

LAB5-b: Opracowanie bazowej sieci neuronowej do modelowania problemu wyprzedzającego przepływu ciepła w materiale 2D

dr inż. Konrad M. Gruszka,*

5 września 2024

Streszczenie

Ten dokument stanowi instrukcję do wykonania ćwiczenia z zagadnienia transferu ciepła w dwuwymiarowym, stacjonarnym przypadku do przedmiotu Rozwiązywanie Zadań Odwrotnych. Bazując na niniejszym opisie należy zmodyfikować wcześniej napisany algorytm MRS a także zaprojektować i zaimplementować SSN do modelowania przepływu ciepła w materiale 2D (stacjonarnie).

1 Informacje wstępne

Wprowadzenie

Instrukcja bazuje na wcześniej napisanym oprogramowaniu implementującym metodę MRS do rozwiązywania dwuwymiarowego problemu transferu ciepła dla przypadku jednorodnego. Konieczna będzie modyfikacja kodu źródłowego dodająca nową funkcjonalność w postaci eksportowania wyników MRS do plików, które będzie można następnie wczytywać i traktować jako zbiory uczące i weryfikujące.

Cel ćwiczenia

Celem tego ćwiczenia jest zrozumienie podstawowych zasad modelowania procesów fizycznych oraz zastosowanie metod uczenia maszynowego do rozwiązania problemu przewidywania rozkładu temperatury w dwuwymiarowym obszarze. W ramach zadania należy stworzyć program w Pythonie, który będzie wczytywał dane generowane przez metodę różnic skończonych, a następnie na ich podstawie trenować sieć neuronową przewidującą rozkład temperatury w materiale na podstawie podanych warunków brzegowych i początkowych.

Środowisko pracy

- Python w wersji > 3
- Visual Studio Code z dodatkiem Jupyter Notebook

Biblioteki

- numpy
- matplotlib
- time
- keras
- sklern
- os - do wczytywania/zapisywania plików

*Katedra Informatyki, Wydział Inżynierii Mechanicznej i Informatyki (kgruszka@icis.pcz.pl)

2 Modyfikacja kodu MRS 2D w przypadku jednorodnym stacjonarnym

Zakres zadania:

1. Generowanie danych wejściowych i ich zapis do plików
 - (a) Użyj metody różnic skończonych (MRS) do symulacji przepływu ciepła w dwuwymiarowym materiale.
 - (b) Przygotuj skrypt generujący pliki z temperaturami w węzłach dla różnych warunków brzegowych i początkowych.
2. Przetwarzanie danych
 - (a) Opracuj funkcję wczytującą wygenerowane pliki i przekształcającą dane wejściowe na format odpowiedni do uczenia sieci neuronowej.

Wymagania funkcjonalne algorytmu

1. Program ma korzystać z wcześniej napisanego algorytmu MRS który należy skopiować do nowo tworzonego kodu
2. Podczas generowania danych należy posługiwać się wersją z ustawialną zbieżnością (parametr *tolerance*)
3. Warunki brzegowe mają być wygenerowane losowo, ale należy zadbać o to aby dane te się nie powtarzały
4. Pliki z danymi muszą zawierać informacje o temperaturze w każdym węźle a ponadto:
 - (a) Wygenerowane pliki mają trafiać do podkatalogu "data\"
 - (b) Nazwa pliku musi zawierać informacje o unikalnym numerze pliku, warunkach brzegowych z wszystkich czterech stron materiału oraz o iteracji po której osiągnięto zbieżność, wszystko oddzielone znakiem "-" przykładowa nazwa pliku: "70_28_106_300_400_1234.txt", gdzie:
 - i. 70 - to unikalny numer pliku
 - ii. 28 - temperatura z lewej strony materiału
 - iii. 106 - temperatura z prawej strony materiału
 - iv. 300- temperatura z górnej strony materiału
 - v. 406 - temperatura z dolnej strony materiału
 - vi. 1234 - numer iteracji po jakiej osiągnięto zbieżność
 - (c) Pierwsza linia w pliku ma zawierać następujące informacje oddzielone przecinkami (przykładowo: 28,106,0.01,3805):
 - i. 28 - T_L - warunek brzegowy (lewa krawędź)
 - ii. 106 - T_P - warunek brzegowy (prawa krawędź)
 - iii. 300 - T_L - warunek brzegowy (górna krawędź)
 - iv. 406 - T_P - warunek brzegowy (prawa krawędź)
 - v. 0.01 - tolerancja (użyty parametr progu zbieżności)
 - vi. 1234 - numer iteracji po którym osiągnięto zadaną zbieżność
 - (d) Dane zawierające temperatury w poszczególnych węzłach (od pierwszego do ostatniego) mają być oddzielone przecinkiem, z wyjątkiem ostatniej temperatury w ostatnim węźle.
5. Skrypt ma umożliwić wywołanie go z linii poleceń wraz z parametrami symulacji oraz bezpośrednio z VSCode z dodatkiem Jupyter.
6. Skrypt musi posiadać możliwość zdefiniowania liczby generowanych plików oraz przedziału temperatur w jakim ma generować warunki brzegowe.

