Solutions for Artifical Neural Networks

September 21, 2020

Department of Physics
University of Gothenburg
Göteborg, Sweden 2020

Chapter 2

2.3 Cross-talk term

(a) Apply $\boldsymbol{x}^{(\nu)}$ to the network: $S_i = \operatorname{sgn}\big(\sum_{j=1}^{N} w_{ij} x_j^{(\nu)}\big)$. For $\boldsymbol{x}^{(\nu)}$ to remain unchanged under this update we must require that $S_i = x_i^{(\nu)}$. In other words

$$x_i^{(\nu)} = \operatorname{sgn}\Big(\sum_{j=1}^{N} w_{ij} x_j^{(\nu)}\Big). \tag{1}$$

This condition evaluates to:

$$\begin{aligned}
x_i^{(\nu)} &= \operatorname{sgn}\Big(\sum_{j=1}^{N} w_{ij} x_j^{(\nu)}\Big) = \operatorname{sgn}\Big(\sum_{\substack{j=1\\ j\neq i}}^{p}\Big(\frac{1}{N}\sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)}\Big) x_j^{(\nu)}\Big) \\
&= \operatorname{sgn}\Big(\frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N} x_i^{(\nu)} \underbrace{x_j^{(\nu)} x_j^{(\nu)}}_{=1} + \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}\Big) \\
&= \operatorname{sgn}\Big(\frac{N-1}{N} x_i^{(\nu)} + \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}\Big) \\
&= \operatorname{sgn}\Big(x_i^{(\nu)} - \frac{1}{N} x_i^{(\nu)} + \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}\Big) \\
&= \operatorname{sgn}\Big(x_i^{(\nu)} + \text{corrections}\Big).
\end{aligned}$$

The corrections are given by:

$$\text{corrections} = -\frac{1}{N} x_i^{(\nu)} + \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}.$$

So condition (1) is satisfied if $\big|\text{corrections}\big| < 1$. Now define $C_i^{(\nu)}$ as follows:

$$C_i^{(\nu)} \equiv \frac{1}{N} - \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} x_i^{(\nu)} \quad \big(= \text{corrections} \times (-x_i^{(\nu)})\big).$$

Multiply both sides of Equation (1) by $-x_i^{(\nu)}$ and rewrite this condition as

$$-1 = \operatorname{sgn}\big(-1 + C_i^{(\nu)}\big).$$

This condition is satisfied provided that $C_i^{(\nu)} < 1$. No limits have been taken so far.
(b) Assume random patterns $\boldsymbol{x}^{(\nu)}$ with bits

$$x_i^{(\mu)} = \begin{cases} +1 \text{ with probability } \frac{1}{2}, \\ -1 \text{ with probability } \frac{1}{2}. \end{cases}$$

The bit $x_i^{(\nu)}$ is unchanged after a single asynchronous update if $C_i^{(\nu)} < 1$, as derived in task (a). Thus, the probability that the bit $x_i^{(\nu)}$ is changed (although it shouldn't be) is given by

$$P_{\text{error}^{t=1}} = \text{prob}(C_i^{(\nu)} > 1).$$

To evaluate $P_{\text{error}}^{t=1}$, consider $C_i^{(\nu)}$ in the limit of $N \gg 1$:

$$C_i^{(\nu)} \approx -\frac{1}{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{\substack{\mu=1 \\ \mu\neq\nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} x_i^{(\nu)} = -\frac{1}{N} \sum_{k=1}^{(N-1)(p-1)} z_k,$$

where z_k are independent random variables with

$$z_k = \begin{cases} +1 \text{ with probability } \frac{1}{2}, \\ -1 \text{ with probability } \frac{1}{2}. \end{cases}$$

Since the variables z_k are indenpendently identically distributed (with zero mean and unit variance), we can use the central limit theorem. It follows that the distribution of $C_i^{(\nu)}$ has the following properties:

- $C_i^{(\nu)}$ is approximately Gaussian distributed in the limit $N \gg 1$.
- The mean of $C_i^{(\nu)}$ is equal to zero (since the mean of the x_k-variables vanishes).
- The variance σ^2 of $C_i^{(\nu)}$ is $\sigma^2 = \frac{1}{N^2}(N-1)(p-1) \approx \frac{p}{N}$ in the limit $N \gg 1$ and $p \gg 1$.

Thus it follows that

$$P_{\text{error}}^{t=1} = \text{prob}(C_i^{(\nu)} > 1) = \int_1^\infty \mathrm{d}x \, \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} = \frac{1}{2}\left[1 - \text{erf}\left(\frac{1}{\sqrt{2\sigma^2}}\right)\right],$$

where the error function $\text{erf}(z)$ is defined as

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \mathrm{d}y \, \mathrm{e}^{-\frac{y^2}{2}}.$$

We conclude that

$$P_{\text{error}}^{t=1} = \frac{1}{2}\left[1 - \text{erf}\left(\frac{1}{\sqrt{2\alpha}}\right)\right].$$

Here $\alpha = p/N$ is the storage capacity.

2.11 Hopfield net with four neurons

One stored pattern in the network: $\boldsymbol{x}^{(1)} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$. Weight matrix:

$$\mathbb{W} = \frac{1}{N}\boldsymbol{x}^{(1)}\boldsymbol{x}^{(1)\mathsf{T}} = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

Feeding the 2^4 patterns gives:

1. Take $\boldsymbol{S}(t=0) = \boldsymbol{x}^{(1)} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$.
   To compute $\boldsymbol{S}(t=1)$ must first evaluate $\mathbb{W}\boldsymbol{S}(t=0) = \frac{1}{4}\boldsymbol{x}^{(1)}\boldsymbol{x}^{(1)\mathsf{T}}\boldsymbol{x}^{(1)}$. Then take sgn function componentwise. Obtain $\boldsymbol{S}(t=1) = \boldsymbol{x}^{(1)}$.

2. $\boldsymbol{S}(t=0) = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$, $\mathbb{W}\boldsymbol{S}(t=0) = \frac{1}{4}\begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$.
   Take the sgn function componentwise to obtain $\boldsymbol{S}(t=1) = \boldsymbol{x}^{(1)}$. The other cases are analogous. The results are given below.

3. $\boldsymbol{S}(t=0) = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$, $\boldsymbol{S}(t=1) = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \boldsymbol{x}^{(1)}$

4. $\boldsymbol{S}(t=0) = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$, $\boldsymbol{S}(t=1) = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \boldsymbol{x}^{(1)}$

5. $\boldsymbol{S}(t=0) = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$, $\boldsymbol{S}(t=1) = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \boldsymbol{x}^{(1)}$

6. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\1\\-1\\-1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

7. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\-1\\-1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

8. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\-1\\1\\-1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

9. $\boldsymbol{S}(t=0)=\begin{bmatrix}1\\1\\1\\-1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

10. $\boldsymbol{S}(t=0)=\begin{bmatrix}1\\1\\-1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

11. $\boldsymbol{S}(t=0)=\begin{bmatrix}1\\-1\\1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$.

12. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}-1\\1\\1\\1\end{bmatrix}=-\boldsymbol{x}^{(1)}$

13. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\1\\-1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}-1\\1\\1\\1\end{bmatrix}=-\boldsymbol{x}^{(1)}$

14. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\-1\\1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}-1\\1\\1\\1\end{bmatrix}=-\boldsymbol{x}^{(1)}$

15. $\boldsymbol{S}(t=0)=\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}-1\\1\\1\\1\end{bmatrix}=-\boldsymbol{x}^{(1)}$

16. $\boldsymbol{S}(t=0)=\begin{bmatrix}-1\\1\\1\\1\end{bmatrix}$, $\boldsymbol{S}(t=1)=\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$.

In summary:

1. When we feed one of the first five patterns, the stored pattern is retrieved. Pattern no. 1 is the stored patterns, and patterns 2 to 5 differ in only one bit compared with the stored pattern.

2. When we feed a pattern that has more than two bit differences, then the network retrieves the inverted version of the stored pattern (patterns 12-16).

3. When exactly two bits are distorted, the network fails (patterns 6 to 11).

2.12 Recognising letters with a Hopfield net

(a) Define

$$Q^{(\mu,\nu)} = \sum_{j=1}^{N} x_j^{(\mu)} x_j^{(\nu)}. \tag{2}$$

The bit j contributes with $+1$ to $Q^{(\mu,\nu)}$ if $x_j^{(\mu)} = x_j^{(\nu)}$, and with -1 if $x_j^{(\mu)} \neq x_j^{(\nu)}$. Since the number of bits are $N = 32$, we have $Q^{(\mu,\nu)} = 32 - 2H^{(\mu,\nu)}$, where $H^{(\mu,\nu)}$ is the number of bits that are different in patterns μ and ν (the Hamming distance). By looking at the figures, we find the following:

- $H^{(1,1)} = 0 \Rightarrow Q^{(1,1)} = 32$
- $H^{(1,2)} = 13 \Rightarrow Q^{(1,2)} = 6$
- $H^{(1,3)} = 2 \Rightarrow Q^{(1,3)} = 28$
- $H^{(1,4)} = 32 \Rightarrow Q^{(1,4)} = -32$
- $H^{(1,5)} = 16 \Rightarrow Q^{(1,5)} = 0$
- $H^{(2,1)} = H^{(1,2)} = 13 \Rightarrow Q^{(2,1)} = 6$
- $H^{(2,2)} = 0 \Rightarrow Q^{(2,2)} = 32$
- $H^{(2,3)} = 15 \Rightarrow Q^{(2,3)} = 2$
- $H^{(2,4)} = 32 - H^{(2,1)} = 19 \Rightarrow Q^{(2,4)} = -6$
- $H^{(2,5)} = 19 \Rightarrow Q^{(2,5)} = -6$

(b) We have that

$$\begin{aligned} b_i^{(\nu)} &= \sum_j w_{ij} x_j^{(\nu)} = \sum_j \frac{1}{32}\left(x_i^{(1)} x_j^{(1)} + x_i^{(2)} x_j^{(2)}\right) x_j^{(\nu)} \\ &= \frac{1}{32} x_i^{(1)} \sum_j x_j^{(1)} x_j^{(\nu)} + \frac{1}{32} x_i^{(2)} \sum_j x_j^{(2)} x_j^{(\nu)} = \frac{1}{32} x_i^{(1)} Q^{(1,\nu)} + \frac{1}{32} x_i^{(2)} Q^{(2,\nu)}. \end{aligned}$$

From (a) we have that:

$$
\begin{aligned}
b_i^{(1)} &= \frac{Q^{(1,1)}}{32} x_i^{(1)} + \frac{Q^{(2,1)}}{32} t_i^{(2)} = x_i^{(1)} + \frac{6}{32} x_i^{(2)}, \\
b_i^{(2)} &= \frac{Q^{(1,2)}}{32} x_i^{(1)} + \frac{Q^{(2,2)}}{32} x_i^{(2)} = \frac{6}{32} x_i^{(1)} + x_i^{(2)}, \\
b_i^{(3)} &= \frac{Q^{(1,3)}}{32} x_i^{(1)} + \frac{Q^{(2,3)}}{32} x_i^{(2)} = \frac{28}{32} x_i^{(1)} + \frac{2}{32} x_i^{(2)}, \\
b_i^{(4)} &= \frac{Q^{(1,4)}}{32} x_i^{(1)} + \frac{Q^{(2,4)}}{32} x_i^{(2)} = -x_i^{(1)} - \frac{6}{32} x_i^{(2)}, \\
b_i^{(5)} &= \frac{Q^{(1,5)}}{32} x_i^{(1)} + \frac{Q^{(2,5)}}{32} x_i^{(2)} = -\frac{6}{32} x_i^{(2)}.
\end{aligned}
$$

(c) From (b), we find that:

$$
\begin{aligned}
x_i^{(1)} &\to \operatorname{sgn}(b_i^{(1)}) = x_i^{(1)}, \\
x_i^{(2)} &\to \operatorname{sgn}(b_i^{(2)}) = x_i^{(2)}, \\
x_i^{(3)} &\to \operatorname{sgn}(b_i^{(3)}) = x_i^{(1)}, \\
x_i^{(4)} &\to \operatorname{sgn}(b_i^{(4)}) = -x_i^{(1)} = x_i^{(4)}, \\
x_i^{(5)} &\to \operatorname{sgn}(b_i^{(5)}) = -x_i^{(2)}.
\end{aligned}
$$

Thus, patterns $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(4)}$ are stable.

2.13 Diluted Hopfield net

In this exercise only a small fraction of connections is active, the others are not active. Even so, the network is able to learn, as will be shown here. $S_i = \mathrm{sgn}(b_i)$, $b_i = \sum_{j=1}^{N} w_{ij} S_j$, $w_{ij} = \frac{K_{ij}}{K} \sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)}$,

$$K_{ij} = \begin{cases} 1 \text{ with probability } \frac{K}{N}, \\ 0 \text{ with probability } 1 - \frac{K}{N}. \end{cases}$$

Limit considered: $N \gg 1$, $p \gg 1$, $K \gg 1$, $N \gg p$, $N \gg K$. a) $\boldsymbol{S} = \boldsymbol{x}^{(\nu)}$,

$$b_i = \sum_{j=1}^{N} w_{ij} S_j = \sum_{j=1}^{N} w_{ij} x_j^{(\nu)} = \sum_{j=1}^{N} \frac{K_{ij}}{K} \sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} =$$

$$= \sum_{j=1}^{N} \frac{K_{ij}}{K} x_i^{(\nu)} \underbrace{x_j^{(\nu)} x_j^{(\nu)}}_{=1} + \frac{K_{ii}}{K} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{p} \underbrace{x_i^{(\mu)} x_i^{(\mu)}}_{=1} x_i^{(\nu)} + \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{K_{ij}}{K} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} =$$

$$= \sum_{j=1}^{N} \frac{K_{ij}}{K} x_i^{(\nu)} + \frac{K_{ii}}{K} \frac{p-1}{N} x_i^{(\nu)} + \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{K_{ij}}{K} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)},$$

which gives

$$\langle b_i \rangle_{\mathrm{c}} = \sum_{j=1}^{N} \Big[1 \frac{K}{N} + 0 \Big(1 - \frac{K}{N} \Big) \Big] \frac{x_i^{(\nu)}}{K} + \Big[1 \frac{K}{N} + 0 (1 - \frac{K}{N}) \Big] \frac{p-1}{N} \frac{x_i^{(\nu)}}{K} + \underbrace{\frac{1}{K} \sum_{\substack{j=1 \\ j \neq i}}^{N} \Big[1 \frac{K}{N} + 0 \Big(1 - \frac{K}{N} \Big) \Big] \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}}_{= \langle \text{cross-talk term} \rangle} =$$

$$\approx x_i^{(\nu)} + \underbrace{\frac{1}{K} \sum_{\substack{j=1 \\ j \neq i}}^{N} \Big(\frac{K}{N} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} \Big)}_{= \langle \text{cross-talk term} \rangle}$$

In the last line, the limit $N \gg 1$, $N \gg p$ was used: in this limit, it follows that the contribution of the second term in the penultimate line is negligible, i.e. $\frac{1}{N} \frac{p-1}{N} x_i^{(\nu)} \approx 0$

(b) $\langle C_i^{(\nu)} \rangle_{\mathrm{c}} = \langle \text{cross-talk term} \rangle =$ does not depend on N because it is the sum of $(N-1)(p-1)\frac{K}{N} \approx pK$ random numbers that are ± 1 with equal probability, divided by K (the remaining terms are zero). Thus $\langle C_i^{(\nu)} \rangle_{\mathrm{c}}$ is the sum of $\approx Kp$ random numbers that are ± 1 with equal probability, divided by K. Note that here we used that in the limit $N \gg 1$, $p \gg 1$, it follows that $N - 1 \approx N$, and $p - 1 \approx p$. In the limit of $K \gg 1$ and

$p \gg 1$, we use the CLT (Central Limit Theorem) to conclude that $\langle C_i^{(\nu)} \rangle_{\rm c}$ is Gaussian distributed with mean 0. The variance is computed as follows. Since each term is ± 1 divided by K, then the variance of each term is $1/K^2$. There are approximately Kp terms in the sum, and so the variance of $\langle C_i^{(\nu)} \rangle_{\rm c}$ is approximately equal to $= \frac{Kp}{K^2} = \frac{p}{K}$. Note that the variance depends on K because, as explained above, the number of terms that enter the sum depends on K.

