Exercise solutions

Chapter 2. Fundamentals of numerical optimization

Exercise 2.1

a) Using derivative formula and the fact that the derivative of a sum is the sum of derivatives we have

$$g'(w) = qw + r, \tag{1}$$

and

$$g''(w) = q. \tag{2}$$

b) Using the chain rule on the $\cos(\cdot)$ part, the fact that the derivative of a sum is the sum of derivatives, and derivative formulae we have

$$g'(w) = \sin\left(2\pi w^2\right) 4\pi w + 2w. \tag{3}$$

Likewise taking the second derivative we differentiate the above (additionally using the product rules) as

$$g''(w) = \cos\left(2\pi w^2\right) (4\pi w)^2 + \sin\left(2\pi w^2\right) 4\pi + 2. \tag{4}$$

c) Call the p^{th} summand $h_p(w) = \log\left(1 + e^{-a_p w}\right)$. Then using the chain rule since $\frac{d}{dt}\log(t) = \frac{1}{t}$ and $\frac{d}{dw}\left(1 + e^{-a_p w}\right) = -a_p e^{-a_p w}$ together we have

$$\frac{d}{dw} h_p(w) = \frac{1}{1 + e^{-a_p w}} \left(-a_p e^{-a_p w}\right) = -\frac{a_p e^{-a_p w}}{1 + e^{-a_p w}} = -\frac{a_p}{e^{a_p w} + e^{a_p w} e^{-a_p w}} = -\frac{a_p}{1 + e^{a_p w}}. \tag{5}$$

Now using this result, and since the derivative of a sum is the sum of the derivatives and $g(w) = \sum_{p=1}^{P} h_p(w)$, we have $\frac{d}{dw} g(w) = \sum_{p=1}^{P} \frac{d}{dw} h_p(w)$ and so

$$g'(w) = -\sum_{p=1}^{P} \frac{a_p}{1 + e^{a_p w}}. \tag{6}$$

To compute the second derivative let us again do so by first differentiating the above summand-by-summand. Denote the p^{th} summand above as $h_p(w) = \frac{a_p}{1+e^{a_p w}}$. To compute its derivative we must apply the product and chain rules once again, we have $h'_p(w) = -\frac{a_p}{(1+e^{a_p w})^2} a_p e^{a_p w} = -\frac{e^{a_p w}}{(1+e^{a_p w})^2} a_p^2$. We can then compute the full second derivative as $g''(w) = -\sum_{p=1}^{P} h'_p(w)$ or likewise

$$g''(w) = \sum_{p=1}^{P} \frac{e^{a_p w}}{(1+e^{a_p w})^2} a_p^2. \tag{7}$$

Exercise 2.2

a) Writing out g in terms of the individual entries of $\mathbf{w}$ we have $g(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} w_n Q_{nm} w_m + \sum_{n=1}^{N} r_n w_n + d$ then taking the j^{th} partial derivative we have, since the derivative of a sum is the sum of derivatives $\frac{\partial}{\partial w_j} g(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \frac{\partial}{\partial w_j}(w_n Q_{nm} w_m) + \sum_{n=1}^{N} \frac{\partial}{\partial w_j}(r_n w_n)$ where the d vanishes since it is a constant and $\frac{\partial}{\partial w_j} d = 0$. Now evaluating each derivative we apply the product rule to each $w_n Q_{nm} w_m$ (and remembering that all other terms in w_k where $k \neq j$ are constant and thus vanish when taking the w_j partial derivative) we have

$$\frac{\partial}{\partial w_j} g(\mathbf{w}) = \frac{1}{2}\left(\sum_{n=1}^{N} w_n Q_{nj} + \sum_{m=1}^{N} Q_{jm} w_m\right) + r_j. \tag{8}$$

All together the gradient can then be written compactly as

$$\nabla g(\mathbf{w}) = \frac{1}{2}\left(\mathbf{Q} + \mathbf{Q}^T\right)\mathbf{w} + \mathbf{r}, \tag{9}$$

and because $\mathbf{Q}$ is symmetric this is equivalently

$$\nabla g(\mathbf{w}) = \mathbf{Q}\mathbf{w} + \mathbf{r}. \tag{10}$$

To compute the Hessian we compute mixed partial derivatives of the form $\frac{\partial^2}{\partial w_i \partial w_j} g(\mathbf{w})$. To do this efficiently we can take the partial $\frac{\partial}{\partial w_i}$ of equation (8) since $\frac{\partial^2}{\partial w_i \partial w_j} g(\mathbf{w}) = \frac{\partial}{\partial w_i}\left(\frac{\partial}{\partial w_j} g(\mathbf{w})\right)$ which gives

$$\frac{\partial^2}{\partial w_i \partial w_j} g(\mathbf{w}) = \frac{1}{2}\left(Q_{ij} + Q_{ji}\right). \tag{11}$$

All together we then have that the full Hessian matrix is

$$\nabla^2 g\left(\mathbf{w}\right) = \frac{1}{2}\left(\mathbf{Q} + \mathbf{Q}^T\right), \tag{12}$$

and because $\mathbf{Q}$ is symmetric this is equivalently

$$\nabla^2 g\left(\mathbf{w}\right) = \mathbf{Q}. \tag{13}$$

b) Writing out g in terms of individual entries of $\mathbf{w}$ we have $g\left(\mathbf{w}\right) = -\cos\left(2\pi \sum_{n=1}^{N} w_n^2\right) + \sum_{n=1}^{N} w_n^2$, now taking the j^{th} partial we have

$$\frac{\partial}{\partial w_j} g\left(\mathbf{w}\right) = \sin\left(2\pi \sum_{n=1}^{N} w_n^2\right) 4\pi w_j + 2w_j. \tag{14}$$

From this we can see that the full gradient then takes the form

$$\nabla g\left(\mathbf{w}\right) = \sin\left(2\pi \mathbf{w}^T\mathbf{w}\right) 4\pi\mathbf{w} + 2\mathbf{w}. \tag{15}$$

To compute the second derivatives we can take the partial $\frac{\partial}{\partial w_i}$ of equation (14) which gives

$$\frac{\partial^2}{\partial w_i \partial w_j} g\left(\mathbf{w}\right) = \begin{cases} \cos\left(2\pi \sum_{n=1}^{N} w_n^2\right)\left(4\pi\right)^2 w_i w_j + \sin\left(2\pi \sum_{n=1}^{N} w_n^2\right) 4\pi + 2 & \text{if } i = j \\ \cos\left(2\pi \sum_{n=1}^{N} w_n^2\right)\left(4\pi\right)^2 w_i w_j & \text{else.} \end{cases} \tag{16}$$

All together then denoting $\mathbf{I}_{N\times N}$ the $N \times N$ identity matrix we may write the Hessian as

$$\nabla^2 g\left(\mathbf{w}\right) = \cos\left(2\pi \mathbf{w}^T\mathbf{w}\right)\left(4\pi\right)^2 \mathbf{w}\mathbf{w}^T + \left(2 + \sin\left(2\pi \mathbf{w}^T\mathbf{w}\right) 4\pi\right)\mathbf{I}_{N\times N}. \tag{17}$$

c) Denote by $h_p\left(\mathbf{w}\right) = \log\left(1 + e^{-\mathbf{a}_p^T\mathbf{w}}\right) = \log\left(1 + e^{-\sum_{n=1}^{N} a_{pn} w_n}\right)$ one of the summands of g. Then using the chain rule twice the j^{th} partial can be written as

$$\frac{\partial}{\partial w_j} h_p\left(\mathbf{w}\right) = \frac{1}{1 + e^{-\mathbf{a}_p^T\mathbf{w}}} e^{-\mathbf{a}_p^T\mathbf{w}}\left(-a_{pj}\right). \tag{18}$$

Since $\frac{1}{1+e^{-\mathbf{a}_p^T\mathbf{w}}}e^{-\mathbf{a}_p^T\mathbf{w}} = \frac{1}{e^{\mathbf{a}_p^T\mathbf{w}}+e^{\mathbf{a}_p^T\mathbf{w}}e^{-\mathbf{a}_p^T\mathbf{w}}} = \frac{1}{1+e^{\mathbf{a}_p^T\mathbf{w}}}$ we can rewrite the above more compactly as

$$\frac{\partial}{\partial w_j}h_p\left(\mathbf{w}\right) = -\frac{a_{pj}}{1+e^{\mathbf{a}_p^T\mathbf{w}}}, \tag{19}$$

and summing over p gives

$$\frac{\partial}{\partial w_j}g\left(\mathbf{w}\right) = -\sum_{p=1}^{P}\frac{a_{pj}}{1+e^{\mathbf{a}_p^T\mathbf{w}}}. \tag{20}$$

The full gradient of g is then given by

$$\nabla g\left(\mathbf{w}\right) = -\sum_{p=1}^{P}\frac{\mathbf{a}_p}{1+e^{\mathbf{a}_p^T\mathbf{w}}}. \tag{21}$$

Computing the second partial derivatives from equation (20) we have

$$\frac{\partial^2}{\partial w_i \partial w_j}g\left(\mathbf{w}\right) = \sum_{p=1}^{P}\frac{e^{\mathbf{a}_p^T\mathbf{w}}}{\left(1+e^{\mathbf{a}_p^T\mathbf{w}}\right)^2}a_{pi}a_{pj}, \tag{22}$$

and so we may write the full Hessian compactly as

$$\nabla^2 g\left(\mathbf{w}\right) = \sum_{p=1}^{P}\frac{e^{\mathbf{a}_p^T\mathbf{w}}}{\left(1+e^{\mathbf{a}_p^T\mathbf{w}}\right)^2}\mathbf{a}_p\mathbf{a}_p^T. \tag{23}$$

Exercise 2.3

We verify $\mathbf{X}\mathbf{Y}^T = \sum_{p=1}^{P}\mathbf{x}_p\mathbf{y}_p^T$ by showing that the (i,j) th entry of $\mathbf{X}\mathbf{Y}^T$ and $\sum_{p=1}^{P}\mathbf{x}_p\mathbf{y}_p^T$ are indeed identical. First note that the (i,j) th entry of $\mathbf{X}\mathbf{Y}^T$ can be written as the inner product of the i th row of $\mathbf{X}$ and the j th column of $\mathbf{Y}^T$, or equivalently the j th row of $\mathbf{Y}$. Therefore we have

$$\left(\mathbf{X}\mathbf{Y}^T\right)_{i,j} = \sum_{p=1}^{P}x_{i,p}y_{j,p}. \tag{24}$$

Now note that the (i,j) th entry of $\sum_{p=1}^{P}\mathbf{x}_p\mathbf{y}_p^T$ is simply the sum of the the (i,j) th entries of each summand $\mathbf{x}_p\mathbf{y}_p^T$

$$\left(\sum_{p=1}^{P}\mathbf{x}_p\mathbf{y}_p^T\right)_{i,j} = \sum_{p=1}^{P}\left(\mathbf{x}_p\mathbf{y}_p^T\right)_{i,j}. \tag{25}$$

Since the i th row of $\mathbf{x}_p$ and the j th column of $\mathbf{y}_p^T$ each consists of only one entry ($x_{i,p}$ and $y_{j,p}$ respectively), the right hand side of (25) can be written as $\sum_{p=1}^{P} x_{i,p}y_{j,p}$, which is identical to the expression in (24).

Exercise 2.4

a) The first and second order derivatives of g can be easily calculated as $g'(w) = \frac{2we^{w^2}}{1+e^{w^2}}$ and $g''(w) = \frac{2e^{w^2}\left(1+2w^2+e^{w^2}\right)}{\left(1+e^{w^2}\right)^2}$, respectively. The first order Taylor series approximation to g (around the point v) is then given as

$$h(w) = g(v) + g'(v)(w-v)$$

$$= \log(1+e^{v^2}) + \frac{2ve^{v^2}}{1+e^{v^2}}(w-v). \tag{26}$$

The second order Taylor series approximation to g (around the point v) is calculated as

$$h(w) = g(v) + g'(v)(w-v) + \frac{1}{2}g''(v)(w-v)^2$$

$$= \log(1+e^{v^2}) + \frac{2ve^{v^2}}{1+e^{v^2}}(w-v) + \frac{e^{v^2}\left(1+2v^2+e^{v^2}\right)}{\left(1+e^{v^2}\right)^2}(w-v)^2. \tag{27}$$

b) Assuming the matrix $\mathbf{Q}$ is symmetric, the gradient and Hessian of g can be found as $\nabla g(\mathbf{w}) = \frac{1}{2}\left(\mathbf{Q}+\mathbf{Q}^T\right)\mathbf{w} + \mathbf{r} = \mathbf{Q}\mathbf{w} + \mathbf{r}$ and $\nabla^2 g(\mathbf{w}) = \frac{1}{2}\left(\mathbf{Q}+\mathbf{Q}^T\right) = \mathbf{Q}$, respectively. The first order Taylor series approximation to g (around the point $\mathbf{v}$) is then given as

$$h(\mathbf{w}) = g(\mathbf{v}) + [\nabla g(\mathbf{v})]^T(\mathbf{w}-\mathbf{v})$$

$$= \frac{1}{2}\mathbf{v}^T\mathbf{Q}\mathbf{v} + \mathbf{r}^T\mathbf{v} + d + (\mathbf{Q}\mathbf{v}+\mathbf{r})^T(\mathbf{w}-\mathbf{v}). \tag{28}$$

The second order Taylor series approximation to g (around the point $\mathbf{v}$) is given by

$$h(\mathbf{w}) = g(\mathbf{v}) + [\nabla g(\mathbf{v})]^T(\mathbf{w}-\mathbf{v}) + \frac{1}{2}(\mathbf{w}-\mathbf{v})^T\nabla^2 g(\mathbf{v})(\mathbf{w}-\mathbf{v})$$

$$= \frac{1}{2}\mathbf{v}^T\mathbf{Q}\mathbf{v} + \mathbf{r}^T\mathbf{v} + d + (\mathbf{Q}\mathbf{v}+\mathbf{r})^T(\mathbf{w}-\mathbf{v}) + \frac{1}{2}(\mathbf{w}-\mathbf{v})^T\mathbf{Q}(\mathbf{w}-\mathbf{v})$$

$$= \frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w} + \mathbf{r}^T\mathbf{w} + d = g(\mathbf{w}). \tag{29}$$

The fact that the second order Taylor series (or quadratic) approximation to g is g itself makes sense, considering that g is a *quadratic* function!

