# Sprawozdanie

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

## 1 Przygotowanie danych

### 1.1 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. Implementacja zainspirowana źródłem: https://datascience.stackexchange.com/questions/5990/what-is-a-good-way-to-transform-cyclic-ordinal-attributes

### 1.2 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
    a. Trening: 15010 57.09%
    b. Walidacja 2502 9.52%
    c. Test 8780 33.39%
2. Podział drugi: trening 2009, 2010, 2011
    a. Trening: 22454 64.20%
    b. Walidacja 3742 10.70%
    c. Test: 8780 25.10%

### 1.3 Zestawy cech wejściowych

#### 1.3.1 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 |

### 1.3.2 Skład zbiorów

1. Wszystkie kolumny. Kod „**wszystkie**”

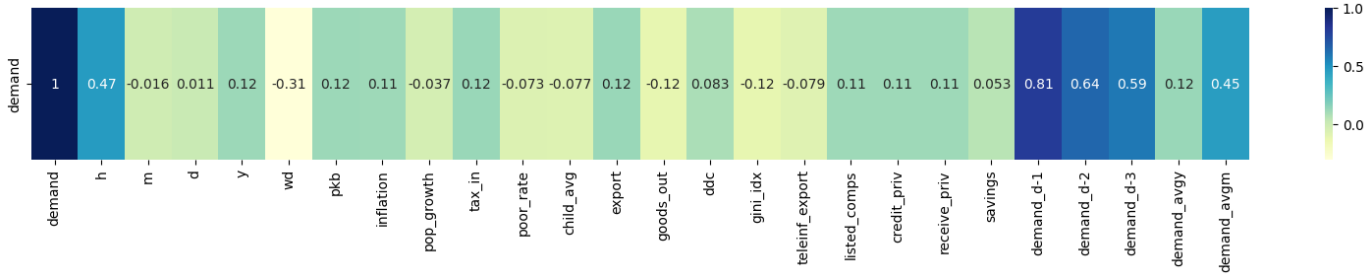
['sin_h','cos_h', 'sin_m','cos_m', 'sin_d','cos_d', 'y', 'sin_wd','cos_wd', 'pkb', 'inflation', 'pop_growth', 'tax_in', 'poor_rate', 'child_avg', 'export', 'goods_out', 'ddc', 'gini_idx', 'articles', 'teleinf_export', 'listed_comps', 'credit_priv', 'receive_priv', 'savings', 'demand_d-1', 'demand_d-2', 'demand_d-3', 'demand_avgy', 'demand_avgm']

2. Tylko dane historyczne o zapotrzebowaniu. Kod „**historia zapotrzebowania**”

['demand_d-1','demand_d-2','demand_d-3','demand_avgy','demand_avgm']

3. Tylko kolumny z korelacją większą niż 0.1 (wart. Bezwzględna). 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 (potencjalnie mniej użytecznych) danych na wyniki.
   Kod „**korelacja > 0.1**”

['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']




Prosta wizualizacja przedstawiająca korelacje zmiennych wejściowych z wyjściem. Użyte zostały te, których wartość bezwzględna była większa od 0.1

Wszystkie zbiory były podzielone na podzbiory treningowy i walidacyjny, gdzie treningowy był używany tylko do uczenia modeli, natomiast na walidacyjnym wykonywane były prognozy pozwalające na dostosowanie hiperparametrów i innych ustawień.

Co do samych zmiennych, najbardziej wartościowe okazują się godzina, dzień tygodnia oraz wartości cofnięte. Oczywiście godzina i dzień tygodnia są ważne, ale można podejrzewać, że to nie ich dokładna wartość ma znaczenie ale bardziej ich dane przedziały np. poranek, południe, bądź dzień tygodnia, sobota, niedziela. Ciekawym dalszym badaniem było by użycie takich zmiennych o charakterze jakościowym, by sprawdzić czy ta hipoteza jest słuszna.

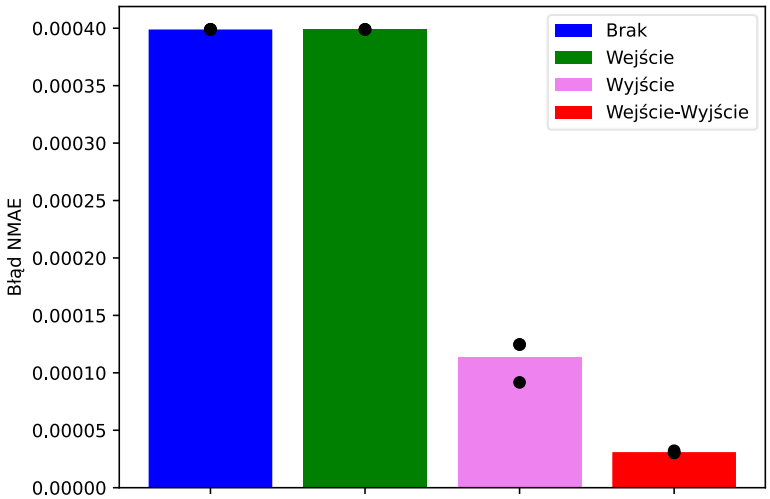
Dane historyczne w postaci średniej miesiąca są również bardzo wartościowe w przeciwieństwie do grupowania według roku. Te dwie zmienne mają jednak pewną cechę, która utrudnia ich użycie w produkcji, a mianowicie trzeba znać pełen zakres czasowy by je obliczyć. W sytuacji rzeczywistej prognozując np. w czasie wiosny, średnia roczna była by niemożliwa do obliczenia, a przybliżenie wykorzystując dane z początku roku przyniosło by złe wyniki, bo zużycie w zimie może być zauważalnie wyższe. Oznacza to, że użycie takich zmiennych może być bez sensu, mimo że poprawią wyniki przy projektowaniu modelu.

Wartości cofnięte o 1, 2 oraz 3 dni są wszystkie bardzo użyteczne jednak najmniejsze cofnięcie bez zaskoczenia jest najlepiej skorelowane. Ciekawe jest zjawisko, że korelacja dla cofnięć o 2 i 3 dni jest bardzo podobna do siebie.

Pozostałe dane związane z krajem, ekonomią itp. są raczej słabo skorelowane, ale wiedząc, że korelacja określa tylko liniowe zależności nie należy ich wyrzucać bez sprawdzenia ich znaczenia – jedną z metod jest właśnie badanie wielu zbiorów cech wejściowych przeprowadzone dalej.

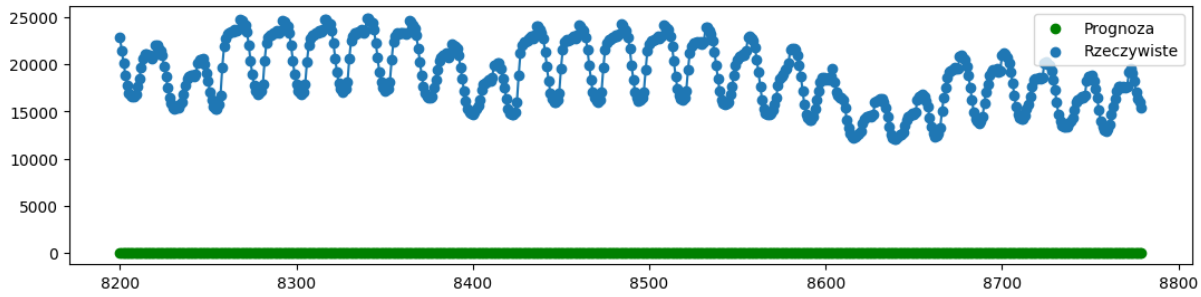
# 2 Normalizacja

Badanie wpływu normalizacji odbyło się jako pierwsze z użyciem domyślnych hiperparametrów dostarczonych w szkielecie programu. Użyta została normalizacja metodą Min-Max.




