Chair of Decision Sciences and Systems
TUM School of Computation, Information and Technology
Technical University of Munich

Business Analytics & Machine Learning
Tutorial sheet 12: SGD and Neural Networks – Solution

Prof. Dr. Martin Bichler, Prof. Dr. Jalal Etesami
Julius Durmann, Markus Ewert, Johannes Knörr, Yutong Chao
January 30, 2024

Exercise T12.1 *Linear Neural Network*

This subsection is regarding linear networks. For input $x \in \mathbb{R}^{d_0}$, a deep linear network $F : \mathbb{R}^{d_0} \to \mathbb{R}$ of depth K will output $F(x) = W_K W_{K-1} ... W_1 x$, where each W_j is a matrix of appropriate dimension. We aim to train F to minimize the mean squared error loss on predicting real-valued scalar labels y. The loss is specified by

$$l(F) = \frac{1}{n} \sum_{i=1}^{n} (y_i - F(x_i))^2.$$

where $\{(x_i, y_i)\}_{i=1,...n}$ is our dataset.

1. Determine whether the following statement is true or false.
   - For $K = 1$, we recover the linear regression (with no bias term).
   - For $K = 2$, if there exists a pair of matrix W_1, W_2 that minimizes l, then there are infinite pairs of matrices that minimizes l.
   - This network with increasing depth K doesn't allow one to model more complex relationship between x and y.
   - $W_K \in \mathbb{R}^{d_1 \times d_2}$ can be a matrix ($d_1, d_2 > 1$).
2. You plan to train this model with stochastic gradient descent and batch size 1. In each batch, you minimize $l_x(F) = (y - F(x))^2$, for a fixed data point x. For simplicity, suppose $K = 3$ and W_3 is a scalar. Then, what is $\frac{\partial l_x}{\partial W_3}$?

Solution

1.
   - True. From their definition.
   - True. If W_1 and W_2 minimizes l, so AW_1 and $W_2 A^{-1}$ also minimize it, where A is an invertible matrix.
   - True. Since for any K, the function class remains the same as linear regression. It always only contains functions of the form $\omega^T x$.
   - False. Because the output is a scalar. W_K must be a row vector.
2. We denotes $z_3(x) = W_3 W_2 W_1 x$ and $g(x) = y - z_3(x)$. By applying chain rules, we have,

$$\frac{\partial l_x}{\partial W_3} = \frac{\partial l_x}{\partial g(x)} \frac{\partial g(x)}{\partial z_3(x)} \frac{\partial z_3(x)}{\partial W_3} = -2(y - F(x)) W_2 W_1 x.$$

Exercise T12.2 *Gradients of a fully connected neural network*

Consider a fully connected neural network, which consists of

- an input layer (l=0) representing two-dimentional data points

$$x = a^{[0]} = \left(a_1^{[0]}, a_2^{[0]}\right) \in \mathbb{R}^2$$

- a hidden layer (l=1) with 2 nodes, each with a sigmoid activation function $g_1^{[1]} \equiv \sigma, g_2^{[1]} \equiv \sigma$
- an output layer (l=2) with one node with a sigmoid activation function, i.e. $g^{[2]} \equiv \sigma$
- the weight matrix and bias between the input layer and the hidden layer are $W^{[1]} \in \mathbb{R}^{2\times 2}$ and $b^{[1]} \in \mathbb{R}^{1\times 2}$
- the weight matrix and bias between the hidden layer and the output layer are $W^{[2]} \in \mathbb{R}^{2\times 1}$ and $b^{[2]} \in \mathbb{R}^{1\times 1}$

The loss function is chosen to be the *cross-entropy loss*

$$\ell(y, \hat{y}) = -[y \ln(\hat{y}) + (1-y)\ln(1-\hat{y})]$$

a) How many trainable parameters does it have?

b) Write $\hat{Y}$ as a function of X (use matrix notation).

c) Compute the empirical risk $\mathcal{L}$ for the following data points and initial weights

$$X = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$W^{[1]} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad b^{[1]} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad W^{[2]} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad b^{[2]} = \begin{pmatrix} 0 \end{pmatrix}$$

d) Compute the partial derivatives of $\mathcal{L}$ w.r.t. all trainable parameters.

e) Perform one update step of gradient descent using a learning rate of $\alpha = 1$.

f) Compute the empirical risk $\mathcal{L}$ for the data (X, Y) from c) with the updated weights. Discuss the result!

