Chapter 7: Supervised Hebbian Learning

Brandon Morgan

1/13/2021

E7.4

As stated in the answer to E7.1, the *hardlim* transition function could be used instead of the *hardlims*, where a binary response would be used instead bipolar. From example E7.1, the input vectors, p_i, will be inputted by column, where an empty tile is represented as a 0 and a filled tile as 1. Thus, $p_1^t = [0, 0, 1, 1]$ and $p_2^t = [1, 1, 0, 1]$; thus, $P = [p_1, p_2]$:

$$P = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

1

Two vectors are othorgonal if their dot product is zero, $p_1^t p_2 = 0$:

```r
p1 = matrix(c(0, 0, 1, 1), ncol=1)
p2 = matrix(c(1, 1, 0, 1), ncol=1)
t(p1)%*%p2
```

```
##      [,1]
## [1,]    1
```

As we can see, the dot product is not zero; therefore our vectors are not orthogonal.

2

The Hebb Rule is given by $W = TP^t$, where $T = P$ for autoassociator network, to get $W = PP^t$. However, from P7.7, when *hardlim* is used for binary transtion, we use the following rule instead: $W'p' + b = Wp$, where $W' = 2W$ and $b = -W1$, where 1 is a vector ones 1's.

Here we get the following normal weights:

```r
P = matrix(c(p1, p2), ncol=2)
W=P%*%t(P)
W
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    0    1
## [2,]    1    1    0    1
## [3,]    0    0    1    1
## [4,]    1    1    1    2
```

Now we can calculate W':

```
W_dash = 2*W
W_dash
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    2    0    2
## [2,]    2    2    0    2
## [3,]    0    0    2    2
## [4,]    2    2    2    4
```

and bias $b = -W1$

```
ones = matrix(1, ncol=1, nrow=4)
b = -W%*%ones
b
```

```
##      [,1]
## [1,]   -3
## [2,]   -3
## [3,]   -2
## [4,]   -5
```

3

We can convert our new input into the following vector: $p_t^t = [1, 1, 1, 1]$. Now we can test our new input pattern by $a = hardlim(W'p_t + b)$:

```
pt = matrix(c(1, 1, 1, 1), ncol=1)
W_dash%*%pt+b
```

```
##      [,1]
## [1,]    3
## [2,]    3
## [3,]    2
## [4,]    5
```

The hardlim function states that every entry of $a = (Wp_t)_i < 0$ is assigned to 0 and $a = (Wp_t)_i \geq 0$ is assigned to 1. Thus, we will get the following output vectors $a = [1, 1, 1, 1]$:

```
matrix(c(1,1, 1, 1), ncol=1)
```

```
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
```

This output does not match any input as the input vectors were not orthogonal.