UNIVERZITET U ZENICI
POLITEHNIČKI FAKULTET

Predikcija vremenskih uslova korištenjem LSTM neuronskih mreža

PROJEKAT

Predmet: *Rudarenje podataka*
Profesor: *doc. dr. Adnan Dželihodžić*
Asistenti: mr. *Faris Hambo, mr. Ahmed Mujić*
Studenti: *Mahir Prcanović II-111, Mirzad Varupa II-115*
Studij: *Softversko inženjerstvo, II ciklus studija*

Zenica, *juni* 2025.

SADRŽAJ

1. Opis projekta

Ovaj projekt ima za cilj razvoj prediktivnog modela koji koristi vremenske serije za predviđanje vremenskih uslova, konkretno temperature. Model je baziran na rekurentnim neuronskim mrežama, tačnije LSTM (Long Short-Term Memory) arhitekturi, koja je posebno pogodna za obradu sekvencijalnih podataka poput vremenskih serija. Cilj modela je da, na osnovu prethodnih N dana meteoroloških podataka, predvidi temperaturu za naredni dan. Kroz proces pripreme podataka, treninga modela i evaluacije rezultata, testirane su različite konfiguracije modela, kao i metode za poboljšanje tačnosti predikcije. Eksperimentisalo se s dužinom vremenskog prozora (window size), dubinom LSTM slojeva, te tehnikama poput dropout regularizacije radi onemogućavanja overfittinga.

U krajnjem cilju, ovaj projekt prikazuje kako se moderni alati dubokog učenja mogu primjeniti na realan vremenski dataset kako bi se dobile korisne i praktične vremenske prognoze. Iako je model razvijen kao akademski eksperiment, rezultati pokazuju da ovakav pristup ima potencijal za širu primjenu.

2. Opis dataseta

Dataset koji se koristio za analizu i predikciju vremenskih uslova sadrži satne meteorološke podatke za nekoliko velikih gradova širom svijeta. Podaci su prikupljeni i javno dostupni putem platforme *Kaggle* pod nazivom *Historical Hourly Weather Data*. Glavni cilj ovog dataseta je da omogući detaljnu vremensku analizu i modeliranje na bazi vremenskih serija. Dataset se sastoji od više CSV fajlova, pri čemu svaki fajl sadrži različite vremenske varijable:

- temperature.csv – satne temperature izražene u Kelvinima za 35 gradova, uključujući New York, Los Angeles, San Francisco, Toronto i druge.
- humidity.csv – relativna vlažnost vazduha izražena u procentima.
- pressure.csv – atmosferski pritisak u hektopaskalima (hPa).
- wind_speed.csv – brzina vjetra izražena u metrima u sekundi (m/s).
- wind_direction.csv – smjer vjetra izražen u stepenima.
- weather_description.csv – tekstualni opisi vremenskih uslova poput „broken clouds“, „light rain“, „sky is clear“ i slično.
- city_attributes.csv – atributi gradova kao što su zemlja, geografska širina i dužina

Svi vremenski podaci su registrovani u formatu vremenskih serija sa vremenskim pečatom (datetime), i predstavljaju kontinuirana mjerenja u satnim intervalima. Važni aspekti dataseta su svakako:

- Jedinstvenost podataka: Svaki grad ima svoje kolone u fajlovima, sa vrijednostima koje su vezane za taj vremenski pečat.
- Nedostajući podaci: Dataset sadrži praznine (NaN vrijednosti), koje su popunjene tehnikama popunjavanja kao što su *forward fill i backward fill*, kako bi se osigurao kontinuitet vremenskih serija.
- Temperatura u Kelvinima: Sve temperaturne vrijednosti su prvobitno izražene u Kelvinima i u ovom radu su konvertovane u Celzijuse radi lakše interpretacije.
- Kategorijalni podaci (opis vremena): Vremenski opisi su tekstualni podaci koji su kodirani u binarne kategorije korištenjem one-hot encoding kako bi se mogli koristiti u mašinskom učenju.

3. Opis problema

U ovom projektu cilj je predvidjeti vremenske uslove na osnovu vremenskih serija podataka, sa fokusom na ključnu promjenjivu, temperaturu. Preciznije, cilj je izgraditi model koji može da predvidi temperaturu za naredni vremenski korak na osnovu prethodnih vremenskih podataka. Glavni izazovi su:

- Sekvencijalnost podataka: vremenski podaci su vremenski zavisni, trenutna temperatura zavisi od prethodnih vremenskih podataka. Model mora razumjeti ove vremenske zavisnosti.
- Preciznost predikcije: predviđanje temperature je regresioni zadatak, gdje je cilj minimizirati greške i predvidjeti što tačniju vrijednost.

