Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNIA Winter Term 2019/2020

Exercise Sheet 6

Philip Georgis [s8phgeor], Pauline Sander [s8pasand], Vilém Zouhar [vizo00001]

(Solutions)

Deadline: 5.1.2020

Exercises

Exercise 6.1 - Network scheme

Definitions:
$\mathbf{X} \in \mathbf{R}^{nx3}$ contains the input data (*age*, *class*, *survived*)
$\mathbf{y} \in \mathbf{R}^{nx1}$ contains the *ticket prices*
$\mathbf{b} \in 1^{nx1}$ a bias vector of ones (only one because they have the same shape for both layers)
$\mathbf{W}^{(1)} \in [0,1]^{4x4}$ weight matrix mapping from input to hidden layer
$\mathbf{W}^{(2)} \in [0,1]^{5x1}$ weight matrix mapping from hidden to output layer
concat($\mathbf{X}$,$\mathbf{y}$) - operation that appends vector $\mathbf{y}$ as a column to matrix $\mathbf{X}$

Formula:

$$\hat{y} = \mathbf{e}^1 = concat(ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}), \mathbf{b}) \times \mathbf{W}^{(2)} \tag{1}$$

$$Loss = \mathbf{L} = MSE(\hat{y}, \mathbf{y}) \tag{2}$$

$$= \frac{1}{n} \sum^{n} \left(concat(ReLU(concat(\mathbf{X}_n, \mathbf{b}_n) \times \mathbf{W}_n^{(1)}), \mathbf{b}_n) \times \mathbf{W}_n^{(2)} - \mathbf{y}_n \right)^2 \tag{3}$$

This gives us the following network scheme (see next page):

[1] This is the name in the computational graph in exercise 6.2.

$$[\textbf{\textit{ReIU}}\,(\,(\mathbf{X}^{(nx3)}\ \text{concat}\ \mathbf{b}^{(nx1)})\ \ x\ \mathbf{W}^{[1](4x4)})\qquad \text{concat}\ \mathbf{b}^{(nx1)}]\qquad x\ \mathbf{W}^{[2](5x1)}\qquad = \mathbf{e}^{(nx1)}$$

