Podczas sprawdzania i poszukiwania najlepszego modelu zmieniałem takie hiperparametry jak:

* Optymalizator
* BatchSize
* Liczba Warstw Neuronowych Gęstych (Dense)
* Liczba Neurnów w warstwach gęstych
* Liczba epok

W najlepszym moim modelu udało mi się podnieść dokładność modelu do 0.9185 z domyślnych 0.9127(występującym tylko zbiorze uczcącym) oraz z domyślnych 0.9102(przy podziale na zbiór uczący i walidacyjny).

Podane przeze mnie wartości są z historii fitowania modelu.

Uzyskałem to poprzez zmianę optymalizatora na SGD, dodanie podziału na zbiór walidacyjny (20% zbioru danych) i uczący (80% zbioru danych), dodaniu warstw z neuronami gęstymi, dodanie minibatcha oraz zwiększeniu liczby epok, dzięki czemu sieć poprawnie klasyfikuje ubrania przedstawione w teście który znajduje się w notatniku.

Warstwy spłaszczającej nie zmieniałem.

Mój najlepszy model posiada następujące hiperparametry (znajduje się on w folderze model1):

* Optymalizator: SGD
* BatchSize: 32
* Liczba Warstw Neuronowych Gęstych (Dense): 5
* Liczba Neurnów w warstwach gęstych (numery odpowiadają warstwom)

1. 128
2. 64
3. 32
4. 16
5. 10

* Liczba epok: 27
* Validation split 0.2

Przetestowałem również takie modele z hiperparametrami, które również dawały dobre rezultaty klasyfikacji ubrań w teście oraz dobrze się uczyły:

1.Model drugi (znajduje się on w folderze model2):

* Optymalizator: Adam (learing rate 0.0009)
* BatchSize: 16
* Liczba Warstw Neuronowych Gęstych (Dense): 3
* Liczba Neurnów w warstwach gęstych (numery odpowiadają warstwom)

1. 128
2. 32
3. 10

* Liczba epok: 8
* Validation split 0.2

2.Model drugi (znajduje się on w folderze model3):

* Optymalizator: SGD (learing rate 0.005)
* BatchSize: None
* Liczba Warstw Neuronowych Gęstych (Dense): 5
* Liczba Neurnów w warstwach gęstych (numery odpowiadają warstwom)

1. 128
2. 64
3. 32
4. 16
5. 10

* Liczba epok: 37
* Validation split 0.2

Przetestowałem również wiele innych modeli z hiperamatrami, które nie dawały dobrych rezultaty klasyfikacji ubrań w teście, ale w miarę dobrze się uczyły (patrząc na funkcję błędu walidacyjnego i funkcję błędu). Jako przykład takiego modelu podaję model znajdujący się w folderze model4:

* Optymalizator: ADAM (learing rate 0.0009)
* BatchSize: None
* Liczba Warstw Neuronowych Gęstych (Dense): 3(między warstwą nr 1 i 2 znajduję się warstwa porzucająca o współczynniku porzucania 0.2)
* Liczba Neurnów w warstwach gęstych (numery odpowiadają warstwom)

1. 128
2. 64
3. 10

* Liczba epok: 10
* Validation split 0.2