Sprawozdanie

MSIO – sieci LSTM (ćw. 2, 6, 7)  
Filip Horst 311257

# Przygotowanie danych

## Istotne elementy przekształceń

* Cofnięty szereg czasowy został zrealizowany poprzez obliczenie cofniętej daty dla każdego wiersza, wykonanie klasycznego LEFT JOIN tabeli samej ze sobą z kluczami: data cofnięta – data pomiaru, a następnie usunięcie wierszy posiadających puste komórki
* Cechy h, m, d, wd zostały zamienione na dane cykliczne z wykorzystaniem funkcji trygonometrycznych. Metoda zainspirowana źródłem: <https://datascience.stackexchange.com/questions/5990/what-is-a-good-way-to-transform-cyclic-ordinal-attributes>

## Podział na zbiory

W obu przypadkach podział został wykonany tak, aby zapewnić mniej więcej te same proporcje zbioru treningowego do walidacyjnego. Zbiór testowy składał się w obu wersjach tylko z danych 2012 roku.

1. Podział pierwszy: trening na latach 2010, 2011
   1. Trening: 15010 57.09%
   2. Walidacja 2502 9.52%
   3. Test 8780 33.39%
2. Podział drugi trening 2009, 2010, 2011
   1. Trening: 22454 64.20%
   2. Walidacja 3742 10.70%
   3. Test: 8780 25.10%

## Zestawy cech wejściowych

### Słownik kodowania

|  |  |
| --- | --- |
| Kod | Znaczenie |
| Y, Sin\_h, cos\_h, sin\_m, cos\_m, sin\_d, cos\_d, Sin\_wd, cos\_wd | Data. Godzina, miesiąc, dzień i dzień tygodnia (wd) zapisane w formie cyklicznej |
| Pkb | PKB kraju [miliardy USD] |
| Inflation | Stopa inflacji |
| Pop\_growth | Tempo wzrostu liczby ludności |
| Tax\_in | Wpływy podatkowe [%PKB] |
| Poor\_rate | Wskaźnik liczby osób ubogich według oceny krajowej |
| Child\_avg | Średnia dzietność na kobietę |
| Export | Eksport |
| Goods\_out | Wydatki PKB %-Saldo zewn. Towarów i usług |
| Ddc | Wsp. Zależnośći demograficznej |
| Gini\_idx | Wskaźnik Giniego |
| Articles | Artykuły prasowe naukowe i techniczne [tys. szt] |
| Teleinf\_export | Eksport towarów teleifnormatycznych jako % całkowitego eksportu |
| Listed\_comps | Krajowe spółki giełdowe łącznie |
| Credit\_priv | Kredyt krajowy udzielony dla sektora prywatnego (% PKB) |
| Receive\_priv | Należności od sektora prywatnego (roczny wzrost jako % podaży pieniądza) |
| Savings | Oszczędności (% PKB) |
| Demand\_d-1, demand\_d-2, demand\_d-3, demand\_avgy, demand\_avgm | Cofnięte wartości zapotrzebowania oraz średnie roczne/miesięczne |

### Skład zbiorów

1. Wszystkie kolumny

['sin\_h','cos\_h', 'sin\_m','cos\_m', 'sin\_d','cos\_d', 'y', 'sin\_wd','cos\_wd', 'pkb', 'inflation', 'pop\_growth', 'tax\_in', 'poor\_rate', 'child\_avg', 'export', 'goods\_out', 'ddc', 'gini\_idx', 'articles', 'teleinf\_export', 'listed\_comps', 'credit\_priv', 'receive\_priv', 'savings', 'demand\_d-1', 'demand\_d-2', 'demand\_d-3', 'demand\_avgy', 'demand\_avgm']

