ROB Laboratorium 5&6 - sprawozdanie

Mateusz Wasiak

1 czerwca 2020

Wstęp

Podczas tego zadania laboratoryjnego wykonałem implementację referencyjną wraz z lekkimi modyfikacjami w celu usprawnienia działania sieci. Pliki te mają standardowe nazwy takie jak dostarczone z zadaniem laboratoryjnym. Drugi punkt dotyczył wykonania poprawek podobnych do tych zawartych w artykule. Pliki stworzone do tej części laboratorium mają przedrostek *modified*. Stan generatora losowwego znajduje się w pliku *rndstate.txt*. Wywołanie wersji referencyjnej nie znajduje się w pliku mainscript. W tym celu należy wywołać *ann_training*. Analogicznie zmodyfikowana wersja wymaga wywyłania *modifiedAnn_training*

1 Referencyjna implementacja uczenia sieci

W trakcie wykonywania zadania laboratoryjnego zmodyfikowałem również funkcję *ann_training* by umożliwić przetwarzanie sekwencyjne.

```
terr = 0;
for i = 1:rows(tvec)
    [layerWeights singleTerr] = backprop(tvec(i, :),
    tlab(i, :), layerWeights, learningRate);
    terr += singleTerr;
endfor
terr /= rows(tvec);
```

1.1 Usprawnienia i eksperymenty

Zadanie laboratoryjne wymagało eksperymentowania na sieci. Wymogiem było pozostanie przy wstecznej propagacji błędu oraz pozostanie przy klasyfikacji przez w pełni połączone sieci. Do eksperymentów wykorzystałem zmodyfikowany plik *ann_training*. Wykonywałem testy dla różnej liczby warstw (od 1 do 4), oraz o różnej liczbie neuronów w warstwie (25, 50, 100, 150).

Najlepsze rezultaty otrzymałem dla dwóch warstw o liczbie neuronów w każdej warstwie 100. Ograniczyłem się do przedstawienia tylko macierzy pomyłek dla zbioru testowego, by sprawozdanie nie było zbyt obszerne.

1.2 Wyniki

Przedstawiona sieć osiągnęła najlepszy współczynnik błędu dla zbioru testowego w 48 epoce - 12.17% w tym samym momencie zbiór treningowy miał współczynnik błędu równy 8.64%.

Epoka	Czas trwania	Stopa błędu zb. tren.	Stopa błędu zb. test.
1	163.90036201477051	0.28189999999999998	0.28260000000000002
2	157.60338592529297	0.22068333333333334	0.2291
3	168.37966108322144	0.1797	0.1925
4	178.03525114059448	0.16096666666666667	0.1749
5	175.38025903701782	0.15176666666666666	0.16750000000000001
6	172.04319214820862	0.14574999999999999	0.16309999999999999
7	172.3576672077179	0.14044999999999999	0.1585
8	173.52324795722961	0.1363	0.15479999999999999
9	171.30492901802063	0.13343333333333332	0.15179999999999999
10	170.67542314529419	0.13031666666666666	0.14960000000000001
11	171.05622601509094	0.1278	0.1467
12	173.97120118141174	0.12556666666666666	0.14330000000000001
13	169.76893496513367	0.12358333333333334	0.14230000000000001
14	170.0781888961792	0.12171666666666667	0.14080000000000001
15	154.13210201263428	0.11963333333333333	0.1386
16	149.03255009651184	0.11771666666666666	0.13669999999999999
17	176.748694896698	0.11561666666666667	0.1361
18	150.42839503288269	0.11425	0.13500000000000001
19	162.40335202217102	0.11291666666666667	0.1343
20	158.78724598884583	0.11156666666666666	0.13389999999999999
21	150.09618592262268	0.11033333333333334	0.13250000000000001
22	161.64896392822266	0.10871666666666667	0.13220000000000001
23	160.11274909973145	0.1074	0.13189999999999999
24	151.79807901382446	0.10639999999999999	0.13159999999999999
25	160.23871397972107	0.10521666666666667	0.13070000000000001

Tabela 1: Wyniki działania podczas epok dla sieci [100, 100] learningRate = 0.01 Częśc 1.

