m2-hw2

February 4, 2024

1 Porblem 2 (30 points)

1.1 Problem Description

In this problem you will use linear least squares to fit a linear function to a 3D temperature field, with x,y,z locations and an associated temperature T.

Fill out the notebook as instructed, making the requested plots and printing necessary values.

You are welcome to use any of the code provided in the lecture activities.

Summary of deliverables: Results: - Predicted temperature `T(5,5,5)` using a hand-coded LLS squares model with a linear function - Direction of travel from `(5,5,5)` to experience the greatest decrease in temperature

Discussion: - Reasoning for how we can use our fitted function to determine the direction of greatest decrease in temperature

Imports and Utility Functions:

[12]:
```
import numpy as np
```

1.2 Load the data

The data is contained in `tempfield.npy` and can be loaded with `np.load(tempfield.npy)`. The first three columns correspond to the `x`, `y`, and `z` locations of the data points, and the 4th column corresponds to the temperature `T` at the respective point. Store the data as you see fit.

[13]:
```
# YORU CODE GOES HERE
tempfield_data = np.load("tempfield.npy")
x = np.array(tempfield_data[:,0])
y = np.array(tempfield_data[:,1])
z = np.array(tempfield_data[:,2])
T = np.array(tempfield_data[:,3])
```

1.3 LLS Regression in 3D

Now fit a linear function to the data using the closed for of LLS regression. Use your fitted function to report the predicted temperature at `x = 5, y = 5, z = 5`. You are free to add regularization to your model, but this is not required and will not be graded.

```
[14]: # YOUR CODE GOES HERE
X = np.c_[x, y, z, np.ones(len(x))]
w = np.linalg.inv(X.T @ X) @ X.T @ T.reshape(-1, 1)
# Given x, y, z coordinates, predict the temperature at that point
x_pred, y_pred, z_pred = 5, 5, 5

# temperature prediction logic
T_pred = np.array([x_pred, y_pred, z_pred, 1])@ w
print(f'The predicted temperature at ({x_pred}, {y_pred}, {z_pred}) is:␣
 ↪{T_pred[0]}')

# direction of travel logic
direction = -1*w[0:3].flatten()
direction/np.linalg.norm(direction)
print(f'The direction of travel to experience the greatest decrease in␣
 ↪temperature is: {direction}')
```

```
The predicted temperature at (5, 5, 5) is: 21.389073383508922
The direction of travel to experience the greatest decrease in temperature is: [
0.13421269 -0.11741862 -0.72677374]
```

1.4 Gradient Intuition

Using the function you fit in the previous part, in which direction should one move from the point `p = (5,5,5)` to experience the largest decrease in temperature in the immediate neighborhood of the point? Report the specific direction, along with your reasoning.

1.4.1 *Your answer goes here*

The negative of the gradient moves you in the direction of the maximun decreasing rate of a function. One should move in the direction of [0.13421269 -0.11741862 -0.72677374] from the point point `p = (5,5,5)` to experience the largest decrease in temperature.