WOJSKOWA AKADEMIA TECHNICZNA

LABORATORIUM
METODY INFORMATYCZNEGO WSPOMAGANIA DECYZJI

Stopień, imię i nazwisko prowadzącego	Stopień, imię i nazwisko słuchacza	Grupa szkoleniowa
dr Jarosław Olejniczak	***inż. Grzegorz Pol***	***I0G1S4***

Data wykonania ćwiczenia
15.01.2011 r.

SPRAWOZDANIE
Z
PRACY LABORATORYJNEJ
NR 5

Temat: Sieci neuronowe. Agencja nieruchomości

1. Treść zadania

Wygenerować dane losowych preferencji dla 100 przypadków mieszkań dla cech: Duże (D), Tanie (T), Uzbrojone (U), Blisko Centrum (BC). Każda z tych cech może przybrać wartości wag od 0 do 8. Należy wykorzystać funkcję LOS().

Zbudować funkcję klasyfikującą każdą ze 100 nieruchomości do grupy 1 lub drugiej według odpowiednio przyjętej reguły np.: Jeśli (D)>=5 i (T)>=6 i (U)>=7 i (BC)>=4 to grupa pierwsza. Pozostałe mieszkania są zaliczane do grupy 2.

Zbudować sieć neuronową dostosowaną do klasyfikacji mieszkań do grupy pierwszej lub drugiej. Sformułować wnioski dotyczące wygenerowanej sieci.

2. Sposób realizacji

W celu realizacji postawionego w treści zadania stworzyłem w programie kalkulacyjnym Excel plik *arkusz.xls*, który wykorzystałem w celu wygenerowania wag w przedziale od 0 do 8 dla wybranych cech mieszkań. Wykonałem to przy pomocy funkcji: *=ZAOKR.DO.CAŁK(LOS()*(0+0)+0)*.

Kolejny krok polegał na przydzieleniu nieruchomości do grup pierwszej bądź drugiej. Do tego stworzyłem funkcję, która przydziela do pierwszej grupy wyłącznie nieruchomości spełniające poniższą zależność: *=JEŻELI((Arkusz1!A2>1)*(Arkusz1!B2>1)*(Arkusz1!C2>1)*(Arkusz1!D2>=0);1;2)*. Funkcja ta mniej więcej po równo przydziela nieruchomości do dwóch grup

W celu zdobycia wyciągnięcia wniosków zrealizowałem zadanie na trzech sieciach neuronowych różniących się ilością parametrów (2,2,2,1 | 10,10,10,1 | 40,40,40,1)

3. Realizacja w aplikacji R

```
R Console
> library(RODBC)
> library(AMORE)
> setwd("C:/Users/Admin/Desktop/miw")
> plik <- odbcConnectExcel("arkusz.xls")
> WE <- sqlQuery(plik , "select * from [Arkusz1$]")
> WY <- sqlQuery(plik , "select * from [Arkusz2$]")
> VAL_WE <- sqlQuery(plik , "select * from [Arkusz3$]")
> VAL_WY <- sqlQuery(plik , "select * from [Arkusz4$]")
> WE=as.matrix(WE)
> WY=as.matrix(WY)
> VAL_WE=as.matrix(VAL_WE)
> VAL_WY=as.matrix(VAL_WY)
> net <- newff(n.neurons=c(40,40,40,1), learning.rate.global=1e-2, momentum.global=0.5, error.cr$
> result <- train(net, WE, WY, VAL_WE, VAL_WY, error.criterium="LMS", report=TRUE, show.step=200$
index.show: 1 LMS       TRAIN: 0.026922973164542        VAL: 0.0534916542121662          BEST NET
index.show: 2 LMS       TRAIN: 0.019563228228927        VAL: 0.0733338476253887
index.show: 3 LMS       TRAIN: 0.00704296070775551      VAL: 0.0574554851649879
index.show: 4 LMS       TRAIN: 0.00241826430192698      VAL: 0.0484976994805192          BEST NET
index.show: 5 LMS       TRAIN: 0.00116014809950494      VAL: 0.0452860559760846          BEST NET
index.show: 6 LMS       TRAIN: 0.000865449054407807     VAL: 0.0442840780148048          BEST NET
index.show: 7 LMS       TRAIN: 0.000834563355186235     VAL: 0.0440531674911195          BEST NET
index.show: 8 LMS       TRAIN: 0.000849938844328716     VAL: 0.0440662677723918
index.show: 9 LMS       TRAIN: 0.0008547550880579       VAL: 0.0441547707272196
index.show: 10 LMS      TRAIN: 0.000841576938544398     VAL: 0.0442786422937656
```

