

## KAPITEL 4

Design og test af neuralt netværk til løsning af andengrads-  
ligninger med reelle rødder (NN1 X)

---

# KAPITEL 4

#### 4. DESIGN OG TEST AF NEURALT NETVÆRK TIL LØSNING AF ANDENGRADSLIGNINGER MED REELLE RØDDER (NN1\_X)

##### 4.1 Indledning

Dette kapitel omhandler opsætning, træning og test af et neuralt netværk til løsning af andengradsligninger med reelle rødder. Der vil ikke blive beskrevet hvordan de forskellige parametres opsætning i programpakken BrainMaker Professionel (v2.5) skal udføres, da dette blot vil blive en gentagelse af manualen.

Med en ligning på formen:

$$A \cdot x^2 + B \cdot x + C = 0$$

skal netværket lære sig at bestemme de to reelle rødder heri,  $X_1$  og  $X_2$ , efter at være blevet trænet med et givet antal sammenhørende A-, B-, C- og  $X_1$ -,  $X_2$ -værdier.

Opsætningen af neurale netværk omfatter tre elementer, som hver især indeholder et antal parametre:

1. Netværk
  - a) Valg af netværkstype
  - b) Valg af neuron-type
  - c) Valg af netværksarkitektur
  
2. Data
  - a) Valg af data-repræsentation
  
3. Træning
  - a) Valg af træningsalgoritme
  - b) Valg af træningsparametre
  - c) Valg af træningsdata

Til opgaven er valgt et feedforward-netværk af neuroner med en sigmoid-overføringsfunktion (hyperbolsk tangens overføringsfunktion) med forstærkningen 1.

Netværkets arkitektur, det vil sige dets lagopdeling og neuronbestykning fastlægges ved en skønsom kombination af teori, håndregler og den erfaring, man uvægerligt erhverver sig ved at måtte prøve sig frem.

Netværket har tre neuroner i input-laget (én for hver af ligningskoefficienterne A, B og C) og to neuroner i output-laget (én for hver af rødderne X1 og X2). Indledningsvis vælges to skjulte lag med tre neuroner i hvert efter tommelfingerreglen om at bestyke skjulte lag med et antal neuroner svarende til gennemsnittet af input- og output-neuroner.

Valg af data-repræsentation er i dette tilfælde ikke det store problem, da både input- og output-data har numerisk form.

Træningssalgoritmen vælges den tidligere omtalte "Backpropagation of Error".

Træningsdata i nødvendig mængde genereres ved hjælp af et C-program: ANDENG.EXE.

Nu er neurale net ikke kendt for nøjagtighed, når det gælder numeriske bestemmelser. Neurale netværk er langt bedre til at vise resultater indeholdende tendenser eller "retninger".

På trods af ovenstående gøres forsøget for at undersøge påstanden nærmere.

## 4.2 Afgrænsninger

Hvis netværket skal kunne løse opgaven med et overkommeligt antal trænings-data er det nødvendigt med nogle afgrænsninger. Disse afgrænsninger består i at holde de tilfældigt udvalgte værdier for A-, B- og C-koefficienter inden for visse intervaller, således:

$$0 < A < 10, \quad -10 < B < 10, \quad -10 < C < 10$$

## 4.3 Datarepræsentation

I bilag 1 er vist koden ANDENG.C til generering af andengrads-ligninger med reelle rødder. Data er lagret i en matrix med 150 rækker og 5 søjler, hvoraf de tre første søjler indeholder ligningernes A-, B- og C-koefficienter, og de to sidste søjler rummer ligningsrødderne:

A	B	C	X1	X2
2	-5	1	2.281	0.219
4	3	-8	1.088	-1.84
.	.	.	.	.
.	.	.	.	.

etc.

Filen: NN1\_1.DAT rummer denne matrix.

## 4.4 Bedømmelseskriterier

Som før nævnt er neurale net ikke kendt for nøjagtighed, når det gælder numerisk præcision. Derfor er det ikke muligt at opstille egentlige kriterier til netværksafprøvningen, men det forventes

at netværket finder løsninger, som er i "nærheden" af ligningsrødderne. I punkt 4.6 **Netværksafprøvning** er de tre ligningskoefficienter A, B og C samt det matematisk beregnede resultat og netværkets løsning tilgængelige efter kørsel af et trænet netværk med nye datasæt.

#### 4.5 Opsætning og træning af netværk

Netværket kaldes til løsning af andengradsligninger med reelle rødder NN1\_4.

Som nævnt lægges ud med to skjulte lag à tre neuroner. Der tilføjes én neuron i hvert af de skjulte lag hver gang Root Mean Squared (RMS) Error ikke er blevet forbedret med 0.1 eller mere over 200 gennemløb.

Programmet BrainMaker har to parametre med stor indflydelse på træning af netværket: Træningstolerancen og Indlæringsraten.

Træningstolerancen sættes til 0.1, hvilket er den værdi netværks-output må variere med. Hvis en output-neuron kan antage værdier mellem 0 og 1, og den tilsigtede output-værdi er 0.5, skal netværkets output-værdi altså ligge mellem 0.4 og 0.6 for at blive betragtet som korrekt. Hvis træningstolerancen sættes til 0, skal netværkets output være lig med mønster-output'et, hvilket vil medføre at netværket aldrig bliver færdigt med at træne.

