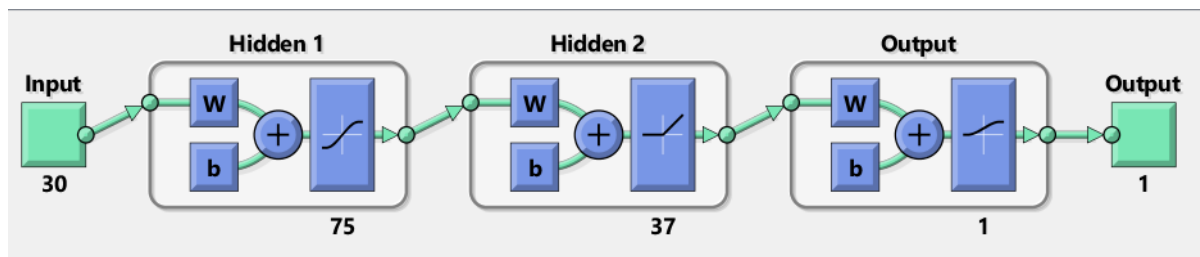


Am impartit problema initiala de clasificare a exemplelor in 10 clase: 0, 1, .. 9 in 2 subprobleme:

- ❖ clasificarea tuturor exemplelor in 2 clase: 0(neapartinand clasei 9) si 1(apartinand clasei 9)
- ❖ clasificarea exemplelor din clasele: 0, 1, .. 8

Pentru fiecare subproblema, am construit o retea neuronală pentru a o rezolva. In final, am combinat rezultatele celor doua modele asa cum este descris mai jos:

#### Arhitectura retelei I



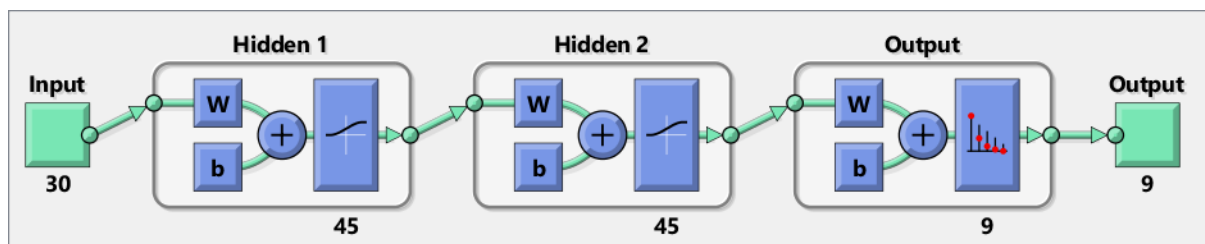
Reteaua rezolva problema clasificarii exemplelor in clasa 0(neapartinand clasei initiale 9) si 1(apartinand clasei initiale 9)

Reteaua(patternnet) este alcatuita din 3 straturi:

- ❖ 2 straturi ascunse (primul de 75 de perceptroni cu functia de transfer „tansig”, al doilea de 37 de perceptroni cu functia de transfer „poslin/RELU”)
- ❖ 1 strat de iesire cu 1 perceptron cu functia de transfer „logsig”

Reteaua se antreneaza cu functia de antrenare „traingdx”, functia de performanta „crossentropy”. Foloseste 80% din date pentru antrenare si 20% pentru validare si se opreste cand misclasifica la rand 100 de exemple (net9.trainParam.max\_fail = 100). De asemenea, reteaaua are la dispozitie 5000 de epoci pentru a se antrena si are setata eroarea de atins la 1e-5.

#### Arhitectura retelei 2



Reteaua clasifica exemplele din clasele: 0, 1, .. 8. Pentru antrenare nu am folosit clasele initiale, ci am transformat fiecare clasa x intr-un vector coloana cu 9 linii(una pentru fiecare clasa) si am setat linia x(corespunzatoare clasei respective) la 1.

