STATISTICAL MACHINE LEARNING (WS2020)
SEMINAR ON ARTIFICIAL NEURAL NETWORKS

Assignment 1. a) Recall the negative log likelihood for the linear regression:

$$\mathcal{L}(\boldsymbol{w}) = \frac{m}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\right) \tag{1}$$

where:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1n} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_{m1} & \dots & x_{mn} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_m^T \end{pmatrix} \tag{2}$$

is the input matrix (input dimension n, the number of data samples m),

$$\boldsymbol{y} = \left(y_1, \dots, y_m\right)^T, \quad y_i \in \mathbb{R}^m \tag{3}$$

the target vector and

$$\boldsymbol{w} = \left(w_0, \dots, w_n\right)^T, \quad w_i \in \mathbb{R}^m \tag{4}$$

is the weight vector. Prove that the MLE solution is:

$$\boldsymbol{w}^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{y} \tag{5}$$

b) When L2 regularization is involved we get the following loss function:

$$\mathcal{L}(\boldsymbol{w}) = \left(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\right) + \lambda\boldsymbol{w}^T\boldsymbol{w}, \quad \lambda > 0 \tag{6}$$

Prove that the MLE solution is now:

$$\boldsymbol{w}^* = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\boldsymbol{y} \tag{7}$$

Hint: use matrix calculus identities for differentiation to simplify your computations.

Assignment 2. a) Give the backward message for the multinomial cross entropy layer for a single dataset sample:

$$l(\boldsymbol{p}, \boldsymbol{t}) = \sum_{j=1}^{K} t_j \log(p_j) \tag{8}$$

where $\boldsymbol{p} = \left(p_1, \dots, p_K\right)^T$ is an input vector and $\boldsymbol{t}$ a one-hot encoded target vector (for the target class $k \in \{1, \dots, K\}$ we have $t_k = 1$ and $t_l = 0$ for $l \neq k$).

b) Give the backward message for the softmax layer:

$$p_j(\boldsymbol{s}) = \frac{e^{s_j}}{\sum_{i=1}^{K} e^{s_i}}, \quad j \in \{1, \ldots, K\} \tag{9}$$

c) Give the backward message for the linear layer:

$$s_j(\boldsymbol{x}, \mathbf{W}) = \sum_{i=1}^{N} w_{ij} x_i, \quad j \in \{1, \ldots, K\} \tag{10}$$

where $\boldsymbol{x} = \big(x_1, \ldots, x_N\big)^T$ is an input vector and w_{ij} a weight of the connection between input i and output j. Give also the parameter message $\frac{\partial s_j}{\partial w_{ij}}$.

d) Combine all layers to form the logistic regression model:

$$l(\boldsymbol{p}(\boldsymbol{s}(\boldsymbol{x}, \mathbf{W})), \boldsymbol{t}) \tag{11}$$

and use the previous results to get its gradient with respect to the parameters:

$$\frac{\partial l}{\partial w_{ij}}, \quad i \in \{1, \ldots, N\},\ j \in \{1, \ldots, K\} \tag{12}$$

Assignment 3. Consider the following neural network:

$$y_k(\boldsymbol{x}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}) = \sum_{j=1}^{M} w_{jk}^{(2)} \tanh\left(\sum_{i=1}^{N} w_{ij}^{(1)} x_i\right), \tag{13}$$

having outputs denoted $k \in \{1, \ldots, K\}$. Draw the network. Swapping positions of any two neurons in the first linear layer will not change the output of the network. Similarly changing sign of all input and output connection weights of a neuron in the same layer will not change the network's output (because $\tanh(-x) = -\tanh(x)$). Such configurations are called the *weight-space symmetries*. Compute the maximum number of networks preserving output for the same input based on the defined weight-space symmetries.

Assignment 4. Define backward messages for the following layers. Note that all outputs of these layers are computed independently so we can omit subscripts, i.e., $y(x) = y_j(x)$ for each layer output j.

a) Sigmoid:

$$y(x) = \frac{1}{1 + e^{-x}}. \tag{14}$$

Express the backward message $y'(x)$ using the forward message $y(x)$.

b) Hyperbolic tangent:

$$y(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{15}$$

Express the backward message $y'(x)$ using the forward message $y(x)$.

c) Rectified Linear Unit (ReLU):

$$y(x) = \max(0, x). \tag{16}$$

d) Parametric ReLU (PReLU):

$$y(x, a) = \max(0, x) + a \min(0, x). \tag{17}$$

Derive also the parameter message $\frac{\partial y(x,a)}{\partial a}$.