Indlæringsraten er en parameter, der gør det muligt at fastlægge, hvor meget netværket skal justere på de synaptiske vægte, afhængigt af den procentdel af trænings-data netværket har bestemt korrekt. Det er anbefalelsesværdigt at reducere indlæringsraten noget, når der arbejdes på at få de sidste data (facts) rigtige, det vil sige at lade BrainMaker tage mindre

"justerings-skridt" sidst i træningsforløbet. Dette kan forhindre netværket i at justere forbi et globalt minimum. Følgende opsætning er valgt:

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

Der bør på den anden side ikke nedjusteres for kraftigt på indlæringsraten, da dette vil bevirke, at netværket bliver meget lang tid om at træne færdigt, og der er risiko for, at det havner i et lokalt minimum.

BrainMaker kan oprette to filer indeholdende status for trænings- og testforløbet. Den første fil benævnes: [Filnavn].STS, og indeholder træningsstatistik. Den anden fil: [Filnavn].STA indeholder teststatistik. Først nævnte fil rummer for hvert gennemløb følgende parametre:

- 1) Den af netværket forbrugte træningstid.
- 2) Indlæringsraten.
- 3) Træningstolerancen.
- 4) Det totale antal gange, mængden af træningsdata er gennemløbet.
- 5) Antallet af rigtige træningsdata (GOOD) i hvert gennemløb.
- 6) Antallet af forkerte træningsdata (BAD) i hvert gennemløb.
- 7) Antallet af forkerte output-neuroner (BadOutput) i hvert gennemløb.
- 8) Antallet af forkerte output-neuroner hidtil i forløbet (TotalBadOutput).

- 9) Middelfejlen (AvgError) på netværkets output i forhold til det tilsigtede output for hvert gennemløb.
- 10) RMS fejlen (Root Mean Squared Error) på netværkets output i forhold til det tilsigtede output for hvert gennemløb.
- 11) Determinations-koefficienten for hver output-neuron ( $R^2$  [Outputneuronnavn]).
- 12) Antal gange hver output-neuron er forkert i hvert gennemløb (Bad [Outputneuronnavn]).

I en datamængde, som skal påtrykkes BrainMaker, kan nogle datasæt reserveres til test af netværket. I dette tilfælde bruges 10% af de 150 datasæt (ligningskoefficienter og rødder til test).

Det kan vælges, at teste et netværk efter et vist antal træningsgennemløb. Programmet skriver da en linie med teststatistik, til filen [Filnavn].STA efter hver test. Hvis det er valgt samtidigt at lagre netværket, kan der efter endt træning findes det netværk, som tester bedst.

Det vælges at udføre netværkstest efter hvert 20. gennemløb med de til rådighed værende testdata.

Det vælges også at lagre netværket på harddisken efter hvert 20. gennemløb. Filerne, hvori de trænede netværk lagres, benævnes: RUN00020.NET, RUN00040.NET etc. Hvis netværkstræningen er meget tidskrævende, og der lagres med små periodiske intervaller på harddisken, bliver denne hurtigt fyldt.

Efter endt træning viser det sig, at der er blevet tilføjet en enkelt neuron i hvert af de skjulte lag, og hele træningen af netværket er tilendebragt efter 680 iterationer. Træningsforløbet virkede fornuftigt, og for at opnå et bedre generaliserende netværk, trænes det selv samme net en gang til med "støj" tillagt netværkets input-data.

Der lægges 5% støj (5% afvigelse) på indgangen, og netværket trænes under navnet NN1\_5.

Træningen standses efter 1200 iterationer, selvom den ikke er tilendebragt. Dette gøres, da det for det første kan tage meget lang tid at træne et netværk med støj færdigt og for det andet ikke nødvendigvis bliver bedre efter lang tids træning. Der er størst interesse i at finde det netværk, som har de mindste afvigelser ved test.

Efter 480 gennemløb er:

GOOD = 15

BAD = 0

AvgError = 0.0403

RMSError = 0.0524

R<sup>2</sup> : X1 = 0.9383

R<sup>2</sup> : X2 = 0.9466

Bad X1 = 0

Bad X2 = 0

hvilket er de bedste testdata i træningsforløbet.

Netværket findes på medfølgende diskette i directory A:\NN1\_5.

