### 案例 6：基于 GRU-Attention 模型的流域长期降水量预测

* **问题背景**：降水量是影响农业生产、水资源管理和防洪抗旱的重要因素。某流域的降水量具有明显的季节性和年际变化特征，且受厄尔尼诺、拉尼娜等气候现象影响，长期降水量预测难度较大，需要关注关键时间节点的气候特征。
* **问题描述**：某流域管理局需要对该流域未来 5 年的年度降水量进行预测。要求模型能够高效处理长序列的历史降水数据，重点关注影响降水量的关键季节转换节点（如厄尔尼诺现象发生的月份），提高长期预测的准确性，为流域水资源规划和防灾减灾提供依据。
* **数据情况**：提供该流域过去 50 年的月度降水量数据（单位：mm），同时提供全球气候指数（如厄尔尼诺指数、南方涛动指数等）、大气环流数据等。数据存在部分年份的个别月份数据缺失。

### 案例 6：GRU-Attention 模型流域长期降水量预测代码

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.preprocessing import MinMaxScaler  from tensorflow.keras.models import Model  from tensorflow.keras.layers import Input, GRU, Dense, Attention, Concatenate  from sklearn.metrics import mean\_squared\_error  import tensorflow as tf  # 数据加载与预处理  data = pd.read\_csv('basin\_precipitation.csv', parse\_dates=['date'], index\_col='date')  precip\_data = data['precipitation'].values.reshape(-1, 1)  # 数据归一化  scaler = MinMaxScaler(feature\_range=(0, 1))  precip\_scaled = scaler.fit\_transform(precip\_data)  # 构建序列数据  def create\_sequences(data, time\_steps=12, pred\_steps=1):  X, y = [], []  for i in range(len(data) - time\_steps - pred\_steps + 1):  X.append(data[i:i+time\_steps, 0])  y.append(data[i+time\_steps:i+time\_steps+pred\_steps, 0])  return np.array(X), np.array(y).squeeze()  time\_steps = 12 # 12个月的历史数据  X, y = create\_sequences(precip\_scaled, time\_steps)  # 划分训练集和测试集  train\_size = int(len(X) \* 0.8)  X\_train, X\_test = X[:train\_size], X[train\_size:]  y\_train, y\_test = y[:train\_size], y[train\_size:]  # 重塑输入形状 [samples, time steps, features]  X\_train = X\_train.reshape(X\_train.shape[0], X\_train.shape[1], 1)  X\_test = X\_test.reshape(X\_test.shape[0], X\_test.shape[1], 1)  # 构建GRU-Attention模型  inputs = Input(shape=(time\_steps, 1))  gru\_out = GRU(64, return\_sequences=True)(inputs) # 返回所有时间步的输出用于注意力机制  # 注意力机制层  attention = Attention()([gru\_out, gru\_out])  attention = tf.reduce\_sum(attention, axis=1) # 聚合注意力权重  # 输出层  outputs = Dense(1)(attention)  model = Model(inputs=inputs, outputs=outputs)  model.compile(optimizer='adam', loss='mse')  # 模型训练  history = model.fit(X\_train, y\_train, epochs=50, batch\_size=16, validation\_data=(X\_test, y\_test), verbose=1)  # 模型预测  y\_pred = model.predict(X\_test)  # 反归一化  y\_pred\_actual = scaler.inverse\_transform(y\_pred.reshape(-1, 1))  y\_test\_actual = scaler.inverse\_transform(y\_test.reshape(-1, 1))  # 评估模型  mse = mean\_squared\_error(y\_test\_actual, y\_pred\_actual)  print(f'GRU-Attention模型MSE: {mse}')  # 可视化结果  plt.figure(figsize=(12, 6))  plt.plot(y\_test\_actual, label='实际降水量')  plt.plot(y\_pred\_actual, label='预测降水量')  plt.legend()  plt.savefig('precipitation\_prediction.png')  plt.show()  # 保存模型  model.save('gru\_attention\_precip.h5') |