WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI

Sprawozdanie

Zaawansowane metody uczenia maszynowego

Sieci Neuronowe – pojedyncza warstwa ukryta

Autor:

Karol Baranowski

Prowadzący:

mgr inż. Przemysław Czuba

Spis treści

Zadanie

Dla zbioru danych użytego w przykładzie PlanarClassification zaimplementować sieć neuronową przy uzyciu framework'a PyTorch.

- Zaimplementować strukturę analogiczną jak w przykładzie
- Wykonać porównania wyników oraz czasów uczenia dla:
  - Różnej liczby neuronów w warstwie ukrytej
  - Różnych funkcji aktywacji (sigmoid, tanh, ReLU)•
- (dla chętnych) Użyć innego zbioru danych do klasyfikacji binarnej

Wstęp

Sieć ma za zadanie na podstawie dwóch wartości punktu, rozpoznawać jego kolor wykorzystując klasyfikację binarną. Zadanie zrealizowano używając frameworka PyTorch. Jako dane do trenowania sieci przyjęto zbiór danych z przykładu *PlanarClassification*. Następnie wygenerowano losowe dane testowe o wielkości 20% danych treningowych z tego samego przykładu ale używając innego żarna.

Porównanie wyników

Porównano działanie trzech funkcji aktywacji (sigmoid, tanh, ReLU) na dwóch warstwach – ukrytej i wyjściowej, łącznie dziewięć testów. Dla każdego testu sieć trenowano oddzielnie używając 1, 2 ,3 ,4 ,5, 10, 20 i 50 neuronów w warstwie ukrytej analogicznie jak w załączonym do zadania przykładzie *neural_net.py*.

Poniżej wyniki testów i omówienie najważniejszych obserwacji:

relu – relu

```
Activation function on hidden layer: relu
Activation function on output layer: relu
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.28543
                Cost after 100 epochs:   0.24977
                Cost after 10000 epochs: 0.22532
                Time of training:        3.0519 s
                Accuracy:                61.25%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.62500
                Cost after 100 epochs:   0.32042
                Cost after 10000 epochs: 0.20820
                Time of training:        3.1494 s
                Accuracy:                73.75%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.75227
                Cost after 100 epochs:   0.32131
                Cost after 10000 epochs: 0.22126
                Time of training:        3.4574 s
                Accuracy:                56.25%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.42156
                Cost after 100 epochs:   0.27822
                Cost after 10000 epochs: 0.15798
                Time of training:        3.8837 s
                Accuracy:                80.00%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.48032
                Cost after 100 epochs:   0.42505
                Cost after 10000 epochs: 0.17748
                Time of training:        4.3402 s
                Accuracy:                73.75%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.49727
                Cost after 100 epochs:   0.38698
                Cost after 10000 epochs: 0.09975
                Time of training:        4.1027 s
                Accuracy:                81.25%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.38687
                Cost after 100 epochs:   0.26678
                Cost after 10000 epochs: 0.09123
                Time of training:        5.4077 s
                Accuracy:                82.50%
```

Zastosowanie ReLu na warstwie ukrytej i wyjścia dało wysoki wynik już od użycia 4 neuronów w ukrytej warstwie. ReLu szybko radzi sobie z ujemnymi wartościami ponieważ przybliża je do 0. Wyliczony koszt dla 20 i 50 neuronów jest najniższy ze wszystkich testów. Zazwyczaj ReLu używa się do warstw ukrytych, gdyż przyspiesza uczenie.

relu – sigmoid

```
Activation function on hidden layer: relu
Activation function on output layer: sigmoid
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.30098
                Cost after 100 epochs:   0.29255
                Cost after 10000 epochs: 0.23116
                Time of training:        3.0823 s
                Accuracy:                55.00%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.27002
                Cost after 100 epochs:   0.26605
                Cost after 10000 epochs: 0.22370
                Time of training:        3.6828 s
                Accuracy:                61.25%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.27691
                Cost after 100 epochs:   0.26993
                Cost after 10000 epochs: 0.22483
                Time of training:        3.7037 s
                Accuracy:                68.75%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.26461
                Cost after 100 epochs:   0.26000
                Cost after 10000 epochs: 0.22453
                Time of training:        3.7004 s
                Accuracy:                61.25%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.25001
                Cost after 100 epochs:   0.24693
                Cost after 10000 epochs: 0.22425
                Time of training:        3.5939 s
                Accuracy:                60.00%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.26559
                Cost after 100 epochs:   0.24736
                Cost after 10000 epochs: 0.17268
                Time of training:        4.0321 s
                Accuracy:                78.75%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.26375
                Cost after 100 epochs:   0.24236
                Cost after 10000 epochs: 0.13219
                Time of training:        5.0175 s
                Accuracy:                81.25%
```