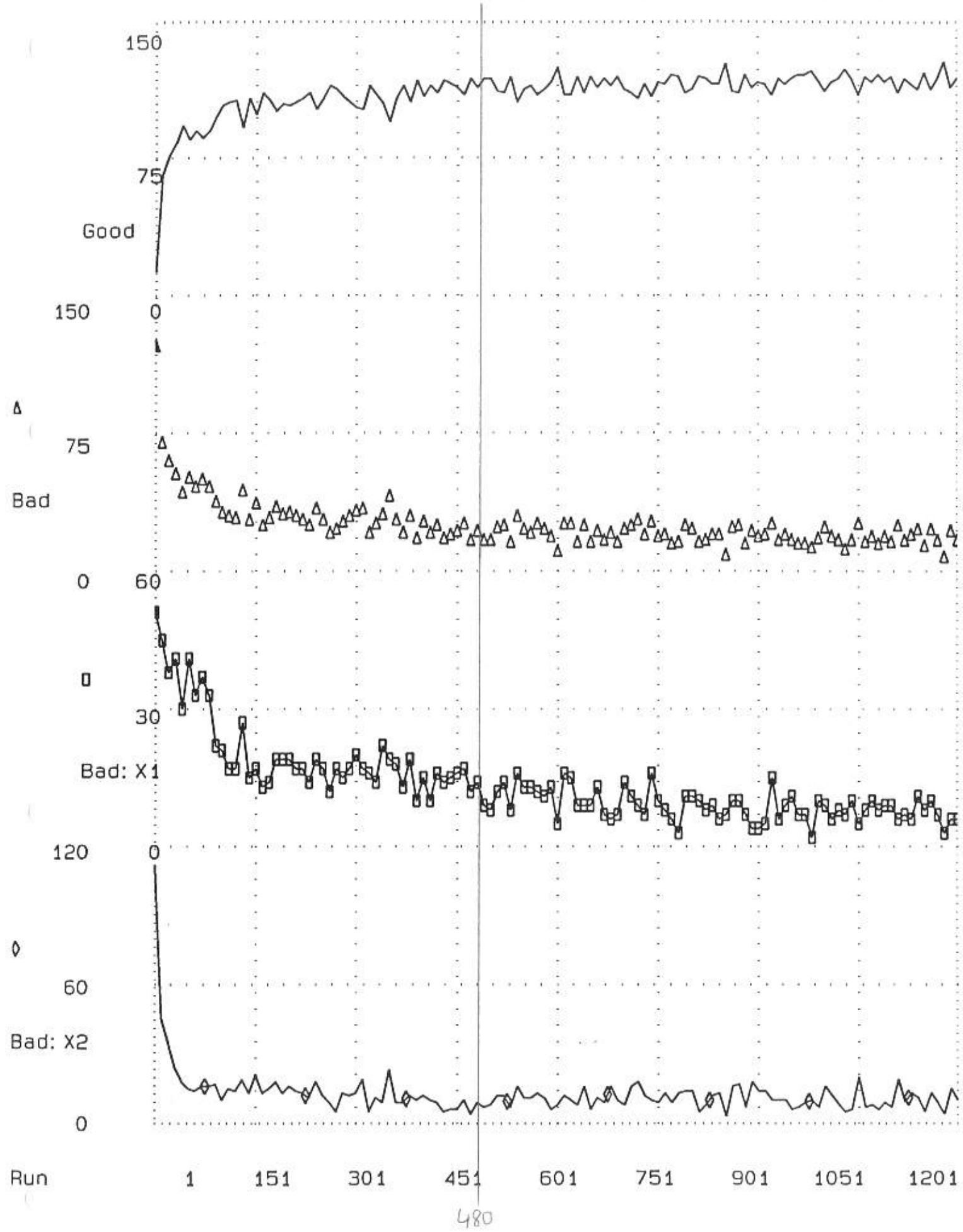
Nedenfor er vist nogle grafer med indhold fra NN1\_5.STS (Graf 4.1 og 4.2) og NN1\_5.STA (Graf 4.3 og 4.4).

Graf 4.1 viser øverst antallet af rigtige træningsdata (GOOD) som funktion af antal gennemløb. Næstøverst vises antallet af forkerte træningsdata (BAD) som funktion af antal gennemløb. Og de to sidste i graf 4.1 viser antal gange hver output-neuron X1 og X2 er forkerte i træningsforløbet som funktion af antal gennemløb.

