

Projekt z przedmiotu Metody Inteligencji Obliczeniowej

Wizualizacja procesu uczenia SSN

Rafał Jakubczyk
Agnieszka Lasek
Wiktor Żychowicz

1. Opis projektu

W ramach projektu należało zaprojektować prostą sieć neuronową, realizującą dowolne zadanie ze zbioru benchmarkowego UCI MLR <http://archive.ics.uci.edu/ml/datasets.php>, a następnie przedstawić w sposób graficzny, jak w trakcie procesu uczenia zmieniają się poszczególne wagi i biasy.

Sieć neuronowa została utworzona na podstawie zbioru danych Irysów ze strony: <https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>.

Głównym narzędziem wykorzystanym do zaprezentowania wizualizacji poszczególnych parametrów sieci było Weights & Biases. Wykorzystanie biblioteki wandb umożliwiło połączenie się z narzędziem i zaprezentowanie w sposób graficzny zmian występujących w trakcie uczenia pomiędzy poszczególnymi wagami, biasami, wartościami funkcji straty oraz innymi wybranymi parametrami.

Kod źródłowy projektu został udostępniony jako repozytorium na Githubie i jest dostępny pod linkiem:
<https://github.com/MIO-Proj/WizualizacjaProcesuUczeniaSSN> .

Przykładowa wizualizacja procesu uczenia sieci dostępna jest pod linkiem: <https://wandb.ai/laseka/MIO-SSNvisualization/runs/10qojfer?workspace=user-laseka> i została wygenerowana dla 150 epok oraz 2 warstw ukrytych po 12 neuronów każda.

2. Podział pracy

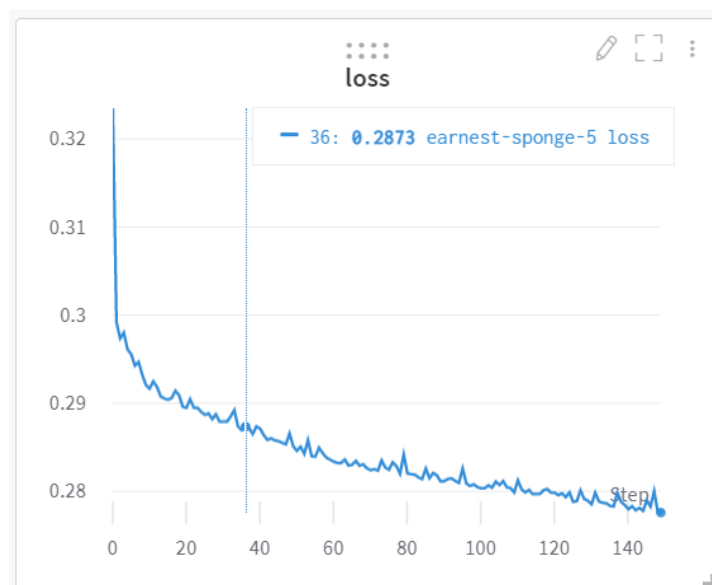
Podział pracy między członkami zespołu wyglądał następująco:

- Rafał Jakubczyk - stworzenie modelu sieci neuronowej, który realizuje problem klasyfikacji dla datasetu Iris oraz tworzenie wizualizacji przy wykorzystaniu biblioteki matplotlib
- Agnieszka Lasek - integracja utworzonej sieci neuronowej z narzędziem Weights & Biases oraz stworzenie w nim wizualizacji procesu uczenia, stworzenie dokumentacji projektu
- Wiktor Żychowicz - stworzenie funkcji odpowiadających za przetwarzanie danych z sieci neuronowej, tworzenie wizualizacji przy wykorzystaniu biblioteki matplotlib

