Krzysztof Kaflowski

Podstawy Sztucznej Inteligencji

IS

Rozpoznawanie dużych liter

1. Celem projektu było stworzenie wielowarstowej sieci neuronowej, która potrafi rozpoznać konkretne litery alfabetu zawarte w ASCII.

**2a)**

Sieć neuronowa (sztuczna sieć neuronowa) – ogólna nazwa [struktur matematycznych](https://pl.wikipedia.org/wiki/Struktura_matematyczna) i ich programowych lub sprzętowych modeli, realizujących obliczenia lub [przetwarzanie sygnałów](https://pl.wikipedia.org/wiki/Przetwarzanie_sygna%C5%82%C3%B3w) poprzez rzędy elementów, zwanych [sztucznymi neuronami](https://pl.wikipedia.org/wiki/Sztuczny_neuron), wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa [naturalnych neuronów](https://pl.wikipedia.org/wiki/Neuron), łączących je [synaps](https://pl.wikipedia.org/wiki/Synapsa), oraz [układów nerwowych](https://pl.wikipedia.org/wiki/Uk%C5%82ad_nerwowy), w szczególności [mózgu](https://pl.wikipedia.org/wiki/Kresom%C3%B3zgowie).

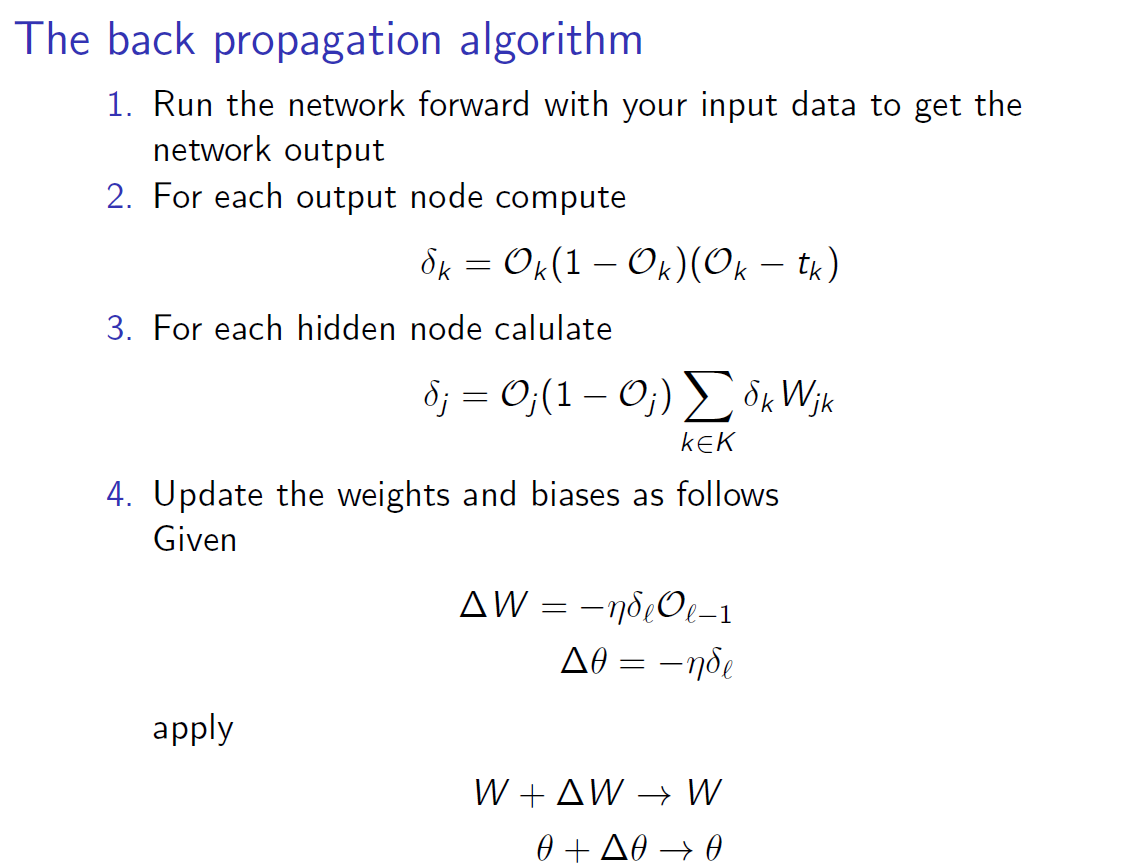
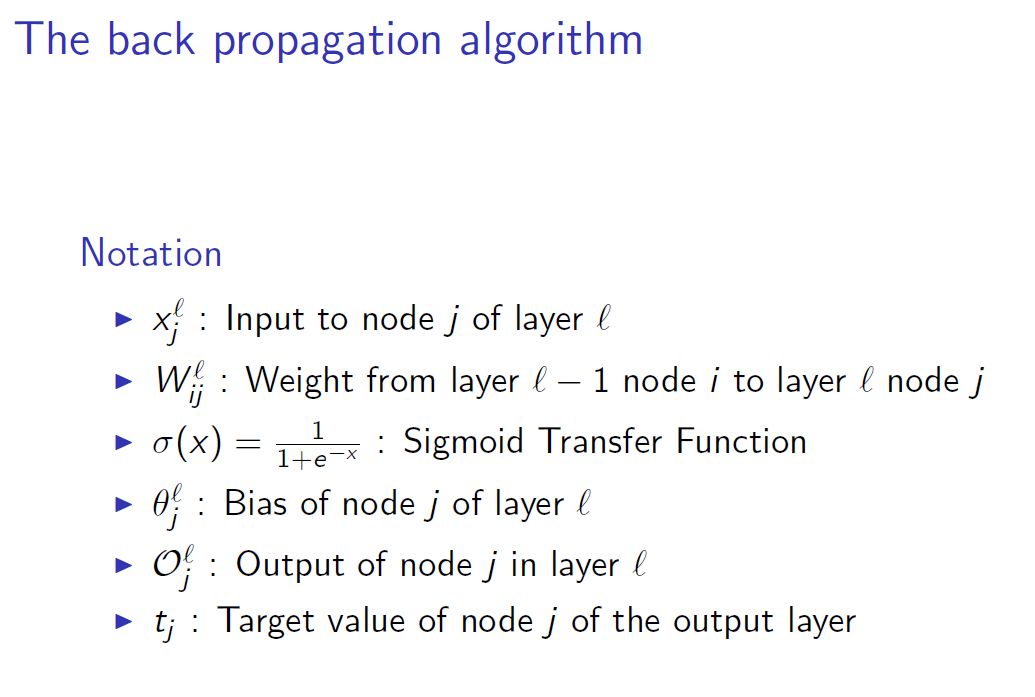