Błąd średni NMAE na zbiorze walidacyjnym 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.

# 3 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 hiperparametrów sieci po kolei metodą zachłanną

Jest to oczywiste, że te założenia obniżają jakość modelu końcowego oraz utrudniają wybór rzeczywiście optymalnych hiperparametrów, ale był to wybór świadomy podjęty głównie na podstawie ogromnej czasochłonności projektu przy dostępnych możliwościach. Po wstępnym wyborze hiperparametrów przy tych większych ograniczeniach, najlepsze z nich zostały zbadane dokładniej dla większej liczby epok.

Wadą metody zachłannej jest to, że nie można przygotować sensownych wykresów zależności błędów od ustawień – praktycznie wszystko było by parabolą. W zastosowanym podejściu po dobraniu danego ustawienia dla jednego hiperparametru, na samym końcu było ono ponownie sprawdzane, więc praca była w pewnym stopniu iteracyjna – pozwoliło to lepiej dopasować hiperparametry do problemu.

# 4 Hiperparametry LSTM

Zawiera dokładniejsze wnioski dla poszczególnych zestawów zakresu danych wejściowych i cech

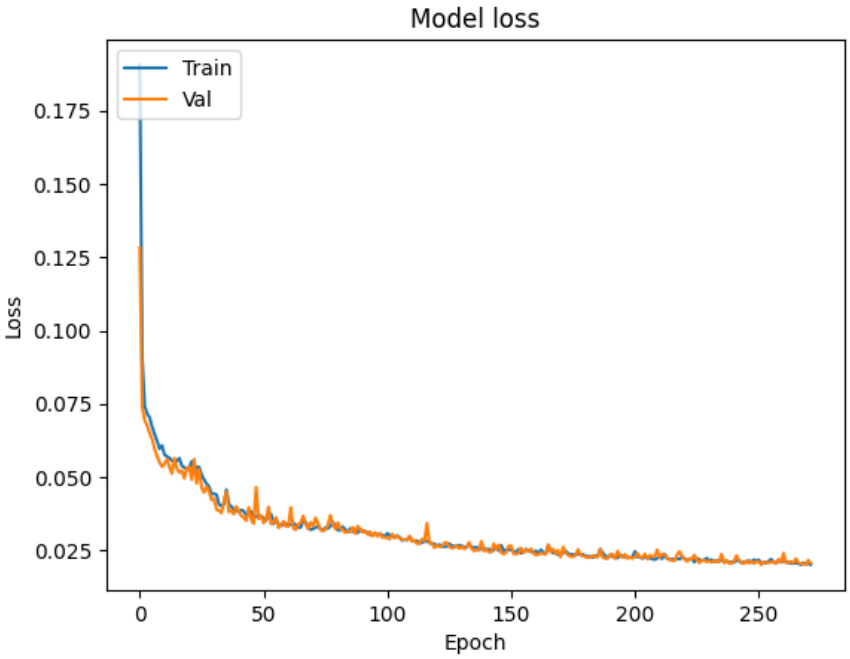
## 4.1 Zbiór 10,11 – wszystkie kolumny

Wybrane hiperparametry:

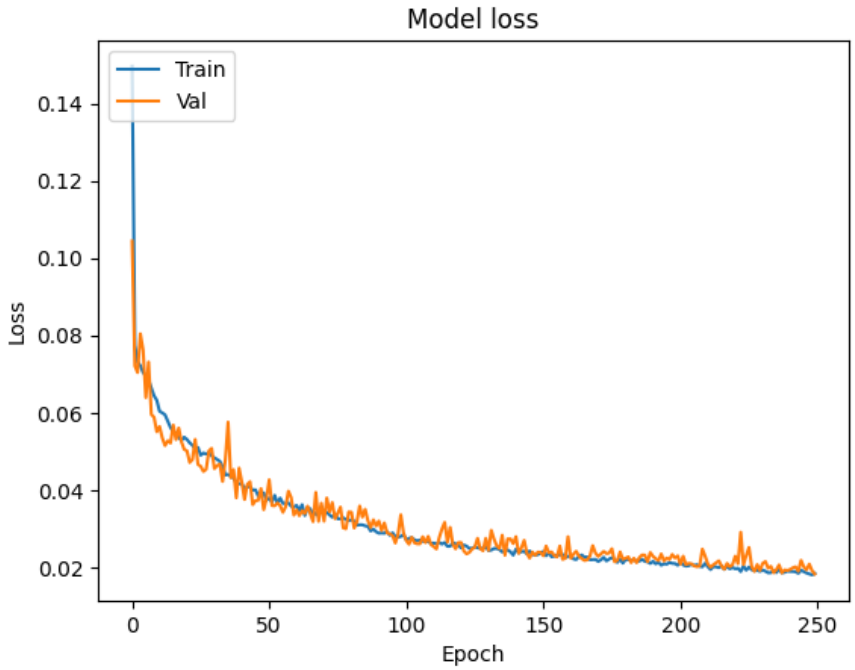
- 2 warstwy o rozmiarach 68, 25
- F. Aktywacji tanh

| 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 |

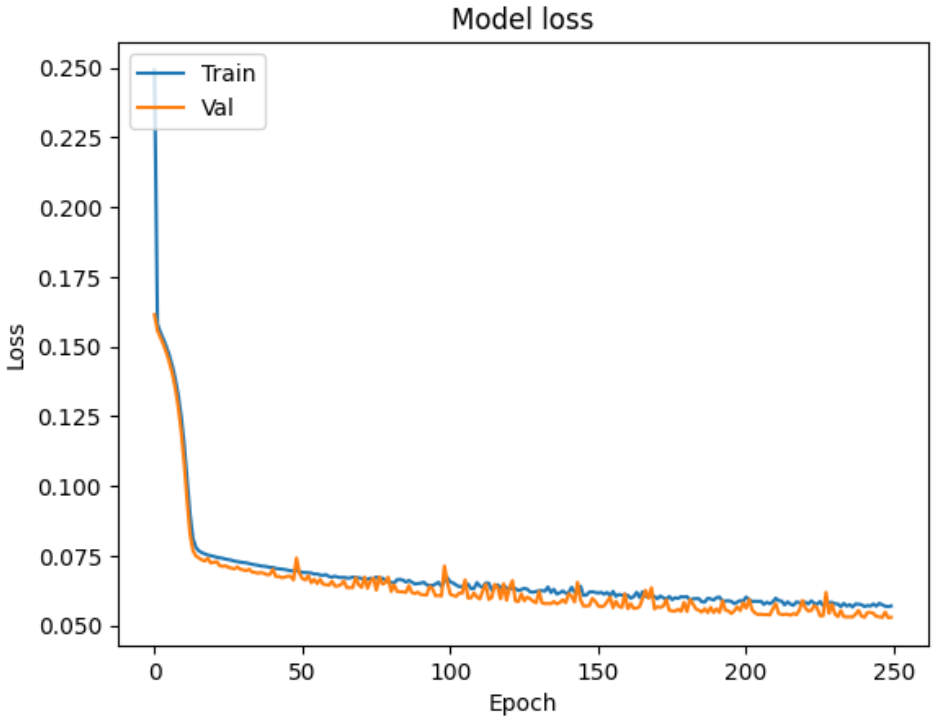
Na podstawie prostych badań statystycznych udało się ustalić, że AdaGrad jest gorszym optymalizatorem w tym problemie. Spośród Adam oraz Nadam okazało się, że wersja „N" daje podobne wyniki, ale jest bardziej konsekwentna pod względem czasu wykonania.