Reteaua(patternnet) este alcatuita din 3 straturi:

- ❖ 2 straturi ascunse, amandoua cu 45 de perceptroni si functia de transfer „logsig”
- ❖ 1 strat de iesire cu 9 perceptroni cu functia de transfer „softmax”

Reteaua se antreneaza cu functia de antrenare „traingdx”, functia de performanta „crossentropy”. Foloseste 80% din date pentru antrenare si 20% pentru validare si se opreste cand misclasifica la rand 25 de exemple (net9.trainParam.max\_fail = 25). De asemenea, reseaua are la dispozitie 5000 de epoci pentru a se antrena si are setata eroarea de atins la 1e-5.

#### Antrenarea si obtinea claselor finale

Prima retea folosea toate cele 26700 de exemple pentru a le clasifica cu 0 si 1, asa cum am descris mai sus. Alegerea modelului o facem in urmatorul fel: am impartit datele in 10 submultimi de 2670 de exemple fiecare, la fiecare pas i (de la 1 la 10) alegeam a i-a multime ca fiind de testare, iar restul le combinam intr-o multime de antrenare. Antrenam reseaua pe datele corespunzatoare antrenarii si evaluam performanta atat de multimea de antrenare cat si pe cea de testare. Performanta finala a modelului o calculam ca o medie ponderata:  $\text{perfTotal} = 0.25 * \text{perfTrain} + 0.75 * \text{perfTest}$ . Alegeam din cele 10 modele antrenate pe cel cu perfTotal maxima pentru a fi folosit mai departe.

A doua retea folosea doar exemplele care aveau etichete intre 0 si 8 (6704 exemple). Alegerea modelului o facem in urmatorul fel: am impartit datele in 8 submultimi de 838 de exemple fiecare, la fiecare pas i (de la 1 la 8) alegeam a i-a multime ca fiind de testare, iar restul le combinam intr-o multime de antrenare. Antrenam reseaua pe datele corespunzatoare antrenarii si evaluam performanta atat de multimea de antrenare cat si pe cea de testare. Performanta finala a modelului o calculam ca o medie ponderata:  $\text{perfTotal} = 0.25 * \text{perfTrain} + 0.75 * \text{perfTest}$ . Alegeam din cele 8 modele antrenate pe cel cu perfTotal maxima pentru a fi folosit mai departe.

In final, dupa ce obtineam cele doua modele „net9” pentru prima clasificare si „netDiff” pentru cea de-a doua clasificare, combinam rezultatele lor astfel:

Pentru datele pentru care voiam sa prezic etichetele:

- ❖ le clasificam cu clasele 0 si 1 folosind net9
- ❖ exemplele care obtinusera clasa 0 in prima clasificare (clasificate ca nu fiind din clasa 9), le clasificam cu netDiff in clasele corespunzatoare
- ❖ in vectorul de etichete finale, 1-urile se inmulteau cu 9, iar 0-urile erau inlocuite cu clasele obtinute prin clasificarea cu netDiff

La o antrenare, am obtinut urmatoarele:

#### Reteaua I

PerfTrain PerfTest PerfTotal

"0.931 0.932 0.932"

"0.940 0.928 0.931"

"0.938 0.927 0.929"

"0.938 0.931 0.932"

"0.937 0.930 0.932"

"0.937 0.924 0.927"

"0.936 0.930 0.931"

"0.937 0.927 0.930"

"0.935 0.931 0.932"

"0.936 0.936 0.936"

meanAcc9 = 0.9296 (media performantelor de testare pe cele 10 multimi de testare)

Matricele de confuzie pentru cele 10 multimi de testare sunt:

560	137
45	1928

562	128
63	1917
530	125
71	1944

546	126
59	1939

553	129
57	1931

571	127
75	1897

563	120
67	1920

519	138
56	1957

486	123
62	1999

560	101
70	1939

$M(i, j)$  = numarul de exemple din clasa i care au fost clasificate in clasa j

Reteaua II

PerfTrain PerfTest PerfTotal

"0.855 0.809 0.821"

"0.852 0.823 0.831"

"0.857 0.811 0.823"

"0.847 0.813 0.821"

"0.827 0.817 0.820"

"0.850 0.840 0.842"

"0.841 0.825 0.829"

"0.845 0.817 0.824"

meanAccDiff = 0.8195 (media performantelor de testare pe cele 8 multimi de testare)

Matricele de confuzie pentru cele 8 multimi de testare sunt:

1.

