b) is no longer accurate because we had assumed z_k did not depend on $W_{i,j}$ if $k \neq j$. However, by the mechanics of the softmax function, that is no longer true, so we have to sum over all z_k.

$$\frac{\partial L}{\partial W_{i,j}} = \sum_{k=1}^N \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial s_j}\frac{\partial s_j}{\partial W_{i,j}}$$

$$\frac{\partial L}{\partial z_k} = -\frac{y_k}{z_k}$$

$$\frac{\partial z_k}{\partial s_j} = \begin{cases} z_k(1-z_k) & \text{if } i = k \\ -z_i z_k & \text{if } i \neq k \end{cases}$$

$$\frac{\partial s_j}{\partial W_{i,j}} = h_i$$

$$\begin{aligned}
\frac{\partial L}{\partial W_{i,j}} &= \sum_{k=1}^{N} \frac{\partial L}{\partial z_k} \frac{\partial z_k}{\partial s_j} \frac{\partial s_j}{\partial W_{i,j}} \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(\frac{\partial L}{\partial z_j} \frac{\partial z_j}{\partial s_j} + \sum_{k \neq j}^{N} \frac{\partial L}{\partial z_k} \frac{\partial z_k}{\partial s_j}\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(-\frac{y_j}{z_j} z_j(1-z_j) + \sum_{k \neq j}^{N} -\frac{y_k}{z_k}(-z_j z_k)\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(-y_j(1-z_j) + \sum_{k \neq j}^{N} y_k z_j\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(-y_j + y_j z_j + \sum_{k \neq j}^{N} y_k z_j\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(-y_j + \sum_{k=1}^{N} y_k z_j\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} \Big(-y_j + z_j \sum_{k=1}^{N} y_k\Big) \\
&= \frac{\partial s_j}{\partial W_{i,j}} (-y_j + z_j) \\
&= h_i(-y_j + z_j)
\end{aligned}$$

2 Kernel PCA

Let $X \in \mathbb{R}^{n \times d}$ be a matrix with rows $x_1^T \dots x_n^T$, and $\phi : X \longrightarrow X'$ be a feature map with associated kernel $k(x,y) = \langle \phi(x_1), \phi(x_2) \rangle$. We will attempt to perform kernel PCA on X using the feature mapping ϕ. For this problem, assume that the data is already centered.

(a) Let $\Sigma = \frac{1}{n} \sum_{i=1}^n x_i x_i^T$ be the sample covariance matrix. Recall that performing PCA involves solving for the eigenvectors and eigenvalues of Σ.

Show if v is a solution (an eigenvector of Σ, e.g. $\Sigma v = \lambda v$), then v is in the range of X^T, or $v = X^T \alpha$ for some α.

Solution:

$$\Sigma v = \frac{1}{n} \sum_{i=1}^n (x_i x_i^T) v = \sum_{i=1}^n \frac{1}{n} x_i \langle v, x_i \rangle = X^T \alpha$$

where $\alpha_i = \frac{1}{n} \langle v, x_i \rangle$.

(b) Show if (α, λ) is a solution to $XX^T \alpha = \lambda \alpha$, then $(v = X^T \alpha, \lambda)$ solves $X^T X v = \lambda v$.

Solution:

If (α, λ) solves $XX^T \alpha = \lambda \alpha$, then $Xv = \lambda \alpha$ and by left multiplying each side by X^T, we have $X^T X v = \lambda X^T \alpha = \lambda v$.

(c) Kernel PCA solves $\phi(\Sigma) v = \lambda v$, where $\phi(\Sigma) = \frac{1}{n} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T$. Explain how to find all λ satisfying this equation without explicitly computing $\phi(x)$.

Solution:

By the second problem, we know we can just solve $\phi(X) \phi(X)^T \alpha = \lambda \alpha$ where $\phi(X)$ is just a matrix with rows $\phi(x_i)^T$. Notice that $(\phi(X) \phi(X)^T)_{i,j} = k(x_i, x_j) = K_{i,j}$, where K is our kernel matrix. Thus we can compute our kernel matrix K, and then solve $K \alpha = \lambda \alpha$.

(d) In regular PCA, the projection coefficient of x onto a principal component v is $\langle x, v \rangle$. In feature space, the coefficient is $\langle \phi(x), v \rangle$. How do we compute this without explicitly computing $\phi(x)$?

