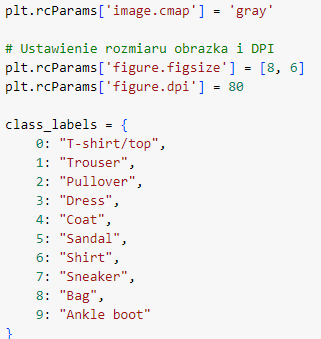
**Z lewej czy z prawej**

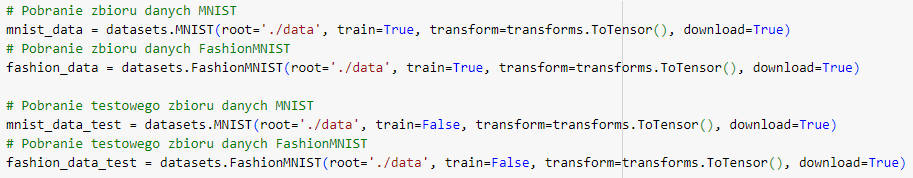
**Bartosz Dudek**

Do obliczeń wykorzystałem PyTorch’a.

1. Ustawienie matplotliba oraz stworzenie listy z kategoriami ubrań

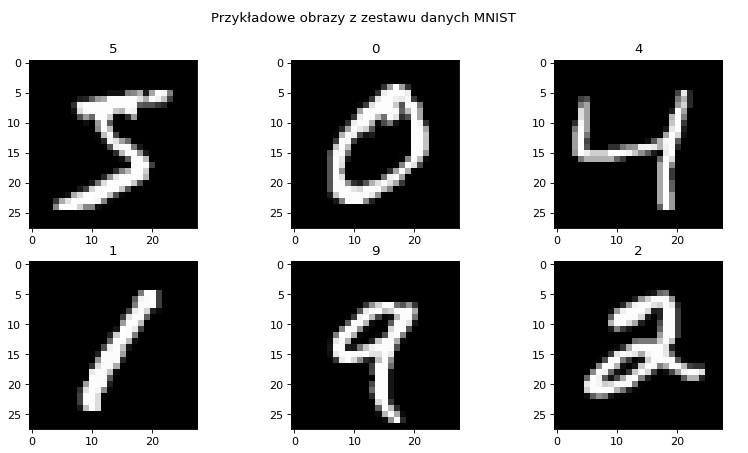


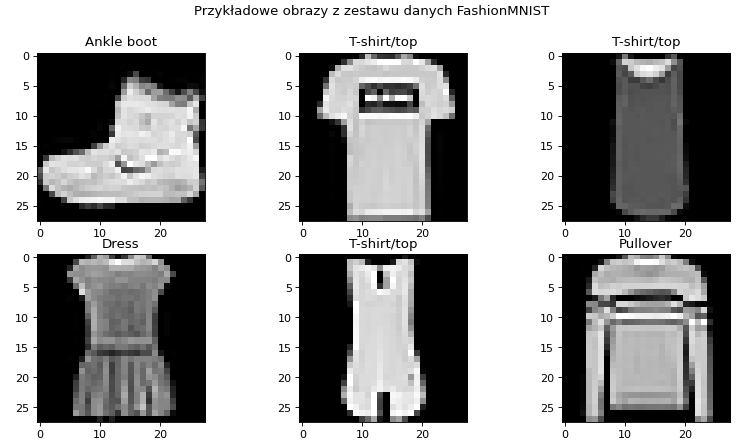
1. Wczytanie zbioru treningowego i testowego z pytorch’a



