

LAB 01 - WPROWADZENIE DO GŁĘBOKIEGO UCZENIA

Karol Działowski

nr albumu: 39259
przedmiot: Uczenie Maszynowe 2

Szczecin, 21 grudnia 2020

Spis treści

1 Cel laboratorium	1
2 Wyniki eksperymentu dla 100 epok	2
2.1 Optymizator	3
2.2 Krok uczenia	3
2.3 Funkcja aktywacji	3
3 Wpływ liczby epok na jakość modelu	3
4 Wnioski	4
5 Sprawdzenie działania klasyfikatora przy uwzględnianiu zniekształceń	5
5.1 Wnioski	7

1 Cel laboratorium

Celem laboratorium było zapoznanie się z funkcjonalnością pakietów *TensorFlow* i *Keras*. W tym celu zaimplementowano perceptron wielowarstwowy. Przetestowano zaimplementowany model badając wpływ następujących parametrów:

- funkcja aktywacji (*sigmoid*, *hard_sigmoid*, *tanh*, *linear*, *relu*, *softmax*)
- liczba epok uczenia
- optymizator (*adam*, *sgd*, *adadelata*, *adagrad*, *rmsprop*)

- krok uczenia

Wyżej wymienione parametry przetestowano opierając się o funkcję kosztu (*loss*) i dokładność (*accuracy*).

2 Wyniki eksperymentu dla 100 epok

W poniższej tabeli przedstawiono 10 najlepszych modeli pod względem dokładności modelu.

Optymizator	Liczba epok	Learning rate	Funkcja aktywacji	Loss	Acc
Adagrad	100	0.10	sigmoid	0.263177	0.9271
Adagrad	100	0.05	softmax	0.264076	0.9258
Adagrad	100	0.02	softmax	0.269774	0.9254
Adagrad	100	0.05	sigmoid	0.264211	0.9253
SGD	100	0.10	sigmoid	0.265435	0.9252
Adadelta	100	0.10	softmax	0.272011	0.9248
SGD	100	0.10	softmax	0.265876	0.9248
Adagrad	100	0.10	softmax	0.265207	0.9247
Adadelta	100	0.10	sigmoid	0.271624	0.9246
Adagrad	100	0.02	sigmoid	0.269853	0.9246

Tabela 1: Najlepsze kombinacje parametrów dla 100 epok

Optymizator	Liczba epok	Learning rate	Funkcja aktywacji	Loss	Acc
Adadelta	100	0.01	linear	7.417104	0.0435
Adadelta	100	0.01	tanh	7.687346	0.0461
Adadelta	100	0.02	linear	7.867214	0.0534
Adadelta	100	0.05	linear	8.153485	0.0536
Adadelta	100	0.10	linear	7.819568	0.0596
Adadelta	100	0.02	tanh	8.111119	0.0797
SGD	100	0.05	tanh	NaN	0.0980
SGD	100	0.05	linear	NaN	0.0980
SGD	100	0.05	relu	NaN	0.0980
Adam	100	0.05	tanh	NaN	0.0980

Tabela 2: Najgorsze kombinacje parametrów dla 100 epok

2.1 Optymizator

optimizer	loss	acc
Adagrad	2.813160	0.509700
Adam	3.461718	0.500542
RMSprop	3.293974	0.421842
SGD	1.780027	0.419479
Adadelta	3.847173	0.410487

Tabela 3: Porównanie optymizatorów dla 100 epok

2.2 Krok uczenia

lr	loss	acc
0.01	3.352127	0.474937
0.05	3.236070	0.450133
0.02	2.010228	0.446010
0.10	3.268945	0.438560

Tabela 4: Porównanie kroku uczenia dla 100 epok

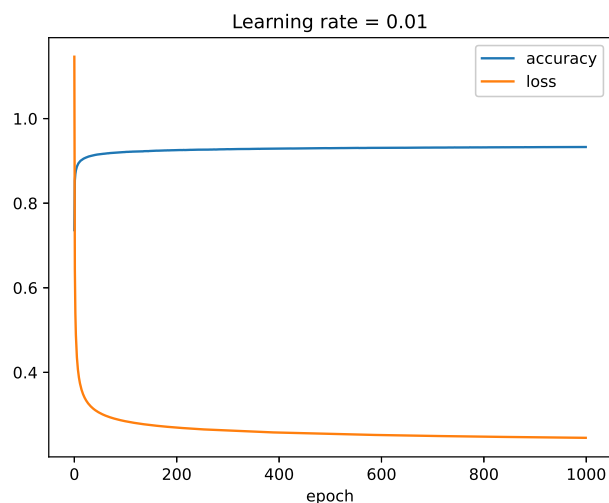
2.3 Funkcja aktywacji

activation	loss	acc
softmax	0.456624	0.911985
sigmoid	0.459970	0.910780
hard_sigmoid	1.563927	0.544735
linear	9.218337	0.148435
tanh	7.706767	0.100355
relu	1.811965	0.098170