(c) $m_\nu = \frac{1}{N}\sum_{i=1}^N x_i^{(\nu)} \langle \langle S_i \rangle_{\rm c} \rangle_{\rm time}$. In the limit of $N \gg 1$, we use the approximation $\langle \langle S_i \rangle_{\rm c} \rangle_{\rm time} \approx \langle S_i \rangle_{\rm c} \approx {\rm sgn}(\langle b_i \rangle_{\rm c})$ to obtain that

$$m_\nu = \frac{1}{N}\sum_{i=1}^N x_i^{(\nu)} {\rm sgn}(\langle b_i \rangle_{\rm c}) = \frac{1}{N}\sum_{i=1}^N {\rm sgn}\Big(x_i^{(\nu)} \langle \sum_{j=1}^N w_{ij} S_j \rangle_{\rm c} \Big) =$$

$$= \frac{1}{N}\sum_{i=1}^N {\rm sgn}\Big(x_i^{(\nu)} \langle \sum_{j=1}^N \frac{K_{ij}}{K} \sum_{\mu=1}^p x_i^{(\mu)} x_j^{(\mu)} S_j \rangle_{\rm c} \Big) \Longrightarrow$$

$$m_\nu = \frac{1}{N}\sum_{i=1}^N {\rm sgn}\Big[\langle \sum_{j=1}^N \frac{K_{ij}}{K} x_i^{(\nu)} x_j^{(\nu)} x_i^{(\nu)} S_j \rangle + x_i^{(\nu)} \underbrace{\langle \sum_{j=1}^N \frac{K_{ij}}{K} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p x_i^{(\mu)} x_j^{(\mu)} S_j \rangle}_{= \langle C_i^{(\nu)} \rangle_{\rm c}} \Big] =$$

$$= \frac{1}{N}\sum_{i=1}^N {\rm sgn}\Big[\Big\langle \sum_{j=1}^N \frac{K_{ij}}{K} x_j^{(\nu)} S_j \Big\rangle + x_i^{(\nu)} \Big\langle C_i^{(\nu)} \Big\rangle_{\rm c} \Big].$$

We have that $\Big\langle \sum_{j=1}^N \frac{K_{ij}}{K} x_j^{(\nu)} S_j \Big\rangle \approx m_\nu$ in the limit of $K \gg 1$. It follows that

$$m_\nu \approx \frac{1}{N}\sum_{i=1}^N {\rm sgn}\Big[m_\nu + x_i^{(\nu)} \Big\langle C_i^{(\nu)} \Big\rangle_{\rm c} \Big],$$

where $\Big\langle C_i^{(\nu)} \Big\rangle_{\rm c}$ is Gaussian distributed with mean 0 and variance approximately equal to $\frac{p}{K}$

$$\Longrightarrow m_\nu \approx \int d\Big\langle C_i^{(\nu)} \Big\rangle_{\rm c} p(\Big\langle C_i^{(\nu)} \Big\rangle_{\rm c}) {\rm sgn}(m_\nu + x_i^{(\nu)} \Big\langle C_i^{(\nu)} \Big\rangle_{\rm c}),$$

where $p(C) = \frac{1}{\sqrt{2\pi \frac{p}{K}}} {\bf e}^{-\frac{C^2}{2\frac{p}{K}}}$. Now, we need to split the integral to account for the fact that $x_i^{(\nu)} = \pm 1$ with equal probability:

$$m_\nu \approx \frac{1}{2} \int d\Big\langle C_i^{(\nu)} \Big\rangle_{\rm c} p(\Big\langle C_i^{(\nu)} \Big\rangle_{\rm c}) {\rm sgn}(m_\nu + \Big\langle C_i^{(\nu)} \Big\rangle_{\rm c})$$

$$+\frac{1}{2}\int d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\mathrm{sgn}(m_\nu-\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})=\Big[\text{Assuming } m_\nu\geq 0\Big]=$$

$$\frac{1}{2}\Big[-\int_{-\infty}^{-m_\nu} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})+\int_{-m_\nu}^{\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\Big]+$$

$$\frac{1}{2}\Big[-\int_{m_\nu}^{-\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})+\int_{-\infty}^{m_\nu} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\Big]=$$

$$=\frac{1}{2}\Big[-\Big(1-\int_{-m_\nu}^{\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\Big)+\int_{-m_\nu}^{\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\Big]+$$

$$\frac{1}{2}\Big[-\int_{m_\nu}^{\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})+1-\int_{m_\nu}^{\infty} d\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}} p(\left\langle C_i^{(\nu)}\right\rangle_{\mathrm{c}})\Big]=$$

$$=\frac{1}{2}\Big(1-\mathrm{erf}\Big(\frac{-m_\nu}{\sqrt{2\frac{p}{K}}}\Big)\Big)-\frac{1}{2}\Big(1-\mathrm{erf}\Big(\frac{m_\nu}{\sqrt{2\frac{p}{K}}}\Big)\Big)=\frac{1}{2}\Big[\mathrm{erf}\Big(\frac{m_\nu}{\sqrt{2\frac{p}{K}}}\Big)-\underbrace{\mathrm{erf}\Big(\frac{-m_\nu}{\sqrt{2\frac{p}{K}}}\Big)}_{=-\mathrm{erf}\left(\frac{m_\nu}{\sqrt{2\frac{p}{K}}}\right)}\Big].$$

It follows that $m_\nu \approx \mathrm{erf}\big(m_\nu/\sqrt{2\frac{p}{K}}\big)$. The same conclusion holds if we instead assumed that $m_\nu<0$ where we assumed that $m_\nu\geq 0$.

Table 1: Signs of $s_\mu = x_j^{(\mu)} x_j^{(\text{mix})}$. See also Table 2.1 in Lecture notes.

$x_i^{(1)}$	$x_i^{(2)}$	$x_i^{(3)}$	$x_i^{(\text{mix})}$	s_1	s_2	s_3
-1	-1	-1	-1	1	1	1
-1	-1	1	-1	1	1	-1
-1	1	-1	-1	1	-1	1
-1	1	1	1	-1	1	1
1	-1	-1	-1	-1	1	1
1	-1	1	1	1	-1	1
1	1	-1	1	1	1	-1
1	1	1	1	1	1	1

This result is essentially the same as for the fully connected Hopfield network obtained under the mean-field approximation, but here, in the argument of the function we have p/K instead of the storage capacity p/N.

2.14 Mixed states

(a)

$$\sum_{j=1}^{N} w_{ij} x_j^{(\text{mix})} = \sum_{j=1}^{N} \frac{1}{N} \sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\text{mix})} = \sum_{\mu=1}^{p} x_i^{(\mu)} \frac{1}{N} \sum_{j=1}^{N} x_j^{(\mu)} x_j^{(\text{mix})} = \sum_{\mu=1}^{p} x_i^{(\mu)} \langle s_\mu \rangle.$$

(b) We find that $\langle s_1 \rangle$, $\langle s_2 \rangle$ and $\langle s_3 \rangle$ are the average of independent random variables X such that $p(X=-1) = \frac{1}{4}$ and $p(X=1) = \frac{3}{4}$. Thus, the expected value of X is $\mathbb{E}[X] = -\frac{1}{4} + \frac{3}{4} = \frac{1}{2}$ and the variance of X is $\text{Var}[X] = \mathbb{E}[(X-\frac{1}{2})^2] = \frac{1}{4}(-\frac{3}{2})^2 + \frac{3}{4}(\frac{1}{2})^2 = \frac{12}{16} = \frac{3}{4}$. Thus, $\mathbb{E}[\langle s_\mu \rangle] = \frac{1}{2}$ for $\mu = 1,2,3$ and $\text{Var}[\langle s_\mu \rangle] = \frac{1}{N^2} N \frac{3}{4} = \frac{3}{4N} \to 0$ as $N \to \infty$. Thus, $\langle s_\mu \rangle = \frac{1}{2}$ for $\mu = 1,2,3$.

For $\mu > 3$, we find that $\langle s_\mu \rangle$ is $\frac{1}{N}$ times the sum of N independent random variables X with $p(X=-1) = \frac{1}{2}$ and $p(X=1) = \frac{1}{2}$. Thus, $\mathbb{E}[X] = 0$ and $\text{Var}[X] = 1$ and we conclude that, for $\mu > 3$, $\mathbb{E}[\langle s_\mu \rangle] = 0$ and $\text{Var}[\langle s_\mu \rangle] = \frac{1}{N^2} N = \frac{1}{N} \to 0$ as $N \to \infty$ and thus $\langle s_\mu \rangle = 0$ for $\mu > 3$.

(c)

$$\sum_{j=1}^{N} w_{ij} x_j^{(\text{mix})} = \sum_{\mu=1}^{3} x_i^{(\mu)} \underbrace{\langle s_\mu \rangle}_{=\frac{1}{2}} + \sum_{\mu>3} x_i^{(\mu)} \underbrace{\langle s_\mu \rangle}_{=0} = \sum_{\mu=1}^{3} \frac{1}{2} x_i^{(\mu)} = \frac{1}{2}(x_i^{(1)} + x_i^{(2)} + x_i^{(3)}).$$

The cross-talk term can be neglected since $\langle s_\mu \rangle = 0$ for $\mu > 3$ (from the previous part of this exercise).

(d) We apply the update rule and use (c) to conclude that

$$\mathrm{sgn}\Big(\sum_{j=1}^{N} w_{ij}\, x_j^{(\mathrm{mix})}\Big) = \mathrm{sgn}\Big(\frac{1}{2}(x_i^{(1)} + x_i^{(2)} + x_i^{(3)})\Big) = x_i^{(\mathrm{mix})}.$$

Chapter 3

Chapter 4

4.10 McCulloch-Pitts dynamics for restricted Boltzmann machine

The deterministic analogue is

$$v_j' = \mathrm{sgn}\big[b_j^{(v)}\big] \quad \text{with} \quad b_j^{(v)} = \sum_{i=1}^{M} h_i\, w_{ij}, \tag{3}$$

for the visible neurons, and

$$h_i' = \mathrm{sgn}\big[b_j^{(v)}\big] \quad \text{with} \quad b_i^{(h)} = \sum_{j=1}^{N} w_{ij}\, v_j. \tag{4}$$

for the hidden neurons.

Say that the hidden neurons $h_{m1}, h_{m2}, \ldots$ change sign when they update. We have

$$h_i' = h_i - 2h_i\delta_{i,m1} - 2h_i\delta_{i,m2} - \ldots \tag{5}$$

and

$$H' = -\sum_{ij} w_{ij} h_i' v_j = H + 2\sum_{ij} w_{ij} h_i \delta_{i,m1} v_j + 2\sum_{ij} w_{ij} h_i \delta_{i,m2} v_j + \ldots \tag{6}$$

$$= H + 2\sum_{j} w_{m1,j} h_{m1} v_j + 2\sum_{j} w_{m2,j} h_{m2} v_j + \ldots \tag{7}$$

$$= H + 2h_{m1} b_{m1}^{(h)} + 2h_{m2} b_{m2}^{(h)} + \ldots \tag{8}$$

$$= H - 2\mathrm{sgn}\big[b_{m1}^{(h)}\big] b_{m1}^{(h)} - 2\mathrm{sgn}\big[b_{m2}^{(h)}\big] b_{m2}^{(h)} - \ldots \tag{9}$$

Say that the visible neurons $v_{m1}, v_{m2}, \ldots$ change sign when they update. We have

$$v_j' = v_j - 2v_j\delta_{j,m2} - 2v_j\delta_{j,m2} - \ldots \tag{10}$$

and

$$H' = -\sum_{ij} w_{ij} h_i v'_j = H + 2\sum_{ij} w_{ij} h_i \delta_{j,m1} v_j + 2\sum_{ij} w_{ij} h_i \delta_{j,m2} v_j + \dots \tag{11}$$

$$= H + 2\sum_{i} w_{i,m1} h_i v_{m1} + 2\sum_{i} w_{i,m2} h_i v_{m2} + \dots \tag{12}$$

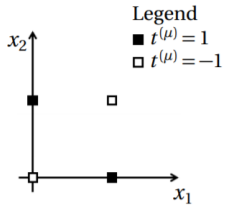
$$= H + 2b^{(v)}_{m1} v_{m1} + 2b^{(v)}_{m2} v_{m2} + \dots \tag{13}$$

$$= H - 2b^{(v)}_{m1} \mathrm{sgn}\left[b^{(v)}_{m1}\right] - 2b^{(v)}_{m2} \mathrm{sgn}\left[b^{(v)}_{m1}\right] - \dots \tag{14}$$

