Tomasz Lenart, gr. proj. 2, rok 3, IS

Budowa i działanie sieci jednowarstwowej

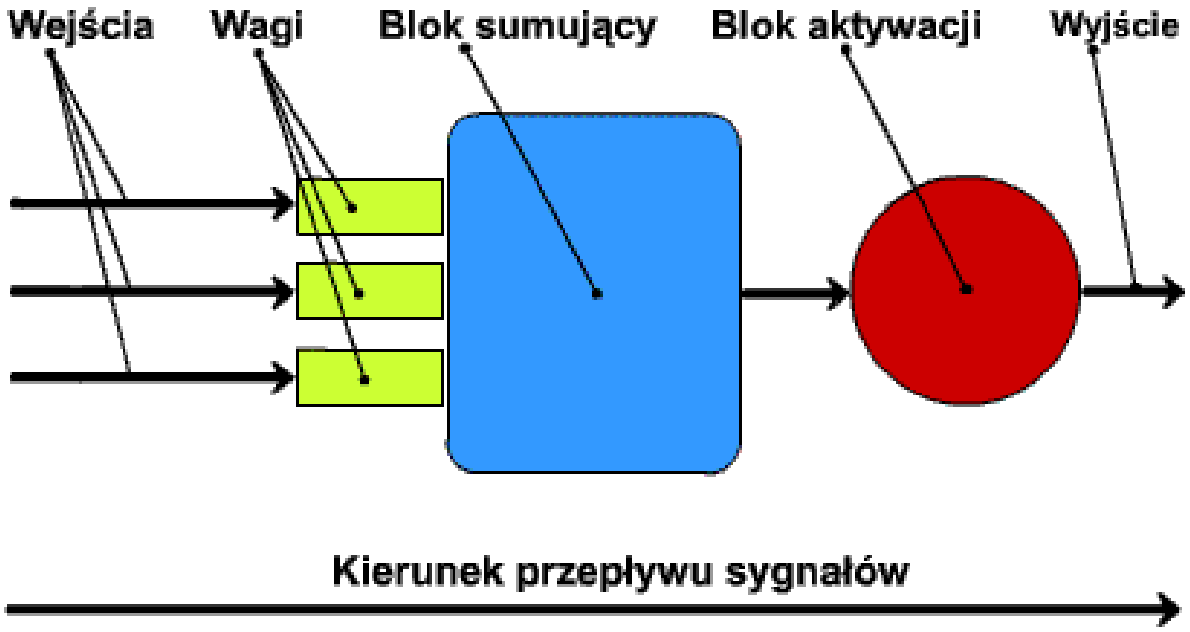
Cel ćwiczenia:

Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter

Sieć neuronowa (sztuczna sieć neuronowa) – ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu.

Czasem nazwą sztuczne sieci neuronowe określa się interdyscyplinarną dziedzinę wiedzy zajmującą się konstrukcją, trenowaniem i badaniem możliwości tego rodzaju sieci.

Schemat budowy sieci:



Ogólny opis budowy i działania sieci:

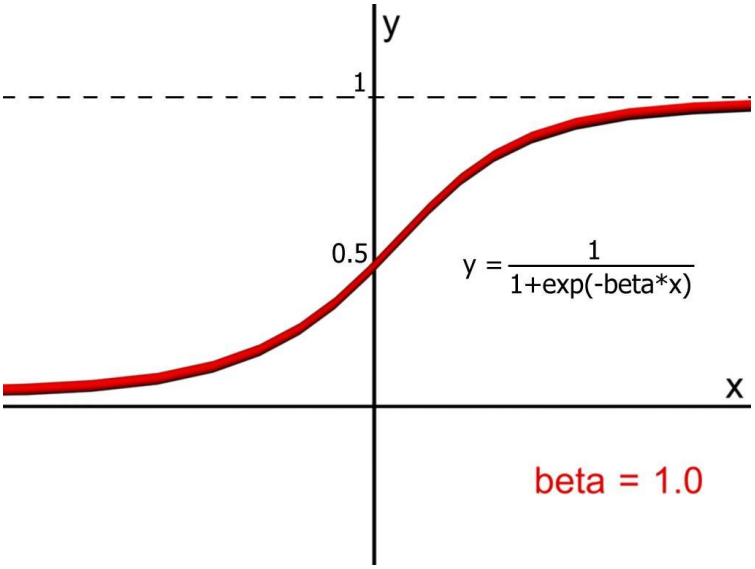
Wejścia dostarczają sygnał, który następnie jest mnożony przez współczynniki wag, następnie w bloku sumowania następuje sumowanie pomnożonych sygnałów. Wynikiem tego otrzymujemy sygnał zwany potencjałem membranowym. Następnie sygnał przetworzony zostaje w bloku aktywacji, który w zależności od potrzeb może być opisany różnymi funkcjami - zwanymi funkcjami aktywacji. Wartość funkcji aktywacji jest sygnałem wyjściowym neuronu i propagowana jest do neuronów warstwy następnej. Funkcja aktywacji przybiera jedną z trzech postaci:

- Skoku jednostkowego – tzw. Funkcja progowa
- Liniowa
- Nieliniowa

W tym algorytmie zastosowano funkcję sigmoidalną w postaci:

$$y(x) = \frac{1}{1 + e^{-\beta x}}$$

Przebieg tej funkcji wygląda następująco:



W przypadku uczenia rozpoznawania liter, percepton obsługuje 35 wejść, ponieważ tyle znaków zawiera matryca pojedynczej litery. Każde z wejść ma przypisaną wagę, która na początku jest generowana losowa z przedziału {-0.5, 0.5}.

Algorym zastosowany w projekcie opiera się na modyfikacji wag każdego z wejść. Wagi modyfikowane są do momentu w którym wartość zwracana przez percepton jest możliwie najbliższa wartości oczekiwanej.

Algorytm:

1. Losowe inicjowanie wag
2. Obliczanie wartości zwracanej przez percepton poprzez blok sumujący i funkcję aktywacji
3. Sprawdzenie poprawności zwroconej wartości – w przypadku błędnego wyniku modyfikacja wag w następujący sposób:

$$w1 += \mathrm{n} * (y - yn) * x1$$

$$w2 += \mathrm{n} * (y - yn) * x2$$

...

$$B += \mathrm{n} * (y - yn) * yn$$

w1,w2...-wagi

$n - współczynk\ uczenia$

x1,x2...-wartości na wejściu neuron

y- spodziewana wartość na wyjściu

yn- rzeczywista wartość na wyjściu

yn- rzeczywista wartość na wyjściu

Wyniki:

Do sprawdzenia poprawności działania neuronu wykorzystano małe i duże litery. Testy zostały przeprowadzone dla dwóch małych i dwóch dużych liter – a,D,E,n - z następującymi współczynnikami uczenia:

- 0,0001
- 0,001
- 0,01
- 0,1
- 0,5
- 0,8

Dla każdego wspólczynnika przeprowadzono 10 prób. Wyniki zebrano w tabelach.

Współczynnik 0,0001		Współczynnik 0,001	
L.p.	% poprawnych	L.p.	% poprawnych
1	100	1	100
2	100	2	75
3	75	3	75
4	75	4	50
5	50	5	75
6	75	6	100
7	75	7	100
8	75	8	50
9	50	9	75
10	50	10	75
Średnia:	72,5	Średnia:	77,5

Współczynnik 0,01		Współczynnik 0,1	
L.p.	% poprawnych	L.p.	% poprawnych
1	75	1	100
2	100	2	100
3	75	3	100
4	100	4	100
5	75	5	100
6	100	6	100
7	100	7	100
8	100	8	100
9	100	9	100
10	100	10	100
Średnia:	92,5	Średnia:	100

Współczynnik 0,5		Współczynnik 0,9	
L.p.	% poprawnych	L.p.	% poprawnych
1	100	1	100
2	100	2	100
3	100	3	100
4	100	4	100
5	100	5	100
6	100	6	100
7	100	7	100
8	100	8	100
9	100	9	100
10	100	10	100
Średnia:	100	Średnia:	100

Po podsumowaniu wyników otrzymano następujący wykres.

