

# CIFAR100 classification project

Agnieszka Żaba

## 1 Przygotowania modelu

Do klasyfikacji danych ze zbioru CIFAR100 stworzyłam sieć kolnwolucyjną. Ze względu na ograniczenia czasowe dla uczenia na początku sieć była bardzo krótka. Udało się jednak na Colabie włączyć opcję korzystania z GPU, dzięki czemu zwiększyły się możliwości. Na początku stworzyłam prostą sieć zawierającą 3 warstwy konwolucyjne z funkcją aktywacji ReLu. Jako funkcję kosztu od samego początku wykorzystałam Sparse Categorical Crossentropy.

Uczenie wywoływałam wielokrotnie zmieniając za każdym razem któryś z parametrów. Początkowo chciałam dostosować learning rate, ale ze względu na ograniczenie czasowe zostałam przy domyślnym  $lr = 0.001$ . Przy zmniejszonym kroku po 10 epokach otrzymałam lepsze wyniki na zbiorze treningowym, ale wciąż na zbiorze walidacyjnym nie były zadowalające.

Ustawiłam więc rozmiar batcha jako 32 i dodałam normalizację batchów po każdej warstwie konwolucyjnej. Udało się przyspieszyć proces uczenia. Dodatkowo, żeby zminimalizować rozbieżność między zbiorem treningowym i walidacyjnym dodałam regularyzację. Na początku L1L2, ale samo l2 miało lepsze wyniki.

Aby uniknąć overfittingu dodałam także dropout z parametrem 0.5. Najlepsze wyniki, accuracy około 0.58 na testowym i 0.49 na walidacyjnym, otrzymałam wykorzystując sieć, która miała ponad 2.5 mln parametrów.

## 2 Konstrukcja sieci

Augmentacja danych - rotacja, przesunięcie w poziomie, poziome odbicie

Optimizer - Adam

Learning rate - 0.001

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 32, 32, 32)	896
batch_normalization (Batch Normalization)	(None, 32, 32, 32)	128
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 30, 30, 64)	18496
batch_normalization_1 (Batch Normalization)	(None, 30, 30, 64)	256
conv2d_2 (Conv2D)	(None, 28, 28, 64)	36928
batch_normalization_2 (Batch Normalization)	(None, 28, 28, 64)	256
max_pooling2d (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 14, 14, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 12, 12, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_2 (Dropout)	(None, 6, 6, 64)	0
conv2d_4 (Conv2D)	(None, 1, 1, 64)	36928
batch_normalization_4 (Batch Normalization)	(None, 1, 1, 64)	256
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 100)	6500
=====		
Total params: 137,828		
Trainable params: 137,252		
Non-trainable params: 576		

### 3 Wyniki

W modelu bez augmentacji po 100 epokach:

train accuracy = 0.4081  
validation accuracy = 0.4400

W modelu bez augmentacji przez 15 epok:

train accuracy = 0.3601  
validation accuracy = 0.3543

I po uczeniu z augmentacjš od 16 do 100 epoki:

train accuracy = 0.3713  
validation accuracy = 0.4330