Chapter 5

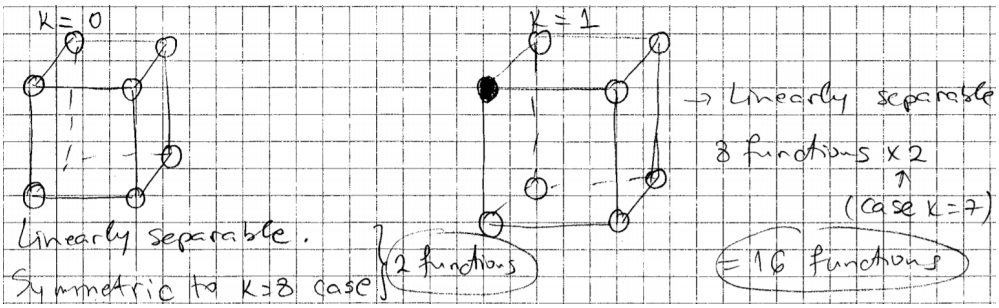
5.2 Boolean functions

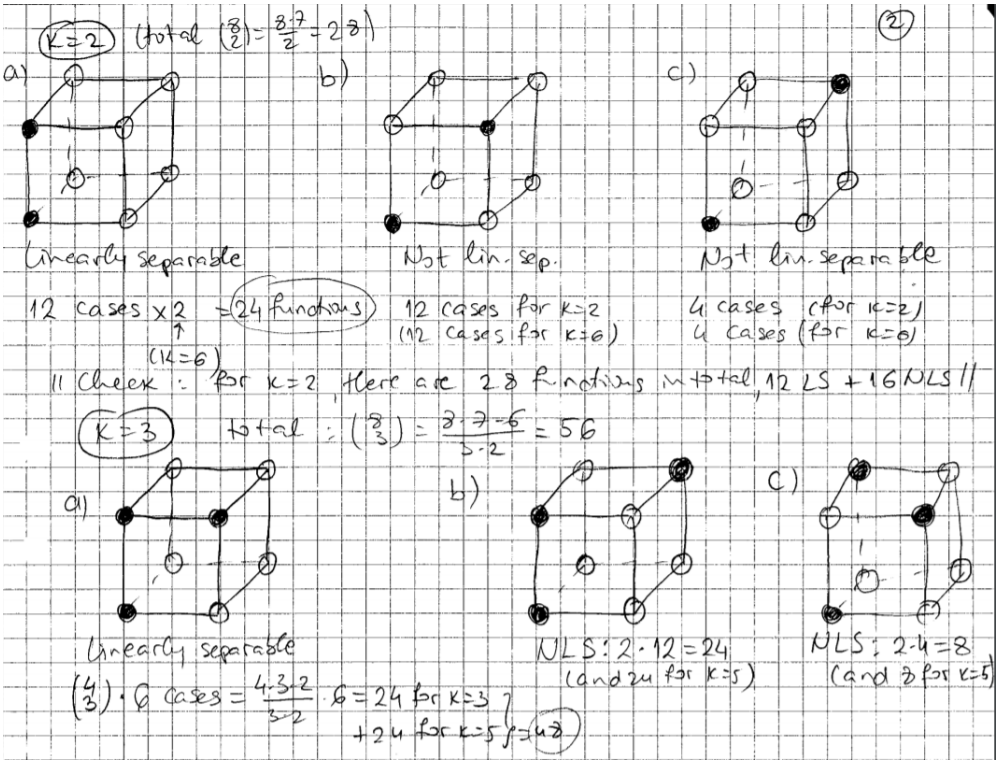
(a)

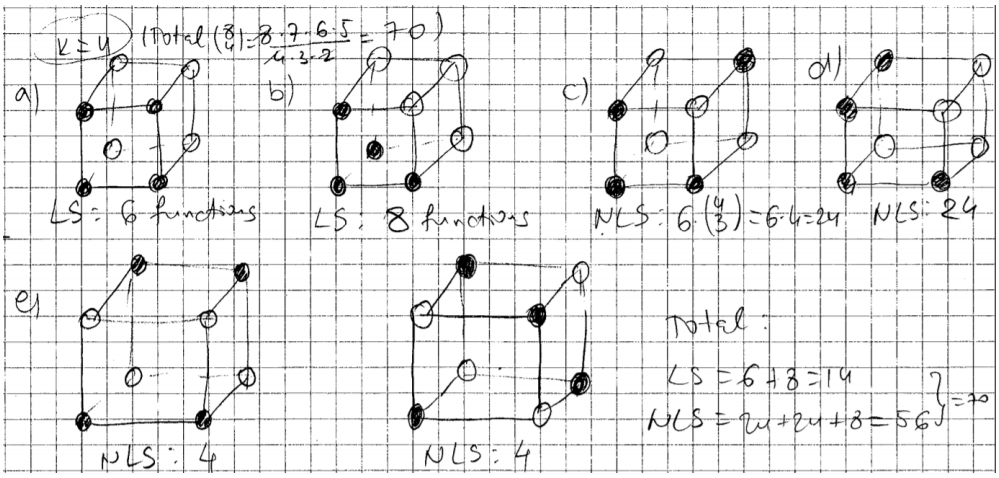


In order for the problem to be solved by a simple perceptron with two input terminals and one output unit (no hidden layer), it must be linearly separable, i.e. there must exist a plane such that all inputs mapping to the target output $+1$ have to be on one side of the plane and all inputs mapping to the target output -1 have to be on the other side of the plane. By looking at the figure above, this is impossible for the Boolean XOR problem.

(b) 3D Boolean functions. Consider a function that has k input patterns that map to the target output $+1$ and $8-k$ input patterns that map to -1.




K=2 total $\binom{8}{2}=\frac{8\cdot7}{2}=28$
a)
b)
c)
linearly separable
Not lin. sep.
Not lin. separable
12 cases x 2 = 24 functions
(1K=6)
12 cases for K=2
(12 cases for K=6)
4 cases (for K=2)
4 cases (for K=6)
// Check: For K=2 there are 28 functions in total, 12 LS + 16 NLS //
K=3 total: $\binom{8}{3}=\frac{8\cdot7\cdot6}{3\cdot2}=56$
a)
b)
c)
linearly separable
$\binom{4}{3}\cdot 6$ cases $=\frac{4\cdot3\cdot2}{3\cdot2}\cdot 6=24$ for K=3
+24 for K=5 } =48
NLS: 2·12=24
(and 24 for K=5)
NLS: 2·4=8
(and 8 for K=5)


K=4 (Total $\binom{8}{4}=\frac{8\cdot7\cdot6\cdot5}{4\cdot3\cdot2}=70$)
a)
b)
c)
d)
LS: 6 functions
LS: 8 functions
NLS: $6\binom{4}{3}=6\cdot4=24$
NLS: 24
e)
NLS: 4
NLS: 4
Total:
LS = 6+8=14
NLS = 24+24+8=56 } =70

5.5 Boolean functions

(a) Here, $N=3$. Using that $\theta_i=2$ and $w_{jk}=x_k^{(j)}$, we conclude that

$$-\theta_i+\sum_k w_{ik}x_k^{(\mu)}=-2+\sum_k x_k^{(i)}x_k^{(\mu)}\begin{cases}>0 \text{ if } i=\mu,\\ \leq 0 \text{ if } i\neq\mu.\end{cases}$$

Thus

$$v^{(i,\mu)}=\begin{cases}1 \text{ if } i=\mu,\\ 0 \text{ if } i\neq\mu.\end{cases}$$

From figure 5.16 in the lecture notes, we can see that there are exactly 4 of the 8 possible inputs $\boldsymbol{x}^{(\mu)}$ that are to be mapped to $O^{(\mu)}=1$. These are $\boldsymbol{x}^{(\mu)}$ for $\mu=2,4,5$ and 7. These inputs will assign, respectively:

$$\boldsymbol{v}^{(2)}=\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix},\boldsymbol{v}^{(4)}=\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix},\boldsymbol{v}^{(5)}=\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}, \text{ and } \boldsymbol{v}^{(7)}=\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}.$$

The weights $\boldsymbol{W}$ must satisfy the following

$$\boldsymbol{O}^{(\mu)}=\mathbb{W}^T\boldsymbol{v}^{(\mu)}=\begin{cases}1 \text{ for } \mu\in\{2,4,5,7\},\\ -1 \text{ for } \mu\in\{1,3,6,8\}.\end{cases}$$

This is achieved by letting

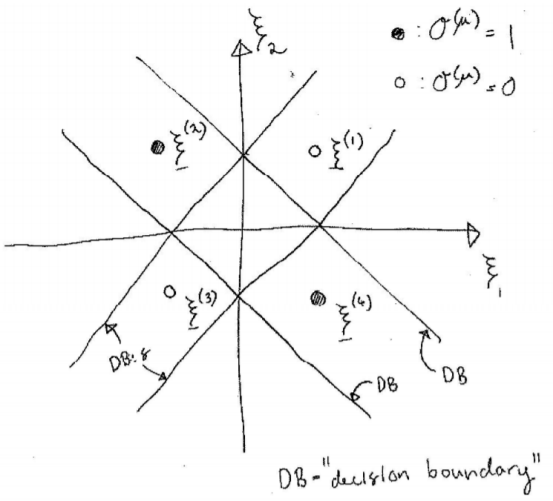
$$\boldsymbol{W}=\begin{bmatrix}-1\\1\\-1\\1\\1\\-1\\1\\-1\end{bmatrix}.$$

Note that this solution only works for $N=3$. If $N=2$, there is no solution since $-2+\sum_k x_k^{(j)}x_k^{(\mu)}\leq 0$ for all i and μ, i.e. $v^{(i,\mu)}=0$ for all i and μ.

(b) The solution in a) implies separating each corner $\boldsymbol{x}^{(\mu)}$ of the cube of input patterns by letting

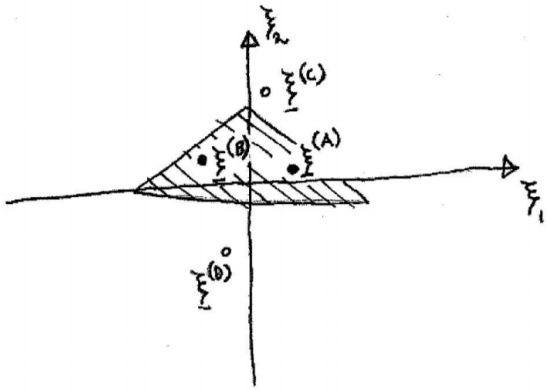
$$v^{(i,\mu)} = \begin{cases} 1 \text{ if } i = \mu, \\ 0 \text{ if } i \neq \mu. \end{cases}$$

Thus, the solution requires $2^3 = 8$ hidden neurons. The analogous solution in 2D is to separate each corner of a square, and it requires $2^N = 2^2 = 4$ neurons. The decision boundaries of the hidden neurons are shown here:



5.6 Linearly inseparable problem

(a)



In the figure above, $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}^{(B)}$ are to have output 1 and $\boldsymbol{x}^{(C)}$ and $\boldsymbol{x}^{(D)}$ are to have output 0. There is no straight line that can separate patterns $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}^{(B)}$ from patterns $\boldsymbol{x}^{(C)}$ and $\boldsymbol{x}^{(D)}$.

(b) The triangle corners are:

$$\boldsymbol{x}^{(1)}=\begin{bmatrix}-4\\0\end{bmatrix},\boldsymbol{x}^{(2)}=\begin{bmatrix}4\\-1\end{bmatrix},\boldsymbol{x}^{(3)}=\begin{bmatrix}0\\3\end{bmatrix}.$$

Let $b_1=0$ at $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$. Here, b_1 is the argument of the activation function for the hidden neuron one. This implies

$$\begin{cases}0=w_{11}x_1^{(1)}+w_{12}x_2^{(1)}-\theta_1=-4w_{11}-\theta_1\\0=w_{11}x_1^{(2)}+w_{12}x_2^{(2)}-\theta_1=4w_{11}-w_{12}-\theta_1\end{cases}\implies\begin{cases}\theta_1=-4w_{11}\\w_{12}=4w_{11}-\theta_1=8w_{11}\end{cases}$$

We choose $w_{11}=1$, $w_{12}=8$ and $\theta_1=-4$. Now, let $b_2=0$ at $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(3)}$. This implies

$$\begin{cases}0=w_{21}x_1^{(2)}+w_{22}x_2^{(2)}-\theta_2=-4w_{21}-w_{22}-\theta_2\\0=w_{21}x_1^{(3)}+w_{22}x_2^{(3)}-\theta_1=3w_{22}-\theta_2\end{cases}\implies\begin{cases}w_{22}=4w_{21}-\theta_2=4w_{21}-3w_{22}\implies w_{22}=w_{21}\\\theta_2=3w_{22}\end{cases}$$

We choose $w_{21}=w_{22}=1$ and $\theta_2=3$. Now, let $b_3=0$ at $\boldsymbol{x}^{(3)}$ and $\boldsymbol{x}^{(1)}$. This implies

$$\begin{cases}0=w_{31}x_1^{(3)}+w_{32}x_2^{(3)}-\theta_2=-3w_{32}-\theta_3\\0=w_{31}x_1^{(1)}+w_{32}x_2^{(1)}-\theta_3=-4w_{31}-\theta_3\end{cases}\implies\begin{cases}3w_{32}=-4w_{31}\\\theta_3=3w_{32}\end{cases}$$

We choose $w_{32}=4$, $w_{31}=-3$ and $\theta_3=12$.

