

Podstawy Sztucznej Inteligencji- Sprawozdanie

Scenariusz 3

Cel ćwiczenia

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

Zadania do wykonania

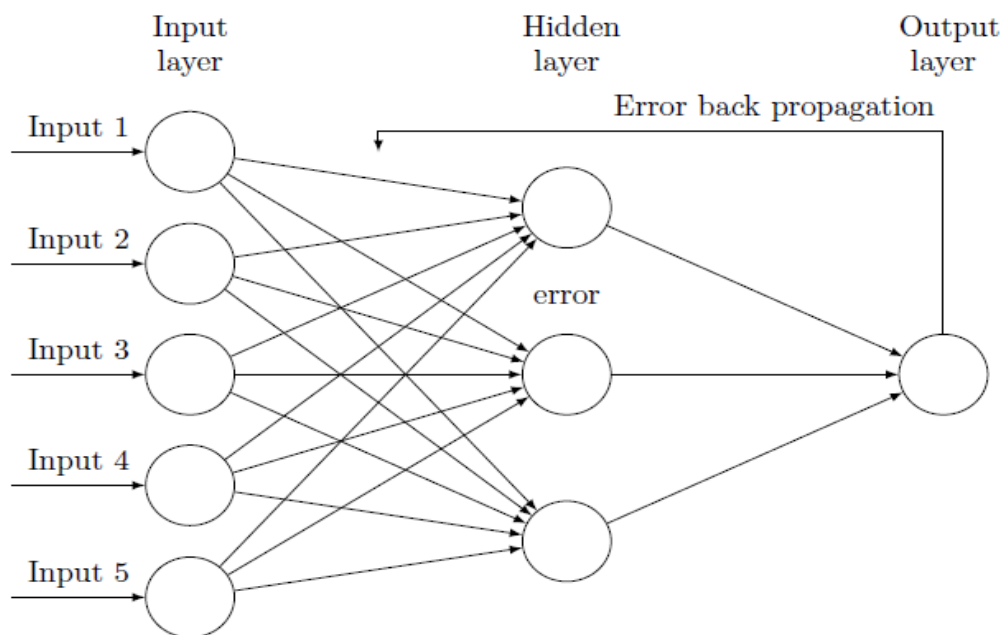
- Wygenerowanie danych uczących i testujących, zawierających 20 dużych liter dowolnie wybranego alfabetu.
- Przygotowanie wielowarstwowej sieci oraz algorytmu wstecznej propagacji błędów.
- Uczenie sieci dla różnych współczynników uczenia i bezwładności.
- Testowanie sieci.

Specyfikacja

Do wykonania ćwiczenia wykorzystałem sieć typu FeedForward z neuronami sigmoidalnymi. Algorytm, z którego korzystałem to algorytm wstecznej propagacji błędów rozpoznawania. Polega on na takiej zmianie wag sygnałów wejściowych każdego neuronu w każdej warstwie, by wartość błędów dla kolejnych par uczących zawartych w zbiorze uczącym była jak najmniejsza. W tym celu wykorzystywana jest metoda gradientowa najszybszego spadku.

Schemat krokowy wygląda następująco:

1. Inicjalizacja sieci i algorytmu.
2. Obliczanie wartości wyjściowej sieci na podstawie danych.
3. Obliczanie błędów sieci.
4. Korekta wag.
5. Czy się nauczona?
 - a. TAK- przejdź dalej.
 - b. NIE- wróć do punktu 2.
6. Koniec.



Schemat sieci typu FeedForward korzystającym z algorytmu propagacji wstecznej

Wykonanie zadania

Sieć składa się z składników:

- 35 neuronów liniowych warstwy wejściowej (każdy neuron odpowiada jednemu znakowi z tablicy).
- 25 neuronów sigmoidalnych warstwy ukrytej z dołączonym biosem.
- 20 neuronów liniowych warstwy wyjściowej (każdy neuron odpowiada za jeden element tablicy, która identyfikuje literę).