Solution:

$$
\begin{aligned}
\langle v, \phi(x) \rangle &= \langle \phi(X)^T \alpha, \phi(x) \rangle \\
&= \alpha^T \phi(X) \phi(x) \\
&= \alpha^T \begin{bmatrix} \langle \phi(x_1), \phi(x) \rangle \\ \vdots \\ \langle \phi(x_n), \phi(x) \rangle \end{bmatrix} \\
&= \alpha^T \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{bmatrix}
\end{aligned}
$$

With this formulation, we can compute $k(x_1,x) \dots k(x_n,x)$ instead of computing $\phi(x)$.

(e) To get the correct projection coefficient in regular PCA, we always normalize eigenvectors v so that $\langle x, v \rangle$ gives the correct coefficient. How can we equivalently ensure proper normalization in our kernel PCA?

Solution:

We have

$$\langle v, v \rangle = \langle X^T \alpha, X^T \alpha \rangle = \alpha^T X X^T \alpha = \alpha^T (\lambda \alpha) = \lambda \langle \alpha, \alpha \rangle.$$

Thus, to ensure $\langle v, v \rangle = 1$, we can scale α such that $\langle \alpha, \alpha \rangle = \frac{1}{\lambda}$.

3 QDA Isocontours

Given the 2×2 matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

(a) Determine the eigenvalues and set of orthonormal eigenvectors of A

Solution:

$$Eigenvalues = \begin{cases} \lambda_1 = 3 \\ \lambda_2 = 1 \end{cases}$$

$$Eigenvector = \begin{cases} u_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\ u_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \end{cases}$$

(b) Determine the shape of the set given by constraint $x^\top Ax = 1$, where $x \in \mathbb{R}^2$ and $\|x\|_2 = 1$

Solution: The shape of the set is an ellipse with principle axes u_1 with length $1/\sqrt{3}$ and u_2 with length 1

(c) Recall that the quadratic for an anisotropic distribution of class C is given by the following formula

$$Q_C(x) = -\frac{1}{2}(x-\mu_c)^\top \Sigma_C^{-1}(x-\mu_C) - \frac{1}{2}|\Sigma_C| + \ln \pi_C$$

Assume that we are given two classes: C and D described by the following parameters:

$$\Sigma_C^{-1} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}, \Sigma_D^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

$$\mu_C = \mu_D = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\pi_C = \pi_D$$

Also recall that the Bayes decision boundary is the set of points where each class is equally likely. Determine the formula for the Bayes decision boundary.

Solution: Calculate the Bayes decision boundary

$$0 = Q_C(x) - Q_D(x)$$

Abbreviating μ_C and μ_D as μ,

$$0 = -\frac{1}{2}(x-\mu)^\top(\Sigma_C^{-1} - \Sigma_D^{-1})(x-\mu) - \frac{1}{2}\ln|\Sigma_C| + \frac{1}{2}\ln|\Sigma_D|$$

$$\frac{1}{2}(x-\mu)^\top \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} (x-\mu) = -\frac{1}{2}\ln|\Sigma_C| + \frac{1}{2}\ln|\Sigma_D|$$

$$(x-\mu)^\top \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} (x-\mu) = -\ln|\Sigma_C| + \ln|\Sigma_D|$$

$$(x-\mu)^\top \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} (x-\mu) = \ln(\frac{16}{3})$$

(d) Assume that the Bayes decision boundary has the following form:

$$(x-\mu)^\top \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} (x-\mu) = 1$$

Draw out the Bayes decision boundary.

Solution:

It becomes the same shape as in part (b).

(e) Assume now that we are given a new Bayes decision boundary

$$x^\top \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x = \ln(\frac{4}{3})$$

What would this decision boundary look like?

Solution: A hyperbola

$$x_1^2 - x_2^2 = \ln(\frac{4}{3})$$

4 SVM

(a) In a soft margin SVM, if we increase C, which of the following are likely to happen? (Select all that apply)

(a) The margin will grow wider.

(b) All slack variables will go to 0.

(c) The norm $||w||_2$ will grow larger.

(d) There will be more points inside of the margin.

Solution: C

(b) If a hard margin svm tries to minimize $||w||_2^2$ subject to $y_i(w^T x_i + b) \geq c$ for some c, what will the width of the slab (the empty region) be?

(a) $\frac{2}{||w||_2^2}$

(b) $\frac{2c}{||w||_2^2}$

(c) $\frac{c}{||w||_2^2}$

(d) $\frac{2}{c||w||_2^2}$

Solution: B

(c) The shortest distance from a point z to a hyperplane $w^T x = 0$ is

(a) $w^T z$

(b) $\frac{w^T z}{||w||_2}$

(c) $\frac{w^T z}{||w||_2^2}$

(d) $||w||_2||z||_2$

Solution: B

(d) Which of the following is true about SVM's? (Select all that apply)

(a) For soft margin SVM's, a solution exists if and only if the data is linearly separable.

