

Deep Learning 2 (na 27.10.24)

1. *Wizja komputerowa.*

Wizja komputerowa to dziedzina, która polega na filtrowaniu obrazów w celu znajdowania na nich pożądaných artefaktów/wzorów.

- a. Obrazy są danymi sieci konwolucyjnych. Proszę zapoznać się z dowolnym API zarządzania obrazami (np. najbardziej popularna biblioteka Pillow) i pokazać w kodzie metody: załadowanie obrazu, wyświetlenie, zapis do pliku, wypisanie informacji o obrazie (format, tryb koloru, rozmiar), zmiana rozmiaru, operacje: obrót, odwrócenie, wycinek; przygotowanie obrazu do sieci neuronowej: skalowanie, centrowanie, standaryzacja, itp.
- b. Proszę napisać kod w Pythonie, który ładuje i wizualizuje obraz, wykonuje na obrazie konwolucję zadanym filtrem (lub zestawem filtrów), na koniec wizualizuje i zapisuje obraz wynikowy do pliku. Filtry zapisać jako macierze kwadratowe o nieparzystej długości boku. Jeśli filtr da się zadawać analitycznie, dodatkowym parametrem wywołania powinien być rozmiar. Obraz wynikowy powinien być tej samej wielkości, co obraz wejściowy – jeśli to niemożliwe, należy rozszerzyć obraz wierszami/kolumnami zer na zewnątrz (tzw. 0-padding).
- c. Zaznajomić się z wykładem „Sieci konwolucyjne” i zademonstrować w kodzie działanie konwolucji jako metody. Uwzględnić hiperparametry: Stride (S), 0-Padding (P), rozmiar maski filtru (F). Przy wybranym F i S algorytm ma sam ustalić najmniejsze P, tak, aby obraz wynikowy miał ten sam rozmiar, co wejściowy.

Do kodu dołączyć obraz/y wejściowe użyte w kodzie.

2. *Rozszerzenie danych (data augmentation): klasa ImageDataGenerator z biblioteki Keras.*

Jest to automat do „tworzenia” dodatkowych obrazów. Jak używać tego generatora podczas uczenia?

3. *Sieć CNN.*

Dla wybranego przez siebie zbioru obrazów proszę znaleźć dobry model CNN. Architekturę modelu należy testować „przyrostowo”, obserwując krzywe uczenia (wykres straty i dowolnej metryki w funkcji numeru epoki).

Sieć CNN składa się z pewnej liczby bloków typu (w Keras): Conv2D, MaxPooling2D, Dropout, BatchNormalization. Warstwy te i ich hiperparametry zostały przedstawione na wykładzie „Sieci konwolucyjne”.

4. *Transfer learning.*

Dla wybranego zbioru obrazów proszę przetestować wydajność sieci CNN o znanej architekturze (keras.io/api/applications). Dostosowanie topologii wstępnie wyuczonej sieci do naszego zbioru danych polega na:

- a. dostosowaniu rozmiaru wejścia do konkretnej sieci,
- b. zmianie wyjścia (adekwatnie do liczby klas problemu),
- c. douczeniu sieci na wybranym przez nas zbiorze.

Jakie mogą być różne tryby takiego uczenia?