Wprowadzenie do sztucznej inteligencji, Lista 2

Radosław Wojtczak

254607

0.1 Wprowadzenie

Celem zadania było wykorzystanie zbioru danych **THE MNIST DATABASE of handwritten digits** i przy pomocy biblioteki **TensorFlow** należało stworzyć i wytrenować sieć neuronową rozpoznającą cyfry z podanego zbioru.

0.2 Rozwiązanie

Ów program został napisany w języku programowania python, przy pomocy narzędzia "Google Colab".
Na samym początkowy dokonujemy importu wszystkich potrzebnych bibliotek, w tym wcześniej wspomnianego **TensorFlow'a**. Przygotowujemy odpowiednio dane i dzielimy je na dwie grupy:

- *train*- odpowiedzialne za trenowanie sieci neuronowej
- *test*- testowy zbiór danych.

Będziemy pracować według następującego przepływu roboczego: najpierw zajmiemy się treningiem sieci neuronowej na danych treningowych, następnie sieć nauczy się kojarzyć obrazi z etykietami i wygeneruje swoje przewidywania dotyczące zbioru test. Dodatkowo sprawdzimy jak powyższa sieć poradzi sobie z specjalną serią danych przygotowaną przez autora tego sprawozdania, jak i innej osoby trzeciej.
Utworzona sieć składa się z sekwencji trzech warstw *Dense*, które są ze sobą połączone w sposób gęsty. Trzecia z warstw jest dziesięcioelementową warstwą *softmax*- warstwa ta zwróci tablicę 10 wartości prawdopodobieństwa (suma wszystkich tych wartości jest równa 1). Każdy z tych wyników określa prawdopodobieństwo tego, że na danym obrazie przedstawiono daną cyfrę (obraz może przedstawiać jedną z dostępnych dziesięciu cyfr). Dodatkowo na etapie kompilacji dochodzi do ustalnie odpowiedniej funkcji straty, optymalizatora oraz metryki monitorowania podczas trenowania i testowania (za to wszystko odpowiada wywołanie *model.compile*).
Następnie przeprowadzamy trenowanie sieci i sprawdzamy jak zareaguje na testowe dane.

0.3 Otrzymane wyniki dla danych z pakietu

Po wykonaniu 3 *Epochów* nasza sieć osiągnęła wynik w okolicach **98,5%** dokładności dla danych treningowych (Uwaga: każde uruchomienie sieci niezacznie wpływa na zmianę ów dokładności), jednakże zauważamy, że ta wartość dla danych testowych jest mniejsza: wynosi **97,5%**. Wynika to między innymi z **nadmiernego dopasowania** (modele uczenia maszynowego mają tendencję do niższej dokładności przetwarzania nowych danych, niż miało to miejsce w przypadku danych treningowych). Mimo wszystko zauważamy, że ów wartość jest na bardzo zadowalającym poziomie.

0.4 Otrzymane wyniki dla danych własnych

Poza danymi z wcześniej wspomnianego pakietu sieć została sprawdzona przy użyciu trzech dodatkowych zestawów danych (każdy z nich składał się z 5 powtórzeń każdej z cyfr). W skład tych pakietów wchodziły:

1. Odręcznie napisane cyfry przez autora sprawozdania
2. Cyfry napisane przy użyciu programu graficznego przez autora sprawozdania
3. Odręcznie napisane cyfry przez osobę trzecią

Każdy z powyższych zestawów znajduje się w plikach źródłowych załączonych wraz ze sprawozdaniem. Poniższe tabele przedstawiają otrzymane wyniki dla powyższych danych:

Liczba	Uzyskane wyniki	Procent dokładności
0	$\{5, 3, 2, 5, 3\}$	0%
1	$\{1, 1, 1, 3, 1\}$	80%
2	$\{2, 2, 2, 6, 3\}$	60%
3	$\{3, 3, 3, 3, 3\}$	100%
4	$\{6, 8, 5, 8, 6\}$	0%
5	$\{5, 3, 5, 5, 3\}$	60%
6	$\{5, 6, 6, 5, 2\}$	40%
7	$\{3, 2, 2, 8, 3\}$	0%
8	$\{3, 3, 5, 8, 3\}$	20%
9	$\{8, 3, 9, 9, 8\}$	40%

Table 1: Przykładowe wyniki dla odręcznych cyfr autora, wartość średnia:38%

Liczba	Uzyskane wyniki	Procent dokładności
0	$\{6, 6, 5, 0, 6\}$	20%
1	$\{1, 5, 3, 3, 1\}$	40%
2	$\{2, 2, 3, 2, 2\}$	80%
3	$\{6, 3, 3, 3, 3\}$	80%
4	$\{6, 6, 5, 6, 6\}$	0%
5	$\{6, 5, 5, 5, 5\}$	80%
6	$\{3, 6, 5, 6, 5\}$	40%
7	$\{6, 2, 8, 2, 8\}$	0%
8	$\{8, 8, 8, 8, 8\}$	100%
9	$\{3, 5, 3, 3, 3\}$	0%

Table 2: Przykładowe wyniki dla "komputerowo napisanych" cyfr autora, wartość średnia:44%

UWAGA: Kolumna **Uzyskane wyniki** przedstawia zbiór odpowiedzi sieci na daną liczbę. Każda z liczb została powtórzona pięc razy, dlatego wynikiem jest zbiór zawierający pięć różnych odpowiedzi sieci. Dodatkowo otrzymane wyniki z każdym uruchomieniem lekko się od siebie różnia, w tabelach są przedstawione przykładowe wyniki dla jednego uruchomienia

Liczba	Uzyskane wyniki	Procent dokładności
0	$\{0, 0, 6, 3, 6\}$	40%
1	$\{8, 3, 8, 8, 1\}$	20%
2	$\{8, 2, 3, 2, 2\}$	60%
3	$\{3, 6, 3, 3, 3\}$	80%
4	$\{1, 3, 7, 8, 7\}$	0%
5	$\{5, 5, 5, 5, 6\}$	80%
6	$\{6, 6, 6, 5, 6\}$	80%
7	$\{3, 3, 7, 5, 3\}$	20%
8	$\{5, 3, 3, 3, 3\}$	0%
9	$\{3, 2, 3, 3, 3\}$	0%

Table 3: Przykładowe wyniki dla odręcznie napisanych cyfr osoby trzeciej, wartość średnia:38%

0.5 Interpretacja wyników i wnioski

Zauważamy, iż dla naszych danych sieć poradziła sobie bardzo słabo, procentowa wartość dokładności mieściła się w przedziale $[40, 45]\%$ dla większości wykonanych testów. Tak masywna różnica w otrzymanych rezultatach wynika z różnic w zapisie cyfr między Europejczykami a Amerykanami. Najłatwiej można to zaobserwować na przykładzie cyfry **4**, którą sieć w większości przypadków myli z **6**, czy z **9**, która jest nagminnie mylona z cyfrą **3**. Istnieją również przypadki odwrotne, z którymi nasza sieć radzi sobie z bardzo dużą skutecznością- mowa tu o cyfrach takich jak **3**, **5**, czy **2**.
Wniosek: Przy trenowania sieciu neuronowych należy uważnie wziąć pod uwagę dane, które będą wykorzystywane do jej trenowania jak i testowania. Przy danych do trenowania, które będą pochodzić np. z innych kręgów kulturowych niż dane do testowania, wytrenowana sieć neuronowa może mieć spory problem z poprawnym dopasowaniem otrzymanych danych względem trenowanych wzorców. Dodatkowo trzeba uważać na takie zjawiska jak **nadmierne dopasowanie**, które również negatywnie wpływają na wyniki osiągane przez sieci neuronowe