DEEP LEARNING (SS2024)
SEMINAR 1

Assignment 1. Consider an artificial neuron

$$y = f\Big(\sum_{i=1}^{n} w_i x_i + b\Big)$$

for inputs $x \in \mathbb{R}^n$ with weights $w \in \mathbb{R}^n$, bias $b \in \mathbb{R}$ and activation function f. Show that the *affine mapping* given by (w, b) can be replaced by a *linear mapping* if we extend the input space by one dimension. Give a geometric interpretation.

Hint: A linear mapping has no constant bias term. Essentially we want to obtain $y = f(\bar{w}^\mathsf{T}\bar{x})$ for a suitable representation of x by $\bar{x}$. This transformation is quite common. It simplifies both derivations and implementations (*e.g.* in regression, logistic regression, Perceptron algorithm). It is used in the first lab.

Assignment 2 (Softmax)**.** The softmax function is defined as

$$\mathrm{softmax}\colon \mathbb{R}^n \to \mathbb{R}^n_+\colon s \mapsto \frac{e^{s_i}}{\sum_j e^{s_j}},$$

where the inputs s are often called *scores* or *logits*. It is a standard way to parameterise a categorical distribution. It is commonly used in neural network classifiers, to define the predictive probability of class labels. Show the following properties of this function:

a) Scores are defined up to an additive constant, i.e. show that $\mathrm{softmax}(s)$ is invariant to adding the same number to all scores s.

b) Deciding for the best class can be performed either based on probabilities or scores, i.e. show that $\arg\max_i \mathrm{softmax}(s)_i = \arg\max_i (s_i)$.

c) Let us verify that a multinomial regression model that uses softmax, is equivalent to a logistic regression model in the case of two classes. In the next lab, we will ask you to implement a NN for a two class problem, and this exercise should help you to avoid confusion.

Consider a classification problem with features $x \in \mathbb{R}^n$ and two classes $y \in \{1, 2\}$. A neural network for this problem may be designed to output probabilities of the two classes as the vector with components

$$p(y{=}k \,|\, x) = \mathrm{softmax}_k(Wx + b), \quad k = 1, 2 \tag{1}$$

using weights $W \in \mathbb{R}^{2\times n}$, $b \in \mathbb{R}^2$. Another network for the same problem is designed to output real-valued scores

$$a = v^\mathsf{T} x + c, \tag{2}$$

using weights $v \in \mathbb{R}^n$, $c \in \mathbb{R}$. It defines the predictive probability as

$$p(y{=}1 \,|\, x) = \sigma(a) \;\text{ and }\; p(y{=}2 \,|\, x) = 1 - \sigma(a) = \sigma(-a), \tag{3}$$

where σ is the logistic sigmoid function: $\sigma(a) = \frac{1}{1+e^{-a}}$. Show that each model of the form (1) has an equivalent model of the form (2)-(3) and vice-versa.

Assignment 3. Consider a two-layer network

$$x^2 = F(x^0) = f \circ A^2 \circ f \circ A^1 x^0$$

with affine mappings $A^k x^{k-1} = W^k x^{k-1} + b^k$, $k = 1, 2$ and element-wise activation function f.

a) Assume that the activation function f is the identity mapping $f\colon x \to x$. Show that the network is equivalent to a network with only one affine layer. It follows that the identity activation function is not a good choice in most cases.

b) Assume that the activation function is ReLU, i.e. $f(x) = \max(0, x)$. Show that rescaling $(W^1, b^1) \to (\lambda W^1, \lambda b^1)$ and $(W^2, b^2) \to (\lambda^{-1} W^2, b^2)$ with some positive λ keeps the network mapping F unchanged.
Note: The equivalence of the forward pass does not imply equivalence w.r.t. training. In fact, we will discuss later that such transforms substantially affect gradient descent algorithms.

Assignment 4 (Overfitting)**.** In this exercise we discuss the overfitting problem on the example of a simple logistic regression model (2-class classification). The model is defined as follows:

$$p(y \,|\, x; w) = \sigma(y w^T x),$$

where $y = \pm 1$ is the class, $x \in \mathbb{R}^n$ is the feature vector, $w \in \mathbb{R}^n$ is a parameter vector and σ is the logistic sigmoid function. Given training data $\mathcal{T} = \{(x_j, y_j) \mid j = 1 \ldots m\}$, we want to estimate w by maximising the (conditional) log-likelihood $\sum_j \log p(y_j \,|\, x_j; w)$.

a) Let us assume that the training data are linearly separable. Prove that in this case the logistic regression problem has no finite optimal solution. This can be shown e.g. by noting that for any w that achieves a correct classification, taking $w' = \alpha w$ with $\alpha > 1$ achieves a higher likelihood.

b) Show that subtracting the regularizer on the weight norm $\lambda \|w\|^2$ with some $\lambda > 0$ fixes this problem, *i.e.* that the optimal solution is finite.