

KAPITEL 6

Design og test af neuralt netværk til optisk karaktergenkendelse (NN3 X)

KAPITEL 6

6 DESIGN OG TEST AF NEURALT NETVÆRK TIL OPTISK KARAKTERGENKENDELSE (NN3_X)

6.1 Indledning

I denne opgave skal det undersøges, om et neuralt netværk kan "lære sig" de ti karakterer: 0 1 2 3 4 5 6 7 8 9 ud fra billeder af disse. Et sådant netværk kaldes for et optisk karakter genkendelses netværk (Optical Character Recognition OCR) og trænes ved hjælp af billed-input og numeriske output-data.

Til opgaven vælges:

Netværkstype:	Feedforward
Neurontype:	Sigmoid-overføringsfunktion (hyperbolsk tangens-overføringsfunktion) med forstærkning 1.
Netværksarkitektur:	Et input-lag, 2 skjulte lag og 1 output-lag
Antal input-neuroner:	$8 \cdot 8 = 64$
Antal output-neuroner:	10
Antal neuroner i skjulte lag ved start:	25 og 20

6.2 Afgrænsninger

Der opereres kun med de 10 karakterer fra titalssystemet, repræsenteret i et $8 \cdot 8$ matrix format.

6.3 Datarepræsentation

Følgende format bruges til generering af træningsdata:

facts

```
.....  
..XXXXX.  
.X....XX  
.X...X.X  
.X..X..X  
.X.X...X  
.XX....X  
..XXXXX.
```

0

```
.....  
....X...  
...XX...  
....X...  
....X...  
....X...  
....X...  
....X...  
..XXXXX.
```

1

```
.....  
..XXXXX.  
.X....X  
.....X  
...XXX.  
...X....  
..X....X  
.XXXXXXXX
```

2

```
.....  
..XXXXX.  
.X....X  
.....X  
...XXX.  
.....X  
.X....X  
..XXXXX.
```

3

```
.....  
....XXX.  
...X..X.  
..X...X.  
.X....X.  
.XXXXXXXX  
.....X.  
.....X.
```

4

```
.....  
.XXXXXXXX  
.X.....  
.X.....  
.XXXXXX.  
.....X  
.X.....X  
..XXXXX.
```

5

```
.....  
..XXXXX.  
.X.....  
.X.....  
.XXXXXX.  
.X.....X  
.X.....X  
..XXXXX.
```

6

```
.....  
.XXXXXXXX  
.....X  
.....X.  
.....X.  
....X...  
...X....  
...X....
```

7

```
.....  
..XXXXX.  
.X.....X  
.X.....X  
..XXXXX.  
.X.....X  
.X.....X  
..XXXXX.
```

8

```
.....  
..XXXXX.  
.X.....X  
.X.....X  
..XXXXXX  
.....X  
.....X  
..XXXXX.  
9
```

6.4 Bedømmelseskriterier

Der vil i bedømmelsen selvsagt blive lagt vægt på, at netværket kan genkende de påtrykte input-data (cifre). Der stilles dog ingen procentklassifikation i denne opgave, da det også ønskes undersøgt, hvordan netværket reagerer på forskellige udformninger af de præsenterede cifre.

6.5 Opsætning og træning af netværk

Til løsning af opgaven er valgt et netværk (NN3_1) med to skjulte lag med 25 neuroner i det første og 20 neuroner i det andet lag. Ved adskillige forsøg har det vist sig, at det bedste antal skjulte neuroner er ganske svært at fastlægge. Der blev gjort forsøg med følgende netværkskonfigurationer, hvori antallet af skjulte neuroner og skjulte lag blev varieret:

1. Ét skjult lag med 10 neuroner.
2. Ét skjult lag med 15 neuroner.
3. To skjulte lag med henholdsvis 15 og 10 neuroner.
4. Ét skjult lag med 25 neuroner.
5. To skjulte lag med henholdsvis 25 og 20 neuroner.

I første forsøg anvendtes for få neuroner, idet de påtrykte data ikke blev genkendt særligt godt. Øvrige forsøg viste bedre genkendelsesevne end forsøg 1 og afveg kun lidt fra hinanden. Sidste forsøg valgtes som det endelige netværk. Alle fem netværk blev helt færdige med at træne, før de blev taget i brug.

Træningstolerancen sættes til 0.1

og ved :	% rigtige svar	anvendes Indlæringsraten
	0 %	1.0
	50 %	1.0
	75 %	0.9
	90 %	0.8

Det vælges, at teste og lagre netværket for hvert femte gennemløb.

Netværkstræningen er udført med 10 % støj på input-data.

Netværket er også trænet med tilsløring (blurring), hvilket ligeledes skulle give netværket større generaliseringsevne. Nettet får herved præsenteret hvert sæt trænings-data to gange: Én gang normalt og én gang tilsløret. Træningen ophører, når alle trænings-data - både tilslørede og utilslørede - er nået under den angivne trænings-tolerance.