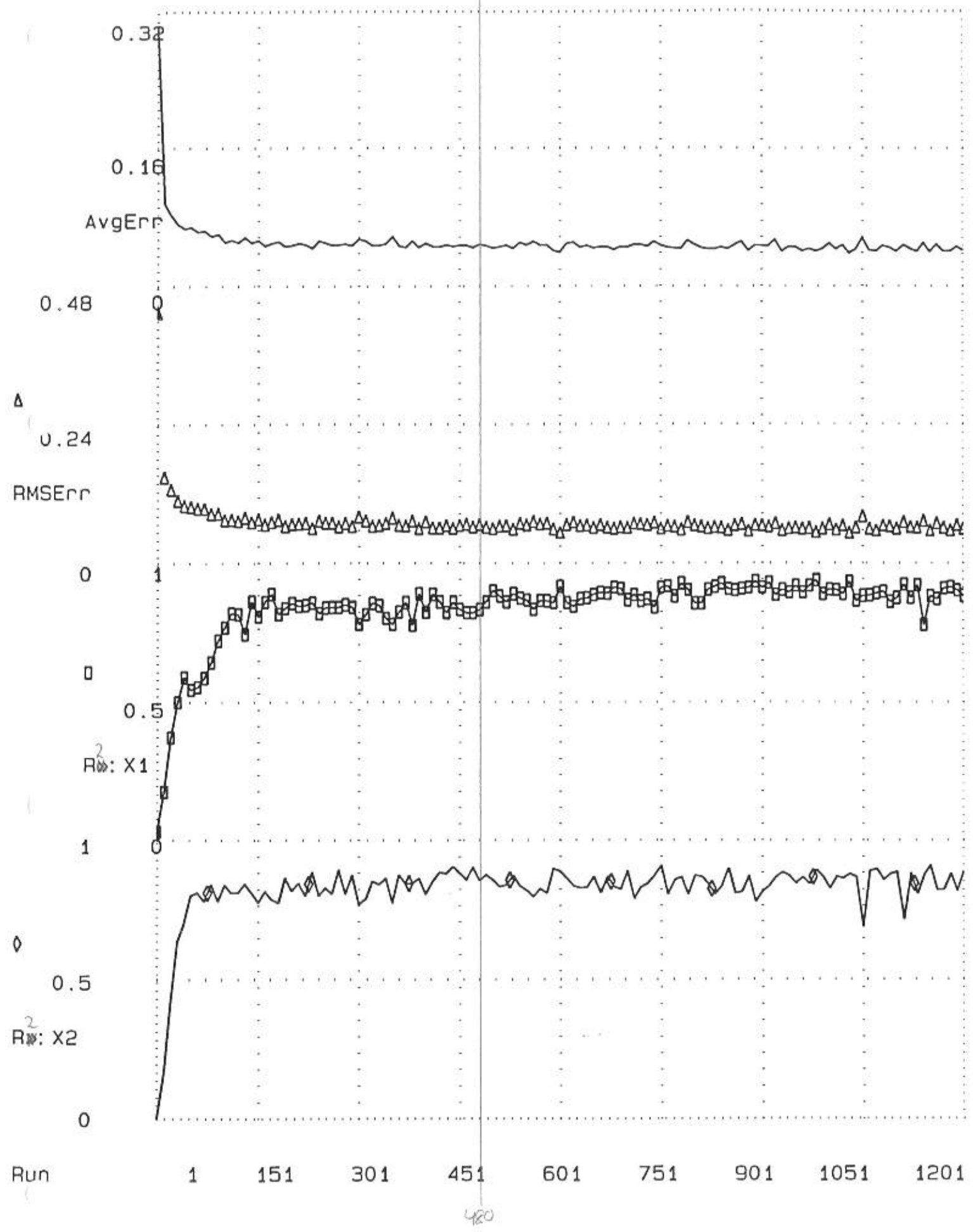


Graf 4.2 viser øverst middelfejlen på netværkets output i forhold til det tilsigtede output i træningsforløbet som funktion af antal gennemløb. Den næstøverste viser RMS fejlen på netværkets output i forhold til det tilsigtede output i træningsforløbet som funktion af antal gennemløb. Og de to sidste i graf 4.2 viser determinations-koefficienten for hver output-neuron i træningsforløbet som funktion af antal gennemløb.

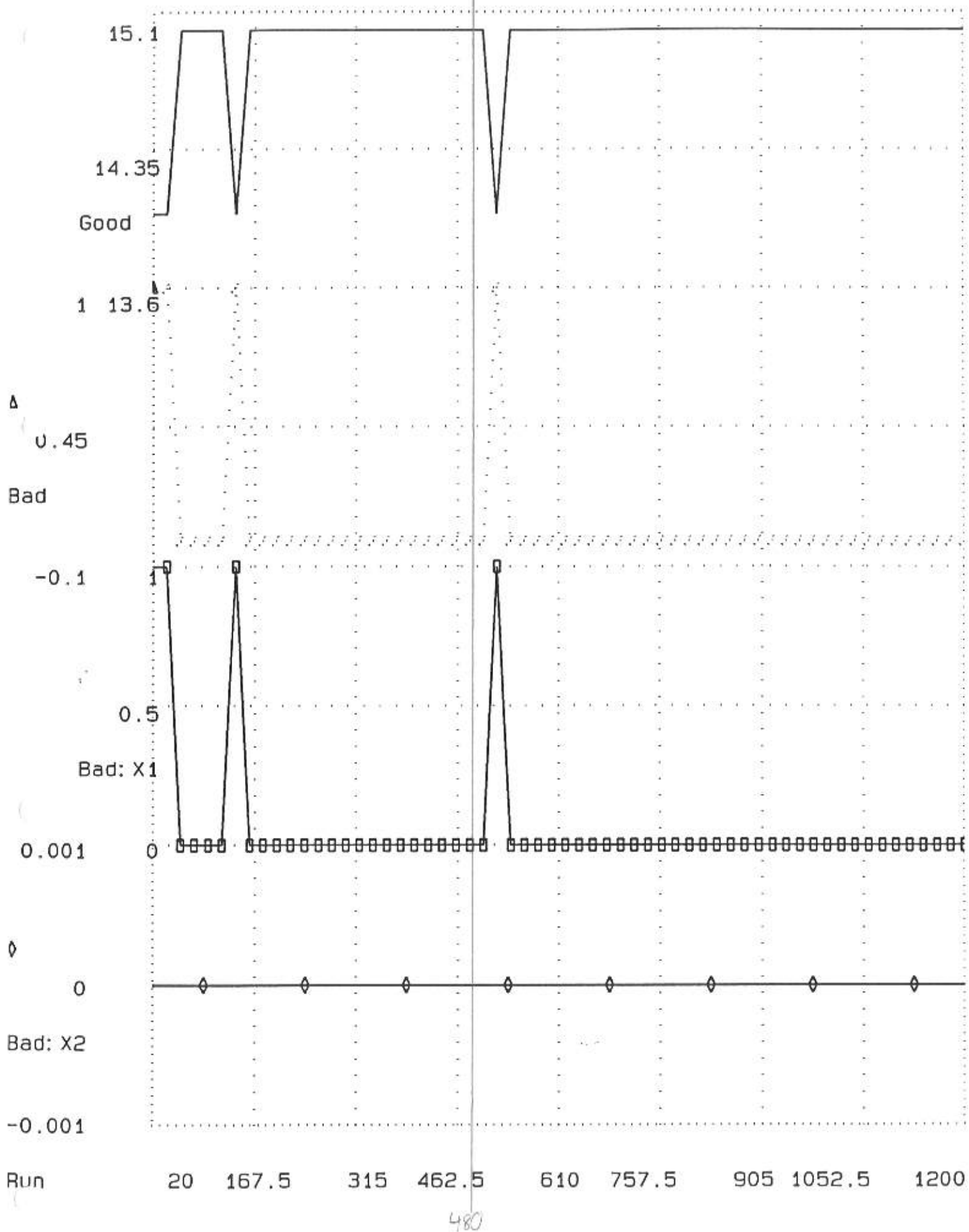
Graf 4.3 og graf 4.4 viser de samme parametre, som graf 4.1 og 4.2, men er afbildet ved test af netværket. Via graf 4.3 og 4.4 kan det konstateres at netværket tester bedst ved gennemløb 480.