Exercise 2.5

Let $\mathbf{n}^T \begin{bmatrix} h & w_1 & w_2 & \cdots & w_N \end{bmatrix}^T + \gamma = 0$ be the equation for the tangent hyperplane, with normal vector $\mathbf{n}$ and bias γ. From Equation (2.3) in the book we know that $h(\mathbf{w}) = g(\mathbf{v}) + \nabla g(\mathbf{v})^T(\mathbf{w} - \mathbf{v})$. A simple rearrangement then gives $h(\mathbf{w}) - \nabla g(\mathbf{v})^T\mathbf{w} + \nabla g(\mathbf{v})^T - g(\mathbf{v}) = 0$. By inspection it is now easy to see that $\mathbf{n} = \begin{bmatrix} 1 \\ -\nabla g(\mathbf{v}) \end{bmatrix}$ and $\gamma = \nabla g(\mathbf{v})^T - g(\mathbf{v})$.

Exercise 2.6

a) Setting the derivative of $g(w)$ to zero gives $g'(w) = \log(w) - \log(1-w) = 0$. Using the property of log, that $\log(a) - \log(b) = \log\left(\frac{a}{b}\right)$ this can be written equivalently as $\log\left(\frac{w}{1-w}\right) = 0$, and exponentiating each side this is equivalently $\frac{w}{1-w} = e^0 = 1$. Rearranging this is $w = 1 - w$, or equivalently $w = \frac{1}{2}$.

b) Setting the gradient of $g(\mathbf{w})$ to $\mathbf{0}_{2\times 1}$ gives $\nabla g(\mathbf{w}) = \mathbf{Q}\mathbf{w} + \mathbf{r} = \mathbf{0}_{2\times 1}$. Solving for $\mathbf{w}$ then gives $\mathbf{w} = -\mathbf{Q}^{-1}\mathbf{r}$ which, with the given values for $\mathbf{Q}$ and $\mathbf{r}$, has a numerical value of $\mathbf{w} = \begin{bmatrix} -0.57 \\ -0.57 \end{bmatrix}$.

Exercise 2.7

a) $g(w) = w^2$
$g''(w) = 2 > 0$

b) $g(w) = e^{w^2}$
$g''(w) = \left((2w)^2 + 2\right)e^{w^2}$. Note that both $(2w)^2 + 2$ and e^{w^2} are always positive, and so too is their product.

c) $g(w) = \log(1 + e^w)$
$g''(w) = \frac{e^w}{(1+e^w)^2}$. Note that both the numerator and denominator are positive, and so too is the whole fraction.

d) $g(w) = -\log(w)$

$g''(w) = \frac{1}{w^2} > 0$

Exercise 2.8

a) $g(w) = w \tanh(w) = w\frac{1-e^{-w}}{1+e^{-w}}$

Setting $g'(w) = \frac{1-e^{-w}+2we^{-w}}{1+e^{-w}}$ to zero gives $1 - e^{-w} + 2we^{-w} = 0$ or equivalently $e^w = 1 - 2w$. Figure 1 shows that functions e^w (in black) and $1 - 2w$ (in blue) only intersect at $w = 0$.

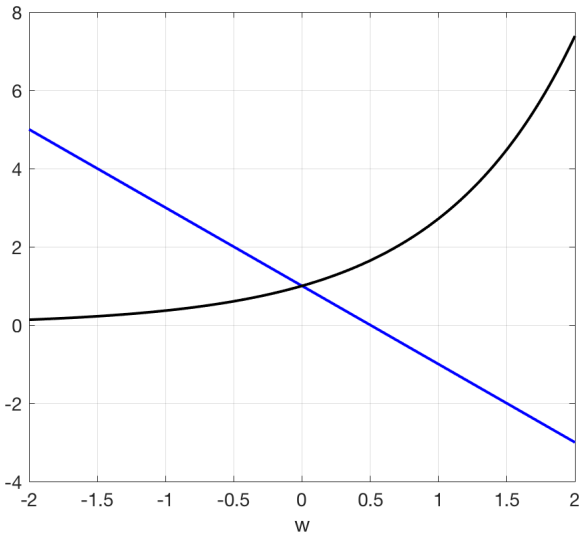

Figure 1. $w = 0$ is the only solution to $g'(w) = 0$.

Plotting $g(w)$ in Figure 2 we see that while being non-convex, its only stationary point is a global minimum at $w = 0$.

b) $g''(w) = \frac{2e^{-w}}{(1+e^{-w})^2}(2 + e^{-w} - w)$. Note that $\frac{2e^{-w}}{(1+e^{-w})^2}$ is always positive while $2 + e^{-w} - w$ will be negative for large enough values of w. Therefore the second derivative can take on negative values and g cannot be convex.

Exercise 2.9

a) Suppose first that all of the eigenvalues $d_n \geq 0$. Then the curvature function is $\psi(\mathbf{z}) = \mathbf{z}^T\mathbf{Q}\mathbf{z} = \mathbf{z}^T\left(\sum_{n=1}^{N} \mathbf{e}_n\mathbf{e}_n^T d_n\right)\mathbf{z} = \sum_{n=1}^{N}\left(\mathbf{e}_n^T\mathbf{z}\right)^2 d_n$. Since each $\left(\mathbf{e}_n^T\mathbf{z}\right)^2 \geq 0$ together with the assumption that $d_n \geq 0$, we have that $\psi(\mathbf{z}) \geq 0$ and therefore that $\mathbf{Q}$ is positive semidfinite.

b) Now suppose that $\mathbf{Q}$ is positive semidefinite. Then we know that the corresponding curvature function $\psi(\mathbf{z}) = \mathbf{z}^T\mathbf{Q}\mathbf{z} \geq 0$, and using the eigenfactorization of $\mathbf{Q}$ this is equiv-

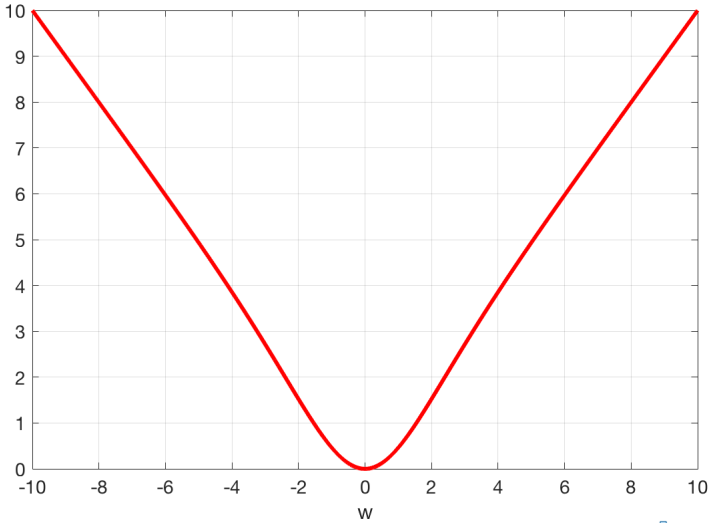

Figure 2. $g(w)$ is nonconvex with no stationary points but a global minimum at $w = 0$.

alently $\psi(\mathbf{z}) = \mathbf{z}^T\mathbf{Q}\mathbf{z} = \mathbf{z}^T\left(\sum_{n=1}^{N}\mathbf{e}_n\mathbf{e}_n^T d_n\right)\mathbf{z} = \sum_{n=1}^{N}\left(\mathbf{e}_n^T\mathbf{z}\right)^2 d_n \geq 0$. Since this sum is *always* nonnegative, so too then must be the eigenvalues $d_n \geq 0$ for all $n = 1...N$. If this were not the case, say one $d_j < 0$, then since the eigenvectors are orthogonal by setting $\mathbf{z} = \mathbf{e}_j$ the curavture function reduces to $\psi(\mathbf{z}) = \sum_{n=1}^{N}\left(\mathbf{e}_n^T\mathbf{e}_j\right)^2 d_n = d_j < 0$ which would contradict our assumption that $\mathbf{Q}$ was positive semidefinite.

c) $\mathbf{Q} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ has two eigenvalues, $d_1 = 0$ and $d_2 = 2$, both nonnegative. Therefore $\mathbf{Q}$ is positive semidefinite and g is convex.

d) Let $\psi(\mathbf{z})$ be the curvature function of the sum

$$\mathbf{Q} + \lambda\mathbf{I}_{N\times N} = \mathbf{E}\mathbf{D}\mathbf{E}^T + \lambda\mathbf{E}\mathbf{E}^T = \sum_{n=1}^{N}\mathbf{e}_n\mathbf{e}_n^T d_n + \sum_{n=1}^{N}\mathbf{e}_n\mathbf{e}_n^T\lambda = \sum_{n=1}^{N}\mathbf{e}_n\mathbf{e}_n^T\left(d_n + \lambda\right). \tag{30}$$

Therefore we have $\psi(\mathbf{z}) = \mathbf{z}^T\left(\sum_{n=1}^{N}\mathbf{e}_n\mathbf{e}_n^T\left(d_n + \lambda\right)\right)\mathbf{z} = \sum_{n=1}^{N}\left(\mathbf{e}_n^T\mathbf{z}\right)^2\left(d_n + \lambda\right)$. Now, since each $\left(\mathbf{e}_n^T\mathbf{z}\right)^2 \geq 0$ if we set λ large enough so that $d_n + \lambda \geq 0$ for all n then so too must the sum be nonnegative. Hence by setting λ to the absolute value of the smallest eigenvalue of $\mathbf{Q}$, the sum $\mathbf{Q} + \lambda\mathbf{I}_{N\times N}$ can be made semipositive definite.

Exercise 2.10

a) The curvature function for $\mathbf{x}\mathbf{x}^T$ is given by $\psi(\mathbf{z}) = \mathbf{z}^T\mathbf{x}\mathbf{x}^T\mathbf{z} = \left(\mathbf{x}^T\mathbf{z}\right)^2 \geq 0$, which is always nonnegative. Therefore all eignevalues of $\mathbf{x}\mathbf{x}^T$ must be nonnegative.

b) The curvature function for $\sum_{p=1}^{P} \delta_p \mathbf{x}_p \mathbf{x}_p^T$ is given by

$$\psi(\mathbf{z}) = \mathbf{z}^T \left(\sum_{p=1}^{P} \delta_p \mathbf{x}_p \mathbf{x}_p^T \right) \mathbf{z} = \sum_{p=1}^{P} \delta_p \left(\mathbf{x}_p^T \mathbf{z} \right)^2, \tag{31}$$

which is always nonnegative if $\delta_p \geq 0$ for all p.

c) In part b), we showed that $\sum_{p=1}^{P} \delta_p \mathbf{x}_p \mathbf{x}_p^T$ has all nonnegative eignevalues. Now note that adding $\lambda \mathbf{I}_{N\times N}$ to $\sum_{p=1}^{P} \delta_p \mathbf{x}_p \mathbf{x}_p^T$ shifts each of its eigenvalues to the right by λ. Therefore all eignevalues of the sum $\sum_{p=1}^{P} \delta_p \mathbf{x}_p \mathbf{x}_p^T + \lambda \mathbf{I}_{N\times N}$ are greater than or equal to $\lambda > 0$, and hence positive.

Exercise 2.11

a) This is the straightforward result of Exercise 2.9.

b) $\nabla^2 g(\mathbf{w}) = \frac{1}{2}\left(\mathbf{Q}^T + \mathbf{Q}\right) = \mathbf{Q}$. Therefore regardless of the values for $\mathbf{r}$ and d, g always defines a convex function as long as the eigenvalues of $\mathbf{Q}$ are all nonnegative.

c) The Hessian of $g(\mathbf{w})$ can be calculated as $\nabla^2 g(\mathbf{w}) = 4\pi^2 \cos\left(2\pi \mathbf{w}^T\mathbf{w}\right) \mathbf{w}\mathbf{w}^T$, with the curvature function given as $\psi(\mathbf{z}) = \mathbf{z}^T\left(\nabla^2 g(\mathbf{w})\right)\mathbf{z} = \mathbf{z}^T\left(4\pi^2 \cos\left(2\pi \mathbf{w}^T\mathbf{w}\right)\mathbf{w}\mathbf{w}^T\right)\mathbf{z} = 4\pi^2 \cos\left(2\pi \mathbf{w}^T\mathbf{w}\right)\left(\mathbf{w}^T\mathbf{z}\right)^2$. Note that when $\cos\left(2\pi \mathbf{w}^T\mathbf{w}\right) < 0$, e.g., with $\mathbf{w} = \begin{bmatrix} 1/2 & 0 \end{bmatrix}^T$, $\psi(\mathbf{z})$ can take on negative values and therefore g cannot be convex.

