

# Leren: Programmeeropdracht 2

Week 2 - Semester 1 2014/15

**Deadline: maandag 10 November, 12.00 uur**

**Deze opdracht mag in teams van twee gemaakt worden**

1. (1 punt) *Multivariate Lineaire Regressie met gradient descent*

Maak op basis van je implementatie van univariate lineaire regressie nu multivariate lineaire regressie met gradient descent. Maak een toplevel aanroep "multiple" waarmee het programma wordt gestart. Input is weer het aantal iteraties en de learning rate. Output moet zijn: de (parameterwaarden van) de regressiefunctie.

Gebruik de data van housesRegr.csv en de data uit de schrijfoopdracht als voorbeeld. Die laatste zijn:

<b>X1</b>	5	5	3
<b>X2</b>	3	5	3
<b>Y</b>	6	6	10

Bouw de implementatie op uit de volgende componenten:

1. Inlezen van de data van een file. (Maak zelf een datafile in een handig formaat heeft. Zonodig converteer je andere data met de hand naar dit formaat, bv. met excel of met een editor.)
2. Berekenen van de gradient als functie van de parameters.
3. Updaten van de parameters
4. Berekenen van de cost  $J(\theta)$ .
5. Belangrijke details van het algorithm zijn de learning rate en de terminatie. Ga uit van een vast aantal iteraties als basis voor terminatie. Varieer de learning rate en het aantal iteraties om de optimale waarden te vinden voor de learning rate en het aantal iteraties.

*Inleveren: code voor gradient, cost function, parameter updating, voorspeller, plot van data, plot van regressiefunctie, resultaat van experimenten.*

2. (1 punt) *Hogere orde polynomen*

Breid het programma uit met een component die kwadraten van alle input variabelen toevoegt. Pas het resultaat toe op de data houses.csv en vergelijk de "loss" met de variant zonder kwadraten. Inleveren: output + korte bespreking van de resultaten.

3. (2 punten) *Logistische Regressie*

Implementeer het algorithm voor logistische regressie mbv. gradient descent. Aanroep van het programma moet zijn "logreg". Gebruik ook hier het aantal iteraties als stopcriterium. Bouw de implementatie op uit de volgende componenten:

1. Inlezen van de data van een file. (Maak zelf een datafile in een handig formaat heeft. Zonodig converteer je andere data met de hand naar dit formaat, bv. met excel of met een editor.)
2. Berekenen van de gradient als functie van de parameters.
3. Updaten van de parameters
4. Berekenen van de cost  $J(\theta)$ .
5. Paarsgewijze vergelijking voor meerdere klassen
6. Belangrijke details van het algorithm zijn de learning rate en de terminatie. Ga uit van een vast aantal iteraties als basis voor terminatie. Varieer de learning rate en het aantal iteraties om de optimale waarden te vinden voor deze data. Gebruik de data in onderstaande tabel en ook de data op digits.csv. Dit zijn pixelwaarden van handgeschreven cijfers 1, 2 en 3. Elk voorbeeld is een matrix van grijswaarden die is gelabeld als 1, 2 of 3.

<b>X1</b>	5	5	3	2
<b>X2</b>	3	5	3	4
<b>Y</b>	0	0	1	1