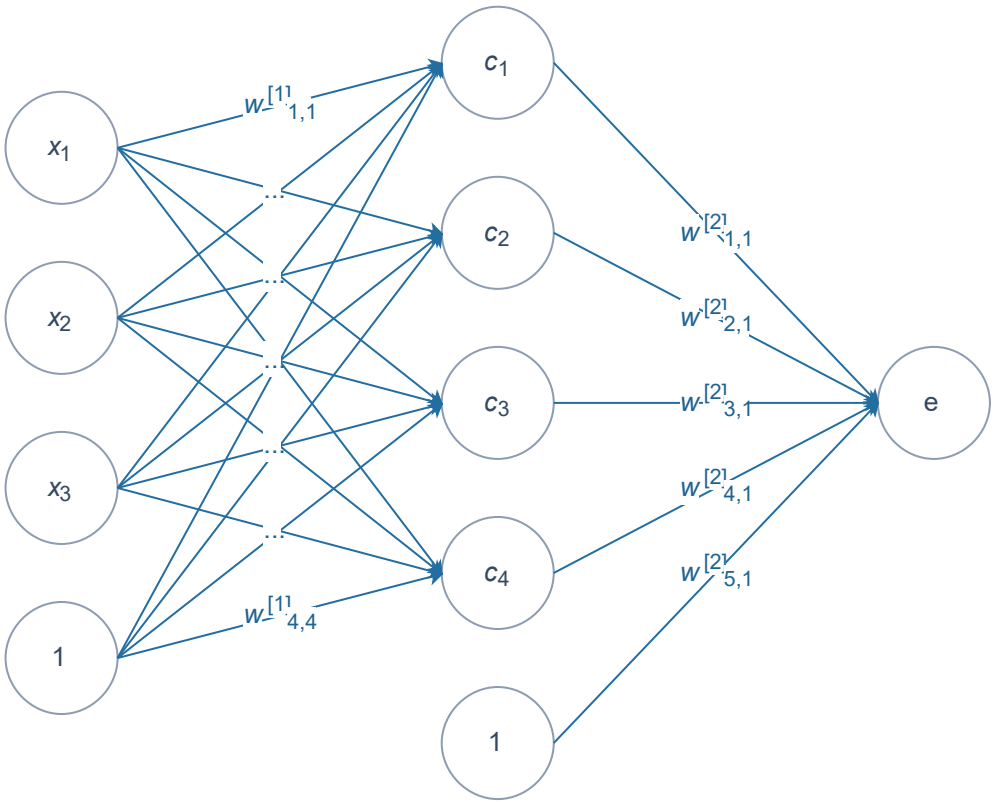
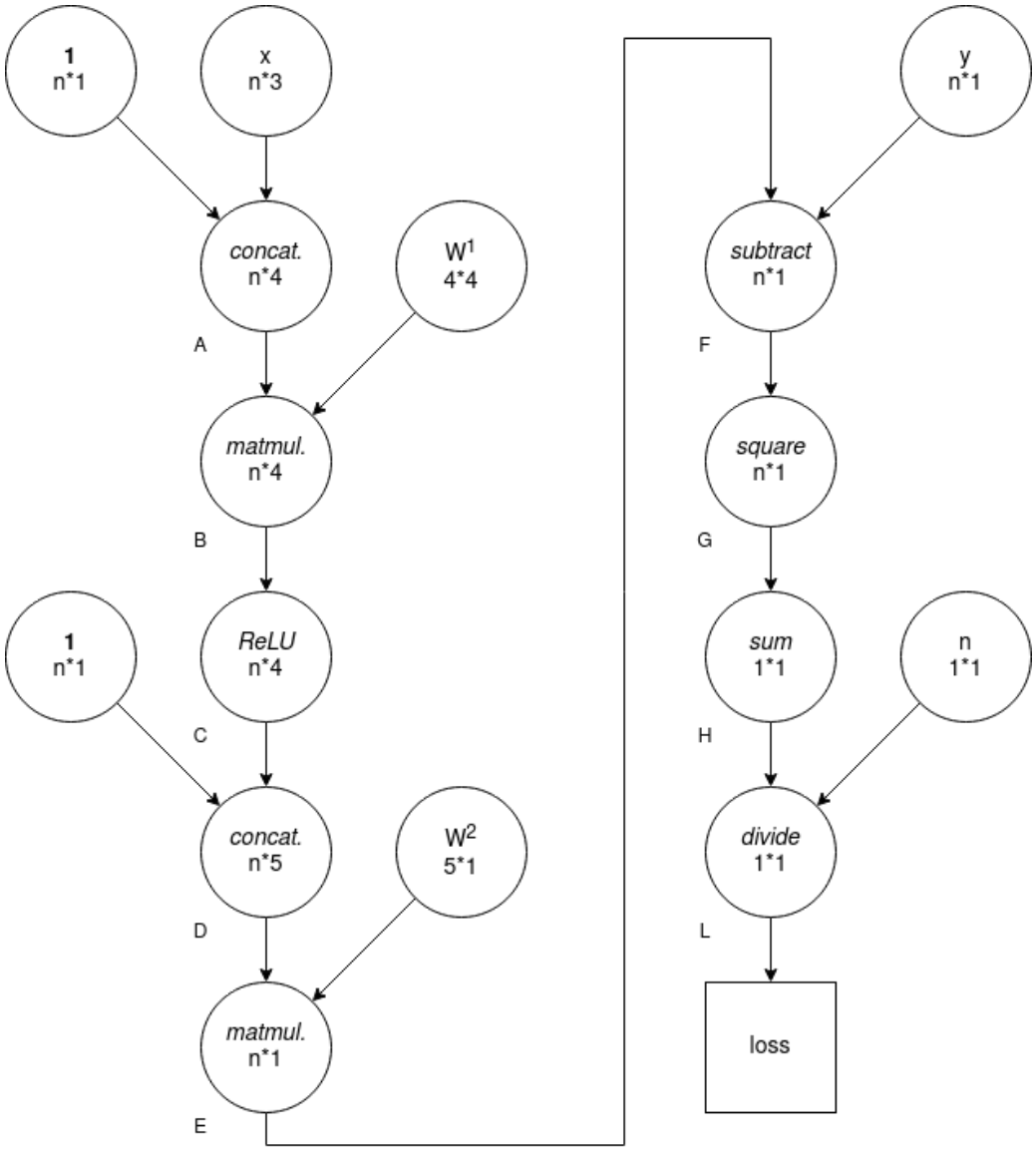


Figure 1: Network scheme

Exercise 6.2 - Computation graph



Exercise 6.3 - Backpropagate

Biases are simply part of the weight matrices. The forward propagation is vectorized.

