Kraków Applied Physics and Computer Science

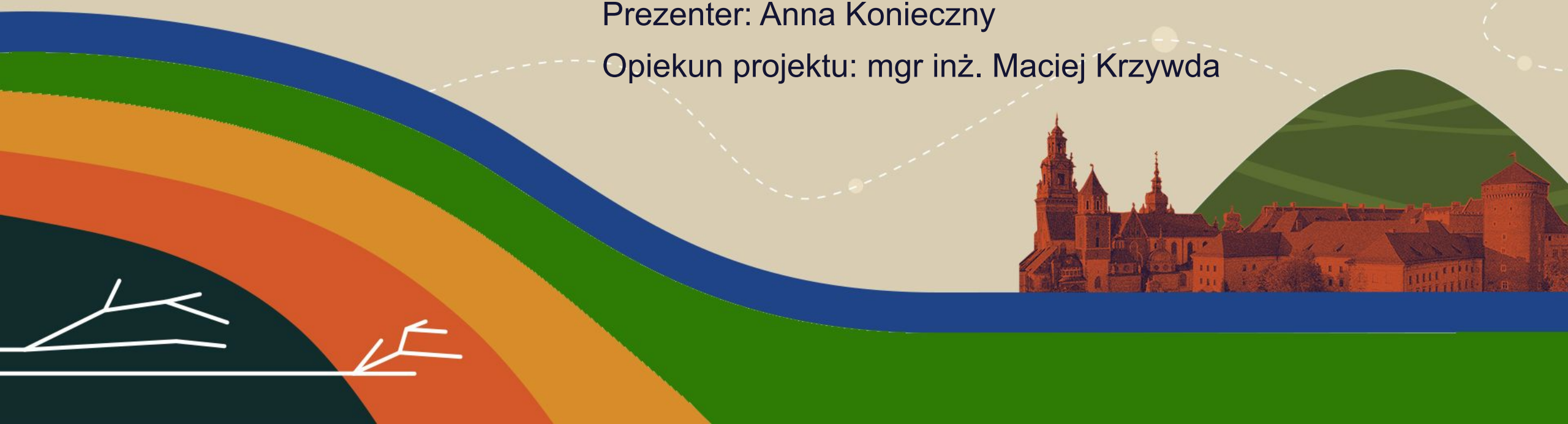
Summer School'22

29th of July 2022

Using Bio-Inspired Algorithms to Optimize Structures of Graph Convolutional Networks

Prezenter: Anna Konieczny

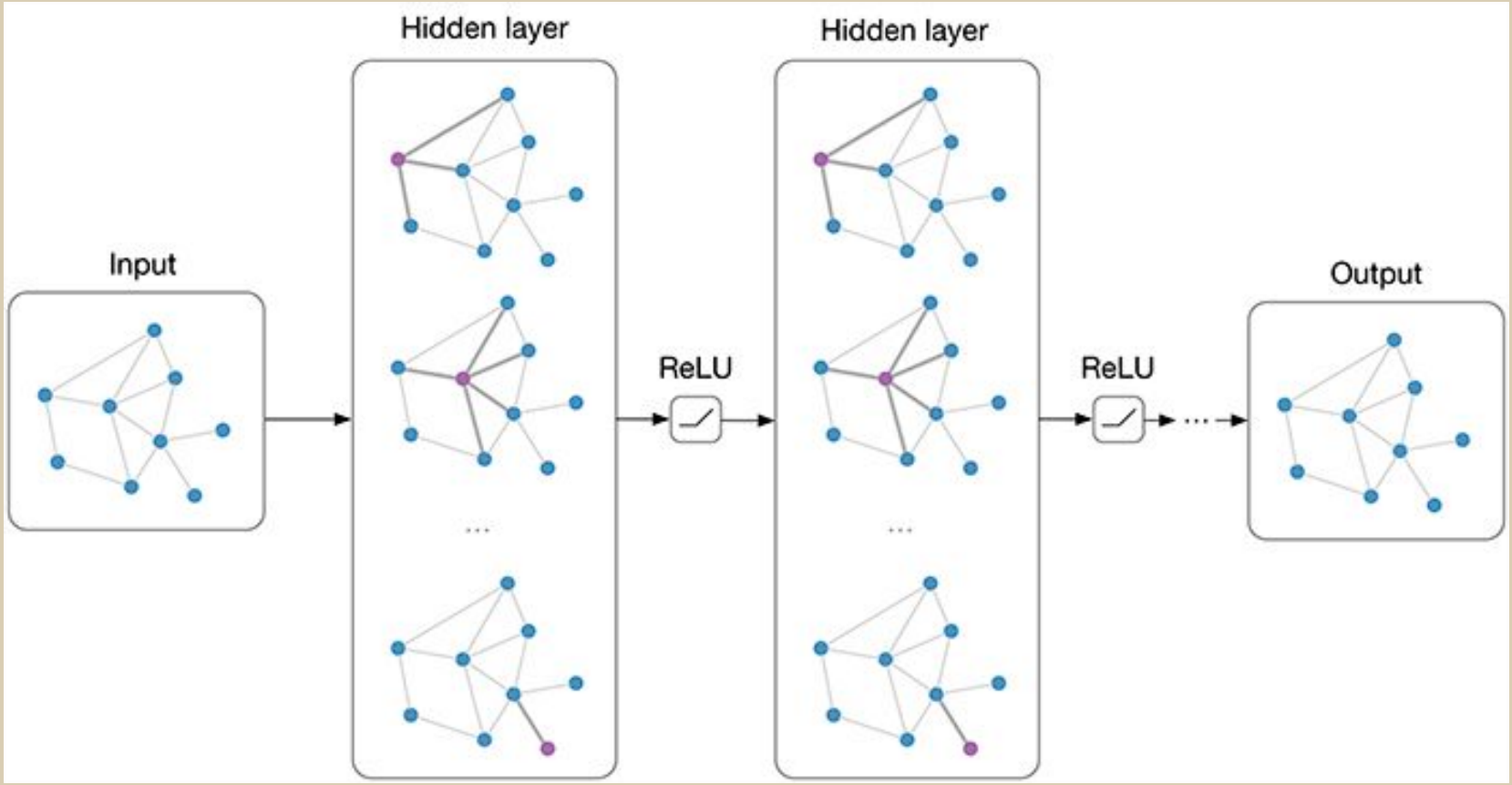
Opiekun projektu: mgr inż. Maciej Krzywda



Plan prezentacji

1. Grafowe konwolucyjne sieci neuronowe - GCN
2. Porównanie sieci GCN oraz CNN
3. Problem optymalizacji architektury
4. Optymalizacja architektury GCN
5. Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury
6. Wykorzystane technologie
7. Wyniki
8. Perspektywy rozwoju

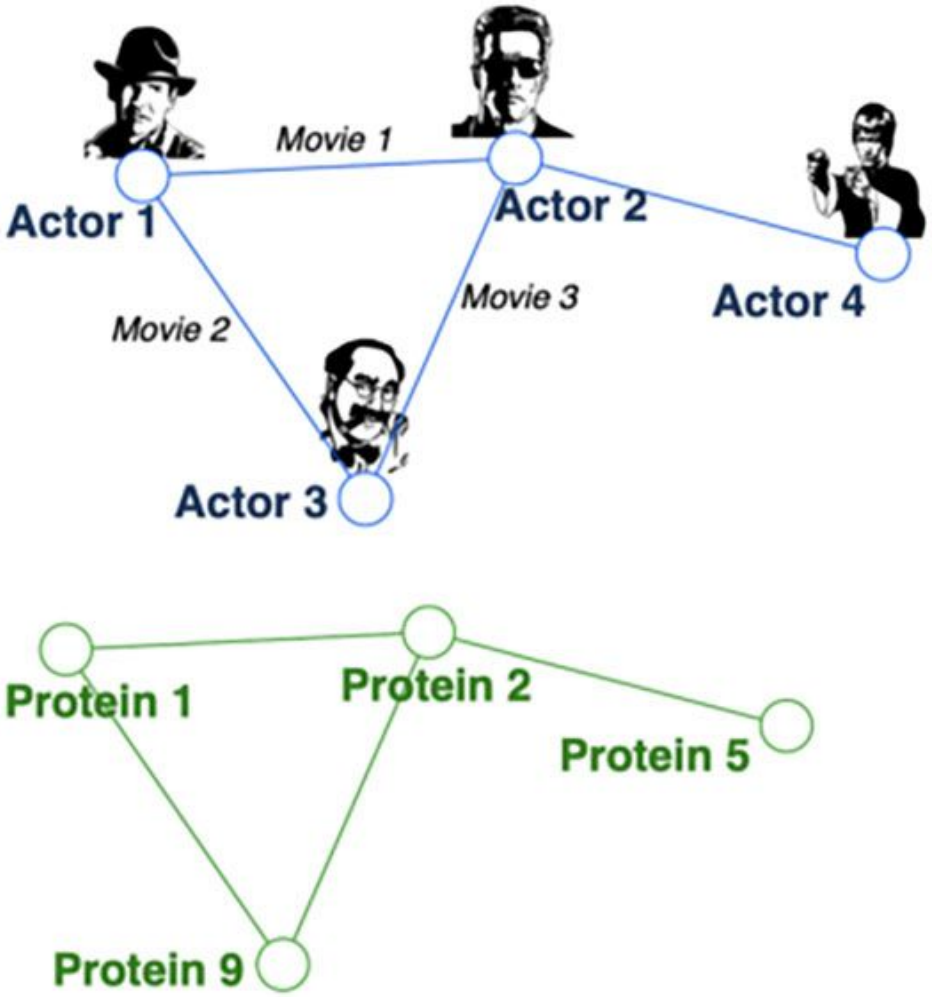
Grafowe sieci konwolucyjne - GCN



Rys.1. Uproszczony schemat grafowej sieci konwolucyjnej [1].