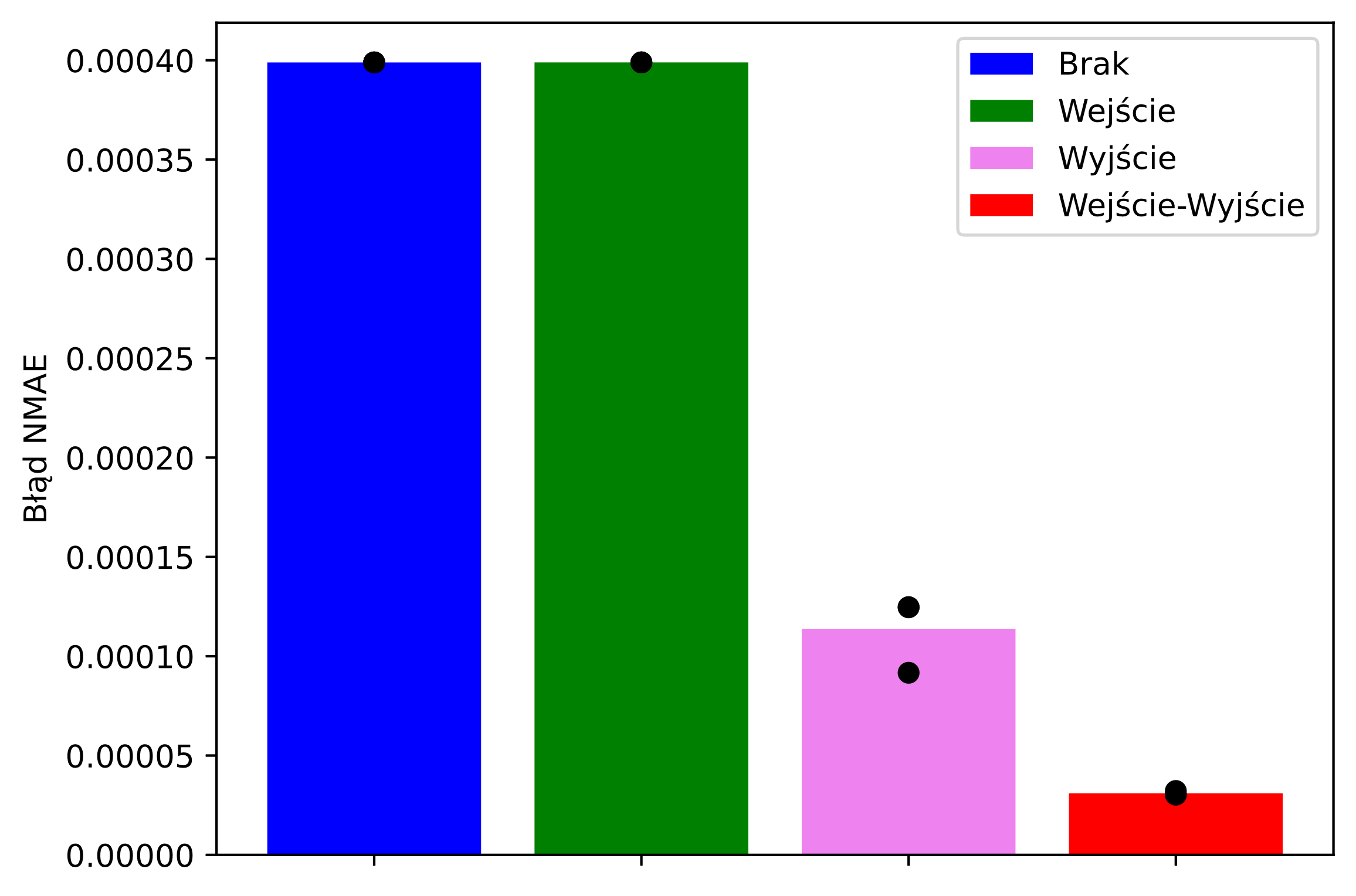
1. Tylko dane historyczne o zapotrzebowaniu

['demand\_d-1','demand\_d-2','demand\_d-3','demand\_avgy','demand\_avgm']

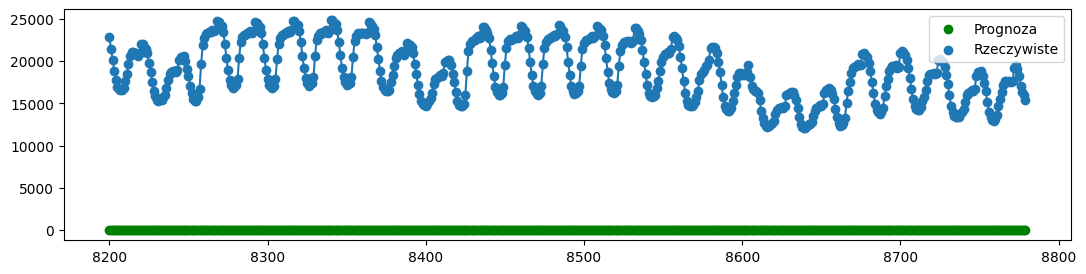
1. Tylko kolumny z korelacją większa niż 0.1. Korelacja dla danych cyklicznych została obliczona przed ich zamianą. Próg korelacji został przyjęty tak, aby trochę zmniejszyć rozmiar zbioru wejściowego i przeanalizować wpływ braku mniej skorelowanych danych na wyniki.