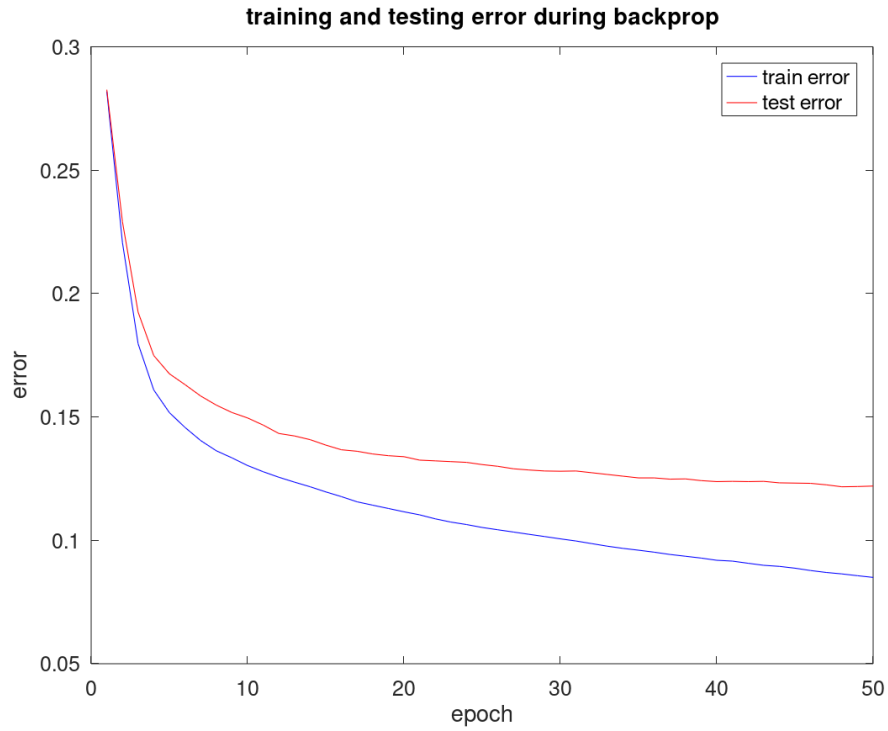
Epoka	Czas trwania	Stopa błędu zb. tren.	Stopa błędu zb. test.
26	161.04131507873535	0.10428333333333334	0.13
27	150.99919509887695	0.10336666666666666	0.129
28	159.88767290115356	0.10243333333333333	0.1285
29	143.35586905479431	0.10153333333333334	0.12809999999999999
30	141.55525398254395	0.10059999999999999	0.128
31	149.61800599098206	0.099733333333333327	0.12809999999999999
32	149.39557003974915	0.098683333333333331	0.12740000000000001
33	141.52372598648071	0.097633333333333336	0.12670000000000001
34	149.41447496414185	0.096750000000000003	0.126
35	148.58663415908813	0.096000000000000002	0.12529999999999999
36	141.54258012771606	0.095200000000000007	0.12529999999999999
37	149.92497611045837	0.09428333333333333	0.12479999999999999
38	148.82761096954346	0.093549999999999994	0.1249
39	142.07028698921204	0.092816666666666672	0.1242
40	150.10235118865967	0.09191666666666666	0.12379999999999999
41	149.67605090141296	0.091583333333333336	0.1239
42	141.60869407653809	0.090716666666666668	0.12379999999999999
43	149.39262509346008	0.089883333333333329	0.1239
44	149.20972514152527	0.089450000000000002	0.12330000000000001
45	141.65762591362	0.088716666666666666	0.1232
46	149.81429195404053	0.087766666666666673	0.1231
47	149.18013501167297	0.086983333333333329	0.1225
48	141.23463988304138	0.086400000000000005	0.1217
49	149.58968591690063	0.085666666666666669	0.12180000000000001
50	149.39906311035156	0.084983333333333327	0.122

Tabela 2: Wyniki działania podczas epok dla sieci [100, 100] learningRate = 0.01 Częśc 2.

