

# Prototypical networks

## DOKUMENTACJA KOŃCOWA

Zespół 12:

Kamil Dąbrowski, 285629

Łukasz Pietraszek, 323581

### 1. Przyjęte założenia projektowe

Sieci prototypowe opierają się na założeniu, że istnieje osadzenie, w którym kilka punktów skupia się wokół pojedynczej reprezentacji prototypu dla każdej klasy. Jego celem jest nauka prototypów dla poszczególnych klas na podstawie uśredniania próbek w przestrzeni cech.

### 2. Krótki opis wybranych lub przydzielonych zbiorów danych

#### 2.1. Omniglot

Zestaw danych Omniglot jest przeznaczony do opracowywania podobnych do ludzkiego uczenia się algorytmów uczenia się. Zawiera 1623 różnych odręcznych znaków z 50 różnych alfabetów. Każda z 1623 symboli została narysowana online przez 20 różnych ludzi.

#### 2.2. Mini-imagenet

Zestaw danych mini-ImageNet został zaproponowany przez Vinyals et al. do oceny uczenia się za pomocą kilku strzałów. Jego złożoność jest wysoka ze względu na użycie obrazów ImageNet, ale wymaga mniej zasobów i infrastruktury niż praca na pełnym zestawie danych ImageNet. W sumie istnieje 100 klas z 600 próbkami kolorowych obrazów 84×84 na klasę. Te 100 klas podzielono odpowiednio na 64, 16 i 20 klas dla zadań próbkowania do meta-treningu, meta-walidacji i meta-testu.

### 3. Krótki opis wybranych lub przydzielonych architektur

Uczenie się kilku strzałów to rodzaj meta-uczenia się. W przeciwieństwie do tradycyjnych modeli, które wymagają dużej ilości danych, aby uzyskać dobre wyniki, modele z funkcją uczenia się kilku strzałów wymagają tylko niewielkiej ilości danych uczących z ograniczoną ilością informacji. W nauce za pomocą kilku strzałów model jest uczony w zakresie różnych powiązanych zadań podczas fazy meta-treningu, aby uogólnić dobrze na zadania niewidoczne z zaledwie kilkoma wystąpieniami w fazie meta-testowania. Klasyfikacja kilku strzałów to metoda, w której klasyfikator musi zostać dostosowany do nowych klas, których nie widać na treningu i podać tylko kilka przykładów każdej z tych klas. Sieci prototypowe uczą się

przestrzeni metrycznej, w której klasyfikację można przeprowadzić przez obliczenie odległości do prototypowych reprezentacji każdej klasy.

#### 4. Biblioteki wybrane do realizacji projektu

Do rozwiązania zadania wykorzystano język programowania Python w wersji 3.9 wraz z bibliotekami:

- conda-forge::jupyterlab
- conda-forge::jupyterlab-lsp
- conda-forge::python-language-server
- conda-forge::nodejs
- numpy
- pandas
- pip
- scikit-learn
- scipy
- scikit-build
- opencv-python
- tqdm
- Gdown
- cudatoolkit=11.1

#### 5. Przeprowadzone eksperymenty

Eksperymenty przeprowadzono dla datasetów Omniglot i Mini-imagenet zgodnie z opisanymi w artykule parametrami. Architektura składa się z czterech bloki konwolucyjne. Każdy blok zawiera 64-filtrową konwolucję  $3 \times 3$ , warstwę normalizacji wsadowej, nieliniowość ReLU i warstwę max-pooling  $2 \times 2$ . Modele porównano pod względem skuteczności treningu dla 1-shot (5-way Acc.), 5-shot (5-way Acc.), 1-shot (20-way Acc.), 5-shot (20-way Acc.). Do treningu modelu dla zbioru danych Omniglot wykorzystano 600 epok, każdy epizod trenujący zawierał 60 klas i po 5 query points na każdą klasę. Natomiast dla zbioru Mini-imagenet wybrano 64 klasy trenujące, 16 klas walidacyjne i 20 klas testujących.

##### Parametry eksperymentów:

###### # GENERAL PARAMS

- ★ *LERNING\_RATE* – szybkość uczenia, domyślnie 0.001,
- ★ *GAMMA* – gamma harmonogramu nauki modelu, domyślnie 0.5,
- ★ *DECAY\_EVERY* – krok harmonogramu tempa nauki, domyślnie 20,
- ★ *MAX\_EPOCH* – maksymalna liczba epok do wytrenowania domyślnie 500,

###### # NETWORK PARAMS

- ★ *X\_DIM* – wymiar X sieci neuronowej, domyślnie (3, 84, 84),
- ★ *HID\_DIM* – wymiar HID sieci neuronowej, domyślnie 64,

- ★  $Z\_DIM$  – wymiar  $Z$  sieci neuronowej, domyślnie 64,
- ★  $N\_TEST$  – liczba uruchomień zbioru testowego, domyślnie 25,

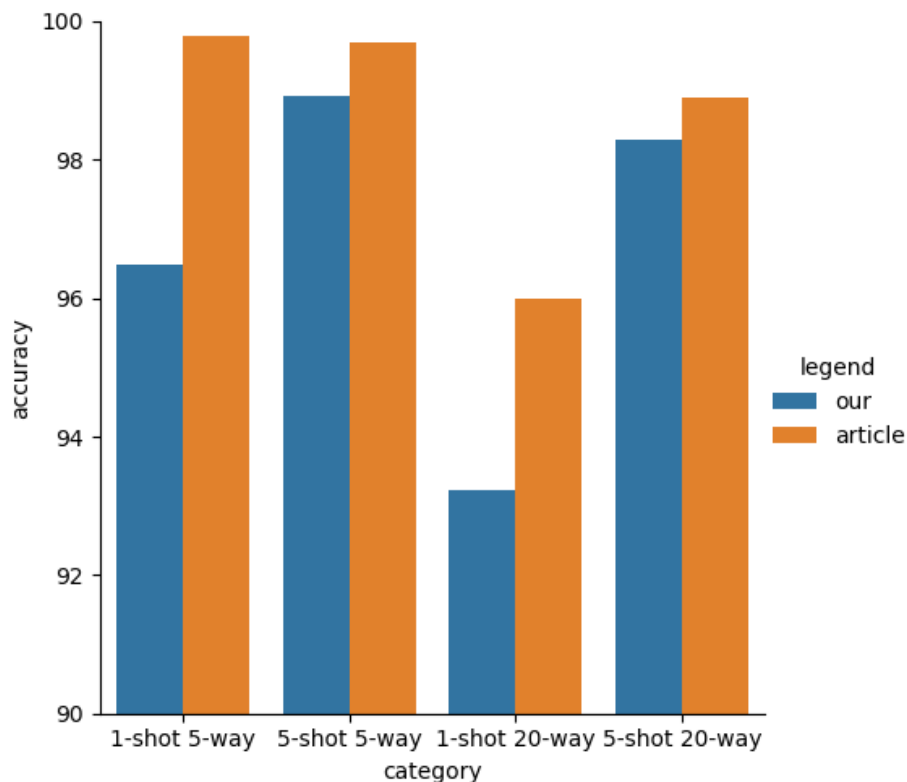
#### # ALGORITHM PARAMS

- ★  $NUM\_WAY$  – droga, wykorzystane w eksperymentach 5 lub 20, domyślnie 20,
- ★  $NUM\_SHOT$  – strzały, wykorzystane w eksperymentach 1 lub 5, domyślnie 1,
- ★  $NUM\_QUERY$  – liczba próbek do wykorzystania na klasę w postaci zapytań walidacyjnych wykorzystane w eksperymentach 5 lub 15, domyślnie 5,
- ★  $EPOCH\_SIZE$  – liczba epok, domyślnie 100,
- ★  $EPOCH\_SIZE\_TEST$  – liczba epok testującej, domyślnie 1000.

## 6. Wyniki eksperymentów

### 6.1. Skuteczność modelu dla datasetu Omniglot.

	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
Osiągnięte wyniki	$96,48 \pm 0,08\%$	$98,92 \pm 0,03\%$	$93,92 \pm 0,05\%$	$98,29 \pm 0,02\%$
Wyniki w artykule	98,8%	99,7%	96,0%	98,9%