Exercise 2.12

This is a programming exercise. See our code repository.

Exercise 2.13

This is a programming exercise. See our code repository.

Exercise 2.14

This is a programming exercise. See our code repository.

Exercise 2.15

Denoting the deisred length as ℓ, using the Pythagorean Theorem and the given Figure we can say that $\ell^2 = \left\|\mathbf{w}^{k-1} - \mathbf{w}^k\right\|_2^2 + \left(l\left(\mathbf{w}^{k-1}\right) - l\left(\mathbf{w}^k\right)\right)^2$.
Next, noting that $l\left(\mathbf{w}^{k-1}\right) = g\left(\mathbf{w}^{k-1}\right)$, that by definition

$$l\left(\mathbf{w}^k\right) = g\left(\mathbf{w}^{k-1}\right) + \nabla g\left(\mathbf{w}^{k-1}\right)^T\left(\mathbf{w}^k - \mathbf{w}^{k-1}\right), \tag{32}$$

and that we know $\left\|\mathbf{w}^{k-1} - \mathbf{w}^k\right\|_2 = \alpha\left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2$ all together we may rewrite this equivalently as

$$\ell^2 = \alpha^2\left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2^2 + \left(\nabla g\left(\mathbf{w}^{k-1}\right)^T\left(\mathbf{w}^k - \mathbf{w}^{k-1}\right)\right)^2. \tag{33}$$

Using the definition of $\mathbf{w}^k = \mathbf{w}^{k-1} - \alpha\nabla g\left(\mathbf{w}^{k-1}\right)$ this can be simplified as $\ell^2 = \alpha^2\left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2^2 + \alpha^2\left(\left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2^2\right)^2$ or equivalently

$$\ell^2 = \alpha^2\left(1 + \left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2^2\right)\left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2^2, \tag{34}$$

and taking the square root of both sides gives the desired solution.

Exercise 2.16

This is a programming exercise. See our code repository.

Exercise 2.17

a) Setting the gradient of $g\left(\mathbf{w}\right) = \log\left(1 + e^{\mathbf{w}^T\mathbf{w}}\right)$ to zero, using the chain rule, gives

$$\nabla g\left(\mathbf{w}\right) = \frac{2e^{\mathbf{w}^T\mathbf{w}}}{1 + e^{\mathbf{w}^T\mathbf{w}}}\mathbf{w} = \mathbf{0}_{N\times 1}. \tag{35}$$

Since the scalar weight $\frac{2e^{\mathbf{w}^T\mathbf{w}}}{1+e^{\mathbf{w}^T\mathbf{w}}} \geq 1$ the only way the equality can occur is when $\mathbf{w} = \mathbf{0}_{N\times 1}$.

b) The Hessian of g, using the chain rule and product rule, may be calculated as

$$\nabla^2 g\left(\mathbf{w}\right) = \frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1 + e^{\mathbf{w}^T\mathbf{w}}\right)^2}\mathbf{w}\mathbf{w}^T + \frac{2e^{\mathbf{w}^T\mathbf{w}}}{1 + e^{\mathbf{w}^T\mathbf{w}}}\mathbf{I}_{N\times N}. \tag{36}$$

Fixing $\mathbf{w}$ for the moment, for any N length column vector $\mathbf{z}$ we have that (where $\mathbf{I}_{N\times N}$ is a $N \times N$ identity)

$$\mathbf{z}^T\nabla^2 g\left(\mathbf{w}\right)\mathbf{z} = \mathbf{z}^T\left(\frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1+e^{\mathbf{w}^T\mathbf{w}}\right)^2}\mathbf{w}\mathbf{w}^T + \frac{2e^{\mathbf{w}^T\mathbf{w}}}{1+e^{\mathbf{w}^T\mathbf{w}}}\mathbf{I}_{N\times N}\right)\mathbf{z}$$

$$= \mathbf{z}^T\nabla^2 g\left(\mathbf{w}\right)\mathbf{z} = \mathbf{z}^T\frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1+e^{\mathbf{w}^T\mathbf{w}}\right)^2}\mathbf{w}\mathbf{w}^T\mathbf{z} + \mathbf{z}^T\frac{2e^{\mathbf{w}^T\mathbf{w}}}{1+e^{\mathbf{w}^T\mathbf{w}}}\mathbf{I}_{N\times N}\mathbf{z}$$

$$= \frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1+e^{\mathbf{w}^T\mathbf{w}}\right)^2}\mathbf{z}^T\mathbf{w}\mathbf{w}^T\mathbf{z} + \frac{2e^{\mathbf{w}^T\mathbf{w}}}{1+e^{\mathbf{w}^T\mathbf{w}}}\mathbf{z}^T\mathbf{z}$$

$$= \frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1+e^{\mathbf{w}^T\mathbf{w}}\right)^2}\left(\mathbf{z}^T\mathbf{w}\right)^2 + \frac{2e^{\mathbf{w}^T\mathbf{w}}}{1+e^{\mathbf{w}^T\mathbf{w}}}\left\|\mathbf{z}\right\|_2^2. \tag{37}$$

Since each component of this expression is nonnegative regardless of the $\mathbf{z}$ chosen, i.e., $\frac{4e^{\mathbf{w}^T\mathbf{w}}}{\left(1+e^{\mathbf{w}^T\mathbf{w}}\right)^2} \geq 0$, $\left(\mathbf{z}^T\mathbf{w}\right)^2 \geq 0$, etc., so too is the sum always nonnegative, regardless of the $\mathbf{z}$ chosen. However this also holds regardless of which $\mathbf{w}$ is chosen - hence we have that

$$\mathbf{z}^T\nabla^2 g\left(\mathbf{w}\right)\mathbf{z} \geq 0, \tag{38}$$

for all $\mathbf{w}$ and $\mathbf{z}$, and therefore (by the second order definition of convexity) g is indeed convex.

c) See our code repository.

d) See our code repository.

Chapter 3. Regression

Exercise 3.1

This is a programming exercise.

Solving the Least Squares problem for the student debt data gives the trend line as $y = b + wx$, where x is year, y is the amount of debt in trillions of dollars, and optimal parameters b and w are found respectively as $b = -160.7$ and $w = .08$ (see our code repository). Therefore in year 2050 ($x_{new} = 2050$), the student debt will reach $y_{new} = -160.7 + .08(2050) \approx 3.9$ trillion dollars if this trend continues.

Exercise 3.2

a) See our code repository.

b) Running the code in part a) returns the optimal parameter values $w_0 = 6.81$ and $w_1 = 0.65$. Exponentiating both sides of Equation (3.30) in the book, we have $y \approx e^{w_0} x^{w_1}$. Plugging the optimal parameter values for $w_0 = 6.81$ and $w_1 = 0.65$ found in part a), the nonlinear relationship between x and y can be approximated as $y \approx 907\, x^{0.65}$.

c) With $x = 10$ kg, we have that $y \approx 4051$ kJ/day. So an animal weighing 10 kg will require $\frac{4051}{4.18} = 969$ kcal per day.

Exercise 3.3

a) Noting that $\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_p$ we can write

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} - y_p\right)^2 = \sum_{p=1}^{P} \left\{ \left(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_p - y_p\right) \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} - y_p\right) \right\}$$

$$= \sum_{p=1}^{P} \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} - \sum_{p=1}^{P} \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_p y_p - \sum_{p=1}^{P} y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} + \sum_{p=1}^{P} y_p^2$$

$$= \frac{1}{2} \tilde{\mathbf{w}}^T \left(2 \sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T \right) \tilde{\mathbf{w}} - \left(2 \sum_{p=1}^{P} y_p \tilde{\mathbf{x}}_p \right)^T \tilde{\mathbf{w}} + \sum_{p=1}^{P} y_p^2. \tag{39}$$

Hence $\mathbf{Q} = 2 \sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T$, $\mathbf{r} = -2 \sum_{p=1}^{P} y_p \tilde{\mathbf{x}}_p$ and $d = \sum_{p=1}^{P} y_p^2$.

b) $\psi\left(\mathbf{z}\right) = \mathbf{z}^T \mathbf{Q} \mathbf{z} = \mathbf{z}^T \left(2 \sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T \right) \mathbf{z} = \sum_{p=1}^{P} 2 \left(\tilde{\mathbf{x}}_p^T \mathbf{z} \right)^2 \geq 0$. So $\mathbf{Q}$ is positive semidefinite with all nonnegative eigenvalues.

c) Hessian of g can be easily calculated as $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \frac{1}{2}\left(\mathbf{Q} + \mathbf{Q}^T\right)$. Now since $\mathbf{Q}^T = \mathbf{Q}$ (see part a), we can simply write $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \frac{1}{2}\left(\mathbf{Q} + \mathbf{Q}\right) = \mathbf{Q}$. Now because $\mathbf{Q}$ has all nonnegative eigenvalues as shown in part b), the function g is convex according to the second order definition of convexity.

d) Knowing the form of gradient $\nabla g\left(\tilde{\mathbf{w}}\right) = \mathbf{Q}\tilde{\mathbf{w}} + \mathbf{r}$ and Hessian $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \mathbf{Q}$, and starting at an arbitrarily chosen point $\mathbf{v}$, the first Newton step is the solution to the following

linear system of equations

$$\left[\nabla^2 g\left(\mathbf{v}\right)\right]\tilde{\mathbf{w}} = \left[\nabla^2 g\left(\mathbf{v}\right)\right]\mathbf{v} - \nabla g\left(\mathbf{v}\right). \tag{40}$$

Plugging $\nabla^2 g\left(\mathbf{v}\right) = \mathbf{Q} = 2\sum_{p=1}^{P}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T$ and $\nabla g\left(\mathbf{v}\right) = \mathbf{Q}\mathbf{v} + \mathbf{r} =$ into the equation above, we have

$$\mathbf{Q}\tilde{\mathbf{w}} = \mathbf{Q}\mathbf{v} - \left(\mathbf{Q}\mathbf{v} + \mathbf{r}\right) = -\mathbf{r}. \tag{41}$$

Plugging again $\mathbf{Q} = 2\sum_{p=1}^{P}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T$ and $\mathbf{r} = -2\sum_{p=1}^{P} y_p\tilde{\mathbf{x}}_p$ from part a) into this equation gives

$$\left(\sum_{p=1}^{P}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T\right)\tilde{\mathbf{w}} = \sum_{p=1}^{P} y_p\tilde{\mathbf{x}}_p. \tag{42}$$

Exercise 3.4

This is a programming exercise. See our code repository.

Exercise 3.5

a) Using the compact notation introduced in (3.16) we can write the cost function g as

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P}\left(\tilde{\mathbf{f}}_p^T\tilde{\mathbf{w}} - y_p\right)^2. \tag{43}$$

Calculating the gradient of g as $\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\tilde{\mathbf{f}}_p^T\tilde{\mathbf{w}} - y_p\right)\tilde{\mathbf{f}}_p$, the first order system $\nabla g\left(\tilde{\mathbf{w}}\right) = \mathbf{0}_{2\times 1}$ then can be written as

$$\sum_{p=1}^{P}\left(\tilde{\mathbf{f}}_p^T\tilde{\mathbf{w}} - y_p\right)\tilde{\mathbf{f}}_p = \mathbf{0}_{2\times 1}, \tag{44}$$

or equivalently

$$\left(\sum_{p=1}^{P}\tilde{\mathbf{f}}_p\tilde{\mathbf{f}}_p^T\right)\tilde{\mathbf{w}} = \sum_{p=1}^{P} y_p\tilde{\mathbf{f}}_p. \tag{45}$$

b) This is a programming exercise. See the code repository.

Exercise 3.6

a) From elementary physics or just by inspecting the data, one can propose the follwing relationship between the input x and output y

$$y = w\sin\left(\frac{\pi x}{180}\right). \tag{46}$$

b) See our code repository.

Exercise 3.7

a) $y = e^{b+wx}$. Note that taking log of both sides gives $\log y = b + wx$, which is linear in both weights b and w.

b) See our code repository.

Exercise 3.8

a) Denoting by y and x the current passing through the circuit and the length of the added wire, respectively, a *potential* model could take the form

$$y = \frac{1}{b + xw}, \tag{47}$$

where b and w are the model parameters to learn. Note that if we did not include b as a parameter, the current would go to infinity at $x = 0$!
If we tranform the output y by taking its inverse $\frac{1}{y}$, we have

$$\frac{1}{y} = b + xw. \tag{48}$$

b) The Least Squares cost function for the Ohm's dataset associated with (48) then takes the form

$$g\left(b, w\right) = \sum_{p=1}^{5}\left(b + x_p w - \frac{1}{y_p}\right)^2, \tag{49}$$

which can be written more compactly as

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{5}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}} - \frac{1}{y_p}\right)^2, \tag{50}$$

introducing the familiar notation

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_p \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ w \end{bmatrix}. \tag{51}$$

Now setting the gradient of g to zero, we have

$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{5}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}} - \frac{1}{y_p}\right)\tilde{\mathbf{x}}_p = \mathbf{0}_{2\times 1}. \tag{52}$$

This system can be solved for optimal parameters using a linear solver or algebraically as $\tilde{\mathbf{w}} = \left(\sum_{p=1}^{5}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T\right)^{-1}\left(\sum_{p=1}^{5}\frac{1}{y_p}\tilde{\mathbf{x}}_p\right)$.

c) See our code repository.

