Prof. Dr. Leif Döring
André Ferdinand, Sara Klein

Reinforcement Learning

12. Solution Sheet

1. Back propagation: Theory

Go through the proof of Theorem 5.3.8 and check that the gradient $\nabla_w F_w(x)$ is computed by $\frac{\partial}{\partial w^l_{i,j}} F_w(x)_m = \delta^l_{i,j} V^l_i$ with $\delta^L_{j,m} = \Phi'(h^L_j)$ if $j = m$ and zero else and $\delta^l_{j,m} = \sigma'(h^{l-1}_j) w^{l+1}_{j,m} \delta^{l+1}_{m,m}$.

Hint: Note that $\nabla_w F_w(x) \in R^k$, i.e. $\delta_{j,m}$ is with respect to the m-th koordinate of $\nabla_w F_w(x)$.

Solution:

First, note that $\nabla_w F_w(x) \in \mathbb{R}^k$, i.e. we have to consider the derivatives in each dimension.

***Output layer**: To compute the partial derivative with respect to the final weights $w^L_{i,j}$ we proceed as follows:*

$$\begin{aligned}\frac{\partial}{\partial w^L_{i,j}} F_w(x)_j &= \frac{\partial}{\partial w^L_{i,j}} \Phi\Big(\underbrace{\sum_{k=1}^{N_{L-1}} w^L_{k,j} V^{L-1}_k}_{=h^L_j}\Big)\\ &= \Phi'(h^L_j) V^{L-1}_i\\ &=: \delta^L_{j,j}\, V^{L-1}_i.\end{aligned}$$

Note that the derivative of $F_w(x)_m$ with respect to $w^L_{i,j}$ is 0 for $j \neq m$.

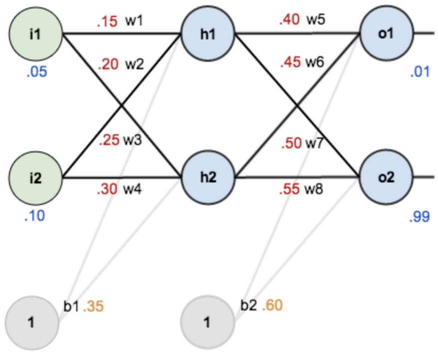
In the last hidden layer we have

$$\begin{aligned}&\frac{\partial}{\partial w^{L-1}_{i,j}} F_w(x)_m\\ &= \frac{\partial}{\partial w^{L-1}_{i,j}} \Phi\Big(\sum_{t=1}^{N_{L-1}} w^L_{t,m} \sigma\Big(\sum_{r=1}^{N_{l-2}} w^{L-1}_{r,t} V^{L-2}_r\Big)\Big)\\ &= \Phi'(h^L_m) \frac{\partial}{\partial w^{L-1}_{i,j}} \sum_{t=1}^{N_{L-1}} w^L_{t,m} \sigma\Big(\sum_{r=1}^{N_{l-2}} w^{L-1}_{r,t} V^{L-2}_r\Big)\\ &= \Phi'(h^L_m) \sum_{t=1}^{N_{L-1}} w^L_{t,m} \sigma'\Big(\sum_{r=1}^{N_{l-2}} w^{L-1}_{r,t} V^{L-2}_r\Big) \frac{\partial}{\partial w^{L-1}_{i,j}} \sum_{r=1}^{N_{l-2}} w^{L-1}_{r,t} V^{L-2}_r\\ &= \Phi'(h^L_m) w^L_{j,m} \sigma'(h^{L-2}_j) V^{L-2}_i\\ &= \delta^L_{m,m} w^L_{j,m} \sigma'(h^{L-2}_j) V^{L-2}_i\\ &= \delta^{L-1}_{j,m} V^{L-2}_i.\end{aligned}$$

And so on...

2. **Back propagation: Praxis**

Consider a neural network with two inputs, two hidden neurons, two output neurons:



The blue numbers are the training inputs/outputs, the red numbers are the weights and the orange numbers the biases. Calculate the forward passes and backward passes from Algorithm 33 in the lecture and update the weights.

Solution:

See https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example

$net_{h_l} \equiv \delta^l$ *and* $out_{h_l} \equiv V^l$ *from Algorithm 33 in the lecture notes.*