Forward pass:

$$
\begin{aligned}
L^{(1x1)} &= \frac{H}{n} \\
H^{(1x1)} &= \sum_{k=1}^{n} G_k \\
G_k^{(1x1)} &= F_k^2 \\
F^{(nx1)} = E - \mathbf{y}^{nx1} &= concat(ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}), \mathbf{b}) \times \mathbf{W}^{(2)} - \mathbf{y} \\
E^{(nx1)} = D \times W^{(2)(5x1)} &= concat(ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}), \mathbf{b}) \times \mathbf{W}^{(2)} \\
D^{(nx5)} = concat(C, \mathbf{b}) &= concat(ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}), \mathbf{b}) \\
C^{(nx4)} = ReLU(B) &= ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}) \\
B^{(nx4)} = A \times W^{(1)(4x4)} &= concat(\mathbf{X}, \mathbf{b}) \times W^{(1)} \\
A^{(nx4)} = concat(\mathbf{X}, \mathbf{b}) &
\end{aligned}
$$

Backpropagation $W^{(2)}$

$$
\begin{aligned}
&\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial G}\frac{\partial G}{\partial F}\frac{\partial F}{\partial E}\frac{\partial E}{\partial w_j^{(2)}} \\
&\frac{\partial L}{\partial H} = \frac{1}{n} \\
&\frac{\partial H}{\partial G} = \sum^{n} \\
&\frac{\partial G}{\partial F} = 2 \cdot F^{(nx1)} = 2F^{T(1xn)} \\
&\frac{\partial F}{\partial E} = 1 \\
&\frac{\partial E}{\partial W^{(2)}} = D \\
&\Rightarrow \frac{\partial L}{\partial W^{(2)}} = \frac{1}{n}\sum^{n}[2 \cdot F^{T(1xn)} \cdot 1 \cdot D^{(nx5)}] = \frac{2}{n}\sum[F^T D]^{(1x5)} = \frac{2}{n}F^T D \\
&\Rightarrow \frac{2}{n}(\hat{y} - y)^{T(1\times n)}\left[concat(ReLU(concat(\mathbf{X}, \mathbf{b}) \times \mathbf{W}^{(1)}), \mathbf{b})\right]^{(n\times 5)}
\end{aligned}
$$

Backpropagation $w_{i,j}^{(1)}$

$$
\begin{aligned}
&\frac{\partial L}{\partial w_{i,j}^{(1)}} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial G}\frac{\partial G}{\partial F}\frac{\partial F}{\partial E}\frac{\partial E}{\partial D}\frac{\partial D}{\partial C}\frac{\partial C}{\partial B_{i,j}}\frac{\partial B_{i,j}}{\partial w_{i,j}^{(1)}} \\
&\frac{\partial E}{\partial D} = W^{(2)} \\
&\frac{\partial D}{\partial C} = thisdoesnotwork \\
&\frac{\partial C}{\partial B_{ij}} = R'(B)_{ij} = \begin{cases} 1 & B_{ij} > 0 \\ 0 & \text{else} \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial B_{ij}}{\partial w_{ij}^{(1)}} = A_{ij} \\
&\Rightarrow \frac{\partial L}{\partial W^{(1)}} = \\
&= \frac{1}{n} \sum_{k=1}^{n} \left[\left[[2 \cdot F \times 1 \times (W^{(2)})^T] \times I^4 \right] \odot R'(B) \right]_{k,*} \times A_{k,*} \\
&= \frac{2}{n} A^T \left[\left[F \times (W^{(2)})^T \times I^4 \right] \odot R'(B) \right] \\
&\Rightarrow \frac{2}{n} (x + J^{n,4})^{T,(4\times n)} \left[\left[(\hat{y} - y) \times (W^{(2)})^T \times I^4 \right]^{(n\times 4)} \odot R'\big((x + J^{n,4}) \times W^{(1)}\big)^{(n\times 4)} \right]^{(n\times 4)}
\end{aligned}
$$

Exercise 6.4 - PyTorch

Tensors are smart objects and the operations on them also produce the gradient and can track history. The partial gradient then has to be defined by the operation. The autograd then constructs a dynamic computational graph, which is an acyclic connected graph from this node with leaves as the input vectors. It is invoked using `backward`.