Exercise 3.9

a) By simple inspection we find that $\mathbf{f}_p = \begin{bmatrix} x_{1,p} \\ x_{2,p} \end{bmatrix}$ and $y_p = 1$ for all p.

b) See our code repository.

Exercise 3.10

a) $\sigma^{-1}\left(\sigma\left(t\right)\right) = \log\left(\frac{\sigma(t)}{1-\sigma(t)}\right) = \log\left(\frac{\frac{1}{1+e^{-t}}}{1-\frac{1}{1+e^{-t}}}\right) = \log\left(\frac{\frac{1}{1+e^{-t}}}{\frac{e^{-t}}{1+e^{-t}}}\right) = \log\left(\frac{1}{e^{-t}}\right) = t.$

b) From (3.23) we have $\sigma\left(b + x_p w\right) \approx y_p$ for all p. Taking the *sigmoid-inverse* from both sides then gives $\sigma^{-1}\left(\sigma\left(b + x_p w\right)\right) \approx \sigma^{-1}\left(y_p\right)$, which simplifies to $b + x_p w \approx \log\left(\frac{y_p}{1-y_p}\right)$.

c) Similar to part b) of Exercise 3.8 we can form the Least Squares cost function

$$g\left(b, w\right) = \sum_{p=1}^{P}\left(b + x_p w - \log\left(\frac{y_p}{1-y_p}\right)\right)^2, \tag{53}$$

which can be written more compactly as

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}} - \log\left(\frac{y_p}{1-y_p}\right)\right)^2, \tag{54}$$

introducing the familiar notation

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_p \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ w \end{bmatrix}. \tag{55}$$

Setting the gradient of g to zero, we have

$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}} - \log\left(\frac{y_p}{1-y_p}\right)\right)\tilde{\mathbf{x}}_p = \mathbf{0}_{2\times 1}. \tag{56}$$

This system can be solved for *optimal* parameters using a linear solver or algebraically as $\tilde{\mathbf{w}} = \left(\sum_{p=1}^{P}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T\right)^{-1}\left(\sum_{p=1}^{P}\log\left(\frac{y_p}{1-y_p}\right)\tilde{\mathbf{x}}_p\right)$. See our repository for the code.

Exercise 3.11

a) Using the compact notation the cost function can be written as

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)^2. \tag{57}$$

Now, the partial derivative with respect to the i th entry in $\tilde{\mathbf{w}}$ can be computed as

$$\begin{aligned}\frac{\partial}{\partial\tilde{w}_i}g\left(\tilde{\mathbf{w}}\right) &= 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\frac{\partial}{\partial\tilde{w}_i}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right) \\ &= 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\sigma'\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\frac{\partial}{\partial\tilde{w}_i}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) \\ &= 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\left(1-\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\right)\tilde{x}_{p,i}.\end{aligned} \tag{58}$$

The full gradient of g is then given by

$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\left(1-\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\right)\tilde{\mathbf{x}}_p. \tag{59}$$

b) See our code repository.

Exercise 3.12

a) See our code repository.

b) To find the stationary points of g we must take its gradient and set it equal to zero. The resulting system of equations however **cannot be solved analytically** and hence an iterative method (e.g., gradient descent) must be employed. The cost function in this case is **nonconvex** (see Figure) which requires us to run gradient descent multiple times with different starting points, and take the weight vector resulting in the smallest objective value as the optimal weight vector.

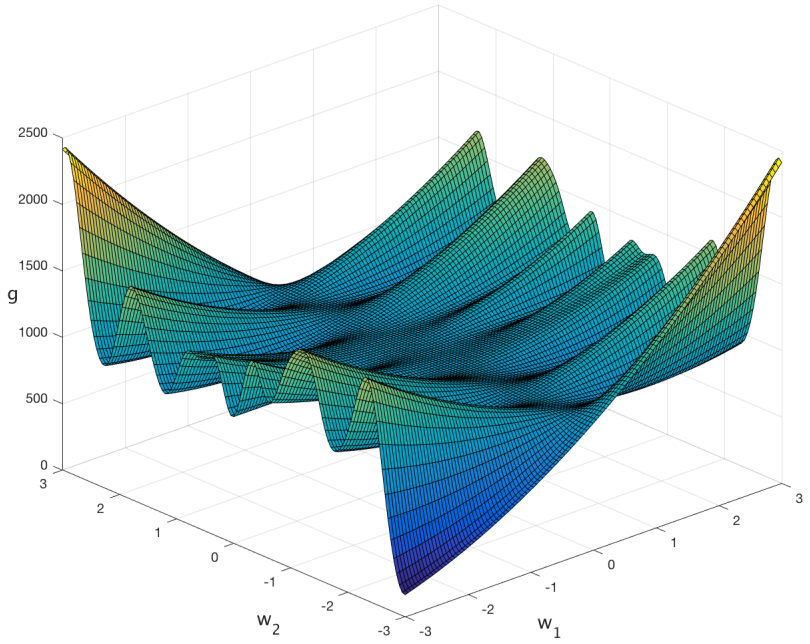

Figure 3. Exercise 3.12

Exercise 3.13

a) Using the compact notation

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_p \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}, \tag{60}$$

the regularized cost function can be written as $g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \left(\sigma\left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) - y_p\right)^2 + \lambda \mathbf{w}^T \mathbf{w}$. The gradient of g with respect to $\tilde{\mathbf{w}}$ can then be written as

$$\nabla g\left(\tilde{\mathbf{w}}\right) = \nabla_{\tilde{\mathbf{w}}} \left(\sum_{p=1}^{P} \left(\sigma\left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) - y_p\right)^2 \right) + \nabla_{\tilde{\mathbf{w}}} \left(\lambda \mathbf{w}^T \mathbf{w}\right). \tag{61}$$

The first part, as shown in Exercise 3.11, can be written as

$$\nabla_{\tilde{\mathbf{w}}} \left(\sum_{p=1}^{P} \left(\sigma \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} \right) - y_p \right)^2 \right)$$

$$= 2 \sum_{p=1}^{P} \left(\sigma \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} \right) - y_p \right) \sigma \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} \right) \left(1 - \sigma \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} \right) \right) \tilde{\mathbf{x}}_p, \tag{62}$$

and the second part as

$$\nabla_{\tilde{\mathbf{w}}} \left(\lambda \mathbf{w}^T \mathbf{w} \right) = \begin{bmatrix} \frac{\partial}{\partial b} \left(\lambda \mathbf{w}^T \mathbf{w} \right) \\ \nabla_{\mathbf{w}} \left(\lambda \mathbf{w}^T \mathbf{w} \right) \end{bmatrix} = \begin{bmatrix} 0 \\ \nabla_{\mathbf{w}} \left(\lambda \mathbf{w}^T \mathbf{w} \right) \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 2\lambda \mathbf{w} \end{bmatrix} = 2\lambda \begin{bmatrix} 0 \\ \mathbf{w} \end{bmatrix}. \tag{63}$$

b) See our code repository.

Exercise 3.14

a) Writing $\frac{\lambda}{2} \left\| \mathbf{w} - \mathbf{w}^{k-1} \right\|_2^2$ equivalently as $\frac{1}{2} \left(\mathbf{w} - \mathbf{w}^{k-1} \right)^T \left(\lambda \mathbf{I}_{N \times N} \right) \left(\mathbf{w} - \mathbf{w}^{k-1} \right)$, h can be written as

how that the first order condition for optimality leads to the following adjusted Newton's system for a stationary point of the above quadratic

$$\left[\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right] \mathbf{w} = \left[\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right] \mathbf{w}^{k-1} - \nabla g \left(\mathbf{w}^{k-1} \right). \tag{64}$$

It is easy to verify that h takes the quadratic form $h(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{Q} \mathbf{w} + \mathbf{r}^T \mathbf{w} + d$ where

$$\mathbf{Q} = \nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N}, \tag{65}$$

$$\mathbf{r} = \nabla g \left(\mathbf{w}^{k-1} \right) - \left(\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right) \mathbf{w}^{k-1}, \tag{66}$$

and

$$d = g \left(\mathbf{w}^{k-1} \right) - \nabla g \left(\mathbf{w}^{k-1} \right)^T \mathbf{w}^{k-1} + \frac{1}{2} \left(\mathbf{w}^{k-1} \right)^T \left[\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right] \mathbf{w}^{k-1}. \tag{67}$$

Setting the gradient of h to zero, $\nabla h(\mathbf{w}) = \mathbf{Q}\mathbf{w} + \mathbf{r} = \mathbf{0}$, we have $\mathbf{Q}\mathbf{w} = -\mathbf{r}$. Substittuting corresponding values for $\mathbf{Q}$ and $\mathbf{r}$ from above gives the desired system

$$\left[\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right] \mathbf{w} = \left[\nabla^2 g \left(\mathbf{w}^{k-1} \right) + \lambda \mathbf{I}_{N \times N} \right] \mathbf{w}^{k-1} - \nabla g \left(\mathbf{w}^{k-1} \right). \tag{68}$$

b) Denoting the set of eigenvalues of $\nabla^2 g\left(\mathbf{w}^{k-1}\right)$ by $\{\delta_n\}_{n=1}^N$, we showed in Exercise 2.9 that the set of eigenvalues for $\nabla^2 g\left(\mathbf{w}^{k-1}\right) + \lambda \mathbf{I}_{N\times N}$ is given as $\{\delta_n + \lambda\}_{n=1}^N$. Therefore by setting λ to any value larger than $\min\{0,\, \delta_1,\, \delta_2,\, \ldots,\, \delta_N\}$, we can gurantee all eigenvalues of $\nabla^2 g\left(\mathbf{w}^{k-1}\right) + \lambda \mathbf{I}_{N\times N}$ are positive.

c) Following the notation defined in part a), the Hessian of h can be written as

$$\nabla^2 h\left(\mathbf{w}\right) = \frac{1}{2}\left(\mathbf{Q}^T + \mathbf{Q}\right) = \mathbf{Q} = \nabla^2 g\left(\mathbf{w}^{k-1}\right) + \lambda \mathbf{I}_{N\times N}. \tag{69}$$

Now, according to part b) setting λ to any value larger than $\min\{0,\, \delta_1,\, \delta_2,\, \ldots,\, \delta_N\}$ makes $\nabla^2 h\left(\mathbf{w}\right)$ positive definite, implying (based on the second order definition of convexity) that h is indeed convex.

Chapter 4. Classification

Exercise 4.1

a) First note that using the compact notation

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_p \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}, \tag{70}$$

and defining the constant vector $\mathbf{c} = -y_p \tilde{\mathbf{x}}_p$, we can write each summand as $h\left(\tilde{\mathbf{w}}\right) = \max\left(0,\, \mathbf{c}^T \tilde{\mathbf{w}}\right)$. According to the zeroth order definition of convexity, h is convex if

$$h\left(\lambda \tilde{\mathbf{w}}_1 + (1-\lambda)\, \tilde{\mathbf{w}}_2\right) \leq \lambda h\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)\, h\left(\tilde{\mathbf{w}}_2\right), \tag{71}$$

for all $0 \leq \lambda \leq 1$. Using the definition of h we need to show that

$$\max\left(0,\, \mathbf{c}^T\left(\lambda \tilde{\mathbf{w}}_1 + (1-\lambda)\, \tilde{\mathbf{w}}_2\right)\right) \leq \lambda \max\left(0,\, \mathbf{c}^T \tilde{\mathbf{w}}_1\right) + (1-\lambda) \max\left(0,\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right). \tag{72}$$

or equivalently

$$\max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 + (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right) \leq \max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1\right) + \max\left(0,\, (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right), \tag{73}$$

using the fact that $\lambda \max\left(0,\, \zeta\right) = \max\left(0,\, \lambda\zeta\right)$ when λ is nonnegative. Now to see that (73) indeed holds, we break it down into four cases:

Case 1. $\lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 \geq 0$ and $(1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2 \geq 0$
In this case both sides of (73) simplify to $\lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 + (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2$ and we have equality.

Case 2. $\lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 < 0$ and $(1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2 < 0$
In this case both sides of (73) simplify to 0 and we have equality again.

Case 3. $\lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 \geq 0$ and $(1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2 < 0$
In this case we have

$$\max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 + (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right) \leq \max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1\right)$$

$$\leq \max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1\right) + \max\left(0,\, (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right). \tag{74}$$

Case 4. $\lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 < 0$ and $(1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2 \geq 0$
In this case we have

$$\max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1 + (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right) \leq \max\left(0,\, (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right)$$

$$\leq \max\left(0,\, \lambda \mathbf{c}^T \tilde{\mathbf{w}}_1\right) + \max\left(0,\, (1-\lambda)\, \mathbf{c}^T \tilde{\mathbf{w}}_2\right). \tag{75}$$

b) We show that

$$f\left(\lambda t + (1-\lambda)\, t\right) \leq \lambda f\left(t\right) + (1-\lambda)\, f\left(t\right), \tag{76}$$

where $f(t) = g(t) + h(t)$, and g and h are both convex. Using the definition of f as $f(t) = g(t) + h(t)$, the left hand side of above can be written as $g\left(\lambda t + (1-\lambda)\, t\right) + h\left(\lambda t + (1-\lambda)\, t\right)$ which is bounded from above by

$$\left[\lambda g\left(t\right) + (1-\lambda)\, g\left(t\right)\right] + \left[\lambda h\left(t\right) + (1-\lambda)\, h\left(t\right)\right], \tag{77}$$

using zeroth order definition of convexity for g and h. Rearranging the terms, (77) can be written as

$$\left[\lambda g\left(t\right) + \lambda h\left(t\right)\right] + \left[(1-\lambda)\, g\left(t\right) + (1-\lambda)\, h\left(t\right)\right] = \lambda f\left(t\right) + (1-\lambda)\, f\left(t\right). \tag{78}$$

Putting the results of part a) and part b) together, the perceptron cost is convex as it is the sum of P convex summands, one for each data point.