(b) For hard margin SVM's, the support vectors are the only points needed to calculate the decision boundary.

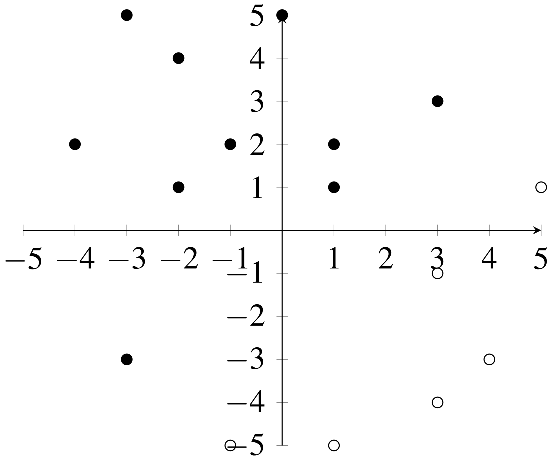
(c) Both hard margin and soft margin SVM's can perfectly separate any training data in some feature space.

(d) Least squares SVM with $l2$-regularization has a closed form solution.

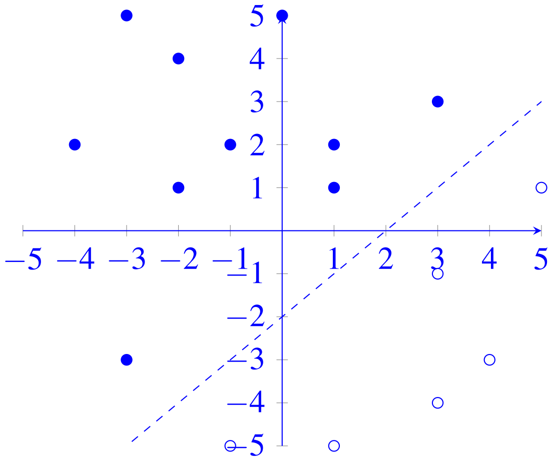
Solution: B, C, D

(e) Let's take a look at a set of points and see what a hard margin SVM would determine as the hyperplane.

i. Sketch and find w, b, for the hyperplane $H = \{w^T x = b\}$ found by a hard margin SVM. What are the support vectors and margin width?



Solution:



By inspection, one can see that the line we are looking for is $x_2 = x_1 - 2$, which has a slope of 1. Thus, we know $w = \begin{bmatrix} c \\ -c \end{bmatrix}$ for some constant c.

We can also see that we have support vectors are $(-5,-1)$, $(-3,-3)$, $(1,1)$, $(3,-1)$, $(5,1)$, and $(3,3)$, but we only need one from each side to find the hyperplane. Thus, since our constraints of $y_i(w^T x_i + b) \geq 1$ have equality for our support vectors:

$$3c - (-1)c + b = -1$$
$$1c - 1c + b = 1$$

Solving this system, we get that $b = 1$, $4c + b = -1$ which means $4c = -2$ or $c = -\frac{1}{2}$.

Thus, our hyperplane $H = w^T x + b$ has $w = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$ and $b = 1$.

The margin width is $\frac{2}{||w||_2}$, or $\frac{2}{\frac{1}{\sqrt{2}}} = 2\sqrt{2}$.

ii. How many points can I remove without affecting the resulting hyperplane? There are 10 closed points and 6 open points.

Solution:

14. I can remove everything except for $(1,1)$, and $(3,-1)$. Note that while there are more support vectors, just keeping these two will yield the same hyperplane.

5 SVM Dual

Recall from Stella's SVM lectures that we can formulate the SVM problem in a different way than its primal.

From lecture, we've seen that a convex optimization with a primal of the form

$$\begin{aligned}&\min_x f(x)\\ \text{s.t. }&g(x) \leq 0\end{aligned}$$

has an associated dual problem, mainly

$$\begin{aligned}&\max_\alpha \min_x \mathscr{L}(x, \alpha)\\ &\quad\text{s.t. } \alpha \geq 0\end{aligned}$$

where $\mathscr{L}(x, \alpha)$ is known as the Lagrangian of the primal. These optimization problems are "equivalent" in the sense that they share the same optimal value and an optimal solution of the primal can be recovered easily from the dual and vice versa.

For the dual to be optimal, recall the KKT conditions:

1. Stationary: $\frac{d\mathscr{L}}{dx} = \frac{df(x)}{dx} + \alpha^T \frac{dg(x)}{dx} = 0$
2. Primal feasibility: $g(x) \leq 0$
3. Dual feasibility: $\alpha \geq 0$
4. Complementary slackness: $\alpha_i g(x)_i = 0$

Using this knowledge, you will now derive dual SVM.

