

MATLAB/ Simulink | Lineare Regression

Beschreibung

- (a) `runScript.m`
- (b) `LinearRegressionDataFormatter.m`
- (c) `LinearRegressionModel.m`
- (d) `GradientDescentOptimizer.m`
- (e) `TemperatureMeasurement.mat`

In dieser Hausaufgabe werden Sie den in der Vorlesung besprochenen Use Case zur Linearen Regression in Matlab umsetzen. Die Programmstruktur ist bereits vorgegeben. Machen Sie sich daher zunächst mit dieser vertraut.

Datei a (`runScript`) dient zum Starten des Programms. Dabei wird zunächst ein **LinearRegressionDataFormatter** Objekt erstellt, welches die Trainingsdaten aus Datei e in eine Struktur lädt, die vom **LinearRegressionModel** Objekt verarbeitet werden kann. Neben den eigentlichen Daten, beinhaltet das **LinearRegressionModel** zusätzlich ein **GradientDescentOptimizer** Objekt, welches den Trainingsvorhang durchführt. Ziel ist es bei der bereits gegebenen learning rate die maximale Anzahl an Iteration so einzustellen, dass in annehmbarer Rechenzeit das Optimum vom Optimierer gefunden wird. Vervollständigen Sie in den Klassen b - d den Code an den entsprechend gekennzeichneten Stellen:

% ===== YOUR CODE HERE =====

Bewertungskriterien

Sie können mit dieser Abgabe 10 Punkte* erreichen. Sie erhalten zunächst für die Umsetzung, welche die Funktionalität des Programms sicherstellt 60% der maximal erreichbaren Punkte. Darüber hinaus, entfallen 40% auf Stiel und performance-orientierte Implementierung. Halten Sie sich dazu, wie in der Vorlesung besprochen, an ein Minimum an Clean-Code Regeln. Des Weiteren vermeiden Sie nach Möglichkeit wie in Listing 1 dargestellt Schleifen und implementieren Sie stattdessen vektorisierte Rechenoperationen (Listing 2).

*In drei über das Semester verteilte Hausaufgaben können insgesamt 30 Punkte erzielt werden. D.h. diese Abgabe geht zu einem Drittel in die Gesamtbewertung ein.

```

1 theta = [1;2];
2 myVec = [1;2;3;4;5];
3 myMat = [ones(length(myVec),1) myVec];
4 resultVec = zeros(length(myVec),1);
5
6 for i = 1:length(myMat)
7     resultVec(i) = myMat(i,1) * theta(1) + myMat(i,2) * theta(2);
8 end

```

Listing 1: bad implementation

```

1 theta = [1;2];
2 myVec = [1;2;3;4;5];
3 myMat = [ones(length(myVec),1) myVec];
4 resultVec = zeros(length(myVec),1);
5
6 resultVec = myMat * theta

```

Listing 2: good implementation

Abgabefrist

Geben Sie ihre Lösung bis **09.04.2021 um 12:00 Uhr mittags** über ihr persönliches Kursrepository ab. Lösungen, welche nach der Deadline gepusht werden, können nicht berücksichtigt werden. Da die Hausaufgaben eine etwaige Klausur ersetzen, führen Plagiate direkt zum Nicht-Bestehen des Kurses.