Państwa zadaniem jest prognozowanie wartości wybranej zmiennej przy wykorzystaniu sieci neuronowych. Prognoza może dotyczyć zarówno wartości danej zmiennej (np. wartość indeksu na zakończenie danego dnia; problem regresyjny) jak i klasy, do której zmienna będzie należeć (np. czy dana drużyna wygra, przegra czy zremisuje mecz; problem klasyfikacyjny).

Opis problemu: Optymalizacja logistyki dostaw (regresja lub klasyfikacja) Opis: Prognozowanie czasu dostawy przesyłek na podstawie historycznych danych logistycznych, takich jak odległość, pogoda, ruch drogowy, itp. Dlaczego interesujące?: Logistyka jest kluczowym elementem e-commerce i przemysłu. Literatura: Istnieją prace badające zastosowanie sztucznej inteligencji w logistyce, np. "Predicting Delivery Time with Machine Learning".

Zbiór danych

<https://www.kaggle.com/code/fareedalianwar/amazon-delivery>

**Kroki Budowy Algorytmu:**

1. **Przygotowanie danych**:
   * Analiza danych i usunięcie braków.
   * Standaryzacja/normalizacja cech, takich jak odległość czy czas.
   * Podział na zbiór uczący, walidacyjny i testowy.
2. **Wybór modelu**:
   * Określenie liczby warstw i neuronów.
   * Wybór funkcji aktywacji oraz optymalizatora.
3. **Trenowanie modelu**:
   * Uczenie modelu poprzez iteracyjne aktualizacje wag.
   * Monitorowanie skuteczności na danych walidacyjnych.
4. **Ewaluacja**:
   * Analiza wyników i dostrajanie hiperparametrów.
   * Porównanie różnych architektur sieci.

**Analiza Parametrów i Ich Wpływ:**

1. **Liczba warstw**:
   * Więcej warstw pozwala na naukę bardziej złożonych wzorców.
   * Zbyt wiele warstw może prowadzić do przeuczenia.
2. **Liczba neuronów**:
   * Więcej neuronów zwiększa zdolność modelu do reprezentacji danych, ale również ryzyko przeuczenia.
3. **Funkcja aktywacji**:
   * ReLU działa dobrze w większości przypadków regresji.
4. **Rozmiar próby uczącej**:
   * Im więcej danych w zbiorze uczącym, tym lepsze rezultaty (43739 próbek to wystarczająco dużo).
5. **Optymalizator i szybkość uczenia**:
   * Adam jest popularnym wyborem ze względu na szybką konwergencję.

pip install scikit-learn

pip install tensorflow

**pip install matplotlib**

**Parametry:**

|  |  |  |
| --- | --- | --- |
| **Argument funkcji** | **Wartość** | **Kto testuje** |
| Liczba neuronów | (Input + output)\*2/3 | Wasiczek |
| Liczba neuronów | Input\*2 > | Wasiczek |
| Liczba neuronów | Mean(Input + output) | Wasiczek |
| Liczba neuronów | 100 | Wasiczek |
| Liczba warstw | 1 | Zima |
| Liczba warstw | 2 | Zima |
| Liczba warstw | 3 | Zima |
| Liczba warstw | (0 lub 4) | Zima |
| Funkcja aktywacji | relu | Zima |
| Funkcja aktywacji | leaky\_relu | Zima |
| Funkcja aktywacji | swish | Zima |
| Funkcja aktywacji | tanh | Zima |
| Funkcja optymalizacji | Adam | Wasiczek |
| Funkcja optymalizacji | SGD | Wasiczek |
| Funkcja optymalizacji | RMSprop | Wasiczek |
| Funkcja optymalizacji | AdaDelta | Wasiczek |