(a) Recall that the primal problem of SVM is of the form

$$\begin{aligned}\max_{w,t\xi} \varepsilon(w,t,\xi) = \max_{w,t,\xi} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n} \xi_i\\ \text{s.t. } y_i(w \cdot x_i - t) \geq 1 - \xi_i,\ \xi_i \geq 0\end{aligned}$$

What is the lagrangian $\mathscr{L}(w,t,\xi,\alpha,\beta)$ for the primal? Use α_i for the $y_i(w \cdot x_i - t) \geq 1 - \xi_i$ constraint and β_i for the $\xi_i \geq 0$ constraint.

Solution: We can rewrite the constraint $y_i(w \cdot x_i - t) \geq 1 - \xi_i$ as $(1 - \xi_i) - y_i(w \cdot x_i - t) \leq 0$ to get it into the form listed at the beginning of the problem. Now, we know the lagrangian is simply

$$\begin{aligned}\mathscr{L}(w,t,\xi,\alpha,\beta) &= \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(-y_i(w\cdot x_i - t) + (1-\xi_i)) + \sum_{i=1}^{n}\beta_i(-\xi_i) \\ &= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n}\alpha_i y_i(w\cdot x_i - t) + \sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}(C-\alpha_i-\beta_i)\xi_i\end{aligned}$$

(b) Using KKT conditions, derive the value of w in terms of dual variables α_i at the optimal point. What is significant about this value of w? Furthermore, prove that at optimum for the dual, $0 \le \alpha_i \le C$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$.

Solution: Using the first KKT condition, we can take gradients/partial derivatives with respect to w, t, and ξ_i and set them to 0 to derive conditions at the optimal point.

We know that $\frac{d\mathscr{L}}{dw} = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0$. Thus, at optimal, $w = \sum_{i=1}^{n}\alpha_i y_i x_i$. This is significant because we've now shown that at optimal, the weight vector w can be written as a linear combination of the sample points x_i.

Similarly, $\frac{d\mathscr{L}}{dt} = \sum_{i=1}^{n}\alpha_i y_i$, so at optimal, $\sum_{i=1}^{n}\alpha_i y_i = 0$.

Lastly, $\frac{d\mathscr{L}}{d\xi_i} = C - \alpha_i - \beta_i$, so $C - \alpha_i - \beta_i = 0$ at optimum.

Since by the third KKT condition α_i and β_i must be dual feasible, or $\alpha_i \ge 0$ and $\beta_i \ge 0$, we have that $\alpha_i = C - \beta_i$ where $\beta_i \ge 0$, which must mean that $\alpha_i \le C$. Thus, $0 \le \alpha_i \le C$ at optimum.

(c) Write the dual problem for SVM in terms of the dual variables $\alpha = \begin{bmatrix}\alpha_1 & \dots & \alpha_n\end{bmatrix}^T$ and matrix Q, where $Q_{ij} = y_i y_j (x_i^T x_j)$.

Solution: In general, we have:

$$\begin{aligned}\mathscr{L}(w,t,\xi,\alpha,\beta) &= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n}\alpha_i y_i(w\cdot x_i - t) + \sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}(C-\alpha_i-\beta_i)\xi_i \\ &= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n}\alpha_i y_i w\cdot x_i - t\sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}(C-\alpha_i-\beta_i)\xi_i\end{aligned}$$

Note that the term $t\sum_{i=1}^{n}\alpha_i y_i = 0$ at optimum, though, since $\sum_{i=1}^{n}\alpha_i y_i = 0$ at optimum. Further, since $C - \alpha_i - \beta_i = 0$ at optimum, then the term $\sum_{i=1}^{n}(C-\alpha_i-\beta_i)\xi_i = 0$ at optimum.

If we replace w with the expression found in part 2 with what we know it will be at optimum, we have:

$$
\begin{aligned}
\mathscr{L}(w,t,\xi,\alpha,\beta) &= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n}\alpha_i y_i (w\cdot x_i) + \sum_{i=1}^{n}\alpha_i \\
&= \frac{1}{2}||\sum_{i=1}^{n}\alpha_i y_i x_i||_2^2 - \sum_{i=1}^{n}\alpha_i y_i((\sum_{j=1}^{n}\alpha_j y_j x_j)\cdot x_i) + \sum_{i=1}^{n}\alpha_i \\
&= \frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i((\sum_{j=1}^{n}\alpha_j y_j x_j)\cdot x_i) - \sum_{i=1}^{n}\alpha_i y_i((\sum_{j=1}^{n}\alpha_j y_j x_j)\cdot x_i) + \sum_{i=1}^{n}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i((\sum_{j=1}^{n}\alpha_j y_j x_j)\cdot x_i) + \sum_{i=1}^{n}\alpha_i \\
&= -\frac{1}{2}\alpha^T Q\alpha + 1^T\alpha
\end{aligned}
$$

