

Podstawy sieci neuronowych

Projekt

Spis treści

1	Projekt - wariant # 1	1
1.1	Opis	1
1.1.1	Wymagania projektu	1
1.1.2	Język programowania	2
1.2	Zadania	2
1.2.1	Zadanie 1 - Klasyfikacja obrazów	2
1.2.2	Zadanie 2 - Aproksymacja funkcji dwuwymiarowej	3
2	Projekt - wariant #2	4
2.1	Opis	4
2.1.1	Wymagania projektu	4
2.1.2	Język programowania	4
3	Wymagania implementacyjne	5
4	Sprawozdanie z projektu	6
4.1	Wymagania	6
4.2	Informacje dodatkowe:	6

1 Projekt - wariant # 1

Projekt obejmuje macierzową implementację wielowarstwowej sieci neuronowej wraz z algorytmem uczenia się i wytrenowanie jej do rozwiązania zadania (1) klasyfikacji i (2) aproksymacji.

1.1 Opis

1.1.1 Wymagania projektu

1. Zaprojektowanie i implementacja sztucznej sieci neuronowej używając macierzy i wektorów
2. Projekt nie korzysta z gotowych bibliotek do implementacji sieci neuronowych.
3. Wytrenowanie sieci do rozwiązania zadania klasyfikacji i aproksymacji.
4. Opis procesu doboru konfiguracji sieci (warstw, nastaw, parametrów, itp.).
5. Przeprowadzenie analizy działania sieci neuronowej.

1.1.2 Język programowania

MATLAB (sugerowany) lub Python; w obu przypadkach jest wymagana własna implementacja na bazie macierzy i wektorów.

1.2 Zadania

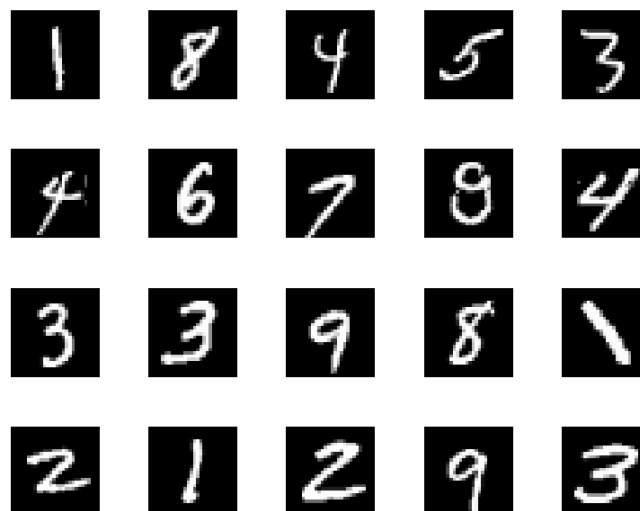
1.2.1 Zadanie 1 - Klasyfikacja obrazów

Zaprojektowanie i implementacja sztucznej sieci neuronowej typu MLP (Multi-Layered Perceptron, wielowarstwowa) do rozwiązania prostego problemu klasyfikacji obrazów dla dwóch zestawów danych składających się z obrazów binarnych (czarno-białych):

- **Cyfry (MINST)**: składa się z wielu obrazów o wymiarach 28×28 przedstawiających skany cyfr od 1 do 9 pisanych odręcznie. Obrazy pochodzą ze zbiorów MINST i są dostępne na stronach:
 - <http://yann.lecun.com/exdb/mnist/> i <https://www.kaggle.com/datasets/hojjatk/mnist-dataset> (+ jak odczytać dane w Pythonie)
 - <http://cis.jhu.edu/~sachin/digit/digit.html> (+ jak odczytać dane w Matlabie).

Uwagi końcowe:

- Sieć jest w stanie zaklasyfikować obraz do odpowiedniej kategorii.
- Wyniki końcowe oceniające efektywność sieci powinny być wyznaczone na podstawie zestawu testowego.



