Grygoruk Piotr 260299

Sztuczna Inteligencja i Inżynieria Wiedzy

Prowadzący: Mgr inż. Michał Karol

1. **Przygotuj zbiór uczący i walidacyjny, wykorzystując dołączony do listy kod procedury ekstrakcji cech. Jeśli zamierzasz korzystać z Weki, zalecane jest wykonanie jednorazowego przekształcenia danych i eksportu do jednego ze zgodnych formatów**

Przetworzenie danych za pomocą senetence transformera

In [3]:
```
ratings_data = pd.read_excel('jester-data-1.xls', header=None)
ratings_data = ratings_data.iloc[:, 1:].replace(99, float('nan'))
ratings = ratings_data.mean()
```

In [4]:
```
jokes_data = []

for i in range(1, 101):
    file_name = f'jokes/init{i}.html'
    with open(file_name, 'r') as file:
        joke_html = file.read()
        soup = BeautifulSoup(joke_html, 'html.parser')
        joke_text = soup.find('font', size='+1').text.strip()
        jokes_data.append(joke_text)
```

In [5]:
```
model = SentenceTransformer('bert-base-cased')
embeddings = model.encode(jokes_data)
```

Podział na zbiór treningowy i walidacyjny

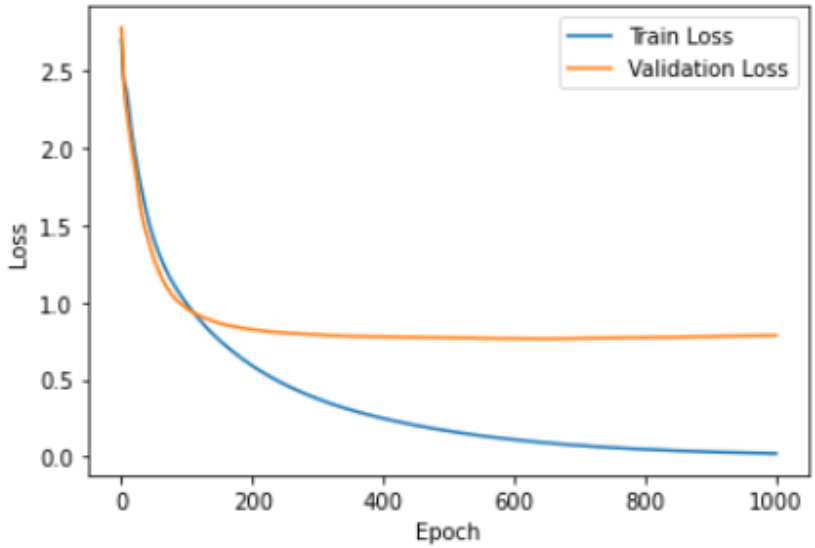
In [6]:
```
train_X, val_X, train_y, val_y = train_test_split(
    embeddings,
    ratings,
    test_size=0.2,
    random_state=3)
```

2. **Przetestuj działanie podstawowego modelu MLP o domyślnej konfiguracji hiperparametrów, ucząc go na danych ze zbioru Jester. Prześledź zachowanie modelu w czasie, wizualizując wartość funkcji kosztu w funkcji liczby epok, zwracając uwagę na wartości dla zbioru uczącego i zbioru walidacyjnego.**

Zdefiniowanie funkcji run z domyślnymi parametrami MLPRegressora – zaimplementowana w celu ułatwienia późniejszych testów.

In [7]:
```
def run(learning_rate_param=0.001, hidden_sizes=(100,)):
    mlp = MLPRegressor(solver='sgd',
                       alpha=0.0,
                       learning_rate='constant',
                       learning_rate_init=learning_rate_param,
                       hidden_layer_sizes=hidden_sizes,
                       random_state=0
                       )
    train_loss = []
    val_loss = []
    epochs = 1000

    for epoch in range(epochs):
        mlp.partial_fit(train_X, train_y)

        pred_train_y = mlp.predict(train_X)
        train_loss.append(mean_squared_error(train_y, pred_train_y))

        pred_val_y = mlp.predict(val_X)
        val_loss.append(mean_squared_error(val_y, pred_val_y))


    return (train_loss, val_loss)
```

