SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 6 Data 10.05.2025 Temat: „Liniowe RNN” Wariant 10	Szymon Nycz Informatyka II stopień, niestacjonarne, 2 semestr, gr.1a TTO

1. Polecenie:

Link do repozytorium: https://github.com/Maciek332/Semestr_3_Nycz/tree/main/MK

Rekurencyjne sieci neuronowe (RNN) to rodzaj modeli głębokiego uczenia, które świetnie radzą sobie z analizą danych sekwencyjnych, takich jak tekst czy sygnały czasowe. W przeciwieństwie do standardowych sieci neuronowych, RNN mają zdolność zapamiętywania wcześniejszych wartości, co pozwala na lepsze rozumienie kontekstu i struktur czasowych. W badanym przypadku skupiono się na uproszczonym wariancie RNN, zwanym liniową siecią rekurencyjną, gdzie pominięto funkcję aktywacji, ograniczając obliczenia do operacji liniowych.

Projekt miał na celu stworzenie minimalistycznej wersji RNN przy użyciu języka Python i biblioteki NumPy. Model trenowano za pomocą algorytmu propagacji wstecznej w czasie (BPTT), a następnie przeanalizowano zmiany gradientów, aby zilustrować efekty ich zanikania i eksplozji. Do treningu użyto losowych binarnych sekwencji o długości 10, a zadaniem sieci było nauczenie się sumowania wartości w każdej sekwencji. Dodatkowo przeprowadzono wizualizację powierzchni błędu oraz gradientów w zależności od wartości wag wejściowych i rekurencyjnych, co umożliwiło lepsze zrozumienie mechanizmów uczenia się.

2. Opis programu opracowanego

```python
import numpy as np

n_sequences = 30
sequence_length = 20
possible_values = np.array([0.0, 0.2, 0.4, 0.6, 0.8, 1.0])

np.random.seed(42)
X = np.random.choice(possible_values, size=(n_sequences, sequence_length))

t = X.sum(axis=1)

for i in range(5):
    print(f"X[{i}]: {X[i]}")
    print(f"t[{i}]: {t[i]}\n")
```

✓ 0.0s

```
X[0]: [0.6 0.8 0.4 0.8 0.8 0.2 0.4 0.4 0.4 0.8 0.6 0.4 1.  0.8 0.2 0.6 1.  1.
 0.2 0.6]
t[0]: 12.0

X[1]: [0.8 0.  0.6 0.2 1.  0.8 0.6 0.  0.  0.4 0.4 0.2 0.6 0.6 1.  1.  1.  0.4
 0.6 0.6]
t[1]: 10.799999999999999

X[2]: [0.  0.4 0.8 0.4 0.8 0.  0.2 0.6 0.  0.6 1.  0.2 0.2 0.  0.2 0.8 0.2 0.6
 0.6 0.6]
t[2]: 8.2

X[3]: [0.6 0.8 0.4 1.  0.  0.6 0.2 0.6 0.2 1.  1.  1.  0.2 0.6 1.  0.8 0.2 0.2
 0.6 0.2]
t[3]: 11.199999999999998

X[4]: [0.2 1.  0.6 1.  1.  0.6 0.  1.  0.8 0.8 0.2 0.8 0.2 0.  0.6 0.6 0.6 0.8
 0.  0.8]
t[4]: 11.600000000000001
```

3. Wnioski

Przeprowadzony eksperyment wykazał, że nawet najprostsza wersja sieci rekurencyjnej może być użyteczna do analizy kluczowych mechanizmów wpływających na proces uczenia. Obserwacja gradientów ujawniła ich znaczną niestabilność – w zależności od wartości wag mogły szybko zanikać lub gwałtownie rosnąć, co utrudniało skuteczną optymalizację modelu. Wizualizacja powierzchni błędu ukazała złożoną strukturę przestrzeni parametrów, w której lokalne minima są oddzielone stromymi zmianami, co dodatkowo komplikuje proces trenowania. Wykorzystanie algorytmu BPTT pozwoliło na śledzenie ewolucji gradientów w czasie i dostarczyło intuicyjnego wglądu w ich dynamikę. Wyniki eksperymentu podkreślają istotność odpowiedniego doboru hiperparametrów oraz możliwą konieczność zastosowania technik takich jak normalizacja wag, krótsze sekwencje treningowe czy bardziej zaawansowane architektury RNN (np. LSTM, GRU), które skuteczniej radzą sobie z problemami zanikania i eksplozji gradientów.