Wykres 1.

Następnie dla dwóch wybranych liter, ‘n’ oraz ‘A’ odnotowano po 3 dokładne wyniki wartości, jakie zwracała sieć, aby lepiej zobrazować wyżej otrzymane wyniki.

Spodziewane wartości które powinien zwrócić percepton to:

- 0 – dla małej litery
- 1 – dla dużej litery

Otrzymane wyniki:

- Litera E

litera 'E'			
Współczynnik 0,0001		Współczynnik 0,001	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,52	1	0,58
2	0,77	2	0,53
3	0,54	3	0,55
Średnia	0,51	Średnia	0,55
Współczynnik 0,01		Współczynnik 0,1	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,74	1	0,65
2	0,59	2	0,71
3	0,39	3	0,63
Średnia	0,57	Średnia	0,66
Współczynnik 0,5		Współczynnik 0,8	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,71	1	0,98
2	0,97	2	0,77
3	0,72	3	0,9
Średnia	0,80	Średnia	0,88

- Litera n

litera 'n'			
Współczynnik 0,0001		Współczynnik 0,001	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,42	1	0,27
2	0,39	2	0,49
3	0,36	3	0,34
Średnia	0,39	Średnia	0,37
Współczynnik 0,01		Współczynnik 0,1	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,19	1	0,2
2	0,18	2	0,19
3	0,48	3	0,22
Średnia	0,28	Średnia	0,20
Współczynnik 0,5		Współczynnik 0,8	
L.p	Zwrócona wartość:	L.p	Zwrócona wartość:
1	0,01	1	0,001
2	0,05	2	0,006
3	0,04	3	0,08
Średnia	0,03	Średnia	0,03

Przykładowy wynik działania programu

- **Litera E**
  - Współczynnik 0,1

```
11111
10000
10000
11110
10000
10000
11111
 WYNIK: 0.580291723488428: duża litera.
```

- Współczynnik 0,5

```
11111
10000
10000
11110
10000
10000
11111
 WYNIK: 0.9342165276045956: duża litera.
```

- Współczynnik 0,8

```
11111
10000
10000
11110
10000
10000
11111
 WYNIK: 0.9894004049331986: duża litera.
```

- **Litera n**
  - Współczynnik:0,1

```
00000
00000
00000
10100
01010
01010
01010
 WYNIK: 0.42420668900090036: mała litera.
```

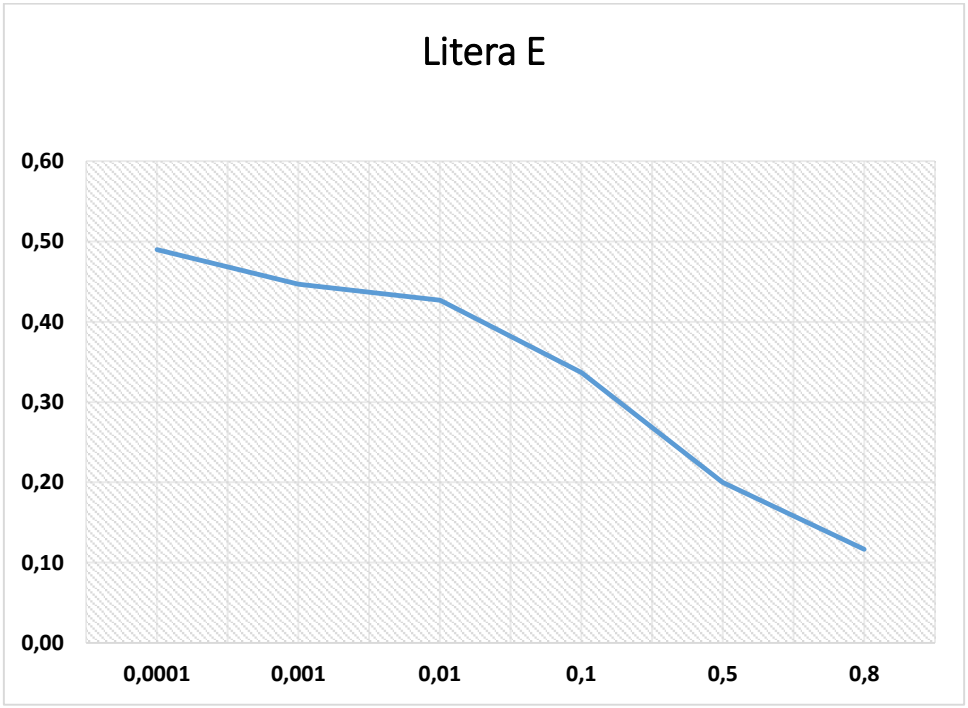
  - Współczynnik: 0,5

```
00000
00000
00000
10100
01010
01010
01010
 WYNIK: 0.03842029208307261: mała litera.
```