Thus, $\min_{w,t,\xi}\mathscr{L}(w,t,\xi,\alpha,\beta) = -\frac{1}{2}\alpha^T Q\alpha + 1^T\alpha$, and thus we have the following dual problem:

$$
\max_{\alpha} -\frac{1}{2}\alpha^T Q\alpha + 1^T\alpha
$$
$$
\text{s.t. } 0 \leq \alpha_i \leq C, \sum_{i=1}^{n}\alpha_i y_i x_i = 0
$$

(d) Prove the following (HINT: use the “Complementary slackness” KKT condition):

(a) $\alpha_i = 0$ means x_i is on or outside of the margin.

Solution: We know at optimum that $C - \alpha_i - \beta_i = 0$, but if $\alpha_i = 0$, then $C = \beta_i \neq 0$. By complementary slackness, $\beta_i \neq 0$ must mean that $\xi_i = 0$, which we know means x_i is on or outside of the margin.

(b) $\alpha_i = C$ means x_i violates the margin.

Solution: Like part (a), $C - \alpha_i - \beta_i = 0$, but if $\alpha_i = C$, then $\beta_i = 0$. By complementary slackness, $\beta_i = 0$ must mean that $\xi_i \neq 0$, which we know means x_i must violate the margin.

(c) $0 < \alpha_i < C$ means x_i is a support vector.

Solution: Since $C - \alpha_i - \beta_i = 0$, this must mean that $\beta_i = C - \alpha_i > 0$, and thus $\xi_i = 0$. Further, since $\alpha > 0$, this must mean that by complementary slackness that $y_i(w\cdot x_i - t) - 1 = 0$, and thus $y_i(w\cdot x_i - t) = 1$, which means this x_i must be a support vector.

6 Logistic Regression

Consider the log-likelihood function for logistic regression

$$\ell(\theta) = \sum_{i=1}^{m} y^{(i)} \ln h(x^{(i)}) + (1-y^{(i)}) \ln(1-h(x^{(i)}))$$

Find the Hessian H of this function , and show that for every z, it holds true that

$$z^\top H z \leq 0$$

(Hint: You might want to start by showing that $\sum_i \sum_j z_i x_i x_j z_j = (x^\top z)^2$)

Solution: Recall that we have $g'(z) = g(z)(1-g(z))$, and thus for $h(x) = g(\theta^\top x)$. We have $\frac{\partial h(x)}{\partial \theta_k} = h(x)(1-h(x))x_k$.

$$\begin{aligned}
\frac{\partial l(\theta)}{\partial \theta_k} &= \sum_{i=1}^{m} (y^{(i)} - h(x^{(i)}))x_k^{(i)} \\
H_{kl} &= \frac{\partial^2 l(\theta)}{\partial \theta_k \partial \theta_l} \\
&= \sum_{i=1}^{m} -\frac{\partial h(x^{(i)})}{\partial \theta_l} x_k^{(i)} \\
&= \sum_{i=1}^{m} -h(x^{(i)})(1-h(x^{(i)}))x_l^{(i)} x_k^{(i)}
\end{aligned}$$

So we have for the hessian matrix H (using that for $X = xx^\top$ if and only if $X_{ij} = x_i x_j$):

$$H = -\sum_{i=1}^{m} h(x^{(i)})(1-h(x^{(i)}))x^{(i)} x^{(i)\top}$$

And to prove that H is negative semi-definite, we show $z^\top H z \leq 0$ for all z

$$\begin{aligned}
z^\top H z &= -z^\top (\sum_{i=1}^{m} h(x^{(i)})(1-h(x^{(i)}))x^{(i)} x^{(i)\top})z \\
&= -\sum_{i=1}^{m} h(x^{(i)})(1-h(x^{(i)}))z^\top x^{(i)} x^{(i)\top} z \\
&= -\sum_{i=1}^{m} h(x^{(i)})(1-h(x^{(i)}))(z^\top x^{(i)})^2 \\
&\leq 0
\end{aligned}$$

with the last inequality holding, since $0 \leq h(x^{(i)}) \leq 1$, which implies $h(x^{(i)}(1-h(x^{(i)})) \geq 0$, and $(z^\top x^{(i)})^2 \geq 0$