Exercise 4.2

The Hessian for the logistic regression cost g is given by

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T. \tag{79}$$

To prove g is convex, we use the second order definition of convexity and show $\nabla^2 g\left(\tilde{\mathbf{w}}\right)$ is positive semidefinite by forming

$$\psi\left(\mathbf{z}\right) = \mathbf{z}^T \nabla^2 g\left(\tilde{\mathbf{w}}\right) \mathbf{z} = \mathbf{z}^T \left[\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T\right] \mathbf{z}$$

$$= \sum_{p=1}^{P} \delta_p \left(\tilde{\mathbf{x}}_p^T \mathbf{z}\right)^2, \tag{80}$$

where $\delta_p = \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right)$. Now note that since $0 \leq \sigma\left(\zeta\right) \leq 1$ we have that $\delta_p \geq 0$, and therefore

$$\mathbf{z}^T \nabla^2 g\left(\tilde{\mathbf{w}}\right) \mathbf{z} = \sum_{p=1}^{P} \delta_p \left(\tilde{\mathbf{x}}_p^T \mathbf{z}\right)^2 \geq 0. \tag{81}$$

Exercise 4.3

a) The softmax cost is given by $g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \log\left(1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}\right)$. The partial derivative of the cost with respect to the i th entry in $\tilde{\mathbf{w}}$ can be computed as

$$\frac{\partial}{\partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial \tilde{w}_i}\left(1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}\right)}{1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}}$$

$$= \sum_{p=1}^{P} \frac{\left(e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}\right) \frac{\partial}{\partial \tilde{w}_i}\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)}{1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}} = -\sum_{p=1}^{P} \frac{e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}}{1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{x}_{p,i}. \tag{82}$$

The full gradient therefore can be written as

$$\nabla g = -\sum_{p=1}^{P} \frac{e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}}{1 + e^{-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{\mathbf{x}}_p. \tag{83}$$

Multiplying both the numerator and denominator of each summand by $e^{y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}$ gives

$$\nabla g = -\sum_{p=1}^{P} \frac{1}{1 + e^{y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{\mathbf{x}}_p = -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{\mathbf{x}}_p. \tag{84}$$

b) $\nabla g = -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{\mathbf{x}}_p$ can be written as $\tilde{\mathbf{X}}\mathbf{r}$, where

$$\tilde{\mathbf{X}} = \left[\begin{array}{cccc} \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_P \end{array}\right], \tag{85}$$

and

$$\mathbf{r} = -\overrightarrow{\sigma}\left(-\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{w}}\right) \odot \mathbf{y}\right) \odot \mathbf{y}. \tag{86}$$

In the equation above, $\odot$ denotes the Hadamard (entry-wise) product,

$$\overrightarrow{\sigma}\left(\left[\begin{array}{c} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_P \end{array}\right]\right) = \left[\begin{array}{c} \sigma(\zeta_1) \\ \sigma(\zeta_2) \\ \vdots \\ \sigma(\zeta_P) \end{array}\right], \tag{87}$$

and

$$\mathbf{y} = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_P \end{array}\right]. \tag{88}$$

c) See our code repository.

Exercise 4.4

a) We show in Exercise 4.3 that

$$\frac{\partial}{\partial \tilde{w}_i} g(\tilde{\mathbf{w}}) = -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{x}_{p,i}. \tag{89}$$

Taking the partial derivative of the above with respect to $\tilde{w}_i$ then gives the (i,j) th entry in $\nabla^2 g$, as

$$\frac{\partial^2}{\partial \tilde{w}_j \partial \tilde{w}_i} g(\tilde{\mathbf{w}}) = -\frac{\partial}{\partial \tilde{w}_j} \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{x}_{p,i}$$

$$= -\sum_{p=1}^{P} \frac{\partial}{\partial \tilde{w}_j}\left(\sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{x}_{p,i}\right) = -\sum_{p=1}^{P} \sigma'\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{x}_{p,i} \frac{\partial}{\partial \tilde{w}_j}\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)$$

$$= -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{x}_{p,i}\left(-y_p \tilde{x}_{p,j}\right)$$

$$= \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1-\sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{x}_{p,i} \tilde{x}_{p,j} y_p^2. \tag{90}$$

Now note that $y_p^2 = 1$ (since $y_p \in \{-1, +1\}$), and we can write the full Hessian as

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1-\sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T. \tag{91}$$

b) $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\left(1-\sigma\left(-y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T$ can be written as $\tilde{\mathbf{X}} \operatorname{diag}\left(\mathbf{r}\right) \tilde{\mathbf{X}}^T$, where

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_P \end{bmatrix}, \tag{92}$$

and

$$\mathbf{r} = \mathbf{s} \odot \left(\mathbf{1}_{P\times 1} - \mathbf{s}\right), \tag{93}$$

wherein $\odot$ denotes the Hadamard (entry-wise) product,

$$s = \overrightarrow{\sigma}\left(-\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{w}}\right) \odot \mathbf{y}\right), \tag{94}$$

$$\overrightarrow{\sigma}\left(\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_P \end{bmatrix}\right) = \begin{bmatrix} \sigma\left(\zeta_1\right) \\ \sigma\left(\zeta_2\right) \\ \vdots \\ \sigma\left(\zeta_P\right) \end{bmatrix}, \tag{95}$$

and

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{bmatrix}. \tag{96}$$

c) See our code repository.

Exercise 4.5

a) Suppose that the hyperplane $b + \mathbf{x}^T \mathbf{w} = 0$ perfectly separates the two classes of data. Thus for every $\mathbf{x}_p$ we have that $-y_p\left(b + \mathbf{x}_p^T \mathbf{w}\right) < 0$. Multiplying b and $\mathbf{w}$ by a constant $C > 1$ then gives

$$-y_p\left(bC + \mathbf{x}_p^T \mathbf{w} C\right) = C\left(-y_p\left(b + \mathbf{x}_p^T \mathbf{w}\right)\right) < -y_p\left(b + \mathbf{x}_p^T \mathbf{w}\right) < 0. \tag{97}$$

Denoting $-y_p\left(bC + \mathbf{x}_p^T\mathbf{w}C\right)$ by α_p, and $-y_p\left(b + \mathbf{x}_p^T\mathbf{w}\right)$ by β_p, we want to show that $\sum_{p=1}^{P}\log\left(1+e^{\alpha_p}\right) < \sum_{p=1}^{P}\log\left(1+e^{\beta_p}\right)$.

Since $\alpha_p < \beta_p$, we have that $e^{\alpha_p} < e^{\beta_p}$. Further, we can write $\log\left(1+e^{\alpha_p}\right) < \log\left(1+e^{\beta_p}\right)$. Summing this inequality over all p then gives the desired inequality.

b) According to part a) if we keep multiplying the hyperplane parameters by $C > 1$, the evaluation of the softmax cost will get smaller and smaller. Theoretically speaking, this means that the minimum of the softmax cost (when the data is separable) is at infinity! Practically speaking, this creates a numerical issue since every computer has a limit on the largest number that can be stored in memory.

Exercise 4.6

a) First note that using the compact notation

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_p \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}, \tag{98}$$

and defining the constant vector $\mathbf{c} = -y_p\tilde{\mathbf{x}}_p$, we can write each summand of the margin and squared margin costs as, $\max\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right)$ and $\max^2\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right)$, respectively. In Exercise 4.1 we use the zeroth order definition of convexity to show that the perceptron cost $h\left(\tilde{\mathbf{w}}\right) = \max\left(0,\, \mathbf{c}^T\tilde{\mathbf{w}}\right)$ is indeed convex. Now note that $\max\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right)$ is just a shifted version of the perceptron, specifically $\max\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right) = h\left(\tilde{\mathbf{w}} + \frac{\mathbf{c}}{\mathbf{c}^T\mathbf{c}}\right)$, and is therefore convex. What remains to show is that a convex function (here $\max\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right)$) maintains its convexity when squared. So given that $g\left(\tilde{\mathbf{w}}\right) = \max\left(0,\, 1+\mathbf{c}^T\tilde{\mathbf{w}}\right)$ is convex we want to prove that $g^2\left(\tilde{\mathbf{w}}\right)$ is convex as well. From the zeroth order definition of convexity for g we know that

$$g\left(\lambda\tilde{\mathbf{w}}_1 + (1-\lambda)\,\tilde{\mathbf{w}}_2\right) \leq \lambda g\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)\,g\left(\tilde{\mathbf{w}}_2\right). \tag{99}$$

Squaring both sides gives

$$g^2\left(\lambda\tilde{\mathbf{w}}_1 + (1-\lambda)\,\tilde{\mathbf{w}}_2\right) \leq \left(\lambda g\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)\,g\left(\tilde{\mathbf{w}}_2\right)\right)^2$$

$$= \lambda^2 g^2\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)^2 g^2\left(\tilde{\mathbf{w}}_2\right) + 2\lambda\,(1-\lambda)\,g\left(\tilde{\mathbf{w}}_1\right)g\left(\tilde{\mathbf{w}}_2\right)$$

$$= \lambda g^2\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)\,g^2\left(\tilde{\mathbf{w}}_2\right) - \lambda\,(1-\lambda)\left(g\left(\tilde{\mathbf{w}}_1\right) - g\left(\tilde{\mathbf{w}}_2\right)\right)^2$$

$$\leq \lambda g^2\left(\tilde{\mathbf{w}}_1\right) + (1-\lambda)\,g^2\left(\tilde{\mathbf{w}}_2\right) - \lambda\,(1-\lambda)\left(g\left(\tilde{\mathbf{w}}_1\right) - g\left(\tilde{\mathbf{w}}_2\right)\right)^2, \tag{100}$$

where the last inequality holds since $\lambda\left(1-\lambda\right)\left(g\left(\tilde{\mathbf{w}}_1\right)-g\left(\tilde{\mathbf{w}}_2\right)\right)^2 \geq 0$.

b) See part b) of Exercise 4.1.

Exercise 4.7

a) Adopting the familiar compact notation, the squared margin cost is given by $g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \max^2\left(0,\, 1 - y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)$. The partial derivative of the cost with respect to the i th entry in $\tilde{\mathbf{w}}$ can then be computed as

$$\frac{\partial}{\partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right)\frac{\partial}{\partial \tilde{w}_i}\left(1 - y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)$$

$$= -2\sum_{p=1}^{P}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{x}_{p,i}. \tag{101}$$

The full gradient therefore can be written as

$$\nabla g = -2\sum_{p=1}^{P}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{\mathbf{x}}_p. \tag{102}$$

c) See our code repository.

Exercise 4.8

See our code repository.

Exercise 4.9

See our code repository.

Exercise 4.10

See our code repository.

Exercise 4.11

a) For the p th data point we want the following approximation to hold

$$\tanh\left(y_p\left(b + \mathbf{x}_p^T \mathbf{w}\right)\right) \approx 1. \tag{103}$$

Using the definition of $\tanh(\cdot)$, (103) can be written as

$$\tanh\left(y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)\right) = 2\sigma\left(y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)\right) - 1 \approx 1, \tag{104}$$

or equivalently

$$\frac{1}{1+e^{-y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)}} \approx 1. \tag{105}$$

For (105) to hold, we must have the following for all $p = 1 \dots P$

$$e^{-y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)} \approx 0. \tag{106}$$

Taking the sum over all data points, we get the desired cost function as

$$h_2(b,\mathbf{w}) = \sum_{p=1}^{P} e^{-y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)}. \tag{107}$$

b) See our code repository.

c) Denote by α the value of $-y\left(b^\star+\mathbf{x}^T\mathbf{w}^\star\right)$ evaluated at the outlier. With h_2 the outlier penalizes the cost function by e^α, which is very large considering that α is a rather large positive value itself. In fact e^α will greatly dominate every other error term introduced by other data points. In order to ameliorate the *exponentially* large contribution of the outlier to the cost function, the classifier then has to move closer to the outlier at the expense of misclassifying two other data points. This however does not happen when using h_1 since the log function suppresses this undesired effect of $e^{(\cdot)}$.

Exercise 4.12

We show that the p th summands in $g(b,\mathbf{w})$ and $h(b,\mathbf{w})$ are identical. Consider the following cases:

Case 1. $y_p = +1$

The p th summand in $g(b,\mathbf{w})$ is given by

$$\log\left(1+e^{-\left(b+\mathbf{x}_p^T\mathbf{w}\right)}\right). \tag{108}$$

With $y_p = +1$, we have $\bar{y}_p = +1$ and the p th summand of $h(b,\mathbf{w})$ can be written as $-\bar{y}_p \log\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right) - (1-\bar{y}_p)\log\left(1-\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right)\right)$, or equivalently

$$-\log\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right). \tag{109}$$

Writing $\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right)$ as $\frac{1}{1+e^{-\left(b+\mathbf{x}_p^T\mathbf{w}\right)}}$, it is clear that the expressions in (108) and (109) are identical.

