

Sprawozdanie z ćwiczenia:

Budowa i działanie sieci wielowarstwowej

1. Cel ćwiczenia

Celem ćwiczenia było poznanie budowy i działania wielowarstwowych sieci neuronowych poprzez uczenie z użyciem algorytmu wstecznej propagacji błęd rozpoznawania konkretnych liter alfabetu.

2. Wstęp teoretyczny

Algorytm uczenia sieci – algorytm propagacji wstecznej, polega na takim dobraniu wag sygnałów wejściowych każdego neuronu w każdej warstwie, aby wartość błędu dla kolejnych par uczących zawartych w zbiorze uczącym była jak najmniejsza. Schemat krokowy algorytmu przedstawia się w następujący sposób:

1. Ustalamy topologię sieci, tzn. liczbę warstw, liczbę neuronów w warstwach.
2. Inicjujemy wagi losowo.
3. Dla danego wektora uczącego obliczamy odpowiedź sieci (warstwa po warstwie).
4. Każdy neuron wyjściowy oblicza swój błąd, oparty na różnicy pomiędzy obliczoną odpowiedzią y oraz poprawną odpowiedzią t .
5. Błędy propagowane są do wcześniejszych warstw.
6. Każdy neuron modyfikuje wagi na podstawie wartości błędu i wielkości przetwarzanych w tym kroku sygnałów.
7. Powtarzamy od punktu 3. dla kolejnych wektorów uczących.
8. Zatrzymujemy się, gdy średni błąd na danych treningowych zostanie przyjęty jako optymalny.

3. Wykonanie zadania

- Wygenerowano dane uczące i testujące, zawierające 20 wielkich liter alfabetu łacińskiego w postaci dwuwymiarowej tablicy 7x5
- Zaimplementowano sieć neuronową.
- Uczono sieci modyfikując współczynnik uczenia oraz współczynnik bezwładności.
- Testowano poprawność działania sieci, sprawdzając zmianę błędu z każdą epoką, oraz próby rozpoznania liter ze szczególnym zwróceniem uwagi na podobieństwo między literami.

→ Program otrzymuje matryce liter, ilość epok oraz współczynnik uczenia. Na tej podstawie program zaczyna „się uczyć”, a po jego zakończeniu następuje testowanie sieci pod kątem zestawu wybranych liter. Działa to na zasadzie porównywania danych testowych z „nauczonymi” oraz zwracania odpowiedniej wartości dla poszczególnych liter.

→ Do 35 wejść odpowiadającym znakom, które reprezentują literę, są przypisywane losowe wagi początkowe. Po wymnożeniu tych wartości, wyliczona zostaje na ich podstawie wartość funkcji aktywacji która podana dalej stanowi wejście dla kolejnych warstw. Warstwa końcowa otrzymuje na wejściu wartości wyjściowe z warstwy ukrytej, następnie po wyznaczeniu wartości funkcji aktywacji wyznaczany jest błąd na podstawie wartości otrzymanej oraz wartości spodziewanej. Następnie dzięki algorytmowi wstecznej propagacji modyfikowane są wagi w warstwie ukrytej.

4. Wyniki:

Działanie programu sprawdzono na zestawie uczącym składającym się z liter A,B,C,D,E,F,G,H,I,J,a,b,c,d,e,f,g,h,i,j oraz zestawu testowego składającego się z liter A,B,C,F,G,a,b,e,i,j

Gdzie zapis liter w formie 0 i 1 wygląda przykładowo:

Litera A :

0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1

Litera B:

1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0

Litera F:

1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0

Wydruk programu zawierał wynik rozpoznawania danej litery. Składał się na niego wynik porównania do litery podanej do testu, jak dwóch dowolnie innych liter, aby potwierdzić poprawność wyniku.

Przykładowy wynik działania programu:

```
Litera A:
Porównanie do litery A Wynik: 0,9725574
Porównanie do litery i Wynik: -0,005273254
Porównanie do litery a Wynik: 0,004080513

Litera B:
Porównanie do litery B Wynik: 0,9538145
Porównanie do litery i Wynik: -0,0006573308
Porównanie do litery a Wynik: 0,004230229

Litera C:
Porównanie do litery C Wynik: 0,9650047
Porównanie do litery i Wynik: -0,0009752001
Porównanie do litery a Wynik: 0,00405529
```

```
Litera i:
Porównanie do litery i Wynik: 0,9716548
Porównanie do litery b Wynik: 0,00235628
Porównanie do litery a Wynik: 0,004645573

Litera a:
Porównanie do litery a Wynik: 0,9607296
Porównanie do litery b Wynik: -0,006961307
Porównanie do litery j Wynik: 0,004054197

Litera b:
Porównanie do litery b Wynik: 0,9569462
Porównanie do litery a Wynik: 0,0005991581
Porównanie do litery j Wynik: -0,0007345995
```

→ **Wpływ współczynnika uczenia na działanie sieci:**

Jako pierwszy, przetestowano wpływ czynnika uczenia się na działanie sieci.