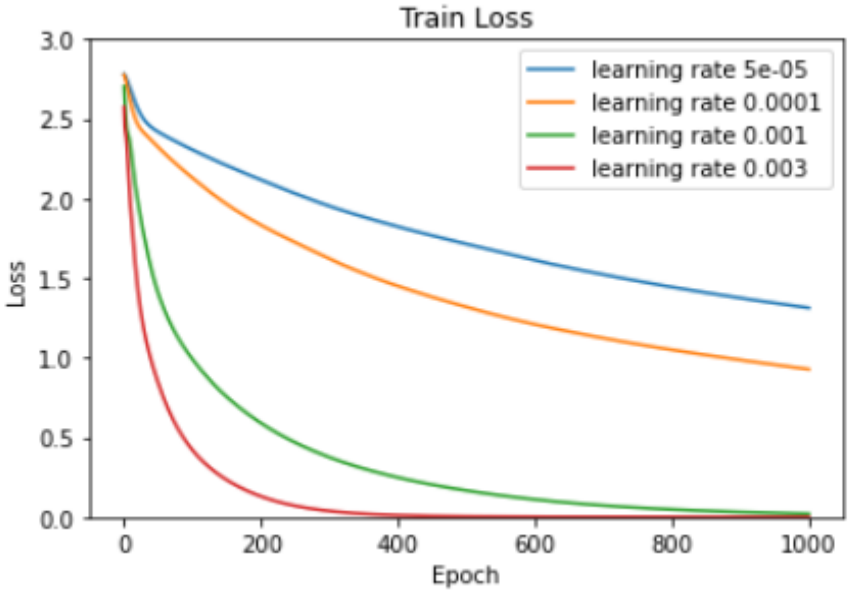
```
train_loss, val_loss = run()
plt.plot(range(len(train_loss)), train_loss, label=f'Train Loss')
plt.plot(range(len(val_loss)), val_loss, label=f'Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

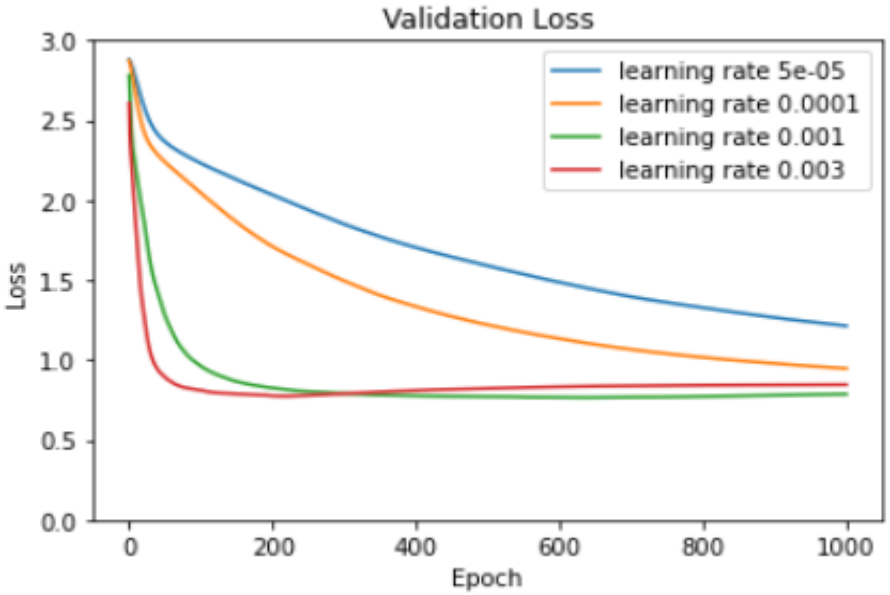


3. **Zbadaj wpływ tempa uczenia (learning rate) na osiągane wyniki: powtórz uczenie dla 3 różnych wartości parametru. Dobierz odpowiednią długość procesu uczenia (liczbę epok) jeśli to konieczne. Przedstaw wyniki na wykresach jak w zadaniu poprzednim. Co dzieje się, gdy tempo uczenia jest zbyt niskie? Co, gdy zbyt wysokie?**

Eksperyment został wykonany z parametrami:

Liczba epok = 1000
Learning rate = [0.00005, 0.0001, 0.001, 0.003]
hidden_layer_sizes = domyślny





W przypadku algorytmu uczenia maszynowego MLPRegressor, tempo uczenia odgrywa ważną rolę w procesie uczenia sieci neuronowej. Tempo uczenia odnosi się do szybkości, z jaką model dostosowuje swoje wagi na podstawie błędu predykcji. Zbyt niskie i zbyt wysokie tempo uczenia może mieć różne konsekwencje

Gdy tempo uczenia jest zbyt niskie:
• Proces uczenia może być powolny i wymagać większej liczby epok, aby model osiągnął zadowalające wyniki.
• Może istnieć ryzyko, że model utknie w lokalnym minimum, niezdolny do znalezienia optymalnego rozwiązania.
• Istnieje możliwość, że model będzie miał trudności z dopasowaniem się do złożonych wzorców w danych, co prowadzi do niższej wydajności predykcyjnej.

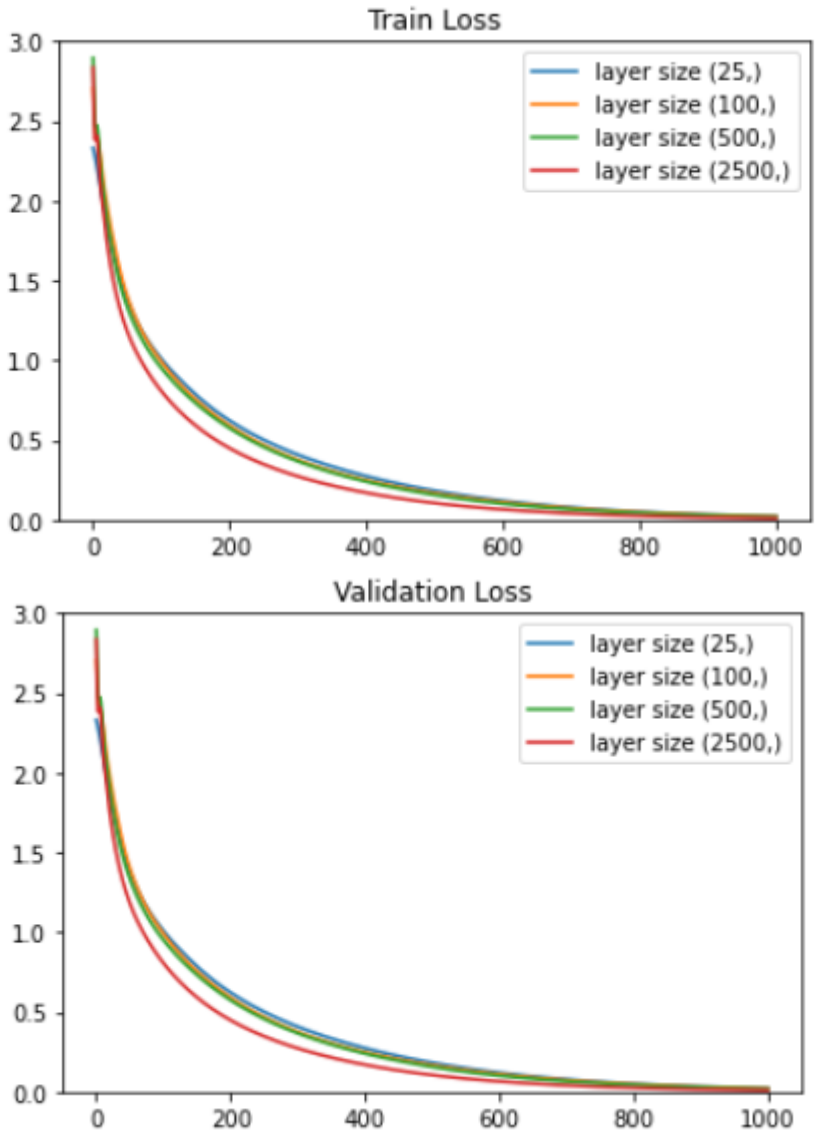
Gdy tempo uczenia jest zbyt wysokie:

• Proces uczenia może być bardzo szybki, co oznacza, że model może szybko dostosować się do danych treningowych.
• Może wystąpić ryzyko, że model będzie nadmiernie reagować na szum w danych treningowych, co prowadzi do przeuczenia.
• Model może mieć trudności z uogólnianiem wzorców na nowe dane, co prowadzi do niższej wydajności predykcyjnej na danych testowych.