Problem rozwiązano za pomocą trzech warstw. Warstwa wejściowa posiada 35 neuronów. Warstwa ukryta posiada 10 neuronów a warstwa wyjściowa posiada 26 neuronów, ponieważ tyle właśnie jest liter w alfabecie. Każdy neuron warstwy ukrytej otrzymuje litere w postaci siatki 5x7 w reprezentacji zero-jedynkowej uzyskanej z warstwy wejściowej. W celu obliczenia błędu oraz poprawiania wag został użyty algorytm wstecznej propagacji błędu.

Algorytm wstecznej propoagacji błędu:

Bardzo dobrze został opisany w prezentacji, którą zamieszciłem w repozytorium projektu pod linkiem:

Krótki opis zawartości powyższej prezentacji i metody działania algorytmu:



Metoda wstecznej propagcji polega na tym, że po przepuszczeniu danych przez wszystkie warstwy, błąd jest obliczany w ostatniej warstwie na podstawie odpowiedzi sieci neuronowej oraz oczekiwanej wartości oraz przemnożony przez pochodną funkcji aktywacji. Następnie błąd w każdej poprzedniej warstwie jest obliczany na podstawie błędu w warstwie wyżej oraz wag warstwy niżej. Jest to suma ważona wag i błędu przemnożona przez pochodną funkcji aktywacji od sumy ważonej z porpzedniej iteracji. Tak jak jest to zapisane w punkcie 2 i 3 powyższego slajdu dla warstwy ukrytej oraz warstwy wyjścia.

Poprawa wag polega na dodaniu to wag obliczonej wartości współczynnika uczenia przemnożonego przez błąd przemnożonego przez odpowiedz neuronu z odwróconym znakiem.

Funkcja aktywacji w tym przypadku to funkcja sigmoidalna o wzorze:

Pochodna funkcji sigmoidalnej po przekształceniach wynosi:

Zaletą funkcji sigmoidalnej jest to, że jej pochodna ma tak łatwą do obliczenia postać po przekształceniu.

Całość przekształcenia znajduje się w prezentacji w pdf pod linkiem umieszczonym wyżej.

W celu zaimplementowania sieci neuronowej wykorzystano bibliotekę Neuroph, która posiada wbudowane wszystkie potrzebne funkcje do łatwego stworzenia sieci. Biblioteka została zaimportowana do Javy.

**2b)**

Utworzono dane uczące oraz testujące, które zawierają siatki liter 5x7 w reprezentacji zero-jedynkowej. Dane uczące są identyczne z testującymi.

