

WSI LISTA 3

Bohdan Tkachenko 256630

1 Zadanie 1

1.1 Wstęp

Zadanie polegało na stworzeniu i wytrenowaniu sieci neuronowej do rozpoznawania ręcznie pisanych cyfr z bazy danych MNIST za pomocą biblioteki TensorFlow.

1.2 Metodologia

Zastosowano prostą architekturę sieci neuronowej, składającą się z trzech warstw:

1. Warstwa **Flatten**, która przekształca format obrazów z dwuwymiarowej tablicy (28 x 28 pikseli) do jednowymiarowej (784 piksele).
2. Warstwa **Dense** z 128 neuronami, używająca funkcji aktywacji 'relu'.
3. Warstwa **Dense** z 10 neuronami (jeden dla każdej klasy cyfr od 0 do 9), używająca funkcji aktywacji 'softmax', która zwraca prawdopodobieństwo przynależności do każdej klasy.

Model został skompilowany z optymalizatorem 'adam' i funkcja straty 'sparse_categorical_crossentropy'. Jako metrykę wybrano 'accuracy'.

Model został wytrenowany na zbiorze treningowym przez 5 epok.

1.3 Wyniki

Po wytrenowaniu modelu, dokonano oceny na zbiorze testowym. Dokładność modelu na zbiorze testowym wyniosła 97.44%.

1.4 Wnioski

Model osiągnął wysoką dokładność na zbiorze testowym, co wskazuje, że jest w stanie skutecznie rozpoznawać ręcznie pisane cyfry. Prosta architektura sieci neuronowej okazała się wystarczająca dla tego zadania. W przyszłości można by rozważyć zastosowanie bardziej złożonych modeli, takich jak sieci konwolucyjne, które mogą być w stanie wykryć bardziej skomplikowane wzorce w obrazach.

2 Zadanie 2

2.1 Wstęp

Zadanie polegało na stworzeniu własnego zbioru testowego na podstawie własnych próbek pisma i sprawdzeniu, jak sieć neuronowa stworzona w poprzednim zadaniu radzi sobie z tym zbiorem.

2.2 Metodologia

Zbiór testowy został stworzony poprzez wczytanie obrazów z folderu 'images', przekształcenie ich do skali szarości, zmianę rozmiaru na 28x28 pikseli i normalizację wartości pikseli do zakresu $[0,1]$.

Architektura sieci neuronowej jest taka sama jak w poprzednim zadaniu, z dodatkową warstwą Dropout po warstwie Dense z 128 neuronami, która pomaga zapobiegać przetrenowaniu sieci przez losowe wyłączanie neuronów podczas treningu.

Model został wytrenowany na zbiorze treningowym MNIST przez 10 epok.

2.3 Wyniki

Dokładność modelu na standardowym zbiorze testowym MNIST wyniosła 98.12%.

Jednakże, dokładność modelu na własnym zbiorze testowym wyniosła tylko 10%, co jest znacznie niższe.

2.4 Wnioski

Wyniki sugerują, że model nie radzi sobie dobrze z rozpoznawaniem ręcznie pisanych cyfr z własnego zbioru testowego. Może to wynikać z różnic w stylu pisania cyfr w porównaniu do stylu używanego w zbiorze MNIST. Może to również wynikać z różnic w jakości obrazów.

3 Zadanie 3

3.1 Wstęp

Zadanie polegało na implementacji sieci neuronowej z jedną warstwą ukrytą, zawierającą k neuronów. Sieć ta miała możliwość wyboru między brakiem normalizacji, normalizacją L1 i L2, a także możliwość wyboru między funkcjami aktywacji ReLU i sigmoid.

3.2 Metodologia

Zbiór danych został wygenerowany losowo, z wartościami wejściowymi z przedziału $[-1, 1]$. Oczekiwana wartość wynosiła 0, jeśli iloczyn dwóch wartości wejściowych był dodatni, a 1 w przeciwnym przypadku.

Sieć neuronowa została zainicjalizowana z losowymi wagami. Wagi te były następnie aktualizowane za pomocą algorytmu propagacji wstecznej z gradientem stochastycznym. Sieć była trenowana przez 10 epok.

3.3 Wyniki

Dokładność modelu na zbiorze testowym wyniosła 88.1%.

3.4 Wnioski

Wyniki sugerują, że sieć neuronowa jest w stanie nauczyć się rozróżniać między dwoma klasami na podstawie iloczynu dwóch wartości wejściowych. Jednakże, dokładność modelu może być zależna od wyboru funkcji aktywacji i metody normalizacji, a także od liczby neuronów w warstwie ukrytej.