

# LAB5: SSN - testowanie właściwości algorytmu wyprzedzającego i rozwiązań dla jedno- i dwu-wymiarowego, stacjonarnego przypadku transferu ciepła w materiałach jednorodnych i niejednorodnych

dr inż. Konrad M. Gruszka,\*

5 stycznia 2025

## Streszczenie

Bazując na niniejszym dokumencie należy rozszerzyć i przetestować pod różnymi kątami algorytmy wyprzedzające wykorzystujące SSN w oparciu o kryteria przedstawione w dalszej części tego dokumentu. Aby rozwiązać zadania z tego dokumentu, konieczne są skrypty utworzone na wcześniejszych zajęciach obejmujące przypadki stacjonarnego transferu ciepła dla jedno- i dwu-wymiarowych dziedzin obliczeniowych.

## Zadania projektowe

Zadania do samodzielnego wykonania

### Algorytm forward

1. Zbadaj wpływ różnych funkcji aktywacji: sigmoid, tanh, linear na dokładność i szybkość uczenia się modelu 1D.
2. Zmieniaj liczbę neuronów w każdej warstwie ukrytej i zbadaj, jak to wpływa na dokładność i zbieżność treningu sieci. Zastosuj 5 różnych konfiguracji poczynając od małej ilości neuronów aż do około 1000 na warstwę. Przetestuj zarówno model **1D** jak i **2D** Opisz je odpowiednio w raporcie.
3. Eksperymentuj z różną liczbą warstw ukrytych przy stałej ilości neuronów w warstwie, zmieniając ilość warstw w modelu (od 1 do 4) i oceniając wpływ na dokładność (**1D**).
4. Modyfikuj systematycznie szybkość uczenia się i analizuj, jak wpływa to na czas potrzebny do zbieżności oraz na dokładność końcową (**2D**).
5. Przeprowadź walidację krzyżową modelu, aby ocenić jego stabilność i zdolność generalizacji na różnych podziałach danych (1D lub 2D).
6. Zmodyfikuj wartość tolerancji zbieżności w wygenerowanych danych dla 0.01 0.05, 0.1, 0.5, 1.0, 5.0 i 10.0 stopni C i trenuj **model wyprzedzający** 1D SSN na tych danych (100 plików dla każdej tolerancji). Oceń jakość predykcji modelu w zależności od użytej tolerancji.
7. Użyj biblioteki matplotlib do wygenerowania wykresów przedstawiających jak zmienia się błąd SSN w trakcie treningu

---

\*Katedra Informatyki, Wydział Informatyki i Sztucznej Inteligencji (kgruszka@icis.pcz.pl)

## Algorytm wsteczny

1. **Dobór funkcji aktywacji.** Zbadaj wpływ różnych funkcji aktywacji w warstwach ukrytych na skuteczność SSN w zadaniu odwrotnym. Przetestuj przynajmniej następujące funkcje:

- ReLU
- tanh
- sigmoid
- LeakyReLU

Oceń modele pod kątem wartości błędu MAE, szybkości zbieżności oraz stabilności uczenia. Testy należy przeprowadzić zarówno dla modelu 1D, jak i jego rozszerzenia do 2D.

2. **Wpływ liczby i rozmiaru warstw ukrytych.** Sprawdź, jak liczba warstw ukrytych oraz liczba neuronów w tych warstwach wpływają na jakość wyznaczania warunków brzegowych. Zalecany zakres testów:

- Liczba warstw: 1–4
- Liczba neuronów: 64, 128, 256, 512, 1024

Analizuj wyniki dla wersji 1D i 2D, a także czas treningu, overfitting i dokładność odwzorowania. Można bazować na parametrach używanych w modelach wyprzedzających.

3. **Wpływ danych treningowych i ich objętości.** Przeanalizuj wpływ liczby przykładów treningowych na jakość działania SSN. W szczególności sprawdź:

- Jak zachowuje się model przy uczonym zbiorze danych: 50, 100, 200, 500 przypadków
- Czy istnieje punkt nasycenia (więcej danych nie poprawia wyników)?
- Jak zmienia się błąd MAE i czas uczenia?

Porównaj również wersję 1D i 2D – które wymagają większego wolumenu danych do podobnej skuteczności?

4. **Różna rozdzielczość siatki (liczba węzłów).** Sprawdź, jak zmienia się jakość odtwarzania warunków brzegowych w zależności od liczby węzłów:

- 1D:  $N = 20, 50, 100, 200$
- 2D: siatki  $10 \times 10, 20 \times 20, 30 \times 30$

Oceń, jak zmiana rozmiaru wejścia (i wyjścia) wpływa na strukturę sieci i efektywność. Czy zwiększanie liczby węzłów pogarsza, poprawia, a może nie wpływa znacząco na wyniki? Do określenia ilości węzłów siatki w zadaniu 2D podejdz elastycznie.

5. **Analiza wpływu wartości warunków brzegowych.** Zbadaj, czy SSN działa lepiej dla pewnych typów warunków brzegowych. Wygeneruj zestawy danych z różnymi przypadkami:

- Warunki równe:  $T_L = T_P$
- Duża różnica:  $T_L = 0^\circ C, T_P = 300^\circ C$
- Mała różnica:  $T_L = 100^\circ C, T_P = 105^\circ C$

Sprawdź, jak model radzi sobie z tymi przypadkami oraz czy wymaga innej liczby epok/trudniej się uczy.

6. **Porównanie jakości modeli 1D i 2D.** Rozszerz model do wersji 2D (siatka rozkładu temperatury) i porównaj jakość działania modelu z wersją 1D.

- Wersja 1D: wektor temperatury + model wyjściowy  $[T_L, T_P]$
- Wersja 2D: siatka  $N \times N$ , wyjście:  $T_{gra}, T_d, T_{lewo}, T_{prawo}$

Oceń różnice w złożoności sieci, czasie uczenia i skuteczności.

7. **Generalizacja do nieznanych warunków.** Sprawdź zdolność SSN do uogólnienia – trenuj model na danych z zakresu  $T_L, T_P \in [0, 200]$  i przetestuj go na danych z zakresu  $[200, 300]$ .

- Czy model poprawnie przewiduje warunki spoza zakresu treningowego?
- Czy SSN wykazuje tendencję do „zgadywania średniej”?

Przygotuj raport końcowy z projektu, który zawiera analizę wszystkich przeprowadzonych eksperymentów, w tym dane (tabele) i wykresy, które ilustrują główne wyniki (ipynb).