Machine Learning Exercise Sheet 8

Deep Learning II

In-class Exercises

See the recording of the in-class exercise for the discussion of the code in the notebook.

Problem 1: See notebook `exercise_inclass_08_pytorch.ipynb` on Moodle.

Homework

Problem 2: You are trying to solve a regression task and you want to choose between two approaches:

1. A simple linear regression model.
2. A feed forward neural network $f_{\boldsymbol{W}}(\boldsymbol{x})$ with L hidden layers, where each hidden layer $l \in \{1, ..., L\}$ has a weight matrix $\boldsymbol{W}_l \in \mathbb{R}^{D \times D}$ and a ReLU activation function. The output layer has a weight matrix $\boldsymbol{W}_{L+1} \in \mathbb{R}^{D \times 1}$ and no activation function.

In both models, there are no bias terms.

Your dataset $\mathcal{D}$ contains data points with nonnegative features $\boldsymbol{x}_n$ and the target y_n is continuous:

$$\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N, \qquad \boldsymbol{x}_n \in \mathbb{R}^D_{\geq 0}, \qquad y_n \in \mathbb{R}$$

Let $\boldsymbol{w}^*_{LS} \in \mathbb{R}^D$ be the optimal weights for the linear regression model corresponding to a *global* minimum of the following least squares optimization problem:

$$\boldsymbol{w}^*_{LS} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^D} \mathcal{L}_{LS}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w} \in \mathbb{R}^D} \frac{1}{2} \sum_{n=1}^N (\boldsymbol{w}^T \boldsymbol{x}_n - y_n)^2$$

Let $\boldsymbol{W}^*_{NN} = \{\boldsymbol{W}^*_1, \ldots, \boldsymbol{W}^*_{L+1}\}$ be the optimal weights for the neural network corresponding to a *global* minimum of the following optimization problem:

$$\boldsymbol{W}^*_{NN} = \arg\min_{\boldsymbol{W}} \mathcal{L}_{NN}(\boldsymbol{W}) = \arg\min_{\boldsymbol{W}} \frac{1}{2} \sum_{n=1}^N (f_{\boldsymbol{W}}(\boldsymbol{x}_n) - y_n)^2$$

a) Assume that the optimal $\boldsymbol{W}^*_{NN}$ you obtain are non-negative.
What will the relation $(<, \leq, =, \geq, >)$ between the neural network loss $\mathcal{L}_{NN}(\boldsymbol{W}^*_{NN})$ and the linear regression loss $\mathcal{L}_{LS}(\boldsymbol{w}^*_{LS})$ be? Provide a mathematical argument to justify your answer.

Upload a single PDF file with your homework solution to Moodle by 15.12.2021, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

Note that for any non-negative $\boldsymbol{x}$ and any non-negative $\boldsymbol{W}$ it holds $\text{ReLU}(\boldsymbol{xW}) = \boldsymbol{xW}$.

Therefore, since our data points have non-negative features $\boldsymbol{x}_i$ and the optimal weights $\boldsymbol{W}^*_{NN}$ are non-negative, every ReLU layer is equivalent to a linear layer when plugging in the optimal weights. This means we can write

$$\begin{aligned} f_{\boldsymbol{W}^*_{NN}}(\boldsymbol{x}_i) &= ReLU(ReLU(ReLU(\boldsymbol{x}_i^T\boldsymbol{W}_1^*)\boldsymbol{W}_2^*)\cdots\boldsymbol{W}_L^*)\boldsymbol{W}_{L+1}^* \\ &= \boldsymbol{x}_i^T\boldsymbol{W}_1^*\boldsymbol{W}_2^*\cdots\boldsymbol{W}_{L+1}^* \\ &= \boldsymbol{x}_i^T\boldsymbol{w}_{NN}^* \end{aligned}$$

where we defined $\boldsymbol{w}^*_{NN} = \boldsymbol{W}_1^*\boldsymbol{W}_2^*\cdots\boldsymbol{W}_{L+1}^*$. From this we can see that the neural network with optimal weights behaves like a linear regression with a different set of weights $\boldsymbol{w}^*_{NN}$.

Note also that linear regression is a special case of the above neural network, i.e. for any weights $\boldsymbol{w}_{LS}$ you can find weights $\boldsymbol{W}_{NN}$ that produce the same output.

Given the above facts and since we the optimal weights correspond to a global minima we can conclude that $\mathcal{L}_{NN}(\boldsymbol{W}^*_{NN}) = \mathcal{L}_{LS}(\boldsymbol{w}^*_{LS})$ and the optimal weights found by solving the least squares optimization problem will be $\boldsymbol{w}^*_{LS} = \boldsymbol{w}^*_{NN}$.

b) In contrast to (a), now assume that the optimal weights $\boldsymbol{w}^*_{LS}$ you obtain are non-negative. What will the relation ($<, \leq, =, \geq, >$) between the linear regression loss $\mathcal{L}_{LS}(\boldsymbol{w}^*_{LS})$ and the neural network loss $\mathcal{L}_{NN}(\boldsymbol{W}^*_{NN})$ be? Provide a mathematical argument to justify your answer.

As stated in (a) linear regression is a special case of the above neural network, i.e. for any weights $\boldsymbol{w}_{LS}$ you can find weights $\boldsymbol{W}_{NN}$ that produce the same output. That is, everything that can be learned with a linear regression can be learned equally well with a neural network.

However, the reverse direction doesn't hold, since in principle neural networks can learn more complicated functions compared to linear regression. Moreover, the given fact that $\boldsymbol{w}^*_{LS}$ are non-negative does not tell us anything about the optimal weights of the neural network $\boldsymbol{W}^*_{NN}$.

Therefore it holds $\mathcal{L}_{NN}(\boldsymbol{W}^*_{NN}) \leq \mathcal{L}_{LS}(\boldsymbol{w}^*_{LS})$ since the neural network can potentially find a better fit for the data (e.g. by taking advantage of non-linearity).

Problem 3: Load the notebook `exercise_08_notebook.ipynb` from Moodle. Fill in the missing code and run the notebook. Export (download) the evaluated notebook as PDF and add it to your submission.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks, consult the Jupyter documentation. Instructions for converting the Jupyter notebooks to PDF are provided on Piazza.

The solution notebook is uploaded to Moodle.