

## KAPITEL 7

## 7 INTEGRERING AF OPTRÆNET NETVÆRK I C-PROGRAMMEL

### 7.1 Brug af medfølgende C-fil: RUNTIME.C

I programpakken: BrainMaker Professionel findes en C-kode: RUNTIME.C, som gør det muligt at integrere og aktivere et færdigtrænet neuralt netværk i et C-miljø. Denne C-kode gør programpakken som helhed til et praktisk anvendeligt værktøj, da det ellers ikke er muligt at distribuere trænedede netværk kommercielt, fordi der på nuværende tidspunkt ikke kan erhverves licens til BrainMaker Professionel. RUNTIME.C-filen må ikke distribueres på source-kodeform, men må gerne videregives/sælges, hvis filen er bygget ind i en anden C-fil og kompileret. Denne C-kode findes i directory A:\RUNTIME på medfølgende diskette.

Når RUNTIME.C er blevet kompileret til RUNTIME.EXE kan et optrænet netværk aktiveres på følgende måde:

```
RUNTIME [FILNAVN].NET [FILNAVN].IN [FILNAVN].OUT
```

hvor [FILNAVN].NET er navnet på det optrænedede netværk - i dette tilfælde - RUN00480.NET fra opgave NN2\_4.

[FILNAVN].IN er en fil indholdende det datasæt, netværket skal påtrykkes - i dette tilfælde TEST.IN, der indeholder følgende ti datasæt med koefficienterne A, B og C:

2	9	1
1	10	5
4	5	10
1	5.2	7
5	8.8	-3
7	3	-9.7
2.2	-4	5
4.4	4	7.8
1	-8.9	1
3	-1.4	4.3

[FILNAVN].OUT er den fil, hvori nettets output ønskes lagret, i dette tilfælde TEST.OUT, der vil rumme en kolonne med ti tal mellem 0 og 1.

Når ovenstående RUNTIME-kommandolinie udføres, skrives følgende på skærmen:

```
Reading weights from run00480.net,  
reading inputs from test.in,  
writing outputs to test.out,  
reading minmax from run00480.net.  
Network size: 4 thresholds, 1 layers, layer sizes: 3  
Done.
```

TEST.OUT viser sig at indeholde følgende tal:

```
0.111  0.950  0.550  
0.000  1.000  0.750  
0.333  0.750  1.000  
0.000  0.760  0.850  
0.444  0.940  0.350  
0.667  0.650  0.015  
0.133  0.300  0.750  
0.378  0.700  0.890  
0.000  0.055  0.550  
0.222  0.430  0.715
```

Dette giver imidlertid ikke megen mening, jvf. ovenstående beskrivelse af det forventede indhold. Ved granskning af RUNTIME.C-koden findes en fejl i koden, se rund ring i bilag 3.

## 7.2 Fejlretning og test af RUNTIME.C

Fejlen omtalt i afsnit 7.1 rettes, og filen kaldes herefter: RUNT\_OK.C. Efter kompilering af denne fil, gøres forsøget endnu en gang:

```
RUNT_OK RUN00480.NET TEST.IN TEST_OK.OUT
```

Herved skrives på skærmen:

```
Reading weights from run00480.net,  
reading inputs from test.in,  
writing outputs to test_ok.out,  
reading minmax from run00480.net.  
Network size: 1 thresholds, 4 layers, layer sizes: 3 7 6 1  
Done.
```

Netværket rummer således 4 lag: Et input-lag med tre neuroner (A, B og C), to skjulte lag med henholdsvis syv og seks neuroner og et output-lag med en neuron (R).

TEST\_OK.OUT indeholder værdierne:

```
0.921  
0.740  
0.010  
0.025  
0.936  
0.998  
0.032  
0.011  
0.983  
0.025
```

Det ønskes testet om teoretisk beregning, BrainMaker.EXE og RUNT\_OK.EXE med værdierne i TEST.IN filen fører til samme resultat, hvilket undersøges nedenfor:

A	B	C	TEORI	BrainMaker	RUNT_OK
2	9	1	1	0.9212	0.921
1	10	5	1	0.7398	0.740
4	5	10	0	0.0101	0.010
1	5.2	7	0	0.0252	0.025
5	8.8	-3	1	0.9361	0.936
7	3	-9.7	1	0.9979	0.998
2.2	-4	5	0	0.0321	0.032
4.4	4	7.8	0	0.0111	0.011
1	-8.9	1	1	0.9832	0.983
3	-1.4	4.3	0	0.0247	0.025

Det ses af tabellen, at de påtrykte data i TEST.IN fører til det rette resultat ved brug af både BrainMaker.EXE og RUNT\_OK.EXE.

### 7.3 Udvikling af C-program: RUN\_NN.C

Det ønskes herefter at videreudvikle RUNT\_OK.C, så det optrænete netværk fra kapitel 5 (NN2\_4) kan afvikles uden brug af BrainMaker Professionel. Den videreudviklede C-kode: RUN\_NN.C er vist i bilag 4 med flowcharts over de nyudviklede rutiner.