Neural network derivation

Machine Learning II
2021-2022 - UMONS
Souhaib Ben Taieb

Let $\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix}$ be a matrix of data points. $\boldsymbol{U} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \end{pmatrix} \in \mathbb{R}^{3\times 2}$, $\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \in \mathbb{R}^3$, $\boldsymbol{W} = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \in \mathbb{R}^{1\times 3}$, $c \in \mathbb{R}$ are neural network parameters.

Forward equations (Scalar form)

$$\begin{aligned} g_{ij} &= u_{j1}x_{i1} + u_{j2}x_{i2} + b_j \\ h_{ij} &= \tanh(g_{ij}) \\ z_i &= w_1 h_{i1} + w_2 h_{i2} + w_3 h_{i3} + c \\ o_i &= \sigma(z_i) \end{aligned}$$

Here, i indexes data points and j indexes hidden units, so $i \in \{1, \ldots, N\}$ and $j \in \{1, 2, 3\}$.

σ is the logistic function defined as

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

and hyperbolic tangent function tanh is defined as

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

Forward equations (Vector form)

$$\begin{aligned} \boldsymbol{G} &= \boldsymbol{X}\boldsymbol{U}^T + \mathbf{1}\boldsymbol{b}^T \qquad (\mathbf{1} \text{ is a vector of } 1's) \\ \boldsymbol{H} &= \tanh(\boldsymbol{G}) \\ \boldsymbol{z} &= \boldsymbol{H}\boldsymbol{W}^T + \mathbf{1}c \\ \boldsymbol{o} &= \sigma(\boldsymbol{z}) \end{aligned}$$

Computational graph

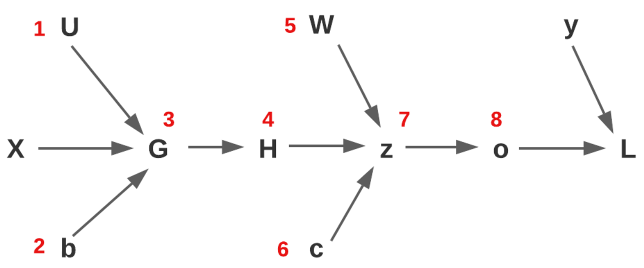

Figure 1: Computational graph

Cost function

$$
\begin{aligned}
\varepsilon(\boldsymbol{z}, \boldsymbol{y}) &= \frac{1}{N}\Big[\sum_{i=1}^{N} \mathcal{L}(z_i, y_i)\Big] \\
\mathcal{L}(z, y) &= y \log(\sigma(z)) + (1-y)\log(1-\sigma(z)) \\
&= y \log(1 + exp(-z)) + (1-y)\log(1 + exp(z))
\end{aligned}
$$

$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$ is the true label.

Backward equations (Scalar form)

$$
\begin{aligned}
\overline{\varepsilon} &= 1 \\
\overline{z_i} &= \overline{\varepsilon}\frac{\partial \varepsilon}{\partial z_i} = \frac{1}{N}(o_i - y_i) \\
\overline{w_j} &= \sum_{i=1}^{N} \overline{z_i}\frac{\partial z_i}{\partial w_j} = \sum_{i=1}^{N} \overline{z_i} h_{ij} \\
\overline{c} &= \sum_{i=1}^{N} \overline{z_i}\frac{\partial z_i}{\partial c} = \sum_{i=1}^{N} \overline{z_i} \\
\overline{h_{ij}} &= \overline{z_i}\frac{\partial z_i}{\partial h_{ij}} = \overline{z_i} w_j \\
\overline{g_{ij}} &= \overline{h_{ij}}\frac{\partial h_{ij}}{\partial g_{ij}} = \overline{h_{ij}}(1 - \tanh^2(g_{ij})) \qquad (As, \frac{\partial \tanh(x)}{\partial x} = 1 - \tanh^2 x) \\
\overline{u_{jk}} &= \sum_{i=1}^{N} \overline{g_{ij}}\frac{\partial g_{ij}}{\partial u_{jk}} = \sum_{i=1}^{N} \overline{g_{ij}} x_{ik} \\
\overline{b_j} &= \sum_{i=1}^{N} \overline{g_{ij}}\frac{\partial g_{ij}}{\partial b_j} = \sum_{i=1}^{N} \overline{g_{ij}}
\end{aligned}
$$

As above, i indexes data points and j indexes hidden units, so $i \in \{1, \dots, N\}$ and $j \in \{1, 2, 3\}$. In addition, k indexes the data dimension so $k \in \{1, 2\}$.

Backward equations (Vector form)

$$
\begin{aligned}
\overline{\boldsymbol{\varepsilon}} &= 1 \\
\overline{\boldsymbol{z}} &= \frac{1}{N}(\boldsymbol{o} - \boldsymbol{y}) \\
\overline{\boldsymbol{W}} &= \boldsymbol{H}^T \overline{\boldsymbol{z}} \\
\overline{c} &= \boldsymbol{z}^T \mathbf{1} \\
\overline{\boldsymbol{H}} &= \overline{\boldsymbol{z}} \boldsymbol{W} \\
\overline{\boldsymbol{G}} &= \overline{\boldsymbol{H}} \odot (1 - \tanh^2(\boldsymbol{G})) \\
\overline{\boldsymbol{U}} &= \overline{\boldsymbol{G}}^T \boldsymbol{X} \\
\overline{\boldsymbol{b}} &= \overline{\boldsymbol{G}}^T \mathbf{1}
\end{aligned}
$$