['sin\_h','cos\_h','y','sin\_wd','cos\_wd','pkb','inflation','tax\_in','export','goods\_out','gini\_idx','listed\_comps','credit\_priv', 'receive\_priv','demand\_d-1','demand\_d-2','demand\_d-3','demand\_avgy','demand\_avgm']

# Normalizacja

Badanie wpływu normalizacji odbyło się jako pierwsze z użyciem domyślnych parametrów dostarczonych w szkielecie programu. Użyta została normalizacja metodą Min-Max.  
  
  
Błąd NMAE w zależności od normalizacji wartości wejściowych i wyjściowych

Na podstawie wykresu można powiedzieć, że normalizacja znacznie zwiększa efektywność nauki sieci i jej jakość w zadanej liczbie iteracji. Spośród scenariuszy z normalizacją tylko wejścia/wyjścia ważniejsze okazało się wyjście.

  
Prognoza zwrócona przez jeden z modeli z normalizacją tylko wejścia w odniesieniu do wartości rzeczywistych

Patrząc na przykładową prognozę dla normalizacji tylko wejścia można powiedzieć, że utworzona sieć jest wadliwa.

**Wniosek**

Normalizacja danych wejściowych jest obowiązkowa w sieci LSTM. Normalizacja wyjścia jest opcjonalna, ale silnie zalecana.

# Przyjęte założenia

Ze względu na ograniczoną moc komputera osobistego oraz brak możliwości „zapuszczenia na noc” (mieszkanie w akademiku + hałasujący system chłodzenia przy intensywnej pracy) przyjęte zostały następujące założenia:

* Batch\_size 512
* Maksymalna liczba epok 250
* Dobór parametrów sieci po kolei metodą zachłanną

Jest to oczywiste, że te założenia obniżają jakość modelu końcowego oraz utrudniają wybór rzeczywiście optymalnych parametrów, ale był to wybór świadomy podjęty głównie na podstawie skrajnej czasochłonności projektu w dostępnych warunkach.

# Parametry LSTM

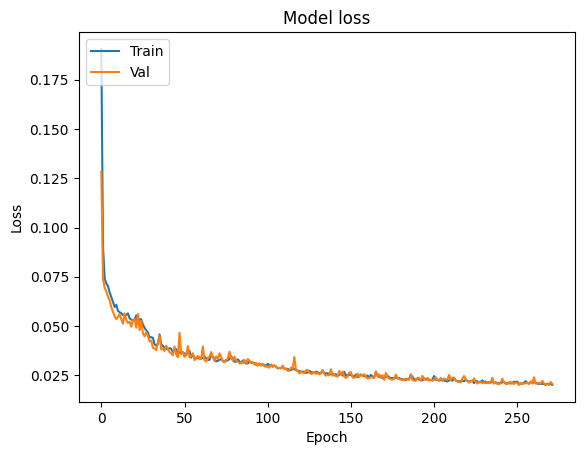
## Zbiór 10,11 – wszystkie kolumny

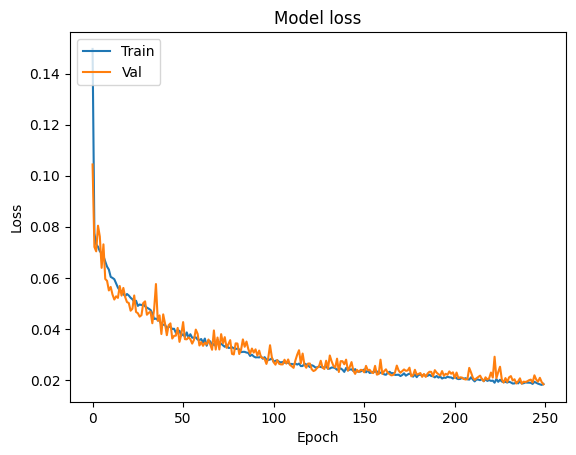
Wybrane parametry:

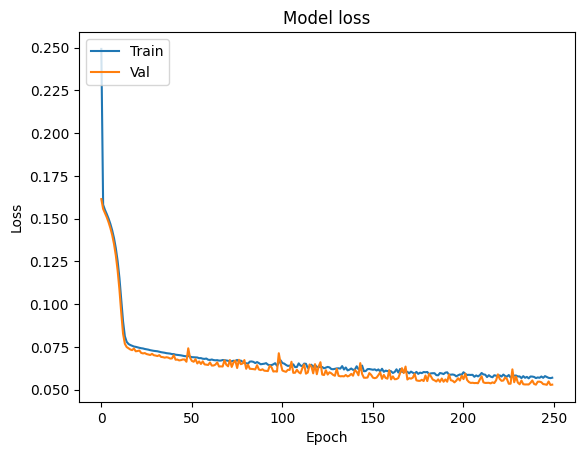
* 2 warstwy o rozmiarach 68, 25
* F. Aktywacji tanh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik MAE\_VAL  (odchylenie std.) | Najlepszy wynik MAE\_VAL |
| Adam | 0.0055 | 103.92 (38.72) | 0.0203 (0.00109) | 0.0192 |
| Nadam | 0.0065 | 67.78 (9.90) | 0.0195 (0.00122) | 0,0183 |
| AdaGrad | 0.035 | 66.65 (17.66) | 0.0545 (0.00170) | 0,0529 |

Na podstawie prostych badań statystycznych udało się ustalić, że AdaGrad jest gorszym optymalizatorem w tym problemie. Spośród Adam oraz Nadam okazało się, że wersja „N” daje podobne wyniki, ale jest bardziej konsekwentna pod względem czasu wykonania.

  
Wykres uczenia dla Adam

  
Wykres uczenia dla Nadam

  
Wykres uczenia dla AdaGrad

Analiza przebiegów uczenia pokazuje, że występują znaczące oscylacje – szczególnie przy metodzie Nadam. Jest to sugestia, że krok uczenia może być zbyt duży jednak takie ustawienie daje lepsze wyniki. Metoda AdaGrad, która osiągnęła gorsze wyniki końcowe ma znacznie „spokojniejszy” wykres uczenia biorąc pod uwagę występowanie oscylacji. Biorąc to pod uwagę, istnieje szansa, że ta metoda szczególnie cierpi przez wymuszone ograniczenie liczby epok.

Najlepszy wynik uzyskany dla optymalizatora Nadam:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Val | AeMax Val |
| 0,018495 | 1,46E-05 | 1,53E-07 | 0,345688 | 0,058697 | 1,29E-05 | 4,94E-05 | 0,697104 |

## Zbiór 10,11 – tylko dane historyczne zapotrzebowania

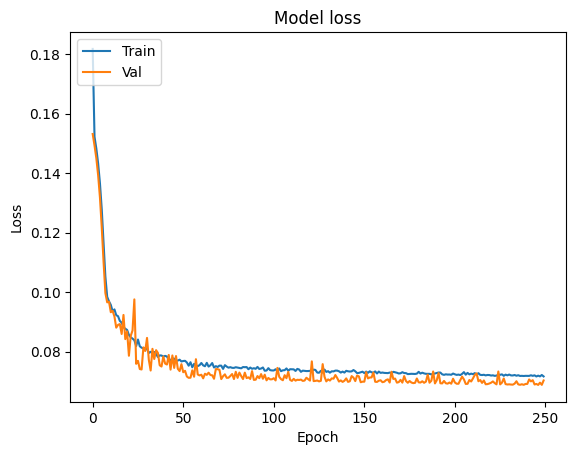
Wybrane parametry:

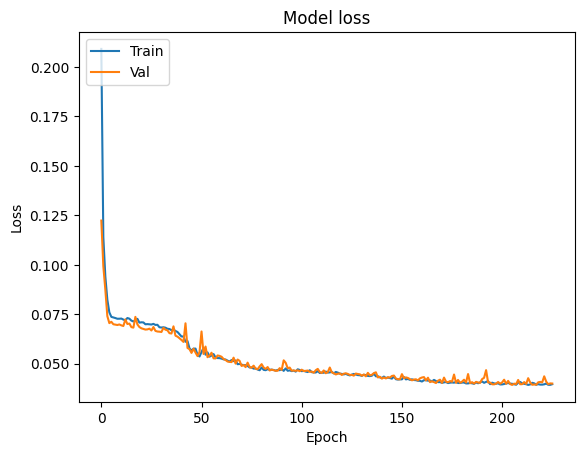
* 2 warstwy o rozmiarach 64 i 25
* F. aktywacji tanh