4. Opis algoritma

Za rješavanje problema predikcije temperature iskorišten je Long Short-Term Memory (LSTM) neuralni mrežni model, koji pripada klasi rekurentnih neuronskih mreža (RNN). Ova arhitektura je posebno pogodna za rad sa sekvencijalnim podacima jer može da „pamti“ informacije iz prošlosti kroz duže vremenske periode.

4.1. Arhitektura modela

Koristio se sekvencijalni model sljedeće strukture:

```
model = Sequential([
    LSTM(128, return_sequences=True, input_shape=(WINDOW_SIZE, X.shape[2])),
    Dropout(0.3),
    LSTM(64),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dense(1)
])
```

Slika 1. LSTM model iskorišten u projektu

LSTM slojevi (128 i 64 jedinice) omogućavaju modelu da uči dugoročne i kratkoročne zavisnosti, droupout slojevi služe kao regularizacija, smanjuju prenaučenost modela, dense sloj omogućava dodatnu nelinearnu transformaciju prije izlaza i izlaz je na kraju jedna vrijednost tj. predikcija temperature za sljedeći vremenski korak.

Model je treniran sa adam optimizatorom, prikazan u kodu na slici 2.

```
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=20, batch_size=64, validation_split=0.1)
```

Slika 2. Treniranje modela

Na slici 2. se nalazi način na koji je model treniran, koristeći Adam optimizator, koji je efikasan algoritam koji kombinuje prednosti drugih algoritama. MSE (mean squared error) je iskorištena kao metrika greške jer se radi o regresionom zadatku. Obuka se vrši funkcijom .fit().

5. Metode iskorištene za poboljšanje algoritma

Kroz razvoj modela za predikciju temperature, primjenjeno je više tehnika za poboljšanje tačnosti, stabilnosti i generalizacije modela. Iako model nije kompleksan, sljedeće metode su bile ključne za optimizaciju performansi:

1. Korištenje vremenskog prozora (sliding window)

Vrijednosti meteoroloških parametara su vremenske serije, što znači da trenutna temperatura zavisi od njenih prethodnih vrijednosti, kao i od drugih faktora u nedavnoj prošlosti. Da bi model imao pristup ovoj vremenskoj dinamici, koristi se tzv. klizni vremenski prozor (WINDOW_SIZE), koji omogućava modelu da u svakom trenutku gleda, npr. posljednjih 24 sata meteoroloških podataka prije nego što pokuša predvidjeti narednu temperaturu.

```
def create_sequences(data, target_col, window):
    X, y = [], []
    target_idx = data.columns.get_loc(target_col)
    for i in range(len(data) - window):
        X.append(data.iloc[i:i+window].values)
        y.append(data.iloc[i+window, target_idx])
    X = np.array(X, dtype=np.float32)
    y = np.array(y, dtype=np.float32)
    return X, y
```

Slikla 3. Kreiranje vremenskih sekvenci

Kreirana je funkcija create_sequences koja prolazi kroz podatke i na određeni vremenski period (u našem slučaju 24) kreira sekvence koje se nalaze u tom prozoru zajedno sa vrijednosti iduće (25te) kao rezultat predikcije.

2. Normalizacija podataka (MinMax skaliranje)

Vrijednosti različitih meteoroloških parametara imaju različite skale (npr. temperatura u Kelvinima, pritisak u hPa, brzina vjetra u m/s). Da bi se spriječilo da karakteristike sa većim numeričkim vrijednostima dominiraju učenja modela, koristi se MinMax skaliranje. Ova metoda sve vrijednosti skalira u opseg [0,1], čime se postiže ravnoteža u doprinosu svakog atributa.

```
numerical_cols = [col for col in data.columns if not col.startswith("weather_description_")]
scaler = MinMaxScaler()
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])
```

Slika 4. Skaliranje podataka sa MinMaxScaler-om

Za skaliranje su iskorištene sve kolone osim „weather_description_“ jer su one tekstualne, dok su sve ostale brojevne kolone.

3. One-hot enkodiranje vremenskih opisa

Kolona weather_description sadrži tekstualne opise poput „clear sky“, „light rain“ itd., koje model ne može direktno koristiti. Da bi se to prevazišlo, koristi se One-hot encoding, koji svaku kategoriju pretvara u niz binarnih kolona.

```
data = pd.get_dummies(data, columns=["weather_description"])
```

Slika 5. One-hot encoding nad weather_description kolonom

4. Raspodjela skupa podataka (train/test split)

Model se trenira na 80% podataka, a testira na preostalih 20%. Time se obezbjeđuje da se model evaluira na podacima koje nikad nije vidio, čime se mjeri njegova stvarna sposobnost generalizacije (prikazano na slici 6).

```
split = int(0.8 * len(X))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]
```

Slika 6. Split podataka na test/train

5. Inverzna transformacija predikcija

Nakon što se dobije predikcija iz modela (u skaliranom obliku), ona se mora vratiti nazad u originalnu temperaturu u stepenima celzijusa. To se radi uz pomoć iste MinMaxScaler instance korištene pri treniranju.

```python
def invert_scale(pred_scaled, y_scaled, scaler, feature_name, numerical_cols):
    idx = numerical_cols.index(feature_name)
    pred_padded = np.zeros((len(pred_scaled), len(numerical_cols)))
    y_padded = np.zeros((len(y_scaled), len(numerical_cols)))
    pred_padded[:, idx] = pred_scaled.flatten()
    y_padded[:, idx] = y_scaled.flatten()
    inv_pred = scaler.inverse_transform(pred_padded)[:, idx]
    inv_y = scaler.inverse_transform(y_padded)[:, idx]
    return inv_pred, inv_y

pred_inv, y_test_inv = invert_scale(pred, y_test, scaler, target_column, numerical_cols)
```

Slika 7. Invertovanje skaliranih podataka nazad u prave vrijednosti

6. Rezultati i njihova interpretacija

Nakon treniranja LSTM modela za predikciju temperature koristeći kombinaciju meteoroloških faktora (vlažnost, pritisak, brzina i smjer vjetra, te vremenski opis), izvršena je evaluacija performansi modela na testnom skupu podataka.

- Evaluacione metrike
    - MAE (Mean Absolute Error) – prosječna apsolutna razlika između stvarnih i predviđenih temperatura.
    - RMSE (Root Mean Squared Error) – korijen srednje kvadratne greške koji je osjetljiviji na veće greške.

Ove vrijednosti znače da model u prosjeku griješi oko 1-2 stepena (u oba smjera), a veće greške su u prosjeku unutar 2-4 stepeni. Ove vrijednosti su prilično dobre kada se uzme u obzir da su ulazni podaci mjereni svakih sat vremena i da se temperatura može brzo promijeniti zbog vremenskih prilika.

- Vizuelna interpretacija

Za dodatnu evaluaciju, modelove predikcije su vizualno upoređene sa stvarnim temperaturama na testnom skupu:

```python
# Prikaz rezultata
plt.figure(figsize=(14, 5))
plt.plot(y_test_inv[:500], label='Prava temperatura')
plt.plot(pred_inv[:500], label='Predviđena temperatura')
plt.title("Predikcija temperature")
plt.legend()
plt.show()
```

Slika 8. Vizuelna interpretacija rezultata

7. Zaključak

U ovom projektu razvijen je model za predikciju temperature koristeći vremenske serije meteoroloških podataka i rekurentne neuronske mreže (RNN), konkretno Long Short-Term Memory (LSTM) arhitekturu. Model koristi prethodne vremenske podatke kroz vremenski prozor (WINDOW_SIZE = 24) kako bi predvidio temperaturu za naredni sat. Korištenjem kombinacije različitih faktora kao što su vlažnost, pritisak, brzina i smjer vjetra, kao i opis vremenskih uslova (uzet kroz one-hot enkodiranje), omogućeno je učenje kompleksnih obrazaca u ponašanju temperature tokom vremena. Podaci su pažljivo očišćeni, skalirani i sekvencionirani prije treniranja modela, čime je osigurano da se svi ulazi nalaze na istoj skali i da model može efikasno učiti odnose među njima. Evaluacija modela pokazuje da je sposoban da s relativno visokom tačnošću predvidi temperaturu – sa prosječnom greškom oko 2 °C. Vizualni prikaz rezultata potvrđuje da model vrlo dobro prati promjene u temperaturi, uključujući i lokalne varijacije. Iako je model već pokazao zadovoljavajuće performanse, postoji prostor za buduća poboljšanja, kao što je uključivanje dodatnih varijabli, eksperimentisanje s različitim arhitekturama neuronskih mreža, kao i korištenje naprednijih metoda obrade vremenskih serija. Ovaj projekat demonstrira praktičnu primjenu dubokog učenja u obradi vremenskih podataka i može poslužiti kao osnova za složenije sisteme predikcije u meteorologiji i srodnim oblastima.