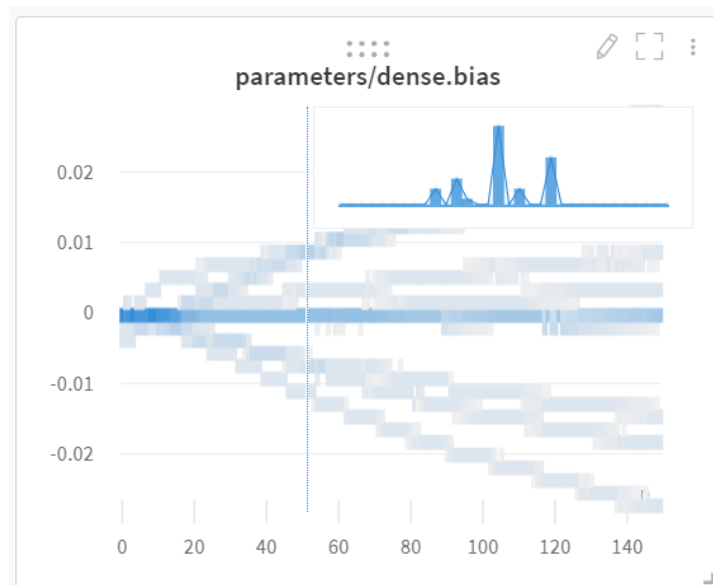
3. Wykorzystane technologie i uzyskane wyniki

Cały projekt został napisany w Pythonie 3, a do jego realizacji zostały wykorzystane m.in. biblioteki: tensorflow, numpy, matplotlib czy wandb. Wizualizacja w projekcie jest realizowana na dwa sposoby: jej najważniejsza część jest przedstawiona na stronie <https://wandb.ai/laseka/MIO-SSNvisualization/runs>.

Przykładowe wizualizacje uzyskane za pomocą wandb:



Wykres przedstawiający zmianę wartości funkcji kosztu dla danych treningowych.

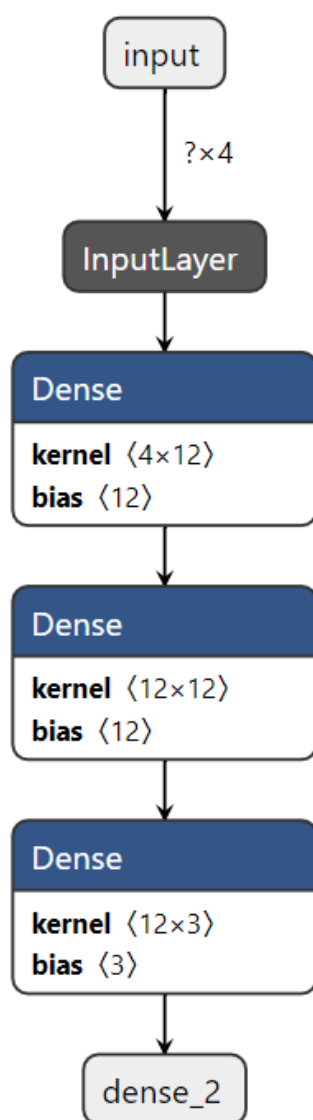


Wykres przedstawiający histogramy biasów dla pierwszej warstwy ukrytej w poszczególnych epokach (natężenie koloru na głównej części wykresu oznacza liczbę neuronów, których bias ma odpowiednią wartość).

Przedstawione tam dane można modyfikować wywołując plik `wandb_visualization.py` dla różnych parametrów uczenia sieci, bądź bezpośrednio w przeglądarce - korzystając z różnorodnych opcji tworzenia nowych wykresów, czy dostosowywania tych już istniejących. Wizualizacje przedstawione w WandB są interaktywne i pozwalają sprawdzić zmiany poszczególnych wartości wag i biasów dla wszystkich etapów uczenia sieci. Wadą jest jednak słaba czytelność niektórych elementów wizualizacji, ze względu na skumulowanie na niektórych wykresach dużej ilości danych.

```
wandb: Run summary:
wandb:      epoch 149
wandb:      loss 0.27758
wandb:    val_loss 0.25346
wandb:    _runtime 19
wandb:   _timestamp 1631475104
wandb:      _step 149
wandb: best_val_loss 0.25311
wandb:   best_epoch 138
wandb: Run history:
wandb:  epoch -----
wandb:   loss -----
wandb: val_loss -----
wandb:  _runtime -----
wandb: _timestamp -----
wandb:    _step -----
wandb:
wandb: Synced 5 W&B file(s), 1 media file(s), 0 artifact file(s) and 1 other file(s)
wandb:
wandb: Synced earnest-sponge-5: https://wandb.ai/laseka/MIO-SSNvisualization/runs/1nt4b0sr
```

Podsumowanie procesu uczenia sieci neuronowej otrzymane dzięki bibliotece wandb po uruchomieniu programu wraz z wygenerowanym linkiem zawierającym szczegóły dotyczące parametrów sieci.



