SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

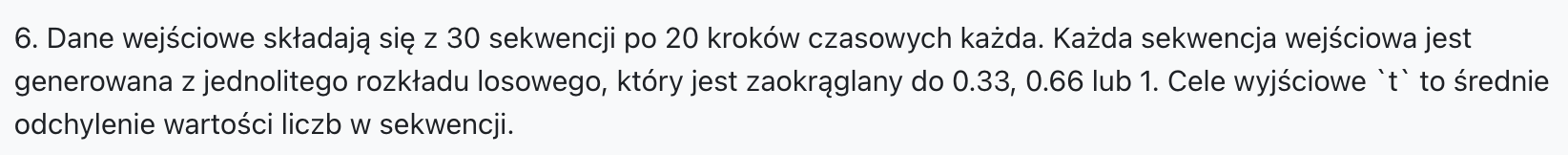
Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

|  |  |
| --- | --- |
| Laboratorium Nr 8  Data 28.11.2023  Temat: Liniowe RNN  Wariant 6 | Rafał Klinowski  Informatyka  II stopień, stacjonarne,  2 semestr, gr. a |

1. Polecenie:

Ćwiczenie polegało na stworzeniu notatnika Jupyter w języku Python do utworzenia rekurencyjnej sieci neuronowej (RNN) oraz przetestowania jej dla odgórnie narzuconych danych wejściowych oraz ustalonej cechy wyjściowej.

Wariant zadania: 6



2. Napisany program, uzyskane wyniki

Podczas realizacji laboratorium skorzystano z funkcji, obliczeń i parametrów zaproponowanych w instrukcji laboratoryjnej.

Pierwszym krokiem było utworzenie, zgodnie z poleceniem, 30 sekwencji po 20 kroków czasowych zaokrąglonych do wartości: 0.33, 0.66 i 1.0. Dla wygenerowanych w sposób losowy (z ustalonym ziarnem w celu zwiększenia powtarzalności eksperymentów) obliczono odchylenie standardowe odpowiednią funkcją.



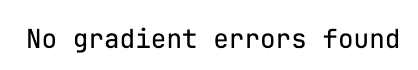
Rysunek 1. Utworzenie sekwencji wejściowych oraz uzyskanie oczekiwanej wartości cechy t jako odchylenie standardowe dla każdej z sekwencji.

Następnie na podstawie instrukcji laboratoryjnej zaimplementowano niezbędne funkcje, w szczególności „forward\_states” oraz „backward\_gradient”.



Rysunek 2. Zdefiniowane funkcje.

Przeprowadzono analizę gradientu i sprawdzono jego poprawność – nie znaleziono nieprawidłowych wartości.



Następnie utworzono wykresy prezentujące propagację wsteczną oraz gradient.



Rysunek 3. Wykresy prezentujące funkcje błędu dla danych wejściowych.



Rysunek 4. Wykres prezentujący gradient propagacji wstecznej w czasie (propagacja od lewej do prawej).

Przeprowadzono optymalizację RProp oraz utworzono wykresy tej optymalizacji.



Rysunek 5. Wykresy przedstawiające przebieg optymalizacji RProp.

Uzyskano następujące wagi modelu:

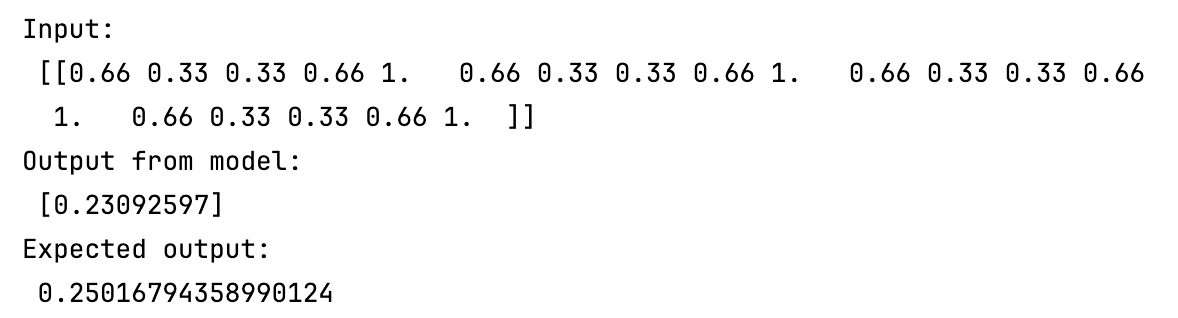


Rysunek 6. Uzyskane po optymalizacji wagi modelu wx oraz wRec.

Na koniec przetestowano model RNN dla przykładowych danych wejściowych, będących również sekwencją 20 danych przyjmujących wartość 0.33, 0.66 lub 1.0.



Rysunek 7. Utworzenie wartości testowych, uzyskanie wyniku z modelu oraz oczekiwanego wyniku – odchylenia standardowego dla tej sekwencji.



Rysunek 8. Uzyskane wyniki; dane testowe, wynik uzyskany z modelu oraz wynik oczekiwany jako odchylenie standardowe dla danych w sekwencji.

Wnioski:

* Uzyskany wynik jest zbliżony do wyniku oczekiwanego, czyli model po optymalizacji i z wykorzystaniem wstecznej propagacji błędu radzi sobie dość dobrze
* W przypadku podania innych danych wejściowych, na przykład mniejszej ilości elementów lub niepoprawnego zakresu, model niepoprawnie obliczy wartość celu
* Implementacja RNN w taki sposób jest dość długa i wymaga sporo pracy, jednak tym samym daje dobre wyniki, dobrze przystosowuje się do różnych wartości celu oraz jest łatwo rozszerzalna i można ją w prosty sposób modyfikować

Repozytorium zawierające uzyskane wyniki wraz z niezbędnymi plikami: <https://github.com/Stukeley/MatematykaKonkretna/tree/master/Lab8>