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 Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,018495 | 1,46E-05 | 1,53E-07 | 0,345688 | 0,058697 | 1,29E-05 | 4,94E-05 | 0,697104 |

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

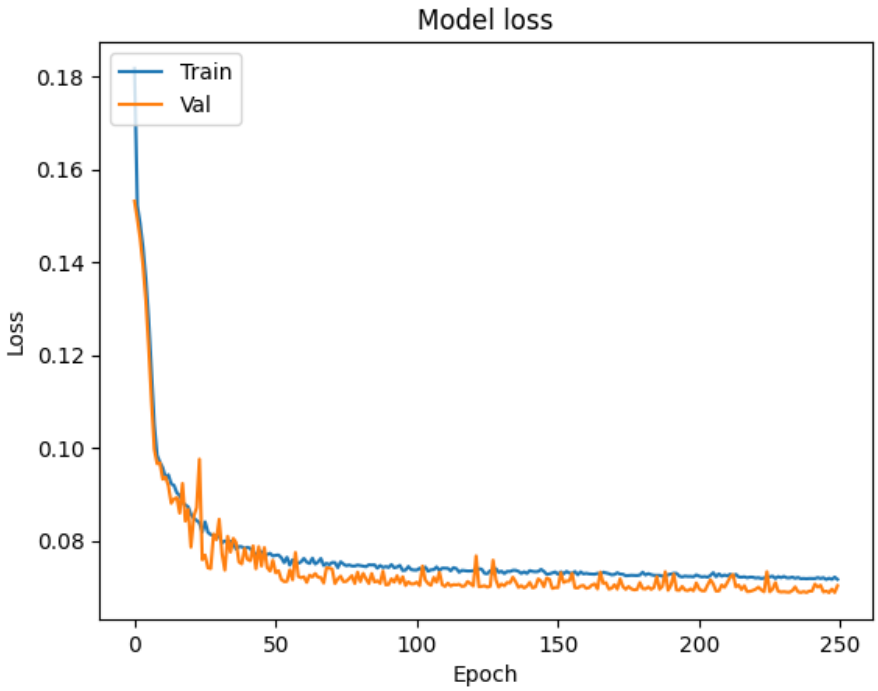
Wybrane hiperparametry:

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

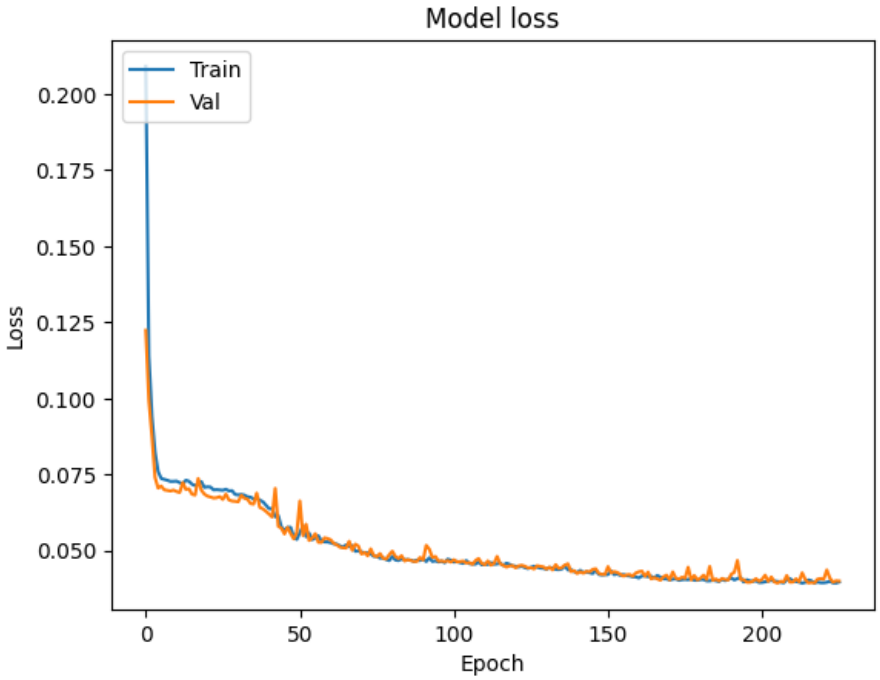
Fakt, że właśnie takie ustawienie dawało najlepsze wyniki jest bardzo zaskakujący. Mimo bardzo poważnego zmniejszenia złożoności danych wejściowych przez ograniczenie liczby kolumn do zaledwie 5, wyniki wciąż były najlepsze dla tak złożonej sieci. Badanie hiperparametrów zostało przeprowadzone startując od wartości domyślnych, czyli warstw wielkości 8 i 4, aby zmniejszyć ryzyko utknięcia w jednym minimum lokalnym. Istnieje jednak szansa, że ustawienia domyślne znajdują się na spadku prowadzącym do tych samych rozmiarów warstw za każdym razem – eksploracja nie przyniosła jednak innych obiecujących kierunków badań.

| 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 |

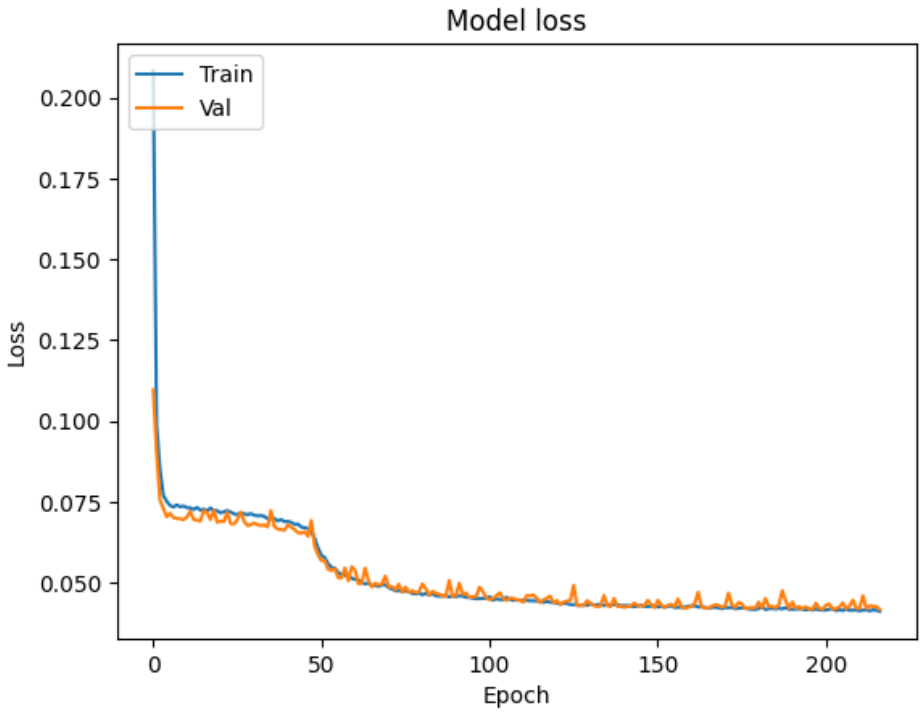
Podobnie jak wcześniej, Adam oraz Nadam dają bardzo podobne wyniki. W tym przypadku to jednak pierwszy z nich był minimalnie lepszy. Co do czasu nauki, wciąż zauważalna jest większa stabilność algorytmu Nadam.




