

WSI - ćwiczenie 5.

Sztuczne sieci neuronowe

Dokumentacja

Treść zadania:

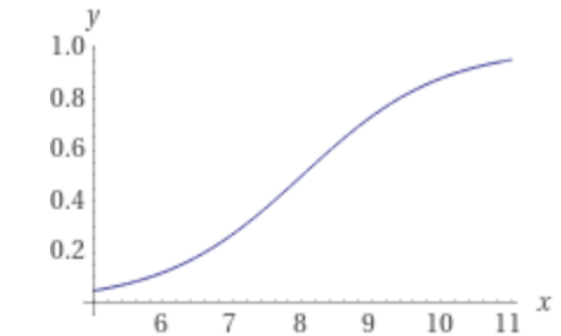
Celem ćwiczenia jest implementacja perceptronu wielowarstwowego oraz wybranego algorytmu optymalizacji gradientowej z algorytmem propagacji wstecznej. Następnie należy wytrenować perceptron wielowarstwowy do klasyfikacji zbioru danych MNIST.

Implementacja

Założenia:

W implementacji w rozwiązaniu zadanie ustanowiliśmy następujące założenia:

- pierwsza warstwa neuronów jest warstwą wejściową i posiada 64 neuronów
- ostatnia warstwa jest warstwą wyjściową i posiada 10 neuronów
- ilość warstw pomiędzy (tzw. hidden layers) ustawiona jest w zależności od testowania
- parametr alpha o wartościach 0.1 lub 0.5
- podział 1797 elementowego zbioru MNIST - grupa trenująca posiada 1203, testująca 297 i walidacyjna 297 elementów
- funkcją aktywacją jest funkcja sigmoida
- wartości próbek są od 0 do 16 co przy większych wartościach po przejściu przez funkcję aktywacyjną daje bardzo małe różnice, więc od wzoru funkcji sigmoidy odejmujemy wartość 8 (wizualizacja poniżej)



Wyniki:

Przeprowadziliśmy testy dla różnych wartości ukrytych warstw, α oraz wielkości próbek dla tych samych wartości wyjściowych (simulate.py):

```
Hidden Layer: [50, 40, 30, 20], Alpha: 0.5, Sample Size: 3  
124 297
```

```
Hidden Layer: [50, 40, 30, 20], Alpha: 0.5, Sample Size: 5  
64 297
```

```
Hidden Layer: [50, 40, 30, 20], Alpha: 0.1, Sample Size: 3  
87 297
```

```
Hidden Layer: [50, 40, 30, 20], Alpha: 0.1, Sample Size: 5  
29 297
```

```
Hidden Layer: [50, 30, 30, 30, 20], Alpha: 0.5, Sample Size: 3  
31 297
```

```
Hidden Layer: [50, 30, 30, 30, 20], Alpha: 0.5, Sample Size: 5  
29 297
```

```
Hidden Layer: [50, 30, 30, 30, 20], Alpha: 0.1, Sample Size: 3  
79 297
```

```
Hidden Layer: [50, 30, 30, 30, 20], Alpha: 0.1, Sample Size: 5  
78 297
```

Best Params:
Hidden Layer: [50, 40, 30, 20], Alpha: 0.5, Sample Size: 3
199 297

W poniższych tabelach przedstawione zostały wyniki dla większej ilości testów, gdzie wartością wynikową jest procentowa poprawność algorytmu.

Tabela dla wartości ukrytych warstw:

- [50, 40, 30, 20]

Test	Parameters			
	Alpha: 0.1 Sample size: 3	Alpha: 0.1 Sample size: 5	Alpha: 0.5 Sample size: 3	Alpha: 0.5 Sample size: 5
1	0.32	0.2	0.61	0.34
2	0.19	0.09	0.51	0.35
3	0.32	0.2	0.62	0.59
4	0.18	0.22	0.54	0.36
5	0.32	0.17	0.66	0.39
6	0.31	0.18	0.55	0.51
7	0.07	0.16	0.56	0.45
8	0.1	0.22	0.49	0.29

- [50, 30, 30, 30, 20]

Test	Parameters			
	Alpha: 0.1 Sample size: 3	Alpha: 0.1 Sample size: 5	Alpha: 0.5 Sample size: 3	Alpha: 0.5 Sample size: 5
1	0.17	0.24	0.42	0.37
2	0.09	0.19	0.52	0.26
3	0.13	0.15	0.43	0.33
4	0.24	0.12	0.26	0.27
5	0.24	0.14	0.61	0.25
6	0.1	0.18	0.27	0.28
7	0.25	0.22	0.31	0.3
8	0.14	0.19	0.23	0.4

Obserwacje:

Algorytm osiągnął najlepsze wyniki dla parametrów - alpha = 0.5 i sample size = 3, gorsze wyniki występowały dla wartości alpha = 0.1

Następnie przeprowadziliśmy testy pokazujące przebieg nauki algorytmu dla dwóch różnych wartości wejściowych (learning_example.py):

Przykładowo

```
First number to recognition: 3
Second number to recognition: 7
```

Z wyników możemy zauważyć, że dla wyjściowych liczb zgodnych z podanymi próbkami algorytm działa poprawnie, dla innych wartości wyjściowych nie co wynika z braku innych wartości wejściowych.

Test	Wyjściowa wartość	Procentowa zgodność dla liczby									
		0	1	2	3	4	5	6	7	8	9
1	3	0.014	0.017	0.0096	0.717	0.023	0.026	0.024	0.302	0.016	0.019
2	1	0.027	0.022	0.015	0.794	0.032	0.026	0.026	0.478	0.0514	0.024
3	2	0.019	0.032	0.024	0.647	0.016	0.024	0.046	0.434	0.042	0.039
4	7	0.023	0.029	0.049	0.092	0.011	0.021	0.036	0.894	0.051	0.027
5	4	0.009	0.025	0.049	0.397	0.187	0.026	0.087	0.851	0.067	0.032
6	0	0.015	0.018	0.015	0.47	0.012	0.016	0.023	0.51	0.013	0.024

Wnioski:

Wyniki w dużej mierze zależą od parametru alpha, dla dużych wartości osiąga on zadowalające wyniki, natomiast dla małych odwrotne rezultaty. Może to wynikać z ilości próbek testujących, sam algorytm ze względu na swoją złożoność jest bardzo trudny do debugowania co bezpośrednio wpływa na wyciągane wnioski.