

Ćwiczenie RNN

Rekurencyjna sieć neuronowa

Część teoretyczna

Wykład na temat rekurencyjnych sieci neuronowych (RNN).

Zadania pomocnicze

Zapoznaj się z tematem predykcji szeregów czasowych w środowisku TensorFlow

https://www.tensorflow.org/tutorials/structured_data/time_series?hl=pl [*]

Zapoznaj się z danymi – dane opisano w <https://www.manning.com/books/deep-learning-with-python>, rozdz. 6.3.1, a także w <https://www.bgc-jena.mpg.de/wetter/>

Zadania do wykonania

Wzorując się na przykładzie podanym na wykładzie oraz pod ww. linkiem [*], zbuduj model predykcyjny oparty na sieci RNN do prognozowania temperatury atmosferycznej w horyzoncie równym 1 (na kolejną godzinę).

1. Zaimportuj niezbędne moduły.
2. Wczytaj dane

```
zip_path = tf.keras.utils.get_file(
    origin='https://storage.googleapis.com/tensorflow/tf-keras-
datasets/jena_climate_2009_2016.csv.zip',
    fname='jena_climate_2009_2016.csv.zip',
    extract=True)
csv_path, _ = os.path.splitext(zip_path)
df = pd.read_csv(csv_path)

# Slice [start:stop:step], starting from index 5 take every 6th record.
df = df[5::6]

date_time = pd.to_datetime(df.pop('Date Time'), format='%d.%m.%Y %H:%M:%S')

df.head()
```

3. Podaj podstawowe statystyki danych (`df.describe().transpose()`). Wyznacz korelacje pomiędzy szeregami czasowymi. Zwizualizuj powiązania pomiędzy szeregami (`sns.pairplot`). Zwizualizuj wszystkie szeregi czasowe w całym zakresie czasowym (2009-2017) oraz w zakresie obejmującym jeden dowolny miesiąc (patrz [*]).
4. Przygotuj dane – wytnij z oryginalnych danych 2 lata (zależnie od `nr_gr`). Przyjmij, że zbiór testowy stanowi ostatni tydzień tego okresu, a zbiór treningowy pozostałą część. Ustal rozmiar okna wejściowego (`look_back`) na 24 godz.

```
nr_gr = ?
start = int(nr_gr*300*24)
```

```

end = int(start + 730*24)
df = df[start:end]

look_back = 24
test_size = 7*24
train_size = (end-start) - test_size
train = df[:train_size]
test = df[train_size:]

```

5. Dokonaj standaryzacji każdego szeregu w części treningowej i testowej (`StandardScaler()`). Do dalszych analiz wybierz szereg czasowy temperatury atmosferycznej (`train1 = train[:,1].reshape(-1,1)`, `test1 = test[:,1].reshape(-1,1)`). Model prognostyczny będzie się uczył prognozować temperaturę na chwilę $t+1$ na podstawie temperatur w chwilach $t-24, \dots, t$ (bez wykorzystania pozostałych zmiennych). Zdefiniuj zbiory treningowy i testowy z wykorzystaniem funkcji `create_dataset()` (patrz wykład).
6. Zbuduj model sieci RNN z jedną warstwą rekurencyjną LSTM. Skompiluj model. Przeprowadź uczenie modelu. Eksperymentalnie dobierz podstawowe hiperparametry: liczbę neuronów w LSTM, liczbę epok i rozmiar *batcha* (możesz przyjąć domyślnie wartości: 16, 30, 500). Wyznacz błędy RMSE, MAE i ME (pamiętaj o destandaryzacji wyników). Pokaż krzywą zbieżności (*loss curve*). Pokaż wykresy oryginalnego szeregu i jego prognoz (część treningowa i testowa) w całym dwuletnim zakresie czasowym oraz w zakresie obejmującym ostatnie dwa tygodnie, tj. ostatni tydzień treningowy i tydzień testowy (na oddzielnym rysunku).
7. Czy dodanie dodatkowych zmiennych wejściowych (egzogenicznych) poprawi wyniki? Sprawdź to eksperymentalnie, powtarzając p. 6.

Wskazówka:

- rozszerz dane wejściowe o dodatkowe atrybuty, np. ciśnienie atmosferyczne p (mbar) i gęstość powietrza ρ (g/m^3) (możesz użyć innych zmiennych). W tym celu użyj funkcji `create_dataset()` dla tych nowych zmiennych, podobnie jak wcześniej do danych temperaturowych. Otrzymane w wyniku zbiory `trainX0` i `testX0` oraz `trainX10` i `testX10` połącz ze zbiorami otrzymanymi wcześniej dla temperatury (`trainX1` i `testX1`):

```

trainXa = np.concatenate((trainX1,trainX0,trainX10),axis=1)
testXa = np.concatenate((testX1,testX0,testX10),axis=1)

```

Nowa struktura danych wejściowych (`trainXa`, `testXa`) wygląda tak:

T (degC)			p (mbar)			rho (g/m**3)		
-0.18	...	0.35	0.11	...	-0.02	-0.19	...	0.52
-0.32	...	0.87	0.21	...	-0.31	-0.28	...	0.27
...
24			24			24		
72								

Co powinno znaleźć się w sprawozdaniu

- A) Cel ćwiczenia.
- B) Treść zadania.
- C) Opis używanej w ćwiczeniu sieci neuronowej (nie kopiuj treści wykładu, poszukaj w literaturze i Internecie).

- D) Metodyka rozwiązania – poszczególne instrukcje z wynikami i komentarzem (zachowaj numerację zadań).
- E) Wnioski końcowe.

Zadania dodatkowe dla ambitnych

1. Rozwiąż postawiony problem za pomocą sieci MLP i modelu opartego na drzewach decyzyjnych. Porównaj wyniki z wynikami RNN.
2. Utwórz model sieci RNN do prognoz o dłuższym horyzoncie, np. 24 godz. Możliwe są tu dwa warianty: (i) prognozowanie rekurencyjne – prognoza dla chwili $t+1$ staje się wejściem modelu dla prognozy dla chwili $t+2$ itd. lub (ii) prognozowanie bezpośrednie – model prognozuje wektor obejmujący prognozy dla $t+1, \dots, t+24$.
3. Wykonaj to ćwiczenie w innym środowisku, np. R, C#, Matlab, ...

Przykładowe zagadnienia i pytania zaliczeniowe

1. Cel i plan ćwiczenia.
2. Materiał ze sprawozdania.
3. Modele prostych sieci rekurencyjnych.
4. Modele z bramkami: LSTM, GRU.
5. Wsteczna propagacja błędów w czasie.
6. Modele hybrydowe – ogólna zasada działania.
7. Modele ze ścieżką kontekstową – ogólna zasada działania.

Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.
2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.