DATA20001 Deep Learning

Exercise 1 **Due Tuesday November 7, before 12:00 PM (noon)**

Rules:

1. Return your solutions in Moodle by the deadline (which is usually just before the next exercise session).
2. The submission consists of a single PDF file containing your answers to all questions.
3. Add an estimate of how many hours you worked on the problems in the beginning of your report. This does not affect your grading in any way, we are just interested in feedback about exercises. Also other feedback is very welcome!
4. The content of the PDF is free form, e.g., LaTeX or Microsoft Office is OK. If you have done calculations by hand, you can also just return a scanned copy of this.
5. If something is unclear you can ask for clarifications in the Moodle discussion group.

---

1 Activation functions (1pt)

Motivation: activation functions and their derivatives are crucial when calculating the forward activation of a network and propagating errors of neural networks. Having a good understanding of these functions helps in programming and designing networks in the future.

(a) σ is the logistic sigmoid. Prove that $\tanh(x) = 2\sigma(2x) - 1$.

(b) Prove that $\sigma'(x) = \sigma(x)(1 - \sigma(x))$.

(c) On the basis of the two previous points, express $\tanh'$ as a function of σ'.

(d) What is the deriative of ReLU? Recall that the ReLU activation function is defined as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Highlight what is special about ReLU's derivative, and how it has been solved in practice according to literature.

2 Forming neural networks (1pt)

Motivation: neural networks tend to form logical operands within their neurons during training. Typically we don't care about that, but instead parameterize a large enough network and hope it catches all the relevant phenomenon within the data. Still, for particular problems is may be useful to have as small a network as possible.

Your task is to formulate the smallest possible network for the following data. Find out both the topology and the corresponding weights.

The table below shows the input x (three input nodes) and the corresponding desired output y, one example per line.

x_1	x_2	x_3	y
1	1	1	0
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

Hence, for the input $1, 1, 1$ the network should produce the output 0, and so on. Furthermore, the network is required to:

(1) Be feedforward

(2) Use sigmoid or ReLU activations.

(3) The nodes should include bias.