Reprezentacja alfabetu postaci zero-jedynkowej:

A

0 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

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

C

0 1 1 1 0

1 0 0 0 1

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

1 0 0 0 1

0 1 1 1 0

D

1 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 0

E

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 1 1 1 1

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

G

0 1 1 1 0

1 0 0 0 1

1 0 0 0 0

1 0 1 1 1

1 0 0 0 1

1 0 0 0 1

0 1 1 1 0

H

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

I

0 1 1 1 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 1 1 1 0

J

1 1 1 1 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

1 0 0 0 1

0 1 1 1 0

K

1 0 0 0 1

1 0 0 1 0

1 0 1 0 0

1 1 0 0 0

1 0 1 0 0

1 0 0 1 0

1 0 0 0 1

L

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

1 1 1 1 1

M

1 0 0 0 1

1 1 0 1 1

1 0 1 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

N

1 0 0 0 1

1 0 0 0 1

1 1 0 0 1

1 0 1 0 1

1 0 0 1 1

1 0 0 0 1

1 0 0 0 1

O

0 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

0 1 1 1 0

P

1 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 1 1 1 0

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

Q

0 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 1 0 1

1 0 0 1 0

0 1 1 0 1

R

1 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 1 1 1 0

1 0 1 0 0

1 0 0 1 0

1 0 0 0 1

S

0 1 1 1 0

1 0 0 0 1

1 0 0 0 0

0 1 1 1 0

0 0 0 0 1

1 0 0 0 1

0 1 1 1 0

T

1 1 1 1 1

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

U

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

0 1 1 1 0

V

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

0 1 0 1 0

0 0 1 0 0

W

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 0 0 1

1 0 1 0 1

1 0 1 0 1

0 1 0 1 0

X

1 0 0 0 1

1 0 0 0 1

0 1 0 1 0

0 0 1 0 0

0 1 0 1 0

1 0 0 0 1

1 0 0 0 1

Y

1 0 0 0 1

1 0 0 0 1

0 1 0 1 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

Z

1 1 1 1 1

0 0 0 0 1

0 0 0 1 0

0 0 1 0 0

0 1 0 0 0

1 0 0 0 0

1 1 1 1 1

Dane uczące oraz testujące wyglądają następująco:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ze względu na wygodę obliczeń cała reprezentacja jest w jednej linii, gdzie pierwsze 35 cyfr to właśnie litera, a następne 26 cyfr to oznaczenie którą reprezentuje literę.

2c)

W celu przetestowania poprawności działania sieci neuronowej wykonano 3 rodzaje testów:

* Stała liczba neuronów w warstwie ukrytej(10) oraz stały współczynnik uczenia(0,1)
* Stała liczba neuronów w warstwie ukrytej(10 oraz stały błąd maksymalny(0,01)
* Stały współczynnik uczenia(0,1) oraz stały błąd maksymalny(0,05 i 0,1)

Stała liczba neuronów w warstwie ukrytej(10) oraz stały współczynnik uczenia(0,1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| staly wspolczynnik uczenia | 0.1 |  |  |  |
| L.P. | MaxError | Liczba iteracji | Liczba Błędów | Błąd uczenia |
| 1 | 0,001 | 7489 | 0 | 9.999017183015793E-4 |
| 2 | 0,05 | 760 | 0 | 0.04996448363628038 |
| 3 | 0,1 | 504 | 0 | 0.09991435304441959 |
| 4 | 0,25 | 257 | 3 | 0.2498702664950261 |
| 5 | 0,5 | 5 | 23 | 0.49826143672759965 |
| 6 | 0,75 | 2 | 24 | 0.7146398609798011 |
| 7 | 1 | 2 | 25 | 0.6763210632196294 |

Wykres liczby iteracji w zależności od maksymalnego błędu uczenia.

Wykres liczby błędu rozpoznawania liter(na 26 wszystkich) od maksymalnego błędu uczenia.

Stała liczba neuronów w warstwie ukrytej(10 oraz stały błąd maksymalny(0,01):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| staly MaxError | 0,01 |  |  |  |
| L.P. | współczynnik uczenia | liczba iteracji | Liczba błędów | Bład uczenia |
| 1 | 0.01 | 19531 | 0 | 0.009999644628075278 |
| 2 | 0.05 | 4054 | 0 | 0.009994252021489802 |
| 3 | 0.1 | 2192 | 0 | 0.009996877637697207 |
| 4 | 0.5 | 4837 | 0 | 0.009756926062756837 |
| 5 | 0,75 | 309 | 0 | 0.009943804496053246 |

Wykres liczby iteracji w zależności od współczynnika uczenia.

Liczba błędów wynosiła w każdym przypadku 0 więc nie ma sensu zamieszczać wykresu.