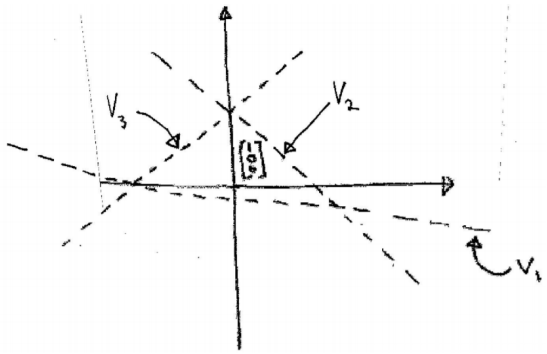
In summary:

$$\boldsymbol{w} = \begin{bmatrix} 1 & 8 \\ 1 & 1 \\ -3 & 4 \end{bmatrix} \text{ and } \boldsymbol{\theta} = \begin{bmatrix} -4 \\ 3 \\ 12 \end{bmatrix}.$$

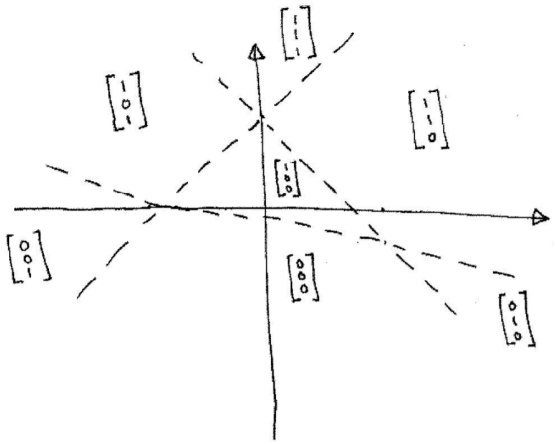
The origin maps to

$$\boldsymbol{V} = H(\boldsymbol{w} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \boldsymbol{\theta}) = H\left(\begin{bmatrix} 4 \\ -3 \\ -12 \end{bmatrix} \right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

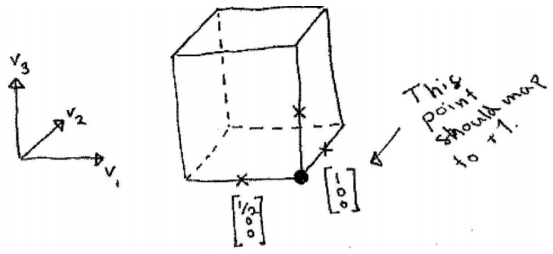
We know that the origin maps to $\boldsymbol{V} = [1,0,0]^T$ and that the hidden neurons change values at the dashed lines in the following figure:



Thus, we can conclude that the regions in input space maps to the following regions in the hidden space:



We want $\boldsymbol{V} = [1,0,0]^T$ to map to 1 and all other possible values of $\boldsymbol{V}$ to map to 0. The hidden space can be illustrated like this:



$\boldsymbol{W}$ must be normal to the plane passing through the crosses in this picture. Also, $\boldsymbol{W}$ points to $\boldsymbol{V} = [1,0,0]^T$ from $\boldsymbol{V} = [0,1,1]^T$. We may choose

$$\boldsymbol{W} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}.$$

We know that the point $\boldsymbol{V} = [\frac{1}{2},0,0]^T$ lies on the decision boundary we are looking for. So

$$\boldsymbol{W}^T \cdot \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \end{bmatrix} - \Theta = 0 \Longrightarrow \Theta = \frac{1}{2}.$$

5.7 Perceptron with one hidden layer

a) $\boldsymbol{W} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}, \Theta = \frac{1}{2},$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \Theta = -\frac{1}{2} \implies \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ maps to } O = 0,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \Theta = 1 - \frac{1}{2} = \frac{1}{2} \implies \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ maps to } O = 1,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \Theta = 2 - \frac{1}{2} = \frac{3}{2} \implies \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \text{ maps to } O = 1,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \Theta = -1 - \frac{1}{2} = -\frac{3}{2} \implies \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ maps to } O = 0,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} - \Theta = 0 - \frac{1}{2} = \frac{1}{2} \implies \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ maps to } O = 0,$$

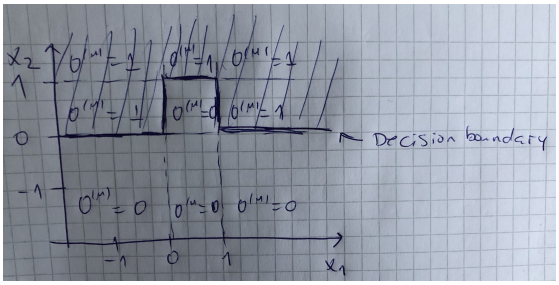
$$\boldsymbol{W}^T \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \Theta = 2 - \frac{1}{2} = \frac{3}{2} \implies \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ maps to } O = 1,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \Theta = 0 - \frac{1}{2} = -\frac{1}{2} \implies \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ maps to } O = 0,$$

$$\boldsymbol{W}^T \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} - \Theta = 1 - \frac{1}{2} = \frac{1}{2} \implies \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ maps to } O = 1,$$

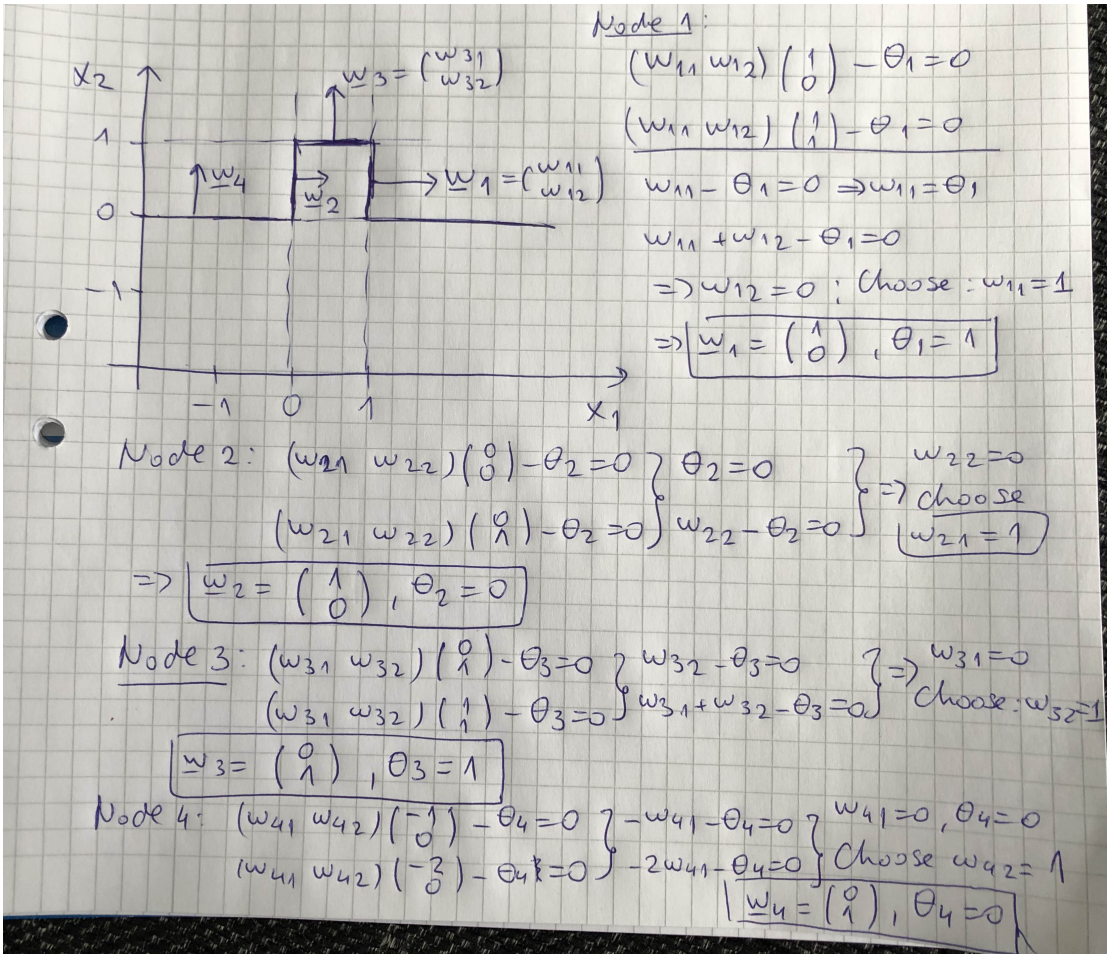
$$\boldsymbol{W}^T \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \Theta = 2 - \frac{1}{2} = \frac{3}{2} \implies \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ maps to } O = 1.$$

Illustration:



(b) Consider the sequence of patterns $\boldsymbol{x}^{(1)} = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix}$, $\boldsymbol{x}^{(2)} = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}$, and $\boldsymbol{x}^{(3)} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$. The corresponding sequence of $v_3^{(\mu)}$ is $v_3^{(1)} = 0$, $v_3^{(2)} = 1$ and $v_3^{(3)} = 0$. For a monotonously increasing sequence of patterns where $x_1^{(\mu)}$ stays constant and $x_2^{(\mu)}$ is monotonously increasing, the function $-\Theta_3 + \sum_j w_{3j} x_j^{(\mu)}$ must increase (decrease) monotonously if w_{32} is positive (negative) or stay constant if $w_{32} = 0$. This isn't the case for the sequence of patterns $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(3)}$.

(c) The solution is shown in the following figure:



5.8 Multilayer perceptron

The solid line passes through the points $(0,1)$ and $(1,0)$:

$$(1,0) \Longrightarrow w_{11}\cdot 1 + w_{12}\cdot 0 - \theta_1 = 0 \Longrightarrow w_{11} = \theta_1,$$

$$(0,1) \Longrightarrow w_{11}\cdot 0 + w_{12}\cdot 1 - \theta_1 = 0 \Longrightarrow w_{12} = \theta_1.$$

So $w_{11} = w_{12} = \theta_1$. The dash-dotted line passes through the points $(0.5,1)$ and $(0.5,0)$:

$$(0.5,0) \Longrightarrow 0.5\,w_{21} + 0\cdot w_{22} - \theta_2 = 0 \Longrightarrow \theta_2 = 0.5\,w_{21}$$

$$(0.5,1) \Longrightarrow 0.5\,w_{21} + w_{22} - \theta_2 = 0 \Longrightarrow w_{22} = 0.$$

So $w_{22} = 0$ and $\theta_2 = 0.5\,w_{21}$. The dashed line passes through the points $(0,0.8)$ and $(1,0.8)$:

$$(0,0.8) \Longrightarrow 0\cdot w_{31} + 0.8\,w_{32} - \theta_3 = 0 \Longrightarrow \theta_3 = 0.8\,w_{32}$$

$$(1,0.8) \Longrightarrow 1 \cdot w_{31} + 0.8 w_{32} - \theta_3 = 0 \Longrightarrow w_{31} = 0.$$

So $w_{31} = 0$ and $\theta_3 = 0.8 w_{32}$.

These conditions are realized by, for instance, $\boldsymbol{w} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \theta = \begin{bmatrix} 1 \\ 0.5 \\ 0.8 \end{bmatrix}$.

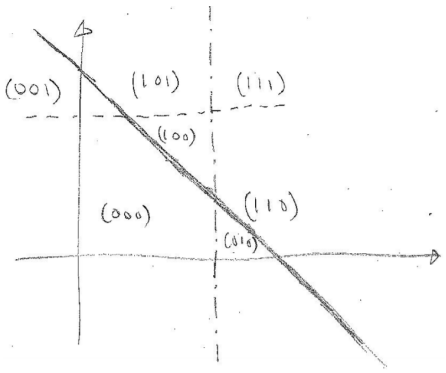
Evaluate the output at $(x_1, x_2) = (0,0)$:

$$V_1 = \theta\left[1 \cdot 0 + 1 \cdot 0 - 1\right] = 0$$

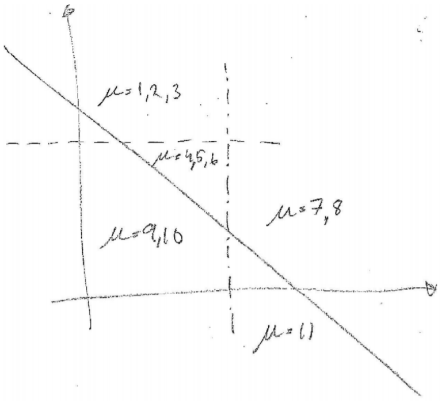
$$V_2 = \theta\left[1 \cdot 0 + 0 \cdot 0 - 0.5\right] = 0$$

$$V_3 = \theta\left[0 \cdot 0 + 1 \cdot 0 - 0.8\right] = 0.$$

So, the origin in the input space maps to $\boldsymbol{V} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. Note that V_1 flips at the solid line, V_2 flips at the dash-dotted line and V_3 flips at the dashed line. Given that the origin in the input space is $(0,0,0)$, we find:



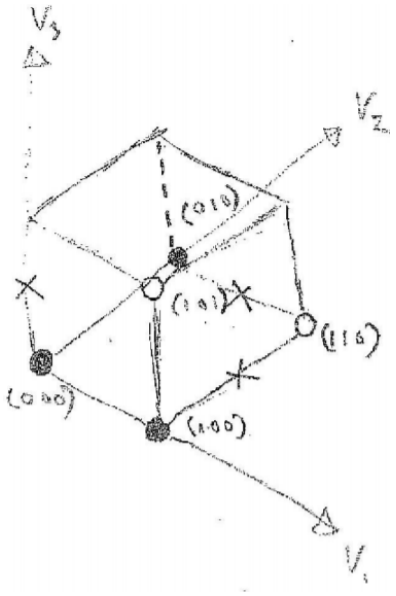
The input patterns are located according to this figure:



So the coordinates in the transformed space are:

μ	$\boldsymbol{V}$
1,2,3	(1,0,1)
4,5,6	(1,0,0)
7,8	(1,1,0)
9,10	(0,0,0)
11	(0,1,0)

Graphical illustration of hidden space:



Yes, the problem is linearly separable since we can separate the output with the plane passing through the crosses in the figure.

e) The crosses drawn at the last figure in assignment (d) are $P_1 = (1, \frac{1}{2}, 0)$, $P_2 = (\frac{1}{2}, 1, 0)$ and $P_3 = (0, 0, \frac{1}{2})$. Decision boundary at plane containing these points imply $W_1 V_1 + W_2 V_2 + W_3 V_3 - \Theta = 0$:

$$P_1 \implies W_1 + \frac{1}{2} W_2 = \Theta \tag{15}$$

$$P_2 \implies \frac{1}{2} W_1 + W_2 = \Theta \tag{16}$$

$$P_3 \implies \frac{1}{2}W_3 = \Theta \tag{17}$$

$$(15)+(16) \implies W_1 + \frac{1}{2}W_2 = \frac{1}{2}W_1 + W_2 \implies W_1 = W_2 \tag{18}$$

$$(15)+(18) \implies \frac{3}{2}W_1 = \frac{3}{2}W_2 = \Theta \tag{19}$$

$$(17)+(19) \implies \frac{1}{2}W_3 = \frac{3}{2}W_1 = \frac{3}{2}W_2 = \Theta \tag{20}$$

Note that we want the origin of the transformed space to be mapped to $+1$ (from patterns 9 and 10). Thus, $0 \cdot W_1 + 0 \cdot W_2 + 0 \cdot W_3 - \Theta > 1 \implies \Theta < -1$. We may choose $\Theta = -2$, $\boldsymbol{W} = \begin{bmatrix} -\frac{4}{3} \\ -\frac{4}{3} \\ -4 \end{bmatrix}$.