Case 2. $y_p = -1$

The p th summand in $g\left(b,\mathbf{w}\right)$ is given by

$$\log\left(1+e^{\left(b+\mathbf{x}_p^T\mathbf{w}\right)}\right). \tag{110}$$

With $y_p = -1$, we have $\bar{y}_p = 0$ and the p th summand of $h\left(b,\mathbf{w}\right)$ can be written as $-\bar{y}_p \log\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right) - (1-\bar{y}_p)\log\left(1-\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right)\right)$, or equivalently

$$-\log\left(1-\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right)\right). \tag{111}$$

Now note that $1-\sigma\left(b+\mathbf{x}_p^T\mathbf{w}\right) = \frac{e^{-\left(b+\mathbf{x}_p^T\mathbf{w}\right)}}{1+e^{-\left(b+\mathbf{x}_p^T\mathbf{w}\right)}} = \frac{1}{1+e^{\left(b+\mathbf{x}_p^T\mathbf{w}\right)}}$. Hence, the expressions in (110) and (111) are indeed identical.

Exercise 4.13

This is a programming exercise. See our code repository.

Exercise 4.14

This is a programming exercise. See our code repository.

Exercise 4.15

a) Here we compute the gradient of the k th classifier $\tilde{\mathbf{w}}_k$ from the softmax multiclass cost function

$$g = \sum_{c=1}^{C}\sum_{p\in\Omega_c}\log\left(\sum_{j=1}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_c\right)}\right). \tag{112}$$

Since this is a sum over all of the P points we can compute its gradient by computing the gradient per point $\tilde{\mathbf{x}}_p$ and then summing the results. So we compute the gradient for an individual summand corresponding to an arbitrray point $\tilde{\mathbf{x}}_p$. We can do this in two cases, combining the results afterwards.

Suppose first that this point first is in class k. All that remains from the original cost function above is then just

$$\log\left(1+\sum_{\substack{j=1\\ j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k\right)}\right). \tag{113}$$

Using the chain rule and elementary calculus rules for taking the derivative of $\log(\cdot)$ and $e^{(\cdot)}$ the gradient of this is then given as

$$\frac{-\sum\limits_{\substack{j=1\\j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k)}\tilde{\mathbf{x}}_p}{1+\sum\limits_{\substack{j=1\\j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k)}}. \tag{114}$$

Adding and subtracing 1 in the numerator of this fraction allows us to rewrite the above as

$$\frac{1-\left(1+\sum\limits_{\substack{j=1\\j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k)}\right)}{1+\sum\limits_{\substack{j=1\\j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k)}}\tilde{\mathbf{x}}_p = \left(\frac{1}{1+\sum\limits_{\substack{j=1\\j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_k)}}-1\right)\tilde{\mathbf{x}}_p. \tag{115}$$

Now suppose $\tilde{\mathbf{x}}_p$ is not in class k but some class $c \neq k$. Then we have that what falls out from the original cost function - with respect to $\tilde{\mathbf{w}}_k$ - is just

$$\log\left(\sum_{j=1}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_c)}\right). \tag{116}$$

The gradient of this summation with respect to $\tilde{\mathbf{w}}_k$ can then be computed once again using the chain rule and elementary calculus rules for taking the derivative of $\log(\cdot)$ and $e^{(\cdot)}$ as

$$\log\left(1+\sum_{\substack{j=1\\j\neq c}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_c)}\right), \tag{117}$$

and the resulting gradient is then

$$\frac{e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_k-\tilde{\mathbf{w}}_c)}\tilde{\mathbf{x}}_p}{1+\sum\limits_{\substack{j=1\\j\neq c}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_c)}}. \tag{118}$$

Multiplying the top and bottom of this fraction by $e^{-\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_k-\tilde{\mathbf{w}}_c)}$ then gives

$$\frac{1}{e^{-\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_k-\tilde{\mathbf{w}}_c)}\left(1+\sum\limits_{\substack{j=1\\j\neq c}}^{C} e^{\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_j-\tilde{\mathbf{w}}_c)}\right)}\tilde{\mathbf{x}}_p. \tag{119}$$

Now using the exponential rule $e^a e^b = e^{a+b}$ we can simplify the denominator of the fraction above to

$$e^{-\tilde{\mathbf{x}}_p^T(\tilde{\mathbf{w}}_k - \tilde{\mathbf{w}}_c)} + \sum_{\substack{j=1 \\ j\neq c}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_k\right)} = 1 + \sum_{\substack{j=1 \\ j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_k\right)}. \tag{120}$$

This means that the fraction itself can then be written equivalently as

$$\left(\frac{1}{1 + \sum_{\substack{j=1 \\ j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_k\right)}} \right) \tilde{\mathbf{x}}_p. \tag{121}$$

Together then, we can write the gradient of the multiclass softmax cost over one point $\tilde{\mathbf{x}}_p$ as

$$\left(\frac{1}{1 + \sum_{\substack{j=1 \\ j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_k\right)}} - 1_{p\in\Omega_k} \right) \tilde{\mathbf{x}}_p. \tag{122}$$

This means that over all points the full gradient with respect to $\tilde{\mathbf{w}}_k$ then becomes

$$\nabla g\left(\tilde{\mathbf{w}}_k\right) = \sum_{p=1}^{P} \left(\frac{1}{1 + \sum_{\substack{j=1 \\ j\neq k}}^{C} e^{\tilde{\mathbf{x}}_p^T\left(\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_k\right)}} - 1_{p\in\Omega_k} \right) \tilde{\mathbf{x}}_p. \tag{123}$$

b) See our code repository.

Exercise 4.16

This is a programming exercise. See our code repository.

Exercise 4.17

With $C = 2$ the multiclass softmax cost

$$\sum_{c=1}^{C} \sum_{p\in\Omega_c} \log\left(1 + \sum_{\substack{j=1 \\ j\neq c}}^{C} e^{\left(b_j - b_c\right) + \mathbf{x}_p^T\left(\mathbf{w}_j - \mathbf{w}_c\right)} \right), \tag{124}$$

reduces to

$$\sum_{p\in\Omega_1}\log\left(1+e^{\left(b_2-b_1\right)+\mathbf{x}_p^T\left(\mathbf{w}_2-\mathbf{w}_1\right)}\right)+\sum_{p\in\Omega_2}\log\left(1+e^{\left(b_1-b_2\right)+\mathbf{x}_p^T\left(\mathbf{w}_1-\mathbf{w}_2\right)}\right). \tag{125}$$

Now note that because we have that $y_p=\begin{cases}-1 & p\in\Omega_1\\ +1 & p\in\Omega_2\end{cases}$, the cost in (125) can be written equivalently as

$$\sum_{p\in\Omega_1}\log\left(1+e^{-y_p\left(\left(b_2-b_1\right)+\mathbf{x}_p^T\left(\mathbf{w}_2-\mathbf{w}_1\right)\right)}\right)+\sum_{p\in\Omega_2}\log\left(1+e^{-y_p\left(\left(b_2-b_1\right)+\mathbf{x}_p^T\left(\mathbf{w}_2-\mathbf{w}_1\right)\right)}\right), \tag{126}$$

which can then be written in a more compact form as

$$\sum_{p=1}^{P}\log\left(1+e^{-y_p\left(\left(b_2-b_1\right)+\mathbf{x}_p^T\left(\mathbf{w}_2-\mathbf{w}_1\right)\right)}\right). \tag{127}$$

Finally letting $b=b_2-b_1$ and $\mathbf{w}=\mathbf{w}_2-\mathbf{w}_1$, we arrive at the familiar two-class softmax cost function

$$\sum_{p=1}^{P}\log\left(1+e^{-y_p\left(b+\mathbf{x}_p^T\mathbf{w}\right)}\right). \tag{128}$$

Exercise 4.19

This is a programming exercise. See our code repository.

Exercise 4.20

This is a programming exercise. See our code repository.

Exercise 4.21

This is a programming exercise. See our code repository.

Chapter 5. Automatic feature design for regression

Exercise 5.1

Note that by concatenating w_m's into the column vector $\mathbf{w}$, and $\mathbf{x}_m$'s into the columns of matrix $\mathbf{X}$, the objective function can be written as

$$g\left(\mathbf{w}\right)=\left\|\mathbf{X}\mathbf{w}-\mathbf{y}\right\|_2^2=\left(\mathbf{X}\mathbf{w}-\mathbf{y}\right)^T\left(\mathbf{X}\mathbf{w}-\mathbf{y}\right)=\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}-2\mathbf{y}^T\mathbf{X}\mathbf{w}+\mathbf{y}^T\mathbf{y}. \tag{129}$$

Setting the gradient to zero, we have

$$\nabla g = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = \mathbf{0}, \tag{130}$$

which gives the linear system

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}, \tag{131}$$

to be solved for $\mathbf{w}$. Now because $\mathbf{X}^T\mathbf{X} = S\mathbf{I}_{M\times M}$, this system reduces to

$$\mathbf{w} = \frac{1}{S}\mathbf{X}^T\mathbf{y}, \tag{132}$$

In other words, we have $w_j = \frac{1}{S}\mathbf{x}_j^T\mathbf{y}$ for all j. Note that orthogonality in this case removes the need for solving a general linear system of equations through elimination, matrix inversion or other methods.

Exercise 5.2

Taking the derivative of the objective in w_j we have

$$\frac{\partial}{\partial w_j}\int_0^1 \left(\sum_{m=0}^{\infty} f_m(x)\, w_m - y(x)\right)^2 dx = 2\int_0^1 f_j(x)\left(\sum_{m=0}^{\infty} f_m(x)\, w_m - y(x)\right) dx = 0. \tag{133}$$

This is equivalently written as

$$\sum_{m=0}^{\infty}\left(\int_0^1 f_j(x)\, f_m(x)\, dx\right) w_m = \int_0^1 f_j(x)\, y(x)\, dx. \tag{134}$$

By orthogonality of the basis functions, this reduces to

$$\left(\int_0^1 f_j(x)\, f_j(x)\, dx\right) w_j = \int_0^1 f_j(x)\, y(x)\, dx. \tag{135}$$

giving

$$w_j = \frac{1}{S}\int_0^1 f_j(x)\, y(x)\, dx. \tag{136}$$

Exercise 5.3

Setting the derivative of the objective in w_0 gives

$$2\int_0^1 \left(w_0 + \sum_{k=1}^{K}\left(w_{2k-1}\sin(2\pi kx) + w_{2k}\cos(2\pi kx)\right) - y(x)\right) dx = 0. \tag{137}$$

Since $\int_0^1 \sin(2\pi kx)\,dx = 0$ and $\int_0^1 \cos(2\pi kx)\,dx = 0$ for all $k \geq 1$ solving for w_0 in the above gives

$$w_0 = \int_0^1 y(x)\,dx. \tag{138}$$

Next, setting the derivative of the objective in w_{2j-1} to zero gives

$$2\int_0^1 \sin(2\pi jx)\left(w_0 + \sum_{k=1}^{K}\left(w_{2k-1}\sin(2\pi kx) + w_{2k}\cos(2\pi kx)\right) - y(x)\right)dx = 0. \tag{139}$$

With the orthogonality conditions stated in the problem this reduces to

$$w_{2j-1}\int_0^1 \sin^2(2\pi jx)\,dx - \int_0^1 \sin(2\pi jx)\,y(x)\,dx = 0, \tag{140}$$

which is equivalently

$$w_{2j-1} = 2\int_0^1 \sin(2\pi jx)\,y(x)\,dx. \tag{141}$$

Finally, setting the derivative in w_{2j} to zero, and following the same reasoning, it easily follows that

$$w_{2j} = 2\int_0^1 \cos(2\pi jx)\,y(x)\,dx. \tag{142}$$

Exercise 5.4

Taking the partial derivative of the Least Squares cost $\int_0^1 \left(\sum_{m=0}^{D} x^m w_m - y(x)\right)^2 dx$ with respect to w_j, we have

$$\begin{aligned}
&\tfrac{\partial}{\partial w_j}\int_0^1 \left(\sum_{m=0}^{D} x^m w_m - y(x)\right)^2 dx = \int_0^1 \tfrac{\partial}{\partial w_j}\left(\sum_{m=0}^{D} x^m w_m - y(x)\right)^2 dx \\
&= \int_0^1 2\left(\sum_{m=0}^{D} x^m w_m - y(x)\right)x^j dx = 2\int_0^1 \left(\sum_{m=0}^{D} x^{m+j} w_m - y(x)\,x^j\right)dx \\
&\qquad = 2\int_0^1 \left(\sum_{m=0}^{D} x^{m+j} w_m\right)dx - 2\int_0^1 y(x)\,x^j dx \\
&\qquad = 2\sum_{m=0}^{D}\left(\int_0^1 x^{m+j}dx\right)w_m - 2\int_0^1 y(x)\,x^j dx.
\end{aligned} \tag{143}$$