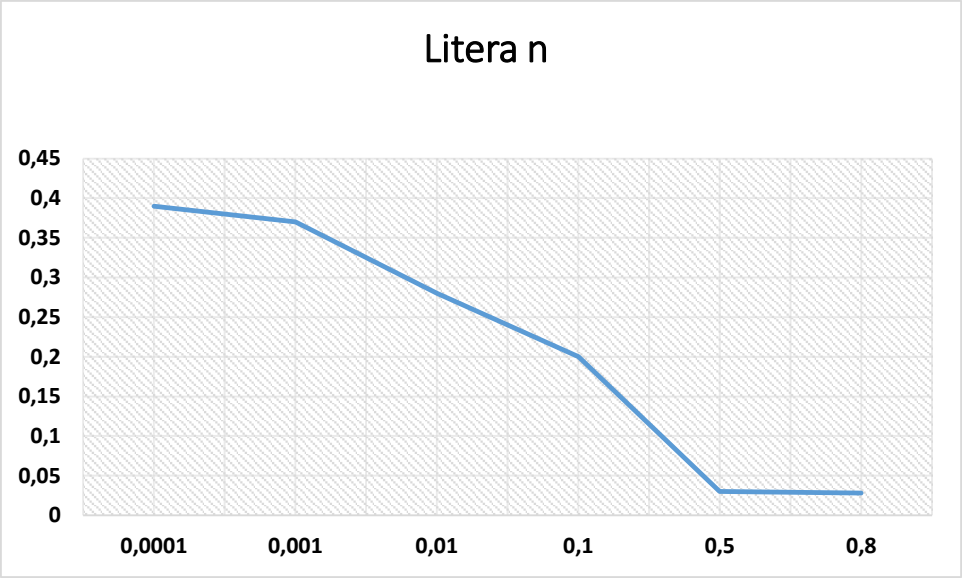
  - Współczynnik: 0,8

```
00000
00000
00000
10100
01010
01010
01010
 WYNIK: 0.0022877820631073964: mała litera.
```

Wykresy obrazujące wartość błędu pomiędzy wnikiem preceptonu a wartością oczekiwaną w zależności od współczynników uczenia:



Wykres 2.



Wykres 3.

Na koniec przetestowano również poprawność działania w zależności od ilości danych uczących. Testy przeprowadzono pięć razy dla 5 oraz 10 liter. Działanie programu sprawdzane było na tych samych czterech literach co w pierwszym przypadku. Zastosowano wspolczynnik: 0.1.

Otrzymano następujące wyniki:

Liczba danych wejściowych: 5	
L.p.	% poprawnych
1	75
2	100
3	50
4	75
5	50
Średnia	70

Liczba danych wejściowych: 10	
L.p.	% poprawnych
1	75
2	75
3	100
4	75
5	50
Średnia	75

Wykres przedstawiający procent poprawnych wyników w zależności od ilości danych.

