DAT157 Obligatory assignment 1

145157 Pål Vårdal Gjerde
145165 Martin Fure

Exercise 1

1) When asked about the letters with no noise, both paradigms are able to recognise the letters perfectly. With noise introduced, performance of both networks drops sharply; the adaline network gets correct results somewhat more often, but still not often enough to be useful.
2) The performance of the MLP network is significantly better; as with the two former, it is able to recognise the letters when presented without noise, but even with 20% noise, it recognises the presented letter most of the time.

Exercise 2

First, two adalines were trained to recognise OR-logic, where one was trained on (P or Q) and the other on (not P or not Q). Then a third was trained to recognise AND-logic. Finally, to use this madaline, the two inputs are sent through both the first adalines, and their outputs are sent through the third. This correctly returns the XOR output of the two.

Exercise 3

Each three letter codon is converted into a twelve binary digit input pattern, where each individual letter is converted to four binary digits as specified in the assignment, and these three four-digit series are combined. The output layer has a size of 21: 20 different amino acids, plus the STOP code. The network is then trained on each possible codon and its corresponding amino acid/STOP code. Finally, it is then tested on the same set of codons.

Most of the outputs have the correct answer, with the output value for that answer being larger than 0.9. Occasionally, no particular output in the answer has a high value; in the code, this results in the highest (but still low) output being printed as a mistake, but this should probably be recognised as the network not having an answer.

Exercise 4

Each network is trained with each of the three different sets of letters, learning which shape corresponds to which letter. The intention is that the networks learn to recognise a letter's shape even when the letter looks slightly different (e.g. is from a different font). Then, each letter from each font is tested.

The perceptron network recognises most letters, but there's a noticeable amount of errors. It will occasionally return several possibilities for an input pattern, or none at all. In contrast, the MLP network has not yet been observed to make incorrect guesses.

Exercise 5

Input:

-1	-1	-1	-1	-1	1
-1	1	1	-1	-1	-1
1	-1	-1	1	-1	1

Assuming the nodes are named A, B, C, D, E, F, this causes the following changes in the edges' weights (+ and - mean +1 and -1, respectively):

	Input 1	Input 2	Input 3	Weight
AB	+	-	-	-1
AC	+	-	-	-1
AD	+	+	+	3
AE	+	+	-	1
AF	-	+	+	1
BC	+	+	+	3
BD	+	-	-	-1
BE	+	-	+	1
BF	-	-	-	-3
CD	+	-	-	-1
CE	+	-	+	1
CF	-	-	-	-3
DE	+	+	-	1
DF	-	+	+	1
EF	-	+	-	-1

Exercise 6

Note: For most of the assignment, JavANN 1.5 has been used. For this part, version 1.4 is necessary.

a) In order to get any meaningful output at all from the Hopfield network, larger cell patterns were necessary. These can be seen in the file named «oppg6cancers.txt», and are loaded using the same reader class as exercise 4 used for its fonts. Of the 4 cells provided, the Hopfield was only able to conclusively recall 1 (the third one); the recollection of the other 3 were very similar to each other, albeit not identical.
b) The BAM network was able to recall each of the four cancer cells as well as their associated output patterns.
c) In this case, cell 3 was chosen to test with because it's the one Hopfield recognised clearly. Sadly, even as little as 5% noise seems to utterly confuse poor Hopfield. BAM, on the other hand, tolerates a greater degree of noise and still returns the correct cells.

Screen dumps from running the code

Exercise 1

```
Results from perceptron network:
Test of perfect T: T, N
Test of perfect O: T, N
Test of perfect N: T, N
Test of T with 10% predetermined noise: T, N
Test of T with 20% predetermined noise: T, N
Test of T with 10% random noise: T, N
Test of T with 20% random noise: T, N

Results from adaline network:
Test of perfect T: T
Test of perfect O: T
Test of perfect N: T
Test of T with 10% predetermined noise: T
Test of T with 20% predetermined noise: T
Test of T with 10% random noise: T
Test of T with 20% random noise: T

Results from MLP network:
Test of perfect T: T (53,22%), O (2,27%), N (14,51%)
Test of perfect O: T (53,22%), O (2,27%), N (14,51%)
Test of perfect N: T (53,22%), O (2,27%), N (14,51%)
Test of T with 10% predetermined noise: T (53,22%), O (2,27%), N (14,51%)
Test of T with 20% predetermined noise: T (53,22%), O (2,27%), N (14,51%)
Test of T with 10% random noise: T (53,22%), O (2,27%), N (14,51%)
Test of T with 20% random noise: T (53,22%), O (2,27%), N (14,51%)
```