Macierz pomyłek:

$$test\_cfmx = \begin{vmatrix} 817 & 0 & 14 & 53 & 2 & 3 & 102 & 0 & 9 & 0 & 0 \\ 1 & 961 & 2 & 24 & 4 & 0 & 6 & 0 & 2 & 0 & 0 \\ 11 & 0 & 767 & 17 & 112 & 1 & 88 & 0 & 4 & 0 & 0 \\ 15 & 8 & 9 & 910 & 23 & 0 & 31 & 0 & 4 & 0 & 0 \\ 0 & 0 & 63 & 45 & 822 & 0 & 66 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 941 & 0 & 36 & 1 & 21 & 0 \\ 110 & 0 & 66 & 46 & 81 & 0 & 677 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 & 0 & 963 & 1 & 20 & 0 \\ 1 & 0 & 3 & 8 & 4 & 1 & 6 & 5 & 972 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 1 & 41 & 0 & 950 & 0 \end{vmatrix}$$

Na podstawie macierzy pomyłek można zauważyć, że największy problem w klasyfikacji sprawiła sieci klasa 7. Najprawdopodobniej jest ona podobna do pozostałych klas (według macierzy pomyłek do 1 i 3 najbardziej).



Rysunek 1: Wykres przedstawiający stopę błędu dla dla sieci [100, 100] learningRate = 0.01

Powyżej znajduje się wykres 1 przedstawiający stopę błędu (przyjaźniejszą graficzną interpretację tabeli z wynikami)

2 Modyfikacja algorytmu uczenia

Po przeczytaniu dołączonego artykułu i przejrzeniu kilku dodatkowych materiałów wykonałem kilka ulepszeń:

2.1 Shuffling

Zastosowałem losowe przemieszczenie próbek zbioru uczącego:

```
setsNumber = rows(tvec);
indices = randperm(setsNumber);
tvec = tvec(indices, :);
tlab = tlab(indices, :);
```

2.2 Normalizacja wejścia

Wejście zostało znromalizowane względem wartości z zbioru uczącego

```
meanTvec = mean(tvec);
stdTvec = std(tvec);
stdTvec(stdTvec == 0) = 1;

tvec = bsxfun(@minus, tvec, meanTvec);
tvec = bsxfun(@rdivide, tvec, stdTvec);

meanTstv = mean(tstv);
stdTstv = std(tstv);
stdTstv(stdTstv == 0) = 1;

tstv = bsxfun(@minus, tstv, meanTstv);
tstv = bsxfun(@rdivide, tstv, stdTstv);
```

2.3 Funkcja aktywacji

Po przeczytaniu kilku dodatkowych materiałów zauważyłem, że w najnowszych artykułach najczęściej występuje funkcja ReLU jako funkcja aktywacji. Uznałem, że podejmę się wykorzystania tej właśnie funkcji w celu ulepszenia działania sieci, zamiast proponowanej w zamieszczonym artykule funkcji *tanh*. Niestety nie udało mi się osiągnąć lepszych wyników niż dla rozwiązania referencyjnego. Osiągnąłem w najlepszym przypadku 13.4% błędów. Wykorzystałem zatem metodę przedstawioną w artykule. Najlepsze wyniki osiągnąłem dla stałej o wartośći 0.001. Poniższy kod przedstawia implemetację actf i actdf dla wybranej funkcji.

```
function res = modifiedActf(sfvalue)
        res = 1.7159 * tanh(2/3 * sfvalue) + 0.0001*sfvalue;
endfunction

function res = modifiedActdf(sfvalue)
    res = 1.14393 * (sech(2/3 * sfvalue)).^2 + 0.0001;
endfunction
```

2.4 Wyniki

Przedstawiona sieć z naniesionymi modyfikacjami osiągnęła najlepszy współczynnik błędu dla zbioru testowego w 48 epoce - 12.37% w tym samym momencie zbiór treningowy miał współczynnik błędu równy 7.92%.