Chapter 6

6.2 Principal-component analysis

Check that $\langle x_j \rangle = 0$:

$$\sum_{\mu=1}^{5} x_1^{(\mu)} = -6-2+1+1+5 = 0\,,$$

$$\sum_{\mu=1}^{5} x_2^{(\mu)} = -5-4+2+3+4 = 0\,.$$

So the data has zero mean in all components. This means that the matrix $\mathbb{C}'$ equals the data-covariance matrix $\mathbb{C}$. Therefore drop the prime in the following:

$$\mathbb{C} = \frac{1}{5}\sum_{\mu=1}^{5} \boldsymbol{x}^{(\mu)}\boldsymbol{x}^{(\mu)\mathsf{T}} \quad \text{(because data has zero mean)}.$$

We have

$$\boldsymbol{x}^{(1)}\boldsymbol{x}^{(1)\mathsf{T}} = \begin{bmatrix} 36 & 30 \\ 30 & 25 \end{bmatrix},$$

$$\boldsymbol{x}^{(2)}\boldsymbol{x}^{(2)\mathsf{T}} = \begin{bmatrix} 4 & 8 \\ 8 & 16 \end{bmatrix},$$

$$\boldsymbol{x}^{(3)}\boldsymbol{x}^{(3)\mathsf{T}} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix},$$

$$\boldsymbol{x}^{(4)}\boldsymbol{x}^{(4)\mathsf{T}} = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix},$$

$$\boldsymbol{x}^{(5)}\boldsymbol{x}^{(5)\mathsf{T}} = \begin{bmatrix} 25 & 20 \\ 20 & 16 \end{bmatrix},$$

We compute the elements of $\mathbb{C}$:

$$5C_{11} = 36+4+4+1+25 = 70$$

$$5C_{12} = 5C_{21} = 30+8+4+3+20 = 65$$

$$5C_{22} = 25+16+4+9+16 = 70$$

We find

$$\mathbb{C} = \begin{bmatrix} -13 & 13 \\ 13 & -13 \end{bmatrix}.$$

Eigenvalues:

$$0 = \begin{vmatrix} 14-\lambda & 13 \\ 13 & 14-\lambda \end{vmatrix} = (14-\lambda)^2 - 13^2 = \lambda^2 - 28\lambda + 27 \implies \lambda = 14 \pm \sqrt{14^2 - 27} = 14 \pm \sqrt{169} = 14 \pm 13.$$

This gives the maximal eigenvalue $\lambda_{\max} = 27$. Eigenvector $\boldsymbol{u}$:

$$\begin{bmatrix} 14 & 13 \\ 13 & 14 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0 \quad \implies u_1 = u_2.$$

So the normalised eigenvector corresponding to the largest eigenvalue of $\mathbb{C}$ is given by

$$\boldsymbol{u} = \frac{\pm 1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

This is the principal component.

6.7 Backpropagation

Discussion. Refer to Section 6 in Lecture notes.

6.8 Stochastic gradient descent

Solution starts on the next page.

③ $H = \frac{1}{2} \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})^2$

$V_\ell^{(1,\mu)} = g(b_\ell^{(1,\mu)})$, $b_\ell^{(1,\mu)} = \sum_s w_{\ell s}^{(1)} x_s^{(\mu)} - \theta_\ell^{(1)}$

$V_i^{(2,\mu)} = g(b_i^{(2,\mu)})$, $b_i^{(2,\mu)} = \sum_\ell w_{i\ell}^{(2)} V_\ell^{(1,\mu)} - \theta_i^{(2)}$

$O^{(\mu)} = g(b_1^{(\mu)})$, $b_1^{(\mu)} = \sum_i W_{1i} V_i^{(2,\mu)} - \textcircled{H}_1$

} Forward propagation

Back-propagation

$\delta W_{1m} = -\eta \frac{\partial H}{\partial W_{1m}}$, $\eta > 0$, learning rate

$\delta \textcircled{H}_1 = -\eta \frac{\partial H}{\partial \textcircled{H}_1}$

$\delta w_{mj}^{(2)} = -\eta \frac{\partial H}{\partial w_{mj}^{(2)}}$

$\delta \theta_m^{(2)} = -\eta \frac{\partial H}{\partial \theta_m^{(2)}}$

$\delta w_{jk}^{(1)} = -\eta \frac{\partial H}{\partial w_{jk}^{(1)}}$

$\delta \theta_j^{(1)} = -\eta \frac{\partial H}{\partial \theta_j^{(1)}}$

$\delta W_{1m} = -\eta \frac{\partial H}{\partial O^{(\mu)}} \frac{\partial O^{(\mu)}}{\partial W_{1m}} = +\eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)}) \cdot \frac{dO^{(\mu)}}{db_1^{(\mu)}} \frac{\partial b_1^{(\mu)}}{\partial W_{1m}} =$

$= \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)}) \cdot g'(b_1^{(\mu)}) \cdot \frac{\partial}{\partial W_{1m}} \left(\sum_i W_{1i} V_i^{(2,\mu)} - \textcircled{H}_1 \right) \Rightarrow$

$$\delta W_{1m} = \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)}) \sum_i \delta_{im} V_i^{(2,\mu)} =$$

$$\delta_{im} = \begin{cases} 1, & i = m \\ 0, & i \neq m \end{cases}$$

$$\boxed{\delta W_{1m} = \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)})\, V_m^{(2,\mu)}}$$

$$\delta \Theta_1 = -\eta \frac{\partial H}{\partial O^{(\mu)}} \frac{\partial O^{(\mu)}}{\partial b_1} = \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)}) \frac{dO^{(\mu)}}{db_1^{(\mu)}} \frac{\partial b_1^{(\mu)}}{\partial \Theta_1} =$$

$$= \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)})\, (-1)$$

$$\boxed{\delta \Theta_1 = -\eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)})}$$

$$\delta w_{mj}^{(2)} = -\eta \frac{\partial H}{\partial O^{(\mu)}} \frac{dO^{(\mu)}}{db_1^{(\mu)}} \frac{\partial b_1^{(\mu)}}{\partial w_{mj}^{(2)}} =$$

$$= \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)}) \sum_i W_{1i}\, g'(b_i^{(2,\mu)}) \frac{\partial b_i^{(2,\mu)}}{\partial w_{mj}^{(2)}} =$$

$$= \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)}) g'(b_1^{(\mu)}) \sum_i W_{1i}\, g'(b_i^{(2,\mu)}) \sum_\ell \delta_{im} \delta_{\ell j} V_\ell^{(1,\mu)} =$$

$$\boxed{\delta w_{mj}^{(2)} = \eta \sum_{\mu=1}^{P} (t^{(\mu)} - O^{(\mu)})\, g'(b_1^{(\mu)})\, W_{1m}\, g'(b_m^{(2,\mu)})\, V_j^{(1,\mu)}}$$

$$\delta\theta_m^{(2)} = -\eta \frac{\partial H}{\partial O^{(M)}} \frac{dO^{(M)}}{db_1^{(M)}} \cdot \frac{\partial b_1^{(M)}}{\partial \theta_m^{(2)}} =$$

$$= \eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)}) \cdot g'(b_1^{(M)}) \sum_i W_{1i}\, g'(b_i^{(2,M)})\, \delta_{im}(-1)$$

$$\boxed{\delta\theta_m^{(2)} = -\eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)})\, g'(b_1^{(M)})\, W_{1m}\, g'(b_m^{(2,M)})}$$

$$\delta w_{jk}^{(1)} = -\eta \frac{\partial H}{\partial w_{jk}^{(1)}} = -\eta \frac{\partial H}{\partial O^{(M)}} \frac{\partial O^{(M)}}{\partial b_1^{(M)}} \frac{\partial b_1^{(M)}}{\partial w_{jk}^{(1)}}$$

$$= +\eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)})\, g'(b_1^{(M)}) \sum_i W_{1i}\, g'(b_i^{(2,M)}) \frac{\partial b_i^{(2,M)}}{\partial w_{jk}^{(1)}} =$$

$$= \eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)})\, g'(b_1^{(M)}) \sum_i W_{1i}\, g'(b_i^{(2,M)}) \sum_\ell w_{i\ell}^{(2)}\, g'(b_\ell^{(1,M)}) \cdot$$

$$\cdot \frac{\partial b_\ell^{(1,M)}}{\partial w_{jk}^{(1)}} =$$

$$= \eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)})\, g'(b_1^{(M)}) \sum_i W_{1i}\, g'(b_i^{(2,M)}) \sum_\ell w_{i\ell}^{(2)}\, g'(b_\ell^{(1,M)}) \cdot$$

$$\cdot \delta_{\ell j}\delta_{sk} x_s^{(M)} =$$

$$\boxed{\delta w_{jk}^{(1)} = \eta \sum_{M=1}^{P} (t^{(M)} - O^{(M)})\, g'(b_1^{(M)}) \sum_i W_{1i}\, g'(b_i^{(2,M)})\, w_{ij}^{(2)}\, g'(b_j^{(1,M)})\, x_k^{(M)}}$$

$$\delta\theta_j^{(1)} = -\eta \frac{\partial H}{\partial \theta_j^{(1)}} = \ldots \text{ instead of } \frac{\partial b_\ell^{(1,M)}}{\partial w_{jk}^{(1)}} \text{ use } \frac{\partial b_\ell^{(1,M)}}{\partial \theta_j^{(1)}}$$

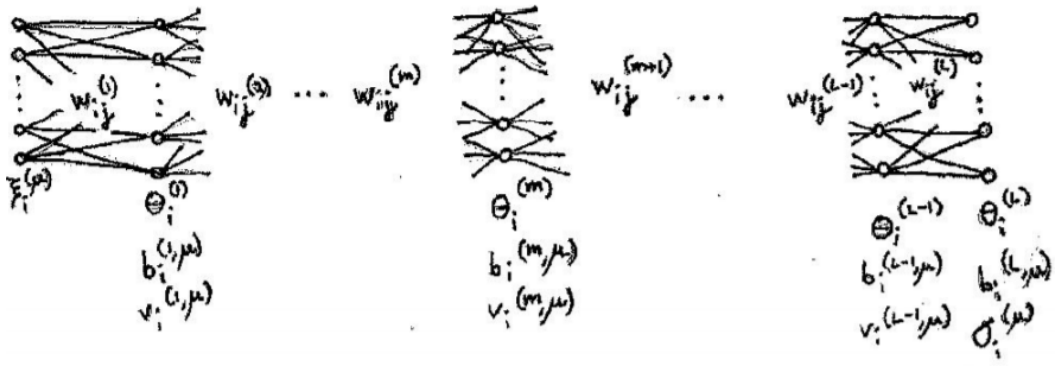
Find:

$$\delta\theta_j^{(1)} = -\eta \sum_{\mu=1}^{p} (t^{(\mu)} - O^{(\mu)}) g'(b_1^{(\mu)}) \sum_i W_{1i} g'(b_i^{(2,\mu)}) w_{ij}^{(2)} g'(b_j^{(1,\mu)})$$

If batch mode → summation over μ remains.

If stochastic gradient-descent : each update is based on a single randomly chosen pattern $\underline{\mu}$ (summation over $\underline{\mu}$ should be omitted).

6.9 Multi-layer perceptron



Let N_m denote the number of weights $w_{ij}^{(m)}$ in the mth layer. Let n_m denote the number of hidden units $v_i^{(m,\mu)}$ for $i = 1,\ldots,L-1$, let n_0 denote the number of input units and let n_L denote the number of output units. The number of weights are $\sum_{m=1}^{L} N_m = \sum_{m=1}^{L} n_{m-1} n_m$ and the number of thresholds are $\sum_{m=1}^{L} n_m$.

$$\begin{aligned}\frac{\partial}{\partial w_{qr}^{(p)}} v_i^{(m,\mu)} &= \frac{\partial}{\partial w_{qr}^{(p)}} g\Big(b_i^{(m,\mu)}\Big) = g'\Big(b_i^{(m,\mu)}\Big) \frac{\partial}{\partial w_{qr}^{(p)}} b_i^{(m,\mu)} \\ &= g'\Big(b_i^{(m,\mu)}\Big) \frac{\partial}{\partial w_{qr}^{(p)}} \Big(-\theta_i^{(m)} + \sum_j w_{ij}^{(m)} v_j^{(m-1,\mu)}\Big) \\ &= g'\Big(b_i^{(m,\mu)}\Big) \Big(\sum_j \frac{\partial}{\partial w_{qr}^{(p)}} w_{ij}^{(m)} v_j^{(m-1,\mu)}\Big).\end{aligned}$$

Using that $p < m$, we obtain

$$\frac{\partial}{\partial w_{qr}^{(p)}} v_i^{(m,\mu)} = g'\Big(b_i^{(m,\mu)}\Big) \Big(\sum_j w_{ij}^{(m)} \frac{\partial v_j^{(m-1,\mu)}}{\partial w_{qr}^{(p)}}\Big). \tag{21}$$

We have $\frac{\partial}{\partial w_{qr}^{(p)}} v_i^{(m,\mu)} = g'\Big(b_i^{(m,\mu)}\Big) \Big(\sum_j \frac{\partial}{\partial w_{qr}^{(p)}} w_{ij}^{(m)} v_j^{(m-1,\mu)}\Big)$. But since $p = m$, we find:

$$\frac{\partial}{\partial w_{qr}^{(p)}} v_i^{(m,\mu)} = g'\Big(b_i^{(m,\mu)}\Big) \sum_j \delta_{qi} \delta_{rj} v_j^{(m-1,\mu)} = g'\Big(b_i^{(m,\mu)}\Big) \delta_{qi} v_r^{(m-1,\mu)}. \tag{22}$$