Efter 95 iterationer fremkommer de bedste testdata i træningsforløbet, hvor både tilslørede (10 stk.) og utilslørede (10 stk.) input-data giver rigtige outputs:

GOOD = 20 (10 stk. + 10 stk.)

BAD = 0

AvgError = 0.0237

RMSError = 0.0354

R^2 : 0 = 0.9958

R^2 : 1 = 0.9946

R^2 : 2 = 0.9938

R^2 : 3 = 0.9949

R^2 : 4 = 0.9994

R^2 : 5 = 0.9937

R^2 : 6 = 0.9867

R^2 : 7 = 0.9975

R^2 : 8 = 0.9851

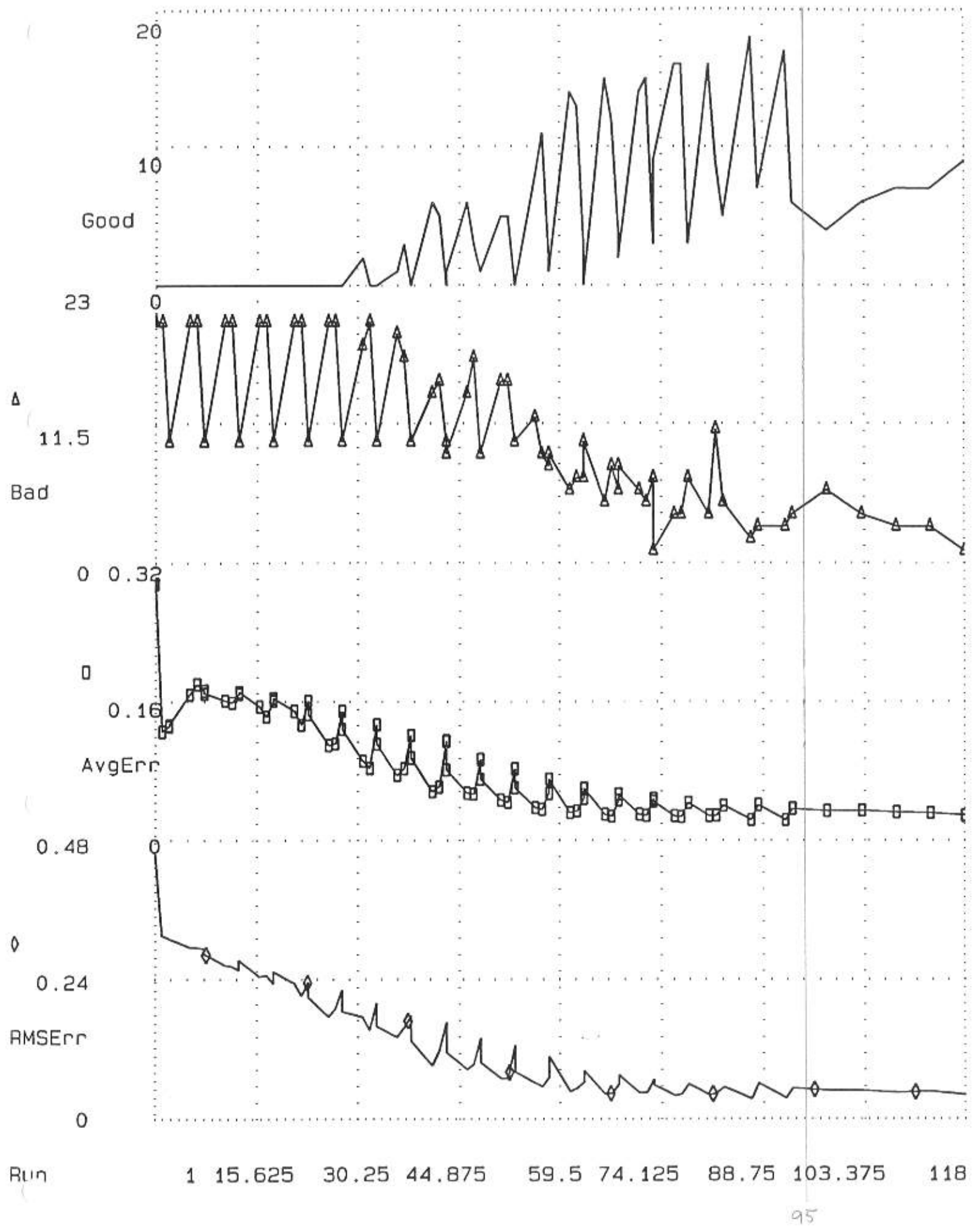
R^2 : 9 = 0.9984