Solution

a) The neural network has 9 trainable parameters consisting of 6 weights and 3 biases.

b) For a data matrix with n data points $X \in \mathbb{R}^{n\times 2}$ and the n-dimensional one-vector $\vec{1} \in \{1\}^n$, the output of the neural network can be written as

$$\hat{Y} = g^{[2]}\left(g^{[1]}\left(X \cdot W^{[1]} + \vec{1} \cdot b^{[1]}\right) \cdot W^{[2]} + \vec{1} \cdot b^{[2]}\right).$$

c) Let

$$\begin{aligned} Z^{[1]} &:= X \cdot W^{[1]} + \vec{1} \cdot b^{[1]} \\ A^{[1]} &:= \sigma\left(Z^{[1]}\right) \\ Z^{[2]} &:= A^{[1]} \cdot W^{[2]} + \vec{1} \cdot b^{[2]} \\ A^{[2]} &:= \sigma\left(Z^{[2]}\right) \end{aligned}$$

be auxiliary variables, where σ is the component-wise sigmoid function. $Z^{[k]}$ describes the incoming signals of layer k, right before the activation function, whereas $A^{[k]}$ describes the outgoing signal of layer k, right after the activation function. These auxiliary variables are saved and used to compute the gradient during the backpropagation later. The final outgoing signal is equal to the predicted output, i.e. $A^{[2]} = \hat{Y}$.

In order to compute $\mathcal{L}(Y, \hat{Y})$, the input X needs to be propagated iteratively through each layer of the neural network (forward-pass), which, at the end, yields the output $\hat{Y}$.

$$
\begin{aligned}
Z^{[1]} &= X \cdot W^{[1]} + \vec{1} \cdot b^{[1]} = X, \\
A^{[1]} &= \begin{pmatrix} \sigma(X_{11}) & \sigma(X_{12}) \\ \sigma(X_{21}) & \sigma(X_{22}) \\ \sigma(X_{31}) & \sigma(X_{32}) \\ \sigma(X_{41}) & \sigma(X_{42}) \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.731 \\ 0.731 & 0.5 \\ 0.731 & 0.731 \end{pmatrix}, \\
Z^{[2]} &= \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.731 \\ 0.731 & 0.5 \\ 0.731 & 0.731 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.231 \\ 0.231 \\ 0 \end{pmatrix} \\
A^{[2]} &= \begin{pmatrix} \sigma(0) \\ \sigma(-0.231) \\ \sigma(0.231) \\ \sigma(0) \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.443 \\ 0.557 \\ 0.5 \end{pmatrix}. \\
\hat{Y} &= A^{[2]}
\end{aligned}
$$

The empirical risk can then be computed as

$$
\begin{aligned}
\mathcal{L}(Y, \hat{Y}) &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \ell\left(Y_i, \hat{Y}_i\right) \\
&= \tfrac{1}{4} \cdot (\\
&\quad -(0 \cdot \ln(0.5) + (1-0) \cdot \ln(1-0.5)) \\
&\quad -(1 \cdot \ln(0.443) + (1-1) \cdot \ln(1-0.443)) \\
&\quad -(1 \cdot \ln(0.557) + (1-1) \cdot \ln(1-0.557)) \\
&\quad -(1 \cdot \ln(0.5) + (1-1) \cdot \ln(1-0.5))) \\
&\approx 0.25 \cdot (0.693 + 0.814 + 0.585 + 0.693) \\
&\approx 0.696
\end{aligned}
$$

d) Trainable parameters are $W^{[1]}, W^{[2]}, b^{[1]}$ and $b^{[2]}$, and derivatives of the empirical loss w.r.t. these parameters can be computed via chain-rule.

