

---

# Laboratorium

Logika rozmyta – kontroler rozmyty.

Tworzenie zdjęć generatywną siecią przeciwstawną (GAN).

Tworzenie tekstu LSTM.

---

## Zadanie 1

Użyjemy paczki `simpful`, by stworzyć prosty kontroler rozmyty do obliczania napiwków (przykład był na wykładzie).

a) Poczytaj o paczce `simpful` np. tutaj:

- <https://pypi.org/project/simpful/>
- [https://pythonhosted.org/scikit-fuzzy/auto\\_examples/plot\\_tipping\\_problem\\_newapi.html](https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html)
- [https://www.researchgate.net/publication/346395808\\_Simpful\\_A\\_User-Friendly\\_Python\\_Library\\_for\\_Fuzzy\\_Logic](https://www.researchgate.net/publication/346395808_Simpful_A_User-Friendly_Python_Library_for_Fuzzy_Logic)

Zwróć uwagę na wstawki kodu i sposób tworzenia kontrolerów (zmienne, reguły, wyostrzanie).

- b) Zainstaluj paczkę i skopiuj z wybranej strony kod tworzący system do dawania napiwków (3 zmienne lingwistyczne, 3 reguły).
- c) Wyświetl wykresy zmiennych lingwistycznych.
- d) Przetestuj działanie kontrolera. Daj kilka danych (liczby dla jedzenia i obsługi) i wyświetl jaki napiwek (0-30%) proponuje system dla tych inputów.

## Zadanie 2

Na wykładzie były prezentowane programy `gan01.py`, ..., `gan07.py`. Zostały stworzone na podstawie samouczka, do którego warto zajrzeć

- <https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/>

Wygenerujemy obrazki pisanych ręcznie cyfr (MNIST).

- a) Ściągnij pliki pythonowe z wykładu i przejrzyj je. Możesz je uruchomić.
- b) Szczególnie ważny jest plik `gan07.py`. Uruchom go ustawiając liczbę epok na 3 (trenowanie powinno zająć około 10 minut). W międzyczasie możesz pomyśleć nad zadaniem 2.
- c) Sprawdź jak wyglądają wygenerowane obrazki po każdej epoce (w `generated_plot_eXXX.png`).
- d) Czy model GAN da się zapisywać (wagi sieci w pliku)? Czy taki zapisany model da się potem załadować i dotrenować na kolejnych próbkach danych? Spróbuj znaleźć sposób na to.

- e) Jako praca domowa: wytrenuj model na większej liczbie epok (np. kilka godzin trenowania). Jak będą wyglądały wygenerowane obrazki? Pokaż je.

### Zadanie 3

Wykonaj następujące polecenia związane z sieciami rekurencyjnymi:

- a) Uruchom i przeanalizuj pliki z wykładu o sieciach rekurencyjnych:
- rnn01.py (prosta demonstracja)
  - rnn02.py (Badanie jak działa sieć rekurencyjna)
  - rnn03.py (Przewidywanie liczby plam na słońcu w danym miesiącu -RNN)
  - lstm01.py (Przewidywanie liczby plam na słońcu w danym miesiącu - LSTM)
  - lstm02.py (Uczenie generowania tekstu przez LSTM litera po literze)
  - lstm03.py (Uruchomienie generatora LSTM z poprzedniego zadania)
  - lstm04.py (Uczenie generowania tekstu przez LSTM słowo po słowie)
  - lstm05.py (Uruchomienie generatora LSTM z poprzedniego zadania)
- Uwaga: w zadaniach lstm02.py i lstm04.py uczenie może trwać wiele godzin. Znacznie zmniejsz liczbę epok, by trenowanie trwało parę minut.
- b) Zademonstruj jak działają generatory tekstu (lstm03 i lstm05) po Twoim krótkim treningu.
- c) Dotrenuj model z lstm02 i lstm04 o parę epok, wykorzystując jako bazę startową już wytrenowany przez Ciebie model zapisany w pliku hdf5. Następnie pokaż, czy generowany tekst jest trochę lepszy.

### Zadanie 4

To zadanie jest dość luźne, ale ciekaw jestem Państwa znalezisk i eksperymentów.

- Znajdź gotowy wytrenowany model do generacji zdjęć lub generacji tekstu w Internecie. Można skorzystać z modeli dostępnych na githubie, zamieszczanych przez wielu twórców, albo zajrzeć na stronę <https://huggingface.co/> która oferuje dostęp do wielu modeli.
- Zademonstruj jego działanie: wygeneruj kilka zdjęć lub fragmenty tekstu. Niektóre bardziej obciążające CPU/GPU można spróbować odpalić na google collab.
- Krótko opowiedz o tym modelu (struktura, pamięć).