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. Jest to jednak skutek bardzo dużego kroku uczenia ustawionego jako eksperyment, czy AdaGrad jest w stanie dorównać szybkością zbieżności pozostałym optymalizatorom – na bazie tych wyników odpowiedź brzmi nie.

Najlepszy wynik uzyskano dla optymalizatora Adam:

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

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

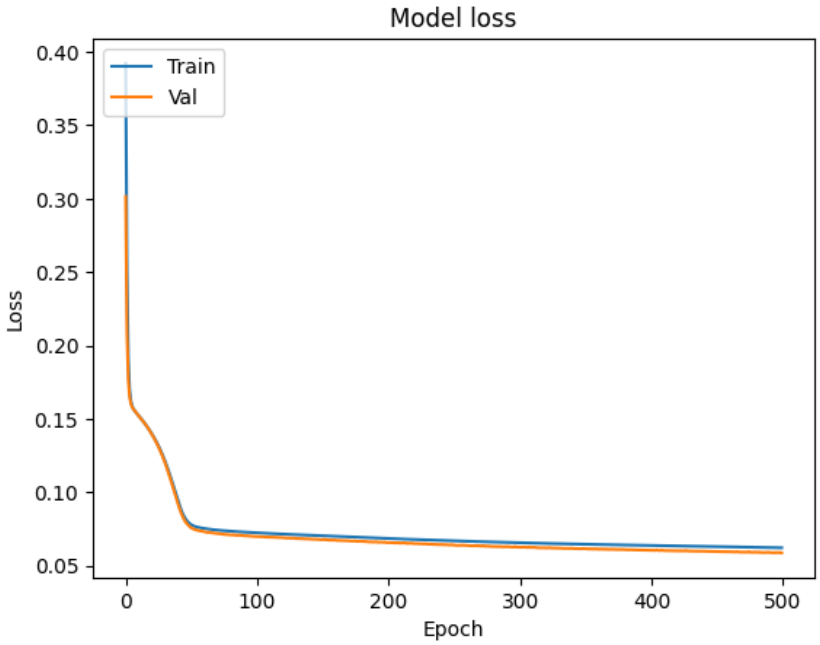
Wybrane hiperparametry:

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

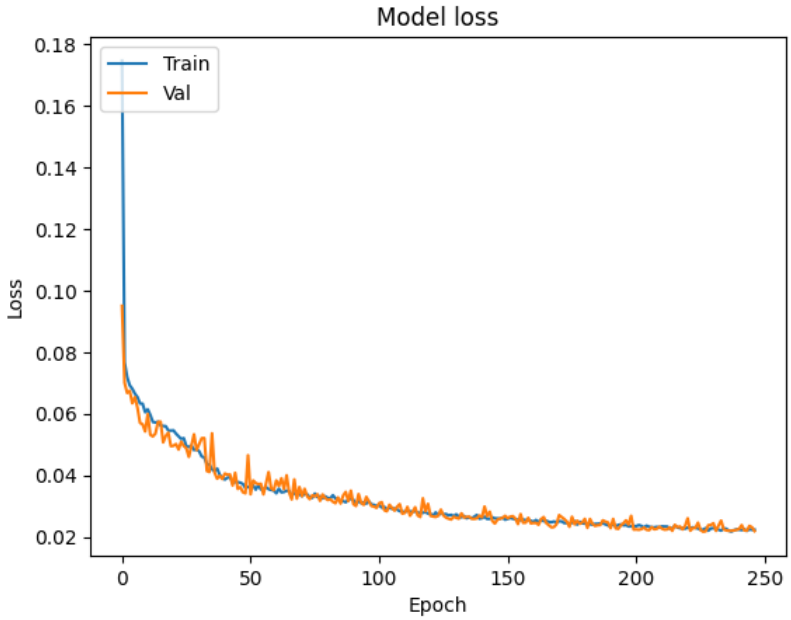
| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE_VAL (odchylenie std.) | Najlepszy wynik MAE_VAL |
|---|---|---|---|---|
| Adam | 0.008 | 87.37 (7.13) | 0.02085 (0.00089) | 0.01983 |
| Nadam | 0.0045 | 80.40 (7.61) | 0.02348 (0.00284) | 0.02172 |
| AdaGrad | 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.




Wykres uczenia dla 500 iteracji Adagrad




Wykres uczenia dla Nadam

Wykresy uczenia wyglądają analogicznie do poprzednich przypadków – duże oscylacje dla Nadam oraz spokojny, ale wolno minimalizujący błąd przebieg dla Adagrad.

Najlepsze wyniki dla optymalizatora Adam:

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0,019835 | 1,57E-05 | 1,89E-06 | 0,393794 | 0,207848 | 4,55E-05 | 5,31E-06 | 0,899838 |

## 4.4 Zbiór 09,10,11 – wszystkie kolumny

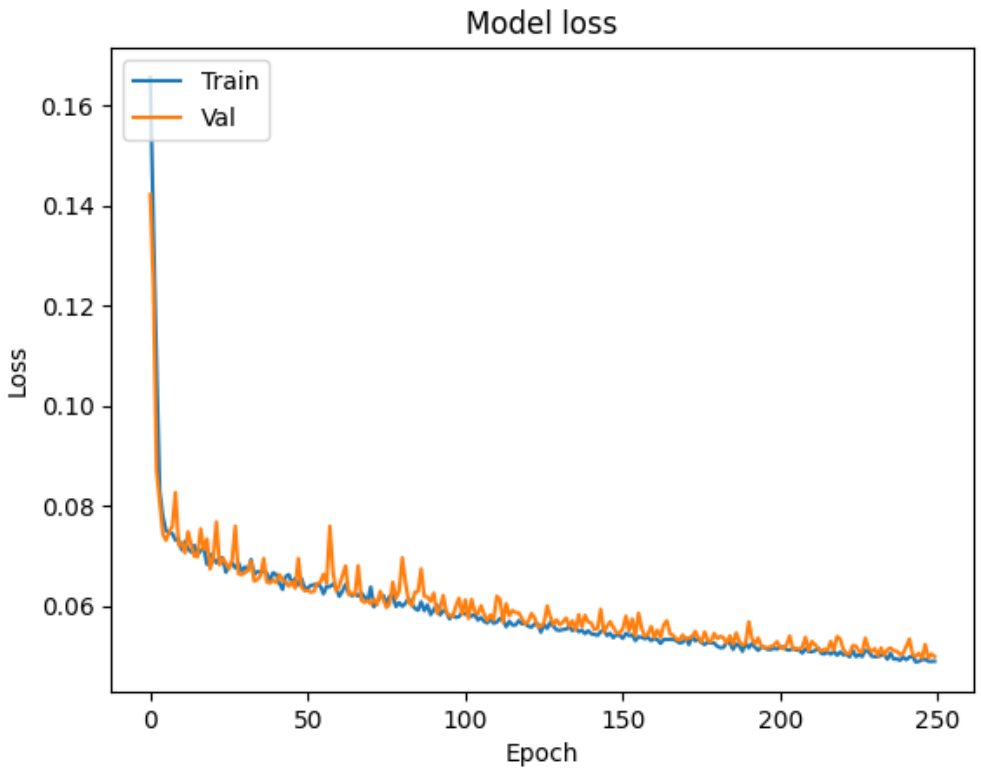
Wybrane hiperparametry:

- 2 warstwy o rozmiarach 65 i 30
- Funkcja aktywacji tanh

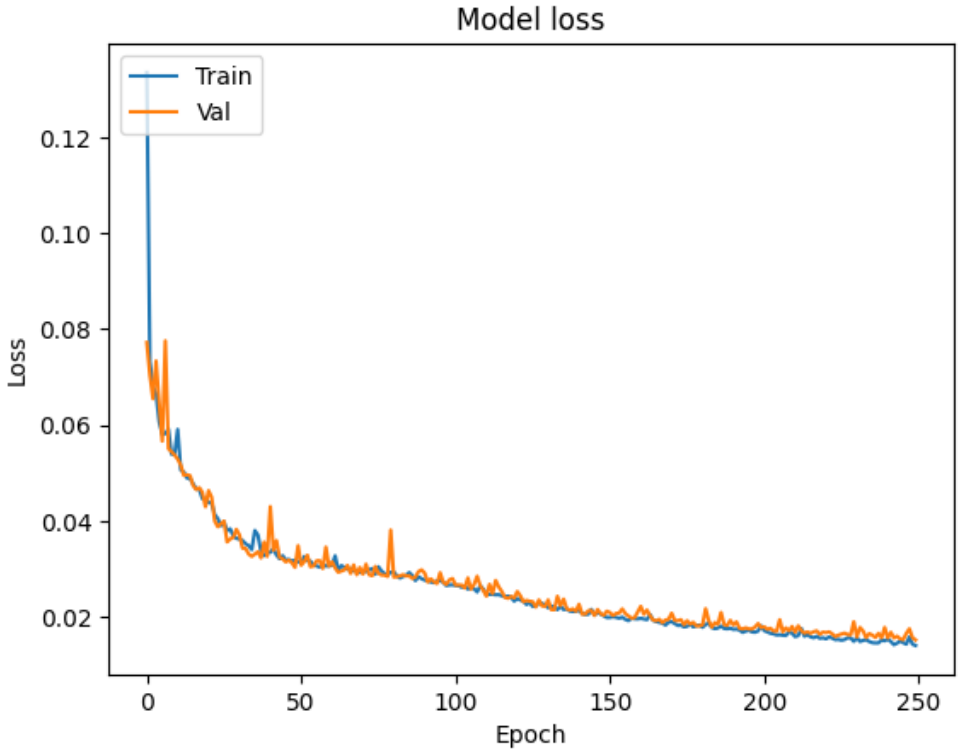
Można zaobserwować, że hiperparametry są podobne do zbioru z tylko dwóch lat, jednak co jest zaskakujące to fakt, że sieć dla trzech lat jest mniej złożona jak chodzi o liczbę neuronów. Pewną hipotezą tego zjawiska może być, że dodatkowe dane są na tyle wartościowe jak chodzi o modelowanie procesu, że pozwalają ograniczyć złożoność samego modelu.

| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE_VAL (odchylenie std.) | Najlepszy wynik MAE_VAL |
|---|---|---|---|---|
| Adam | 0.009 | 69.38 (9.25) | 0.01679 (0.00092) | 0.01588 |
| Nadam | 0.01 | 72.30 (6.76) | 0.01472 (0.00060) | 0.01407 |
| AdaGrad | 0.08 | 63.89 (1.84) | 0.05107 (0.00143) | 0.04990 |

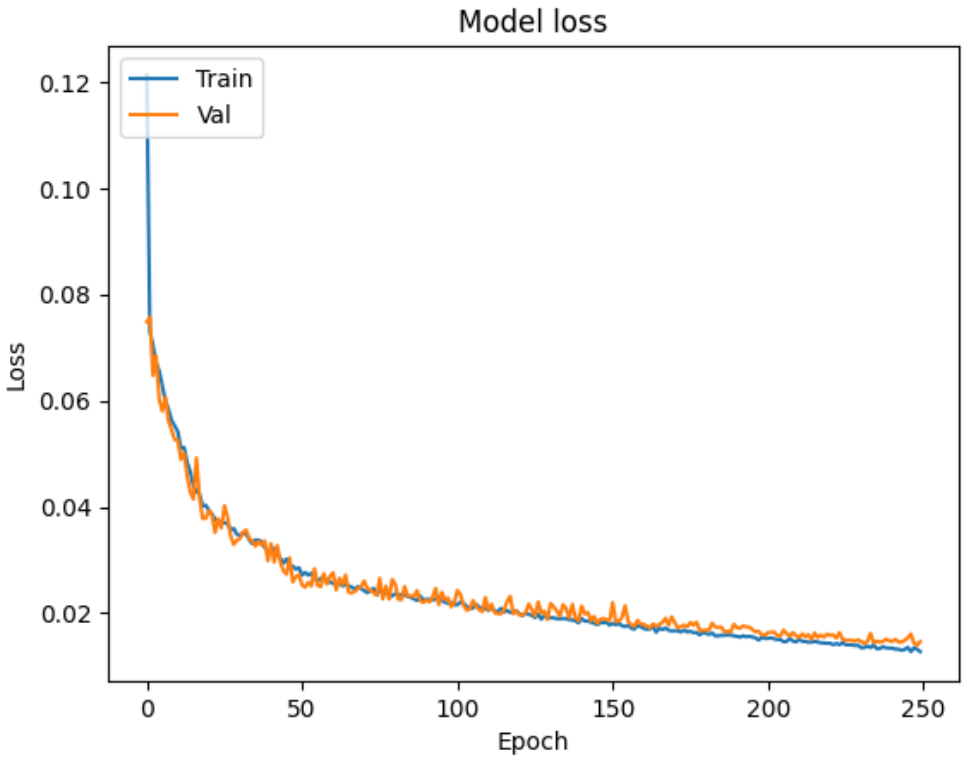
Wyniki na zbiorze walidacyjnym w porównaniu do poprzednich modeli się poprawiły, jednak teraz jest używany zbiór trzyletni, co oznacza że zawartość zbioru walidacyjnego się zmieniła i nie można uznać tych zmian za reprezentatywne – odpowiedź da dopiero porównanie najlepszych modeli na zbiorze testowym.




Wykres uczenia dla AdaGrad




Wykres uczenia dla Adam




Wykres uczenia dla Nadam

Wykresy uczenia pokazują, że mimo dużego kroku oscylacje nie są nieakceptowalnie duże. Co prawda, ich wygląd wciąż jest daleki od ideału, ale zarówno na podstawie względnego odchylenia

standardowego, jak i na bazie oceny wizualnej zdają się być lepsze niż przy zbiorze dwuletnim. Wciąż widoczne jest zjawisko braku wypłaszczenia powodowane ograniczeniem liczby epok.