Setting the above to zero for $j = 0, \ldots, D$ we have

$$\sum_{m=0}^{D}\left(\int_0^1 x^{m+j}dx\right)w_m = \int_0^1 y(x)\,x^j dx \qquad \text{for all } j. \tag{144}$$

Noting that $\int_0^1 x^{m+j}dx = \frac{1}{m+j+1}$, we can write the linear system of $D+1$ equations above as

$$\mathbf{Pw} = \mathbf{d}, \tag{145}$$

where $\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \dots & w_D \end{bmatrix}^T$, $P_{m,j} = \frac{1}{m+j+1}$ $(0 \leq m, j \leq D)$, and $h_j = \int_0^1 y(x)\, x^j dx$ $(0 \leq j \leq D)$.

$$\mathbf{P} = \begin{bmatrix} 1 & 1/2 & 1/3 & \cdots & 1/D+1 \\ 1/2 & 1/3 & 1/4 & \cdots & 1/D+2 \\ 1/3 & 1/4 & 1/5 & \cdots & 1/D+3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/D+1 & 1/D+2 & 1/D+3 & \cdots & 1/2D+1 \end{bmatrix}, \tag{146}$$

and

$$\mathbf{d} = \begin{bmatrix} \int_0^1 y(x)\, dx \\ \int_0^1 y(x)\, x dx \\ \int_0^1 y(x)\, x^2 dx \\ \vdots \\ \int_0^1 y(x)\, x^D dx \end{bmatrix}. \tag{147}$$

Exercise 5.5

$$w_0 + \sum_{m=1}^{M} \cos(2\pi m x)\, w_{2m-1} + \sin(2\pi m x)\, w_{2m}$$

$$= w_0 + \sum_{m=1}^{M} \frac{1}{2}\left(e^{2\pi i m x} + e^{-2\pi i m x}\right) w_{2m-1} + \frac{1}{2i}\left(e^{2\pi i m x} - e^{-2\pi i m x}\right) w_{2m}$$

$$= w_0 + \sum_{m=1}^{M} \frac{1}{2}\left(w_{2m-1} - i w_{2m}\right) e^{2\pi i m x} + \frac{1}{2}\left(w_{2m-1} + i w_{2m}\right) e^{-2\pi i m x}$$

$$= w_0 + \sum_{m=1}^{M} \frac{1}{2}\left(w_{2m-1} - i w_{2m}\right) e^{2\pi i m x} + \sum_{m=1}^{M} \frac{1}{2}\left(w_{2m-1} + i w_{2m}\right) e^{-2\pi i m x}$$

$$= w_0 + \sum_{m=1}^{M} \frac{1}{2}\left(w_{2m-1} - i w_{2m}\right) e^{2\pi i m x} + \sum_{m=-1}^{-M} \frac{1}{2}\left(w_{1-2m} + i w_{-2m}\right) e^{2\pi i m x}$$

$$= w'_0 e^{2\pi i 0} + \sum_{m=1}^{M} w'_m e^{2\pi i m x} + \sum_{m=-1}^{-M} w'_m e^{2\pi i m x} = \sum_{m=-M}^{M} w'_m e^{2\pi i m x}. \tag{148}$$

Exercise 5.6

a) As shown pictorially in Figure 4, with a 3-hidden layer neural network basis approximation we have

$$Q = (N+1)\,M_3 + (M_3+1)\,M_2 + (M_2+1)\,M + M + 1. \tag{149}$$

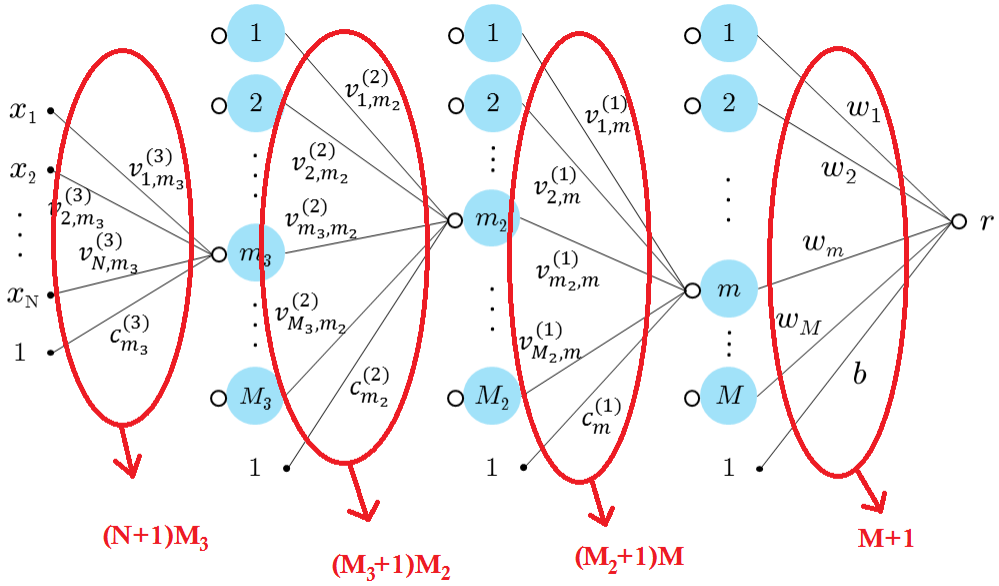

Figure 4. Calculating the total number of weights in a three hidden-layer feedforward neural network.

In general, defining $M_0 = 1$, $M_1 = M$, and $M_{L+1} = N$, the total number of parameters in a fully-connected feedforward neural network with L hidden layers can be written as $Q = \sum_{k=1}^{L+1} (M_k + 1)\, M_{k-1}$.

b) According to part a), Q can be written as $Q = N \cdot M_L + \left(M_L + \sum_{k=1}^{L} (M_k + 1)\, M_{k-1} \right)$, where the expression inside the parantheses is constant with respect to N. Note that Q is independent of the number of data points P. As we will see in Chapter 7, this is not the case with kernel methods (used for fixed basis regression) in which case Q depends on P.

Exercise 5.7

This is a programming exercise. See our code repository.

Exercise 5.8

This is a programming exercise. See our code repository.

Exercise 5.9

a) The Least Squares cost function is given by

$$g = \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right)^2 . \tag{150}$$

Taking the partial derivative of g with respect to b gives

$$\frac{\partial}{\partial b} g = \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial b} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \right)$$

$$= 2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) . \tag{151}$$

Taking the partial derivative of g with respect to w_n gives $(1 \leq n \leq M)$

$$\frac{\partial}{\partial w_n} g = \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial w_n} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \right)$$

$$= 2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) a \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) . \tag{152}$$

Taking the partial derivative of g with respect to c_n gives $(1 \leq n \leq M)$

$$\frac{\partial}{\partial c_n} g = \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial c_n} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \right)$$

$$= \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial c_n} \left(a \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n \right) \right)$$

$$= 2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) a' \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n . \tag{153}$$

Taking the partial derivative of g with respect to the j th entry in $\mathbf{v}_n$ gives $(1 \leq n \leq M)$

$$\frac{\partial}{\partial v_{n,j}} g = \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial v_{n,j}} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \right)$$

$$= \sum_{p=1}^{P} \left(2 \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) \frac{\partial}{\partial v_{n,j}} \left(a \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n \right) \right)$$

$$= 2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) a' \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n \frac{\partial}{\partial v_{n,j}} \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right)$$

$$= 2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) a' \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n x_{p,j}. \tag{154}$$

Therefore the full gradient with respect to $\mathbf{v}_n$ can be written as

$$2 \sum_{p=1}^{P} \left(b + \sum_{m=1}^{M} a \left(c_m + \mathbf{x}_p^T \mathbf{v}_m \right) w_m - y_p \right) a' \left(c_n + \mathbf{x}_p^T \mathbf{v}_n \right) w_n \mathbf{x}_p. \tag{155}$$

c) See our code repository.

Exercise 5.10

Eric should choose $D = 5$ as it produces minimal testing error.

Stanley could choose either $D = 3$ or $D = 8$ but $D = 3$ produces a simpler model (Occam's razor).

Even though Kyle could choose $D = 10$, the decreasing trend in both the training and testing error curves suggests that we are still in the underfitting zone! So Kyle should try even larger values for D until the testing error ceases to decrease or goes back up.

In Kenny's plot, the training error does not consistently go down which is indicative of a problem in his implementation. Therefore he should look into his implementation for debugging!

Exercise 5.11

This is a programming exercise. See our code repository.

Exercise 5.12

This is a programming exercise. See our code repository.

Exercise 5.13

This is a programming exercise. See our code repository.

Chapter 6. Automatic feature design for classification

Exercise 6.1

a) Using the compact notation the softmax cost can be written as

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \log\left(1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}\right). \tag{156}$$

The partial derivative of the cost with respect to the i th entry in $\tilde{\mathbf{w}}$ can be computed as

$$\frac{\partial}{\partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial \tilde{w}_i}\left(1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}\right)}{1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}}$$

$$= \sum_{p=1}^{P} \frac{\left(e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}\right) \frac{\partial}{\partial \tilde{w}_i}\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)}{1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}} = -\sum_{p=1}^{P} \frac{e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}}{1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{f}_{p,i}. \tag{157}$$

The full gradient therefore can be written as

$$\nabla g = -\sum_{p=1}^{P} \frac{e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}}{1 + e^{-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{\mathbf{f}}_p. \tag{158}$$

Multiplying both the numerator and denominator of each summand by $e^{y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}$ gives

$$\nabla g = -\sum_{p=1}^{P} \frac{1}{1 + e^{y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}}} y_p \tilde{\mathbf{f}}_p = -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{\mathbf{f}}_p. \tag{159}$$

To find the Hessian, we can write

$$\frac{\partial^2}{\partial \tilde{w}_j \partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = -\frac{\partial}{\partial \tilde{w}_j} \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{f}_{p,i}$$

$$= -\sum_{p=1}^{P} \frac{\partial}{\partial \tilde{w}_j}\left(\sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{f}_{p,i}\right) = -\sum_{p=1}^{P} \sigma'\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) y_p \tilde{f}_{p,i} \frac{\partial}{\partial \tilde{w}_j}\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)$$

$$= -\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{f}_{p,i}\left(-y_p \tilde{f}_{p,j}\right)$$

$$= \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{f}_{p,i} \tilde{f}_{p,j} y_p^2. \tag{160}$$

Now note that $y_p^2 = 1$ (since $y_p \in \{-1, +1\}$), and we can write the full Hessian as

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{f}}_p \tilde{\mathbf{f}}_p^T. \tag{161}$$

b) To prove g is convex, we use the second order definition of convexity and show $\nabla^2 g\left(\tilde{\mathbf{w}}\right)$ is positive semidefinite by forming

$$\begin{aligned} \psi\left(\mathbf{z}\right) = \mathbf{z}^T \nabla^2 g\left(\tilde{\mathbf{w}}\right) \mathbf{z} &= \mathbf{z}^T \left[\sum_{p=1}^{P} \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) \tilde{\mathbf{f}}_p \tilde{\mathbf{f}}_p^T\right] \mathbf{z} \\ &= \sum_{p=1}^{P} \delta_p \left(\tilde{\mathbf{f}}_p^T \mathbf{z}\right)^2, \end{aligned} \tag{162}$$

where $\delta_p = \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\left(1 - \sigma\left(-y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right)$. Now note that since $0 \leq \sigma\left(\zeta\right) \leq 1$ we have that $\delta_p \geq 0$, and therefore

$$\mathbf{z}^T \nabla^2 g\left(\tilde{\mathbf{w}}\right) \mathbf{z} = \sum_{p=1}^{P} \delta_p \left(\tilde{\mathbf{f}}_p^T \mathbf{z}\right)^2 \geq 0. \tag{163}$$

Exercise 6.2

a) The squared margin cost in this case is given by $g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \max^2\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)$. The partial derivative of the cost with respect to the i th entry in $\tilde{\mathbf{w}}$ can then be computed as

$$\begin{aligned} \frac{\partial}{\partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) &= 2 \sum_{p=1}^{P} \left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) \frac{\partial}{\partial \tilde{w}_i}\left(1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) \\ &= -2 \sum_{p=1}^{P} \left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{f}_{p,i}. \end{aligned} \tag{164}$$

The full gradient therefore can be written as

$$\nabla g = -2 \sum_{p=1}^{P} \left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{\mathbf{f}}_p. \tag{165}$$

To find the Hessian, we can write

$$\frac{\partial^2}{\partial \tilde{w}_j \partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = \frac{\partial}{\partial \tilde{w}_j}\left(-2 \sum_{p=1}^{P} \left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{f}_{p,i}\right)$$

$$= \frac{\partial}{\partial \tilde{w}_j}\left(-2\sum_{p=1}^{P}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{f}_{p,i}\right)$$

$$= -2\left(\sum_{p=1}^{P}\frac{\partial}{\partial \tilde{w}_j}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) y_p \tilde{f}_{p,i}\right).$$

Note that although the "max" function is not differentiable, since

$$\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) = \begin{cases} 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}} & \text{if } 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}} > 0 \\ 0 & \text{else,} \end{cases}$$

we may generalize the gradient definition and write

$$\frac{\partial}{\partial \tilde{w}_j}\left(\max\left(0,\, 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right)\right) = \begin{cases} \frac{\partial}{\partial \tilde{w}_j}\left(1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) & \text{if } 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}} > 0 \\ 0 & \text{else.} \end{cases}$$

Denoting $\Omega_{\tilde{\mathbf{w}}} = \left\{p | 1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right\}$ and noting that $\frac{\partial}{\partial \tilde{w}_j}\left(1 - y_p \tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}}\right) = -y_p \tilde{f}_{p,j}$, the generalized Hessian can be written entry-wise as

$$\frac{\partial^2}{\partial \tilde{w}_j \partial \tilde{w}_i} g\left(\tilde{\mathbf{w}}\right) = -2\left(\sum_{p \in \Omega_{\tilde{\mathbf{w}}}} - y_p^2 \tilde{f}_{p,i} \tilde{f}_{p,j}\right),$$

and the Hessian matrix as

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p \in \Omega_{\tilde{\mathbf{w}}}} \tilde{\mathbf{f}}_p \tilde{\mathbf{f}}_p^T,$$

since $y_p^2 = 1$.

b) See Exercises 4.1 and 4.6.

