Programming with neural networks: Exercise sheet 4 SS 2020

University of Würzburg - Chair for Computer Science VI Prof. Dr. F. Doll, A. Hekalo, M. Krug

Exercise sheet: 4

Edited on June 14th

Task 1: Fully Connected

A weight matrix w , a bias vector b and an input x *of* a fully are given Connected Layer, as discussed in the lecture:

$$w = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

$$b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

(a) Calculate the forward pass

Solution:

$$y = x * w + b$$

$$y = \begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

$$y = \begin{bmatrix} 5 & 11 & 17 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

$$y = \begin{bmatrix} 6 & 13 & 20 \end{bmatrix}$$

(b) Calculate the vector δx . The following error vector $\delta y =$ is given $\begin{bmatrix} 0 & 0.5 & 1 \end{bmatrix}$

Solution:

$$\delta x = \delta y * w_T$$

$$w_T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$\delta x = \begin{bmatrix} 0 & 0.5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Page 4

5 6

$\delta x = \begin{bmatrix} 6.5 & 8 \end{bmatrix}$

(c) Calculate the weight updates ΔW and Δb for the above information.

Solution:

$$\Delta w = x^T \cdot \delta y$$

$$\Delta w = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0.5 & 1 \end{bmatrix}$$

$$\Delta w = \begin{bmatrix} 0 & 0.5 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$\Delta b = \delta y$$

Task 2: Convolution-2D

Let the convolution 2D layer be introduced as in the lecture.

(a) Change the filter f (with 2 channels)

$$f_0 = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 7 & 9 & 11 \\ 8 & 10 & 12 \end{bmatrix} \quad f_1 = \begin{bmatrix} 13 & 15 & 17 \\ 24 & 16 & 18 \end{bmatrix} \begin{bmatrix} 19 & 21 & 23 \\ 20 & 22 & 24 \end{bmatrix}$$

so that it can be used for the backward pass.

Solution: Swap channels and filters:

$$f_0 = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 13 & 15 & 17 \\ 24 & 16 & 18 \end{bmatrix} \quad f_1 = \begin{bmatrix} 7 & 9 & 11 \\ 8 & 10 & 12 \end{bmatrix} \begin{bmatrix} 19 & 21 & 23 \\ 20 & 22 & 24 \end{bmatrix}$$

Then rotate 180 degrees:

$$f_0 = \begin{bmatrix} 6 & 4 & 2 \\ 5 & 3 & 1 \end{bmatrix} \begin{bmatrix} 18 & 16 & 24 \\ 17 & 15 & 13 \end{bmatrix} \quad f_1 = \begin{bmatrix} 12 & 10 & 8 \\ 11 & 9 & 7 \end{bmatrix} \begin{bmatrix} 24 & 22 & 20 \\ 23 & 21 & 19 \end{bmatrix}$$

Shape of the input [20 , 18 , 5]
Shape of the filter [3 , 4 , 5 , 1337]
Stride: $s = 2$
Padding: $p = 3$

Solution:

$$o_x = floor\left(\frac{(x + 2p_x) - k_x}{stride_x}\right) + 1 = \frac{20 + 6 - 3}{2} + 1 = 12$$

$$o_y = floor\left(\frac{(y + 2p_y) - k_y}{stride_y}\right) + 1 = \frac{18 + 6 - 4}{2} + 1 = 11$$

$$o_c = 1337$$

Task 3: Recurrent networks

The following are the forward equations of a so-called peephole LSTM.
shows.

$$f_t = x_t \cdot W_f + c_{t\text{-}1} \cdot W_f + b_f$$
$$f_t = \sigma(f_t)$$

$$i_t = x_t \cdot W_i + c_{t-1} \cdot W_{Ri} + b_i$$
$$i_t = \sigma(i_t)$$

$$o_t = x_t \cdot W_o + c_{t\text{-}1} \cdot W_{Ro} + b_o$$
$$o_t = \sigma(o_t)$$

$$c_t = f_t \cdot c_{t\text{-}1} + i_t \cdot \tanh(x_t \cdot W_c + b_c)$$
$$h_t = \tanh(o_t \cdot c_t)$$

Solution: $\delta c_t = \delta h_t \cdot (1 - tanh^2(o_t \cdot c_t)) \cdot o_t + \delta c_{t+1} \cdot f_{t+1} + \delta o_{t+1} \cdot W^T_{Ro} + .delta.f_{t+1} \cdot W^T_{Rf} + \delta i_{t+1} * W^T_{Ri}$

(a) Calculate the equation for δc_t

Exercise 4: Reverse Mode Automatic Differentation

Given is the function $L = \cos(x_1 \cdot x_2) + x_1 \cdot x_2 + x_1$

(a) Calculate the gradient of the function L manually

Solution: $\nabla L = \begin{bmatrix} -\sin(x_1 \cdot x_2) \cdot x_2 + x_2 + 1 \\ \text{-Sin}(x_1 \cdot x_2) \cdot x_1 + x_1 \end{bmatrix}$

(b) Calculate the gradient of L at $(1, 2)$

Solution: $\nabla L = \begin{bmatrix} -\sin(1 \cdot 2) \cdot 2 + 2 + 1 \\ -\sin(1\ 2)\ 1 + 1 \end{bmatrix} = \begin{bmatrix} 1.19th \\ 0.09 \end{bmatrix}$

(c) Calculate the gradient of L at point $(1, 2)$ using reverse mode AD

Solution: Forward:

$$v_0 = x_1$$

$$v_1 = x_2$$

$$v_2 = v_0 \cdot v_1$$

$$v_3 = cos(v_2)$$

$$v_4 = v_3 + v_2 + v_0 = L$$

Backward:

$$\frac{\partial L}{\partial v_4} = 1$$

$$\frac{\partial L}{\partial v_3} = 1 \text{ x } 1$$

$$\frac{\partial L}{\partial v_1} = (-0.91 + 1)\, v_0 = 0.09 \cdot 1 = 0.09$$

$$\frac{\partial L}{\partial v_0} = (-0.91 + 1)\, v_1 + 1 = 1.19\text{th}$$

Exercise 5: Average Pooling

Let average pooling be defined so that the operation of pooling is the average of all Takes values and writes them to the output. Describe how to create the deltas in the Identify backward pass.

Solution:

For a filter with k entries, each δy *has* to be divided into k elements. There must therefore be a routine that is used for an output index, the Index of the upper left element of the filter matrix is determined. About this index you can then divide the deltas. Since every δx of several δy *makes* a contribution you should always update with + =, and at the beginning set all deltas to 0.