

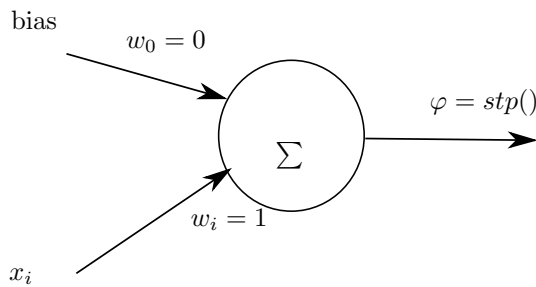
ML II: Exercise 2

Tobias Graf, Philipp Rentzsch, Lina Gundelwein

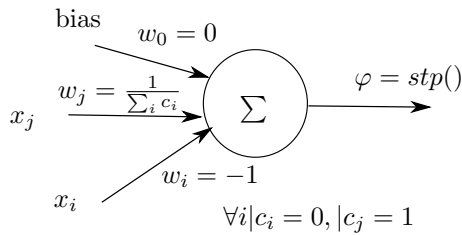
May 20, 2016

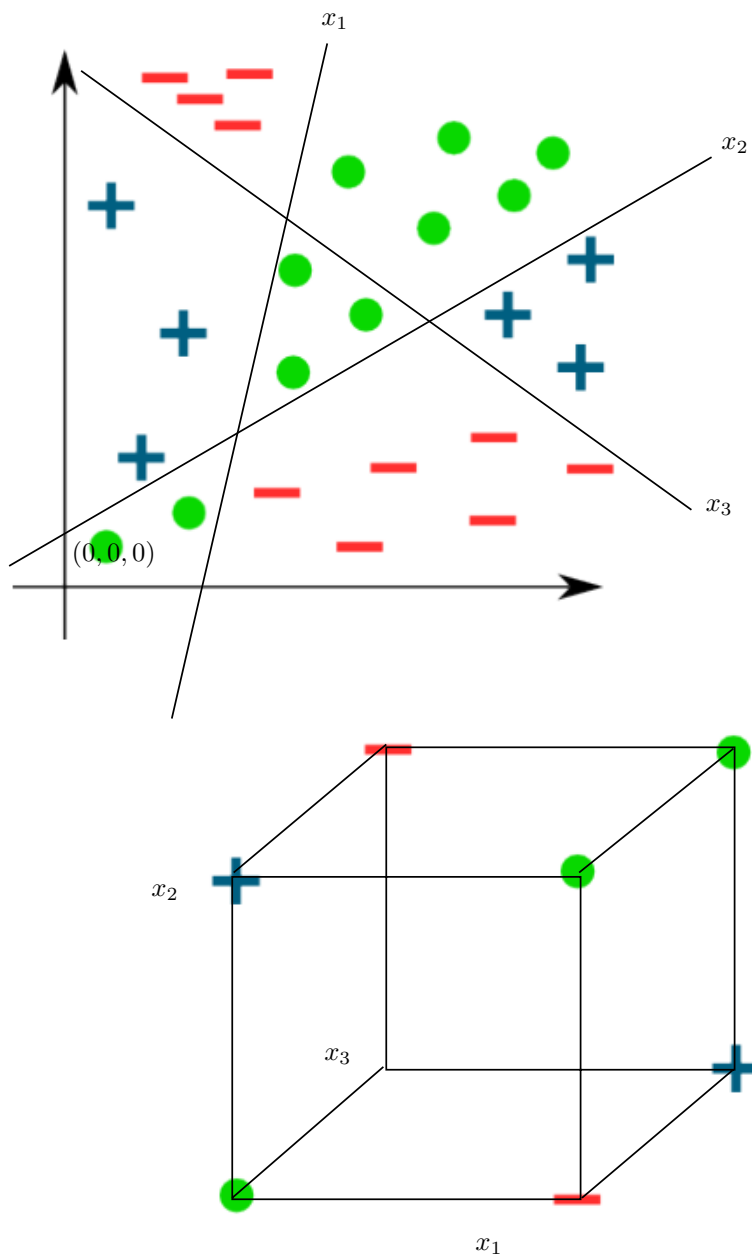
2.1 Simple Networks

1. Logical OR on binary input vector: The weight from the input bias is 0, for all x_i is $w_i = 0$. The binary step function is used as activation.



