

Metody Inteligencji Obliczeniowej



Temat projektu: Kompresja danych z użyciem SSN

Szymon Pietraszek, Karol Śmiałek, Kacper Żółkiewski
14.06.2022

1. Opis problemu

Celem projektu było zaimplementowanie sieci neuronowej typu MLP (*Multilayer Perceptron*) do kompresji danych. Przy realizacji projektu wykorzystano przebieg EKG z portalu <https://www.kaggle.com/datasets/shayanfazeli/heartbeat>. Do implementacji autoenkodera użyliśmy biblioteki Keras. Jest to API do biblioteki TensorFlow. Przy pomocy tej biblioteki stworzony został model SSN, który umożliwia kompresję i dekompresję przebiegu EKG.

2. Konfiguracja sieci

Sieć składa się z 4 warstw odpowiednio o rozmiarach [64, 16, 64, 187], trzy pierwsze warstwy posiadają funkcję aktywacji typu 'relu' natomiast w czwartej warstwie zastosowaliśmy funkcję aktywacji typu 'sigmoid'. Stworzyliśmy własną klasę ECGAutoencoder, która daje nam dostęp do trzech MLP (encoder, decoder, autoencoder).

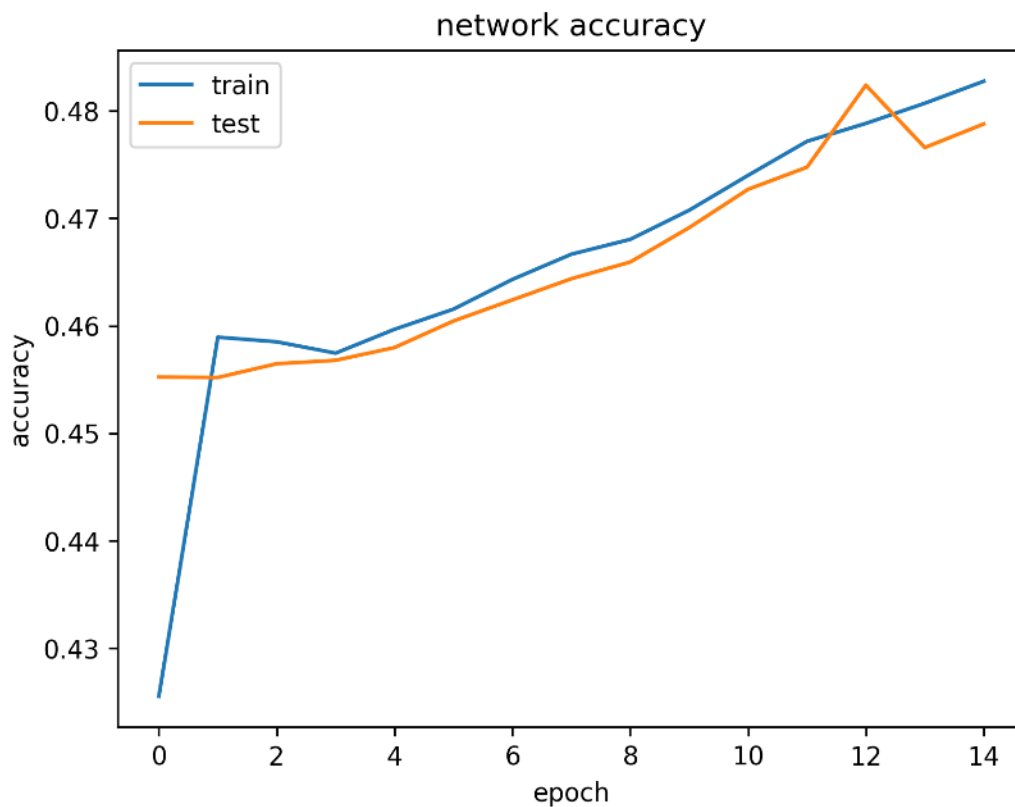
Klasa przyjmuje za argumenty:

- Rozmiar danych wejściowych: 187 próbek na wejście
- Lista rozmiarów warstw,
- Lista typów aktywacji warstw,
- Indeks warstwy, która ma być wyjściem dekodera

