FACIT

TDA 231 Machine Learning: Homework 3
Goal: Feed-forward neural networks

Mikael Kågebäck
kageback@chalmers.se

Due Date: May 7, 2018

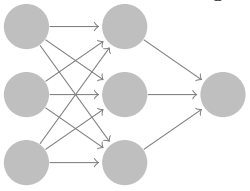
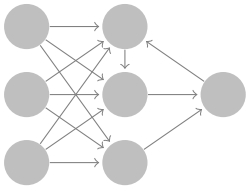
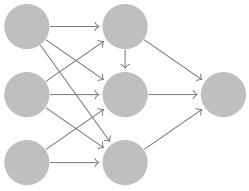
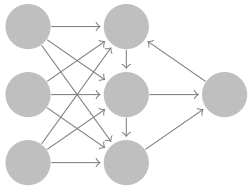
General guidelines:

1. All solutions to theoretical problems, and discussion regarding practical problems, should be submitted in a single file named *report.pdf*
2. All matlab files have to be submitted as a single zip file named *code.zip*.
3. The report should clearly indicate your name, personal number and email address
4. All datasets can be downloaded from the course website.
5. All plots, tables and additional information should be included in *report.pdf*
6. For questions regarding the homework contact Mikael Kågebäck (kageback@chalmers.se)

1 Theoretical problems

Problem 1.1 [Topological properties, 2 points]

(a) Which of the following neural networks are examples of feed-forward neural networks?

(1)  (2)  (3)  (4) 

(b) In general: Which topological properties must the network fulfill to be a feed-forward neural network?

Answer: (1) and (3)

Problem 1.2 [Committee, 2 points]

Brian wants to make his feed-forward network (with no hidden units) using a linear output neuron more powerful. He decides to combine the predictions of two networks by averaging them. The first network has

weights w_1 and the second network has weights w_2. The prediction of this committee for an example x is therefore:

$$y = \frac{1}{2}\mathbf{w}_1^T\mathbf{x} + \frac{1}{2}\mathbf{w}_2^T\mathbf{x}$$

Can we get the exact same predictions as this combination of networks by using a single feed-forward network (again with no hidden units) using a linear output neuron and weights $\mathbf{w}_3 = f(\mathbf{w}_1, \mathbf{w}_2)$, where $f(\mathbf{w}_1, \mathbf{w}_2)$ is some linear function? Prove your claim!

Problem 1.3 [Backpropagation - shallow network, 2 points]

Consider a neural network with only one training case with input $x = (x_1, x_2, \ldots, x_n)^T$ and correct output $t \in \{0, 1\}$. There is only one output neuron, which is linear, i.e. $y = w^T x$ (notice that there are no biases). The cost function is a simple squared error ($E = \frac{1}{2}(t-y)^2$). The network has no hidden units, so the inputs are directly connected to the output neuron with weights $w = (w_1, w_2, \ldots, w_n)^T$. We're in the process of training the neural network with the backpropagation algorithm. What will the algorithm add to w_i for the next iteration if we use a step size (also known as a learning rate) of λ?

Answer: $-\lambda(\mathbf{w}^T\mathbf{x} - t)x_i$

Problem 1.4 [Backpropagation, 4 points]

Derive general expressions for the partial derivatives of an error function E, surrounding a neuron j, in the feed-forward neural network depicted in Figure 1.

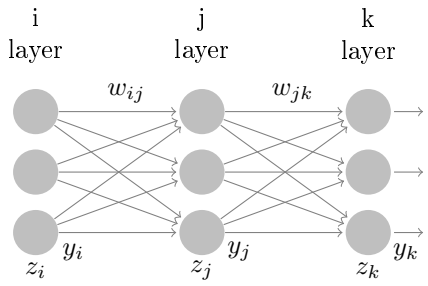


Figure 1: Three layer feed-forward neural network. Each layer labeled by its receptive index variable.

For convenience, we may consider only one training example and ignore the bias term. Forward propagation of the input z_i is done as follows.

$$y_i = g(z_i)$$
$$z_j = \sum_i w_{ij} y_i$$
$$y_j = g(z_j)$$
$$z_k = \sum_j w_{jk} y_j$$
$$y_k = g(z_k)$$

Where $g(z)$ is some differentiable function (e.g. the logistic function).

Now it is your job to do the back propagation. The incoming error derivatives $\frac{\partial E}{\partial y_k}$ are here given and can be used to compute downstream derivatives using the chain-rule. More precisely, give expressions where

each factor is a computable derivative (or already computed as in the case of $\frac{\partial E}{\partial y_k}$). Tips: It is ok to reuse computed factors in subsequent subproblems.

Questions and answers:

(a) $\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k}\frac{dy_k}{dz_k}$

(b) $\frac{\partial E}{\partial z_j} = \sum_k \left(\frac{\partial E}{\partial z_k}\frac{dz_k}{dy_j}\right)\frac{dy_j}{dz_j}$

(c) $\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial w_{jk}}$

(d) $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}}$