Przeprowadzono testy dla czynników: 0.0003, 0.003, 0.03, 0.1

Parametry testu: 50 neuronów w warstwie ukrytej, 1000 epok

Współczynnik	0,0003		Współczynnik	0,003
Litera	Wynik		Litera	Wynik
A	0,7288316		A	0,8534140
B	0,3580596		B	0,8754204
C	0,5471140		C	0,9265290
F	0,4782749		F	0,8680400
G	0,4362798		G	0,4216510
j	0,5582035		j	0,8870566
i	0,6808268		i	0,7954872
a	0,5994798		a	0,8265480
b	0,5089510		b	0,6859621
e	0,5591820		e	0,9035285
Średnia:	0,5455203		Średnia:	0,8043637

Współczynnik	0,03		Współczynnik	0,1
Litera	Wynik		Litera	Wynik
A	0,9682270		A	0,9786480
B	0,9713381		B	0,9846510
C	0,9773825		C	0,9891560
F	0,9598167		F	0,9912350
G	0,9718974		G	0,9853210
j	0,9659587		j	0,9789560
i	0,9699287		i	0,9864520
a	0,6954321		a	0,9789490
b	0,9698108		b	0,9876580
e	0,9773387		e	0,9876840
Średnia:	0,9427131		Średnia:	0,9848710

Na podstawie wyników zawartych w tabelach widać, że najsłabsze wyniki były przy współczynniku uczenia wynoszącym 0,0003. Wynosiły około 0,5.

Gdy współczynnik rósł (0,003), wyniki też rosły. Najbliższe jedynce były wyniki przy współczynniku 0,1.

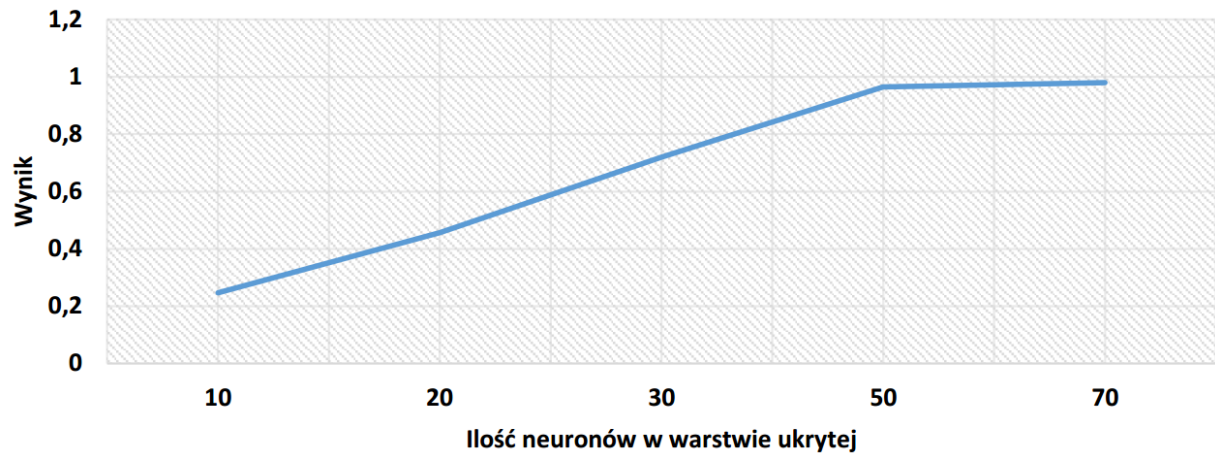
→ Wpływ ilości neuronów warstwie ukrytej na działanie sieci, przy współczynniku uczenia = 0,1 i 1000 epok:

Ilość neuronów	10		Ilość neuronów	20
Litera	Wynik		Litera	Wynik
A	0,335		A	0,363
B	0,133		B	0,395
C	0,292		C	0,406
F	0,240		F	0,421
G	0,163		G	0,465
j	0,247		j	0,426
i	0,268		i	0,883
a	0,190		a	0,303
b	0,247		b	0,434
e	0,352		e	0,467
Średnia:	0,2466		Średnia:	0,4563

Ilość neuronów	50		Ilość neuronów	70
Litera	Wynik		Litera	Wynik
A	0,967		A	0,967
B	0,961		B	0,961
C	0,968		C	0,968
F	0,968		F	0,968
G	0,964		G	0,964
j	0,958		j	0,958
i	0,967		i	0,967
a	0,966		a	0,966
b	0,968		b	0,968
e	0,967		e	0,967
Średnia:	0,9654		Średnia:	0,9798

Im więcej neuronów w ukrytej warstwie, tym lepsze wyniki. Jednak powyżej 50 dalszy wzrost poprawy wyników jest nieznaczny.

Wpływ ilości neuronów warstwy ukrytej na poprawność wyników



Wnioski:

- Wraz ze wzrostem współczynnika uczenia na przedziale $\{0.0003, 0,1\}$ szybkość uczenia wyraźnie rośnie.
- Ilość neuronów w warstwie ukrytej ma duże znaczenie podczas uczenia sieci, ponieważ przy zbyt małej ilości neuronów, sieć nie była w stanie rozpoznać żadnej litery. Efektywność rozpoznawania liter wraz ze wzrostem ilości neuronów do pewnego momentu rośnie w sposób zbliżony do liniowego

Źródła/literatura:

- http://edward_ch.republika.pl/sneuro.html,
<http://web.archive.org/web/20160309093826/http://ac-it.pl/algorytm-wstecznejpropagacji-bledow> ,
http://edward_ch.republika.pl/sneuro.html - opis i sposób działania algorytmu
- <http://www.ai.c-labtech.net/sn/litery.html> - matryce liter
- https://www.youtube.com/watch?v=L_PByyJ9g-I – implementacja algorytmu