

Najlepsze wyniki są osiągane, gdy jest dodana kolejna warstwa neuronów.

Dodanie kolejnej warstwy gęstej zwiększa szybkość uczenia kosztem czasu uczenia i zwiększa też dokładność modelu i spadek funkcji błędu.

Używanie optymalizatora Adam sprawia, że przy tych samych wartościach współczynnika uczenia model się nie uczy jako powinien przy tej samej ilości epok (100) (następuje przy początkowych epokach gwałtowny spadek funkcji błędu (walidacyjnej i treningowej), a potem funkcja błędu jest praktycznie stała i niewielkimi wahaniami), reasumując model się szybko uczy na początku i jest nauczony, ale potem już nie jest w stanie zaktualizować tak wag sieci by jeszcze zmniejszyć wartość funkcji błędu.

Gdy używamy optymalizatora sgd (modele w folderach hiperparametry 1, 2 i 3) spadek funkcji błędu (walidacyjnej i treningowej jest widoczny podczas całego procesu uczenia (na przełomie wszystkich epok)

Zmiana współczynnik uczenia przy optymalizatorze SGD sprawia, że model uczy się wcześniej (hiperparametry 2 i 3) niż przy domyślnym współczynniku uczenia zastosowanym w optymalizatorze SGD.

Mój najlepszy model posiada następujące hiperparametry (znajduje się on w folderze hiperparamtry 3):

- Optymalizator: SGD (learning rate 0.2)
- Liczba Warstw Neuronowych Gęstych (Dense): 2
- Liczba Neuronów w warstwach gęstych (numery odpowiadają warstwom)
  1. 3
  2. 1
- Liczba epok: 100
- Validation split 0.2

Przetestowałem również takie modele z hiperparametrami, które również dawały dobre rezultaty klasyfikacji punktu do danego gangu oraz dobrze się uczyły:

1. Model pierwszy (znajduje się on w folderze hiperparametry 1):

- Optymalizator: SGD (learning rate 0.1)
- Liczba Warstw Neuronowych Gęstych (Dense): 1
- Liczba Neuronów w warstwach gęstych (numery porządkowe odpowiadają numer warstwom)
  1. 3
- Liczba epok: 100
- Validation split 0.2

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

- Optymalizator: SGD (learning rate 0.2)
- Liczba Warstw Neuronowych Gęstych (Dense): 1
- Liczba Neuronów w warstwach gęstych (numery porządkowe odpowiadają numer warstwom)
  1. 3
- Liczba epok: 100
- Validation split 0.2

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

- Optymalizator: Adam (learning rate 0.1)
- Liczba Warstw Neuronowych Gęstych (Dense): 2
- Liczba Neuronów w warstwach gęstych (numery porządkowe odpowiadają numer warstwom)
  1. 3
  2. 1
- Liczba epok: 100
- Validation split 0.2