SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

|  |  |
| --- | --- |
| Laboratorium Nr 6  Data 14.06.2025  Temat: „Liniowe RNN”  Wariant 10 | Anna Więzik  Informatyka  II stopień, niestacjonarne,  2 semestr, gr.1a TTO |

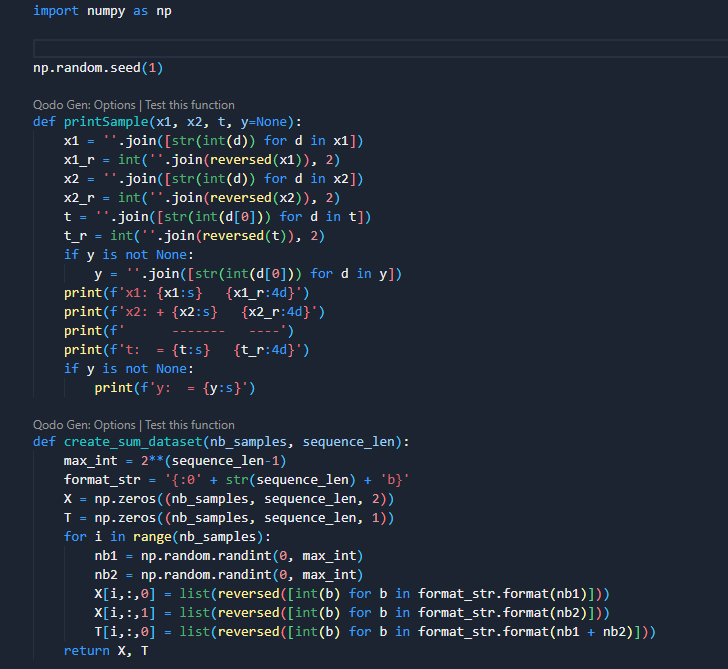
1. Polecenie:

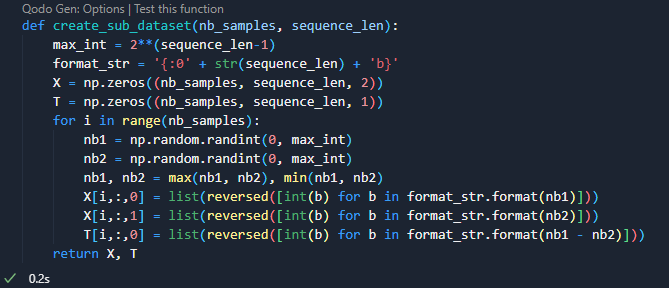
Link do repozytorium: <https://github.com/AnaShiro/MK_2025>

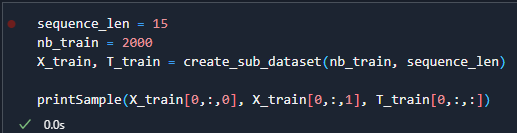
Rekurencyjne sieci neuronowe (RNN) to architektury neuronowe zaprojektowane do przetwarzania sekwencji danych. Charakteryzują się tym, że potrafią przechowywać informacje o wcześniejszych elementach sekwencji w postaci stanów ukrytych, co czyni je szczególnie przydatnymi w zadaniach, gdzie ważny jest kontekst czasowy. W omawianym zadaniu celem było opracowanie nieliniowej sieci RNN, która realizuje zadanie binarnego dodawania dwóch liczb 6-bitowych (rozszerzonych do 7 bitów), przy czym dane wejściowe, stany pośrednie i wyjścia są reprezentowane jako tensory trzeciego rzędu.

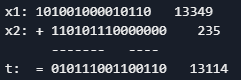
Architektura sieci składa się z kilku kluczowych komponentów: warstwy liniowej przetwarzającej dane wejściowe, warstwy rekurencyjnej aktualizującej stan sieci w kolejnych krokach czasowych, oraz warstwy klasyfikacyjnej z funkcją logistyczną obliczającej prawdopodobieństwo wygenerowania bitu wyjściowego. Przetwarzanie tensorowe umożliwia jednoczesne operacje na wielu próbkach i krokach czasowych, co znacznie zwiększa efektywność obliczeń. W celu zapewnienia poprawności implementacji przeprowadzono kontrolę gradientów, a do optymalizacji parametrów wykorzystano algorytm RMSProp z momentem Niestierowa, który pozwala skutecznie trenować sieci głębokie mimo złożoności powierzchni błędu.

1. Opis programu opracowanego









1. Wnioski

Zadanie pokazało, że rekurencyjne sieci neuronowe mogą skutecznie rozwiązywać zadania binarnego dodawania, o ile odpowiednio zaimplementowane są mechanizmy propagacji w czasie oraz funkcje aktywacji. Wykorzystanie tensorów trzeciego rzędu do reprezentacji wejść, stanów i wyjść umożliwiło jednoczesne i wydajne przetwarzanie wielu próbek. Kontrola gradientów potwierdziła poprawność zaimplementowanej wstecznej propagacji, a zastosowanie RMSProp z pędem Niestierowa przyczyniło się do stabilnej i skutecznej optymalizacji parametrów sieci.

Pomimo prostoty zadania (dodawanie binarne), jego realizacja wymagała złożonego przetwarzania sekwencyjnego oraz uwzględnienia wpływu wcześniejszych kroków czasowych na aktualny stan. Pokazuje to potencjał RNN do zastosowań w bardziej złożonych problemach szeregów czasowych czy przetwarzania języka naturalnego. Przeprowadzone eksperymenty potwierdziły, że odpowiednia konstrukcja architektury, inicjalizacja wag oraz dobór hiperparametrów mają kluczowy wpływ na sukces treningu i ogólną jakość działania modelu.