We have $w_{qr}^{(L-2)} \leftarrow w_{qr}^{(L-2)} + \delta w_{qr}^{(L-2)}$, where

$$\delta w_{qr}^{(L-2)} = -\eta \frac{\partial H}{\partial w_{qr}^{(L-2)}}.$$

From the definition of the energy function, we have

$$\frac{\partial H}{\partial w_{qr}^{(L-2)}} = \frac{\partial}{\partial w_{qr}^{(L-2)}} \frac{1}{2} \sum_{\mu} \sum_{i} \left(O_i^{(\mu)} - t_i^{(\mu)} \right)^2 \tag{23}$$

$$= \sum_{\mu} \sum_{i} \left(O_i^{(\mu)} - t_i^{(\mu)} \right) \frac{\partial O_i^{(\mu)}}{\partial w_{qr}^{(L-2)}} \tag{24}$$

We have:

$$\frac{\partial v_i^{(m,\mu)}}{\partial w_{qr}^{(p)}} = \begin{cases} g'\left(b_i^{(m,\mu)}\right) \sum_j w_{ij}^{(m)} \frac{\partial v_j^{(m-1,\mu)}}{\partial w_{qr}^{(p)}} \text{ if } p < m \\ g'\left(b_i^{(m,\mu)}\right) \delta_{qi} v_r^{(m-1,\mu)} \text{ if } p = m. \end{cases} \tag{25}$$

Define $v_i^{(L,\mu)} = O_i^{(\mu)}$. We have:

$$\begin{aligned} \frac{\partial O_i^{(\mu)}}{\partial w_{qr}^{(L-2)}} &= \frac{\partial v_i^{(L,\mu)}}{\partial w_{qr}^{(L-2)}} \\ &\quad \{\text{Insert from (25). Use that } L-2 < L. \} \\ &= g'\left(b_i^{(L,\mu)}\right) \sum_j w_{ij}^{(L)} \frac{\partial v_j^{(L-1,\mu)}}{\partial w_{qr}^{(L-2)}} \\ &\quad \{\text{Insert from (25). Use that } L-2 < L-1. \} \\ &= g'\left(b_i^{(L,\mu)}\right) \sum_j w_{ij}^{(L)} g'\left(b_j^{(L-1,\mu)}\right) \sum_k w_{jk}^{(L-1)} \frac{\partial v_k^{(L-2,\mu)}}{\partial w_{qr}^{(L-2)}} \\ &\quad \{\text{Insert from (25). Use that } L-2 = L-2. \} \\ &= g'\left(b_i^{(L,\mu)}\right) \sum_j w_{ij}^{(L)} g'\left(b_j^{(L-1,\mu)}\right) \sum_k w_{jk}^{(L-1)} g'\left(b_k^{(L-2,\mu)}\right) \delta_{qk} v_r^{(L-3,\mu)} \\ &= g'\left(b_i^{(L,\mu)}\right) \sum_j w_{ij}^{(L)} g'\left(b_j^{(L-1,\mu)}\right) w_{jq}^{(L-1)} g'\left(b_q^{(L-2,\mu)}\right) v_r^{(L-3,\mu)}. \end{aligned}$$

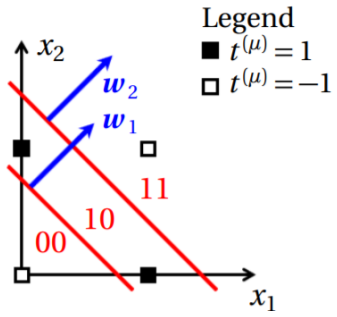
Chapter 7

7.5 Parity function

According to Lecture notes, the thresholds in the hidden layer could be $\frac{1}{2}$ and $\frac{3}{2}$ and all weights (in the hidden layer) equal to unity. The output neuron has weights ± 1 and threshold $\frac{1}{2}$:

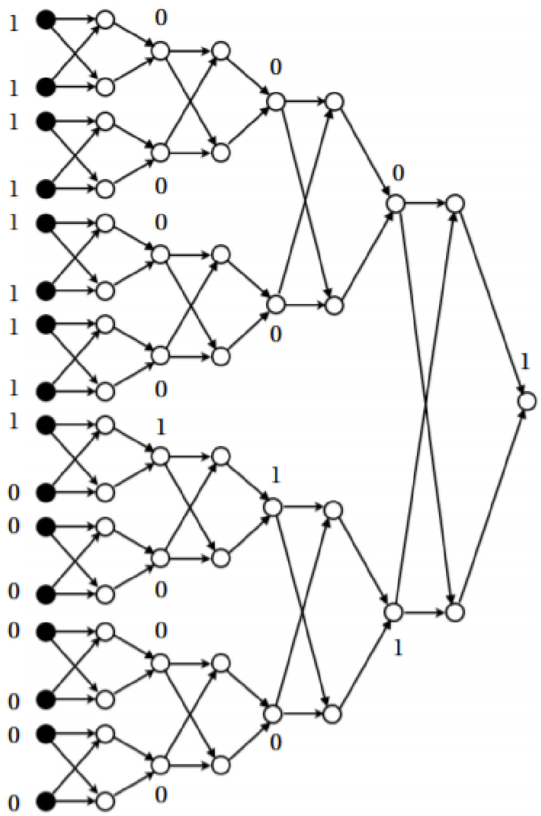
$$O_1 = \theta_H\Big(V_1 - V_2 - \frac{1}{2}\Big).$$

Graphical illustration of the input space:



The hidden space has the following coordinates:

V_1	V_2	t
0	0	-1
1	0	+1
0	1	+1
1	1	-1



Solution of the parity problem for N-dimensional inputs. The network is built from XOR units. Each XOR unit has a hidden layer with two neurons. Above only the states of the inputs and outputs of the XOR units are shown, not those of the hidden neurons. In total, the whole network has O(N) neurons.

7.7 Softmax outputs

a) $b_m^{(L,\mu)} = -\theta_m^{(L)} + \sum_k w_{mk}^{(L)} V_k^{(L-1,\mu)}$:

$$\frac{\partial O_i^{(\mu)}}{\partial b_j^{(L,\mu)}} = \left(\sum_m \exp\left(b_m^{(L,\mu)}\right)\right)^{-1} \exp\left(b_i^{(L,\mu)}\right)\delta_{ij} + \exp\left(b_i^{(L,\mu)}\right)(-1)\left(\sum_m \exp\left(b_m^{(L,\mu)}\right)\right)^{-2} \sum_m \exp\left(b_m^{(L,\mu)}\right)\delta_{mj} =$$

$$= \frac{\delta_{ij}\exp\left(b_i^{(L,\mu)}\right)}{\sum_m \exp\left(b_m^{(L,\mu)}\right)} - \frac{\exp\left(b_i^{(L,\mu)} + b_j^{(L,\mu)}\right)}{\left(\sum_m \exp\left(b_m^{(L,\mu)}\right)\right)^2} = O_i^{(\mu)}\left(\delta_{ij} - O_j^{(\mu)}\right)$$

(b)

$$\delta w_{nq}^{(L)} = -\eta \frac{\partial H}{\partial w_{nq}^{(L)}} = \eta \sum_{i,\mu} \frac{t_i^{(\mu)}}{O_i^{(\mu)}} \frac{\partial O_i^{(\mu)}}{\partial w_{nq}^{(L)}} = \eta \sum_{i,j,\mu} \frac{t_i^{(\mu)}}{O_i^{(\mu)}} \frac{\partial O_i^{(\mu)}}{\partial b_j^{(L,\mu)}} \frac{\partial b_j^{(L,\mu)}}{\partial w_{nq}^{(L)}} =$$

$$= \eta \sum_{i,j,\mu} \frac{t_i^{(\mu)}}{O_i^{(\mu)}} \left(O_i^{(\mu)}\delta_{ij} - O_i^{(\mu)} O_j^{(\mu)}\right) \sum_k V_k^{(L-1,\mu)} \delta_{nj}\delta_{qk} =$$

$$= \eta \sum_{i,j,\mu} t_i^{(\mu)} \left(\delta_{ij} - O_j^{(\mu)}\right) V_q^{(L-1,\mu)} \delta_{nj} =$$

$$= \eta \left(\sum_\mu t_n^{(\mu)} V_q^{(L-1,\mu)} - \sum_\mu O_n^{(\mu)} V_q^{(L-1,\mu)} \underbrace{\sum_i t_i^{(\mu)}}_{=1}\right) = \ldots =$$

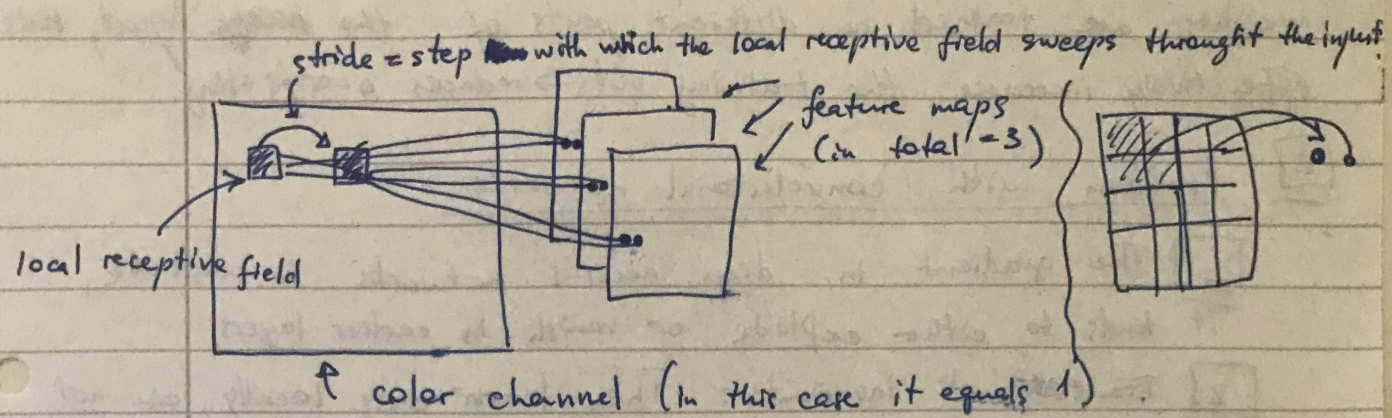
$$= \eta \sum_\mu (t_n^{(\mu)} - O_n^{(\mu)}) V_q^{(L-1,\mu)}.$$

7.8 Convolutional neural net

See next page.

Problem #3

[a] Bernhard's book, page 126.



A convolution layer sweeps through 2d input data by applying convolution operation to the input points located only within the same small areas, called receptive fields. One output corresponds to one area of input. The receptive field may in general move at different rate, called stride. The output neurons obtained from one sweep of the layer also build 2d layers. All neurons in the same 2d output layer correspond to the same weights and threshhold used for performing the convolution operation, called a feature map. In general we can separately perform convolution for different sets of weights and threshholds, creating stacks of 2d output layers. A convolution layer may also be applied to several 2d input layers („channels") using shared feature maps for all input layers. A pooling layer also sweeps through 2d input but it applies „max operation" instead of convolution. Also the stride is taken a such that the areas corresponding to the same input layer would not overlap and would cover the input.

[b] The number of neurons doesn't scale with the size of input. This is the main advantage of this network. Small number of neurons regularizes the network, it reduces

Problem #3

the risk of overfitting. Also, each feature map is applied to different parts of 2d input. This means that all weights are trained on different parts of the input, this effectively increases the training set → reduces overfitting.

c) Problem with convolutional networks:

1. The gradient in deep neural networks is unstable, it tends to either explode or vanish in earlier layers.

2. Translational invariants: They learn only locally, can not learn global features, can mistake for example a leopard-patterned sofa for a leopard. (Example from book)

7.9 Feature map

The kernel, 3×3 max pooling and two outputs gives the following matrix sizes:

- Input layer: 7×5 bits
- Feature map: 5×3 bits
- Output of max-pooling layer: 3×1 bits
- Output layer: 2×3 weight matrix and 2×1 threshold vector

2	2	2
2	3	3
3	2	2
3	2	2
3	2	1

Table 2: Feature map of pattern 1

3	2	2
3	3	3
4	2	3
3	3	3
3	2	2

Table 3: Feature map of pattern 2

3
3
3

Table 4: Max pooling of pattern 1

4
4
4

Table 5: Max pooling of pattern 2

Output layer:

With weights

$$\boldsymbol{w} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{26}$$

we have for the first and second pattern:

$$\boldsymbol{w} \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{w} \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \end{bmatrix}. \tag{27}$$

With the thresholds

$$\boldsymbol{\theta} = \begin{bmatrix} -7/2 \\ 7/2 \end{bmatrix} \tag{28}$$

we have

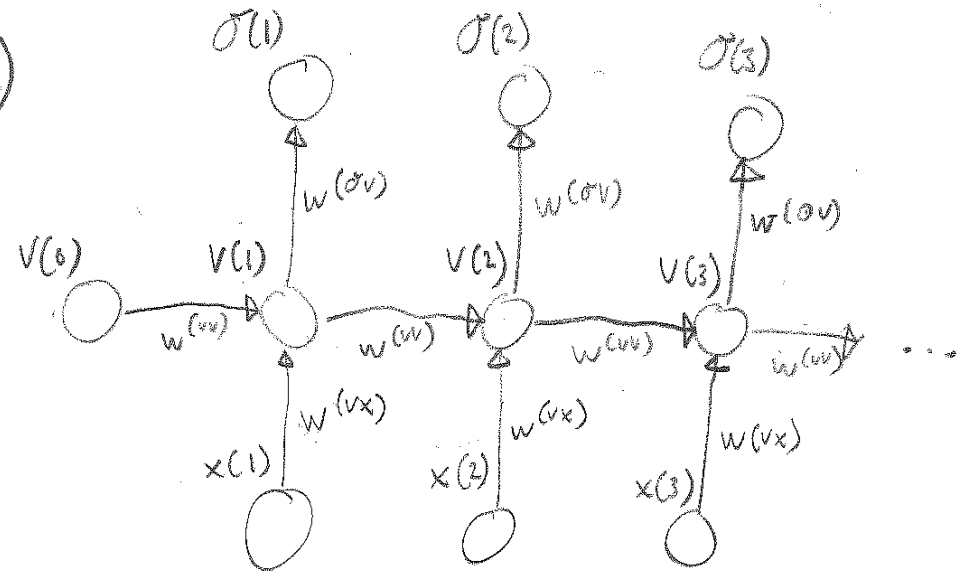
$$\boldsymbol{O}^{(1)} = g\left(\begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{O}^{(2)} = g\left(\begin{bmatrix} -1/2 \\ 1/2 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{29}$$

Chapter 8

8.3 Recurrent network

See next page.

4a)



The layout reflects that the network learns a sequence of input/output pairs.