Sieć neuronowa (2,2,2,1)			Sieć neuronowa (10,10,10,1)			Sieć neuronowa (40,40,40,1)		
1	2	1.952555	1	1	0.9822185	1	1	0.9758745
2	2	2.044184	2	2	1.9670308	2	2	2.0093244
3	1	1.383858	3	2	1.9842562	3	2	2.0385232
4	2	1.362445	4	2	2.0999684	4	2	2.0982447
5	1	1.362445	5	1	0.9992730	5	1	0.9516988
6	2	1.520953	6	1	0.9955760	6	1	0.9417310
7	1	1.383546	7	2	1.8323909	7	2	1.7957501
8	1	1.383546	8	2	1.9916305	8	2	2.0085268
9	1	1.380411	9	1	0.9891310	9	1	1.0527599
10	1	1.371070	10	1	1.0068087	10	1	0.9833775
11	2	1.929884	11	2	1.8116661	11	2	1.9236933
12	1	1.327744	12	1	1.0128598	12	1	0.9966903
13	2	1.382943	13	2	1.8268968	13	2	1.8783260
14	1	1.382299	14	2	2.0198938	14	2	1.9887815
15	1	1.379371	15	2	1.9914412	15	2	1.9847307
16	1	1.383546	16	1	0.9908914	16	1	0.9748245
17	1	1.371070	17	2	1.5325400	17	2	1.8599796
18	1	1.374002	18	2	2.0129342	18	2	2.0075633
19	2	1.756049	19	2	1.8240223	19	2	2.1347794
20	2	1.383862	20	2	1.9852106	20	2	1.9668731
21	2	1.383157	21	2	2.0452312	21	2	2.0893856
22	2	1.367253	22	1	1.0009070	22	1	0.9984538
23	2	1.383862	23	1	1.0034197	23	1	0.9793618
24	1	1.383449	24	1	0.9807850	24	1	0.9533838
25	2	2.015302	25	2	1.9896241	25	2	1.9786079
26	1	1.383449	26	2	1.9850304	26	2	1.9320471
27	2	2.015302	27	1	1.0043774	27	1	1.0064122
28	1	1.383862	28	2	2.0555849	28	2	2.0144194
29	1	1.356272	29	1	1.1164440	29	1	1.1928062
30	2	1.348505	30	1	0.9897457	30	1	0.9889039
31	1	1.383861	31	2	2.0241110	31	2	1.9339744
32	2	1.727704	32	1	1.0063070	32	1	0.9791654
33	2	1.929884	33	1	0.9743293	33	1	0.9812165
34	2	2.015302	34	1	1.3388249	34	1	0.8822216
35	1	1.315462	35	1	0.9827982	35	1	0.9812818
36	2	2.055315	36	2	1.3427467	36	2	1.9056912
37	1	1.383830	37	2	1.9392140	37	2	2.1554983
38	2	1.929884	38	2	1.9310262	38	2	1.9950592
39	1	1.362445	39	2	1.9057995	39	2	1.7613247
40	1	1.383546	40	2	1.9724630	40	2	1.9939925
41	1	1.383546	41	2	1.9858377	41	2	1.9654425
42	2	1.881268	42	2	1.4275031	42	2	1.4713011
43	1	1.383862	43	2	2.1251624	43	2	2.2066656
44	2	1.383546	44	2	2.0116435	44	2	2.0126850
45	2	1.327666	45	1	2.0116902	45	1	1.7932115
46	2	1.381210	46	1	0.9984248	46	1	0.9972320
47	2	1.315654	47	2	2.0150366	47	2	2.0031798
48	2	1.951347	48	1	1.0042952	48	1	1.0035725
49	1	1.381826	49	1	0.9724591	49	1	0.9577914
50	2	1.383723	50	2	2.0240402	50	2	2.0043122

