

WSI Sprawozdanie Lab 5

Jakub Śledź

Dane:

Użyto zbioru danych MNIST. Użyłem sklearn.datasets żeby załadować go do pamięci. Wynikiem klasyfikacji jest rozpoznanie cyfry z zakresu 0-9 więc wynik koduję sposobem one-hot (przykładowo: wartość 3 została zakodowana jako [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]). Dane dzielę na 60% danych trenujących i po 20% walidujących i testujących. Używam StandardScaler aby dane wejściowe miały średnią 0 i odchylenie standardowe 1.

Algorytm:

Ilość i liczebność warstw parametryzowana, tak jak również funkcje aktywacji. Wagi poszczególnych połączeń są inicjalizowane małymi losowymi wartościami. Finalnie zdecydowałem się na 2 warstwy po 64 neurony i warstwę wyjściową o 10 neuronach. Wejście jest propagowane przez kolejne warstwy poprzez wyliczenie ważonej sumy i nałożenie funkcji aktywacji aby wyprodukować wyjście. Wartości wyjściowe warstw są zapisywane na potrzeby mechanizmu propagacji wstecznej. Wynikiem działania sieci jest zwrócenie wartości najbardziej prawdopodobnej przy użyciu funkcji softmax

Propagacja wsteczna błędu:

Obliczam błąd pomiędzy wartością przewidzianą przez sieć a wartością prawdziwą i wyliczam gradient funkcji błędu. Błąd jest propagowany wstecznie przy użyciu pochodnych funkcji aktywacyjnej a następnie wagi są aktualizowane proporcjonalnie do współczynnika uczenia.

Wyniki:

Sieć opisana w sprawozdaniu:

```
Epoch 0, Loss: 2.302619098460113
Epoch 100, Loss: 2.3014342593497163
Epoch 200, Loss: 2.300105901481533
Epoch 300, Loss: 2.2934843958429503
Epoch 400, Loss: 2.0606184414949023
Epoch 500, Loss: 0.9052831627718819
Epoch 600, Loss: 0.37918704457183033
Epoch 700, Loss: 0.190651067739473
Epoch 800, Loss: 0.10124373661509542
Epoch 900, Loss: 0.06123267748714997
Training Time: 8.38 seconds
Validation Accuracy: 0.9637883008356546
Test Accuracy: 0.9472222222222222
Classification Report:
              precision    recall  f1-score   support

     0           0.97       0.97       0.97         38
     1           0.95       0.98       0.96         42
     2           0.97       0.95       0.96         40
     3           0.97       0.91       0.94         35
     4           0.97       0.97       0.97         39
     5           0.94       0.94       0.94         32
     6           0.97       0.97       0.97         34
     7           0.89       0.97       0.93         32
     8           0.97       0.87       0.92         38
     9           0.85       0.93       0.89         30

 accuracy              0.95         360
  macro avg           0.95         0.95         0.95         360
 weighted avg         0.95         0.95         0.95         360
```

Sieć o warstwach [32, 32, 10]

```
Epoch 0, Loss: 2.3025804704986803
Epoch 100, Loss: 2.3017619134713274
Epoch 200, Loss: 2.301471502417717
Epoch 300, Loss: 2.300970641856747
Epoch 400, Loss: 2.299071128494586
Epoch 500, Loss: 2.2762669647041456
Epoch 600, Loss: 1.753021828461603
Epoch 700, Loss: 0.8366645527164095
Epoch 800, Loss: 0.34808536371701165
Epoch 900, Loss: 0.17926436331563303
Training Time: 2.33 seconds
Validation Accuracy: 0.958217270194986
Test Accuracy: 0.9361111111111111
Classification Report:
```

	precision	recall	f1-score	support
0	0.95	0.97	0.96	38
1	0.91	1.00	0.95	42
2	0.95	0.93	0.94	40
3	0.91	0.89	0.90	35
4	0.95	0.95	0.95	39
5	0.94	0.94	0.94	32
6	1.00	0.97	0.99	34
7	0.94	0.97	0.95	32
8	0.89	0.84	0.86	38
9	0.93	0.90	0.92	30
accuracy			0.94	360
macro avg	0.94	0.94	0.94	360
weighted avg	0.94	0.94	0.94	360

Sieć o warstwach [64,10]

```

Epoch 0, Loss: 11.82942567908387
Epoch 100, Loss: 0.6875161931748056
Epoch 200, Loss: 0.2910139925718272
Epoch 300, Loss: 0.18636159623728163
Epoch 400, Loss: 0.15112456526064422
Epoch 500, Loss: 0.13014056504344296
Epoch 600, Loss: 0.10207385989742374
Epoch 700, Loss: 0.09075472479911689
Epoch 800, Loss: 0.08189829463301354
Epoch 900, Loss: 0.07448526273412963
Training Time: 2.33 seconds
Validation Accuracy: 0.9860724233983287
Test Accuracy: 0.9861111111111112
Classification Report:

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	38
1	1.00	1.00	1.00	42
2	0.98	1.00	0.99	40
3	1.00	0.94	0.97	35
4	1.00	1.00	1.00	39
5	0.94	1.00	0.97	32
6	1.00	0.97	0.99	34
7	0.97	0.97	0.97	32
8	1.00	1.00	1.00	38
9	0.97	0.97	0.97	30
accuracy			0.99	360
macro avg	0.99	0.98	0.98	360
weighted avg	0.99	0.99	0.99	360

Jak widać zmniejszenie ilości neuronów przyspiesza proces uczenia ale niekoniecznie przekłada się na pogorszenie wyników. Najlepszy wynik osiągnęła stosunkowo prosta sieć [64,10]. To prawdopodobnie dlatego, że to zagadnienie klasyfikacji jest stosunkowo proste więc stworzenie sieci gdzie każdy pixel wejściowego obrazka ma odpowiadający sobie neuron a warstwa wyjściowa jest wielkości zbioru możliwych wyjść jest optymalne. Dodatkowe warstwy dodają wyłącznie niepotrzebną abstrakcję zaciemniającą istotę problemu.