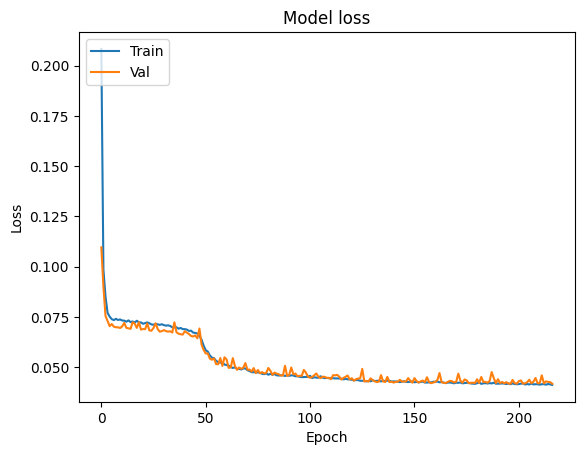
Fakt, że właśnie takie ustawienie dawało najlepsze wyniki jest bardzo zaskakujący. Mimo bardzo poważnego zmniejszenia złożoności danych wejściowych przez ograniczenie liczby kolumn do zaledwie 5, wyniki wciąż były najlepsze dla tak złożonej sieci. Badanie parametrów zostało przeprowadzone startując od wartości domyślnych czyli warstw wielkości 8 i 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE\_VAL  (odchylenie std.) | Najlepszy wynik MAE\_VAL |
| Adam | 0.005 | 52.35 (7.11) | 0.0406 (0.0012) | 0.0396 |
| Nadam | 0.0045 | 53.09 (3.06) | 0.0420 (0.0014) | 0.0403 |
| AdaGrad | 0.1 | 50.63 (28.66) | 0.0714 (0.0036) | 0.0692 |

Podobnie jak wcześniej, Adam oraz Nadam dają bardzo podobne wyniki. W tym przypadku to jednak pierwszy z nich był minimalnie lepszy. Co do czasu nauki, wciąż zauważalna jest większa stabilność algorytmu Nadam.

  
Wykres nauki dla AdaGrad

  
Wykres nauki dla Adam

  
Wykres nauki dla Nadam

Z obserwacji wykresów uczenia można ponownie wywnioskować, że liczba epok jest zbyt mała. Ponadto, widoczne są oscylacje, szczególnie w przypadku AdaGrad.

Najlepszy wynik uzyskano dla optymalizatora Adam:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Val | AeMax Val |
| 0,039696 | 3,14E-05 | 1,65E-05 | 0,426179 | 0,041778 | 9,15E-06 | 7,53E-06 | 0,531195 |

## Zbiór 10,11 – kolumny korelacja > 0.1

Wybrane parametry:

* 2 warstwy o rozmiarach 75 i 17
* Funkcja aktywacji tanh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE\_VAL  (odchylenie std.) | Najlepszy wynik MAE\_VAL |
| Adam | 0.008 | 87.37 (7.13) | 0.02085 (0.00089) | 0.01983 |
| Nadam | 0.0045 | 80.40 (7.61) | 0.02348 (0.00284) | 0.02172 |
| AdaGrad | 0.008 | 98.15 (10.96) | 0.06498 (0.00041) | 0.06472 |

Podobnie do poprzednich przypadków, optymalizator AdaGrad wyróżniał się gładszym, ale wolniejszym wykresem uczenia. Niestety ze względu na czasochłonność ćwiczenia na komputerze osobistym (prawie 1.5 minut na naukę jednego modelu) nie udało się zbadać dalszych etapów nauki. Wyjątkiem były testy pojedynczych ustawień dla 500 iteracji, co mimo znacznego wydłużenia czasu nauki wciąż nie pozwoliło się zbliżyć do punktu przegięcia – kompromisu błędu treningowego i walidacyjnego. Optymalizatory Adam i Nadam osiągały lepsze wyniki, ale jak widać po odchyleniu standardowym oscylacje były dość spore.

Najlepsze wyniki dla optymalizatora Adam:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Val | AeMax Val |
| 0,019835 | 1,57E-05 | 1,89E-06 | 0,393794 | 0,207848 | 4,55E-05 | 5,31E-06 | 0,899838 |

## Zbiór 09,10,11 – wszystkie kolumny

## Zbiór 09,10,11 – tylko dane historyczne zapotrzebowania

## Zbiór 09,10,11 – kolumny korelacja > 0.1

# Wyniki łączne i wnioski

# Problemy obliczeniowe