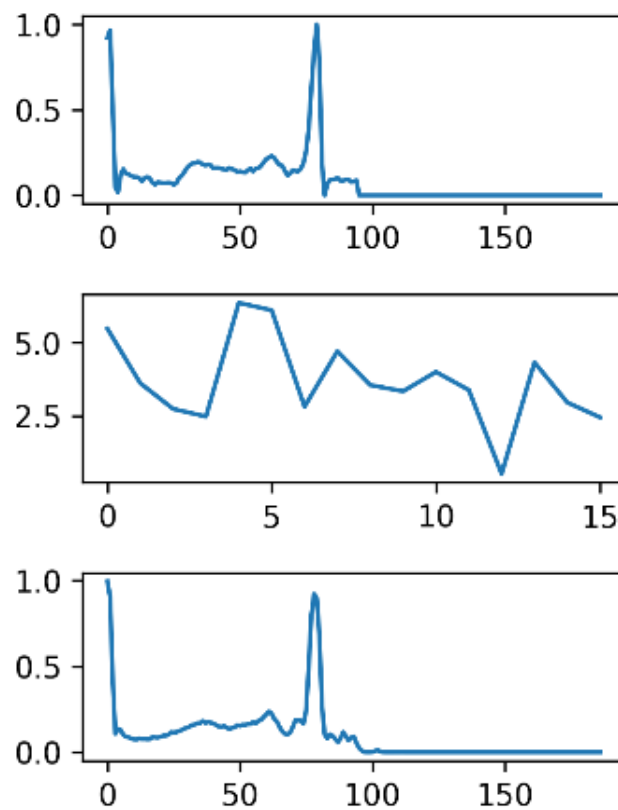
Liczbę epok ustawiono na 15, natomiast wartość batch_size na 256. Jako funkcji straty użyto binary_crossentropy.

3. Wyniki

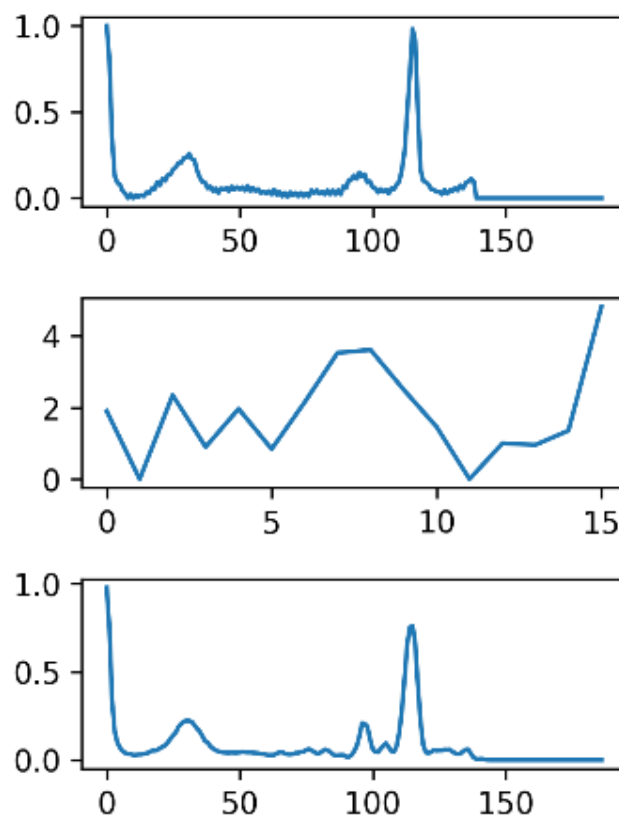
Na poniższym rysunku przedstawiony został wykres opisujący dokładność działania sztucznej sieci neuronowej. Wszystkie wykresy narysowano z wykorzystaniem biblioteki matplotlib.pyplot