4b)

$$V(t) = g\left(w^{(vv)} V(t-1) + w^{(vx)} x(t) - \theta^{(v)}\right)$$

$$O(t) = g\left(w^{(ov)} V(t) - \theta^{(o)}\right)$$

4c)

$$\Delta w^{(ov)} = -\eta \frac{\partial H}{\partial w^{(ov)}} \qquad (*)$$

We first compute:

$$\frac{\partial H}{\partial w^{(ov)}} = \frac{\partial}{\partial w^{(ov)}} \frac{1}{2} \sum_t E_t^2 = \sum_t E_t \frac{\partial}{\partial w^{(ov)}} \left(y(t) - O(t)\right)$$

$$= -\sum_t E_t \frac{\partial}{\partial w^{(ov)}} g\Big(\underbrace{w^{(ov)} V(t) - \theta^{(o)}}_{\equiv B(t)}\Big)$$

$$= -\sum_t E_t g'(B(t)) V(t) \qquad (**)$$

We now insert (2**) into (*):

$$\Delta w(conv) = \eta \sum_t E_t \, g'(B(t)) \, V(t)$$

4d) Trained through backpropagation throug time. Hidden units replaced by LSTM-units.

For (1): A sentence is sequentially fed to the inputs, word by word, until an end-of-sentence (EOS)-tag is encountered.

(2) The output is now the next word in the translated sentence.

* If step (2) is run for the first time the output is the first word of the translated sentence.
* If the output is the "EOS-tag": TERMINATE.

(3) Feed the output as the next input. Go to (2).

Unstable gradient problem can be overcome by limiting the error propagation backwards in time.

Chapter 9

9.3 Oja's rule

Network output $y = \boldsymbol{w}\cdot\boldsymbol{x} \equiv \sum_{j=1}^{N} w_j x_j = \boldsymbol{w}^\mathsf{T}\boldsymbol{x} = \boldsymbol{x}^\mathsf{T}\boldsymbol{w}$. Learning rule $\delta w_j = \eta y(x_j - y\,w_j)$. Steady state $\boldsymbol{w}^*$. See Lecture notes Section 9.1, and Hertz *et al.* (1991), Section 8.2.

(a) Prove that $\boldsymbol{w}^*$ maximizes $\langle y^2\rangle$ using that $|\boldsymbol{w}^*|^2 = 1$, and that $\boldsymbol{w}^*$ is the leading eigenvector of the matrix $\mathbb{C}'$ that has elements $C'_{ij} = \langle x_i x_j\rangle$ (not the covariance matrix). First, write

$$\langle y^2\rangle = \langle(\boldsymbol{w}^\mathsf{T}\boldsymbol{x})(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})\rangle = \boldsymbol{w}^\mathsf{T}\mathbb{C}'\boldsymbol{w}.$$

For $\boldsymbol{w} = \boldsymbol{w}^*$, find $\langle y^2\rangle_{\boldsymbol{w}^*} = \boldsymbol{w}^{*\mathsf{T}}\underbrace{\mathbb{C}'\boldsymbol{w}^*}_{=\lambda_{\max}\boldsymbol{w}^*} = \lambda_{\max}\underbrace{\boldsymbol{w}^{*\mathsf{T}}\boldsymbol{w}^*}_{=1} = \lambda_{\max}$, where $\lambda_{\max}$ is the maximum eigenvalue of $\mathbb{C}'$. Since $\mathbb{C}'$ is symmetric, it has real eigenvalues λ_α and its eigenvectors $\boldsymbol{u}_\alpha$ are orthogonal, i.e. $\boldsymbol{u}_\alpha\cdot\boldsymbol{u}_\beta = \delta_{\alpha\beta}$, where $\delta_{\alpha\beta}$ is the kronecker delta:

$$\delta_{\alpha\beta} = \begin{cases} 1 \text{ if } \alpha = \beta, \\ 0 \text{ otherwise.} \end{cases}$$

Furthermore, all eigenvalues of $\mathbb{C}'$ non-negative, since

$$\lambda_\alpha = \boldsymbol{u}_\alpha^\mathsf{T}\mathbb{C}'\boldsymbol{u}_\alpha = \boldsymbol{u}_\alpha^\mathsf{T}\langle\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\rangle\boldsymbol{u}_\alpha = \langle|\boldsymbol{u}_\alpha^\mathsf{T}\boldsymbol{x}|^2\rangle \geq 0.$$

For any unit vector $\boldsymbol{w} = \sum_\alpha K_\alpha\boldsymbol{u}_\alpha$ that can be represented as a linear combination of the eigenvectors $\boldsymbol{u}_\alpha$ with coefficients K_α (assuming that $|\boldsymbol{w}|^2 = 1$), we find

$$\langle y^2\rangle_{\boldsymbol{w}} = \Big(\sum_\alpha K_\alpha\boldsymbol{u}_\alpha\Big)^\mathsf{T}\mathbb{C}'\Big(\sum_\beta K_\beta\boldsymbol{u}_\beta\Big) = \sum_{\alpha,\beta} K_\alpha K_\beta\lambda_\beta\underbrace{\boldsymbol{u}_\alpha^\mathsf{T}\boldsymbol{u}_\beta}_{=\delta_{\alpha\beta}} = \sum_\alpha (K_\alpha)^2\lambda_\alpha \leq \lambda_{\max}\sum_\alpha (K_\alpha)^2.$$

From $|\boldsymbol{w}|^2 = 1$, we find $\sum_\alpha (K_\alpha)^2 = 1$. Thus, $\langle y^2\rangle_{\boldsymbol{w}} \leq \lambda_{\max}\sum_\alpha (K_\alpha)^2 = \lambda_{\max}$ and $\langle y^2\rangle_{\boldsymbol{w}^*} = \lambda_{\max}$. This shows that $\langle y^2\rangle_{\boldsymbol{w}^*}$ is maximal in comparison to $\langle y^2\rangle_{\boldsymbol{w}}$ for any other $\boldsymbol{w}$ such that $|\boldsymbol{w}|^2 = 1$.

(b) Assume that $\boldsymbol{w}^*$ is a steady state:

$$0 = \langle\delta\boldsymbol{w}\rangle_{\boldsymbol{w}^*} = \eta\langle y(\boldsymbol{x} - y\,\boldsymbol{w})\rangle_{\boldsymbol{w}^*} = \eta(\mathbb{C}'\boldsymbol{w}^* - (\boldsymbol{w}^*\cdot\mathbb{C}'\boldsymbol{w}^*)\boldsymbol{w}^*).$$

For this equation to hold we must require that $\boldsymbol{w}^*$ is an eigenvector of $\mathbb{C}'$, and that $|\boldsymbol{w}^*|^2 = 1$. Then $\mathbb{C}'\boldsymbol{w}^* = \lambda\boldsymbol{w}^*$ and $\boldsymbol{w}^*\cdot\mathbb{C}'\boldsymbol{w}^* = \lambda$.

Now, we must show that $\boldsymbol{w}^*$ has the maximum eigenvalue, which is denoted by λ_1. First note that in order for the network to converge to a steady state, this steady state must be stable. Therefore, check the stability of $\boldsymbol{w}^*$. Evaluate $\langle \delta \boldsymbol{w} \rangle$ at $\boldsymbol{w} = \boldsymbol{w}^* + \boldsymbol{\epsilon}$, where $|\boldsymbol{\epsilon}|$ is small. To this end, we expand $\langle \delta \boldsymbol{\epsilon} \rangle$ in $\boldsymbol{\epsilon}$ to leading order:

$$\langle \delta \boldsymbol{\epsilon} \rangle \approx \eta \Big[\mathbb{C}' \boldsymbol{\epsilon} - 2(\boldsymbol{\epsilon} \cdot \mathbb{C}' \boldsymbol{w}^*) \boldsymbol{w}^* - (\boldsymbol{w}^* \cdot \mathbb{C}' \boldsymbol{w}^*) \boldsymbol{\epsilon} \Big].$$

We know that $\boldsymbol{w}^*$ must be an eigenvector of $\mathbb{C}$. So we choose a value of α, put $\boldsymbol{w}^* = \boldsymbol{u}_\alpha$ and multiply with $\boldsymbol{u}_\beta$ from the left, to determine the β-th component of $\langle \delta \boldsymbol{\epsilon} \rangle$:

$$\boldsymbol{u}_\beta \langle \delta \boldsymbol{\epsilon} \rangle \approx \eta \big[(\lambda_\beta - \lambda_\alpha) - 2\lambda_\alpha \delta_{\alpha\beta} \big] (\boldsymbol{u}_\beta \cdot \boldsymbol{\epsilon}).$$

Here we used that $\delta_{\alpha\beta}(\boldsymbol{u}_\alpha \cdot \boldsymbol{\epsilon}) = \delta_{\alpha\beta}(\boldsymbol{u}_\beta \cdot \boldsymbol{\epsilon})$. We conclude: if $\lambda_\beta > \lambda_\alpha$ then the component of displacement along $\boldsymbol{u}_\beta$ must grow, on average. So if λ_α is not the largest eigenvalue, the corresponding eigenvector $\boldsymbol{u}_\alpha$ cannot be a steady-state direction. The eigenvector corresponding to λ_1, on the other hand, represents a steady state. So only the choice $\lambda_\alpha = \lambda_1$ leads to a steady state.

(c) See Oja's M-rule in the lecture notes. Deduce normalisation from the steady-state condition $\langle \delta w_{ij} \rangle = 0$.

9.5 Kohonen net

The parameter σ regulates the width of the neighbouring function $\Lambda(i, i_0)$. This function is centred at the winning neuron i_0 ($\Lambda(i_0, i_0) = 1$), and it decreases as the distance from this neuron increases. The update rule assures that the weight assigned to the winning neuron i_0 is moved by a fraction η towards the fed pattern $\boldsymbol{x}$. Moreover, the neighbourhood function assures that the weights assigned to the remaining neurons are also moved towards $\boldsymbol{x}$ with a fraction determined by the width of $\Lambda(i, i_0)$: the fraction is larger the closer the neuron i is to the winning neuron i_0. In the limit of $\sigma \to 0$, the update rule reduces to the simple-competitive learning, because only the winning neuron for pattern $\boldsymbol{x}$ is updated in this case. In other words, the rule becomes:

$$\delta w_{ij} = \begin{cases} \eta(x_j - w_{ij}) \text{ for } i = i_0, \\ 0, \text{otherwise.} \end{cases}$$

M output neurons arranged in a d-dimensional lattice. The position of the i^{th} output neuron in the output space is a d-dimensional vector $\boldsymbol{r}_i$ (each $\boldsymbol{r}_i$ is fixed). Weight matrix $\mathbb{W}$ is an $M \times N$ matrix with elements w_{ij}, $i = 1, \ldots, M$, $j = 1, \ldots, N$. The weight vector $\boldsymbol{w}_i$ assigned to the output neuron i is N-dimensional (as are the input patterns $\boldsymbol{x}$): $\boldsymbol{w}_i = (w_{i1}, \ldots, w_{i,N})^{\mathsf{T}}$. Learning consists of two phases: ordering and convergence phase. Usually, ordering phase lasts for $T_{\text{order}} \approx 1000$ updates. The convergence phase starts after the ordering phase is finished. Initialisation: weights are initialised to random numbers from a relevant interval for a given problem, e.g. from $[-1, 1]$. Set the current update number t to $t = 0$. Learning algorithm:

1. Update $t \leftarrow t + 1$.
2. Draw a random pattern $\boldsymbol{x}$ from the set of input patterns.
3. Competition: find a neuron i_0 such that $\boldsymbol{w}_{i_0}$ is closest to $\boldsymbol{x}$ in the euclidean sense, i.e. $|\boldsymbol{x} - \boldsymbol{w}_{i_0}| \leq |\boldsymbol{x} - \boldsymbol{w}_i|$ for all i.
4. Update weight vectors:
$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \delta \boldsymbol{w}_i,$$
where
$$\delta w_{ij} = \eta(t)\Lambda(i, i_0, t)(x_j - w_{ij}), \text{ and } \Lambda(i, i_0, t) = \exp\left(-\frac{|\boldsymbol{r}_i - \boldsymbol{r}_{i_0}|^2}{2\sigma(t)^2}\right).$$
The dependence on the update number t is as follows: if $t < T_{\text{order}}$ then $\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau}\right)$ and $\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau}\right)$, where $\tau \approx \frac{T_{\text{order}}}{\log(\sigma_0)}$, σ_0 is set to the largest

distance in the output lattice, and η_0 is set to a small (but not too small) value, usually $\eta_0 = 0.1$. When η_0 is not too small, the network is likely to get rid of *kinks* during the ordering phase. If, however, $t > T_{\text{order}}$ (convergence phase, fine tunning), then both η and σ are kept constant and small (usually, $\eta = 0.01$ and $\sigma \approx 1$).

5. Repeat steps 1-4 until convergence.

9.10

See next page.

