Machine Learning for Physicists: Example Problems

(1) You have a neural network with one input layer (N_i neurons), two hidden layers (N_{h1} neurons connected to the input layer, and N_{h2} neurons above that), and one output layer (N_o neurons). They are all fully connected, with sigmoid nonlinearities ($y = \sigma(z)$ for calculating the output y of any neuron). The weights shall be denoted in the form $W^{o,h2}_{kj}$ and so on, likewise the biases b^{h2}_k. The cost function, evaluated for a single input sample $\vec{y}^i$, is $C = \sum_{k=1}^{N_o}(y^o_k - f_k(\vec{y}^i))^2$, where $\vec{f}(\vec{y}^i)$ is the desired function that we want the network to learn. Calculate $\frac{\partial C}{\partial W^{h2,h1}_{mn}}$ directly. Express the result in terms of matrix- and vector-multiplication with suitable matrices and vectors.

(2) You have two input neurons, directly connected to a single output neuron. Each of the input neurons (called y^i_1 and y^i_2) can only have the two values 0 or to 1. (2a) Suggest how to choose the weights and biases such that the network calculates y^i_1 AND (NOT y^i_2). You can assume that the output neuron has a sharp step function as nonlinearity, i.e. $y^o = \theta(z^o)$, where $\theta(z) = 0$ for $z < 0$ and $\theta(z) = 1$ for $z \geq 0$. (2b) Assume that the possible input pairs 00, 01 etc. occur with probabilities p_{00}, p_{01} etc. Now write down explicitly the cost function $\bar{C}$ that is averaged over the training data, for the case where the weights and bias are still not fixed (call them W_1, W_2 and b). (2c) Assuming $p_{00} = p_{01} = \ldots = 1/4$, as well as $b = -1$, plot $\bar{C}$ as a function of W_1 and W_2 [hint: indicate the different regions in the W_1, W_2 plane that have different values of $\bar{C}$ and indicate those values].

(3) You find the following lines in a python/keras program:

```
model=Sequential()
model.add(Dense(3,activation='relu',input_shape=(2,)))
model.add(Dense(1,activation='sigmoid'))
```

Draw a picture of the network layout. Write down explicitly the function which the output neuron of this network calculates (introduce suitable notation for the weights and biases, and define explicitly the nonlinear functions that occur).

(4) [+ bonus points for particularly efficient and practical solutions] Describe the kind of neural network and outline the training procedure with which you would try to solve the following challenge: A little robot can move around on the floor of a large room. It is being fed the picture from a camera. For each second, it can decide to move in either one of 8 directions. Its task is to avoid obstacles (bumping into something is bad, and each collision is registered by an automatic detector) but also to pick up all kind of little waste stuff that is lying on the floor (it has an arm that can grab the waste in front of it and will put it automatically into some basket). After training, this robot should be able to perform this task in arbitrary rooms (not just with some precisely defined geometry and precisely placed obstacles).