Metody głębokiego uczenia - Konwolucyjne sieci neuronowe

Jadwiga Słowik

Wykorzystane narzędzia

- Google colab
- Jupyter-notebook
- Python
- Keras
- Scikit-learn

Zastosowane podejście

- Ze zbioru uczącego wydzielenie zbioru treningowego i walidacyjnego w stosunku 9:1
- Badanie wpływu wybranych hiperparametrów

Początkowa konfiguracja

- Ustawienie stałej uczenia na wartość 0,01
- Wybór optymalizatora (**SGD**, RMSProp, Adagrad, Adam)
- Wybór liczby warstw w MLP - zastosowanie więcej niż dwóch warstw ukrytych pogarsza wyniki klasyfikacji
    - wybrałam dwie warstwy ukryte po 128 neuronów (funkcja aktywacji ReLU)

Kolejne kroki część 2

- Wybór konfiguracji konwolucyjno-poolingowej
  - 128 Conv3x3
  - ReLU
  - ((128 Conv2x2, ReLU) * 3), MaxPooling2x2) * 4
  - Fully connected layer (128, 128, ReLu)
- Uzyskana skuteczność:
  - Dla 5 epok: 62%
  - Dla 40 epok: 76%

Zastosowanie metod regularyzacji

- **Kernel regularizer** dla warstwy fully connected i warstw konwolucyjnych
- **Dropout** (prawdopodobieństwo wygaszenia połączenia 0,2) dla każdej warstwy
- **Batch Normalization**

Wyniki: zastosowanie pierwszej metody spowodowało spadek o 20 pkt. proc. Natomiast zastosowanie dwóch ostatnich - poprawa wyniku o kilka punktów procentowych (accuracy rzędu 79%).

Próba zastosowania data augmentation

- Width_shift_range = 0,1
- Shear_range = 0,2
- Zoom_range = 0,3
- Horizontal_flip, vertical_flip

Niestety nie odnotowałam poprawy w jakości klasyfikacji, od pewnego momentu (~70 epoka) accuracy oscylowało wokół 78%.

Wybór stałej uczenia

- Zastosowanie zmiennej stałej uczenia:
  - [1,30): learning rate równy 0,01
  - [30, 60): learning rate równy 0,005
  - [60, inf): learning rate równy 0,003

  Zastosowany zabieg nie poprawił skuteczności na zbiorze walidacyjnym

Druga próba zastosowania data augmentation

- Rotation_range = 30
- Width_shift_range i width_shift_range równe 0,1
- horizontal_flip i vertical_flip

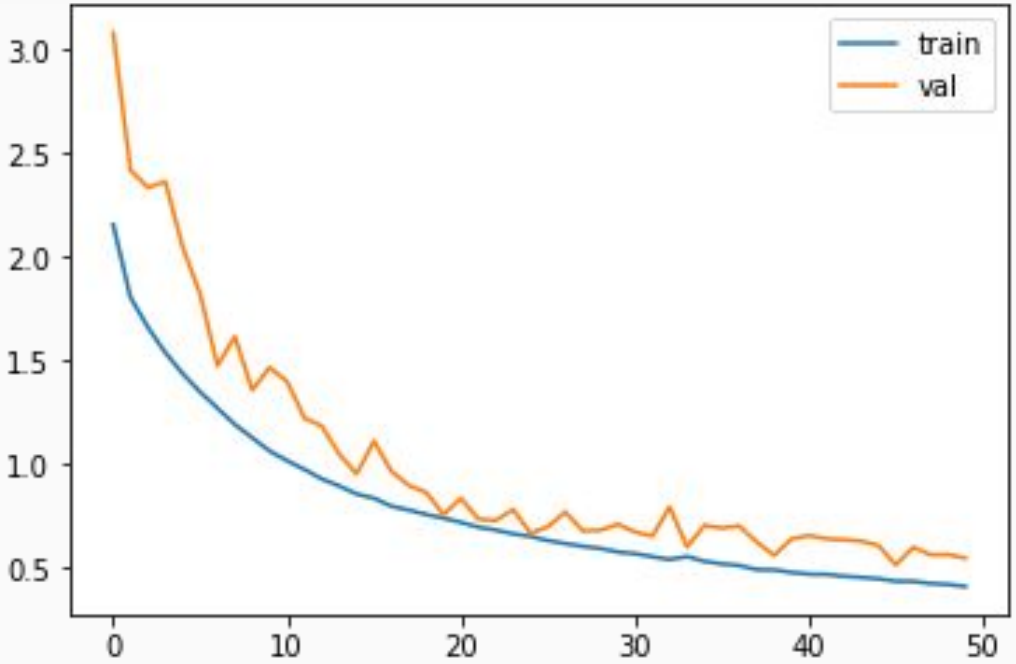
Nadal accuracy na zbiorze walidacyjnym ok. 80%

Ostateczna architektura (accuracy na zbiorze testowym 81,91%)

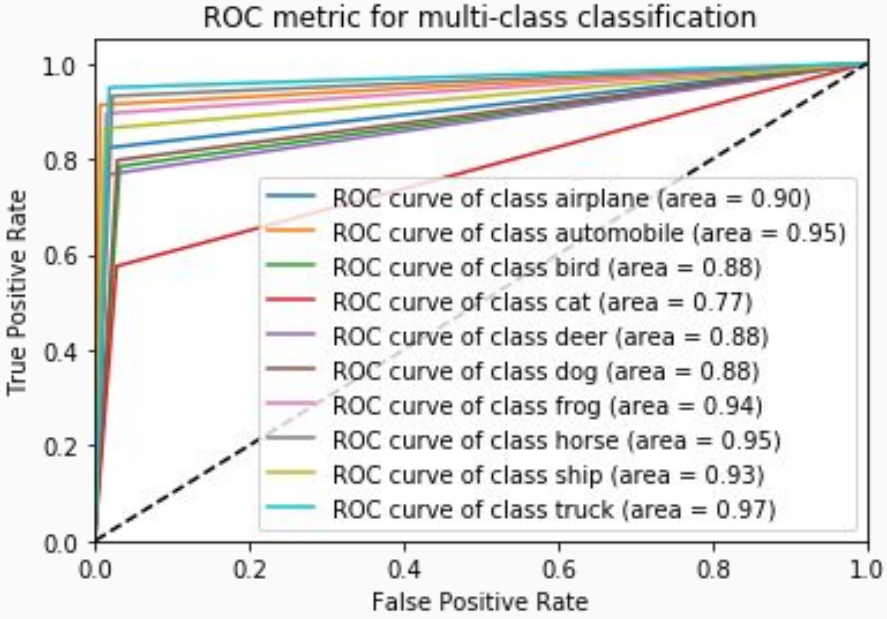
- 128 Conv3x3, przesunięcie 1, ReLu
- 4 warstwy MaxPooling2x2, przesuniecie równe 2
  - (128 Conv2x2, ReLU, Dropout (0,2), BatchNorm) * 3
- (Fully-Connected (128), Dropout (0,2), BatchNorm) * 2
- Softmax 10

50 epok, stałe learning rate = 0,01

Funkcja kosztu



ROC AUC


ROC metric for multi-class classification
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class airplane (area = 0.90)
ROC curve of class automobile (area = 0.95)
ROC curve of class bird (area = 0.88)
ROC curve of class cat (area = 0.77)
ROC curve of class deer (area = 0.88)
ROC curve of class dog (area = 0.88)
ROC curve of class frog (area = 0.94)
ROC curve of class horse (area = 0.95)
ROC curve of class ship (area = 0.93)
ROC curve of class truck (area = 0.97)