Działanie programu testowałem w kilku kombinacjach. Zmiennymi są: współczynnik nauki(learning rate) i współczynnik bezwładności(momentum). Learning rate przyjmował wartości 0.01 i 0.1, a momentum 0.1 i 0.01.

Momentum jest współczynnikiem, od którego zależy korekta wag neuronu. korekta wag neuronu zależy nie tylko od sygnału wejściowego i błędu, jaki neuron popełnił, ale również od tego, jak duża była korekta wag w poprzednim kroku uczenia. W ten sposób szybkość uczenia (wielkość korekty wag) automatycznie maleje w miarę zbliżania się do właściwego rozwiązania, a proces uczenia staje się płynniejszy (więcej o momentum w części "dla fachowców"). Używając momentum można zwiększyć współczynnik uczenia.

Działanie programu

Program po uruchomieniu przystępuje do treningu sieci wg. Algorytmu propagacji wstecznej, po każdej epoce wypisując całkowity błąd uczenia.

Przykładowy output:

Total error: 1.13158360968e-12
Total error: 1.28852802384e-12
Total error: 1.51885551763e-12
Total error: 1.55507251603e-12
Total error: 1.49872158707e-12
Total error: 1.27782405507e-12
Total error: 1.17895624191e-12
Total error: 1.55987479394e-12
Total error: 1.81378932215e-12
Total error: 2.38206170573e-12
Total error: 2.36278397838e-12
Total error: 1.50554206906e-12
Total error: 1.4901123762e-12
Total error: 1.517881615e-12
Total error: 1.51007797193e-12
Total error: 1.13009185887e-12
Total error: 1.04689709008e-12
Total error: 8.34579017022e-13
Total error: 1.02182143901e-12
Total error: 1.15709666617e-12

Po skończonym treningu przechodzi do jej testowania. Program analizuje dane wyjścia porównując je do danych oczekiwanych.

Przykładowy output:

Dla litery C output wynosi:

2.15432596984e-07

2.55901977386e-08

1.00000000513

1.00422877328e-06

3.78723876593e-07

4.1738191553e-07

1.89134079509e-08

5.11354729504e-07

3.33480184633e-07

9.04747482509e-07

6.31162405562e-07

6.5414254441e-07

1.78135975415e-07

1.60962385531e-07

4.06003039327e-07

2.98266923299e-07

3.49109398654e-07

2.24965655271e-08

6.89940120036e-07

1.45316653416e-06

Według danych podanych wyżej mamy analizę litery „C”. Sieć rozpoznała dobrze literę na co wskazują 3 wartości, bo wynosi prawie 1.

Analiza

Proces uczenia odbywał się dla stałej liczby epok wynoszącej 2000.

Dla każdej kombinacji wykonałem po 2 próby.

Na podstawie ogólnego błędu porównywałem wszystkie kombinacje. Ustaloną wartością graniczną była wartość <0.001 .

Do osiągnięcia tego progu najmniej potrzebowała kombinacja learning rate=0.1 i momentum=0.01. Liczba epok wyniosła tylko 900. Następne kombinacje:

- Learning rate=0.1, momentum=0.1, epochs=1200
- Learning rate=0.01, momentum=0.01, epochs=1300
- Learning rate=0.01, momentum=0.1, epochs=1400

Oprócz tego wszystkie kombinacje były bezbłędne. Oprócz tego poszukiwałem również podobieństw wśród liczb, ale takowych nie zauważyłem.

Wnioski

Na podstawie wyników można stwierdzić, że ta sieć neuronowa bardzo dobrze sobie poradziła z powierzonym jej zadaniem. Algorytm wstecznej propagacji celująco zrealizował uczenie tej sieci co pozwoliło na poświęcenie małej ilości epok na naukę. Analiza podobieństw między literami nie znalazła takowego przypadku. Program zrealizował zadanie w 100%, nie popełniał błędów i potrzebował bardzo mało czasu na naukę.