Epoka	Czas trwania	Stopa błędu zb. tren.	Stopa błędu zb. test.
1	137.76694798469543	0.17173333333333332	0.18079999999999999
2	137.99308395385742	0.15373333333333333	0.16700000000000001
3	134.57798790931702	0.14323333333333332	0.1573
4	131.95930480957031	0.13668333333333332	0.15179999999999999
5	131.97356510162354	0.13191666666666665	0.1487
6	131.03740692138672	0.12855	0.1464
7	131.24426198005676	0.12553333333333333	0.14319999999999999
8	131.67844414710999	0.12286666666666667	0.1414
9	131.1017050743103	0.12111666666666666	0.1399
10	131.00423789024353	0.11895	0.13850000000000001
11	131.1301589012146	0.11715	0.13689999999999999
12	130.93589806556702	0.11558333333333333	0.1366
13	131.88487505912781	0.11398333333333334	0.13600000000000001
14	135.39777588844299	0.11235000000000001	0.13569999999999999
15	133.0634400844574	0.11068333333333333	0.1351
16	127.80086898803711	0.10935	0.13469999999999999
17	128.66137385368347	0.10828333333333333	0.1336
18	128.11967206001282	0.10695	0.1328
19	127.86915588378906	0.10539999999999999	0.13220000000000001
20	128.18963813781738	0.10448333333333333	0.13159999999999999
21	128.23892498016357	0.10326666666666667	0.13120000000000001
22	134.17125988006592	0.10249999999999999	0.13059999999999999
23	135.7961368560791	0.10128333333333334	0.13
24	126.59308385848999	0.10050000000000001	0.12959999999999999
25	126.03853893280029	0.099900000000000003	0.1285

Tabela 3: Wyniki działania podczas epok dla zmodyfikowanej sieci [100, 100] learningRate = 0.001 Częśc 1.

Epoka	Czas trwania	Stopa błędu zb. tren.	Stopa błędu zb. test.
26	126.3236517906189	0.098599999999999993	0.1293
27	126.35890102386475	0.097616666666666671	0.12870000000000001
28	125.92740893363953	0.096600000000000005	0.12859999999999999
29	126.02317905426025	0.095366666666666669	0.12859999999999999
30	125.85780811309814	0.094549999999999995	0.12809999999999999
31	125.79930591583252	0.093683333333333327	0.128
32	125.99260807037354	0.092633333333333331	0.12790000000000001
33	125.87819194793701	0.091766666666666663	0.12759999999999999
34	126.04745697975159	0.090466666666666667	0.128
35	126.09811091423035	0.08981666666666667	0.12770000000000001
36	125.76131916046143	0.088733333333333331	0.1273
37	126.41528606414795	0.087866666666666662	0.12659999999999999
38	126.19193005561829	0.087133333333333327	0.12620000000000001
39	126.0334198474884	0.086266666666666672	0.12570000000000001
40	126.04962110519409	0.085466666666666663	0.12540000000000001
41	126.02522706985474	0.084566666666666665	0.12540000000000001
42	125.77459597587585	0.083616666666666672	0.12479999999999999
43	126.23216390609741	0.083000000000000004	0.12479999999999999
44	125.94445896148682	0.082049999999999998	0.125
45	126.47796392440796	0.081216666666666673	0.1246
46	126.8427369594574	0.080716666666666673	0.124
47	125.96741509437561	0.079850000000000004	0.1242
48	126.06781411170959	0.079200000000000007	0.1237
49	130.4129650592804	0.078516666666666665	0.1239
50	124.82935500144958	0.07771666666666667	0.1241