Bad 0 til Bad 9 = 0

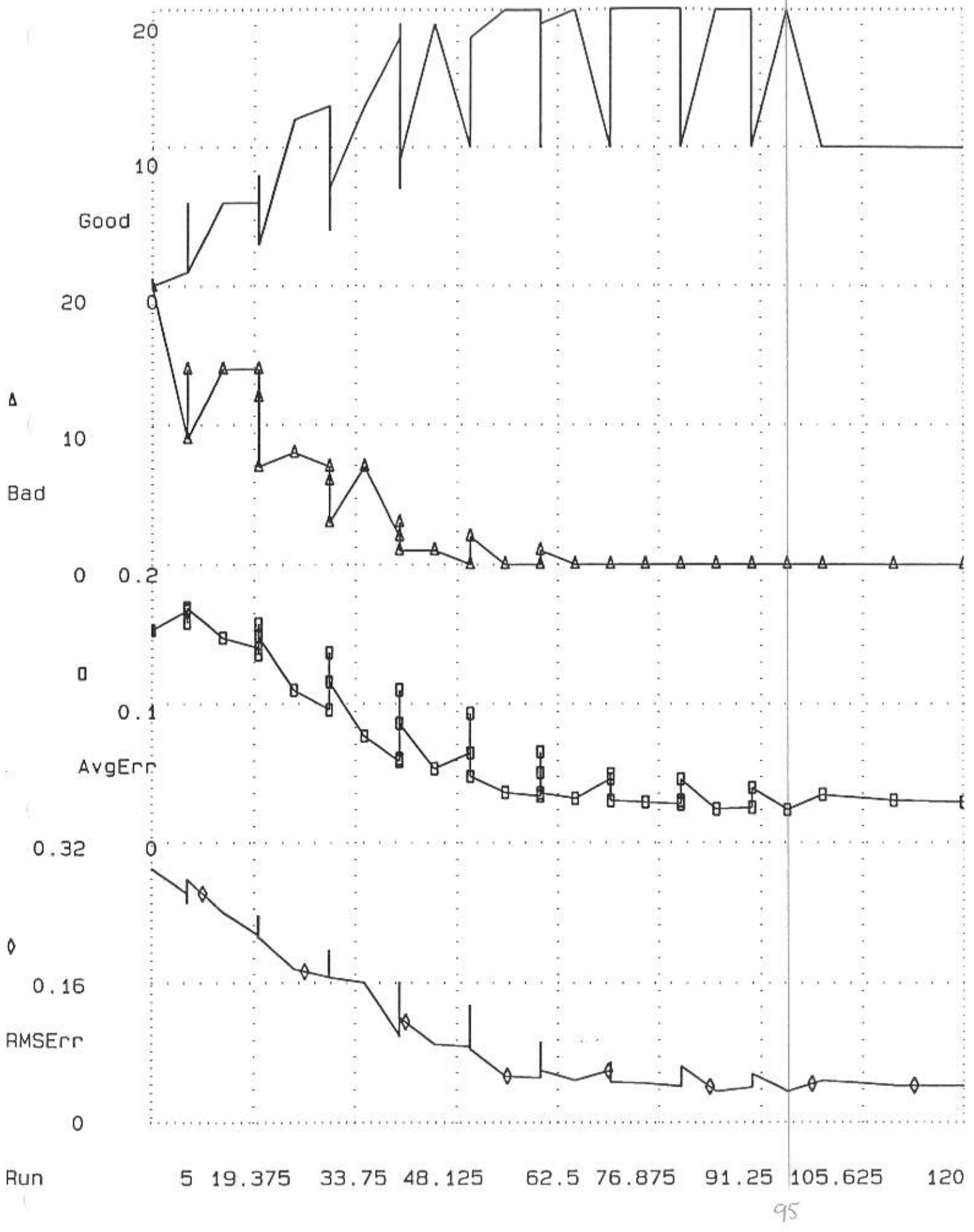
Netværket findes på medfølgende diskette i directory A:\NN3_1

Se trænings- og test-grafer på følgende sider.

Graf 6.1 viser træningsresultater og graf 6.2 viser testresultater.



Graf 6.1 Netdata ved træning af NN3_1 (NN3_1.STS)



Graf 6.2 Netdata ved test under træning NN3_1 (NN3_1.STA)

6.6 Netværksafprøvning

Nedenfor vises resultatet af en kørsel med netværket NN3_1, trænet som omtalt under pkt. 6.5.