• W skrajnych przypadkach, gdy tempo uczenia jest ekstremalnie wysokie, wagi mogą się "rozbiec" i model może nie zbiegać się do optymalnego rozwiązania.

4. Zbadaj wpływ rozmiaru modelu MLP na jakość działania: wykonaj co najmniej 3 eksperymenty dla modeli różniących się liczbą neuronów. Kiedy model przestaje dobrze dopasowywać się do danych? Kiedy zaczyna zanadto dopasowywać się do zbioru uczącego?

Eksperyment został wykonany z parametrami:

Liczba epok = 1000
Learning rate = 0.001
hidden_layer_sizes= [(25,), (100,), (500,), (2500,)]



Model przestaje dobrze dopasowywać się do danych:
Może to zdarzyć się, gdy model jest zbyt mały lub zbyt prosty, aby dobrze reprezentować zależności w danych. W takim przypadku, mimo uczenia się na danych treningowych, model będzie miał trudności z generalizacją na nowe przykłady. Możemy zauważyć, że dokładność modelu na zbiorze testowym będzie niższa niż na zbiorze treningowym. Jeśli ta różnica będzie znacząca, oznacza to, że model nie jest wystarczająco skomplikowany, aby dobrze dopasować się do danych.

Model zanadto dopasowuje się do zbioru uczącego:
To zdarza się, gdy model jest zbyt skomplikowany lub ma zbyt wiele neuronów, co pozwala mu na naukę zbyt szczegółowych cech zbioru uczącego, które mogą być przypadkowe lub nieistotne dla ogólnego wzorca. W takim przypadku model może idealnie dopasować się do danych treningowych, ale będzie słabo generalizować na nowe przykłady. Możemy zauważyć, że dokładność modelu na zbiorze testowym będzie niższa niż oczekiwana, a różnica między wynikami na zbiorze treningowym i testowym będzie znacząca.

5. Wybierz najlepszy uzyskany w drodze powyższych eksperymentów model i przetestuj go w praktyce: znajdź (lub napisz własny) tekst o charakterze dowcipu, przetwórz go na wektor za pomocą używanej w zadaniach metody ekstrakcji cech, a następnie odpytaj model neuronowy. Czy predykcja zgadza się z Twoim oczekiwaniem?

Najlepsze parametry:
Liczba epok = 1000
Learning rate = 0.001
hidden_layer_sizes=(100,)