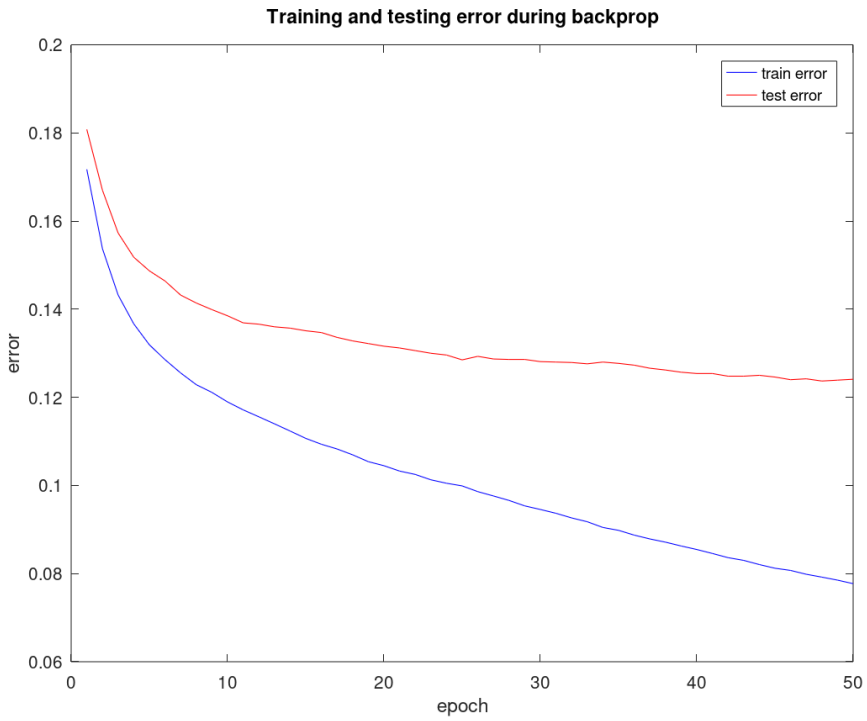
Tabela 4: Wyniki działania podczas epok dla zmodyfikowanej sieci [100, 100] learningRate = 0.001 Częśc 2.

Macierz pomyłek:

$$train\_cfmx = \begin{vmatrix} 5302 & 1 & 37 & 133 & 26 & 3 & 466 & 0 & 32 & 0 & 0 \\ 11 & 5853 & 12 & 98 & 10 & 2 & 11 & 0 & 3 & 0 & 0 \\ 78 & 2 & 5012 & 64 & 624 & 4 & 210 & 0 & 6 & 0 & 0 \\ 99 & 14 & 31 & 5647 & 109 & 1 & 92 & 0 & 6 & 1 & 0 \\ 14 & 6 & 200 & 181 & 5433 & 0 & 151 & 0 & 15 & 0 & 0 \\ 1 & 0 & 2 & 1 & 0 & 5906 & 0 & 72 & 4 & 14 & 0 \\ 432 & 9 & 257 & 117 & 424 & 0 & 4724 & 0 & 36 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 & 57 & 0 & 5834 & 9 & 97 & 0 \\ 20 & 2 & 14 & 28 & 23 & 7 & 40 & 16 & 5848 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 18 & 0 & 200 & 4 & 5778 & 0 \end{vmatrix}$$

$$test\_cfmx = \begin{vmatrix} 821 & 0 & 12 & 31 & 5 & 2 & 118 & 0 & 11 & 0 & 0 \\ 3 & 958 & 5 & 25 & 5 & 0 & 3 & 0 & 1 & 0 & 0 \\ 23 & 1 & 758 & 11 & 133 & 1 & 72 & 0 & 1 & 0 & 0 \\ 24 & 7 & 13 & 903 & 21 & 1 & 27 & 0 & 4 & 0 & 0 \\ 0 & 0 & 66 & 39 & 841 & 2 & 47 & 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 947 & 0 & 27 & 4 & 20 & 0 \\ 116 & 0 & 75 & 30 & 102 & 0 & 666 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 23 & 0 & 961 & 0 & 16 & 0 \\ 5 & 0 & 5 & 8 & 5 & 4 & 9 & 7 & 957 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 1 & 46 & 0 & 947 & 0 \end{vmatrix}$$

Wykres 2 przedstawia współczynnik błędu dla przygotowanej sieci (przyjaźniejszą graficzną interpretację tabeli z wynikami) dla zbioru treningowego i testowego. Analogicznie dla rozwiązania referencyjnego najwięcej błędów wystąpiło dla klasy 7. Niestety każda konfiguracja sieci osiągała wyniki gorsze niż w rozwiązaniu referencyjnym. (Wykonywałem testy dla współczynników uczenia (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1) Architektura sieci o dwóch warstwach w kazdej po 100 neuronów również zwracała najlepsze wyniki.



Rysunek 2: Wykres przedstawiający współczynnik błędu dla dla sieci [100, 100] learningRate = 0.001