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

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 $\frac{\partial l_x}{\partial W_3}$?

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!