Rysunek 1: Cyfry (MINST)

1.2.2 Zadanie 2 - Aproksymacja funkcji dwuwymiarowej

Zadanie aproksymacji można przedstawić jako problem, w którym mając pewną dziedzinę i część wartości funkcji f należy odgadnąć wartość f w innych punktach. Ma to miejsce przy przewidywaniu wyników innych działań, zadaniach uogólnienia czy szukaniu ukrytych zależności w danych. Zakres zadań obejmuje wszelkiego rodzaju klasyfikacje nieznanymi obiektów na podstawie znanych przykładów jak rozpoznawanie mowy, OCR, sterowanie czy prognozowanie trendów.

Definicja. Niech X będzie dowolną przestrzenią wartości i niech $f : X \rightarrow Y$ to będzie pewna nieznana funkcja na X . Mając dany ciąg $(x_1, f(x_1)), \dots, (x_n, f(x_n))$ za pomocą analizy i uogólnień dla dowolnego punktu x_0 ze przestrzeni X znaleźć taką wartość y , która

- $f(x_0) = y$,
- $|f(x_0) - y| \rightarrow \min$,
- $P(f(x_0) = y) \rightarrow \max$.

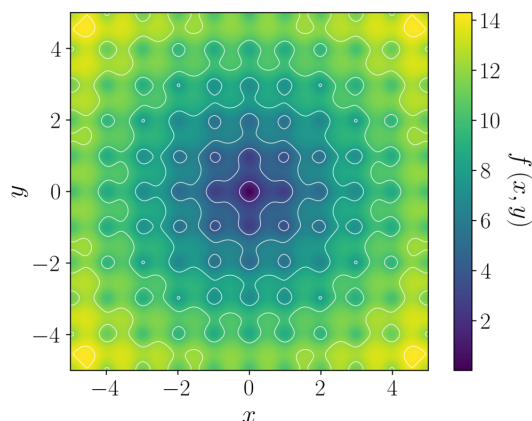
Zadanie:

Na podstawie zadanego zestawu losowych punktów i wartości funkcji w punkcie należy aproksymować dwuwymiarową funkcję Ackley'a $f_A(x_1, x_2)$ na przedziale $[-2, 2]$.

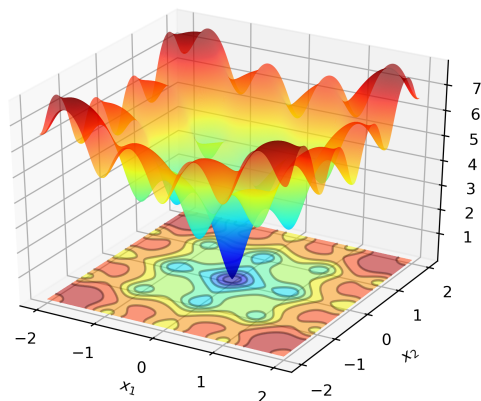
$$f_A(x_1, x_2) = -20 \exp \left[-0.2 \sqrt{0.5(x_1^2 + x_2^2)} \right] - \exp \left[0.5(\cos(2\pi x_1) + \cos(2\pi x_2)) \right] + \exp[1] + 20$$

Uwagi końcowe:

- Przetestowanie jakości aproksymacji używając próbek pobranych w regularnych odstępach, co 0.01.
- Przedstawienie na wspólnym wykresie węzłów aproksymacji oraz wyników z sieci neuronowej.
- Wyznaczenie średniokwadratowego błędu aproksymacji między wynikiem aproksymacji sieci, a rzeczywistą wartością funkcji w danym punkcie.



(a) Funkcja Ackley'a - plot 2D



(b) Funkcja Ackley'a - plot 3D

2 Projekt - wariant #2

Projekt obejmuje zaprojektowanie, implementację i wytrenowanie sztucznej sieci neuronowej do rozwiązania średniozaawansowanego problemu klasyfikacyjnego, aproksymacyjnego lub modelowania. Projekt zakłada wykorzystanie istniejących frameworków do sieci neuronowych i uczenia głębokiego.

2.1 Opis

2.1.1 Wymagania projektu

1. Zadanie obejmuje zaprojektowanie i implementację sieci neuronowej do rozwiązania wybranego problemu klasyfikacyjnego, aproksymacyjnego lub modelowania.
2. Implementacja sieci, jej uczenie i działanie powinny być zademonstrowane na wybranym, przykładowym zbiorze danych.
3. Dopuszczalne jest korzystanie z gotowych bibliotek programistycznych.
4. Zbiór danych sieci powinien być odpowiednio złożony.