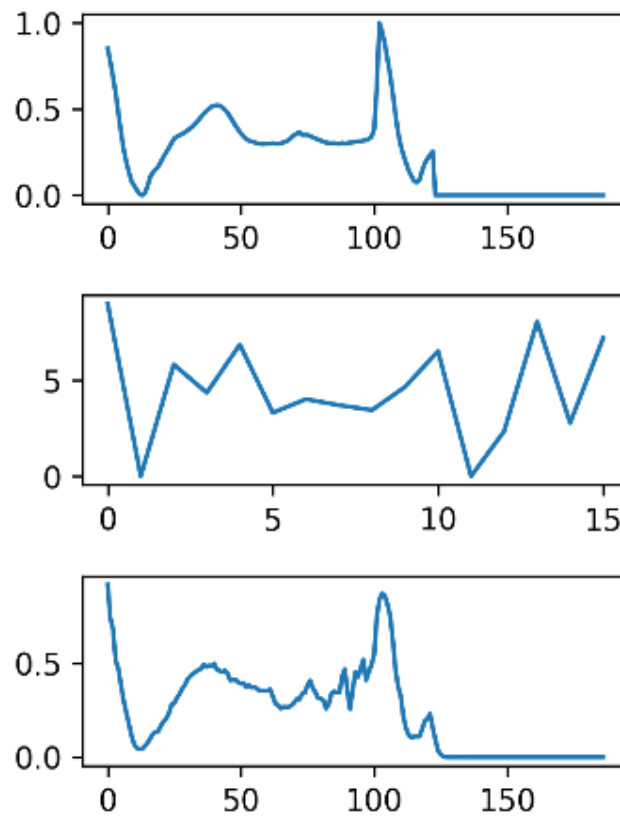
Rysunek 1. Dopasowanie sieci neuronowej



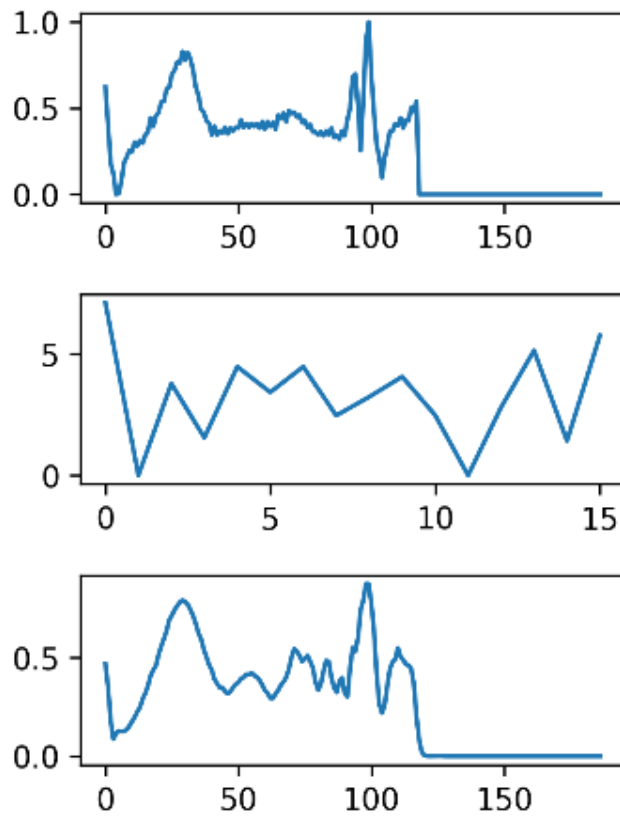
Rysunek 2. Wejściowy sygnał EKG, Skompresowany sygnał EKG, Zdekompresowany sygnał EKG.



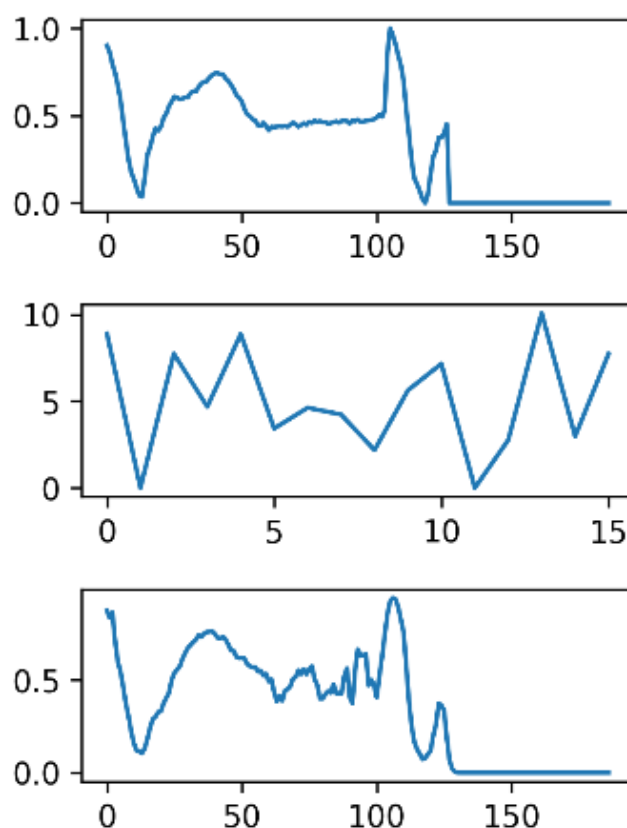
Rysunek 3. Wejściowy sygnał EKG, Skompresowany sygnał EKG, Zdekompresowany sygnał EKG.



Rysunek 4. Wejściowy sygnał EKG, Skompresowany sygnał EKG, Zdekompresowany sygnał EKG.



Rysunek 5. Wejściowy sygnał EKG, Skompresowany sygnał EKG, Zdekompresowany sygnał EKG.



Rysunek 6. Wejściowy sygnał EKG, Skompresowany sygnał EKG, Zdekompresowany sygnał EKG.

Analizując powyższe wykresy możemy stwierdzić że sieć działa prawidłowo, a rezultaty są zadowalające. Ponadto sygnał zdekompresowany przypomina, sygnał wejściowy.

4. Kod Źródłowy

Kod źródłowy dostępny jest w repozytorium <https://github.com/Smialekkarol/MIO>

5. Bibliografia

<https://blog.keras.io/building-autoencoders-in-keras.html>

<https://www.kaggle.com/datasets/shayanfazeli/heartbeat?resource=download>

https://en.wikipedia.org/wiki/Multilayer_perceptron