Kombinacja ta daje wysoką dokładność dla wyższej ilości neuronów w ukrytej warstwie (20). Poza tym od początku wartość kosztu jest stosunkowa niska. Sigmoid nadaje się na funkcję aktywacji warstwy wychodzącej w klasyfikacji binarnej, ponieważ zwraca wynik z przedziału 0 i 1.

relu – tanh

```
Activation function on hidden layer: relu
Activation function on output layer: tanh
        Neurons in hidden layer: 1
                Cost after 1 epoch:      1.80961
                Cost after 100 epochs:   0.30273
                Cost after 10000 epochs: 0.24056
                Time of training:        3.3342 s
                Accuracy:                45.00%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.27044
                Cost after 100 epochs:   0.24869
                Cost after 10000 epochs: 0.23389
                Time of training:        3.7985 s
                Accuracy:                60.00%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.26884
                Cost after 100 epochs:   0.25368
                Cost after 10000 epochs: 0.23545
                Time of training:        3.8444 s
                Accuracy:                60.00%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      1.44692
                Cost after 100 epochs:   0.25101
                Cost after 10000 epochs: 0.21551
                Time of training:        3.7728 s
                Accuracy:                60.00%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.95975
                Cost after 100 epochs:   0.24084
                Cost after 10000 epochs: 0.19796
                Time of training:        3.8918 s
                Accuracy:                71.25%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.33156
                Cost after 100 epochs:   0.24742
                Cost after 10000 epochs: 0.10714
                Time of training:        4.2331 s
                Accuracy:                80.00%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      1.13878
                Cost after 100 epochs:   0.23076
                Cost after 10000 epochs: 0.10011
                Time of training:        6.5994 s
                Accuracy:                82.50%
```

Wyniki podobne jak dla relu-sigmoid, dokładność 80% od 20 neuronów w ukrytej warstwie. Nieco dokładniejsza (~1,5%) dla 50 neuronów ale o 1,5 s wolniejsza.

sigmoid – relu

```
Activation function on hidden layer: sigmoid
Activation function on output layer: relu
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.50000
                Cost after 100 epochs:   0.50000
                Cost after 10000 epochs: 0.50000
                Time of training:        4.7129 s
                Accuracy:                50.00%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.50000
                Cost after 100 epochs:   0.50000
                Cost after 10000 epochs: 0.50000
                Time of training:        4.2647 s
                Accuracy:                50.00%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.24799
                Cost after 100 epochs:   0.24264
                Cost after 10000 epochs: 0.16270
                Time of training:        4.0560 s
                Accuracy:                80.00%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.39298
                Cost after 100 epochs:   0.26711
                Cost after 10000 epochs: 0.17389
                Time of training:        4.1152 s
                Accuracy:                81.25%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.47828
                Cost after 100 epochs:   0.25875
                Cost after 10000 epochs: 0.15296
                Time of training:        4.0819 s
                Accuracy:                81.25%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.34981
                Cost after 100 epochs:   0.24783
                Cost after 10000 epochs: 0.13659
                Time of training:        4.5248 s
                Accuracy:                78.75%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.30819
                Cost after 100 epochs:   0.23316
                Cost after 10000 epochs: 0.13948
                Time of training:        5.6148 s
                Accuracy:                78.75%
```

Czas rośnie stosunkowo wolno wraz z wzrostem ilości neuronów. Od 3 neuronów dokładność wysoka (80%), jednak dokładność nie rośnie wraz z wzrostem ilości neuronów, a wręcz minimalnie opada.

sigmoid – sigmoid

```
Activation function on hidden layer: sigmoid
Activation function on output layer: sigmoid
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.26827
                Cost after 100 epochs:   0.26459
                Cost after 10000 epochs: 0.24126
                Time of training:        3.4010 s
                Accuracy:                63.75%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.27893
                Cost after 100 epochs:   0.27083
                Cost after 10000 epochs: 0.23958
                Time of training:        4.0056 s
                Accuracy:                60.00%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.26151
                Cost after 100 epochs:   0.25607
                Cost after 10000 epochs: 0.22790
                Time of training:        4.0192 s
                Accuracy:                45.00%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.25020
                Cost after 100 epochs:   0.24935
                Cost after 10000 epochs: 0.23215
                Time of training:        4.0137 s
                Accuracy:                58.75%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.25664
                Cost after 100 epochs:   0.25568
                Cost after 10000 epochs: 0.23799
                Time of training:        3.9989 s
                Accuracy:                63.75%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.27403
                Cost after 100 epochs:   0.25790
                Cost after 10000 epochs: 0.22744
                Time of training:        4.3436 s
                Accuracy:                55.00%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.27130
                Cost after 100 epochs:   0.25053
                Cost after 10000 epochs: 0.21126
                Time of training:        5.3340 s
                Accuracy:                63.75%
```