2.1.2 Język programowania

Python. Sugerowane biblioteki do implementacji sieci neuronowych:

- TensorFlow,
- Keras,
- PyTorch.

Przykładowe zestawy danych można znaleźć na stronie

- <https://www.kaggle.com>.

3 Wymagania implementacyjne

1. Przetworzenie danych wejściowych do odpowiedniego formatu
 - odpowiedni format numeryczny
 - odpowiednia liczba wejść i wyjść
2. Analiza problemu i wybór odpowiedniej architektury sieci do rozwiązywanego problemu
3. Zaprojektowanie struktury sieci i dobór odpowiedniej konfiguracji:
 - współczynnik uczenia
 - liczba warstw ukrytych sieci
 - liczba neuronów w poszczególnych warstwach sieci
 - funkcje aktywacji
 - dodatkowe filtry
4. Analiza danych i podział na odpowiednie zestawy (testowy, treningowy, walidacyjny)
5. Implementacja algorytmu uczenia się i wytrenowanie sieci:
 - Ustalenie warunku zatrzymania procesu uczenia się (np. przekroczenie maksymalnego dopuszczalnego błęd lub określonej liczby iteracji)
 - Implementacja algorytmu uczenia się np. metoda największego spadku (metoda gradientowa)
 - Przeprowadzenie serii eksperymentów mających na celu ustalenie optymalnej struktury i konfiguracji sieci
6. Implementacja programu powinna pozwolić na uruchomienie eksperymentów na większą skalę.
7. Wyznaczyć adekwatne kryterium pomiaru błęd do rozwiązywanego problemu:
 - odpowiedni zestaw testowy
 - kryterium błęd, np. błąd średniokwadratowy

4 Sprawozdanie z projektu

Dla obu wariantów proszę przygotować **tylko jedno sprawozdanie**. W przypadku wariantu # 1 należy przeprowadzić testy dla obu zadań, a analizę i wnioski dla całości.

4.1 Wymagania

Należy udokumentować jak przebiegał proces projektowania, implementacji i trenowania sieci oraz przeprowadzić analizę otrzymanych wyników niezależnie od jakości uzyskanych wyników. W raporcie powinien znaleźć się opis:

- Analizy problemu i wyboru odpowiedniej architektury sieci do rozwiązywanego problemu
- Przygotowania architektury programu, w tym wybór odpowiednich bibliotek oraz pozostałych aspektów technicznych - jak format danych.
- Analizy danych i podział na odpowiednie zestawy (testowy, treningowy, walidacyjny).
- Przygotowania danych – wybór formatu danych wejściowych, stworzenie zestawów danych, sformatowanie plików danych, podział na zbiory: walidacyjny, testowy i uczący.
- Wyboru architektury sieci: rodzaju sieci, ilości warstw ukrytych i liczby neuronów w warstwach, dobór nastaw jak typu funkcji aktywacji czy wartość parametru uczenia.
- Trenowania sieci i opisu serii eksperymentów mających na celu ustalenie optymalnej struktury i konfiguracji sieci
- Analiza rozwiązania i wyników
 - Sukcesy i porażki.
 - Jakie problemy pojawiły się w trakcie i jak je rozwiązano?
 - Przeanalizowanie wpływów różnych konfiguracji sieci na wyniki.
 - Porównanie otrzymanych wyników i oczekiwań.

Ponadto:

- Dołączyć odpowiednie grafiki, grafy, tabelki prezentujące zestawy danych.
- Wizualizacje procesu uczenia się, np. wpływ liczby warstw na błąd na wyjściu.
- Wizualizacje wyników, np. przy pomocy macierzy trafień lub mapy błędów.

4.2 Informacje dodatkowe:

- Końcowa prezentacja projektu na zajęciach.
- Sprawozdanie w formacie A4 PDF.
- Streszczenie sprawozdania w postaci slajdów (PDF lub PowerPoint).
- Projekt (kod + sprawozdanie PDF + slajdy) udostępnione w postaci repozytorium.