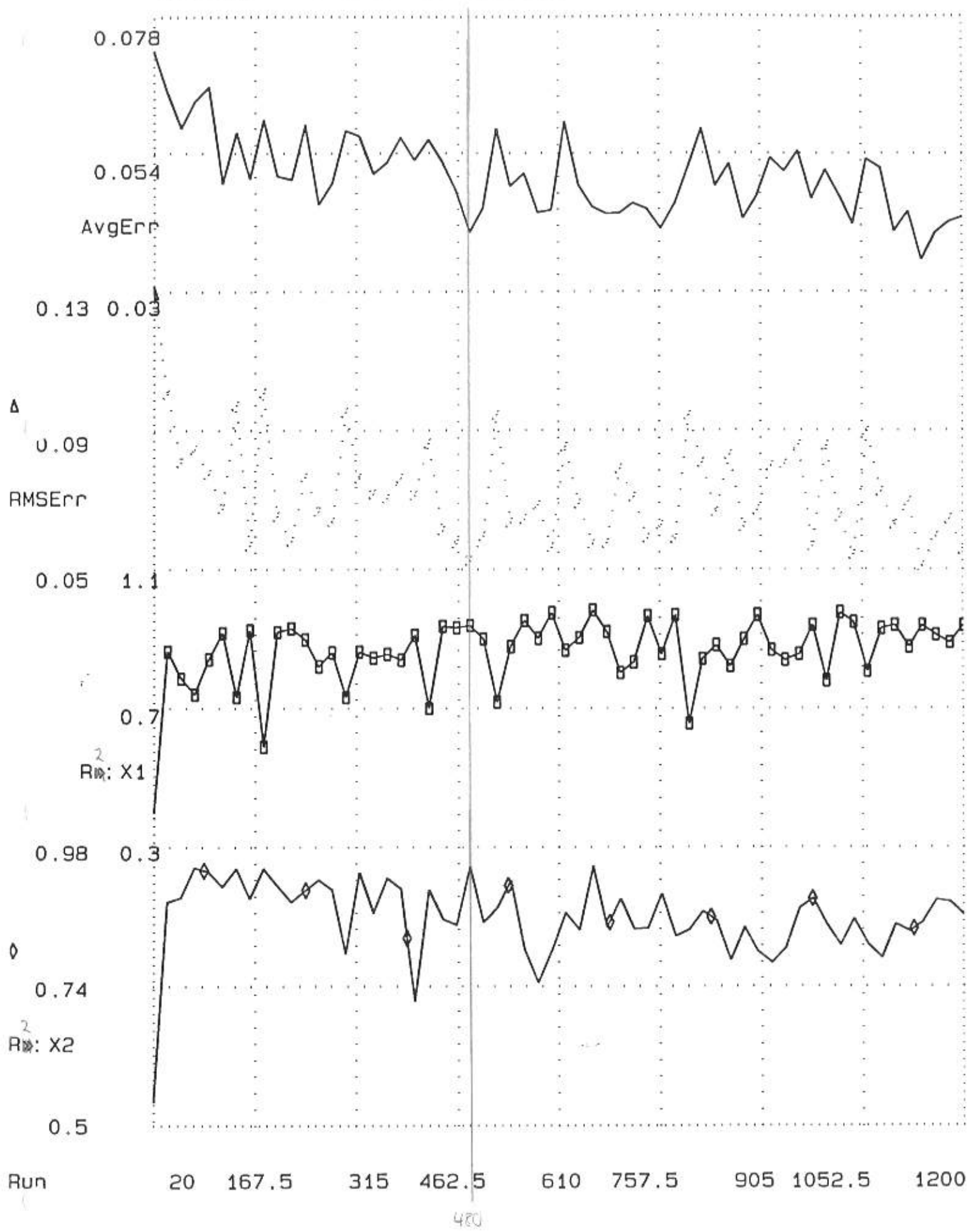
Graf 4.1 Netdata ved træning af NN1\_5 (NN1\_5.STS)



Graf 4.2 Netdata ved træning af NN1\_5 (NN1\_5.STS)



Graf 4.3 Netdata ved test under træning af NN1\_5 (NN1\_5.STA)



Graf 4.4 Netdata ved test under træning af NN1\_5 (NN1\_5.STA)

#### 4.6 Netværksafprøvning

Herefter påtrykkes netværket 150 nye A, B og C-koefficienter med dertil hørende reelle rødder, som det skal tage stilling til, ud fra, hvad det har lært sig under træningen.