Niska dokładność (60%) prawie dla każdej ilości neuronów, w szczególności dla 3 (45%).

sigmoid – tanh

```
Activation function on hidden layer: sigmoid
Activation function on output layer: tanh
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.69554
                Cost after 100 epochs:   0.32161
                Cost after 10000 epochs: 0.24121
                Time of training:        3.5595 s
                Accuracy:                50.00%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      1.17165
                Cost after 100 epochs:   0.27461
                Cost after 10000 epochs: 0.22273
                Time of training:        3.9825 s
                Accuracy:                51.25%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      1.38313
                Cost after 100 epochs:   0.24022
                Cost after 10000 epochs: 0.15483
                Time of training:        3.9777 s
                Accuracy:                80.00%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.26924
                Cost after 100 epochs:   0.26173
                Cost after 10000 epochs: 0.20595
                Time of training:        4.1236 s
                Accuracy:                71.25%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      1.41704
                Cost after 100 epochs:   0.25335

                Cost after 10000 epochs: 0.22162
                Time of training:        4.0989 s
                Accuracy:                51.25%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.28840
                Cost after 100 epochs:   0.23745
                Cost after 10000 epochs: 0.14029
                Time of training:        4.5608 s
                Accuracy:                81.25%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.71697
                Cost after 100 epochs:   0.24373
                Cost after 10000 epochs: 0.14341
                Time of training:        5.8971 s
                Accuracy:                82.50%
```

Wysoka dokładność dla wielu neuronów (20, 50). Dla małych ilości neuronów dokładność waha się co przedstawiono powyżej w wyniku dla 2,3 i 4 neuronów.

tanh – relu

```
Activation function on hidden layer: tanh
Activation function on output layer: relu
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.25041
                Cost after 100 epochs:   0.24952
                Cost after 10000 epochs: 0.24108
                Time of training:        3.6662 s
                Accuracy:                46.25%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.23174
                Cost after 100 epochs:   0.22376
                Cost after 10000 epochs: 0.21012
                Time of training:        4.3327 s
                Accuracy:                52.50%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.44088
                Cost after 100 epochs:   0.24948
                Cost after 10000 epochs: 0.11674
                Time of training:        4.1510 s
                Accuracy:                82.50%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.38939
                Cost after 100 epochs:   0.32725
                Cost after 10000 epochs: 0.11635
                Time of training:        4.2101 s
                Accuracy:                82.50%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.39877
                Cost after 100 epochs:   0.23982
                Cost after 10000 epochs: 0.11683
                Time of training:        4.1998 s
                Accuracy:                83.75%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.43459
                Cost after 100 epochs:   0.27959
                Cost after 10000 epochs: 0.10718
                Time of training:        4.8268 s
                Accuracy:                80.00%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.48854
                Cost after 100 epochs:   0.27838
                Cost after 10000 epochs: 0.10381
                Time of training:       19.8721 s
                Accuracy:                81.25%
```

Długi czas liczenia dla wielu neuronów. Tanh stosuje się raczej jako warstwę wychodzącą. Dokładność wysoka od 3 neuronów.

tanh – sigmoid

```
Activation function on hidden layer: tanh
Activation function on output layer: sigmoid
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.25771
                Cost after 100 epochs:   0.25330
                Cost after 10000 epochs: 0.22753
                Time of training:        3.5404 s
                Accuracy:                63.75%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      0.25187
                Cost after 100 epochs:   0.24828
                Cost after 10000 epochs: 0.21121
                Time of training:        4.0607 s
                Accuracy:                55.00%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.26222
                Cost after 100 epochs:   0.25584
                Cost after 10000 epochs: 0.18534
                Time of training:        4.1268 s
                Accuracy:                77.50%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      0.29588
                Cost after 100 epochs:   0.28382
                Cost after 10000 epochs: 0.13124
                Time of training:        4.1392 s
                Accuracy:                81.25%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.24380
                Cost after 100 epochs:   0.24154
                Cost after 10000 epochs: 0.13103
                Time of training:        4.3492 s
                Accuracy:                82.50%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.27276
                Cost after 100 epochs:   0.25430
                Cost after 10000 epochs: 0.12389
                Time of training:        4.7602 s
                Accuracy:                83.75%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.24877
                Cost after 100 epochs:   0.22872
                Cost after 10000 epochs: 0.12230
                Time of training:       19.3468 s
                Accuracy:                83.75%
```

Wysoka dokładność od 3 neuronów. Wyniki podobne do tanh-ReLu, chociaż dokładność wyższa dla warstw z 1 i 2 neuronami.

tanh – tanh

```
Activation function on hidden layer: tanh
Activation function on output layer: tanh
        Neurons in hidden layer: 1
                Cost after 1 epoch:      0.37425
                Cost after 100 epochs:   0.25801
                Cost after 10000 epochs: 0.23001
                Time of training:        3.5484 s
                Accuracy:                63.75%
        Neurons in hidden layer: 2
                Cost after 1 epoch:      1.47935
                Cost after 100 epochs:   0.26894
                Cost after 10000 epochs: 0.21498
                Time of training:        3.9742 s
                Accuracy:                67.50%
        Neurons in hidden layer: 3
                Cost after 1 epoch:      0.40129
                Cost after 100 epochs:   0.24914
                Cost after 10000 epochs: 0.11678
                Time of training:        4.1621 s
                Accuracy:                83.75%
        Neurons in hidden layer: 4
                Cost after 1 epoch:      1.16323
                Cost after 100 epochs:   0.32405
                Cost after 10000 epochs: 0.11731
                Time of training:        4.0378 s
                Accuracy:                83.75%
        Neurons in hidden layer: 5
                Cost after 1 epoch:      0.94110
                Cost after 100 epochs:   0.24054
                Cost after 10000 epochs: 0.11931
                Time of training:        4.2478 s
                Accuracy:                85.00%
        Neurons in hidden layer: 20
                Cost after 1 epoch:      0.76288
                Cost after 100 epochs:   0.23236
                Cost after 10000 epochs: 0.11486
                Time of training:        4.7701 s
                Accuracy:                85.00%
        Neurons in hidden layer: 50
                Cost after 1 epoch:      0.70679
                Cost after 100 epochs:   0.22155
                Cost after 10000 epochs: 0.11600
                Time of training:       23.9537 s
                Accuracy:                83.75%
```

Najwyższa dokładność z przeprowadzonych testów. Wyniki najwyższe dla warstw z 5 i 20 neuronami. Dla 50 spada o 1,25% i 4-krotnie wydłuża czas. Dla warstw z 1 lub 2 neuronami wynik dość niski.

Wnioski końcowe

- Cel zadania został osiągnięty. Sieć, w zależności od dobranych parametrów i funkcji aktywacji, uczyła się (minimalizowała funkcję kosztu) wykorzystując dane treningowe, a następnie była w stanie zwrócić poprawnie (najlepiej w 85%) wyniki dla danych testowych, których wcześniej nie analizowała.
- Oprócz ilości warstw oraz użytych funkcji aktywacji, duże znaczenie ma dobranie algorytmu optymalizacji i normalizacji. W zadaniu użyto do estymacji błędu średniokwadratowego (MSE), a do optymalizacji stochastycznego spadku gradientu (SGD). Warto dodać, że wyniki z wysoką poprawnością dla funkcji aktywacji sigmoid uzyskano również używając do normalizacji Binary Cross Entropy (BCELoss), który mierzy entropię pomiędzy elementami warstwy wejścia, a wyjścia. Jednak jest to algorytm stworzony typowo do klasyfikacji binarnej i działa tylko dla wartości z pomiędzy 0 i 1 więc dla danych znormalizowanych funkcją sigmoid. Nie można było tej funkcji zastosować dla danych z tanh i ReLu.
- Średnio czas uczenia dla 10000 iteracji zbioru 800 elementowgo z zadania trwał na sprzęcie autora około 5 sekund dla funkcji sigmoid i ReLu na warstwie ukrytej. Czas uczenia, gdy w tej warstwie użyty był tanh był 4 razy dłuższy dla 50 neuronów w warstwie ukrytej.
- Im więcej neuronów w ukrytej warstwie tym dłuższy czas uczenia i w zależności od funkcji aktywacji – wynik lepszy bądź od pewnej ilości neuronów nie poprawia się, a nawet maleje.
- Learning rate ustawiono na 0,01 – zaobserwowano, że dla takiego parametru dokładność predykcji była najwyższa.
- Czas uczenia można by przyspieszyć używając mocniejszego sprzętu z wieloma kartami graficznymi, gdyż dostępna jest wersja pytorch-cuda wspierająca równoległe obliczenia na wielu kartach.