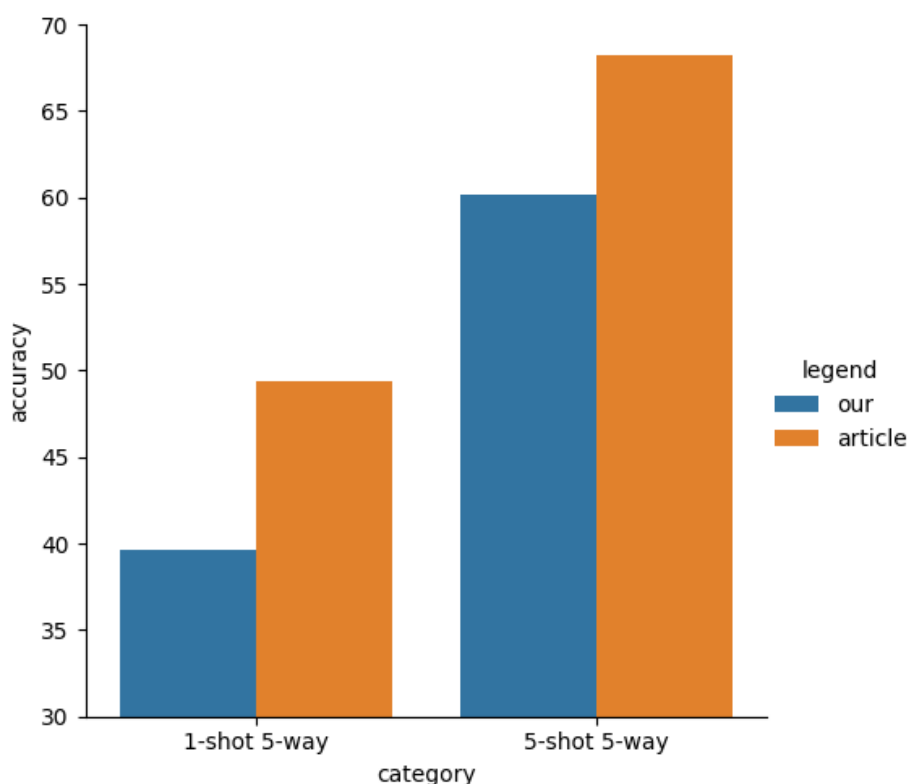


Powyżej zamieszczono tabelę z porównaniem osiągniętych przez nas wyników oraz wyników osiągniętych w artykule. Na wykresie kolorem niebieskim oznaczono nasze wyniki, natomiast kolorem pomarańczowym wyniki osiągnięte w artykule.

Niestety nie udało się osiągnąć identycznych ani tym bardziej lepszych rezultatów niż te przedstawione w artykule, niemniej jednak są one bardzo zbliżone. W przypadku 1-shot zarówno dla 5-way, jak i dla 20-way różnica wynosi około 2 punktów procentowych precyzji na korzyść wyników z artykułu, natomiast w przypadku 5-shot jest to różnica zaledwie niecałego 1 punktu procentowego uzyskanej precyzji na korzyść rezultatów z artykułu.

## 6.2. Skuteczność modelu dla datasetu Mini-imagenet.

	5-way	
	1-shot	5-shot
Osiągnięte wyniki	39,63 ± 0,17%	60,15 ± 0,11%
Wyniki w artykule	49,42 ± 0,78%	68,20 ± 0,66%



Powyżej zamieszczono tabelę z porównaniem osiągniętych przez nas wyników oraz wyników osiągniętych w artykule. Na wykresie kolorem niebieskim oznaczono nasze wyniki, natomiast kolorem pomarańczowym wyniki osiągnięte w artykule.

Niestety w przypadku zbioru danych Mini-imagenet również nie udało się osiągnąć identycznych ani tym bardziej lepszych rezultatów niż te przedstawione w artykule, niemniej jednak są one zbliżone. W przypadku 1-shot dla 5-way różnica wynosi około 10 punktów procentowych precyzji na korzyść wyników z artykułu, natomiast w przypadku 5-shot jest to różnica około 8 punktów procentowych uzyskanej precyzji na korzyść rezultatów z artykułu.

## 7. Wnioski

Zaproponowany przez autorów artykułu pomysł, aby wykorzystać sieci prototypowe do uczenia kilku strzałów okazał się skutecznym i działającym poprawnie pomysłem. Wyuczone sieci za pomocą treningu epizodycznego potrafią z dość dobrą skutecznością ustawić wspomniane strzały.

Uzyskane rezultaty są zadawalające. Udało nam się odtworzyć zaproponowany przez autorów model uczenia za pomocą sieci prototypowych i uzyskać zbliżone wyniki. Rezultaty na zbiorze danych Omniglot są bardziej zbliżone do tych uzyskanych w artykule, jednakże wynika to z tego, że na omawianych danych precyzja wynosi blisko 100% dla 5-way i około 98% dla 20-way. Dataset Mini-imagenet daje nieco gorsze wyniki zarówno w naszych eksperymentach, jak i w artykule (przedział 40-70% precyzji po wytrenowaniu). Przekłada się to na większe różnice na niekorzyść uzyskanych przez nas precyzji w procentach (strata do precyzji osiągniętych w artykule to kolejno około 10 i 8 punktów procentowych dla 1-shot i 5-shot).