Exercise 2

```
True  XOR True:  false
True  XOR False: true
False XOR True:  true
False XOR False: false
```

Exercise 3

```
Best match for UGU: Cysteine, with confidence 94%
Best match for UCU: Serine, with confidence 96%
Best match for AUA: Isoleucine, with confidence 94%
Best match for CUA: Leucine, with confidence 99%
Best match for AUC: Isoleucine, with confidence 95%
Best match for CUC: Leucine, with confidence 95%
Best match for GUA: Valine, with confidence 96%
Best match for AUG: Methionine, with confidence 85%
Best match for CUG: Leucine, with confidence 99%
Best match for GUC: Valine, with confidence 94%
Best match for GUG: Valine, with confidence 95%
Best match for UUA: Leucine, with confidence 93%
Best match for AUU: Isoleucine, with confidence 95%
Best match for AAA: Lysine, with confidence 93%
Best match for UUC: Phenylalanine, with confidence 92%
Best match for CUU: Leucine, with confidence 95%
Best match for CAA: Glutamine, with confidence 95%
Best match for AAC: Asparagine, with confidence 93%
Best match for CAC: Histidine, with confidence 95%
Best match for UUG: Leucine, with confidence 95%
Best match for GUU: Valine, with confidence 95%
Best match for GAA: Glutamic acid, with confidence 91%
Best match for AAG: Lysine, with confidence 92%
Best match for CAG: Glutamine, with confidence 94%
Best match for GAC: Aspartic acid, with confidence 91%
```

```
Best match for GAG: Glutamic acid, with confidence 95%
Best match for UUU: Phenylalanine, with confidence 93%
Best match for UAA: STOP, with confidence 98%
Best match for AAU: Asparagine, with confidence 93%
Best match for UAC: Tyrosine, with confidence 93%
Best match for CAU: Histidine, with confidence 95%
Best match for UAG: STOP, with confidence 93%
Best match for GAU: Aspartic acid, with confidence 91%
Best match for UAU: Tyrosine, with confidence 93%
Best match for AGA: Arginine, with confidence 95%
Best match for CGA: Arginine, with confidence 97%
Best match for AGC: Arginine, with confidence 4% (correct answer: Serine)
Best match for CGC: Arginine, with confidence 95%
Best match for ACA: Threonine, with confidence 96%
Best match for CCA: Proline, with confidence 94%
Best match for GGA: Glycine, with confidence 95%
Best match for ACC: Threonine, with confidence 96%
Best match for AGG: Arginine, with confidence 95%
Best match for CCC: Proline, with confidence 95%
Best match for CGG: Arginine, with confidence 98%
Best match for GGC: Glycine, with confidence 96%
Best match for GCA: Alanine, with confidence 94%
Best match for ACG: Threonine, with confidence 94%
Best match for CCG: Proline, with confidence 97%
Best match for GCC: Alanine, with confidence 96%
Best match for GGG: Glycine, with confidence 98%
Best match for GCG: Alanine, with confidence 97%
Best match for UGA: STOP, with confidence 93%
Best match for AGU: Arginine, with confidence 4% (correct answer: Serine)
Best match for UGC: Cysteine, with confidence 94%
Best match for CGU: Arginine, with confidence 95%
Best match for UCA: Serine, with confidence 97%
Best match for ACU: Threonine, with confidence 96%
Best match for UCC: Serine, with confidence 96%
Best match for UGG: Tryptophan, with confidence 44%
Best match for CCU: Proline, with confidence 95%
Best match for GGU: Glycine, with confidence 96%
Best match for UCG: Serine, with confidence 95%
Best match for GCU: Alanine, with confidence 96%
```

Exercise 4

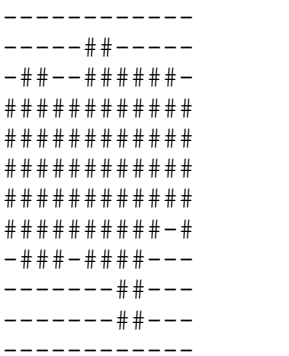
```
Testing a perceptron network:
Font 1
Testing with A: A
Testing with B: B
Testing with C: C
Testing with D: D
Testing with E: B, E
Testing with J: D, J
Testing with K: K

Font 2
Testing with A: A
Testing with B: B
Testing with C: C
Testing with D: D, K
Testing with E: E, K
Testing with J: J
Testing with K: K

Font 3
Testing with A: A
Testing with B: B
```

```
Testing with C: C
Testing with D: D
Testing with E: E
Testing with J: J
Testing with K: K

Testing an MLP network:
Font 1
Testing with A: A
Testing with B: B
Testing with C: C
Testing with D: D
Testing with E: E
Testing with J: J
Testing with K: K

Font 2
Testing with A: A
Testing with B: B
Testing with C: C
Testing with D: D
Testing with E: E
Testing with J: J
Testing with K: K

Font 3
Testing with A: A
Testing with B: B
Testing with C: C
Testing with D: D
Testing with E: E
Testing with J: J
Testing with K: K
```

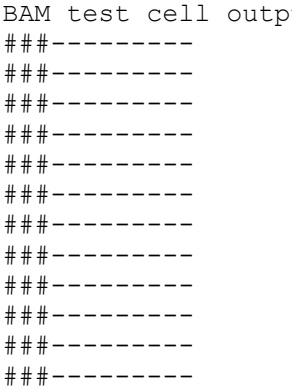
Exercise 6

```
Test cell input 1:
------------
-----##-----
-##--######-
############
############
############
############
##########-#
-###-####---
-------##---
-------##---
------------

Hopfield test cell output 1:
--###-----##
#---###--###
##--###-###-
############
############
############
-###########
##########-#
###########-
#-##---#--#-
---#-###--##
-----##---##
```

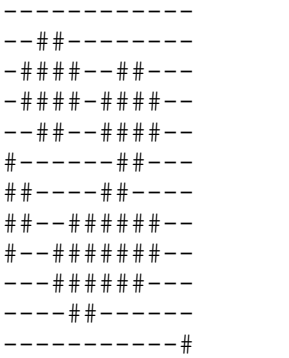
BAM test cell output 1 (in):



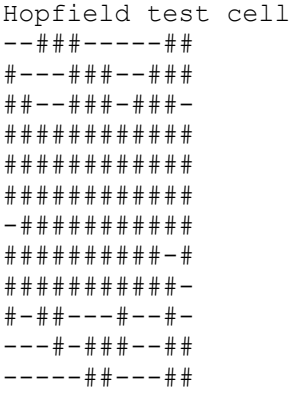
BAM test cell output 1 (out):



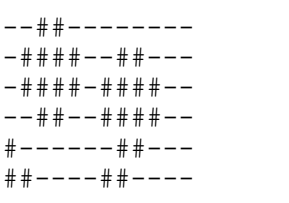
Test cell input 2:



Hopfield test cell output 2:



BAM test cell output 2 (in):



```
##--######--
#--#######--
---######---
----##------
-----------#
```

BAM test cell output 2 (out):

```
---###------
---###------
---###------
---###------
---###------
---###------
---###------
---###------
---###------
---###------
---###------
---###------
```

Test cell input 3:

```
------------
--##---##---
--##---##---
---##-------
---##-------
--------##--
#-------##--
#-##--------
--##--##----
------####--
-##-----##--
-##---------
```

Hopfield test cell output 3:

```
------------
--##---##---
--##---##---
---##-------
---##-------
--------##--
#-------##--
#-##--------
--##--##----
------####--
-##-----##--
-##---------
```

BAM test cell output 3 (in):

```
------------
--##---##---
--##---##---
---##-------
---##-------
--------##--
#-------##--
#-##--------
--##--##----
------####--
-##-----##--
-##---------
```

BAM test cell output 3 (out):

```
------###---
```

```
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
```

Test cell input 4:

```
--###-----##
#--###--####
##--##--###-
##########--
##########
--##########
--##########
##########--
##########-
#-##--##-##-
--##-##--###
-##--##---##
```

Hopfield test cell output 4:

```
--###-----##
#--###--####
##--##--###-
##########--
##########
--##########
--##########
##########--
##########-
#-##--##-##-
--##-##--###
-##--##---##
```

BAM test cell output 4 (in):

```
--###-----##
#--###--####
##--##--###-
##########--
##########
--##########
--##########
##########--
##########-
#-##--##-##-
--##-##--###
-##--##---##
```

BAM test cell output 4 (out):

```
---------###
---------###
---------###
---------###
---------###
---------###
---------###
---------###
---------###
```

```
---------###
---------###
---------###

Test with 20% noise
Input pattern:
-#----------
---##--##-##
--##---##---
---###--#---
---###----#-
--#-#---##--
#-------##-#
-###----#---
-###--##--##
-##---#-#--#
-##-----##--
-##-#--#----

Hopfield output:
--###-----##
#--###--####
##--##--###-
##########--
###########
--##########
--##########
##########--
##########-
#-##--##-##-
--##-##--###
-##--##---##

BAM output (in):
------------
--##---##---
--##---##---
---##-------
---##-------
--------##--
#-------##--
#-##--------
--##--##----
------####--
-##-----##--
-##---------

BAM output (out):
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
------###---
```