93	0	0	0	1	0	4	0	3
0	65	12	6	2	4	2	10	1
0	1	74	1	0	0	3	9	0
0	6	0	72	1	5	1	3	1
4	1	1	0	87	1	1	0	1
0	0	1	9	1	72	2	1	1
0	0	1	1	8	5	60	0	1
0	5	13	5	1	2	1	77	3
3	3	1	1	0	2	2	2	78

2.

87	1	0	0	6	1	3	0	0
0	59	6	7	2	1	2	5	2
0	6	74	3	0	1	2	10	1
0	5	1	69	2	6	1	4	2
1	3	1	1	88	0	4	1	0
0	1	0	7	1	74	2	0	2
2	0	0	2	4	1	86	2	0
1	6	7	2	0	1	2	73	3
1	0	2	2	0	0	2	4	80

3.

84	0	0	0	1	0	1	0	4
0	71	14	4	1	4	0	3	1
0	9	87	3	1	0	2	11	1
0	5	0	59	4	1	1	5	1
4	0	0	0	80	1	3	1	0
0	1	1	6	2	81	2	1	3
2	0	0	4	6	1	78	1	0
0	2	7	3	3	3	1	69	3
3	3	0	2	1	2	1	8	71

4.

76	0	0	0	3	0	0	0	3
0	78	10	4	2	1	0	5	4
0	3	67	0	0	1	1	10	0
0	3	5	73	4	3	1	5	1
2	2	0	1	87	1	5	0	0
0	2	1	6	2	60	0	1	7
1	0	3	0	1	3	78	0	4
0	3	9	7	2	0	1	78	2
5	4	4	2	0	0	1	6	84

5.

76	0	0	0	1	0	1	0	4
0	57	15	7	1	0	2	3	2
0	4	90	1	0	0	1	3	2
0	5	2	88	1	5	5	6	3
1	1	0	0	85	1	2	1	1
2	3	0	6	2	68	3	1	2
0	0	1	1	1	1	74	0	1

0 8 7 3 2 1 2 78 3  
8 1 2 1 0 2 0 8 69

6.

101 0 0 0 3 0 0 0 2  
0 64 16 7 2 0 0 1 1  
0 5 83 1 0 0 0 7 1  
0 6 0 75 2 4 0 0 0  
0 1 0 0 98 2 1 1 0  
1 1 0 9 0 79 3 0 5  
0 0 2 0 5 4 81 0 2  
0 6 6 1 0 2 0 62 6  
7 3 0 0 1 5 1 1 61

7.

102 0 0 0 2 0 2 1 3  
0 62 11 9 0 0 0 1 1  
0 4 88 2 0 0 0 9 1  
0 1 1 72 1 5 0 1 2  
0 0 0 2 84 0 8 0 0  
0 0 0 7 2 73 3 0 2  
2 0 1 1 5 3 80 0 2  
0 7 11 5 1 0 1 65 3  
3 5 1 1 1 2 3 8 65

8.

94 0 0 0 5 0 2 0 4  
0 75 21 6 1 1 2 10 1  
0 2 79 1 0 0 0 8 0  
0 4 1 57 1 7 1 2 0  
3 1 1 0 84 0 9 2 0

1	0	0	6	0	76	3	1	2
1	1	2	1	4	1	77	0	4
0	4	11	1	0	0	0	80	1
3	4	2	1	1	1	0	1	63

$M(i, j)$  = numarul de exemple din clasa  $i$  care au fost clasificate in clasa  $j$