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^{[2]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial W^{[2]}} \\
\frac{\partial \mathcal{L}}{\partial b^{[2]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial b^{[2]}} \\
\frac{\partial \mathcal{L}}{\partial W^{[1]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial A^{[1]}} \cdot \frac{\partial A^{[1]}}{\partial Z^{[1]}} \cdot \frac{\partial Z^{[1]}}{\partial W^{[1]}} \\
\frac{\partial \mathcal{L}}{\partial b^{[1]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial A^{[1]}} \cdot \frac{\partial A^{[1]}}{\partial Z^{[1]}} \cdot \frac{\partial Z^{[1]}}{\partial b^{[1]}}
\end{aligned}
$$

Since the cross-entropy loss and the sigmoid activation functions for the output layer are component-wise operations, they can be combined to a single term. Consider the j-th component of both terms:

$$
\begin{aligned}
\frac{\partial \ell(Y_j, A_j^{[2]})}{\partial A_j^{[2]}} \cdot \frac{\partial A_j^{[2]}}{\partial Z_j^{[2]}} &= \left(\frac{1-Y_j}{1-A_j^{[2]}} - \frac{Y_j}{A_j^{[2]}} \right) \cdot \left(A_j^{[2]} \cdot (1 - A_j^{[2]}) \right) \\
&= \left(\frac{(1-Y_j) \cdot A_j^{[2]}}{(1-A_j^{[2]}) \cdot A_j^{[2]}} - \frac{Y_j \cdot (1-A_j^{[2]})}{A_j^{[2]} \cdot (1-A_j^{[2]})} \right) \cdot \left(A_j^{[2]} \cdot (1 - A_j^{[2]}) \right) \\
&= (1 - Y_j) \cdot A^{[2]} - Y_j \cdot (1 - A^{[2]}) \\
&= A_j^{[2]} - Y_j \cdot A_j^{[2]} - Y_j + Y_j \cdot A_j^{[2]} \\
&= A_j^{[2]} - Y_j
\end{aligned}
$$

This results in

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} &= \frac{1}{4} \cdot \vec{1}^T \cdot \text{diag} \left[\left(A_j^{[2]} - Y_j \right)_{j \in \{1,2,3,4\}} \right] \\
&= 0.25 \cdot \begin{pmatrix} 0.5 - 0 & 0.443 - 1 & 0.557 - 1 & 0.5 - 1 \end{pmatrix} \\
&= \begin{pmatrix} 0.125 & -0.139 & -0.111 & -0.125 \end{pmatrix} \in \mathbb{R}^{1 \times 4}
\end{aligned}
$$

Considering the (linear) term $Z^{[2]} := A^{[1]} \cdot W^{[2]} + \vec{1} \cdot b^{[2]}$, the derivatives can be obtained by $\frac{\partial Z^{[2]}}{\partial W^{[2]}} = A^{[1]}$ and $\frac{\partial Z^{[2]}}{\partial b^{[2]}} = \vec{1}$. This leads to

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^{[2]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial W^{[2]}} \\
&= \frac{1}{4} \cdot \vec{1}^T \cdot \text{diag} \left[\left(A_j^{[2]} - Y_j \right)_{j \in \{1,2,3,4\}} \right] \cdot A^{[1]} \\
&= \begin{pmatrix} 0.125 & -0.139 & -0.111 & -0.125 \end{pmatrix} \cdot \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.731 \\ 0.731 & 0.5 \\ 0.731 & 0.731 \end{pmatrix} \\
&= \begin{pmatrix} -0.179 & -0.186 \end{pmatrix} \in \mathbb{R}^{1 \times 2}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b^{[2]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial b^{[2]}} \\
&= \begin{pmatrix} 0.125 & -0.139 & -0.111 & -0.125 \end{pmatrix} \cdot \vec{1} \\
&= -0.25 \in \mathbb{R}^{1 \times 1}
\end{aligned}
$$

