DEEP LEARNING IN COMPUTER VISION - EXERCISE 4

KARSTEN HERTH, FELIX HUMMEL, FELIX KAMMERLANDER, AND DAVID PALOSCH

Exercise 4.1a

Possible values for $x_1, \dots x_8$ are

$$\begin{aligned} x_1 &= 17, & x_2 &= 13, \\ x_3 &= 3, & x_4 &= 4, \\ x_5 &= 13, & x_6 &= 13, \\ x_7 &= 2, & x_8 &= 3. \end{aligned}$$

Exercise 4.1c

The `tf.nn.conv2d_transpose()` requires to define the output shape since it is not necessarily unique. For example, let the input shape for a convolution be

(a) 4×4,

(b) 3×3.

In both cases we use a filter of size 3×3, striding of 2 and the padding option SAME. Then, in both cases (a) and (b), performing a `conv2d()` yields an output of 2×2. Hence, if we want to go back using `tf.nn.conv2d_transpose()`, we have to tell tensorflow if we came from the 4×4 input or the 3×3 input.

The `tf.layers.conv2d_transpose()` uses the 4×4 output by default. See also our file ex_04_1_c.ipynb, where we implemented this example.

Exercise 4.3

Let $g : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, $A \in \mathbb{R}^{m \times n}$. We consider the composite function $E : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto g(Ax)$.

(a) The chainrule yields

$$\nabla E(x) = (E'(x))^T = (A^T g'(x))^T = \nabla g(x) A.$$

(b) Since the action of a convolutional layer is linear, it can be viewed as a mapping $x \mapsto Mx$ with a matrix M. At any point is the derivative of this mapping (which is necessary for the backpropagation algorithm) given by the transpose matrix M^T. This matrix M^T describes a transpose convolution.

Date: January 23, 2018.