Najlepsze wyniki osiągnięto dla Nadam:

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,014667 | 7,99E-06 | 1,07E-06 | 0,377291 | 0,047836 | 1,05E-05 | 5,55E-06 | 0,557811 |

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

Od tego etapu optymalizator AdaGrad został pominięty z powodu niskiej skuteczności przy przyjętych niesprzyjających założeniach limitu epok.

Wybrane hiperparametry:

- Warstwy o rozmiarach 50 i 20
- Funkcja aktywacji Tanh

| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE_VAL (odchylenie std.) | Najlepszy wynik MAE_VAL |
|---|---|---|---|---|
| Adam | 0.0065 | 56.63 (1.23) | 0.03912 (0.00179) | 0.03796 |
| Nadam | 0.008 | 60.05 (4.66) | 0.03794 (0.00122) | 0.03718 |

Podobnie jak w zbiorze dwuletnim, ten zestaw cech wejściowych okazał się mało skuteczny i jest dużo gorszy od pozostałych dla tego zakresu danych.

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,037182 | 2,03E-05 | 2,37E-06 | 0,465217 | 0,040486 | 8,87E-06 | 5,15E-07 | 0,503959 |

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

Wybrane hiperparametry:

- Warstwy o rozmiarach 55 i 25
- Funkcja aktywacji tanh

| Optymalizator | Dobrany learning rate | Średni czas nauki (odchylenie std.) | Wynik średni MAE_VAL (odchylenie std.) | Najlepszy wynik MAE_VAL |
|---|---|---|---|---|
| Adam | 0.007 | 58.89 (2.59) | 0.02073 (0.00102) | 0.02001 |
| Nadam | 0.008 | 67.01 (4.60) | 0.02052 (0.00099) | 0.01941 |

Wyniki Adam i Nadam są dużo bardziej zbliżone niż w przypadku zbioru dwuletniego.

Najlepsze wyniki dla Nadam:

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,019418 | 1,06E-05 | 1,26E-05 | 0,374894 | 0,133805 | 2,93E-05 | 4,71E-05 | 0,714149 |

# 5 Wyniki łączne

| Zbiór | Hiperparametry | Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|---|---|
| Naiwny | Cofnięcie 1 krok (godzina) | - | - | - | - | 0.03874 | 8.4918E-06 | 4.7079E-06 | 0.1961 |
| 09, 10, 11 – historia zapotrzebowania | [50, 20], Tanh, Nadam, 0.008 | 0,037182 | 2,03E-05 | 2,37E-06 | 0,465217 | 0,040486 | 8,87E-06 | 5,15E-07 | 0,503959 |
| 10, 11 – historia zapotrzebowania | [64, 25], Tanh, Adam, 0.005 | 0,039696 | 3,14E-05 | 1,65E-05 | 0,426179 | 0,041778 | 9,15E-06 | 7,53E-06 | 0,531195 |
| 09, 10, 11 – wszystkie | [65, 30], Tanh Nadam, 0.01 | 0,014667 | 7,99E-06 | 1,07E-06 | 0,377291 | 0,047836 | 1,05E-05 | 5,55E-06 | 0,557811 |
| 10,11 – wszystkie | [68, 25], Tanh, Nadam, 0.0065 | 0,018495 | 1,46E-05 | 1,53E-07 | 0,345688 | 0,058697 | 1,29E-05 | 4,94E-05 | 0,697104 |
| Naiwny | Cofnięcie 24 kroki (dzień) | - | - | - | - | 0.07705 | 1.6913E-05 | 1.4939E-05 | 0.4751 |
| 09, 10, 11 – korelacja > 0.1 | [55,25], Tanh Nadam, 0.008 | 0,019418 | 1,06E-05 | 1,26E-05 | 0,374894 | 0,133805 | 2,93E-05 | 4,71E-05 | 0,714149 |
| 10, 11 – korelacja > 0.1 | [75, 17], Tanh Adam, 0.008 | 0,019835 | 1,57E-05 | 1,89E-06 | 0,393794 | 0,207848 | 4,55E-05 | 5,31E-06 | 0,899838 |

Tabela z wynikami najlepszych modeli dla każdej kombinacji zbiór danych - cechy wejściowe. Posortowany według Mae Test.

* Hiperparametry zostały podane w formie: [warstwy], funkcja aktywacji warstw ukrytych, optymalizator, krok uczenia.

Niestety okazuje się, że **model naiwny z cofnięciem o godzinę jest lepszy od uzyskanych sieci**. Model posiada jednak tylko dane cofnięte o co najmniej dzień, co jest logiczne, ponieważ prognozy na tylko godzinę wstecz były by raczej mało użyteczne. Bardziej adekwatny jest model naiwny z cofnięciem o 24 godziny. Ten model wypada znacznie gorzej i tylko dwa utworzone modele wypadły gorzej od niego.

Według wartości błędu Mae na zbiorze testowym najlepszy jest zbiór „09,10,11-historia zapotrzebowania". Jest to jednak prawdopodobnie **dzieło przypadku**, ponieważ jego błąd na zbiorze walidacyjnym jest znacznie większy niż inne zestawy kolumn. Podobne zjawisko ma jednak miejsce dla zbioru z dwóch lat, dlatego ciężko ocenić czy jest to zasadny wybór.

Spośród pozostałych zbiorów model jest **najskuteczniejszy dla pełnego zestawu danych**. Nic dziwnego biorąc pod uwagę fakt, że sieci głębokie potrzebują dużych ilości informacji oraz

występujące osobiste problemy techniczne (więcej informacji w dalszych sekcjach). Należy wspomnieć, że w zbiorze na bazie podstawowej analizy korelacji nie udało się ustalić cech, które by wręcz przeszkadzały w pracy, więc w najgorszym przypadku były bezużyteczne. Jak wiadomo, w takiej sytuacji jedyną korzyścią ograniczania zbioru cech jest **przyśpieszenie nauki**, co było zauważalne również w tym badaniu, jednak korzyści czasowe w porównaniu do strat w jakości były na tyle małe, że zostały uznane za nieopłacalne.

Ciekawym zjawiskiem jest to, że model naiwny posiada mniejszy błąd maksymalny od modeli. Może to mieć znaczenie w zależności od kontekstu biznesowego. Być może rozwiązywany problem nie wymaga tego, by prognozy były średnio najbliższe prawdzie, lecz tego by nie mogły się pomylić o jakąś wartość maksymalną. Wtedy okazało by się, że to jednak model naiwny jest najlepszy.