```
def my_joke(joke):
    joke_embedding = model.encode([joke])
    joke_embedding = np.reshape(joke_embedding, (1, -1))
    rating_prediction = my_mlp.predict(joke_embedding)
    print("Predykcja oceny żartu:", rating_prediction, "\n")

jokes = [
    "Why did the scarecrow win an award? Because he was outstanding in his field!",
    "Why don't scientists trust atoms anymore? Because they make up everything!",
    "Why couldn't the bicycle stand up by itself? It was two tired",
    "Why don't skeletons fight each other? They don't have the guts!",
    "What do you call a bear with no teeth? A gummy bear!",
    "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dumm
]
```

```
In [14]: for joke in jokes:
             print(joke)
             my_joke(joke)

Why did the scarecrow win an award? Because he was outstanding in his field!
Predykcja oceny żartu: [-0.8665695]

Why don't scientists trust atoms anymore? Because they make up everything!
Predykcja oceny żartu: [-1.3128527]

Why couldn't the bicycle stand up by itself? It was two tired
Predykcja oceny żartu: [-1.7342541]

Why don't skeletons fight each other? They don't have the guts!
Predykcja oceny żartu: [-1.053306]

What do you call a bear with no teeth? A gummy bear!
Predykcja oceny żartu: [-1.41627]

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t
ext ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has s
urvived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popular
ised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishin
g software like Aldus PageMaker including versions of Lorem Ipsum.
Predykcja oceny żartu: [1.207004]
```

Podsumowanie

Widzimy tutaj następujący schemat – im dłuższy jest tekst, tym lepsza jest jego ocena jako żart. Prawdopodobnie ma to związek z zastosowaniem SentenceTransformera. Nie jest to zgodne z moją predykcją. Brak doświadczenia w stosowaniu sieci neuronowych uniemożliwia mi wyciągnięcie bardziej sensownych wniosków. Niemniej jednak lista zadań była interesująca.

6. Źródła

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

https://vitalflux.com/sklearn-neural-network-regression-example-mlpregressor/

Materiały z wykładu

https://chat.openai.com/