^A betyder, at linien er en kommentarlinie, d.v.s at netværket ikke har kendskab til de to korrekte rødder.

```
^A 2.6      -0.1      <---- Matematisk resultat
3.758 -9.78  0.007    <---- Netværksinput (A, B og C)
3.465  0.022        <---- Netværksoutput
```

```
^A 0.92     -5.42
2.492  7.438 -10.1
0.215  -5.2
```

```
^A 0.16     -6.16
1.036  6.266 -2.31
0.203  -6.41
```

```
^A 0.29      0
7.052 -0.02  -2.21
0.888  -0.82
```

```
^A 0.54     -5.54
0.767  4.645 -2.03
0.265  -5.48
```

```
^A 2.07     -0.4
6.608 -10.9  -4.23
1.455  -0.53
```

```
^A 3.35      0.15
2.018  -6.1   1.069
3.405  -0.05
```

```
^A 1         -1
5.766  0.451 -4.58
0.87   -0.85
```

```
^A 0.74     -4.07
3.268  9.113 -9.74
0.22   -4.69
```

$\hat{A}$  0.87      -2.07  
4.933 5.407 -9.27  
0.698 -1.25

$\hat{A}$  -1.15      -7.85  
0.475 7.277 8.82  
0.17 -6.66

$\hat{A}$  1.39      -0.82  
6.935 -3.57 -8.37  
0.958 -0.77

$\hat{A}$  0.75      -2  
4.753 5.67 -7.32  
0.675 -1.29

$\hat{A}$  1      -0.33  
6.57 -4.56 -2.13  
0.928 -0.79

$\hat{A}$  1.7      -0.7  
4.38 -5.77 -5.92  
1.375 -0.59

$\hat{A}$  5      -2  
0.849 -1.76 -8.06  
2.733 -0.9

$\hat{A}$  0.52      -0.64  
8.174 0.924 -4.17  
0.895 -0.81

$\hat{A}$  -0.14      -1.46  
4.903 8.81 1.018  
0.61 -1.44

$\hat{A}$  1.9      -0.1  
5.057 -9.85 -1.28  
1.823 -0.4

$\hat{A}$  0      -1  
7.953 6.784 0.332  
0.858 -0.85

$\hat{A}$  2.21      -0.81  
4.58 -7.94 -9.22  
2.423 -0.19

$\hat{A}$  0.33      -0.33  
9.08 -1.27 -0.44  
0.895 -0.8

$\hat{A}$  0.13     -1.13  
7.287 7.995 -0.96  
0.825 -0.92

$\hat{A}$  0.34     -1.47  
7.594 11.56 -6.18  
0.748 -1.09

$\hat{A}$  0         -4  
2.426 8.805 -1.66  
0.198 -5.02

$\hat{A}$  0.58     -1.08  
8.35 2.931 -4.56  
0.89 -0.82

$\hat{A}$  -0.59     -3.41  
1.991 7.272 3.101  
0.25 -3.89

$\hat{A}$  1.06     -0.56  
10.17 -4.4 -5.25  
0.91 -0.79

$\hat{A}$  0.79     -0.56  
9.42 -3.54 -5.45  
0.913 -0.79

$\hat{A}$  0.63     -0.91  
6.766 0.748 -3.87  
0.885 -0.83

$\hat{A}$  6.46     -0.46  
1.264 -3.91 -4.67  
3.57 -0.2

$\hat{A}$  7.41     -0.41  
0.741 -5.53 -1.2  
6.31 0.319

$\hat{A}$  0.62     -0.32  
10.32 -4.1 -0.86  
0.898 -0.79

$\hat{A}$  0.65     -1.54  
9.332 8.043 -8.35  
0.875 -0.84

$\hat{A}$  0         -1.11  
8.289 9.059 0.624  
0.843 -0.87



^A -0.23 -1.43  
3.209 4.357 1.779  
0.613 -1.45

^A 0.76 -1.64  
8.247 6.53 -10.2  
0.865 -0.86

^A 4 -0.5  
1.807 -7.73 -6.2  
6.188 0.379

^A 3.81 -1.31  
2.007 -4.59 -9.1  
3.47 -0.1

^A 0 -1.17  
5.237 5.978 0.949  
0.758 -1.06

^A 0.87 -0.57  
10.1 -2.64 -4.88  
0.905 -0.79

^A 0.6 -1.47  
9.154 6.417 -5.74  
0.88 -0.82

^A 9.1 -1.1  
0.761 -7.09 -9.73  
6.445 -0.04

^A 0.44 -0.57  
8.133 -0.38 -1.65  
0.893 -0.8

^A 1.12 -0.79  
8.629 -4.09 -7.33  
0.923 -0.78

^A 1.76 -0.09  
5.933 -8.93 0.276  
1.16 -0.67

^A 1.2 -0.83  
7.298 -3.13 -8.55  
0.943 -0.78

$\hat{A}$  0.31     -1.91  
5.164 9.108 -1.64  
0.608 -1.46

$\hat{A}$  1.88     -0.46  
7.135 -8.68 -5.48  
1.103 -0.69

$\hat{A}$  0     -0.17  
5.617 2.33 -0.91  
0.843 -0.89

$\hat{A}$  -0.28     -0.72  
4.615 5.192 -0.25  
0.725 -1.14

$\hat{A}$  1.23     -0.61  
7.557 -5.12 -5.73  
0.94 -0.78

$\hat{A}$  0     -2.33  
2.145 6.237 -1.65  
0.285 -3.53

$\hat{A}$  3.11     -1.61  
1.873 -3.09 -10.6  
2.783 -0.37

$\hat{A}$  0.76     -1.76  
