water-quality-analysis-using-cnn

March 25, 2024

[1]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\joevin\anaconda3\lib\site-
packages (2.16.1)
Requirement already satisfied: tensorflow-intel==2.16.1 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow) (2.16.1)
Requirement already satisfied: absl-py>=1.0.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (24.3.7)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (3.10.0)
Requirement already satisfied: libclang>=13.0.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.3.1 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.3.2)
Requirement already satisfied: opt-einsum>=2.3.2 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (3.3.0)
Requirement already satisfied: packaging in c:\users\joevin\anaconda3\lib\site-
packages (from tensorflow-intel==2.16.1->tensorflow) (23.1)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
```

```
intel==2.16.1->tensorflow) (4.25.3)
Requirement already satisfied: requests<3,>=2.21.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.31.0)
Requirement already satisfied: setuptools in c:\users\joevin\anaconda3\lib\site-
packages (from tensorflow-intel==2.16.1->tensorflow) (68.0.0)
Requirement already satisfied: six>=1.12.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.62.1)
Requirement already satisfied: tensorboard<2.17,>=2.16 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.16.2)
Requirement already satisfied: keras>=3.0.0 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (3.1.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.31.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
c:\users\joevin\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.24.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
c:\users\joevin\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-
intel==2.16.1->tensorflow) (0.38.4)
Requirement already satisfied: rich in c:\users\joevin\anaconda3\lib\site-
packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (13.7.1)
Requirement already satisfied: namex in c:\users\joevin\anaconda3\lib\site-
packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.0.7)
Requirement already satisfied: optree in c:\users\joevin\anaconda3\lib\site-
packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\joevin\anaconda3\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\joevin\anaconda3\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (3.4)
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\joevin\anaconda3\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\joevin\anaconda3\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (2023.7.22)
Requirement already satisfied: markdown>=2.6.8 in
c:\users\joevin\anaconda3\lib\site-packages (from
tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
c:\users\joevin\anaconda3\lib\site-packages (from
tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
c:\users\joevin\anaconda3\lib\site-packages (from
tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
c:\users\joevin\anaconda3\lib\site-packages (from
werkzeug>=1.0.1->tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow)
(2.1.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in
c:\users\joevin\anaconda3\lib\site-packages (from
rich->keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
c:\users\joevin\anaconda3\lib\site-packages (from
rich->keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\joevin\anaconda3\lib\site-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.0.0->tensorflow-
intel==2.16.1->tensorflow) (0.1.0)
```

[2]:
```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

[3]:
```
data = pd.read_csv('water_potability.csv')
X = data.drop('Potability', axis=1)
y = data['Potability']
```

[4]:
```
data.head()
```

[4]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	\
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	

```
   Organic_carbon  Trihalomethanes  Turbidity  Potability
0       10.379783        86.990970   2.963135           0
1       15.180013        56.329076   4.500656           0
2       16.868637        66.420093   3.055934           0
3       18.436524       100.341674   4.628771           0
4       11.558279        31.997993   4.075075           0
```

[5]:
```
data.tail()
```

[5]:
```
            ph    Hardness        Solids  Chloramines     Sulfate  \
3271  4.668102  193.681735  47580.991603     7.166639  359.948574
3272  7.808856  193.553212  17329.802160     8.061362         NaN
3273  9.419510  175.762646  33155.578218     7.350233         NaN
3274  5.126763  230.603758  11983.869376     6.303357         NaN
3275  7.874671  195.102299  17404.177061     7.509306         NaN

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
3271    526.424171       13.894419        66.687695   4.435821           1
3272    392.449580       19.903225              NaN   2.798243           1
3273    432.044783       11.039070        69.845400   3.298875           1
3274    402.883113       11.168946        77.488213   4.708658           1
3275    327.459760       16.140368        78.698446   2.309149           1
```

[6]:
```
data.isnull().sum()
```

[6]:
```
ph                491
Hardness            0
Solids              0
Chloramines         0
Sulfate           781
Conductivity        0
Organic_carbon      0
Trihalomethanes   162
Turbidity           0
Potability          0
dtype: int64
```

[7]:
```
data.dtypes
```

[7]:
```
ph                float64
Hardness          float64
Solids            float64
Chloramines       float64
Sulfate           float64
Conductivity      float64
Organic_carbon    float64
```

```
Trihalomethanes    float64
Turbidity          float64
Potability           int64
dtype: object
```

[8]:
```
data.dropna()
```

[8]:
```
              ph    Hardness        Solids  Chloramines     Sulfate  \
3       8.316766  214.373394  22018.417441     8.059332  356.886136
4       9.092223  181.101509  17978.986339     6.546600  310.135738
5       5.584087  188.313324  28748.687739     7.544869  326.678363
6      10.223862  248.071735  28749.716544     7.513408  393.663396
7       8.635849  203.361523  13672.091764     4.563009  303.309771
…            …           …            …            …           …
3267    8.989900  215.047358  15921.412018     6.297312  312.931022
3268    6.702547  207.321086  17246.920347     7.708117  304.510230
3269   11.491011   94.812545  37188.826022     9.263166  258.930600
3270    6.069616  186.659040  26138.780191     7.747547  345.700257
3271    4.668102  193.681735  47580.991603     7.166639  359.948574

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
3       363.266516       18.436524       100.341674   4.628771           0
4       398.410813       11.558279        31.997993   4.075075           0
5       280.467916        8.399735        54.917862   2.559708           0
6       283.651634       13.789695        84.603556   2.672989           0
7       474.607645       12.363817        62.798309   4.401425           0
…              …               …                …          …           …
3267    390.410231        9.899115        55.069304   4.613843           1
3268    329.266002       16.217303        28.878601   3.442983           1
3269    439.893618       16.172755        41.558501   4.369264           1
3270    415.886955       12.067620        60.419921   3.669712           1
3271    526.424171       13.894419        66.687695   4.435821           1

[2011 rows x 10 columns]
```

[9]:
```
data.fillna(data.mean())
```

[9]:
```
              ph    Hardness        Solids  Chloramines     Sulfate  \
0       7.080795  204.890455  20791.318981     7.300212  368.516441
1       3.716080  129.422921  18630.057858     6.635246  333.775777
2       8.099124  224.236259  19909.541732     9.275884  333.775777
3       8.316766  214.373394  22018.417441     8.059332  356.886136
4       9.092223  181.101509  17978.986339     6.546600  310.135738
…            …           …            …            …           …
3271    4.668102  193.681735  47580.991603     7.166639  359.948574
3272    7.808856  193.553212  17329.802160     8.061362  333.775777
3273    9.419510  175.762646  33155.578218     7.350233  333.775777
```

```
3274  5.126763  230.603758  11983.869376     6.303357  333.775777
3275  7.874671  195.102299  17404.177061     7.509306  333.775777

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
0       564.308654       10.379783        86.990970   2.963135           0
1       592.885359       15.180013        56.329076   4.500656           0
2       418.606213       16.868637        66.420093   3.055934           0
3       363.266516       18.436524       100.341674   4.628771           0
4       398.410813       11.558279        31.997993   4.075075           0
…              …               …                …          …           …
3271    526.424171       13.894419        66.687695   4.435821           1
3272    392.449580       19.903225        66.396293   2.798243           1
3273    432.044783       11.039070        69.845400   3.298875           1
3274    402.883113       11.168946        77.488213   4.708658           1
3275    327.459760       16.140368        78.698446   2.309149           1

[3276 rows x 10 columns]
```

[10]:
```
data.drop_duplicates()
```

[10]:
```
            ph    Hardness        Solids  Chloramines     Sulfate  \
0          NaN  204.890455  20791.318981     7.300212  368.516441
1     3.716080  129.422921  18630.057858     6.635246         NaN
2     8.099124  224.236259  19909.541732     9.275884         NaN
3     8.316766  214.373394  22018.417441     8.059332  356.886136
4     9.092223  181.101509  17978.986339     6.546600  310.135738
…          …           …           …           …         …
3271  4.668102  193.681735  47580.991603     7.166639  359.948574
3272  7.808856  193.553212  17329.802160     8.061362         NaN
3273  9.419510  175.762646  33155.578218     7.350233         NaN
3274  5.126763  230.603758  11983.869376     6.303357         NaN
3275  7.874671  195.102299  17404.177061     7.509306         NaN

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
0       564.308654       10.379783        86.990970   2.963135           0
1       592.885359       15.180013        56.329076   4.500656           0
2       418.606213       16.868637        66.420093   3.055934           0
3       363.266516       18.436524       100.341674   4.628771           0
4       398.410813       11.558279        31.997993   4.075075           0
…              …               …                …          …           …
3271    526.424171       13.894419        66.687695   4.435821           1
3272    392.449580       19.903225              NaN   2.798243           1
3273    432.044783       11.039070        69.845400   3.298875           1
3274    402.883113       11.168946        77.488213   4.708658           1
3275    327.459760       16.140368        78.698446   2.309149           1

[3276 rows x 10 columns]
```

[11]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[12]:
```
X_train = np.array(X_train).reshape(-1, 1, 9, 1)
X_test = np.array(X_test).reshape(-1, 1, 9, 1)
X_train = X_train / 255.0
X_test = X_test / 255.0
```

[13]:
```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), strides=(2, 2), activation='relu',␣
↪input_shape=(9, 1, 1), padding='same'),
    tf.keras.layers.Conv2D(64, (3, 3), strides=(2, 2), activation='relu',␣
↪padding='same'),
    tf.keras.layers.Conv2D(64, (3, 3), strides=(2, 2), activation='relu',␣
↪padding='same'),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=10, batch_size=32,␣
↪validation_data=(X_test, y_test))
```

```
Epoch 1/10
```

```
C:\Users\JOEVIN\anaconda3\Lib\site-
packages\keras\src\layers\convolutional\base_conv.py:99: UserWarning: Do not
pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in the model
instead.
  super().__init__(
```

```
82/82             2s 5ms/step -
accuracy: 0.6005 - loss: 0.6842 - val_accuracy: 0.6280 - val_loss: 0.6577
Epoch 2/10
82/82             0s 3ms/step -
accuracy: 0.6053 - loss: 0.6722 - val_accuracy: 0.6280 - val_loss: 0.6586
Epoch 3/10
82/82             0s 3ms/step -
accuracy: 0.6035 - loss: 0.6723 - val_accuracy: 0.6280 - val_loss: 0.6595
Epoch 4/10
82/82             0s 3ms/step -
accuracy: 0.5994 - loss: 0.6738 - val_accuracy: 0.6280 - val_loss: 0.6594
Epoch 5/10
```

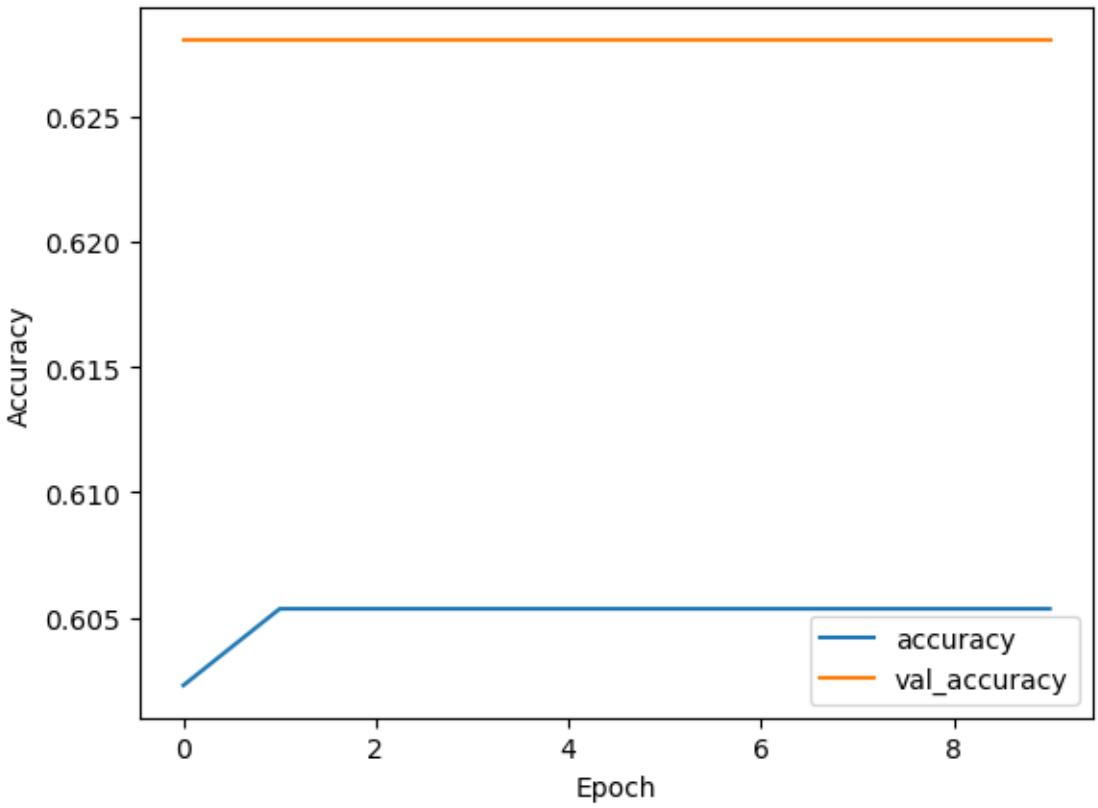
```
82/82             0s 3ms/step -
accuracy: 0.6019 - loss: 0.6725 - val_accuracy: 0.6280 - val_loss: 0.6579
Epoch 6/10
82/82             0s 3ms/step -
accuracy: 0.5917 - loss: 0.6775 - val_accuracy: 0.6280 - val_loss: 0.6596
Epoch 7/10
82/82             0s 3ms/step -
accuracy: 0.6087 - loss: 0.6698 - val_accuracy: 0.6280 - val_loss: 0.6610
Epoch 8/10
82/82             0s 3ms/step -
accuracy: 0.6135 - loss: 0.6688 - val_accuracy: 0.6280 - val_loss: 0.6585
Epoch 9/10
82/82             0s 3ms/step -
accuracy: 0.6106 - loss: 0.6695 - val_accuracy: 0.6280 - val_loss: 0.6585
Epoch 10/10
82/82             0s 3ms/step -
accuracy: 0.6079 - loss: 0.6701 - val_accuracy: 0.6280 - val_loss: 0.6595
```

[14]:
```
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print('Test Accuracy:', test_accuracy)
```

```
21/21             0s 2ms/step -
accuracy: 0.6400 - loss: 0.6562
Test Accuracy: 0.6280487775802612
```

[15]:
```
import matplotlib.pyplot as plt
```

[16]:
```
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.show()
```


0.625
0.620
0.615
0.610
0.605
Accuracy
0
2
4
6
8
Epoch
accuracy
val_accuracy

[ ]: