Chapter 6: Fitting models

Problem 6.1

Show that the derivatives of the least squares loss function in equation 6.5 are given by the expressions in equation 6.7

$$L = \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y + i)^2 \tag{6.5}$$

Taking the derivative of Equation 6.5 with respect to ϕ_0, we get:

$$\frac{\partial L}{\partial \phi_0} = 2(\phi_0 + \phi_1 x_i - y_i)$$

and taking the derivative with respect to ϕ_1, we get:

$$\frac{\partial L}{\partial \phi_1} = 2x_i(\phi_0 + \phi_1 x_i - y_i)$$

Which is the same as shown in equation 6.7.

Problem 6.2

A surface is convex is the eigenvalues of the Hessian $\mathbf{H}[\phi]$ are positive everywhere. In this case, the surface has a unique minimum, and optimisation is easy. Find an algebraic expression for the Hessian matrix,

$$\mathbf{H}[\phi] = \begin{bmatrix} \frac{\partial^2 L}{\partial \phi^2} & \frac{\partial^2 L}{\partial \phi \partial \theta} \\ \frac{\partial^2 L}{\partial \theta \partial \phi} & \frac{\partial^2 L}{\partial \theta^2} \end{bmatrix}$$

for the linear regression model (Equation 6.5). Prove that this function is convex by showing that the eigenvalues are always positive. This can be done by showing that both the trace and the determinant of the matrix are positive.

Start by computing each term in the Hessian matrix where $\phi = \phi_0$ and $\theta = \phi_1$:

$$\frac{\partial^2 L}{\partial \phi^2} = 2\sum_{i=1}^{I} 1 = 2I$$

$$\frac{\partial^2 L}{\partial\phi\partial\theta} = 2\sum_{i=1}^{I} x_i$$

$$\frac{\partial^2 L}{\partial\theta\partial\phi} = 2\sum_{i=1}^{I} x_i$$

$$\frac{\partial^2 L}{\partial\theta^2} = 2\sum_{i=1}^{I} x_i^2$$

Therefore, the Hessian matrix is:

$$\mathbf{H}[\phi] = \begin{bmatrix} 2I & 2\sum_{i=1}^{I} x_i \\ 2\sum_{i=1}^{I} x_i & 2\sum_{i=1}^{I} x_i^2 \end{bmatrix}$$

The trace of this Hessian matrix is:

$$\text{Tr}(\mathbf{H}[\phi]) = 2I + 2\sum_{i=1}^{I} x_i^2$$

The determinant of the Hessian matrix is:

$$\text{Det}(\mathbf{H}[\phi]) = 4I\sum_{i=1}^{I} x_i^2 - 4\left(\sum_{i=1}^{I} x_i\right)^2$$

Which are both positive due to the squared terms. Therefore, the Hessian matrix is positive definite, and the function is convex.

Problem 6.3

Compute the derivates of the least squares loss $L[\phi]$ with respect to the parameters ϕ_0 and ϕ_1 for the Gabor model (Equation 6.8),

$$\text{f}[x, \boldsymbol{\phi}] = \sin[\phi_0 + 0.06 \cdot \phi_1 x] \cdot \exp\left(-\frac{(\phi_0 + 0.06 \cdot \phi_1 x)^2}{32.0}\right) \tag{6.8}$$

The least squares loss function for I training examples is defined as:

$$L = \sum_{i=1}^{I}(\sin[\phi_0 + 0.06 \cdot \phi_1 x_i] \cdot \exp\left(-\frac{(\phi_0 + 0.06 \cdot \phi_1 x_i)^2}{32.0}\right) - y_i)^2$$

Taking the derivative of L with respect to ϕ_0, we get:

$$
\begin{aligned}
\frac{\partial L}{\partial \phi_0} =& 2\sum_{i=1}^{I}\left(\sin[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)-y_i\right)\\
&\times\left(\cos[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)\right.\\
&\left.-\frac{(\phi_0+0.06\cdot\phi_1 x_i)}{16.0}\cdot\sin[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)\right)
\end{aligned}
$$

Taking the derivative of L with respect to ϕ_1, we get:

$$
\begin{aligned}
\frac{\partial L}{\partial \phi_1} =& 2\sum_{i=1}^{I}x_i\left(\sin[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)-y_i\right)\\
&\times\left(\cos[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)\right.\\
&\left.-\frac{0.06\cdot x_i(\phi_0+0.06\cdot\phi_1 x_i)}{16.0}\cdot\sin[\phi_0+0.06\cdot\phi_1 x_i]\cdot\exp\left(-\frac{(\phi_0+0.06\cdot\phi_1 x_i)^2}{32.0}\right)\right)
\end{aligned}
$$

Problem 6.4

The logistic regression model uses a linear function to assign an input $\mathbf{x}$ to one of two classes $y \in \{0,1\}$. For a 1D input and a 1D output, it has two parameters ϕ_0 and ϕ_1, and the model is given by:

$$Pr(y=1|x) = \text{sig}[\phi_0+\phi_1 x],$$

where sig[•] is the logistic sigmoid function:

$$\text{sig}[z] = \frac{1}{1+\exp(-z)}.$$

Plot y against x for this model for different values of ϕ_0 and ϕ_1 and explain the qualitative meaning of each parameter. What is a suitable loss function for this model? Compute the derivatives of this loss function with respect to the parameters. Generate ten data points from a normal distribution with mean -1 and standard deviation 1 and assign them the label $y=0$. Generate another ten data points from a normal distribution with mean 1 ad standard deviation 1 and assign these the label $y=1$. Plot the loss as a heatmap in terms of the two parameters ϕ_0 and ϕ_1. Is this loss function convex? How could you prove this?