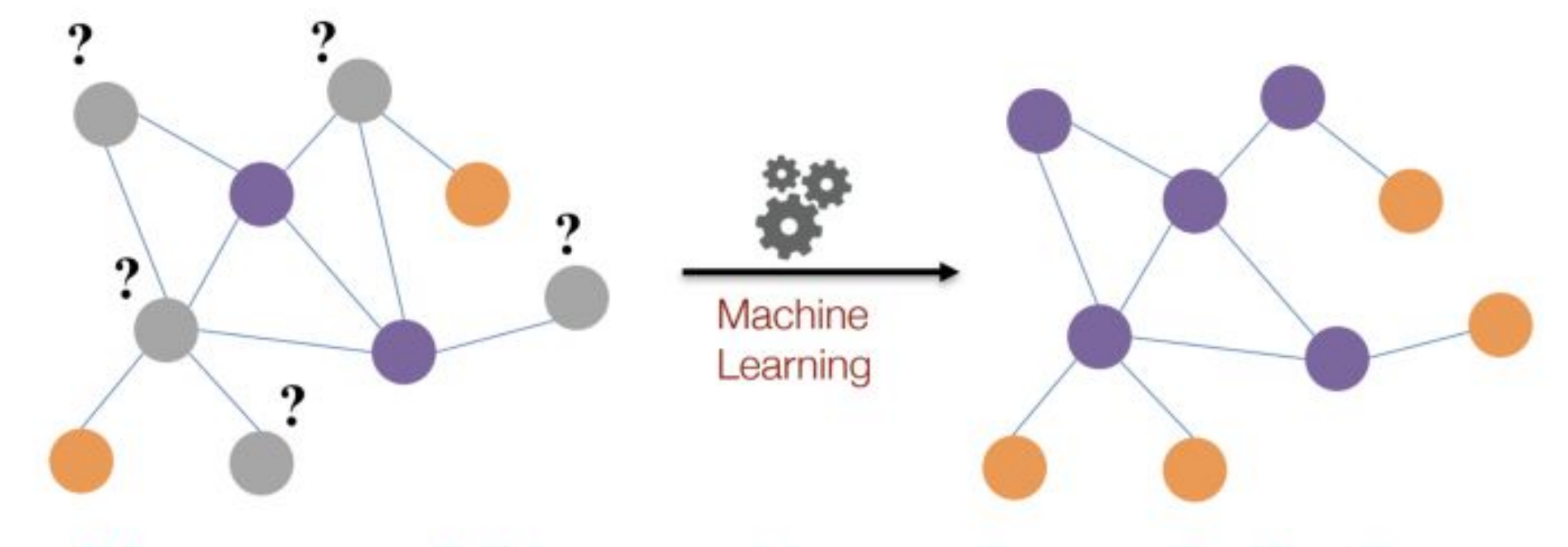
Niektóre z dostępnych zestawów danych z biblioteki pytorch-geometric [2]:

- dane zawierające informacje o interakcjach między proteinami **(PPI)**
- media społecznościowe, np. posty należące do różnych społeczności **(Reddit, Reddit2)**
- dane zawierające produkty ze sklepu Amazon i ich kategorie **(AmazonProducts)**
- dane zawierające 40 943 obiektów, 18 relacji i 151 442 faktów składających się z trzech słów, np. “łóżko jest meblem” **(WordNet18)**
- zestaw danych zawierający 2708 publikacji naukowych zaklasyfikowanych do 7 różnych klas, zawierający 5429 połączeń. Każda publikacja opisana jest wektorem z wartościami zero-jedynkowymi, które wskazują na obecność danego słowa ze słownika **(Cora)**



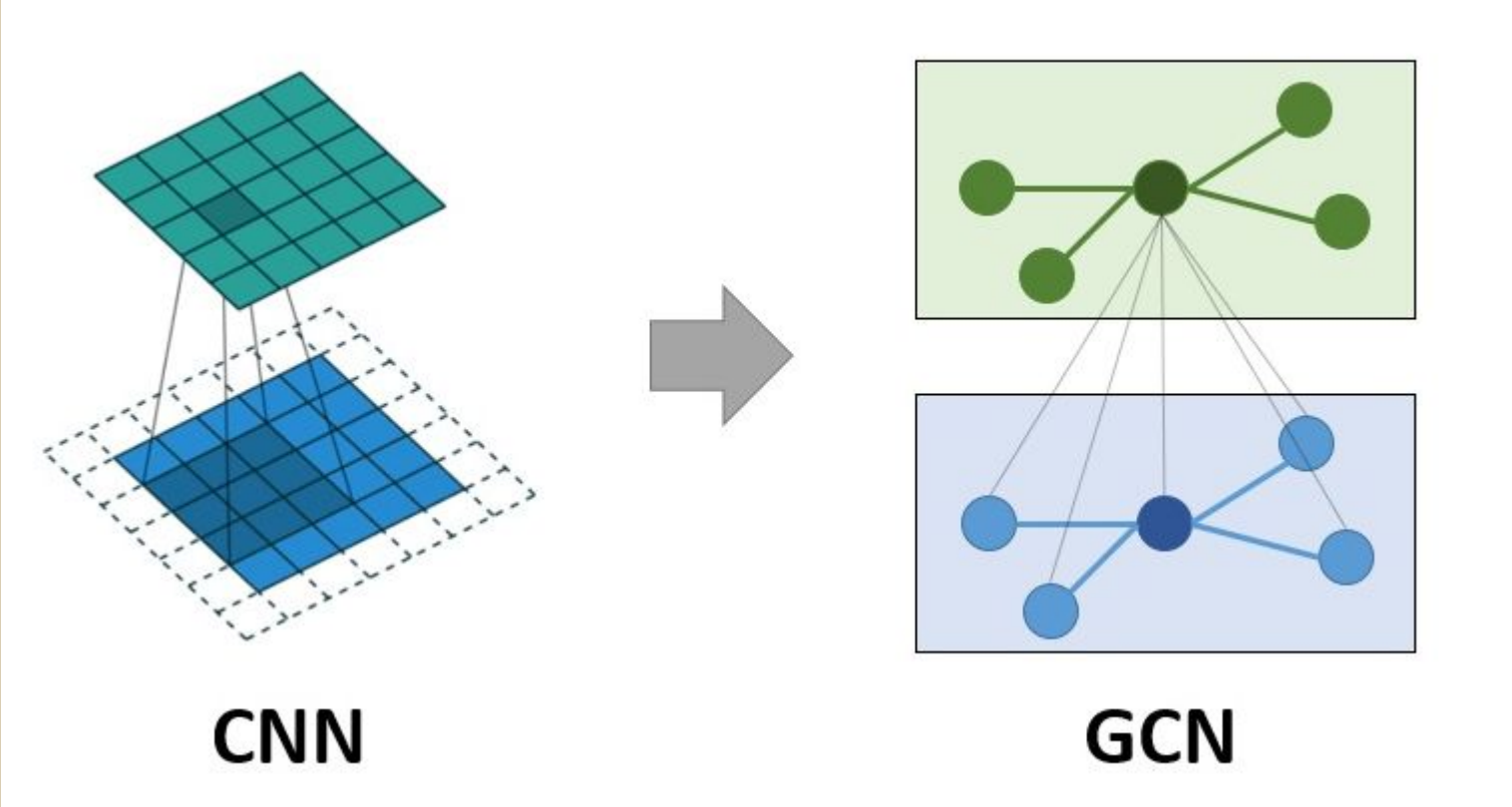
Rys.2. Wizualizacja przykładowych zestawów danych o postaci grafów [2].

Grafowe sieci konwolucyjne - GCN



Rys.3. Problem klasyfikacji węzłów danych o postaci grafów w uczeniu maszynowym [2].

Porównanie sieci GCN oraz CNN



Rys.4. Porównanie działania konwolucyjnej sieci neuronowej oraz grafowej sieci neuronowej [2].

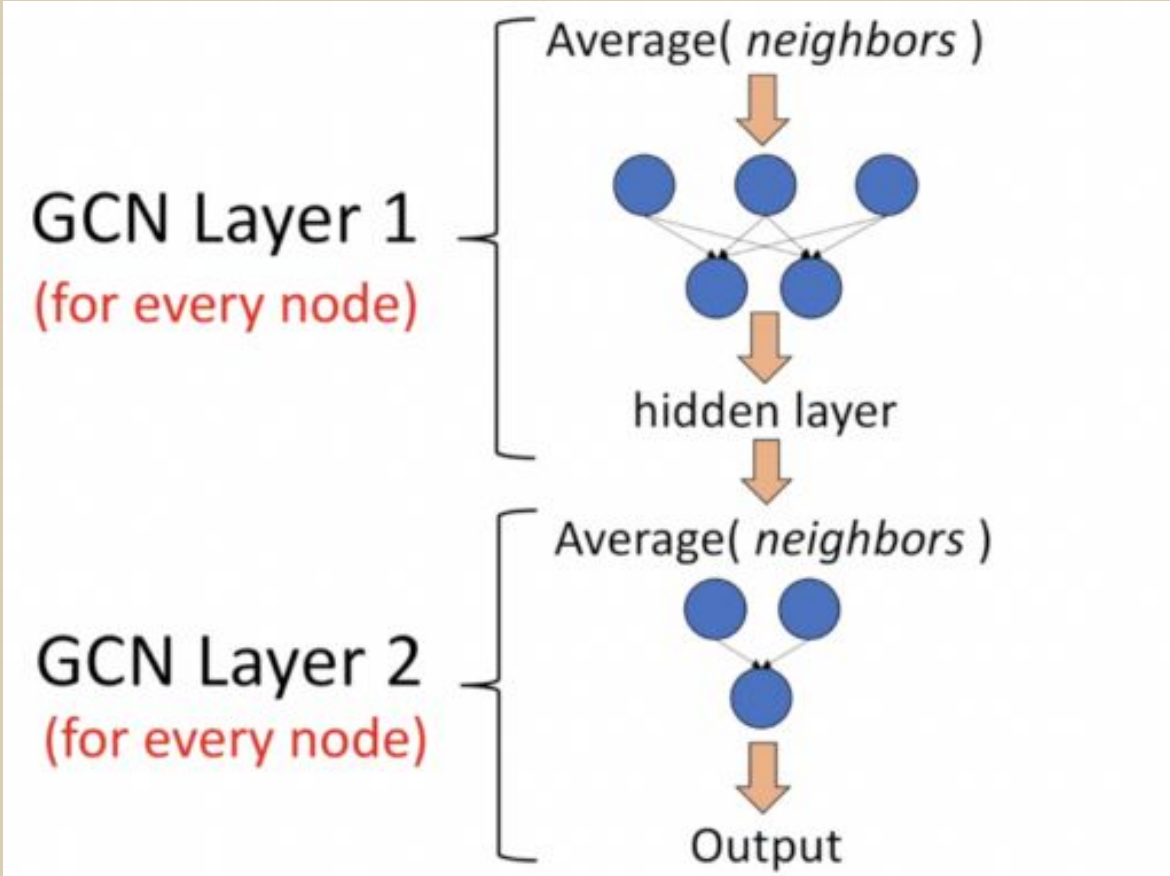
Problem optymalizacji architektury

Poszukiwanie optymalnej architektury sieci neuronowej (Neural Architecture Search - NAS) można podzielić m.in. na następujące etapy:

1. Przeszukiwanie przestrzeni (search space)
2. Metoda optymalizacji
   - algorytmy ewolucyjne
   - uczenie przez wzmacnianie
   - analiza bayesowska
3. Metoda ewaluacji kandydata
   - zredukowanie liczby epok
   - transfer learning

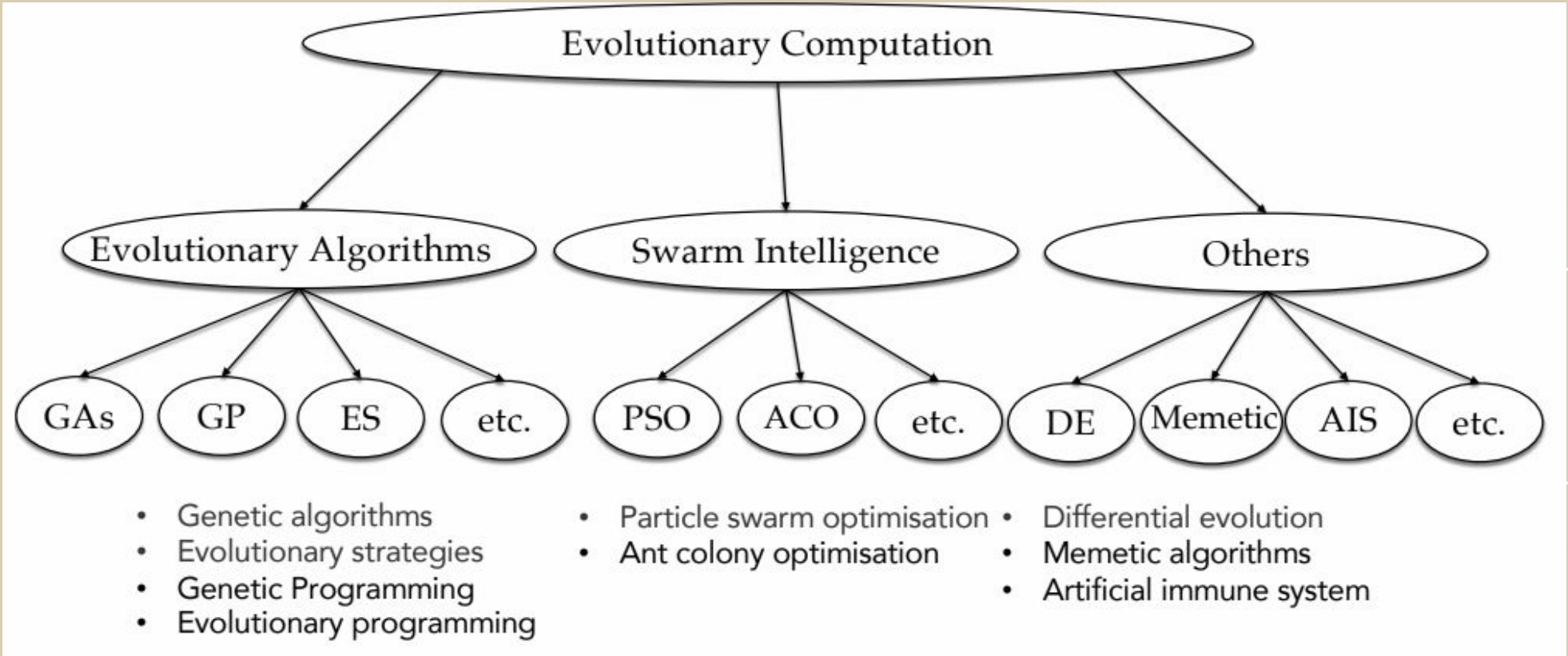
Optymalizacja architektury GCN

- liczba warstw konwolucyjnych: 2 lub 3
- liczba neuronów w warstwie ukrytej: od 20 do 100
- funkcja aktywacji w warstwie ukrytej: ReLU lub LeakyReLU
- obecność lub brak Dropout
- funkcja aktywacji: Linear lub LogSoftmax



Rys.5. Ilustracja obrazująca architekturę grafowej sieci neuronowej [2].

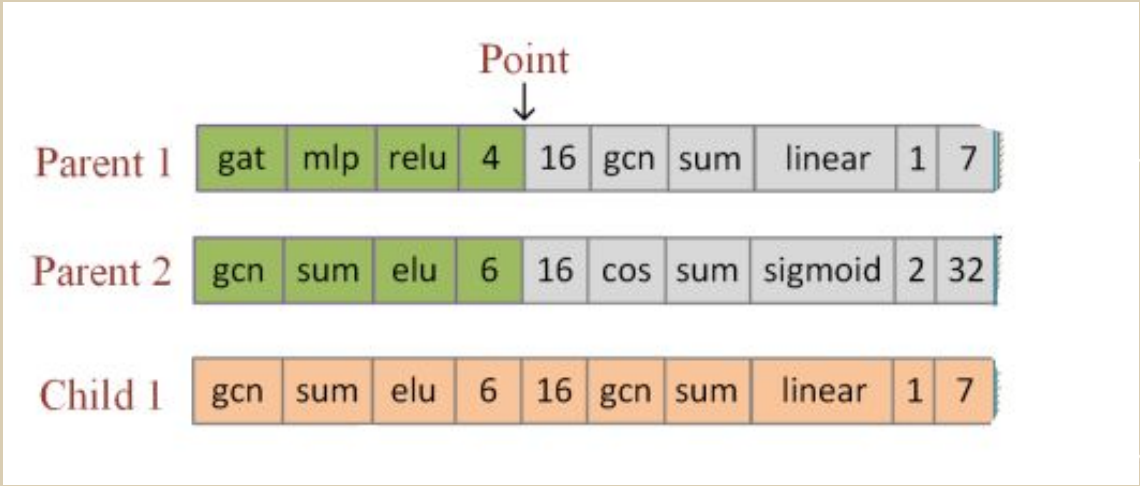
Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury



Rys.6. Ilustracja obrazująca podział dziedzin programowania ewolucyjnego[8].

Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury

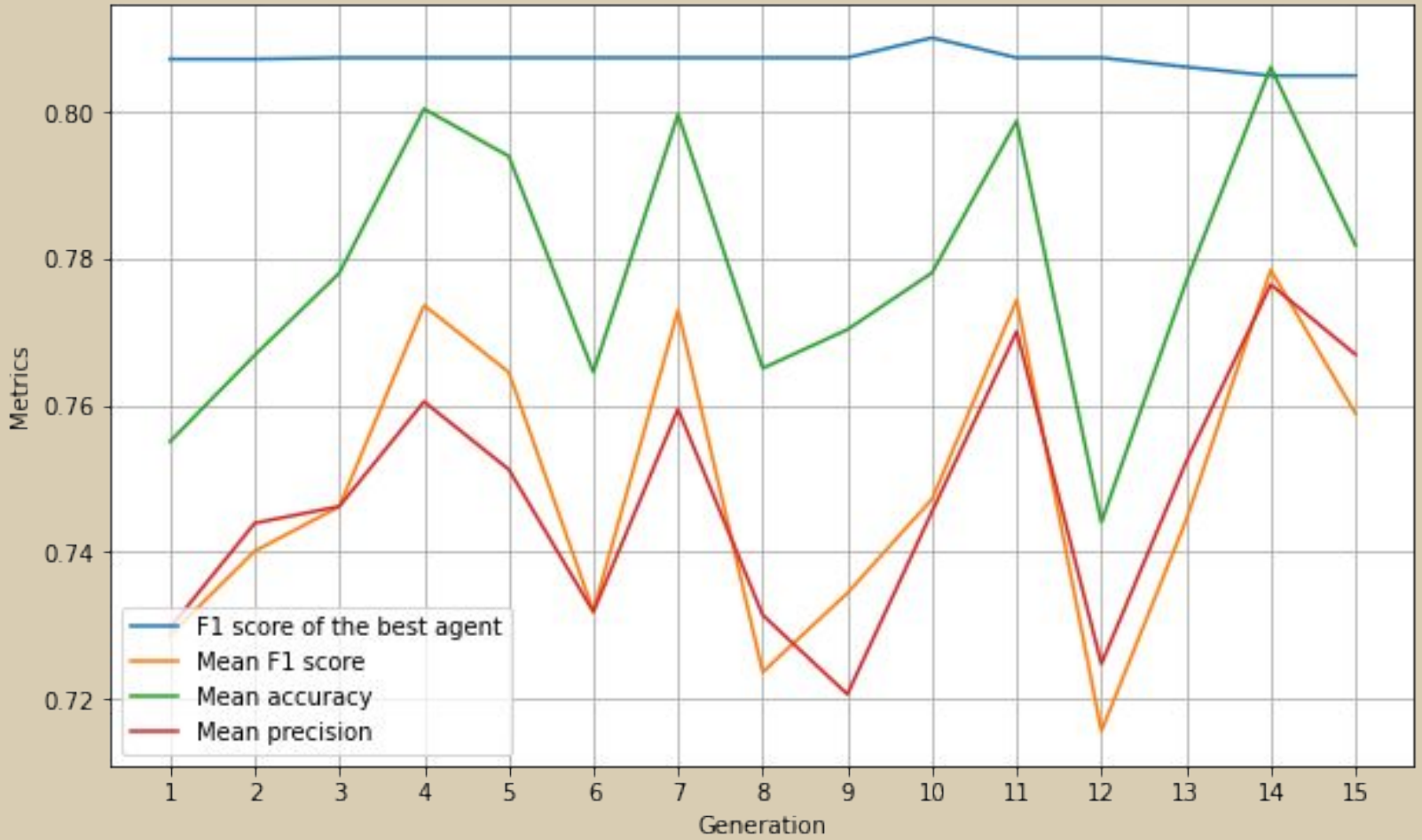
1. Inicjalizacja populacji.
2. Ewaluacja
   - obliczenie funkcji dopasowania
   - selekcja
   - krzyżowanie
   - mutacje
3. Otrzymanie najlepszego osobnika