2. Arbitrary vector: The weights depend the elements of c . w_i is $\frac{1}{\sum_i c_i}$ for all i where c_i is 1 and -1 for all other w_i . The activation is again a binary step resulting 1 for $\tilde{z} = 1$ and 0 otherwise.

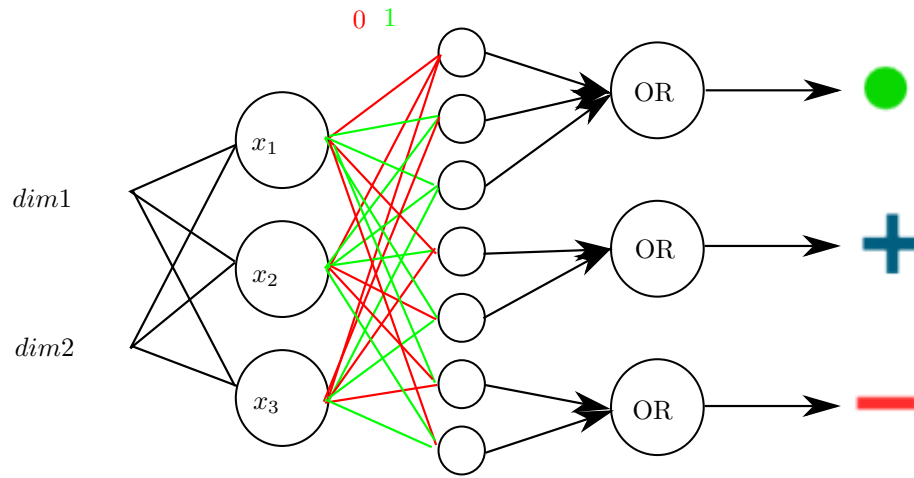




Generalization to higher dimensions: Draw hyperdimensional decision boundaries each dividing the input space into two subspaces. The number of boundaries relates to the dimension of the hypercube.

2.2 3-layer Universal Classifier

The first layer are the discriminative hyperplanes from 2.1.3 that divide the input space into two subgroups. Layer two employs the Arbitrary vector neurons that each decide for one corner of the hypercube whether the data point belongs to the group. The third layer then combines the results from the corners with the same class using the OR neurons from 2.1.1.



Problems we see with these zero loss training classifiers are that they may be over-fitted to the training data.

3.1 Linear Activation Function

Using any linear activation function $\varphi(x) = x$ leads to the following outputs:

$$\begin{aligned}z_1 &= \varphi(B_1 z_0) = B_1 z_0 \\z_2 &= B_2 z_1 = B_2 B_1 z_0\end{aligned}$$

This could be replaced by a single layer with the parameters $B = B_1 B_2$.

3.2 Weight Decay

1.

$$\begin{aligned}Loss(w) &= Loss_0(w) + \frac{\lambda}{2N} w^T w \\ \frac{\partial Loss}{\partial w} &= \frac{\partial}{\partial w} Loss_0 + \frac{\lambda}{N} w \\ w &= w - \tau \frac{\partial}{\partial w} Loss_0 - \tau \frac{\lambda}{N} w \\ &= (1 - \frac{\tau \lambda}{N}) w - \tau \frac{\partial}{\partial w} Loss_0 \\ &\rightarrow \epsilon = \frac{\tau \lambda}{N}\end{aligned}$$

2. The weight decays in proportion to its size. Thus, larger weights are penalized and weights with a small magnitude are preferred which avoids overfitting.

3.

$$\begin{aligned}Loss(w) &= Loss_0(w) + \frac{\lambda}{2N} |w| \\ \frac{\partial Loss}{\partial w} &= \frac{\partial}{\partial w} Loss_0 + \frac{\lambda}{N} \text{sgn}(w) \\ w &= w - \tau \frac{\partial}{\partial w} Loss_0 - \tau \frac{\lambda}{N} \text{sgn}(w)\end{aligned}$$

4. Since the biases are fixed and representing the offset, not the curvature of the model, the regularization has little effect on them.

4. Application

Honestly, very complicated object structure.