# 6 Bardziej rozbudowany model

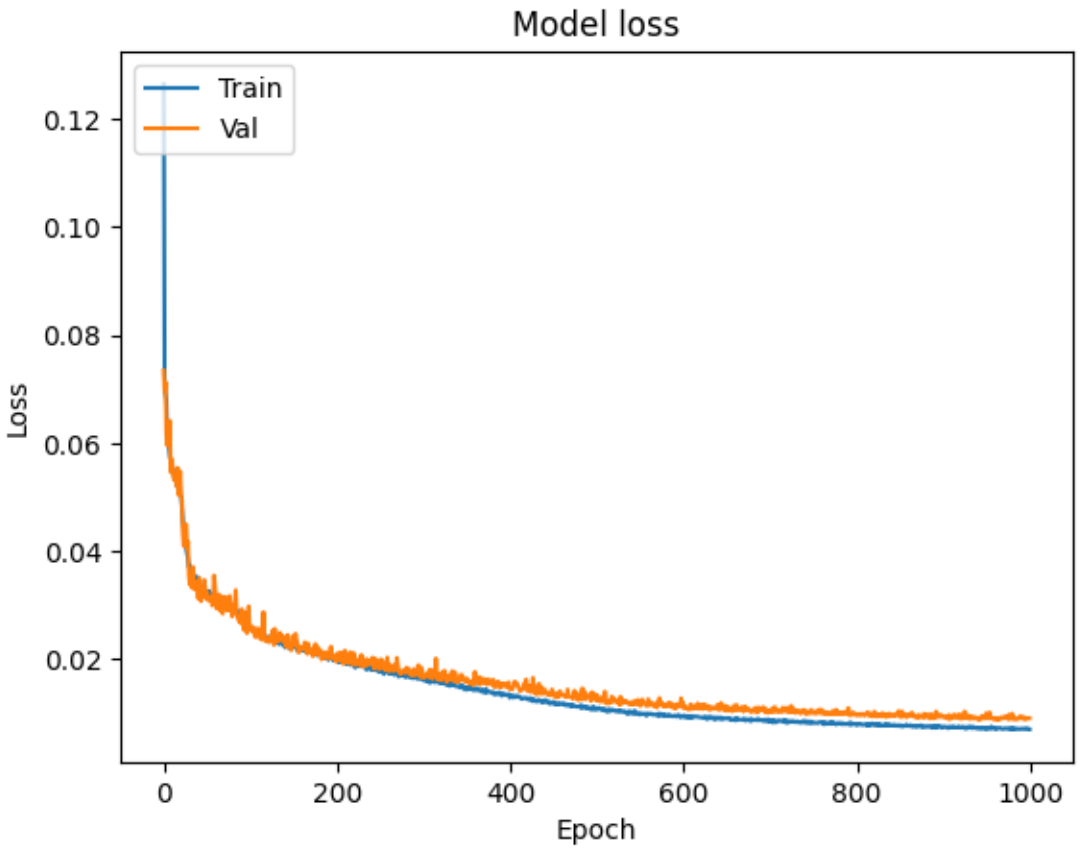
## 6.1 Zestaw trzyletni

Żeby sprawdzić zachowanie modelu przeprowadzony został eksperyment polegający na wykonaniu większej liczby iteracji. Do testu wybrany został model dla zbioru „09,10,11 – wszystkie". Aby uniknąć problemów z oscylacjami, współczynnik uczenia został zmniejszony do 0.005 zamiast 0.01.

Ostatecznie po wczesnym zatrzymaniu po 1192 epokach:

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,008265 | 4,5E-06 | 2,32E-06 | 0,121135 | 0,056535 | 1,2389E-05 | 1,78E-06 | 0,43428 |

Porównując uzyskane wyniki do poprzednich uzyskanych dla tych samych ustawień można zaobserwować prawie dwukrotny spadek wartości MAE Val oraz 20%+ wzrost MAE Test. Oznacza to, że poprawa jakości modelu dla danych walidacyjnych pogarsza go dla danych testowych. Jednocześnie nie są zauważalne problemy z przeuczeniem na wykresie straty dla zbiorów treningowego i walidacyjnego. Podsumowując powstaje podejrzenie, że **zestaw treningowy nie reprezentuje dobrze procesu odpowiadającego za wyniki testowe**.




Wykres nauki dla 1000 epok

Wykres przedstawia tylko pierwsze 1000 z 1192 epok. Zauważalne są pewne oscylacje, jednak oba błędy są konsekwentnie zmniejszane w trakcie nauki. Ten fakt w połączeniu z rosnącym błędem testowym to kolejny dowód na słabą reprezentatywność danych uczących.

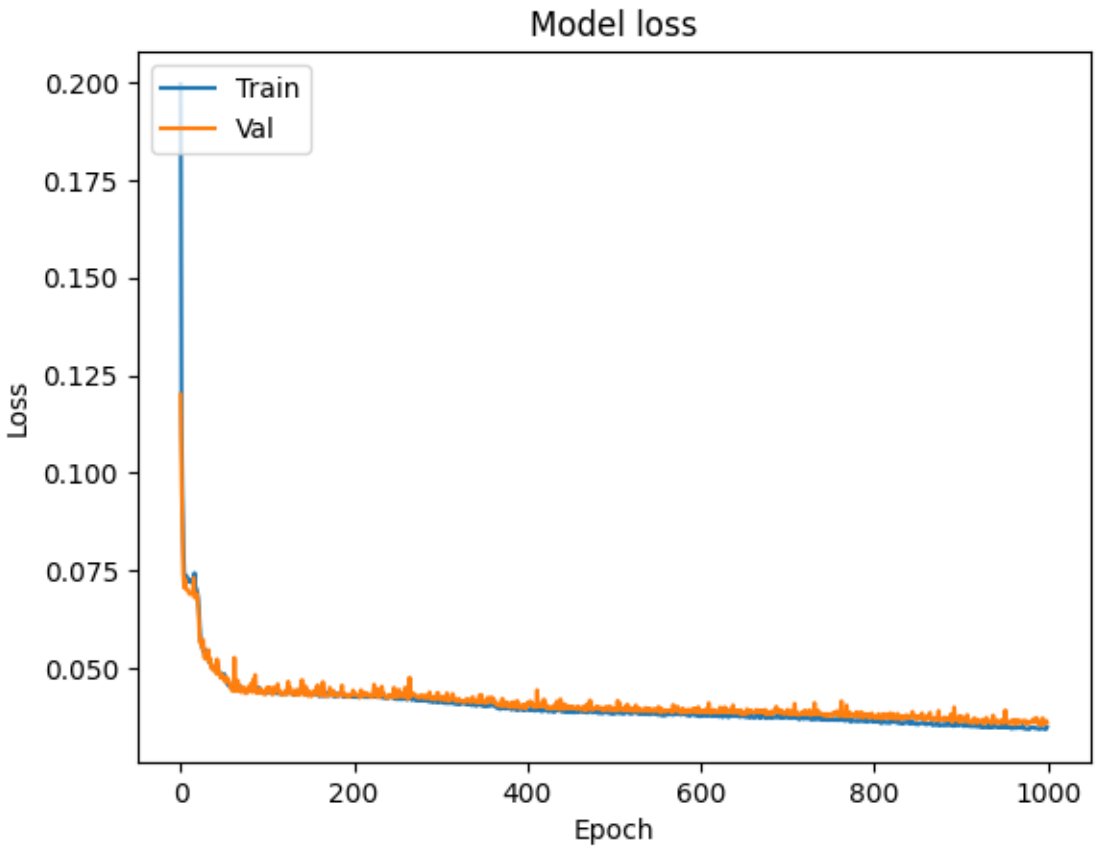
## 6.2 Zestaw dwuletni

Aby sprawdzić, czy problemem istotnie jest zestaw danych, zbadany został bardziej ograniczony zbiór dwuletni. Tym razem cechami wejściowymi były tylko dane historyczne o zapotrzebowaniu, ponieważ najlepiej radziły sobie na zbiorze testowym. Wynik po 1000 epok:

| Mae Val | Nmae Val | AeMin Val | AeMax Val | Mae Test | NMae Test | AeMin Test | AeMax Test |
|---|---|---|---|---|---|---|---|
| 0,035989 | 2,85E-05 | 9,81E-07 | 0,405615 | 0,040032 | 8,77E-06 | 7,66E-06 | 0,637183 |

W tym przypadku odnosząc się do wartości dla tych samych hiperparametrów z mniejszą liczbą epok błąd MAE Val poprawił się o zaledwie mniej niż 10%, natomiast błąd MAE Test zwiększył się o ponad 2%.