Exercise 6.3

This is a programming exercise. See our code repository.

Exercise 6.4

a) The softmax cost in this case is given by

$$g = \sum_{p=1}^{P} \log\left(1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}\right). \tag{166}$$

Taking the partial derivative of g with respect to b gives

$$\frac{\partial}{\partial b}g = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial b}\left(1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}\right)}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)} \frac{\partial}{\partial b}\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)\right)}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{-e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}} y_p. \tag{167}$$

Noting that $\sigma(\zeta) = \frac{e^{\zeta}}{1+e^{\zeta}}$, we can write

$$\frac{\partial}{\partial b}g = -\sum_{p=1}^{P} \sigma\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)\right) y_p. \tag{168}$$

Taking the partial derivative of g with respect to w_n gives ($1 \leq n \leq M$)

$$\frac{\partial}{\partial w_n}g = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial w_n}\left(1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}\right)}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)} \frac{\partial}{\partial w_n}\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)\right)}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}}{1 + e^{-y_p\left(b+\sum_{m=1}^{M} a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)}} \left(-y_p a\left(c_n + \mathbf{x}_p^T\mathbf{v}_n\right)\right)$$

$$= -\sum_{p=1}^{P} \sigma\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)\right) a\left(c_n + \mathbf{x}_p^T\mathbf{v}_n\right) y_p. \tag{169}$$

Taking the partial derivative of g with respect to c_n gives ($1 \leq n \leq M$)

$$\frac{\partial}{\partial c_n} g = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial c_n}\left(1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}\right)}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)} \frac{\partial}{\partial c_n}\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)\right)}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}} \left(-y_p w_n a'\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right) \frac{\partial}{\partial c_n}\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right)\right)$$

$$= -\sum_{p=1}^{P} \sigma\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)\right) a'\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right) w_n y_p. \tag{170}$$

Taking the partial derivative of g with respect to the j th entry in $\mathbf{v}_n$ gives ($1 \leq n \leq M$)

$$\frac{\partial}{\partial v_{n,j}} g = \sum_{p=1}^{P} \frac{\frac{\partial}{\partial v_{n,j}}\left(1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}\right)}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)} \frac{\partial}{\partial v_{n,j}}\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)\right)}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}$$

$$= \sum_{p=1}^{P} \frac{e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}}{1 + e^{-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)}} \left(-y_p w_n a'\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right) \frac{\partial}{\partial v_{n,j}}\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right)\right)$$

$$= -\sum_{p=1}^{P} \sigma\left(-y_p\left(b + \sum_{m=1}^{M} a\left(c_m + \mathbf{x}_p^T \mathbf{v}_m\right) w_m\right)\right) a'\left(c_n + \mathbf{x}_p^T \mathbf{v}_n\right) w_n y_p x_{p,j}. \tag{171}$$

Therefore the full gradient with respect to $\mathbf{v}_n$ can be written as

$$-\sum_{p=1}^{P}\sigma\left(-y_p\left(b+\sum_{m=1}^{M}a\left(c_m+\mathbf{x}_p^T\mathbf{v}_m\right)w_m\right)\right)a'\left(c_n+\mathbf{x}_p^T\mathbf{v}_n\right)w_n y_p\mathbf{x}_p. \tag{172}$$

Exercise 6.5
This is a programming exercise. See our code repository.

Exercise 6.6
This is a programming exercise. See our code repository.

Exercise 6.7
This is a programming exercise. See our code repository.

Exercise 6.8
This is a programming exercise. See our code repository.

Exercise 6.9
This is a programming exercise. See our code repository.

Chapter 8. Advanced gradient schemes

Exercise 8.1
This is a programming exercise. See our code repository.

Exercise 8.2
This is a programming exercise. See our code repository.

Exercise 8.3
This is a programming exercise. See our code repository.

Exercise 8.4
f has Lipschitz continuous gradient with constant J, and g is Lipschitz continuous with constant K, so we can write for all $\mathbf{x}$ and $\mathbf{y}$ in the domain of g

$$\|\nabla f\left(g\left(\mathbf{x}\right)\right)-\nabla f\left(g\left(\mathbf{y}\right)\right)\|_2\leq J\left\|g\left(\mathbf{x}\right)-g\left(\mathbf{y}\right)\right\|_2\leq JK\left\|\mathbf{x}-\mathbf{y}\right\|_2. \tag{173}$$

Therefore, $f\left(g\right)$ has Lipschitz continuous gradient with constant JK.

Exercise 8.5

a) The maximum eigenvalue of a matrix $\mathbf{A}$ can be found, using the Rayleigh quotient, as

$$\max_{\mathbf{x}} \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}. \tag{174}$$

Plugging in $\mathbf{A} = \mathbf{a}\mathbf{a}^T$, we have

$$\frac{\mathbf{x}^T\mathbf{a}\mathbf{a}^T\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\left(\mathbf{x}^T\mathbf{a}\right)\left(\mathbf{a}^T\mathbf{x}\right)}{\mathbf{x}^T\mathbf{x}} = \frac{\left(\mathbf{x}^T\mathbf{a}\right)^2}{\left\|\mathbf{x}\right\|_2^2} = \frac{\left(\left\|\mathbf{x}\right\|_2\left\|\mathbf{a}\right\|_2\cos\left(\theta_{\mathbf{a}}\right)\right)^2}{\left\|\mathbf{x}\right\|_2^2} = \left\|\mathbf{a}\right\|_2^2\cos^2\left(\theta_{\mathbf{a}}\right). \tag{175}$$

where $\theta_{\mathbf{a}}$ is the angle between $\mathbf{a}$ and $\mathbf{x}$. Using a similar argument for $\mathbf{a}\mathbf{a}^T + \mathbf{b}\mathbf{b}^T$ we have

$$\frac{\mathbf{x}^T\left(\mathbf{a}\mathbf{a}^T + \mathbf{b}\mathbf{b}^T\right)\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\left(\mathbf{x}^T\mathbf{a}\mathbf{a}^T\mathbf{x}\right) + \left(\mathbf{x}^T\mathbf{b}\mathbf{b}^T\mathbf{x}\right)}{\mathbf{x}^T\mathbf{x}} = \left\|\mathbf{a}\right\|_2^2\cos^2\left(\theta_{\mathbf{a}}\right) + \left\|\mathbf{b}\right\|_2^2\cos^2\left(\theta_{\mathbf{b}}\right). \tag{176}$$

and hence the maximum eigenvalue of $\mathbf{a}\mathbf{a}^T$ (i.e., $\left\|\mathbf{a}\right\|_2^2$) is always smaller than or equal to that of $\mathbf{a}\mathbf{a}^T + \mathbf{b}\mathbf{b}^T$.

b) Writing the Hessian of the soft margin SVM cost function as

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p\in\Omega_{\tilde{\mathbf{w}}}} \tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T, \tag{177}$$

we have using part a) then since each outer product matrix $\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T$ has nonnegative eigenvalues that the maximum eigenvalue of the Hessian is bounded above by the maximum eigenvalue of the sum of all p points / outer product matrices. In other words that

$$\sum_{p\in\Omega_{\tilde{\mathbf{w}}}} \tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T \preceq \sum_{p=1}^{P} \tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T. \tag{178}$$

The form of the Lipschitz constant in Example 8.8 then follows from this.

c)

Starting with the form of the Hessian given in Example 8.7, $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \tilde{\mathbf{X}}\text{diag}\left(\mathbf{r}\right)\tilde{\mathbf{X}}^T$, note that this can be written as a sum of outer product matrices as

$$\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} r_p\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T. \tag{179}$$

By definition the largest value possible for r_p is $\frac{1}{4}$. Then we have

$$\sum_{p=1}^{P} r_p \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T \preceq \frac{1}{4} \sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T = \frac{1}{4} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T, \tag{180}$$

and the form of the Lipschitz constant follows from this.

Chapter 9. Dimension reduction techniques

Exercise 9.1

This is a programming exercise. See our code repository.

Exercise 9.2

This is a programming exercise. See our code repository.

Exercise 9.3

a) Let us assume that the $N \times N$ symmetric positive semi-definite matrix $\mathbf{X}\mathbf{X}^T$ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$ and corresponding eigenvectors $\mathbf{a}_1 ... \mathbf{a}_N$, with an eigendecomposition $\mathbf{X}\mathbf{X}^T = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T$, where $\mathbf{A}$ is an orthonormal basis whose columns are the eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues along its diagonal. So we can write

$$\mathbf{d}^T \mathbf{X}\mathbf{X}^T \mathbf{d} = \mathbf{d}^T \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T \mathbf{d} = \mathbf{s}^T \mathbf{\Lambda} \mathbf{s}, \tag{181}$$

where $\mathbf{r} = \mathbf{A}^T \mathbf{d}$. Therefore

$$\mathbf{d}^T \mathbf{X}\mathbf{X}^T \mathbf{d} = \sum_{n=1}^{N} \lambda_n r_n^2 \leq \sum_{n=1}^{N} \lambda_1 r_n^2 = \lambda_1 \|\mathbf{r}\|_2^2 = \lambda_1 \mathbf{d}^T \mathbf{A}\mathbf{A}^T \mathbf{d} = \lambda_1 \mathbf{d}^T \mathbf{d} = \lambda_1, \tag{182}$$

where we use the fact that $\mathbf{A}$ is an orthonormal basis and $\mathbf{d}$ has unit length.
Note that if you set $\mathbf{d} = \mathbf{a}_1$ then the objective meets the upper bound since

$$\mathbf{a}_1^T \mathbf{X}\mathbf{X}^T \mathbf{a}_1 = \mathbf{a}_1^T \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T \mathbf{a}_1 = \mathbf{e}_1^T \mathbf{\Lambda} \mathbf{e}_1 = \lambda_1, \tag{183}$$

where $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}^T$ is the first standard basis vector. Hence,

$$\lambda_1 = \max_{\|\mathbf{d}\|_2 = 1} \mathbf{d}^T \mathbf{X}\mathbf{X}^T \mathbf{d}, \tag{184}$$

and the corresponding eigenvector $\mathbf{a}_1$ is the unit length vector that maximizes the objective.

b) We now show among all unit direction vectors perpendicular to $\mathbf{a}_1$, $\mathbf{a}_2$ is the one which maximizes the objective. Using (182) once again we can write

$$\mathbf{d}^T\mathbf{X}\mathbf{X}^T\mathbf{d} = \lambda_1 r_1^2 + \sum_{n=2}^{N} \lambda_n r_n^2 \leq \lambda_2 r_1^2 + \sum_{n=2}^{N} \lambda_2 r_n^2 = \sum_{n=1}^{N} \lambda_2 r_n^2, \tag{185}$$

where the inequality holds since we are only looking for directions perpendicular to $\mathbf{a}_1$, therefore $r_1 = \mathbf{d}^T\mathbf{a}_1 = 0$, and we also have that $\lambda_2 \geq \cdots \geq \lambda_N \geq 0$ by assumption. Hence

$$\mathbf{d}^T\mathbf{X}\mathbf{X}^T\mathbf{d} \leq \sum_{n=1}^{N} \lambda_2 r_n^2 = \lambda_2 \|\mathbf{r}\|_2^2 = \lambda_2 \mathbf{d}^T\mathbf{A}\mathbf{A}^T\mathbf{d} = \lambda_2 \mathbf{d}^T\mathbf{d} = \lambda_2. \tag{186}$$

Finally note that with the input $\mathbf{d} = \mathbf{a}_2$ this upper bound is met.

c) Assume $\mathbf{X}$ has the singular value decomposition $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. Then we can write

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T\left(\mathbf{U}\mathbf{S}\mathbf{V}^T\right)^T = \mathbf{U}\mathbf{S}\mathbf{V}^T\left(\mathbf{V}\mathbf{S}^T\mathbf{U}^T\right) = \mathbf{U}\mathbf{S}\left(\mathbf{V}^T\mathbf{V}\right)\mathbf{S}^T\mathbf{U}^T = \mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{U}^T. \tag{187}$$

From part a) we have the eigendecomposition of $\mathbf{X}\mathbf{X}^T$ as $\mathbf{X}\mathbf{X}^T = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T$. A simple inspection shows $\mathbf{U} = \mathbf{A}$ and $\mathbf{S}\mathbf{S}^T = \mathbf{\Lambda}$.
Note that $\mathbf{S}\mathbf{S}^T$ is a diagonal matrix with singular values squared along its diagnoal.

Exercise 9.4

This is a programming exercise. See our code repository.