Raz jest train=False a raz train=True

1. Wyświetlanie grafik z opisami



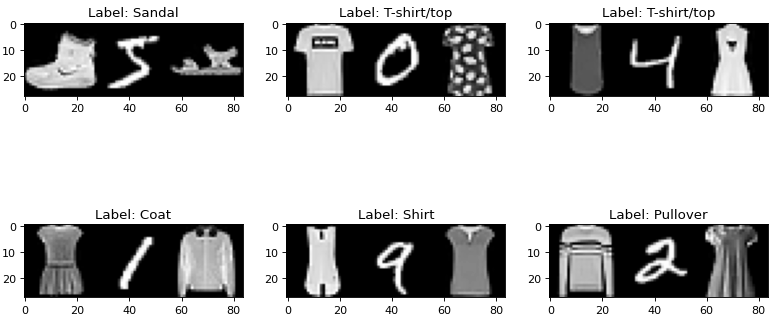


1. Stworzenie odpowiedniego zbioru treningowego

W moim przypadku zrozumiałem plecenie tak, że obrazki z prawej i lewej nie mogą być takie same czyli muszą mieć różne etykiety. Zatem oprócz przemieszania jednego ze zbiorów usunąłem elementy gdzie etykiety się powtarzają na tych samych indexach (zawszę tracę okoł 10% danych ale 54tyś to nadal duzo)



1. Przykłady ze stworzonego zbioru danych



1. Sieć została zaprojektowana zgodnie z danymi w zadaniu - jedyna różnica dropout 0.45 zamiast 0.5
2. Trening sieci



Stworzenie dataloadera



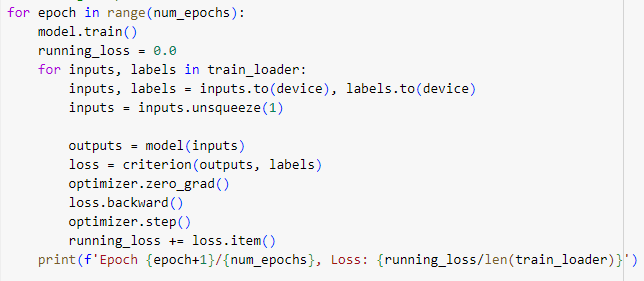
Trening na gpu jeśli dostępne



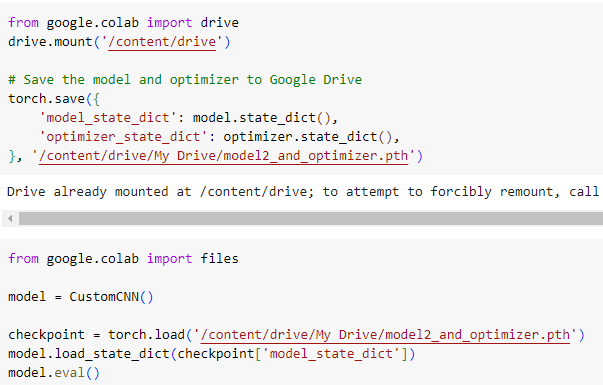
Optymalizator i funkcja kosztu



Ustawienie modelu w tryb treningu, trenowanie przez 20 epok

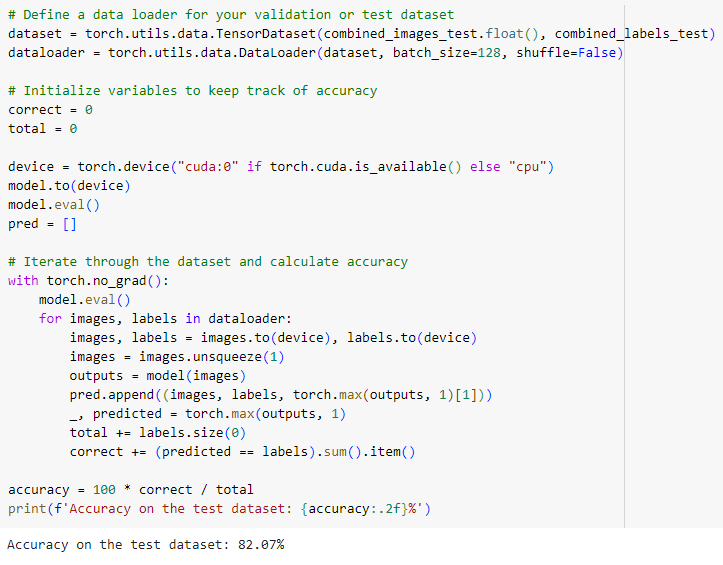


1. Zapis i odczyt modelu i optymalizatora ( w razie potrzeby zakończenia sesji)



1. Obliczenie pozycji na zbiorze testowym

Podobnie jak podczas trenowania tylko ustawiamy na modelu .eval() tryb do testowania.



Precyzja ok 82% jest nieźle.

1. Sprawdzenie kilku przewidzianych etykiet dla obrazków

