# Klasyfikacja dźwięków: Ptak, Nie-Ptak

Dominik Młotkowski
Klaudia Juszczak
Anita Czerniak
Tomasz Krupiński

14 stycznia 2020

Nasze zadanie polega na nauczeniu sieci neuronowej prawidłowo rozpoznawać czy dany dźwięk pochodzi od śpiewu ptaka. Dane potrzebne do realizacji projektu znajdują się na stronie: `http://dcase.community/challenge2018/task-bird-audio-detection`. Są one publicznie dostępne. Wykorzystaliśmy plik "ff1010bird", który zawiera 7690 plików. 7000 danych z tego pliku wykorzystaliśmy jako dane treningowe, przy czym, przy ustalonym parametrze $validation\_split = 0.2$ pobieramy 20% danych walidacyjnych, czyli w tym przypadku 1400 obrazków. Resztę z nich, czyli 690 jako dane testowe.
Każde nagranie trwa 10 sekund. Dodatkowo, do każdego pliku z nagraniami posiadamy plik csv, który wyszczególnia nam nazwy plików oraz odpawiadające im wartości $0 = nie - ptak$ lub $1 = ptak$.

```
In [8]: import pandas as pd
        from pandas import DataFrame
        data = pd.read_csv("./data2/warblrb10k_public_metadata_2018.csv")
        df = DataFrame(data, columns = ['itemid','hasbird'])
        print(df)
```

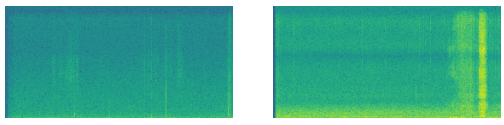
```
                        itemid  hasbird
0     759808e5-f824-401e-9058        1
1     1d94fc4a-1c63-4da0-9cac        1
2     bb0099ce-3073-4613-8557        1
3     c4c67e81-9aa8-4af4-8eb7        1
4     ab322d4b-da69-4b06-a065        0
...                       ...      ...
7995  ca7b3342-17b0-444f-ba2a        1
7996  43071f95-d31e-447b-8786        1
7997  0d4d2fea-743d-46aa-a17f        1
7998  0d34160d-55db-4c70-93fa        1
7999  01539aa0-f482-4a71-a944        1

[8000 rows x 2 columns]
```

Rysunek 1

Wszystkie pliki są w formacie WAV oraz nie są przestrzenne (mono). Próbkowanie wynosi 44.1 $kHz$. Zwykle w wyniku próbkowania otrzymujemy sygnał, który wygląda inaczej niż oryginalny. Próbkowanie polega na odczytywaniu chwilowego poziomu sygnału akustycznego i zapisywanie jego wartości w postaci cyfrowej. Zapis dźwięku jest tym lepszy im częściej jest próbkowany. 44.1 $kHz$ oznacza, że próbkowanie odbywa się 44100 razy na sekundę. Aby użyć naszych danych, które są w formacie audio, musieliśmy przerobić je na obrazki, a dokładniej spektogramy. Użyliśmy do tego transformaty Fouriera($FFT$). Transformacja Fouriera to technika matematyczna stosowana do przekształcania danych czasowych w dane o dziedzinie częstotliwości.

$$f(t) = \sum_{w} Ae^{a\pi iwt}, \quad gdzie$$

$$e^{a\pi iwt} = cosa\pi wt + isina\pi wt$$



(a) (b)

Rysunek 2: Spektogramy

**Model:**

W naszym projekcie najtrafniejszym wyborem była konwolucyjna sieć neuronowa (CNN), ponieważ jest ona najlepsza do prac z obrazami jaki i z dźwiękiem.

```
model = Sequential()
model.add(Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(184, 372, 4)))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(2, activation='softmax'))
model.summary()



model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
print("*** Model compiled ***")
```

(a)

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 184, 372, 16)      592
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 184, 372, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 92, 186, 32)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 92, 186, 32)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 547584)            0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 1095170
=================================================================
Total params: 1,100,402
Trainable params: 1,100,402
Non-trainable params: 0
```

(b)

Rysunek 3: Jedna funkcja gęstości $model.add(Dense(2, activation =' softmax'))$

Jest to model, który oparty jest na 3 epokach. Dodatkowo mamy model, który posiada 10 epok i jego skuteczność jest lepsza o 4%, natomiast loss jest mniejszy o 0.100 od powyższego modelu.