From the definitions of $Z^{[2]}$ and $A^{[1]}$, the respective derivatives result in $\frac{\partial Z^{[2]}}{\partial A^{[1]}} = W^{[2]}$ (linear term), and $\frac{\partial A^{[1]}}{\partial Z^{[1]}} = \left(A_{ij}^{[1]} \cdot (1 - A_{ij}^{[1]})\right)_{i \in \{1,2,3,4\}, j \in \{1,2\}}$ (component-wise sigmoid). Finally, this leads to

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^{[1]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial A^{[1]}} \cdot \frac{\partial A^{[1]}}{\partial Z^{[1]}} \cdot \frac{\partial Z^{[1]}}{\partial W^{[1]}} \\
&= \left(\left(A^{[2]} - Y\right) \cdot \left(W^{[2]}\right)^T \star A^{[1]} \star \left(1 - A^{[1]}\right)\right)^T \cdot X \\
&= \ldots = \\
&= \begin{pmatrix} -0.046 & -0.059 \\ 0.052 & 0.052 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b^{[1]}} &= \frac{\partial \mathcal{L}}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}} \cdot \frac{\partial Z^{[2]}}{\partial A^{[1]}} \cdot \frac{\partial A^{[1]}}{\partial Z^{[1]}} \cdot \frac{\partial Z^{[1]}}{\partial b^{[1]}} \\
&= \left(\left(A^{[2]} - Y\right) \cdot \left(W^{[2]}\right)^T \star A^{[1]} \star \left(1 - A^{[1]}\right)\right)^T \cdot \vec{1} \\
&= \ldots = \\
&= \begin{pmatrix} -0.050 & 0.048 \end{pmatrix}
\end{aligned}$$

e) With the computed derivatives from 1d), updating weights and biases is done according to the gradient descent update rule:

$$
\begin{aligned}
W^{[2]}_{new} &= W^{[2]} - \alpha \cdot \left(\frac{\partial \mathcal{L}}{\partial W^{[2]}}\right)^T \\
&= \begin{pmatrix} 1 \\ -1 \end{pmatrix} - 1.0 \cdot \begin{pmatrix} -0.179 \\ -0.186 \end{pmatrix} = \begin{pmatrix} 1.179 \\ -0.814 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
b^{[2]}_{new} &= b^{[2]} - \alpha \cdot \left(\frac{\partial \mathcal{L}}{\partial b^{[2]}}\right)^T \\
&= \begin{pmatrix} 0 \end{pmatrix} - 1.0 \cdot \begin{pmatrix} -0.25 \end{pmatrix} = \begin{pmatrix} 0.25 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
W^{[1]}_{new} &= W^{[1]} - \alpha \cdot \left(\frac{\partial \mathcal{L}}{\partial W^{[1]}}\right)^T \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - 1.0 \cdot \begin{pmatrix} -0.046 & 0.052 \\ -0.059 & 0.052 \end{pmatrix} = \begin{pmatrix} 1.046 & -0.052 \\ 0.059 & 0.948 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
b^{[1]}_{new} &= b^{[1]} - \alpha \cdot \left(\frac{\partial \mathcal{L}}{\partial b^{[1]}}\right)^T \\
&= \begin{pmatrix} 0 & 0 \end{pmatrix} - 1.0 \cdot \begin{pmatrix} -0.050 & 0.048 \end{pmatrix} = \begin{pmatrix} 0.05 & -0.048 \end{pmatrix}
\end{aligned}
$$

f) Another forward-pass yields

$$\hat{Y}_{new} = g^{[2]}\left(g^{[1]}\left(X \cdot W^{[1]}_{new} + \vec{1} \cdot b^{[1]}_{new}\right) \cdot W^{[2]}_{new} + \vec{1} \cdot b^{[2]}_{new}\right)$$

$$\mathcal{L}(Y, \hat{Y}_{new}) = ... = 0.585$$

As expected, the empirical risk is lower than before. If the learning rate alpha is not sufficiently small, however, it can happen that the empirical risk increases after a weight update.