51	1	1.383783	51	2	1.9226177	51	2	1.9769939
52	1	1.521027	52	1	1.9844763	52	1	1.6519735
53	1	1.383158	53	1	1.0064842	53	1	0.9926616
54	2	2.036358	54	1	1.0004521	54	1	1.0165213
55	1	1.374002	55	2	2.1255239	55	2	2.1782648
56	2	1.995717	56	1	0.9712901	56	1	0.9856817
57	1	1.370470	57	2	2.0534889	57	2	1.9804410
58	2	2.065024	58	2	1.9749378	58	2	1.9801405
59	1	1.370470	59	2	2.0431966	59	2	1.7198134
60	2	2.007698	60	2	1.6223481	60	2	1.7352260
61	1	1.370470	61	1	0.9709747	61	1	0.9690772
62	1	1.338942	62	1	1.7174170	62	1	1.0035206
63	2	1.383322	63	1	0.9891192	63	1	0.9918852
64	2	1.327666	64	1	1.0721058	64	1	0.9846871
65	2	1.383723	65	1	1.6517901	65	1	1.2496507
66	1	1.338942	66	1	0.9903824	66	1	0.9967960
67	1	1.383839	67	2	1.6105128	67	2	1.7442503
68	2	1.315462	68	1	1.0066952	68	1	1.0063021
69	2	1.929884	69	2	1.9945176	69	2	2.0196850
70	2	1.367253	70	1	1.0053417	70	1	0.9997311
71	2	1.383757	71	2	1.9299778	71	2	2.0797841
72	2	1.382943	72	1	1.0047078	72	1	0.9976309
73	2	1.964193	73	1	1.0067908	73	1	0.9881945
74	1	1.307736	74	2	1.9948401	74	2	1.9631532
75	1	1.383757	75	2	1.9519353	75	2	2.1239834
76	2	1.727704	76	2	2.0663342	76	2	2.1189000
77	1	1.383158	77	1	1.0146415	77	1	0.9895960
78	2	1.327744	78	1	0.9998892	78	1	1.0059694
79	2	1.383157	79	2	1.9820488	79	2	1.9161831
80	2	1.383783	80	2	1.9561707	80	2	1.9510638
81	1	1.383864	81	1	0.9738303	81	1	0.9684971
82	1	1.383802	82	2	1.9761082	82	2	1.9777409
83	1	1.362445	83	2	1.9733725	83	2	1.9970087
84	2	1.307736	84	1	1.7273437	84	1	1.2798203
85	1	1.370470	85	1	0.9910835	85	1	0.9918796
86	2	1.978591	86	2	2.0400879	86	2	2.0151466
87	2	1.383723	87	1	1.0057007	87	1	1.0101443
88	1	1.327744	88	1	0.9818412	88	1	1.0620510
89	2	1.371060	89	1	0.9731460	89	1	0.9520852
90	2	1.381210	90	1	1.0011594	90	1	0.9936031
91	2	1.315462	91	2	2.0144762	91	2	1.9887297
92	2	1.756049	92	1	0.9950734	92	1	0.9537197
93	1	1.374002	93	1	1.3325542	93	1	1.0424668
94	2	1.383621	94	1	1.4306559	94	1	0.9808139
95	1	1.383679	95	1	1.0068147	95	1	0.9841960
96	2	2.044184	96	2	2.0262443	96	2	2.0042660
97	2	1.381827	97	2	1.9598640	97	2	1.9920301
98	1	1.381211	98	1	1.0062431	98	1	0.9990359
99	2	1.383757	99	1	1.0049841	99	1	1.0911322
100	2	1.367253	100	1	1.0543915	100	1	0.9921366

4. Obliczenia

Za pomocą arkusza (*wynik.xls*) obliczyłem poprawność klasyfikacji. Wyniosła ona odpowiednio:

- sieć neuronowa o parametrach (2,2,2,1) – poprawność **68%**
- sieć neuronowa o parametrach (10,10,10,1) – poprawność **93%**
- sieć neuronowa o parametrach (40,40,40,1) – poprawność **97%**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1			(2,2,2,1)						(10,10,10,1)						(40,40,40,1)		
2	LP.	GR	WART	ZA	OK?		LP.	GR	WART	ZA	OK?		LP.	GR	WART	ZA	OK?
3	1	2	1,952555	2	1		1	1	0,9822185	1	1		1	1	0,9758745	1	1
4	2	2	2,044184	2	1		2	2	1,9670308	2	1		2	2	2,0093244	2	1
5	3	1	1,383858	1	1		3	2	1,9842562	2	1		3	2	2,0385232	2	1
6	4	2	1,362445	1	0		4	2	2,0999684	2	1		4	2	2,0982447	2	1
7	5	1	1,362445	1	1		5	1	0,999273	1	1		5	1	0,9516988	1	1
8	6	2	1,520953	2	1		6	1	0,995576	1	1		6	1	0,941731	1	1
9	7	1	1,383546	1	1		7	2	1,8323909	2	1		7	2	1,7957501	2	1
10	8	1	1,383546	1	1		8	2	1,9916305	2	1		8	2	2,0085268	2	1
11	9	1	1,380411	1	1		9	1	0,989131	1	1		9	1	1,0527599	1	1
12	10	1	1,37107	1	1		10	1	1,0068087	1	1		10	1	0,9833775	1	1
13	11	2	1,929884	2	1		11	2	1,8116661	2	1		11	2	1,9236933	2	1
14	12	1	1,327744	1	1		12	1	1,0128598	1	1		12	1	0,9966903	1	1
15	13	2	1,382943	1	0		13	2	1,8268968	2	1		13	2	1,878326	2	1
16	14	1	1,382299	1	1		14	2	2,0198938	2	1		14	2	1,9887815	2	1
17	15	1	1,379371	1	1		15	2	1,9914412	2	1		15	2	1,9847307	2	1
18	16	1	1,383546	1	1		16	1	0,9908914	1	1		16	1	0,9748245	1	1
86	84	2	1,307736	1	0		84	1	1,7273437	2	0		84	1	1,2798203	1	1
87	85	1	1,37047	1	1		85	1	0,9910835	1	1		85	1	0,9918796	1	1
88	86	2	1,978591	2	1		86	2	2,0400879	2	1		86	2	2,0151466	2	1
89	87	2	1,383723	1	0		87	1	1,0057007	1	1		87	1	1,0101443	1	1
90	88	1	1,327744	1	1		88	1	0,9818412	1	1		88	1	1,062051	1	1
91	89	2	1,37106	1	0		89	1	0,973146	1	1		89	1	0,9520852	1	1
92	90	2	1,38121	1	0		90	1	1,0011594	1	1		90	1	0,9936031	1	1
93	91	2	1,315462	1	0		91	2	2,0144762	2	1		91	2	1,9887297	2	1
94	92	2	1,756049	2	1		92	1	0,9950734	1	1		92	1	0,9537197	1	1
95	93	1	1,374002	1	1		93	1	1,3325542	1	1		93	1	1,0424668	1	1
96	94	2	1,383621	1	0		94	1	1,4306559	1	1		94	1	0,9808139	1	1
97	95	1	1,383679	1	1		95	1	1,0068147	1	1		95	1	0,984196	1	1
98	96	2	2,044184	2	1		96	2	2,0262443	2	1		96	2	2,004266	2	1
99	97	2	1,381827	1	0		97	2	1,959864	2	1		97	2	1,9920301	2	1
100	98	1	1,381211	1	1		98	1	1,0062431	1	1		98	1	0,9990359	1	1
101	99	2	1,383757	1	0		99	1	1,0049841	1	1		99	1	1,0911322	1	1
102	100	2	1,367253	1	0		100	1	1,0543915	1	1		100	1	0,9921366	1	1
103			SUMA:		68				SUMA:		93				SUMA:		97

5. Wnioski

Na podstawie obliczeń i obsługi programu R mogę wyciągnąć następujące wnioski:

- Czym większa ilość neuronów dla poszczególnych warstw tym wynik jest dokładniejszy
- Dla każdego zadania istnieje rozsądna ilość warstw, której przekroczenie mija się z celem (zwiększenie warstw nie zwiększa w sposób efektywny poprawności)
- Zwiększenie warstw wydłuża w znaczny sposób czas na naukę
- Wynik przy 40 warstwach dla naszego zadania jest dokładny (97%)
- Nauka jest efektywniejsza gdy operujemy na symetrycznym rozkładzie