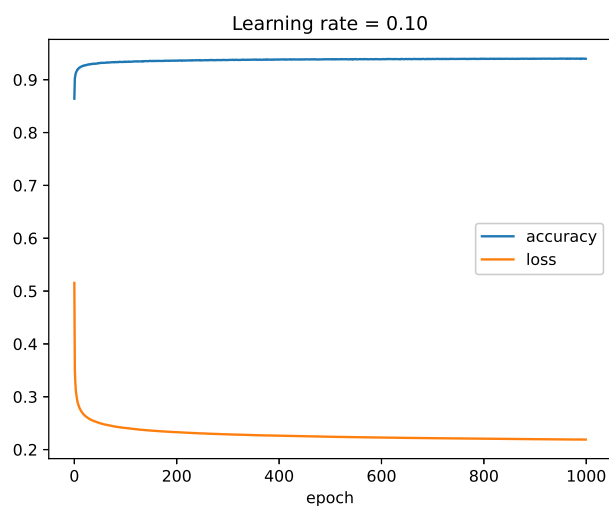
Tabela 5: Porównanie funkcji aktywacji dla 100 epok

3 Wpływ liczby epok na jakość modelu

Zbadano wpływ liczby epok na jakość modelu dla najlepszego modelu z poprzedniego zestawienia, czyli funkcja aktywacji *sigmoid*, optymizator *Adagrad* oraz porównano dwa kroki uczenia: 0.10 i 0.01.



Rysunek 1: Porównanie liczby epok na jakość modelu dla kroku uczenia 0.1



Rysunek 2: Porównanie liczby epok na jakość modelu dla kroku uczenia 0.01

krok uczenia	accuracy	loss
0.01	0.932817	0.245254
0.10	0.939650	0.218975

Tabela 6: Dokładność i funkcja kosztu po 1000 epokach.

4 Wnioski

Największy wpływ na jakość modelu miała wybrana funkcja aktywacji. To od niej zależało, czy model z taką strukturą nauczy się poprawnie i osiągnie zadowalającą dokładność, czy też będzie bezużyteczny.

Jedynymi funkcjami aktywacji jakie osiągały zadowalające wyniki były *softmax* i *sigmoid*. To one osiągnęły dokładność ponad 90% oraz funkcję kosztu poniżej 0.46.

Uśrednione wyniki dla optyimizatorów pokazują, że najlepsze dokładności osiąga optyimizator *Adagrad* oraz *Adam*. *SGD* osiągnął najmniejszą funkcję kosztu dla uśrednionych wyników po wszystkich możliwych funkcjach aktywacji i krokach uczenia.

Nie zaobserwowałem wpływu wielkości kroku uczenia na średnią jakość modelu dla danego problemu i modelu o danej strukturze. Na rysunkach 1 i 2 przedstawiono wpływ kroku uczenia na dokładność i funkcję kosztu.

Duża wartość kroku uczenia pozwala modelowi na szybsze uczenie, mała wartość kroku uczenia pozwala modelowi na znalezienie lepszego rozwiązania kosztem liczby niezbędnych epok. Jednak bardzo mały krok uczenia może spowodować utknięcie w lokalnym minimum [1].

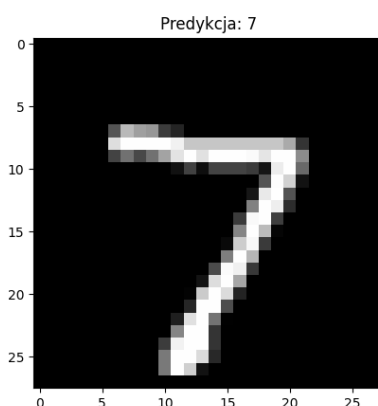
Liczba epok uczenia pozytywnie wpływa na jakość modelu, ale w pewnym momencie występuje efekt nasycenia, gdzie kolejne epoki nie dają znaczącej poprawy jakości. Przedstawiono to na rysunku 1.

