

Sprawozdanie z zadania 1 - PyTorch

Języki programowania do zastosowań biomedycznych, gr. 2

Zadanie: Wykonać polecenia zawarte w linku poniżej:

https://github.com/jarek-pawlowski/machine-learning-applications/blob/main/mnist_in_3_flavours.ipynb

Opis zadania:

Celem zadania było zapoznanie się z aspektami teoretycznymi i praktycznymi biblioteki PyTorch w języku Python. Zwieńczeniem części praktycznej było zadanie projektowe, umieszczone w linku powyżej.

Celem projektu jest stworzenie kompletnego pipeline'u do rozpoznawania cyfr z zestawu danych MNIST z wykorzystaniem sieci konwolucyjnej (CNN) zaimplementowanej w bibliotece PyTorch.

Model uczy się klasyfikować obrazy cyfr (0–9) na podstawie ich reprezentacji pikselowej (28×28 pikseli).

Rozwiązanie:

Do rozwiązania zadania użyto kodu z ww. linku. Dokonano modyfikacji i dodano własne funkcje, zgodnie z założeniami zadania. Kod znajduje się w pliku **main.py**.

Kolejne punkty zadania z ww. linku:

- *apply some regularization technique to Deep model in order to avoid overfitting (start with nonzero weight_decay in optimizer)*

Odpowiedź: Dodano parameter weight_decay:

optimizer = optim.Adam(model.parameters(), lr=model_args['lr'], **weight_decay=1e-4**)

- *try to figure out why the validation loss for CNN model turns to be lower than the train loss (hint: turn off regularization)*

Odpowiedź: Validation loss bywa niższy niż training loss, gdy regularyzacja (np. dropout, L2) działa tylko podczas treningu.

Po wyłączeniu regularyzacji (lub jej braku), różnica znika, a model zachowuje się zgodnie z oczekiwaniami.

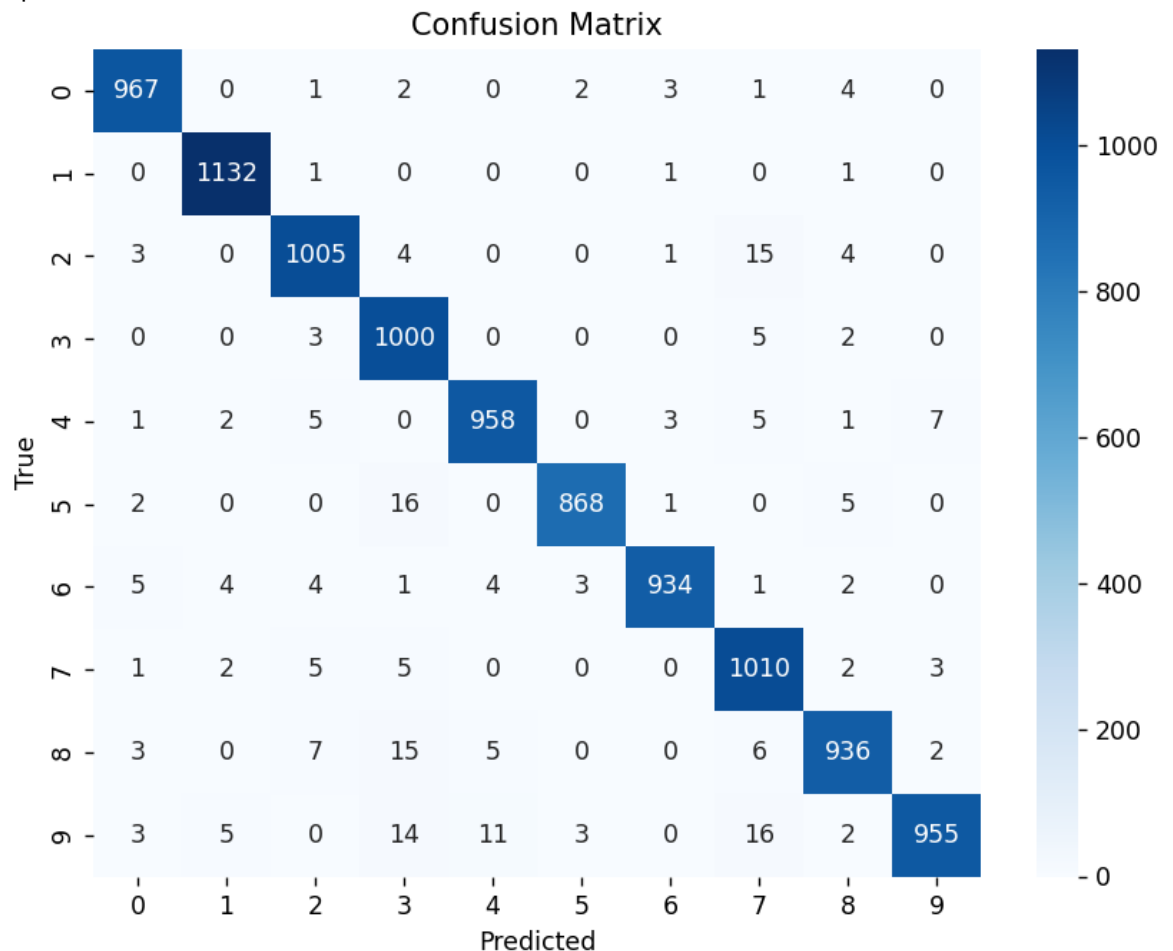
- **tune one of these models to get the *Test Set Accuracy* > 99%,**

Odpowiedź:

```
Validation: Average loss: 0.0339, Accuracy: 9894/10000 (98.94%)
Test: Average loss: 0.0263, Accuracy: 9907/10000 (99.07%)
```

- plot the *confusion matrix* among all of the classes---which of digits are mostly confused with each other?

Odpowiedź:



Na macierzy błędów łatwo zauważyć, że najczęściej mylone są **5 z 3**, **2 z 7** oraz **8 z 3**.

Podsumowanie:

- Architektura CNN została poprawiona pod kątem dokładności i stabilności
- Brak przeuczenia, zachowana generalizacja
- Osiągnięto > 99% dokładności na zbiorze testowym
- Model działa szybko na GPU, ma pełne wsparcie dla walidacji i testu