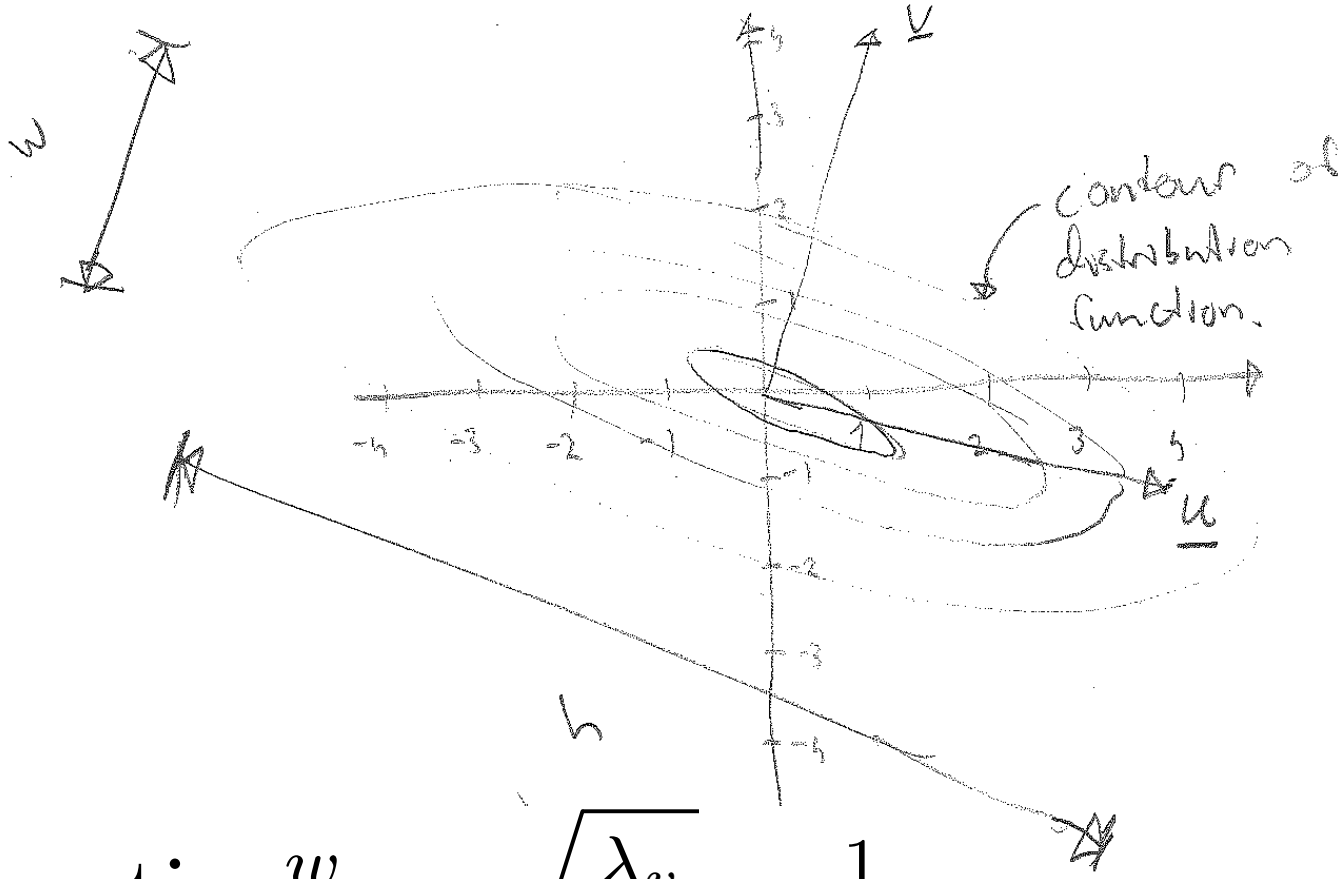
5a)

$$\underline{\underline{C}} = \lambda_u \frac{\underline{u}\underline{u}^T}{\underline{u}^T\underline{u}} + \lambda_v \frac{\underline{v}\underline{v}^T}{\underline{v}^T\underline{v}}$$

$$= 4 \frac{1}{17} \begin{bmatrix} 4 \\ -1 \end{bmatrix} \begin{bmatrix} 4 & -1 \end{bmatrix} + 1 \frac{1}{17} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 4 \end{bmatrix}$$

$$= \frac{4}{17} \begin{bmatrix} 16 & -4 \\ -4 & 1 \end{bmatrix} + \frac{1}{17} \begin{bmatrix} 1 & 4 \\ 4 & 16 \end{bmatrix}$$

$$= \frac{1}{17} \begin{bmatrix} 65 & 12 \\ 12 & 20 \end{bmatrix}$$



ratio $\frac{w}{h} = \sqrt{\frac{\lambda_v}{\lambda_u}} = \frac{1}{2}$

5b)

$$\begin{bmatrix} c_{11} & c_{21} \\ c_{21} & c_{22} \end{bmatrix} \cdot \left\langle (\underline{\xi} - \langle \underline{\xi} \rangle)(\underline{\xi} - \langle \underline{\xi} \rangle)^T \right\rangle$$

$$= \left\langle (\underline{\underline{L}}\,\underline{\xi} - \underline{\underline{L}}\underbrace{\langle \underline{\xi} \rangle}_{=0})(\underline{\underline{L}}\,\underline{\xi} - \underline{\underline{L}}\underbrace{\langle \underline{\xi} \rangle}_{=0})^T \right\rangle$$

$$= \left\langle \underline{\underline{L}}\,\underline{\xi}\,\underline{\xi}^T\,\underline{\underline{L}}^T \right\rangle$$

$$= \underline{\underline{L}} \left\langle \underline{\xi}\,\underline{\xi}^T \right\rangle \underline{\underline{L}}^T$$

$$= \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_{11} & L_{21} \\ 0 & L_{22} \end{bmatrix}$$

$$= \begin{bmatrix} L_{11}^2 & L_{11} L_{21} \\ L_{11} L_{21} & L_{21}^2 + L_{22}^2 \end{bmatrix}$$

Thus:

$$\begin{cases} C_{11} = L_{11}^2 & (1) \\ C_{21} = L_{11} L_{21} & (2) \\ C_{22} = L_{21}^2 + L_{22}^2 & (3) \end{cases}$$

2×1

$2 \times n \,/\, n \times 2$

$$(1) \Rightarrow L_{11} = \overset{+}{(-)} \sqrt{C_{11}}$$

$$(2) \Rightarrow L_{21} = \frac{C_{21}}{L_{11}} = \overset{+}{(-)} \frac{C_{21}}{\sqrt{C_{11}}}$$

$$(3) \Rightarrow L_{22}^2 = C_{22} - L_{21}^2$$

$$= C_{22} - \frac{C_{21}^2}{C_{11}}$$

$$\Rightarrow L_{22} = \overset{+}{(-)} \sqrt{C_{22} - \frac{C_{21}^2}{C_{11}}}$$

5 c)

$$\underline{\underline{C}}^{(\underline{\Sigma})} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(Σ_1 and Σ_2 are independent, with mean zero and variance 1.)

Chapter 10

10.3 Radial basis functions for XOR

(a) A problem is solvable by a simple perceptron if it is linearly separable. For the two-dimensional XOR problem, this means one must find a plane separating the two types of target outputs. Look at this figure:

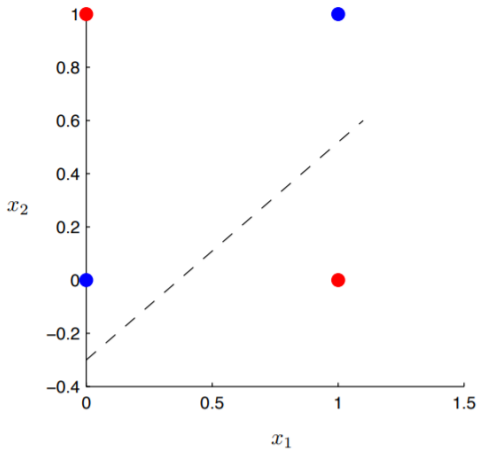


Figure 1: Question 21b: XOR problem. Input patterns (circles) in input space $(x_1,x_2)^\mathsf{T}$. Colours denote target output for a given input pattern: target output 0 is denoted by blue, and target output 1 is denoted by red. Dashed line is an example of a line that tries to solve the XOR problem. In this case, the input $(0,1)^\mathsf{T}$ would be misclassified as having the output 0.

From this figure, we see that no such plane can be found: there will always be at least one point that will be misclassified (the point on the "wrong side" of the plane). An example is shown by a dashed line in the figure: on the right side from the plane, there is only one pattern with the target output 1. On the opposite side of the plane there are two patterns with target output 0, but also one pattern with the target output 1. The latter pattern would be misclassified as the pattern with the output 0.

(b) Applying the radial-basis functions suggested in the task, we find that the input patterns have the following coordinates in the transformed space $(g_1, g_2)^\mathsf{T}$:

$$g_1(\boldsymbol{x}^{(1)}) = \exp(-|(1,1)^\mathsf{T} - (1,1)^\mathsf{T}|^2) = \exp(0) = 1$$

$$g_2(\boldsymbol{x}^{(1)}) = \exp(-|(1,1)^\mathsf{T} - (0,0)^\mathsf{T}|^2) = \exp(-2) \approx 0.14$$

$$g_1(\boldsymbol{x}^{(2)}) = \exp(-|(1,0)^\mathsf{T} - (1,1)^\mathsf{T}|^2) = \exp(-1) \approx 0.37$$

$$g_2(\boldsymbol{x}^{(2)}) = \exp(-|(1,0)^\mathsf{T} - (0,0)^\mathsf{T}|^2) = \exp(-1) \approx 0.37$$

$$g_1(\boldsymbol{x}^{(3)}) = \exp(-|(0,1)^\mathsf{T} - (1,1)^\mathsf{T}|^2) = \exp(-1) \approx 0.37$$
$$g_2(\boldsymbol{x}^{(3)}) = \exp(-|(0,1)^\mathsf{T} - (0,0)^\mathsf{T}|^2) = \exp(-1) \approx 0.37$$
$$g_1(\boldsymbol{x}^{(4)}) = \exp(-|(0,0)^\mathsf{T} - (1,1)^\mathsf{T}|^2) = \exp(-2) \approx 0.14$$
$$g_2(\boldsymbol{x}^{(4)}) = \exp(-|(0,0)^\mathsf{T} - (0,0)^\mathsf{T}|^2) = \exp(0) = 1$$

Note that both input patterns $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(3)}$ are transformed to the same vector in the transformed input space $(g_1, g_2)^\mathsf{T}$. Applying a simple perceptron to the transformed input patterns (in space $(g_1, g_2)^\mathsf{T}$), the problem is linearly separable. Indeed, there is a decision boundary separating the two classes of data: in this case, the network output at the decision boundary is equal to 0.5 (because the activation function is linear):

$$\boldsymbol{w}^\mathsf{T}\boldsymbol{g} - \theta = 0.5 \text{ for } \boldsymbol{g} \text{ at the decision boundary,}$$

where $\boldsymbol{w} = (w_1, w_2)^\mathsf{T}$. [Note: if the target outputs are +1 or -1, then we would require the sigmoid activation function, and that the network output at the boundary is equal to zero.] The weights and the threshold for the simple perception corresponding to the decision boundary shown in the following figure are $w_1 = w_2 = -1$ and $\theta = -1.5$:

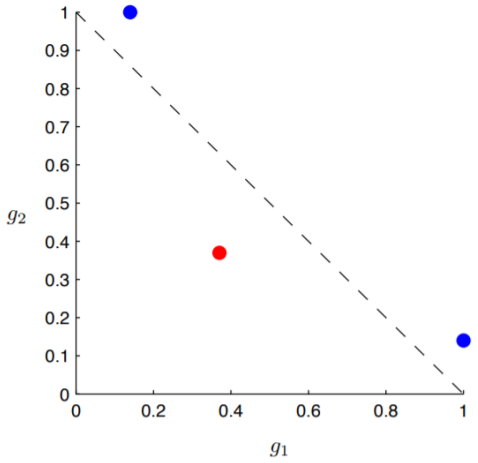


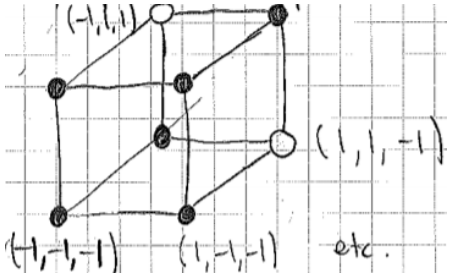
Figure 2: Question 21b: XOR problem in transformed space. Input patterns (circles) in space $(g_1,g_2)^\mathsf{T}$. Colours denote target output for a given input pattern: target output 0 is denoted by blue, and target output 1 is denoted by red. Dashed line is an example of a line solving the XOR problem (decision boundary). In this case, the weight vector and the threshold corresponding to the decision boundary shown are $\boldsymbol{w} = (-1,-1)^\mathsf{T}$ and $\theta = -1.5$.

Note that there are other possible solutions. Particularly, the weight vector $-\boldsymbol{w}$ and the corresponding threshold is also a possible solution, but one must check the network output for the problem given to decide which solution to take. In this task, one requires that the network outputs for the two blue points are smaller than 0.5,

whereas for the red point, the network output is required to be larger than 0.5.

10.3 Radial basis functions

(a) Is the problem linearly separable? Graphical representation:

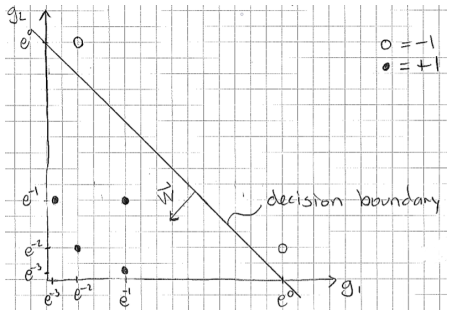


Clearly, no plane separates the white balls from the black balls, i.e. the problem isn't linearly separable.

(b) $\boldsymbol{w}_1 = (-1,1,1)^\mathsf{T}$, $\boldsymbol{w}_2 = (1,1,-1)^\mathsf{T}$, $g_i^{(\mu)} = \exp(-\frac{1}{4}|\boldsymbol{x}^{(\mu)} - \boldsymbol{w}_i|^2)$ for $i = 1,2$.

μ	$\lvert\boldsymbol{x}^{(\mu)} - \boldsymbol{w}_1\rvert^2$	$\lvert\boldsymbol{x}^{(\mu)} - \boldsymbol{w}_2\rvert^2$	$g_1^{(\mu)}$	$g_2^{(\mu)}$	$t^{(\mu)}$
1	8	8	$\exp(-2)$	$\exp(-2)$	1
2	4	12	$\exp(-1)$	$\exp(-3)$	1
3	4	4	$\exp(-1)$	$\exp(-1)$	1
4	0	8	$\exp(0)$	$\exp(-2)$	-1
5	12	4	$\exp(-3)$	$\exp(-1)$	1
6	8	8	$\exp(-2)$	$\exp(-2)$	1
7	8	0	$\exp(-2)$	$\exp(0)$	-1
8	4	4	$\exp(-1)$	$\exp(-1)$	1

$\exp(0) = 1$, $\exp(-1) \approx 0.37$, $\exp(-2) \approx 0.14$, $\exp(-3) \approx 0.05$. Graphical presentation in the $(g_1, g_2)^\mathsf{T}$ space:



(c) Weight vector $\boldsymbol{W}$ and threshold Θ: $W_1 g_1 + W_2 g_2 - \Theta = 0$ on boundary because

sgn() activation function. $\boldsymbol{W} = (-1,-1)^{\mathsf{T}}$, $g_1 = 1$, $g_2 = 0 \implies \Theta = -1$.

Answer: $\boldsymbol{W} = (-1,-1)^{\mathsf{T}}$, $\Theta = -1$.

Chapter 11