Podobny charakter wyników, czyli zmniejszenie błędu na zbiorze walidacyjnym skojarzone ze zwiększeniem błędu na zbiorze testowym jest kolejnym dowodem na słabą reprezentatywność danych uczących.

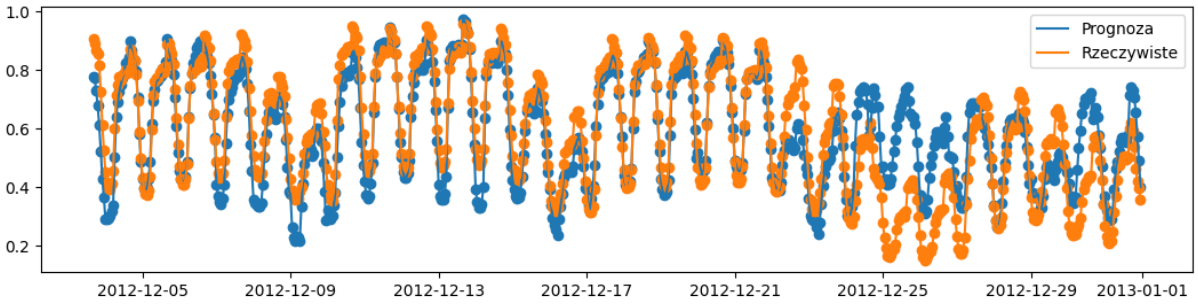



Wykres uczenia dla 1000 epok, zbiór dwuletni

Podobnie jak dla poprzedniego przypadku, dla zbioru dwuletniego zarówno błąd na zbiorze walidacyjnym, jak i treningowym mimo pewnych oscylacji stale maleją.

W tym przypadku zmiany błędów były mniejsze, więc istnieje możliwość występowania zjawiska, że im bardziej odległe dane tym słabiej reprezentują proces.

## 6.3 Interpretacja prognoz




Wykres prognoz dla modelu wykonanego na zbiorze „09,10,11 – wszystkie” oraz ponad 1000 epok.

Od razu zauważalne jest, że na początku wycinka model radzi sobie bardzo dobrze. Nie jest idealny, ale sprawnie dopasowuje się zarówno do dni tygodnia, jak i weekendów gdzie zapotrzebowanie jest mniejsze. W drugiej części (dla próbek po 2021-12-21) następuje jednak nietypowe zjawisko – model bardzo wyraźnie zaniża prognozę dla weekendu oraz zawyża dla pierwszych dni tygodnia, a następnie wraca do normy. Oczywiście w tym przykładzie powodem nietypowej sytuacji są święta. Zapotrzebowanie w takie dni jest bardzo nieprzewidywalne w porównaniu do całego roku i prawdopodobnym powodem skierowania modelu na złe tory są bardzo **nietypowe wartości**

występujące w tych datach w poprzednich latach. Ponadto, model może interpretować je jako zwykłe dni tygodnia, co skutkuje minięciem się z prawdą.

Wracając do lewej części wykresu można zaobserwować, że prognozy są przesunięte w dół prawie dla każdego dnia. Istnieje więc szansa, że na skutek procesów społecznych zapotrzebowanie w mniej odległych latach wzrosło. Dane treningowe pochodzące z przeszłości nie są w stanie tego pokazać, dlatego **powstaje zjawisko pewnego braku reprezentowalności**.

# 7 Pozostałe wnioski

## 7.1 Reprezentatywność danych i wyniki

Udało się utworzyć modele, które były zdecydowanie lepsze od podejścia naiwnego, więc cel został wykonany. Niestety, jak się okazuje dane zarówno w zbiorze dwuletnim, jak i trzyletnim nie są do końca reprezentatywne dla procesu zawartego w zbiorze testowym. Potencjalnym powodem może być zmiana cen energii lub inne procesy społeczno-ekonomiczne.

Naturalnym dalszym krokiem do realizacji tego projektu była by jeszcze dokładniejsza analiza (najlepiej z udziałem eksperta) danych w celu identyfikacji zmian w procesie lub szczególnie wadliwych części zbioru. Przeprowadzone sprawdzenie zachowania modelu było tylko pierwszym krokiem.

Być może skutecznym rozwiązaniem byłoby wstępne dopasowanie modelu do najświeższych danych treningowych, a następnie zasilenie go wycinkiem danych z ostatniego roku (testowego). W ten sposób pierwszy etap utworzył by model wstępny, który drugi etap dopasowałby do charakteru nowoczesnych danych zakładając ich bardzo ograniczoną ilość.

## 7.2 Sieć LSTM

Mimo problemów udało się jednak wykonać wystarczające badania, by zaobserwować pewne zależności dotyczące samego typu sieci:

- Sieci LSTM potrzebują normalizacji wejść o różnorodnych zakresach wartości
- Normalizacja wyjścia jest opcjonalna, ale ma wpływ na szybkość działania i nauki sieci, dlatego jest zalecana
- Sieć LSTM jest obciążającym obliczeniowo zadaniem – bardziej czasochłonnym i potrzebującym więcej pamięci niż klasyczne sieci neuronowe
- LSTM dobrze generalizuje już po niewielkiej liczbie epok, ale w celu osiągnięcia najlepszych rezultatów może potrzebować bardzo długiego czasu
- Optymalizator Adam i Nadam są bardzo podobne. Zazwyczaj Nadam działał trochę szybciej i miał mniejsze oscylacje czasu treningu
- Optymalizator AdaGrad w porównaniu do metod Adam i Nadam ma znacznie wolniejszą zbieżność, ale dzięki temu występuje mniej oscylacji, nawet przy wyższych ustawieniach kroku uczenia
- Sieci LSTM wymagają dużych ilości danych
- Liczba cech użytych do nauki ma wpływ na szybkość i jakość modelu
- LSTM jest odporna na potencjalnie bezużyteczne zmienne wejściowe
- Głównymi hiperparametrami sieci do sterowania złożonością LSTM są liczba i rozmiary warstw, pozostałe hiperparametry służą bardziej do optymalizacji/ogólnego dopasowania do zadania

## 7.3 Trudności w realizacji

Podczas pracy nad tym projektem wielokrotnie napotykane były problemy związane z trudnością obliczeniową zadania. Uczenie sieci trwające ponad 80 sekund nawet dla zaledwie 250 iteracji i batch_size 512, błędy pamięci (czasami na tyle poważne, że powodowały bluescreen), czy umierające jądro python. Wszystkie te problemy sprawiają, że wykonanie celu modelu, czyli utworzenie najwyższej jakości prognoz było bardzo trudne. Ponadto, ze względu na czasochłonny charakter zadania realizacja w trakcie ćwiczeń stacjonarnych przeznaczonych na te działania była niewykonalna (w trakcie zajęć udało się wykonać pracę od wczytania i preprocessingu do pełnej analizy pierwszego zestawu danych i kolumn włącznie).