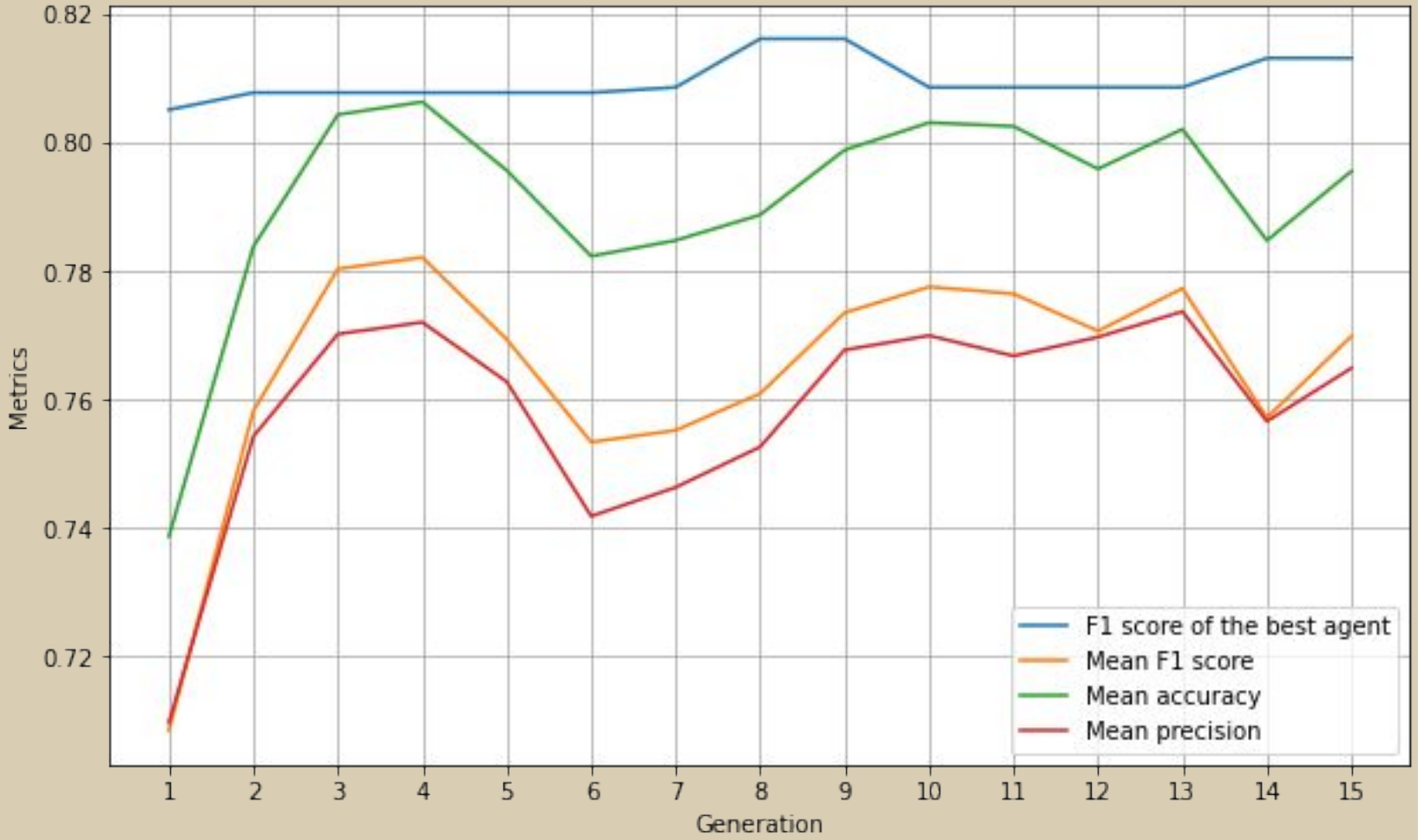
Rys.7. Proces wybierania genów w selekcji naturalnej w algorytmie genetycznym [4].

Wykorzystane technologie

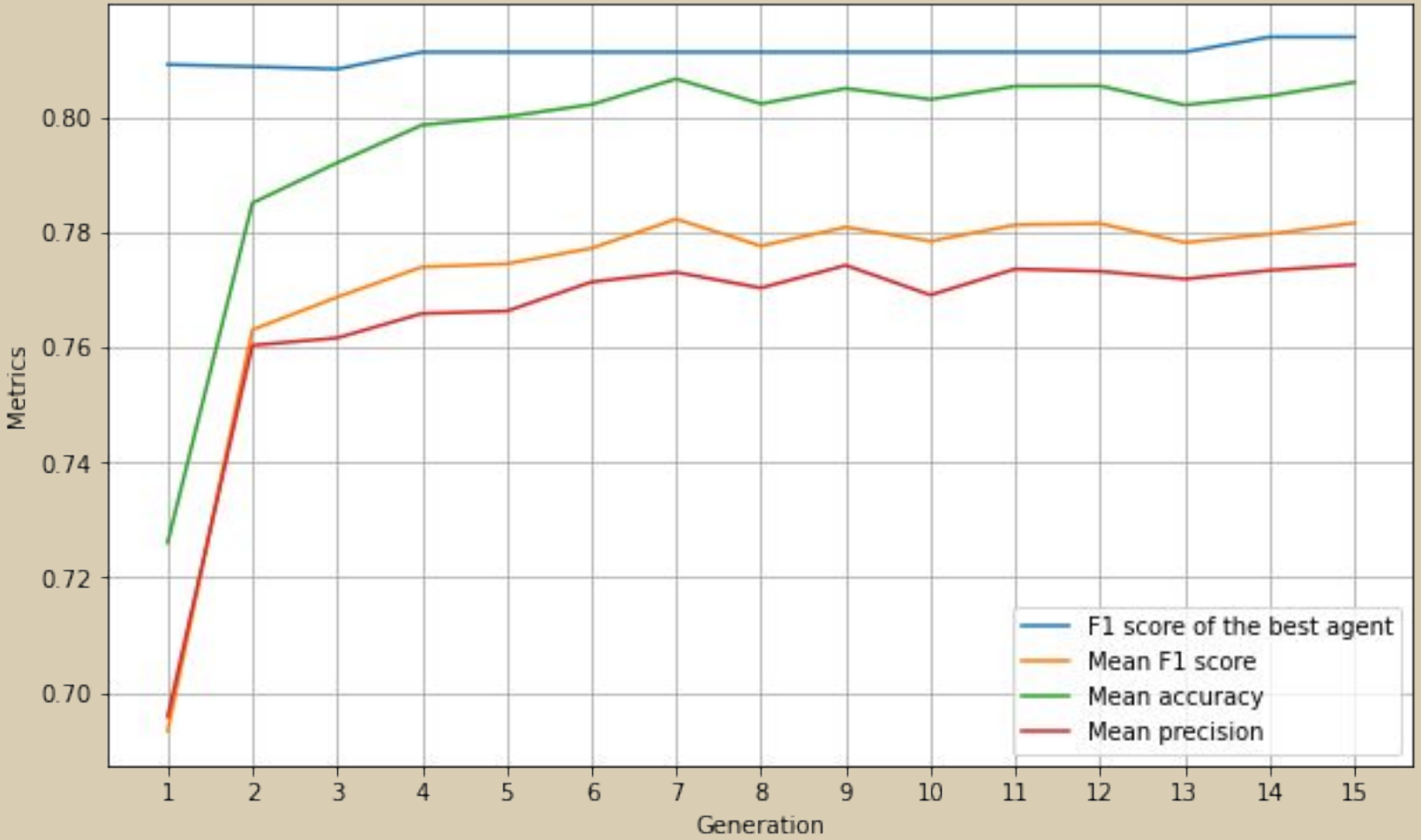
- Google Colaboratory
- Python
- PyTorch
- PyTorch Geometric
- NumPy
- Matplotlib



Rys.8. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 50.



Rys.9. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 200.



Rys.10. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 1000.

Wyniki

Tab.1. Wyniki dokładności oraz wskaźnika F1 dla najlepszego modelu wyselekcjonowanego w algorytmie genetycznym dla populacji o początkowych rozmiarach: 50, 200 oraz 1000. Porównanie z najlepszymi wynikami uzyskanymi również przez optymalizację algorytmem genetycznym z publikacji [4]

Najlepszy model \ Metryka	Dokładność	Wskaźnik F1
rozmiar populacji = 50	0.8306	0.8091
rozmiar populacji = 200	0.8243	0.8078
rozmiar populacji = 1000	0.8300	0.8136
Wyniki z publikacji [4]	0.8420	-

Perspektywy rozwoju

- Zrównoleglenie obliczeń funkcji dopasowania dla każdego agenta, aby przyspieszyć działanie algorytmu
- Stworzenie większej przestrzeni możliwych architektur
- Sprawdzenie działania algorytmu na innych zestawach danych
- Zastosowanie bardziej zaawansowanych algorytmów ewolucyjnych

Bibliografia

[1] http://tkipf.github.io/graph-convolutional-networks/
[2] https://www.topbots.com/graph-convolutional-networks/
[3] https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html
[4] “Evolutionary Architecture Search for Graph Neural Networks” Min Shi, David A. Wilson (2020)
[5] “An introduction to Neural Architecture Search for Convolutional Networks” George Kyriakides, Konstantinos G. Margaritis (2020)
[6] https://medium.com/@brainyloop/introduction-to-evolutionary-algorithms-genetic-algorithm-neuro-evolution-f872fc3eb573
[7] https://nptel.ac.in/courses/112103301
[8] https://homepages.ecs.vuw.ac.nz/~xuebing/slides/WCCI2022_PlenaryTalk_BingXUE_Final.pdf
[9] https://colab.research.google.com/drive/1xeQ0klRhVtTYGA-LwfTMujXblBiOLgqd?usp=sharing

Dziękuję za uwagę!

