HW 9: Results & Conclusions

OCEN 460

Team: _/Sample_Text/

Members: Nate Baker and James Frizzell

Overview:

The purpose of this project is to use existing data on the growth of coral to predict whether coral can grow given oceanographic conditions. The latitude, longitude, depth, temperature, salinity, and dissolved oxygen levels are used to predict a binary value with 1 meaning that coral can grow and 0 meaning that coral cannot grow.

1) Setup Class Structure

This code has been written in an object-oriented fashion so that it may easily be packaged and shipped into production later.

First, we set up the class structure and import the internal libraries used.

```
import random

import tensorflow as tf
import numpy as np
from itertools import product
from statistics import mean, median

class CoralPrediction:
    def __init__(self):
        self.model = None
        self.params = np.zeros(6)
        self.test_lat = []
        self.test_long = []
        self.test_depth = []
        self.test_temp = []
        self.test_sal = []
        self.test_oxy = []

    def set_model(self, modelpath):
        self.model = tf.keras.models.load_model(modelpath)

    def predict(self, params):
        self.params = params
        missing = []
        for i in range(len(self.params)):
            if type(self.params[i]) != float and type(self.params[i])
```

```
!= int:
                missing.append(i)

        if 0 in missing:
            self.test_lat = np.linspace(-90, 90, 90)
        else:
            self.test_lat = [self.params[0]]

        if 1 in missing:
            self.test_long = np.linspace(-180, 180, 180)
        else:
            self.test_long = [self.params[1]]

        if 2 in missing:
            self.test_depth = np.linspace(0, 3000, 50)
        else:
            self.test_depth = [self.params[2]]

        if 3 in missing:
            self.test_temp = np.linspace(-2, 28, 20)
        else:
            self.test_temp = [self.params[3]]

        if 4 in missing:
            self.test_sal = np.linspace(0, 41, 20)
        else:
            self.test_sal = [self.params[4]]

        if 5 in missing:
            self.test_oxy = np.linspace(0.2, 132, 40)
        else:
            self.test_oxy = [self.params[5]]

        cond_list = list(product(*[self.test_lat, self.test_long,
self.test_depth, self.test_temp, self.test_sal, self.test_oxy]))
        if len(cond_list) >= 4000:
            cond_list = random.sample(cond_list, 4000)
        count = 0
        predictions = []
        for cond in cond_list:
            print(count, ' completed out of: ', len(cond_list))
            predictions.append(self.model.predict([list(cond)])[0][0])
            count += 1
        return mean(predictions), median(predictions)
```

The CoralPrediction class has a few properties and methods. Once the object has been instantiated in the main.py function, the user will assign a trained tensorflow model to the predictor using the set_model function.

Then, the user will pass the sea state conditions into the predictor model with the predict function. The code will evaluate the conditions and return the probability of coral being able to grow given the inputs.

Some precautions have been implemented to assist the user in the case that not all metocean data is known. This will be discussed further in section 3.

2) MAIN.py, Runnable function

```
import os
import pathlib
# import CoralClass


def __main__(modelpath, conditions):
    CoralPredictor = CoralPrediction()
    CoralPredictor.set_model(modelpath)
    mean_pred, med_pred = CoralPredictor.predict(conditions)
    print('\n----------------------------------------')
    print('The average likelihood of coral growth is: ',
mean_pred*100, '%')
    print('The median likelihood of coral growth is: ', med_pred*100,
'%')


# if __name__ == '__main__':           Failed implementation of
command-line run functionality
#     import sys
#     args = sys.argv[2:]
#     modelpath = sys.argv[1]
#     pythonname = sys.argv[0]
#     __main__(modelpath, args)

# change this to change the model that will be used to predict the
coral growth
modelpath = str(pathlib.Path(os.getcwd()).parent) + '/coral-
prediction/models/trial0.3.h5'

# Enter the metocean conditions here   [Latitude, Longitude, Depth
(m), Temperature (C), Salinity (ppt), Dissolved O2 (umol/kg)]
# Enter "None" if the data is unknown
conditions = [20, -150, 1000, None, 30, None]

__main__(modelpath, conditions)

0  completed out of:  800
1  completed out of:  800
2  completed out of:  800
3  completed out of:  800
```

```
753  completed out of:  800
754  completed out of:  800
755  completed out of:  800
756  completed out of:  800
757  completed out of:  800
758  completed out of:  800
759  completed out of:  800
760  completed out of:  800
761  completed out of:  800
762  completed out of:  800
763  completed out of:  800
764  completed out of:  800
765  completed out of:  800
766  completed out of:  800
767  completed out of:  800
768  completed out of:  800
769  completed out of:  800
770  completed out of:  800
771  completed out of:  800
772  completed out of:  800
773  completed out of:  800
774  completed out of:  800
775  completed out of:  800
776  completed out of:  800
777  completed out of:  800
778  completed out of:  800
779  completed out of:  800
780  completed out of:  800
781  completed out of:  800
782  completed out of:  800
783  completed out of:  800
784  completed out of:  800
785  completed out of:  800
786  completed out of:  800
787  completed out of:  800
788  completed out of:  800
789  completed out of:  800
790  completed out of:  800
791  completed out of:  800
792  completed out of:  800
793  completed out of:  800
794  completed out of:  800
795  completed out of:  800
796  completed out of:  800
797  completed out of:  800
798  completed out of:  800
799  completed out of:  800

----------------------------------------
```

```
The average likelihood of coral growth is:  63.69994878768921 %
The median likelihood of coral growth is:  98.32048416137695 %
```

This code allows the user to input conditions and see the output. First, the user must specify the path of a trained tensorflow model which will be used to predict the growth of the coral.

Then, the user inputs metocean conditions and runs the program.

The code will report the mean and median likelihood that coral can grown in the given conditions. If all the parameters are specified, the mean will equal the median (since there is only 1 data point that is predicted)

3) Probablistic Modelling

In many real-world cases, not all the data will be known for the location of interest. For example, a climatologist may be studying the South Pacific for coral growth around the Great Barrier Reef. The scientist will have a GPS coordinate of interest (latitude and longitude) and likely will know the ocean depth at that location. However, the temperature, salinity and dissolved oxygen content may be unknown. How can the research continue if the input data is incomplete?

This is solved with probabilistic prediction. In the case that some inputs are not known (are entered as "None" in the inputs) the code will sweep through a range of possible values based on the training data that was earlier used. For each value in the range, the program will predict the coral growth probability. Once all the possible values have been simulated, the code will take the average and median probability and report it back to the user.

4) Conclusions

This project has been successful in predicting the ability of coral to grow in certain metocean conditions by merging data from multiple sources. In the future, a higher