

Wstęp do sztucznej inteligencji 1

Wojciech Typer

Zadanie 1

Do rozwiązania zadania pierwszego wykorzystałem bibliotekę tensorflow. Jako funkcję aktywacji użyłem funkcji swish:

$$f(x) = x \cdot \sigma(x) = x \cdot \frac{1}{1 + e^{-x}}$$

Jako optymalizatora użyłem Adam (Adaptive Moment Estimation). Algorytm ten aktualizuje wagi sieci podczas trenignu, żeby zmniejszyć straty. Jako funkcję straty użyłem sparse categorical crossentropy, która używana jest do klasyfikacji wieloklasowej, gdy etykiety podane są jako liczby.

Model został wytrenowany na 8 epokach. Oto jego wyniki:

- dokładność: 0.9777
- precyzja: 0.9777
- czułość: 0.9775

```
Epoch 1/8
1875/1875 ————— 5s 2ms/step - accuracy: 0.8674 - loss: 0.4508
Epoch 2/8
1875/1875 ————— 4s 2ms/step - accuracy: 0.9628 - loss: 0.1280
Epoch 3/8
1875/1875 ————— 5s 3ms/step - accuracy: 0.9756 - loss: 0.0789
Epoch 4/8
1875/1875 ————— 5s 2ms/step - accuracy: 0.9835 - loss: 0.0533
Epoch 5/8
1875/1875 ————— 5s 2ms/step - accuracy: 0.9873 - loss: 0.0408
Epoch 6/8
1875/1875 ————— 5s 3ms/step - accuracy: 0.9912 - loss: 0.0294
Epoch 7/8
1875/1875 ————— 5s 3ms/step - accuracy: 0.9926 - loss: 0.0239
Epoch 8/8
1875/1875 ————— 5s 3ms/step - accuracy: 0.9949 - loss: 0.0190
313/313 ————— 0s 1ms/step
accuracy: 0.9777
precision: 0.9777
recall: 0.9775
```

Zadanie 2

W drugim zadaniu, model wytrenowany na bazie MNIST osiągnął następujące wyniki:

- dokładność: 0.4333
- precyzja: 0.4881
- czułość: 0.4333

Model nie radził sobie przede wszystkim z rozpoznaniem 9, 8, 6 i 0. Tak prezentują się jego odpowiedzi:

0.1.png	0	5
0.2.png	0	0
0.3.png	0	3
1.1.png	1	1
1.2.png	1	1
1.3.png	1	5
2.1.png	2	2
2.2.png	2	2
2.3.png	2	3
3.1.png	3	3
3.2.png	3	3
3.3.png	3	5
4.1.png	4	5
4.2.png	4	9
4.3.png	4	4
5.1.png	5	5
5.2.png	5	5
5.3.png	5	5
6.1.png	6	8
6.2.png	6	8
6.3.png	6	0
7.1.png	7	7
7.2.png	7	2
7.3.png	7	7
8.1.png	8	3
8.2.png	8	3
8.3.png	8	3
9.1.png	9	6
9.2.png	9	6
9.3.png	9	6

Dla eksperymentu, model przeuczony, wytrenowany na 60 epokach, poradził sobie z rozpoznaniem moich cyfr gorzej:

- dokładność: 0.4000
- precyzja: 0.3362
- czułość: 0.4000

Zadanie 3

Dane podzieliłem następująco: 0.8 na trening i 0.2 na testy. random state = 42 to ziarno generatora liczb pseudolosowych - dzięki temu za każdym razem podział danych będzie jednakowy, dzięki czemu będziemy dostawać powtarzalne wyniki. Model oparty na RandomForestClassifier ma 100 drzew decyzyjnych. Osiągnął on następujące wyniki:

- dokładność: 0.9673
- precyzja: 0.9671
- czułość: 0.9671