Prezenter: Anna Konieczny
Opiekun projektu: mgr inż. Maciej Krzywda

Kraków Applied Physics and Computer Science

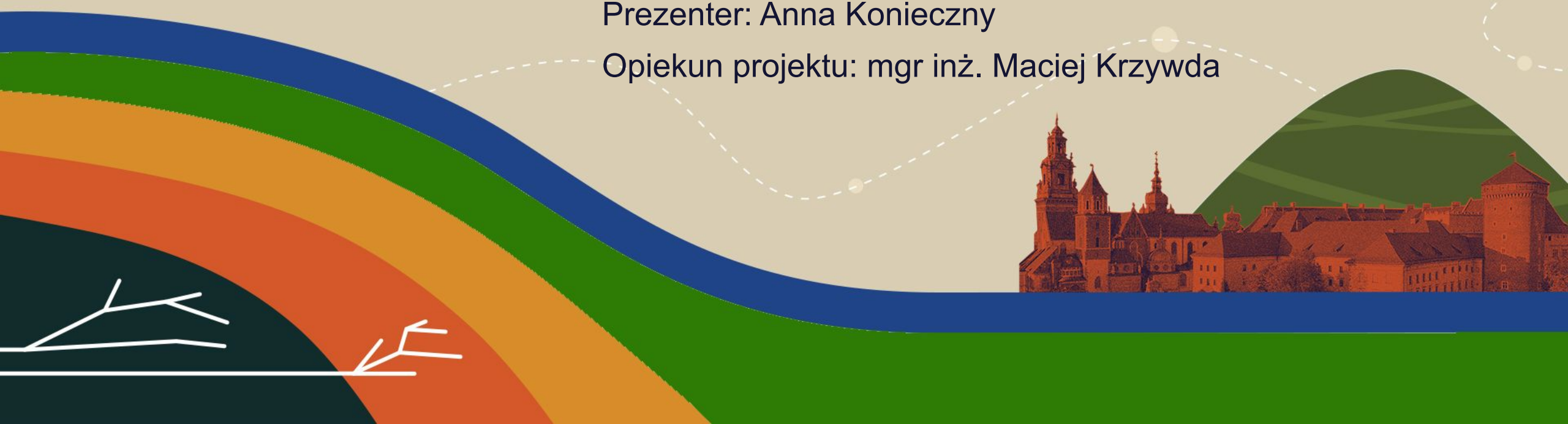
Summer School'22

29th of July 2022

Using Bio-Inspired Algorithms to Optimize Structures of Graph Convolutional Networks

Prezenter: Anna Konieczny

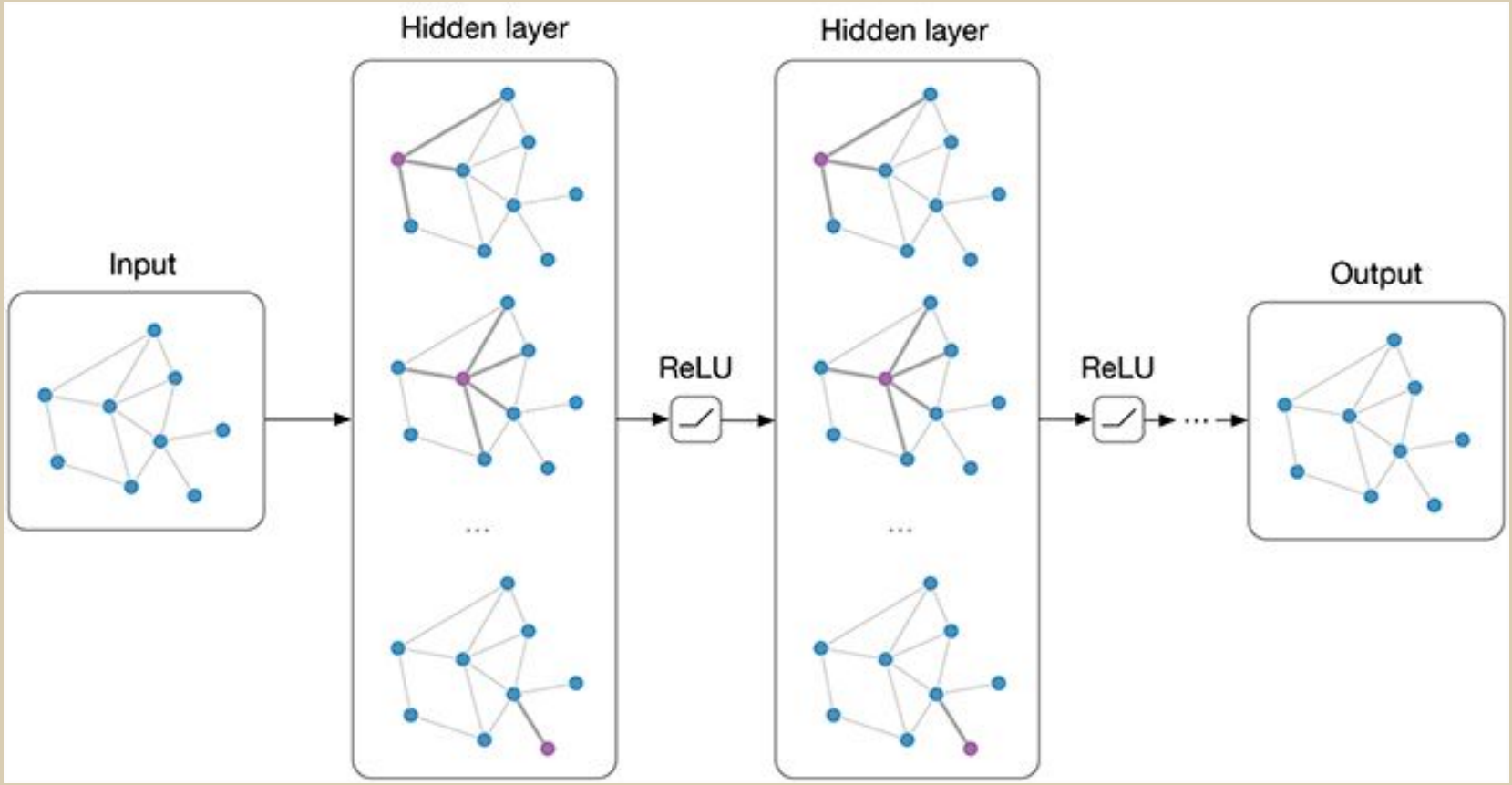
Opiekun projektu: mgr inż. Maciej Krzywda



Plan prezentacji

1. Grafowe konwolucyjne sieci neuronowe - GCN
2. Porównanie sieci GCN oraz CNN
3. Problem optymalizacji architektury
4. Optymalizacja architektury GCN
5. Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury
6. Wykorzystane technologie
7. Wyniki
8. Perspektywy rozwoju

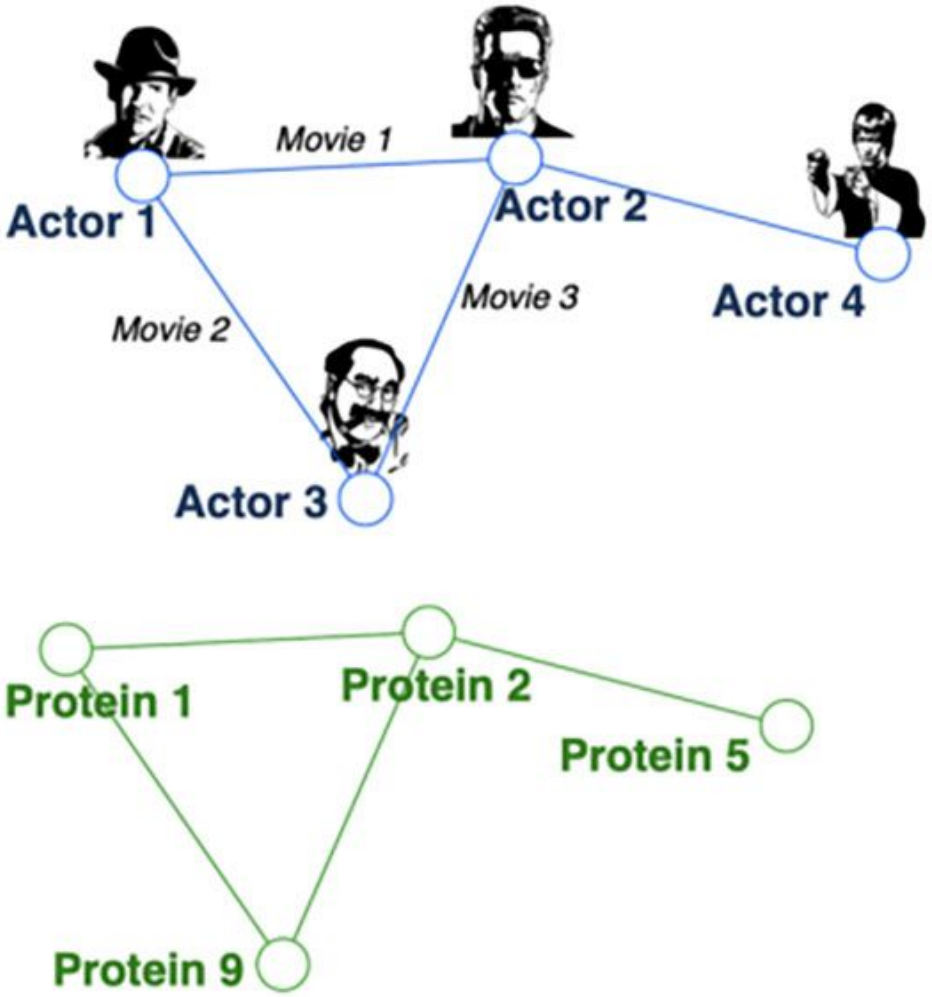
Grafowe sieci konwolucyjne - GCN



Rys.1. Uproszczony schemat grafowej sieci konwolucyjnej [1].

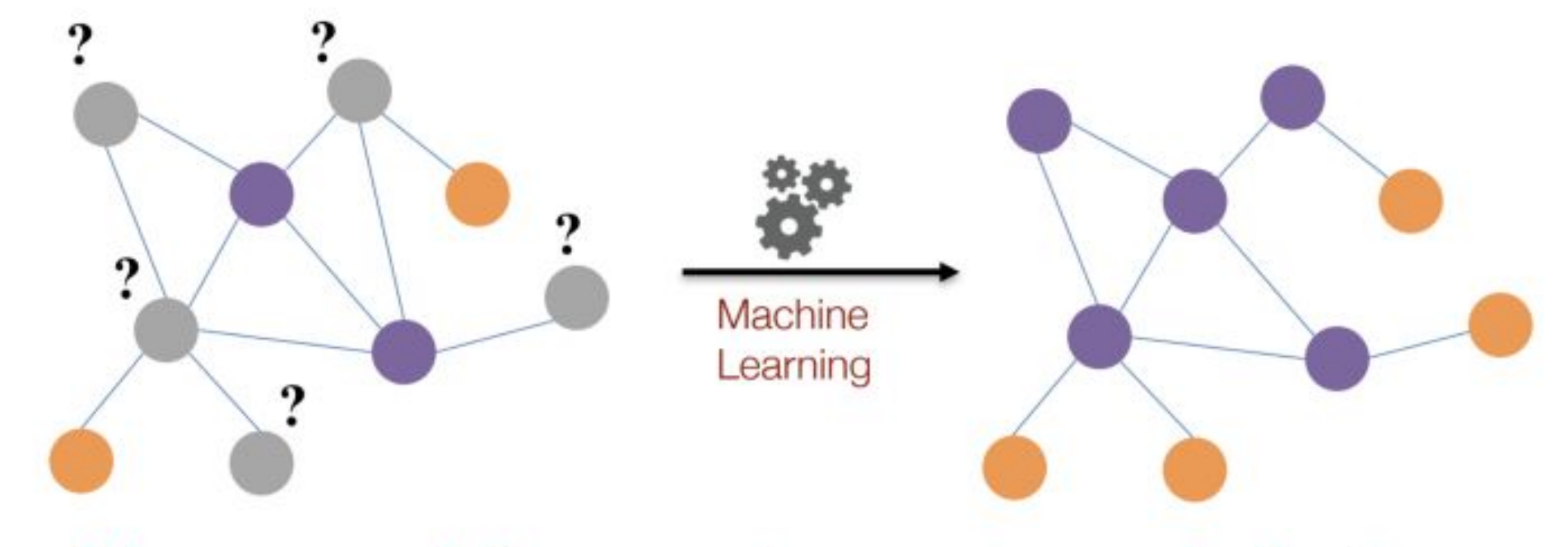
Niektóre z dostępnych zestawów danych z biblioteki pytorch-geometric [2]:

- dane zawierające informacje o interakcjach między proteinami **(PPI)**
- media społecznościowe, np. posty należące do różnych społeczności **(Reddit, Reddit2)**
- dane zawierające produkty ze sklepu Amazon i ich kategorie **(AmazonProducts)**
- dane zawierające 40 943 obiektów, 18 relacji i 151 442 faktów składających się z trzech słów, np. “łóżko jest meblem” **(WordNet18)**
- zestaw danych zawierający 2708 publikacji naukowych zaklasyfikowanych do 7 różnych klas, zawierający 5429 połączeń. Każda publikacja opisana jest wektorem z wartościami zero-jedynkowymi, które wskazują na obecność danego słowa ze słownika **(Cora)**



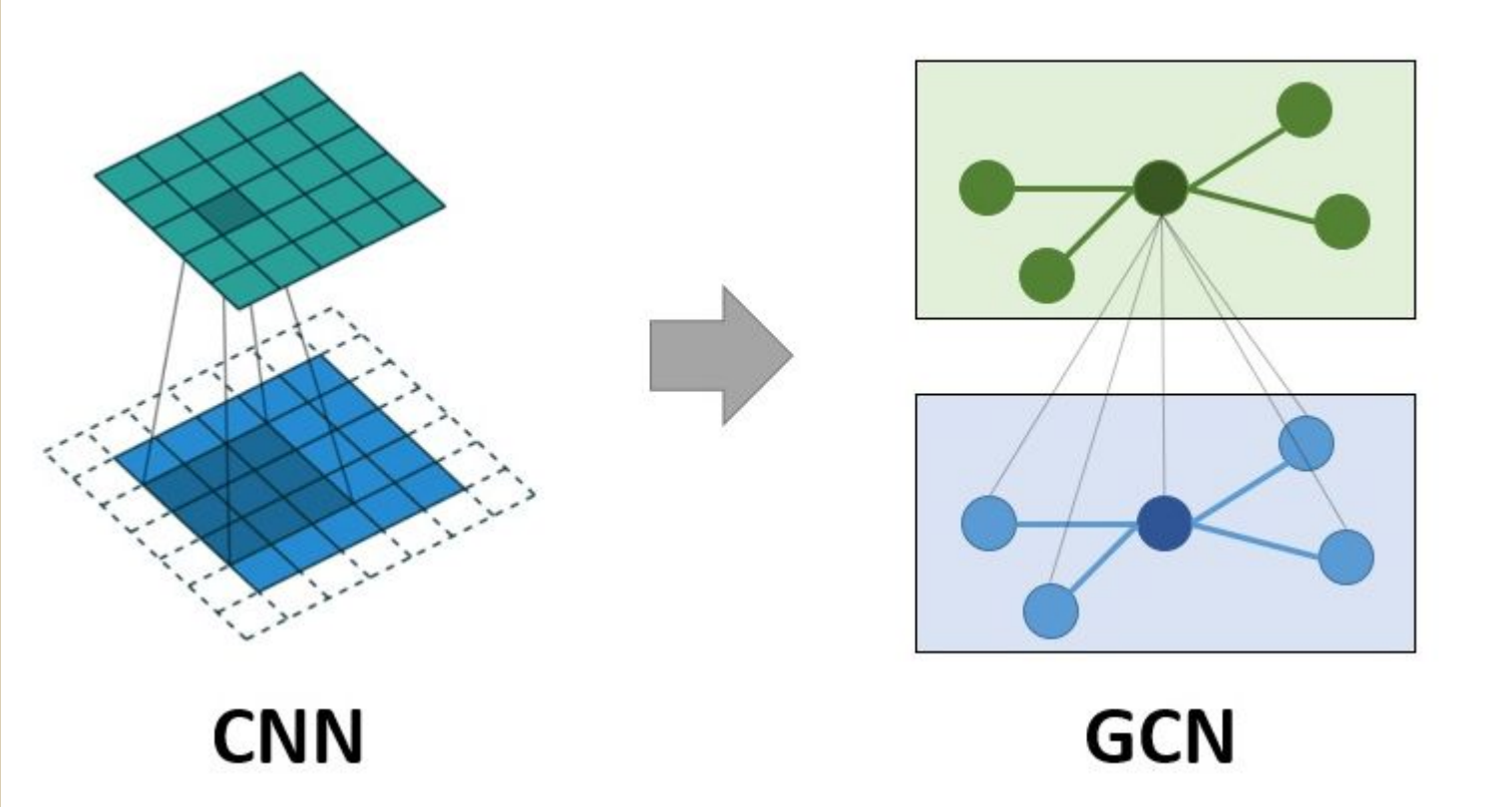
Rys.2. Wizualizacja przykładowych zestawów danych o postaci grafów [2].

Grafowe sieci konwolucyjne - GCN



Rys.3. Problem klasyfikacji węzłów danych o postaci grafów w uczeniu maszynowym [2].

Porównanie sieci GCN oraz CNN



Rys.4. Porównanie działania konwolucyjnej sieci neuronowej oraz grafowej sieci neuronowej [2].

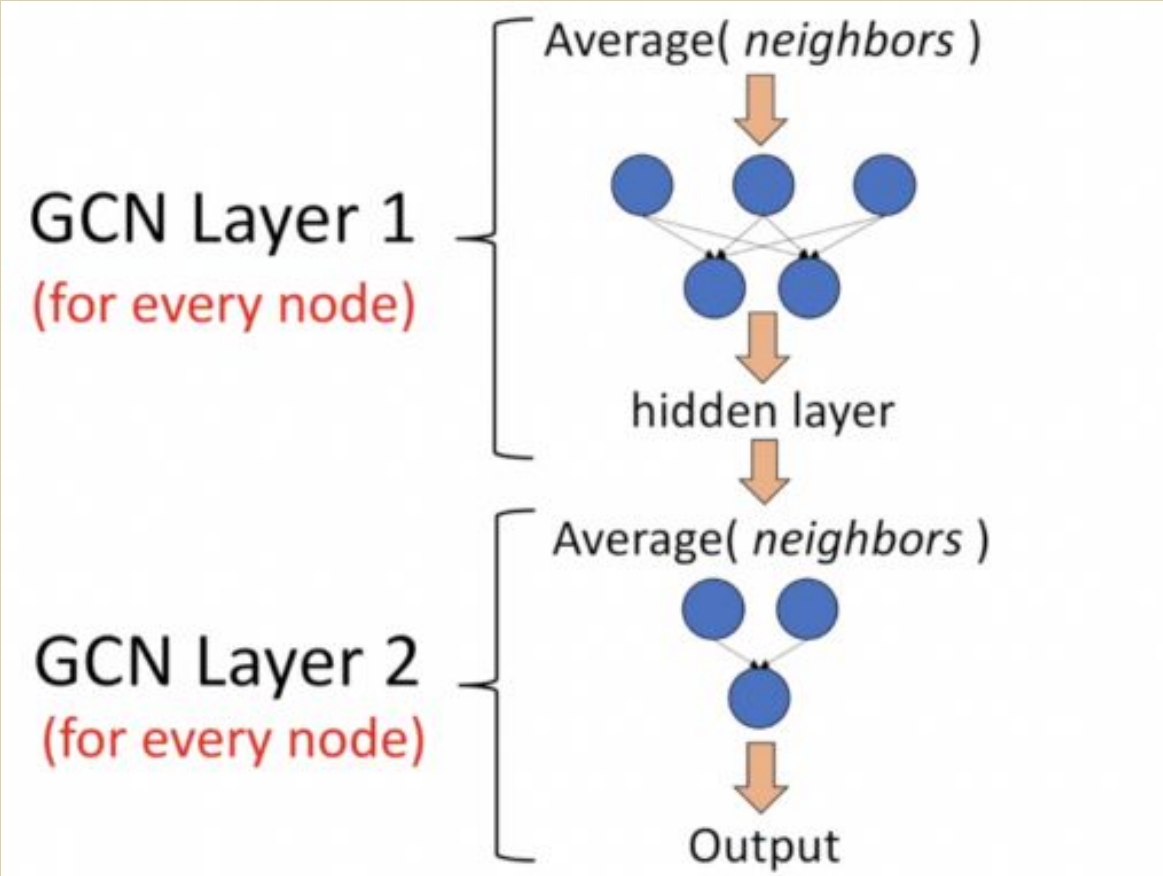
Problem optymalizacji architektury

Poszukiwanie optymalnej architektury sieci neuronowej (Neural Architecture Search - NAS) można podzielić m.in. na następujące etapy:

1. Przeszukiwanie przestrzeni (search space)
2. Metoda optymalizacji
   - algorytmy ewolucyjne
   - uczenie przez wzmacnianie
   - analiza bayesowska
3. Metoda ewaluacji kandydata
   - zredukowanie liczby epok
   - transfer learning

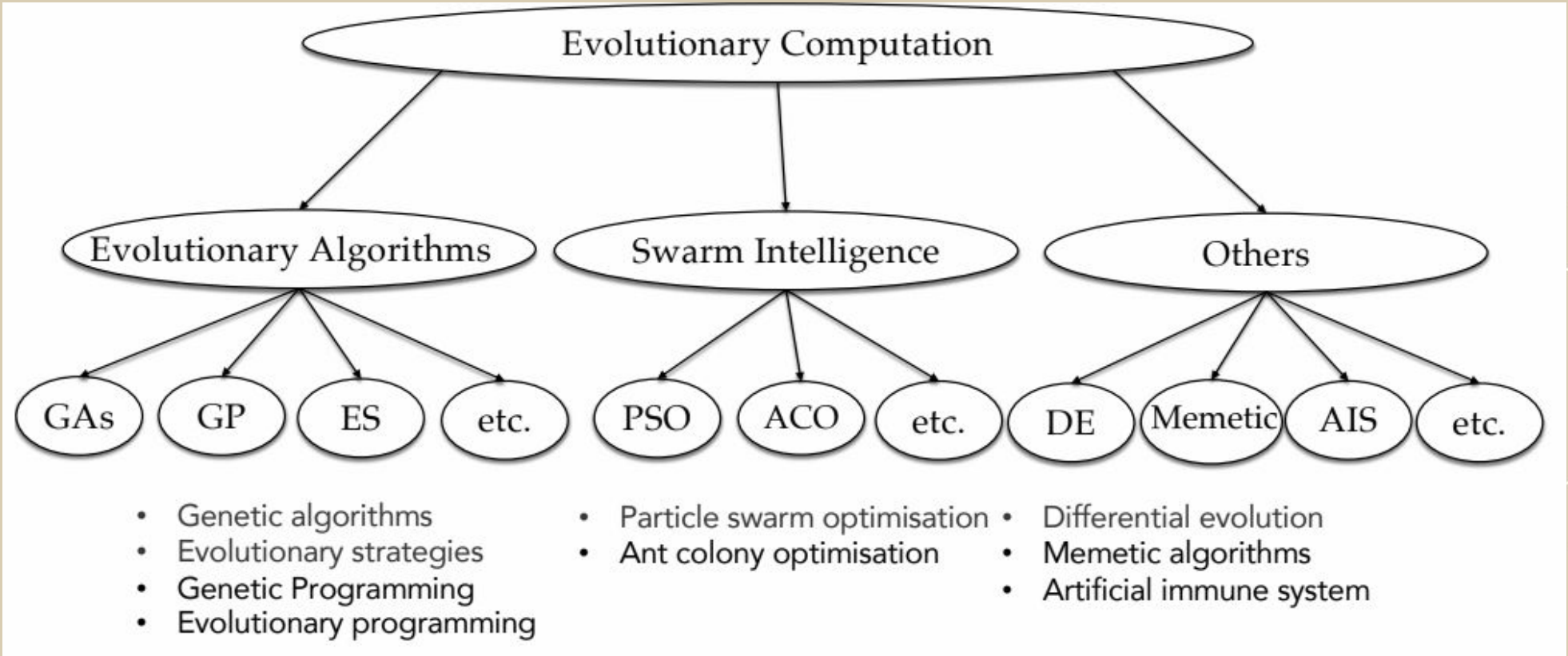
Optymalizacja architektury GCN

- liczba warstw konwolucyjnych: 2 lub 3
- liczba neuronów w warstwie ukrytej: od 20 do 100
- funkcja aktywacji w warstwie ukrytej: ReLU lub LeakyReLU
- obecność lub brak Dropout
- funkcja aktywacji: Linear lub LogSoftmax



Rys.5. Ilustracja obrazująca architekturę grafowej sieci neuronowej [2].

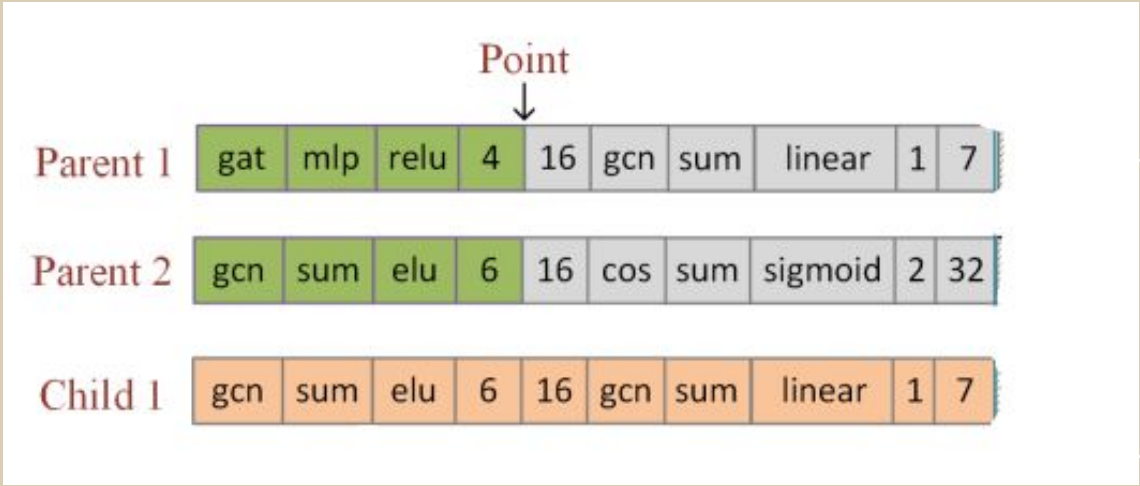
Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury



Rys.6. Ilustracja obrazująca podział dziedzin programowania ewolucyjnego[8].

Wykorzystanie algorytmów genetycznych w celach optymalizacji architektury

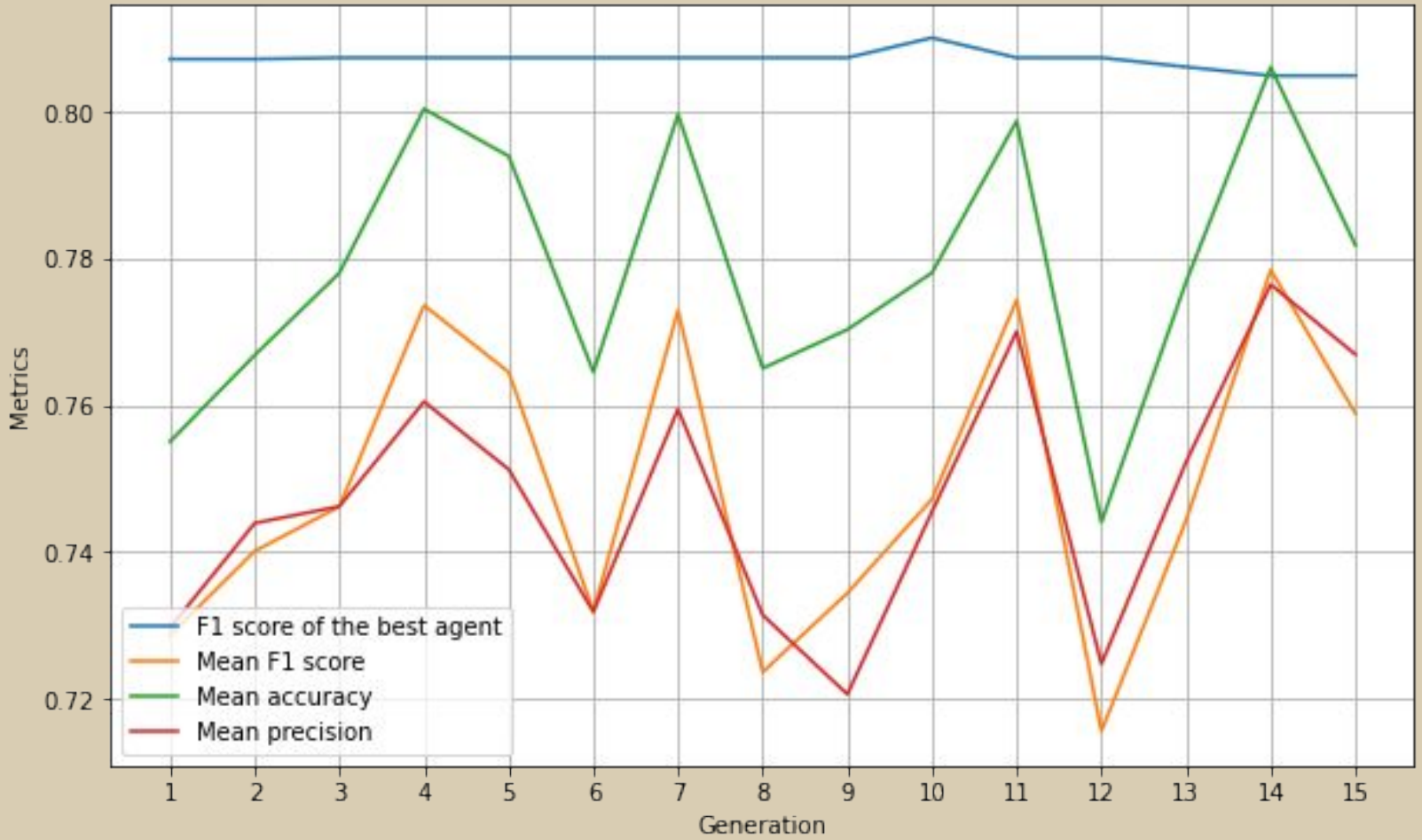
1. Inicjalizacja populacji.
2. Ewaluacja
   - obliczenie funkcji dopasowania
   - selekcja
   - krzyżowanie
   - mutacje
3. Otrzymanie najlepszego osobnika