2.27 4.342 -5  
0.408 -2.6

$\hat{A}$  1.46     -0.68  
8.675 -5.11 -8.45  
0.948 -0.77

$\hat{A}$  1     -0.29  
6.399 -4.07 -1.99  
0.92 -0.79

$\hat{A}$  0.55     -1.45  
10.17 7.047 -8.49  
0.893 -0.81

$\hat{A}$  0.1     -1  
10.66 8.331 0.573  
0.878 -0.81

$\hat{A}$  3.07     -0.57  
3.448 -10.3 -8.42  
4.685 0.19

$\hat{A}$  3.22      0.78  
2.27   -7.81   5.467  
4.1     0.121

$\hat{A}$  0.12      -1.72  
4.353   7.565   -2.28  
0.565   -1.61

$\hat{A}$  2.27      -1.77  
2.165   -1.25   -7.23  
1.2     -1.01

$\hat{A}$  1.57      -0.32  
4.305   -3.65   -3.1  
1.025   -0.77

$\hat{A}$  0.76      -2.36  
5.048   9.352   -9.94  
0.488   -1.97

$\hat{A}$  0.26      -1.26  
8.638   9.85     -0.8  
0.843   -0.87

$\hat{A}$  1.17      -1.92  
3.773   1.725   -9.03  
0.773   -1.16

$\hat{A}$  0.57      -1.57  
10.32   10.53   -8.12  
0.875   -0.84

$\hat{A}$  0.81      -0.31  
7.522   -4.62   -3.51  
0.915   -0.79

$\hat{A}$  2.93      -0.68  
3.907   -8.16   -5.54  
2.693   -0.14

$\hat{A}$  2.39      -0.59  
4.509   -9.02   -5.8  
2.508   -0.18

$\hat{A}$  1.56      -0.27  
7.098   -7.5     -1.45  
0.965   -0.76

$\hat{A}$  1         -0.14  
7.181   -5.26   0.025  
0.91     -0.79

$\hat{A}$  1.09      -0.66  
6.305   -4.98   -4.86  
0.973   -0.76

$\hat{A}$  1          -0.8  
5.997   -1.22   -3.63  
0.895   -0.82

$\hat{A}$  0.33          -2  
3.006   4.977   -1.6  
0.51    -1.85

$\hat{A}$  1.35      -0.64  
7.175   -4.51   -6.93  
0.95    -0.77

$\hat{A}$  1.33      -0.53  
9.915   -8.54   -6.77  
0.953   -0.76

$\hat{A}$  0.76      -1.06  
10.24   2.931   -8.39  
0.903   -0.8

$\hat{A}$  0.34      -1.47  
8.695   9.352   -3.69  
0.853   -0.87

$\hat{A}$  1.37      -0.37  
6.149   -7.78   -2.28  
1.068   -0.71

$\hat{A}$  0.46      -0.31  
7.034   -2.2    -1.32  
0.893   -0.81

$\hat{A}$  0.5          0  
6.296   -3.13   0.652  
0.9      -0.81

$\hat{A}$  1            0  
8.456   -8.47   0.865  
0.923   -0.78

$\hat{A}$  0.22      -2.22  
1.651   5.138   -0.64  
0.298   -3.57

$\hat{A}$  0.18      -1.43  
4.002   4.225   -3.12  
0.678   -1.27

$\hat{A}$  6.54      0.46  
1.363 -6.49 3.055  
5.245 0.267

$\hat{A}$  0.3            -1  
10.13 6.959 -3.63  
0.885 -0.81

$\hat{A}$  0.55      -1.45  
9.39 8.956 -6.52  
0.868 -0.85

$\hat{A}$  1.74      -0.34  
4.973 -8.51 -3.49  
1.603 -0.48

$\hat{A}$  1.18      -0.85  
6.381 -2.61 -5.38  
0.918 -0.8

$\hat{A}$  1.79      -2.79  
2.09 1.696 -9.72  
0.673 -1.97

$\hat{A}$  -1.33            -2  
2.875 7.99 7.512  
0.395 -2.35

$\hat{A}$  0.38            0  
9.251 -3.08 0.295  
0.895 -0.8

$\hat{A}$  0.79      -1.12  
9.537 2.599 -9  
0.903 -0.8

$\hat{A}$  -1            -9  
0.618 9.328 9.043  
0.16 -7.94

$\hat{A}$  0.67      -1.3  
7.164 5.099 -7.01  
0.853 -0.88

$\hat{A}$  1.41      -1.41  
3.433 0.363 -7.63  
0.84 -1.07

$\hat{A}$  2.73      -0.73  
2.914 -6.22 -5.01  
2.678 -0.19

$\hat{A}$  1.41     -6.41  
1.176 5.016 -7.62  
0.258 -5.87

$\hat{A}$  0.27     -0.82  
8.17 3.698 -1.63  
0.88 -0.82

$\hat{A}$  0.65     -1.15  
3.784 2.287 -2.03  
0.748 -1.11

$\hat{A}$  0.75     -1.9  
7.107 8.112 -10.4  
0.793 -1.01

$\hat{A}$  0.58     -0.58  
3.327 0.485 -1.11  
0.808 -1.04

$\hat{A}$  0.53     -2.2  
5.812 10.78 -7.39  
0.558 -1.65

$\hat{A}$  1           0  
5.017 -4.31 0.656  
