### 案例 11：基于 FNN-GA 模型的湖泊水质污染指数预测

* **问题背景**：湖泊水质污染问题日益突出，影响生态环境和人类健康。水质污染指数（如 COD、氨氮、总磷等）的评价标准存在一定模糊性（如 “轻度污染”“中度污染” 的界定），且受工业废水排放、农业面源污染、降雨量、水体自净能力等多因素影响，非线性关系显著。
* **问题描述**：某环境监测部门需要对某重点湖泊未来 1 个月的每周水质污染指数进行预测。要求模型能够处理评价标准中的模糊信息，准确预测污染指数变化趋势，为污染防治措施的制定提供科学依据。
* **数据情况**：提供过去 5 年的每周水质监测数据（包括 COD、氨氮、总磷等污染指数），同时提供周边工业废水排放量、农业化肥使用量、降雨量、风速等数据。数据存在因监测设备故障导致的部分时段缺失，且部分污染指数的划分标准具有模糊性。

### 案例 11：FNN-GA 模型湖泊水质污染指数预测代码

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.preprocessing import MinMaxScaler  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Dense, Activation  from sklearn.metrics import mean\_squared\_error  from geneticalgorithm import geneticalgorithm as ga  # 数据加载与预处理  data = pd.read\_csv('lake\_water\_quality.csv', parse\_dates=['date'], index\_col='date')  pollution\_data = data['pollution\_index'].values.reshape(-1, 1)  # 数据归一化  scaler = MinMaxScaler(feature\_range=(0, 1))  pollution\_scaled = scaler.fit\_transform(pollution\_data)  # 构建数据集  def create\_dataset(data, look\_back=4):  X, y = [], []  for i in range(len(data) - look\_back):  X.append(data[i:i+look\_back, 0])  y.append(data[i+look\_back, 0])  return np.array(X), np.array(y)  look\_back = 4 # 用过去4周数据预测下一周  X, y = create\_dataset(pollution\_scaled, look\_back)  # 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:]  # 定义FNN模型结构（由GA优化参数）  def build\_fnn(hidden\_units, activation):  model = Sequential()  model.add(Dense(hidden\_units[0], input\_dim=look\_back))  model.add(Activation(activation))  for units in hidden\_units[1:]:  model.add(Dense(units))  model.add(Activation(activation))  model.add(Dense(1))  model.compile(loss='mse', optimizer='adam')  return model  # 定义GA优化目标函数  def objective\_function(params):  # 参数解析：隐藏层神经元数(2层)、激活函数类型(0:relu,1:sigmoid)、迭代次数  hidden1 = int(params[0])  hidden2 = int(params[1])  activation\_idx = int(params[2])  epochs = int(params[3])    activation = 'relu' if activation\_idx == 0 else 'sigmoid'  model = build\_fnn([hidden1, hidden2], activation)  model.fit(X\_train, y\_train, epochs=epochs, batch\_size=8, verbose=0)  y\_pred = model.predict(X\_test)  return mean\_squared\_error(y\_test, y\_pred)  # 参数范围  varbound = np.array([  [8, 64], # hidden1  [4, 32], # hidden2  [0, 1], # activation  [20, 100] # epochs  ])  # 遗传算法优化  algorithm\_param = {  'max\_num\_iteration': 30,  'population\_size': 15,  'mutation\_probability': 0.1,  'elit\_ratio': 0.1,  'crossover\_probability': 0.5,  'variable\_type': 'int'  }  model\_ga = ga(  function=objective\_function,  dimension=4,  variable\_type='int',  variable\_boundaries=varbound,  algorithm\_parameters=algorithm\_param  )  model\_ga.run()  best\_params = model\_ga.best\_variable  # 构建最优FNN模型  best\_hidden = [int(best\_params[0]), int(best\_params[1])]  best\_activation = 'relu' if best\_params[2] == 0 else 'sigmoid'  best\_epochs = int(best\_params[3])  final\_model = build\_fnn(best\_hidden, best\_activation)  final\_model.fit(X\_train, y\_train, epochs=best\_epochs, batch\_size=8, verbose=1)  # 模型预测  y\_pred = final\_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'FNN-GA模型MSE: {mse}')  # 可视化结果  plt.figure(figsize=(12, 6))  plt.plot(y\_test\_actual, label='实际污染指数')  plt.plot(y\_pred\_actual, label='预测污染指数')  plt.legend()  plt.savefig('water\_pollution\_prediction.png')  plt.show()  # 保存模型  final\_model.save('fnn\_ga\_water\_quality.h5')  joblib.dump(scaler, 'scaler\_water.pkl') |