Wykres 3

Omówienie wyników i wnioski:

Działanie sztucznego neuronu zalezy od wielu czynników, takich jak:

- Działanie sztucznego neuronu zależy od wielu czynników, takich jak: wartości współczynnika uczenia, ilości danych uczących oraz przydzielanych losowo wag.
- Z wykresu 1 wnioskujemy, że współczynnik uczenia oscylujący w przedziale {0.1 , 0.9} daje wyniki w granicach 100% poprawności. Współczynniki poniżej 0.1 dają dużo gorsze wyniki. Podsumowując, ze wzrostem współcznnika ilość poprawnych wyników rośnie.
- Z wykresu 2 widzimy jak współczynnik uczenia wpływa na wyniki. Wartości dla współczynnika 0,0001 są bardzo bliskie granicy błędu, stąd największa częstotliwość niepoprawnych wartości. Zauważamy również, że im mniejszy współczynnik tym zakres błędu jest dużo większy. Przykładowo, dla współczynnika 0,0001 maksymalna rozpiętość błędu wynosi około 0,2, a dla współczynnika 0,8 wynosi ona jedynie kilka setnych. Ponownie zauważamy, że większy współczynnik gwarantuje poprawniejsze wyniki.
- Wykres 3 przedstawia zalezność ilości uczacych danych od procentu poprawnych wyników. Jednoznacznie możemy stwierdzić, że zbyt mała ilość uczących danych daje znacznie gorsze wyniki.
- Każda próba testowa była przynajmniej kilkukrotnie powtarzana, ponieważ wszystkie wyniki uczenia są zależne od losowo przydzielanych na początku wag, dlatego też mogą zdarzać się niewielkie rozbieżności w otrzymywanych wynikac

*W projekcie został zastosowany tylko ten algorytm który został zaprezentowany na wykładzie.

Źródła:

- Notatki z wykładów
- http://users.pja.edu.pl/~msyd/wyk-nai/multiClassNN-pl.pdf
- http://www.cs.put.poznan.pl/jstefanowski/aed/TPDANN.pdf
- https://en.wikipedia.org/wiki/Artificial_neural_network

Przykładowy wynik działania programu:

```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (17 lis 2017, 01:33:43)
01110
10001
11110
10000
01110
 WYNIK: 0.25657572762940484: mała litera.

11110
10001
10001
10001
10001
10001
11110
 WYNIK: 0.8967324608135596: duża litera.

00000
00000
01110
00001
01111
10001
01111
 WYNIK: 0.32949381426646346: mała litera.

11111
10000
10000
11110
10000
10000
11111
 WYNIK: 0.8740532225918989: duża litera.

00000
00000
00000
10100
01010
01010
01010
 WYNIK: 0.42410140481577496: mała litera.
```

Fragmenty kodu programu:

- Przypisiwanie wag – losowych oraz nowych na podstawie otrzymanych wyników

```
private void losujWagi() {
    Random r = new Random();
    for (int i = 0;i <= rozmiarLitery; ++i) {
        wagi[i] = r.nextDouble() - 0.5;
        System.out.println("Waga[" + i + "] = " + wagi[i]);
    }
}

private void aktualizujWagi(double error, int letter[]) {
    for (int i = 0; i < rozmiarLitery; ++i) {
        wagi[i] += error * wspolczynnikUczenia * letter[i];
    }
    wagi[rozmiarLitery] = error * wspolczynnikUczenia; // bias
}
```

- Fukncja ucząca:

```
private void learn() {
    int epoch = 0;
    while (epoch < 10000) {
        for (int i = 0; i < iloscLiter; ++i) {
            result = sum(x[i]);

            error = y[i] - result;

            if (error * error > 0.25) {
                adjustWeights(error, x[i]);
            }
        }
        ++epoch;
    }
}
```

- Funkcja sumująca:

```
private double sum(int x[]) {
    double sum = 0;
    for (int i = 0; i < rozmiarLitery; ++i) {
        sum += x[i] * wagi[i];
    }
    return 1/(1 + Math.exp(-sum));
}
```