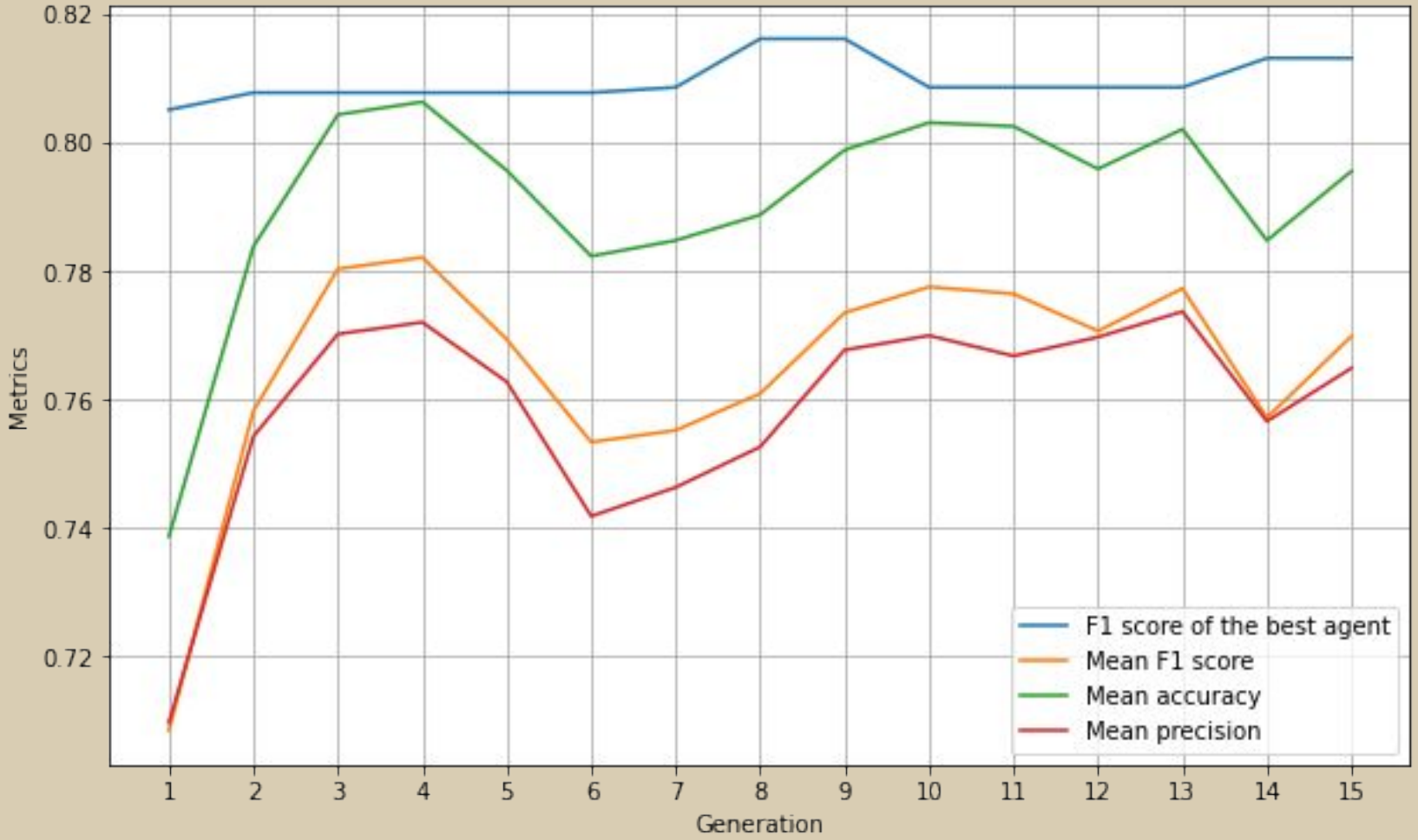
Rys.7. Proces wybierania genów w selekcji naturalnej w algorytmie genetycznym [4].

Wykorzystane technologie

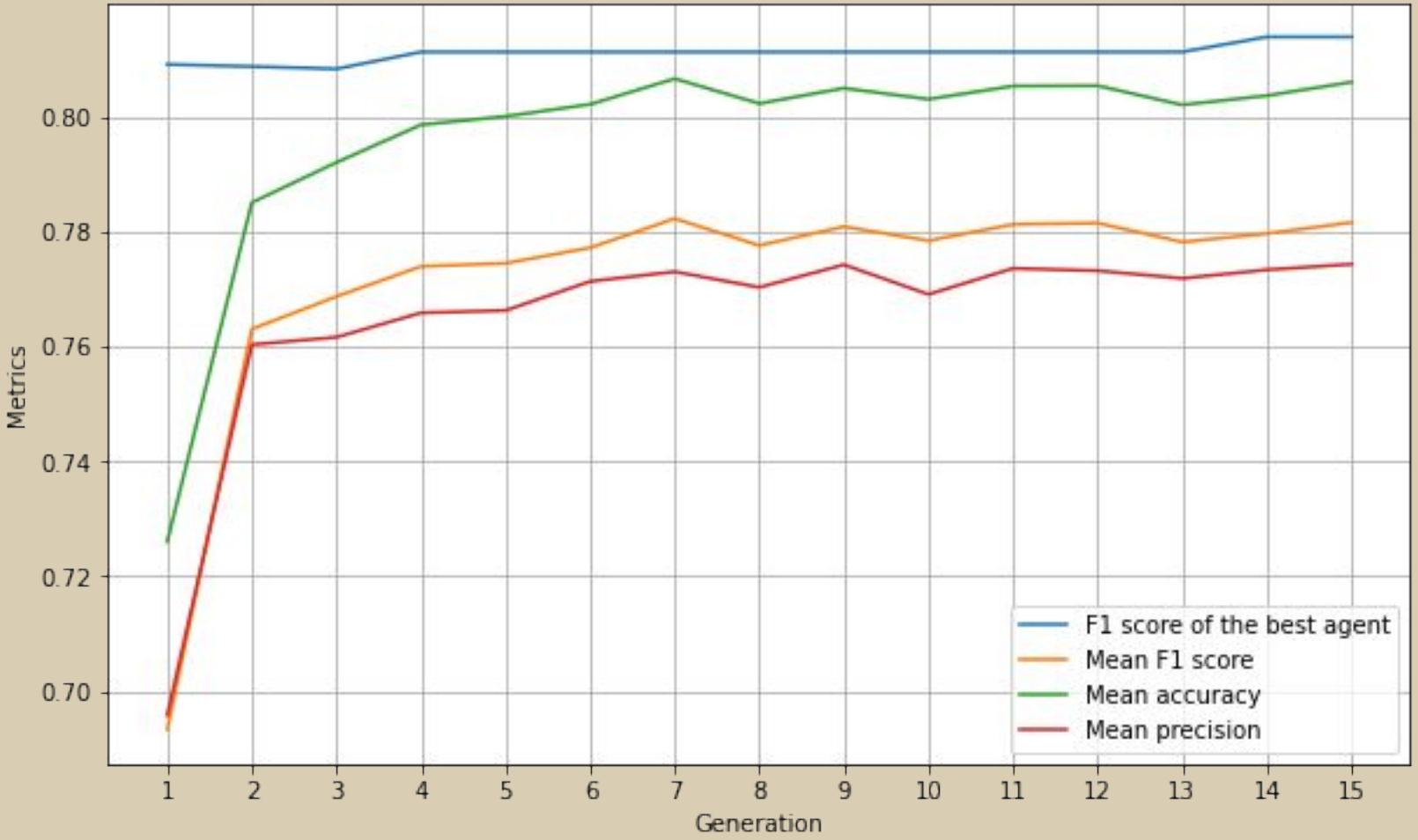
- Google Colaboratory
- Python
- PyTorch
- PyTorch Geometric
- NumPy
- Matplotlib



Rys.8. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 50.



Rys.9. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 200.



Rys.10. Wykres średnich wartości wskaźnika F1, dokładności, precyzji oraz wyniki wskaźnika F1 dla najlepszego osobnika w danej generacji dla początkowej wartości populacji równej 1000.

Wyniki

Tab.1. Wyniki dokładności oraz wskaźnika F1 dla najlepszego modelu wyselekcjonowanego w algorytmie genetycznym dla populacji o początkowych rozmiarach: 50, 200 oraz 1000. Porównanie z najlepszymi wynikami uzyskanymi również przez optymalizację algorytmem genetycznym z publikacji [4]

Najlepszy model \ Metryka	Dokładność	Wskaźnik F1
rozmiar populacji = 50	0.8306	0.8091
rozmiar populacji = 200	0.8243	0.8078
rozmiar populacji = 1000	0.8300	0.8136
Wyniki z publikacji [4]	0.8420	-

Perspektywy rozwoju

- Zrównoleglenie obliczeń funkcji dopasowania dla każdego agenta, aby przyspieszyć działanie algorytmu
- Stworzenie większej przestrzeni możliwych architektur
- Sprawdzenie działania algorytmu na innych zestawach danych
- Zastosowanie bardziej zaawansowanych algorytmów ewolucyjnych

Bibliografia

[1] http://tkipf.github.io/graph-convolutional-networks/
[2] https://www.topbots.com/graph-convolutional-networks/
[3] https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html
[4] “Evolutionary Architecture Search for Graph Neural Networks” Min Shi, David A. Wilson (2020)
[5] “An introduction to Neural Architecture Search for Convolutional Networks” George Kyriakides, Konstantinos G. Margaritis (2020)
[6] https://medium.com/@brainyloop/introduction-to-evolutionary-algorithms-genetic-algorithm-neuro-evolution-f872fc3eb573
[7] https://nptel.ac.in/courses/112103301
[8] https://homepages.ecs.vuw.ac.nz/~xuebing/slides/WCCI2022_PlenaryTalk_BingXUE_Final.pdf
[9] https://colab.research.google.com/drive/1xeQ0klRhVtTYGA-LwfTMujXblBiOLgqd?usp=sharing
[10] https://www.cs.vu.nl/~gusz/ecbook/ecbook-course.html

Dziękuję za uwagę!

Prezenter: Anna Konieczny
Opiekun projektu: mgr inż. Maciej Krzywda