NODE PROPERTIES

type Dense

name dense

ATTRIBUTES

activation relu

batch_input_shape null, 4

bias_initializer Zeros()

dtype float32

kernel_initializer GlorotUniform("GlorotUniform")

trainable true

units 12

use_bias true

INPUTS

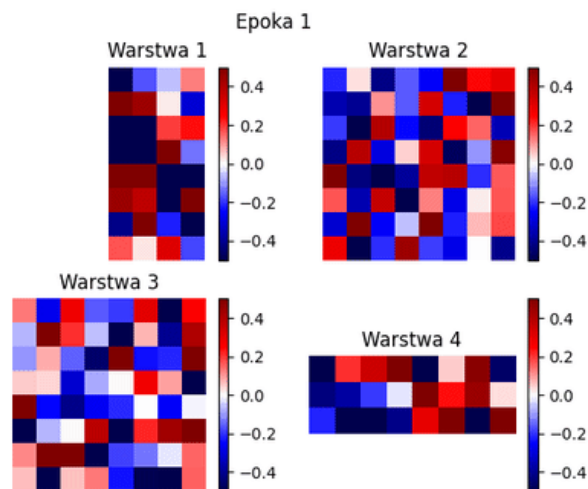
input name: **dense_input**

kernel name: **dense/kernel:0**

Struktura utworzonej przez nas sieci, wygenerowana w wandb (folder Files -> model-best.h5) razem ze szczegółami dotyczącymi poszczególnych warstw.

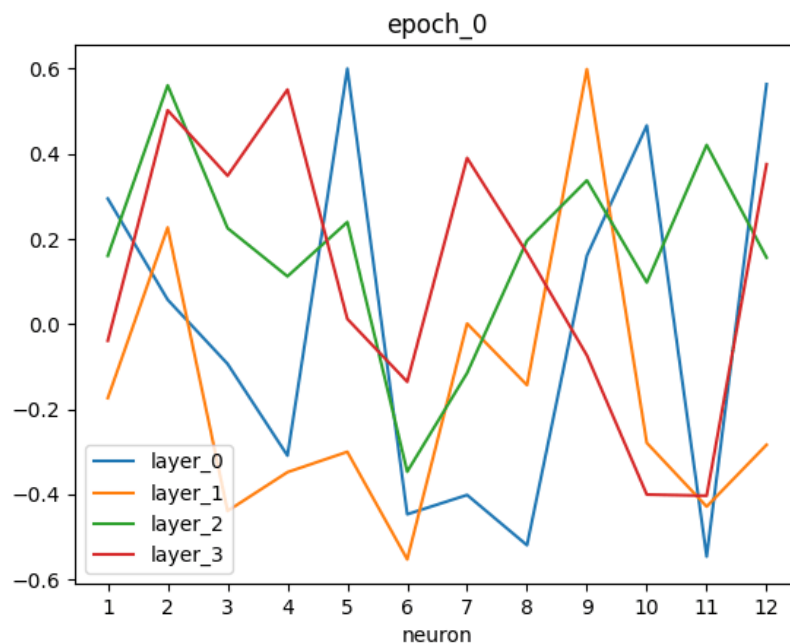
Drugim elementem wykorzystanym w projekcie do wizualizacji procesu uczenia było utworzenie wykresów przedstawiających zmiany wartości m.in. wybranych wag i biasów. Pozwoliło nam to w bardziej czytelny sposób przedstawić, jak zmieniały się poszczególne parametry, jednak uzyskane w ten sposób efekty nie są równie interaktywne, jak wizualizacje w wandb.

Przykładowe wizualizacje otrzymane w ten sposób:



Powyższa animacja przedstawia zmianę wag połączeń między warstwami, na przestrzeni epok. Każda kolumna w warstwie odpowiada wartości wagi neuronu na wejściu warstwy, a wiersze odpowiadają neuronom na wyjściu.

Link do wizualizacji: <https://giphy.com/gifs/3oYj8ZMk25TK79whwX>



Animacja przedstawiająca zmianę biasów dla poszczególnych neuronów w warstwach na przestrzeni epok uczenia.