Answer given

Problem 6.5

Compute the derivates of the least squares loss with respect to the ten parameters of the simple neural network introduced in Equation 3.1:

$$\mathrm{f}[x,\phi] = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x] \tag{3.1}$$

Think carefully about what the derivative of the ReLU function a[•] will be.

The derivate of the least squares loss for the function $\mathrm{f}[x,\phi]$ is given by:

$$\frac{\partial L}{\partial \phi_j} = -2\sum_i (y - \mathrm{f}[x_i,\phi])\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \phi_j}$$

The derivate of the ReLU is zero if the input z is less than zero and one if the input z is greater than zero, which we can write as $\mathbb{I}[z > 0]$. The derivate terms can therefore be written as:

$$\begin{aligned}
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \phi_0} &= 1\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \phi_1} &= \mathrm{a}[\theta_{10} + \theta_{11}x_i]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \phi_2} &= \mathrm{a}[\theta_{20} + \theta_{21}x_i]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \phi_3} &= \mathrm{a}[\theta_{30} + \theta_{31}x_i]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{10}} &= \phi_1 \cdot \mathbb{I}[\theta_{10} + \theta_{11}x_i > 0]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{11}} &= \phi_1 \cdot x_i \cdot \mathbb{I}[\theta_{10} + \theta_{11}x_i > 0]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{20}} &= \phi_2 \cdot \mathbb{I}[\theta_{20} + \theta_{21}x_i > 0]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{21}} &= \phi_2 \cdot x_i \cdot \mathbb{I}[\theta_{20} + \theta_{21}x_i > 0]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{30}} &= \phi_3 \cdot \mathbb{I}[\theta_{30} + \theta_{31}x_i > 0]\\
\frac{\partial \mathrm{f}[x_i,\phi]}{\partial \theta_{31}} &= \phi_3 \cdot x_i \cdot \mathbb{I}[\theta_{30} + \theta_{31}x_i > 0]
\end{aligned}$$

Problem 6.6

> Which of the functions in figure 6.11 is convex? Justify your answer. Characterize each of the points 1-7 as a local minimum, global minimum, or neither.

Only curve b is convex. Points 1, and 3 are local minima. Points 2, 5, 6 and global minima, and points 3, 4, and 7 are neither.

Problem 6.7

> The gradient descent trajectory for path 1 in figure 6.5a oscillates back and forth inefficiently as it moves down the valley toward the minimum. It's also notable that it turns at right angles to the previous direction at each step. Provide a qualitative explanation for these phenomena. Propose a solution that might help prevent this behavior

The gradient descent trajectory for path 1 in figure 6.5a shows gradient descent with line search. The gradient at each step points directly downhill with respect to the contour lines and if the slope was still moving downhill at all in the current direction, then we should continue moving forward in the previous stage. This results in the direction turning at right angles.

The oscillation happens because the trajectory slightly overshoots the center of a descending valley. It has to turn at right angles and then overshoots again in the other direction and so on. This can be prevented by using a smaller learning rate or incorporating momentum into the optimization algorithm.

Problem 6.8

> Can (non-stochastic) gradient descent with a *fixed* learning rate escape local minima?

Yes: the distance moved depends on the gradient at the current point in the learning rate. The movement is agnostic to whether it crosses from valley to valley. In practice, the learning rate is very small, so this is unlikely to happen.

Problem 6.9

> We run the stochastic gradient descent algorithm for 1,000 iterations on a dataset of size 100 with a batch size of 20. How many epochs are we running the algorithm for?

We are running the algorithm for 5 epochs. An epoch is defined as one pass through the entire dataset. (1000 iterations / 100 samples) = 10 iterations per sample. Since the batch size is 20, we need (100 / 20) 5 iterations to pass through the entire dataset.

Problem 6.10

Show that the momentum term $\mathbf{m}_t$ (Equation 6.11) is an infinite weighted sum of the gradients at the previous iterations and derive an expression for the coefficients (weights) of that sum.

$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1-\beta) \cdot \sum_{i \in \text{batch}} \frac{\partial l_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}} \tag{6.11}$$

Equation 6.11 can be rewritten as:

$$\mathbf{m}_{t+1} = \beta \cdot \mathbf{m}_t + (1-\beta) \cdot \nabla l_t$$

Substituting $\mathbf{m}_t$ from the previous step, we get:

$$\mathbf{m}_{t+1} = \beta(\beta \cdot \mathbf{m}_{t-1} + (1-\beta) \cdot \nabla l_{t-1}) + (1-\beta) \cdot \nabla l_t$$

Continuing this process recursively, we can express $\mathbf{m}_{t+1}$ as an infinite series of the past gradients, each multiplied by a coefficient that diminishes exponentially:

$$\mathbf{m}_{t+1} = (1-\beta) \cdot \nabla l_t + \beta(1-\beta) \cdot \nabla l_{t-1} + \beta^2(1-\beta) \cdot \nabla l_{t-2} + \dots$$

This can be generalised as:

$$\mathbf{m}_{t+1} = (1-\beta) \sum_{i=0}^{t} \beta^i \cdot \nabla l_{t-i}$$

Where the coefficient for each gradient ∇l_{t-i} is $(1-\beta)\beta^i$, indicating that each subsequent gradient contributes less to the momentum term. This weighting creates an "exponential moving average" of the gradients, where more recent gradients have a higher influence on the momentum term than older ones.

Problem 6.11

What dimensions will the Hessian have if the model has one million parameters?

The Hessian matrix will have dimensions $10^6 \times 10^6$.