0.963 -0.78

$\hat{A}$  0.9     -0.12  
9.258 -6.6 -0.4  
0.903 -0.79

$\hat{A}$  0.16     -0.78  
7.783 5.016 -1.67  
0.868 -0.84

$\hat{A}$  1.69     -1.19  
1.457 -0.09 -5.82  
1.01 -1.55

$\hat{A}$  1     -1.8  
5.292 4.118 -9.91  
0.778 -1.06

$\hat{A}$  0.61     -4.11  
2.659 6.945 -4.92  
0.29 -3.5

$\hat{A}$  1.62     -0.62  
6.311 -7.54 -6.75  
1.185 -0.65

$\hat{A}$  0.14        0  
6.885 0.167 -0.1  
0.883 -0.82

$\hat{A}$  2.1        -0.1  
4.125 -10.2 -1.6  
3.415 0.017

$\hat{A}$  0.28       -0.39  
9.517 2.658 -0.8  
0.89 -0.8

$\hat{A}$  0.95       -1.06  
8.425 0.211 -9.26  
0.908 -0.8

$\hat{A}$  0.86       -1.86  
4.964 3.253 -7.42  
0.785 -1.03

$\hat{A}$  4.59       -1.09  
2.244 -6.81 -9.49  
4.555 0.149

$\hat{A}$  0        -4.5  
1.73 7.985 0.828  
0.183 -5.59

$\hat{A}$  0.87       -1.54  
5.854 4.259 -6.61  
0.815 -0.96

$\hat{A}$  0.77       -2.27  
3.727 6.398 -7.57  
0.475 -2.05

$\hat{A}$  1.71       0.29  
1.969 -5.23 0.953  
2.725 -0.23

$\hat{A}$  -0.5       -3  
2.255 6.955 2.549  
0.3 -3.22

$\hat{A}$  0.93       -0.43  
9.17 -4.86 -2.45  
0.905 -0.79

$\hat{A}$  1.72       -0.58  
6.909 -8.59 -7.97  
1.278 -0.6

$\hat{A}$  1.47     -1.14  
6.276   -0.73   -9.4  
0.918   -0.82

$\hat{A}$  2.2     -0.53  
5.804   -9.85   -8.73  
2.193   -0.25

$\hat{A}$  0     -0.67  
6.155   4.923   0.953  
0.828   -0.91

$\hat{A}$  2.69     -0.19  
1.693   -5.37   -1.34  
3.698   -0.03

$\hat{A}$  1.11     -3.61  
1.796   5.075   -8.85  
0.278   -4.54

$\hat{A}$  3.62     1.38  
2.385   -10.3   11.3  
5.965   0.437

$\hat{A}$  0.11     -3.11  
2.084   9.293   -2.03  
0.17     -6.48

$\hat{A}$  1.35     -0.57  
8.416   -8.01   -6.78  
0.995   -0.74

$\hat{A}$  1.34     -1.34  
5.111   0.128   -10.4  
0.908   -0.87

$\hat{A}$  1.18     -0.61  
6.316   -4.69   -6.65  
0.985   -0.76

$\hat{A}$  0.86     -3.86  
3.6     9.469   -10.7  
0.24     -4.36

$\hat{A}$  0.44     -1.69  
7.676   9.987   -6.81  
0.795   -0.99

$\hat{A}$  5.19     -0.19  
2.365   -10     -2.51  
6.99     0.531



$\hat{A}$  0.27      -1.47  
5.465 5.265 -2.45  
0.783 -1.01

$\hat{A}$  0.4      -1.4  
8.686 10.22 -5.49  
0.838 -0.9

$\hat{A}$  1      -1.17  
6.412 0.67 -6.64  
0.885 -0.84

$\hat{A}$  8.87      1.13  
0.937 -11.4 10.24  
8.475 0.476

$\hat{A}$  1.67      0  
5.854 -8.6 0.387  
1.135 -0.68

$\hat{A}$  0.33      -1  
8.236 6.92 -1.47  
0.863 -0.85

$\hat{A}$  1.45      -0.55  
10.47 -8.35 -9.48  
0.988 -0.74

$\hat{A}$  0.94      -0.94  
7.377 -0.39 -5.41  
0.898 -0.81

$\hat{A}$  7.85      1.15  
1.055 -7.92 9.697  
6.75 0.5

$\hat{A}$  0.3      0  
9.691 -5.54 -0.73  
0.903 -0.79

$\hat{A}$  -0.72      -2.78  
1.864 7.096 2.512  
0.238 -4.13

$\hat{A}$  0      -0.4  
4.925 0.289 1.023  
0.853 -0.88

#### 4.7 Konklusion

Det fremgår af ovenstående, at netværket ikke er i stand til at bestemme de eksakte værdier af ligningernes rødder, men det fremgår også, at netværket har en ganske god "formodning" om, hvor omkring rødderne ligger. Netværkets resultat ville måske være blevet mere præcist, hvis træningstolerancen var gjort mindre, men det neurale netværk vil aldrig kunne finde de eksakte rødder til en andengradsligning. Med dette forsøg er vist, hvordan et netværk kan konstrueres, og hvilken type opgave, det er mindre velegnet til at løse.