Wydajność obliczeniowa

W celu ustalenia metryk wydajności obliczeniowej oraz ustalenia, w jaki sposób dobór parametrów empirycznych symulacji wpływa na całkowity czas wykonania należy zaimplementować:

- Pomiar czasu generowania jednego pliku
- Pomiar całkowitego czasu wygenerowania wszystkich tworzonych plików,

3 Opracowanie sieci neuronowej do modelowania problemu wyprzedzającego przepływu ciepła w 2D

W tym segmencie zadania należy zbudować model sieci neuronowej, który będzie w stanie nauczyć się i przewidywać rozkład temperatury w dwuwymiarowym materiale na podstawie jego warunków brzegowych i początkowych. W tym celu należy zaimplementować funkcję wczytującą pliki wygenerowane w pierwszej części zadania.

Wczytywanie danych:

Dane z wcześniej wyeksportowanych plików należy wczytać do macierzy *ndarray*, zapewniając odpowiedni wymiar tej macierzy. Przykładowo, dla 50 wyeksportowanych plików przy założeniu że siatka MRS miała 100x100 węzłów to doliczając dodatkowe cechy z pierwszej linii każdego z plików potrzebujemy rozmiaru macierzy 50×10404 elementy. Przy wczytywaniu elementów mogą być przydatne funkcje takie jak *file.readline()*, *os.listdir()*, *.split()* i *.strip()*.

Proces budowy SSN obejmuje kilka kluczowych etapów:

1. Ogólna budowa modelu sieci neuronowej:

- Zdefiniuj architekturę sieci neuronowej, która będzie uczyć się na podstawie wczytanych danych.
- Zastosuj bibliotekę Keras do definicji i trenowania modelu.
- Zaimplementuj podział danych na zestawy uczące i testowe.
- Oceń skuteczność modelu na danych testowych.

2. Wizualizacja wyników:

- Użyj biblioteki matplotlib do porównania przewidywanych rozkładów temperatury z rzeczywistymi danymi.

Poniżej przedstawiono szczegółowe wymagania do budowy modelu sieci neuronowej z wykorzystaniem Keras. Zadanie to podzielono na następujące etapy:

1. Inicjalizacja modelu: Sieć ma realizować model sekwencyjny

2. Dodawanie warstw:

- Warstwa wejściowa: Przyjmie parametry takie jak temperatury brzegowe, tolerancja, liczba iteracji zbieżności.
- Warstwy ukryte: Dodaj kilka warstw gęstych (Dense) z funkcjami aktywacji ReLU. Zalecane są co najmniej dwie warstwy gęste.
- Warstwa wyjściowa: Powinna zwracać rozkład temperatury wzdłuż całego obszaru, co oznacza wartości temperatur dla każdego węzła w siatce MRS.

3. Kompilacja modelu: Skonfiguruj proces uczenia poprzez wyb

Wydajność obliczeniowa

Jak wcześniej wspomniano, należy zbadać wydajność obliczeniową w kontekście generowania danych oraz uczenia sieci neuronowej, aby zrozumieć wpływ różnych parametrów symulacji na czas wykonania zadania.

- Pomiar czasu generowania jednego pliku danych, który obejmuje całą powierzchnię modelu 2D.
- Pomiar całkowitego czasu wygenerowania wszystkich tworzonych plików, zwracając uwagę na optymalizację procesu obliczeniowego.

Wizualizacja wyników

Po wytrenowaniu sieci neuronowej, ważne jest, aby zwizualizować wyniki w celu oceny skuteczności modelu i porównania z danymi rzeczywistymi.

- Wygeneruj mapy ciepła, które ilustrują rozkład temperatury przewidywany przez sieć oraz uzyskany z symulacji MRS.
- Stwórz wykresy różnic temperatur, które pomogą zidentyfikować obszary, gdzie model może wymagać dalszego dostrojenia.

4 Ocena i komentarze końcowe

Forma oddania zadania

Zadanie powinno być przedstawione w formie dokumentu pdf, zawierającego kod źródłowy, wyniki oraz wszelkie niezbędne wyjaśnienia i analizy.

- Dokumentacja powinna zawierać szczegółowe informacje o metodach generowania danych oraz architekturze sieci.
- Należy również dołączyć wykresy wyników, mapy ciepła oraz analizy porównawcze.
- W raporcie powinny znaleźć się wnioski dotyczące skuteczności modelu, potencjalne miejsca na usprawnienia oraz zastosowania praktyczne.

Kryteria oceny

1. Kompletność i jakość implementacji algorytmu MRS i SSN.
2. Dokładność i klarowność dokumentacji.
3. Skuteczność i dokładność modelu, oceniane na podstawie wyników testowych i wizualizacji.
4. Innowacyjność w rozwiązywaniu problemów oraz wprowadzane usprawnienia.

5 Podsumowanie

Opracowanie efektywnej sieci neuronowej do modelowania stacjonarnego transferu ciepła w 2D jest kluczowe dla zrozumienia i optymalizacji procesów termicznych w wielu dziedzinach inżynierii. Dzięki temu ćwiczeniu studenci mają możliwość zgłębienia wiedzy na temat zaawansowanych technik symulacji i uczenia maszynowego, co może przyczynić się do ich przyszłych sukcesów zawodowych w dziedzinie inżynierii i informatyki.