



**Wydział Elektroniki
i Technik Informatycznych**

POLITECHNIKA WARSZAWSKA

Wstęp do Sztucznej Inteligencji

Ćwiczenie nr 5:
Sieci neuronowe

Kaczmarek Robert
293377

Łysomirski Damian
293365

Warszawa 2022

Polecenie

Zadanie polega na zaimplementowaniu, wytrenowaniu i przeprowadzeniu walidacji sieci neuronowej do klasyfikacji ręcznie pisanych cyfr. Wykorzystanym zbiorem danych będzie MNIST.

Użyte narzędzia

Do wykonania zadania zostało użyte środowisko Google Colaboratory z Pythonem w wersji 3.6 oraz następujące biblioteki i moduły:

- pandas
- numpy
- matplotlib
- sklearn.datasets.fetch_openml
- sklearn.model_selection.train_test_split

(W mailu napisał Pan, że można korzystać z `sklearn.datasets.fetch_mldata`, ale okazało się, że już nie działa, dlatego został użyty zamiennik).

Wstępna analiza zbioru

Zbiór danych MNIST zawiera 70000 odręcznie napisanych cyfr. Każda cyfra to obraz o wymiarach 28x28 pikseli. Każdy obraz ma przyporządkowaną do siebie etykietę, która określa jaką cyfrą się na nim znajduje. Zbiór danych został podzielony na zbiór treningowy (59990 obrazów) oraz zbiór testowy (10010 obrazów).

Aby można było korzystać z gotowego zbioru, dane musiały być w odpowiedni sposób przygotowane. Wartości z listy 784 pikseli w obrazach zostały znormalizowane do wartości z przedziału $[0, 1]$.

Etykiety obrazów zawierały cyfry zapisane jako *string*, zostały zamienione na listy 10 elementowe z jedynką na pozycji określającej cyfrę jaką została zapisana na obrazie. Dla przykładu: cyfra 5 została zamieniona na $[0,0,0,0,0,1,0,0,0]$.

Wygląd przykładowej sieci

Przykładowa sieć neuronowa składa się z 784 neuronów warstwy wejściowej, 256 neuronów warstwy ukrytej i 10 neuronów warstwy wyjściowej. Każdy neuron warstwy wejściowej jest połączony z neuronem warstwy ukrytej, a każdy neuron warstwy ukrytej z neuronem warstwy wyjściowej. Wszystkie połączenia mają określone wagi. Do każdego neuronu, oprócz neuronów warstwy wejściowej, dodawany jest *bias*.

Początkowo wagi i biasy inicjalizowane są losowymi wartościami z rozkładu normalnego. Do uczenia się model wykorzystuje propagację w przód oraz wsteczną propagację gradientu. Po inicjalizacji następuje propagacja w przód, czyli dla wejściowych danych X , wag W i biasów b obliczana jest wartość

$$Z = WX + b.$$

Następnie obliczona wartość Z podawana jest do funkcji aktywacji. W zaimplementowanej sieci wykorzystana została funkcja *sigmoidalna* oraz dla ostatniej warstwy funkcja *softmax*. Mając wynik propagacji w przód obliczana jest ocena błędu. Do oceny błędu wykorzystano błąd średniokwadratowy MSE (ang. Mean Square Error) w postaci:

$$q = \mathbb{E}((Y - Y_d)^2)$$

gdzie Y to wartość obliczona z propagacji, a Y_d to spodziewany wynik dla danego przykładu ze zbioru testującego. Następnie następuje wsteczna propagacja gradientu, w której wyznaczony zostaje gradient błędów, po czym aktualizowane są wartości parametrów W oraz b :

$$W = W - \beta \nabla q(W)$$

$$b = b - \beta \nabla q(b)$$

gdzie β to współczynnik uczenia się.

Wyniki

Analiza działania zaimplementowanej sieci neuronowej została przeprowadzona pod kątem sprawdzenia wpływu parametrów takich jak: liczba epok, liczba neuronów, liczba warstw ukrytych, współczynnik uczenia się. Znajdujący się w tabeli parametr *Wymiary sieci* określa liczbę neuronów oraz liczbę warstw, przy czym 784 to neurony warstwy wejściowej, a 10 to neurony warstwy wyjściowej. *Poprawność* określa liczbę poprawnie zaklasyfikowanych obrazów w stosunku do liczby wszystkich obrazów testowych.

Wszystkie eksperymenty zostały przeprowadzone dla tych samych ziaren generatora liczb losowych (`random_state = 1`, `np.random.seed(1)`). Dla zaoszczędzenia cennego czasu macierz pomyłek i miary jakości klasyfikacji zostały wyznaczone tylko dla jednego zestawu parametrów.

Tabela 1. Badanie wpływu liczby epok na poprawność działania zaimplementowanego modelu.

Epoka	Poprawność	q	β	Wymiary sieci
1	35,69%	0,127	0,01	[784, 256, 10]
2	46,64%	0,109	0,01	[784, 256, 10]
3	54,87%	0,096	0,01	[784, 256, 10]
4	60,88%	0,087	0,01	[784, 256, 10]
5	64,90%	0,080	0,01	[784, 256, 10]
6	68,04%	0,074	0,01	[784, 256, 10]
7	71,14%	0,070	0,01	[784, 256, 10]
8	73,27%	0,066	0,01	[784, 256, 10]
9	75,01%	0,062	0,01	[784, 256, 10]
10	76,82%	0,059	0,01	[784, 256, 10]
11	78,04%	0,057	0,01	[784, 256, 10]
12	79,17%	0,055	0,01	[784, 256, 10]
13	79,90%	0,053	0,01	[784, 256, 10]
14	80,56%	0,051	0,01	[784, 256, 10]
15	80,96%	0,049	0,01	[784, 256, 10]

Tabela 2. Badanie wpływu zmiany współczynnika uczenia się, liczby neuronów oraz liczby warstw sieci neuronowej.

Epoka	Poprawność	q	β	Wymiary sieci
1	73,17%	0,063	0,1	[784, 256, 10]
5	87,20%	0,031	0,1	[784, 256, 10]
10	89,17%	0,023	0,1	[784, 256, 10]
1	12,55%	0,156	0,001	[784, 256, 10]
5	26,73%	0,140	0,001	[784, 256, 10]
10	35,61%	0,126	0,001	[784, 256, 10]

1	11,98%	0,161	0,0001	[784, 256, 10]
5	11,48%	0,159	0,0001	[784, 256, 10]
10	12,54%	0,156	0,0001	[784, 256, 10]
1	24,69%	0,112	0,01	[784, 256, 128, 10]
5	65,51%	0,072	0,01	[784, 256, 128, 10]
10	76,98%	0,055	0,01	[784, 256, 128, 10]
1	24,74%	0,098	0,01	[784, 256, 128, 64, 10]
5	68,94%	0,067	0,01	[784, 256, 128, 64, 10]
10	78,38%	0,051	0,01	[784, 256, 128, 64, 10]
1	28,06%	0,090	0,01	[784, 256, 128, 64, 32, 10]
5	66,39%	0,070	0,01	[784, 256, 128, 64, 32, 10]
10	77,58%	0,054	0,01	[784, 256, 128, 64, 32, 10]
1	37,07%	0,133	0,01	[784, 512, 10]
5	66,93%	0,082	0,01	[784, 512, 10]
10	74,18%	0,063	0,01	[784, 512, 10]
1	31,08%	0,119	0,01	[784, 128, 10]
5	65,89%	0,077	0,01	[784, 128, 10]
10	77,52%	0,057	0,01	[784, 128, 10]
1	32,40%	0,105	0,01	[784, 64, 10]
5	66,78%	0,070	0,01	[784, 64, 10]
10	77,42%	0,054	0,01	[784, 64, 10]

Ogólna poprawność zaklasyfikowanych obrazów zwiększała się wraz ze epoką, a obliczony błąd MSE zmniejszał się. Przy późniejszych epokach wyniki poprawiały się o niewiele. Oznacza to, że do dobrego wytrenowania modelu potrzeba ok. 20 epok. Przy większej liczbie epok mogłoby dojść do przetrenowania modelu.

Wybór współczynnika uczenia się ma znaczący wpływ na otrzymane wyniki. Dla dużego współczynnika $\beta = 0,1$ udało się uzyskać poprawność na poziomie 90% co jest bardzo dobrym wynikiem. Zmniejszenie tego parametru spowodowało to, że przy tej samej liczbie epok przyrost poprawności i obniżenie błędu w modelu było znacznie mniejsze. Dla $\beta = 0,0001$ poprawność w epoce 5 była mniejsza niż poprawność w epoce 1 (trochę podejrzanie). Możliwe, że przyrost gradientu był zbyt mały co mogło się wiązać z problemem zaniku gradientu.

Jedyny odczuwalny wpływ w przypadku różnej liczby neuronów i różnej liczby warstw to czas trenowania modelu. Im więcej było warstw i neuronów tym więcej czasu zajmowało trenowanie modelu. Czas nie został uwzględniony w tabelce, ponieważ zbyt późno zorientowaliśmy się, że może dobrze byłoby go mierzyć. Przykładowo dla wymiarów sieci [784, 512, 10] trenowanie zajęło około 20 min, a dla [784, 64, 10] około 6 min. Różna liczba neuronów i warstw nie przełożyła się zauważalną zmianę poprawności i błędu.

Tabela 3. Macierz pomyłek dla sieci o wymiarach [784, 256, 10], współczynnika uczenia $\beta = 0,01$ oraz liczbie epok równej 15.

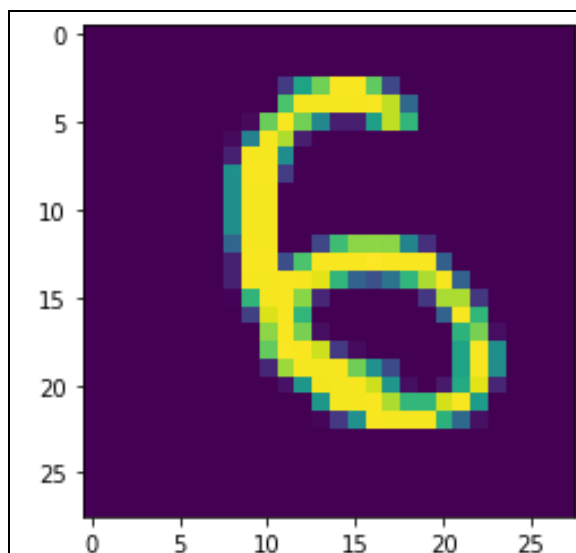
		Cyfra rzeczywista									
		0	1	2	3	4	5	6	7	8	9
Cyfra przewidywana	0	852	0	19	19	3	32	14	12	12	6
	1	2	1126	16	8	3	13	12	14	19	6
	2	9	12	836	45	10	21	25	22	25	11
	3	19	16	31	781	11	77	5	4	68	22
	4	5	3	29	5	785	18	25	27	24	110
	5	28	10	15	78	7	642	23	8	58	13
	6	19	3	27	11	15	37	820	3	19	5
	7	10	0	21	20	14	13	3	892	12	85
	8	5	11	35	50	13	61	8	18	667	29
	9	11	6	7	18	81	17	7	51	32	703

Tabela 4. Miary jakości klasyfikacji dla sieci o wymiarach [784, 256, 10], współczynnika uczenia $\beta = 0,01$ oraz liczbie epok równej 15..

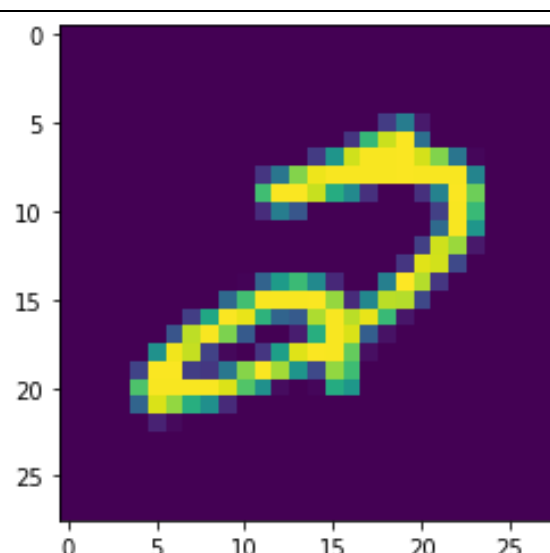
	0	1	2	3	4	5	6	7	8	9
Precyzja	88,75%	94,86%	80,69%	75,46%	83,33%	68,96%	87,05%	84,87%	71,26%	71,01%
Czułość	87,93%	92,37%	82,28%	75,53%	76,14%	72,79%	85,51%	83,36%	74,36%	75,35%
Dokładność	80,96%									

Z wyników zawartych w tabeli nr 4 można wywnioskować, że najłatwiejsza cyfra do zaklasyfikowania to 1. Najtrudniejszą cyfrą do klasyfikacji okazała się 5, ponieważ ma najniższą precyzję i prawie najniższą czułość. Wyniki przedstawione w tabeli nr 3 jasno dają do zrozumienia, że najczęściej mylonymi cyframi są 4 i 9.

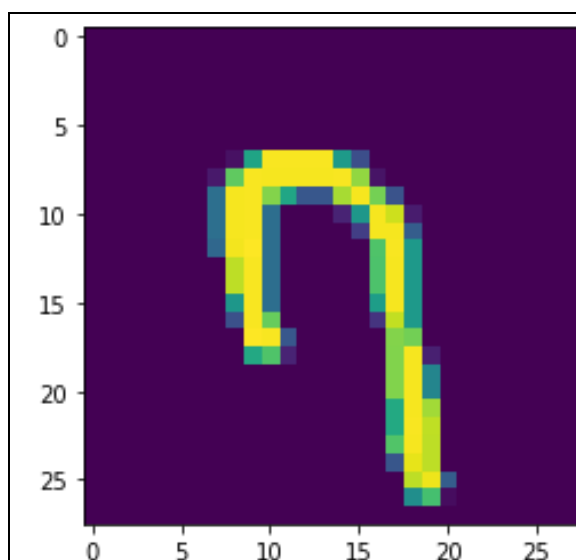
Przykłady poprawnie i niepoprawnie zaklasyfikowanych cyfr zbioru obrazów MNIST zostały przedstawione na rysunkach 1-4.



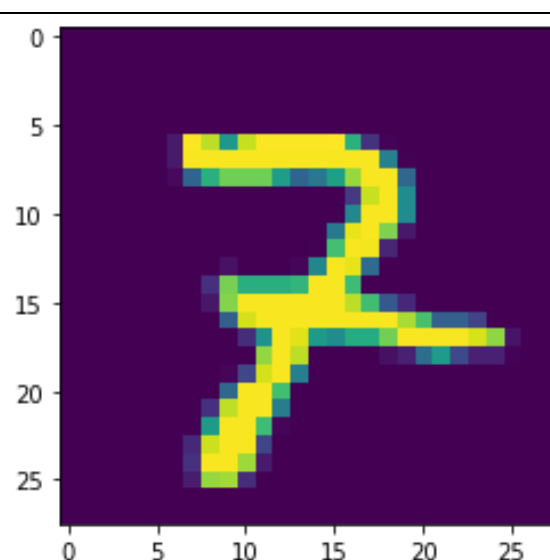
Rysunek 1. Poprawnie zaklasyfikowana cyfra 6.



Rysunek 2. Poprawnie zaklasyfikowana cyfra 2.



Rysunek 3. Niepoprawnie zaklasyfikowana cyfra 7, która została uznana za 9.



Rysunek 4. Niepoprawnie zaklasyfikowana cyfra 7, która została uznana za 2.

Podsumowanie

Przy odpowiednim dobraniu parametrów zaimplementowanej sieci neuronowej możliwe jest uzyskanie ponad 90% poprawności. Niektóre z ręcznie napisanych cyfr ze zbioru nie przypominają wcale cyfr zawartych w dołączonych etykietach. Są to zagadkowe hieroglify, których nawet najstarsi mędrcy, czy najlepsi lekarze nie potrafią zaklasyfikować.

Podział zadań

Damian:

- Wybór architektury sieci neuronowej.
- Przeprowadzenie badań i symulacji.
- Wstępna analiza zbioru.
- Przykłady poprawnie i niepoprawnie zaklasyfikowanych cyfr.

Robert:

- Inicjalizacja wag i biasów.
- Implementacja propagacji wstecznej i propagacji w przód.
- Macierz pomyłek i miary jakości klasyfikacji.
- Przygotowanie zbioru danych MNIST.
- Ostatecznie przygotowanie sprawozdania