Stały współczynnik uczenia(0,1) oraz stały błąd maksymalny(0,05):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stały współczynnik uczenia i maxError | współczynnik uczenia = 0.1 | maxError = 0.05 |  |  |
| L.P. | Liczba neuronów ukrytych | Liczba iteracji | Liczba błędów | Błąd uczenia |
| 1 | 1 | - | - | - |
| 2 | 2 | - | - | - |
| 3 | 7 | 3524 | 2 | 0.049999607842650416 |
| 4 | 10 | 1220 | 0 | 0.04993310332500016 |
| 5 | 15 | 379 | 0 | 0.049821633343280554 |
| 6 | 20 | 284 | 0 | 0.04980737339379792 |
| 7 | 50 | 136 | 0 | 0.0499838314711676 |
| 8 | 100 | 81 | 0 | 0.04898698892441474 |

Wykres liczby iteracji w zależności od liczby neuronów w warstwie ukrytej.

Stały współczynnik uczenia(0,1) oraz stały błąd maksymalny(0,1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stały współczynnik uczenia i maxError | współczynnik uczenia = 0.1 | maxError = 0.1 |  |  |
| L.P. | Liczba neuronów ukrytych | Liczba iteracji | Liczba błędów | Błąd uczenia |
| 1 | 5 | 39429 | 4 | 0.0999837377378111 |
| 2 | 8 | 1126 | 0 | 0.09994157831013414 |
| 3 | 10 | 460 | 0 | 0.09986894655489219 |
| 4 | 15 | 326 | 0 | 0.09945863549425606 |
| 5 | 20 | 212 | 0 | 0.09919280058884664 |
| 6 | 50 | 84 | 0 | 0.09892466751457617 |
| 7 | 100 | 61 | 0 | 0.09891312473633879 |

**2d)**

Analiza:

Najbardziej wpływającym czynnikiem jest maksymalny błąd uczenia. Im mniejszy błąd uczenia tym sieć jest dokładniejsza i praktycznie nie popełnia błędów. Powyżej warotści 0,25 maksymalnego błędu uczenia stosowanie sieci nie ma sensu, bo liczba źle rozpoznanych liter wzrasta bardzo szybko do prawie maksymalnej wartości. Mały maksymalny błąd uczenia niesie jednak za sobą konsekwencje dużego czasu co widać po dużej liczbie iteracji.

Współczynnik uczenia wpływa na szybkość uczenia i jak widać w tabeli oraz wykresie w przedziale <0,05;0,5> utrzymuje się na podobnym poziomie. Różnice mogą stworzyć różne wylosowane wagi za każdym uruchomieniem programu. Co do najmniejszej wartości oraz największej jasno widać, że liczba iteracji wzrosłą kilkuklrotnie, a w najmniejszym spadła nagle kilkukrotnie.

Ostatnim sprawdzanym kryterium była liczba neuronów w warstwie ukrytej. Wykonano tutaj dwa podejścia z różnym maksymalnym błędem uczenia. W obu przypadkach uzyskanie wyniku poniżej 5 neuronów było zbyt czasochłonne. Zgodnie z przewidywaniami im więcej neuronów w warstwie ukrytej tym mniej iteracji jest potrzebnych do ustalenia odpowiednich wag. Idzie to jednak w parze z ilością obliczeń oraz obciążeniem maszyny.

Poniżej 7 neuronów i maksymalnego błędu uczenia wynoszącego 0,05 czas obliczeń był tak czasochłonny, że nie jest on poddawany porównaniom.

Wszystkie testy ze zmiennym współczynnikiem uczenia lub zmiennym maksymalnym błędem uczenia wykonywano dla 10 neuronów w ukrytej warstwie. Było optymalne rozwiązanie, które pozwalało na dość szybkie wykonanie programu przy rozsądnym obciążeniu maszyny.

**2e)**

Wnioski:

Maksymalny błąd uczenia miał największy wpływ na dokładność rozpoznawania liter.

Współczynnik uczenia miał spory wpływ w szybkości uczenia. Ilość iteracji jest uzależniona zarówno od współczynnika uczenia jak i maksymalnego błędu uczenia. W celu najoptymalniejszego działania programu należy wybrać odpowiednie ich wartości mając na względzie czy bardziej zależy nam na czasie czy dokładności obliczeń.

Podobnie jest z liczbą neuronów w warstwie ukrytej. Nie może ona przekroczyć minimalnego progu aby program działał. Stosowanie zbyt dużej ilości może też nie być odpowiednie ze względu na ilość danych jakie musi pomieścić pamięć oraz obciążenie maszyny.

Ustawiając parametry sieci należy próbować różnych parametrów, dopóki jakiś zestaw będzie działał odpowiednio z naszymi wymaganiami. W moim przypadku były to parametry współczynnika uczenia 0,1 maksymalnego błędu uczenia 0,01 oraz 10 neuronów w warstwie ukrytej.

**2f)**

Listing kodu znajduje się na GitHubie: