

Linear Regression Model

Summary

This project has involved writing a protocol in object oriented C++. The task was to create a linear regression model on an old heating system, the current temperature is predicted based on a more advanced sensor which functions as the reference output. We integrated this model into the existing codebase, ensuring it worked seamlessly with the rest of the program. Afterward, we tested it with sensor data to verify accuracy and reliability. This project enhanced our understanding of C++ programming, sensor data processing, and linear regression.

LinReg.h/LinReg.cpp

LinReg construct - This is the construction of the model, this is where we add all the input data for the prediction

getBias() - A get method to retrieve the bias data

getWeight() - A get method to retrieve the weight data

getTrainingSetCount() - The number of stored training sets.

predict() - This is where the prediction calculation is made: $Bias + Weight * input$

train() - This function trains the model, based on *getTrainingSetCount()* which determines how many times the function are going to run, if the input equals zero which in this case it does on the first run reference (*TrainingOutput()*) should equal *myBias*. In all other cases the function does the calculations:

- $Bias += error * LearningRate$
- $Weight += error * LearningRate * input$

This adjusts the prediction with *myLearningRate* which is set as default value 10% for every training session.

Name changes

Button0 ⇒ *predictionButton*

Led1 ⇒ *errorLed*

Timer0 ⇒ *debounceTimer*

Timer1 ⇒ *debounceTimer*

Main.cpp

tempSensorPin{}; - Temperature sensor connected to pin 2.

Vcc{5.0}; - Supply voltage of the temperature sensor.

trainingInput{} - Input value that the program gets trained on.

trainingOutput{} - Output value that the program gets trained on.

```
if(linReg.predict(input) < 0){  
    serial::printf("%d ", static_cast<int>(linReg.predict(input) - 0.5));  
}  
else{  
    serial::printf("%d ", static_cast<int>(linReg.predict(input) + 0.5)); - rounding function for  
positive and negative numbers
```

Referens

<https://github.com/PhilipOberg94/Projekt-1-Linear-regression/>

