Machine Learning for Signal Processing

[5LSL0]

Rik Vullings
Ruud van Sloun
Nishith Chennakeshava
Hans van Gorp

Answers: Nonlinear Models

April 2023

This document outlines all the answers to the questions. As a companion to this answer sheet, we also provide the code answers separately. If an answer is (partially) provided as code we use a 🐍 icon.

Linear models

Exercise 1

[1 pt] We here introduce ourselves the number of examples m. Using the fact that the log of products becomes a sum, and plugging in a Gaussian error distribution, you should arrive at the simple mean squared error loss. Notice how we abuse notation slightly to ignore the variance in the last line, as it has no bearing once we start differentiating the loss.

$$\begin{aligned}
J(\mathrm{X}, \underline{\mathbf{y}}; \underline{\theta}) &= -log \prod_{i=0}^{m-1} \hat{p}_{model}(y^{(i)}; \underline{\mathbf{x}}^{(i)}, \underline{\theta}) \\
&= -log \prod_{i=0}^{m-1} \exp\left\{-\frac{1}{2\sigma^2}\left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^2\right\} \\
&= \sum_{i=0}^{m-1} \frac{1}{2\sigma^2}\left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^2 \\
&= \sum_{i=0}^{m-1} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^2
\end{aligned}$$

Exercise 2

[1 pt] There are three 'tricks' that you can use to make this problem a lot easier (these tricks can come in handy later in the course too):

- First, define the vector of parameters $\underline{\theta}$ by stacking the weight vector with the bias:

$$\underline{\theta} = \begin{bmatrix} \underline{\mathbf{w}} \\ b \end{bmatrix}$$

- Second, because of point one, we have can re-express our linear function as:

$$f(\underline{\mathbf{x}}; \underline{\mathbf{w}}, b) = \underline{\mathbf{w}}^T \underline{\mathbf{x}} + b = \underline{\theta}^T \begin{bmatrix} \underline{\mathbf{x}} \\ 1 \end{bmatrix} = \underline{\theta}^T \underline{\tilde{\mathbf{x}}}$$

- Third, we can parallelize a whole set of operations by introducing the matrix:

$$\mathrm{X} = [\underline{\tilde{\mathbf{x}}}^{(0)}, \underline{\tilde{\mathbf{x}}}^{(1)}, ...\underline{\tilde{\mathbf{x}}}^{(m-1)}]$$

Using these three tricks, we can express our loss function from exercise 1 as:

$$\begin{aligned}
J(\mathrm{X}, \underline{\mathbf{y}}; \underline{\theta}) &= \sum_{i=0}^{m-1} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^2 \\
&= \left(\underline{\mathbf{y}} - \underline{\theta}^T \mathrm{X}\right)\left(\underline{\mathbf{y}} - \underline{\theta}^T \mathrm{X}\right)^T
\end{aligned}$$

This is a linear problem, so we can simply take the gradient of the cost function with respect to $\underline{\theta}$ and equate it to zero.

$$\partial_{\underline{\theta}} J(\mathrm{X}, \underline{\mathbf{y}}; \underline{\theta}) = 2\mathrm{X}\mathrm{X}^T\underline{\theta} - 2\mathrm{X}\underline{\mathbf{y}}^T = 0$$

$$\underline{\theta} = (\mathrm{X}\mathrm{X}^T)^{-1}\mathrm{X}\underline{\mathbf{y}}^T$$

Exercise 3

[1 pt] 🐍 Simply filling in the values into the expression derived above yields $\underline{\mathbf{w}}^T = [0.1, 0.4]$ and $b = 0$. The process is well described since the MSE is very low $\approx 2e^{-32}$

Exercise 4

[1 pt] 🐍 The new parameters values are: $\underline{\mathbf{w}}^T = [0.1011, 0.4107]$ and $b = -0.05023$. The MSE becomes higher. We can obtain improved estimates by taking more measurements.

Exercise 5

[1 pt] following a simmilar approach to exercise 1, we arrive at:

$$\begin{aligned}
J(\mathrm{X}, \underline{\mathbf{y}}; \underline{\theta}) &= -log \prod_{i=0}^{m-1} \hat{p}_{model}(y^{(i)}; \underline{\mathbf{x}}^{(i)}, \underline{\theta}) \\
&= -log \prod_{i=0}^{m-1} \exp\left\{-\frac{1}{2}\left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^T \Sigma^{-1} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]\right\} \\
&= \sum_{i=0}^{m-1} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^T \Sigma^{-1} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right] \\
&= \sum_{i=0}^{m-1} \frac{1}{\sigma_i^2} \left[y^{(i)} - f(\underline{\mathbf{x}}^{(i)}; \underline{\theta})\right]^2
\end{aligned}$$

Conclusion: The balance between the variances of the parameter estimates gives a weighted least squares solution that depends on the structure of the covariance matrix. Because the covariance matrix is a diagonal matrix, the weighted solution per sample depends on the variance at that sample. With lower (and thus more trustworthy) variances counting more towards the final loss.

Exercise 6

[1 pt] 🐍 We use the following inputs and targets:

$$\mathrm{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \ \underline{\mathbf{y}} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T$$

Which results in $\underline{\mathbf{w}}^T = [0, 0]$ and $b = 0.5$.

Nonlinear functions

Exercise 7

[1 pt]

ReLU

Split into two cases, and choose yourself where $x = 0$ belongs to (it does not really matter in the grand scheme of things):

$$\partial_x f(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$$

Sigmoid

Apply the quotient rule (the last part, rewriting to $\sigma(x)(1 - \sigma(x))$ is optional):

$$\partial_x f(x) = \frac{\exp(-x)}{(1 + \exp(-x))^2} = \sigma(x)(1 - \sigma(x))$$

Softmax

Derive partial derivatives of i-th output w.r.t. the j-th input, and split into two cases:

$$\partial_{x_j} f(\underline{\mathbf{x}})_i = \begin{cases} f(\underline{\mathbf{x}})_i \ (1 - f(\underline{\mathbf{x}})_j) & \text{for } i = j \\ -f(\underline{\mathbf{x}})_j \ f(\underline{\mathbf{x}})_i & \text{for } i \neq j \end{cases}$$

Which can be further simplified using the Kronecker delta function:

$$\partial_{x_j} f(\underline{\mathbf{x}})_i = f(\underline{\mathbf{x}})_i \ (\delta_{ij} - f(\underline{\mathbf{x}})_j)$$

Do you notice the similarities between the sigmoid and the softmax?

Exercise 8

[1 pt]

ReLU

The gradient remains 1.

Sigmoid

The gradient becomes 0 and “vanishes”.

Softmax

Let’s say element i in vector $\underline{\mathbf{x}}$ does approach infinity while the rest stays constant: $x_i \to \infty$. Then we have $f(\underline{\mathbf{x}})_i \to 1$ and $f(\underline{\mathbf{x}})_j \to 0$. As a result, $\partial_{x_j} f(\underline{\mathbf{x}})_i \to 0$ for both $i = j$ and $i \neq j$. Thus, in both cases the gradient becomes 0 and “vanishes”.

Shallow (i.e. not deep...) nonlinear models

Exercise 9

[1 pt] Two sequentially applied linear functions can be written as a single linear function, and therefore do not increase the model capacity.

Exercise 10

[1 pt] 🐍 The decision boundary in latent space is found by setting the linear equation there equal to 0.5:

$$\begin{aligned}\left(\underline{\mathbf{w}}^{(2)}\right)^T \underline{\mathbf{h}} + b_2 &= 0.5\\ \begin{bmatrix}1 & -2\end{bmatrix}\begin{bmatrix}h_1\\ h_2\end{bmatrix} &= 0.5\\ h_1 - 2h_2 &= 0.5\end{aligned}$$

Exercise 11

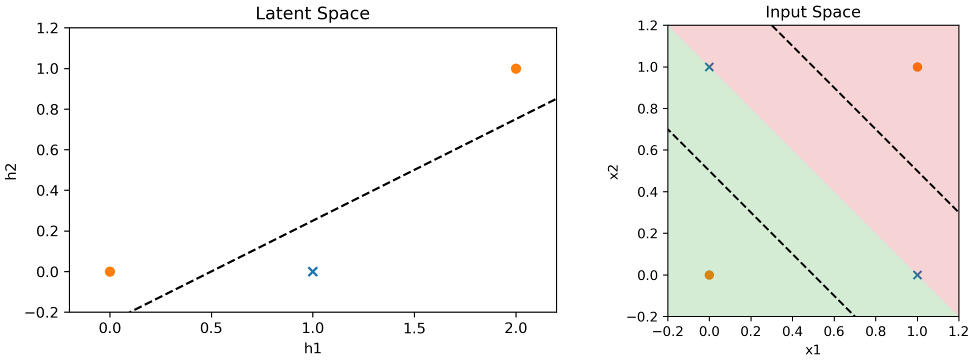
[1 pt] 🐍 Filling in all the values for the full network equation and equating it to 0.5:

$$\begin{aligned}\left(\underline{\mathbf{w}}^{(2)}\right)^T \max\left(0, \mathrm{W}^{(1)}\underline{\mathbf{x}} + \underline{\mathbf{b}}^{(1)}\right) + b^{(2)} &= 0.5\\ \begin{bmatrix}1 & -2\end{bmatrix} \max\left(0, \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ -1\end{bmatrix}\right) &= 0.5\\ \begin{bmatrix}1 & -2\end{bmatrix} \max\left(0, \begin{bmatrix}x_1 + x_2\\ x_1 + x_2 - 1\end{bmatrix}\right) &= 0.5\end{aligned}$$

We can then split this into the two cases where it is possible that the function is equal to 0.5 (the third case is always 0):

$$\begin{aligned}x_1 + x_2 = 1.5 \text{ for } x_1 + x_2 > 1\\ x_1 + x_2 = 0.5 \text{ for } x_1 + x_2 < 1\end{aligned}$$

Your plots should then look like this: (optional shading, red: $x_1+x_2 > 1$, green: $x_1+x_2 < 1$)



Binary classification with logistic regression

Exercise 12

[1 pt] Softmax, as all outputs (probabilities) sum to one.

Exercise 13

[1 pt]

$$\partial_p J = -\sum_i \frac{y^{(i)}}{p^{(i)}} - \frac{1 - y^{(i)}}{1 - p^{(i)}}$$

Exercise 14

[1 pt]

$$\partial_f p = \sigma(f)(1 - \sigma(f))$$

Exercise 15

[1 pt]

$$\partial_{w_j} f = x_j$$

This relates to the LMS filter.

Exercise 16

[1 pt]

$$\partial_{w_j} J = -\sum_i \frac{y^{(i)}}{p^{(i)}} - \frac{1 - y^{(i)}}{1 - p^{(i)}} \sigma(f)(1 - \sigma(f)) x_j^{(i)}$$

Classification with a shallow nonlinear model

Exercise 17

[4 pt]

$$\partial_{\underline{\mathbf{w}}^{(2)}} J = \begin{cases} \underline{\mathbf{0}} & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} < 0 \\ -\left(\sum_i y^{(i)} - \sigma(f^{(2)})\right)\left(\mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)}\right) & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} \geq 0 \end{cases}$$

$$\partial_{\mathrm{W}^{(1)}} J = \begin{cases} \underline{\mathbf{0}} & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} < 0 \\ -\left(\sum_i y^{(i)} - \sigma(f^{(2)})\right)\left(\underline{\mathbf{w}}^{(2)}\right)^T \underline{\mathbf{x}}^{(i)} & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} \geq 0 \end{cases}$$

$$\partial_{b^{(2)}} J = -\left(\sum_i y^{(i)} - \sigma(f^{(2)})\right)$$

$$\partial_{\underline{\mathbf{b}}^{(1)}} J = \begin{cases} \underline{\mathbf{0}} & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} < 0 \\ -\left(\sum_i y^{(i)} - \sigma(f^{(2)})\right)\left(\underline{\mathbf{w}}^{(2)}\right)^T & \text{if } \mathrm{W}^{(1)}\underline{\mathbf{x}}^{(i)} + \underline{\mathbf{b}}^{(1)} \geq 0 \end{cases}$$