5 Sprawdzenie działania klasyfikatora przy uwzględnianiu zniekształceń

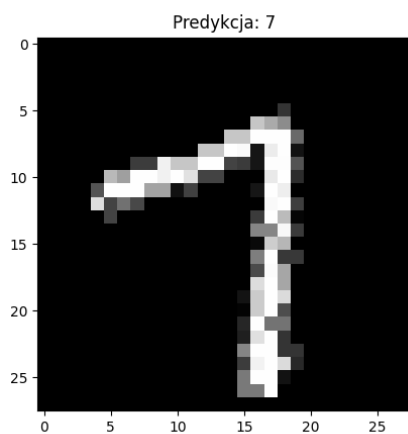
Powyższe testy przeprowadzono na najlepszym modelu dla 100 epok, czyli *opt=Adagrad*, *lr=0.1*, *act=sigmoid*.

Przetestowano klasyfikator na testowym obrazie dokonując następujących zniekształceń:

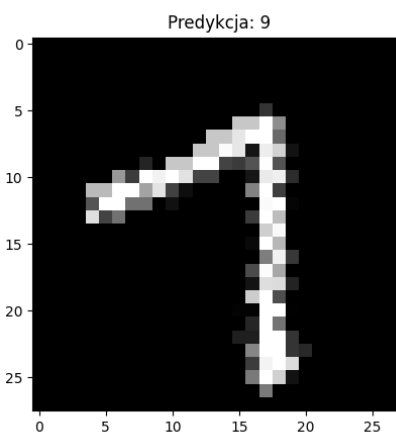
- cyklicznego przesunięcia obiektu
- rotacji obiektu
- dodania szumu



Rysunek 3: Obraz oryginalny

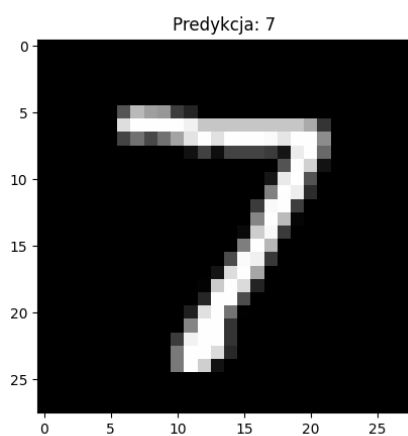


Rotacja 25 stopni

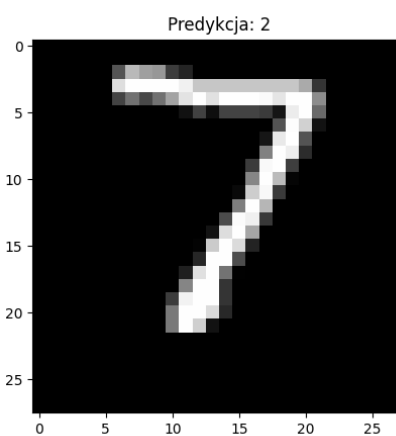


Rotacja 30 stopni

Rysunek 4: Wpływ rotacji na predykcje

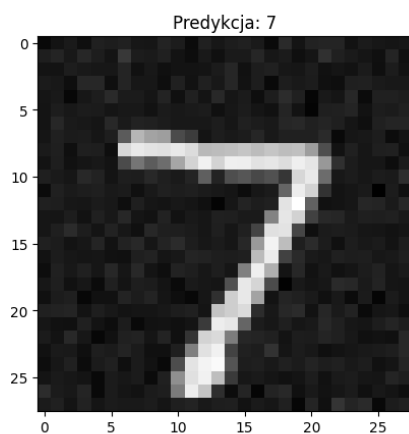


Przesunięcie 2 piksele w osi Y

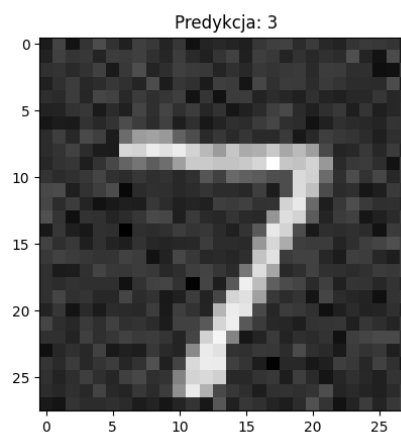


Przesunięcie 5 pikseli w osi Y

Rysunek 5: Wpływ cyklicznego przesunięcia na predykcje



Dodanie szumu od 0 do 10



Dodanie szumu od 0 do 20

Rysunek 6: Wpływ szumu na predykcje

5.1 Wnioski

Testowany klasyfikator nie radzi sobie z rotacjami, przesunięciami czy szumem, co przedstawiono na powyższych przykładach.