Først vises det tals pixel-billede, som påtrykkes netværket, og derunder netværkets output i form af decimal-brøker, der indikerer netværkets "gæt". I eksempel 1 er 7 således nettets bedste gæt, da der står den største decimal-brøk ud for dette tal.

```

_____ 1
.....
  [Pixel image of the digit 7]
  [Pixel image of the digit 7]
.....
.....
.0724 0
.0014 1
.1714 2
.0102 3
.0076 4
.0004 5
.0006 6
.6934 7 <---- OK
.0050 8
.0002 9
    
```

----- 2



- .0040 0
- .0020 1
- .0484 2
- .0002 3
- .0014 4
- .0062 5
- .1796 6 <----- Ikke OK
- .0032 7 <----- Det ønskede
- .1432 8
- .0012 9

----- 3



- .0566 0
- .0126 1
- .1136 2
- .0692 3
- .0064 4
- .0008 5
- .0002 6
- .8816 7 <----- OK
- .0004 8
- .0006 9

----- 4

.
.
.
.
.
· [REDACTED] ·
.
.

.0232 0
.0030 1
.0466 2
.0002 3
.7410 4
.0012 5
.0506 6
.0312 7
.0064 8
.0006 9

<---- Netværket tror at strengen er et 4-tal.

----- 5

.
.
.
.
.
■
.
.

.0074 0
.0012 1
.0784 2
.0002 3
.0034 4
.0010 5
.1264 6
.0086 7
.0696 8
.0002 9

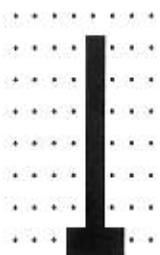
----- 6



.0614 0
.0054 1
.0996 2
.0342 3
.0062 4
.0008 5
.0002 6
.9104 7
.0006 8
.0004 9

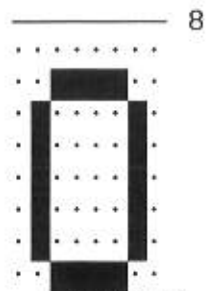
<----- OK

----- 7

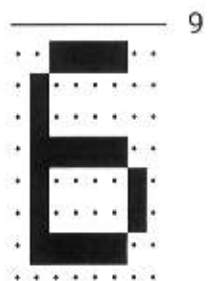


.0004 0
.8306 1
.0636 2
.0104 3
.0012 4
.0434 5
.0082 6
.0376 7
.0034 8
.0020 9

<----- OK



- .0810 0 <----- Det ønskede
- .0014 1
- .0184 2
- .0004 3
- .0316 4
- .0008 5
- .1624 6 <----- Ikke OK
- .0416 7
- .0562 8
- .0002 9



- .0004 0
- .0076 1
- .0130 2
- .0002 3
- .0050 4
- .1344 5 <----- Ikke OK
- .0820 6 <----- Det ønskede
- .0024 7
- .0636 8
- .0004 9

10

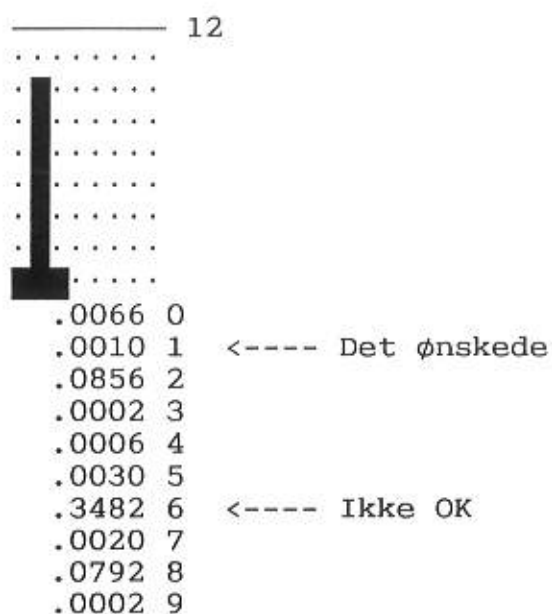


.0004	0	
.0022	1	
.0036	2	
.0002	3	
.0406	4	
.1586	5	<----- Ikke OK
.1142	6	
.0020	7	
.0352	8	
.1550	9	<----- Det ønskede

11



.0002	0	
.1634	1	
.0274	2	
.0010	3	
.0004	4	
.4100	5	<----- OK
.0292	6	
.0004	7	
.0992	8	
.0076	9	



6.7 Konklusion

Det er tydeligvis ikke ligegyldigt, hvor man placerer bogstaverne i 8·8 matrix-feltet. Af eksempel 4 fremgår det, at netværket tror "stregen" i matrix'en er et 4-tal. 4-tallet, som nettet har lært sig, er den eneste karakter med markeringer (X'er) her. Det vil sige, at netværket associerer bestemte tal til bestemte pixels i matrix'en, hvorfor input-tallene ikke kan anbringes vilkårligt i 8·8 feltet. Ønskes dette, skal der opereres med flere forskellige placeringer af hvert enkelt tal, hvilket naturligvis gør træningen mere kompliceret. Hvis netværket var blevet præsenteret for flere data under træningen, ville risikoen være mindre for, at netværket ikke blot lærer de påtrykte data udenad, men opnår en bedre evne til at generalisere.