Tarea 3: Regresión logística

Este código base presenta un optimizador incompleto, que permite explorar los conceptos de descenso de gradiente. Un regresor lineal sirve de base para entender cómo utilizar all optimizador, y con esa información debe implementarse un regresor logístico básico para clasificar el sexo de pingüinos basado en algunas características fenotípicas como longitud y profundidad del pico, longitud de las aletas, peso, etc.

Dependencias

Este código utiliza los paquetes statistics y automatic-differentiation. Desde la terminal de GNU/Octave los instala con:

```
pkg install -forge statistics
pkg install "https://github.com/StevenWaldrip/Automatic-Differentiation/archive/refs/tags/1.0.0.tar.gz"
```

Ejecución de la solución

- Ubiquese en la carpeta `root` del repositorio.

Regresión lineal

- Ejecute el siguiente código en la terminal de GNU/Octave:

```
octave regression_linear.m
```

- Corresponde a la solución de la parte 1 de la tarea.

Regresión logística

- Ejecute el siguiente código en la terminal de GNU/Octave:

```
octave regression_logistic.m
```

- Corresponde a la solución de la parte 2,3,4,5,6,7,8,9,10,11,12 de la tarea.

Nota: Al principio del código se encuentran flags para ejecutar partes especificas de la tarea por defecto se ejecutan todas las partes. Si desea ejecutar solo una parte, ponga la flag en `false`.

Flag	Descripción
`part_four`	Corresponde a la solución 2,3,4,5,6 de la tarea.
`part_seven`	Corresponde a la solución 7,8,9 de la tarea.
`part_extra_points`	Corresponde a la solución 12 de la tarea. Se debe tener la flag `part_seven` en `true`.
`part_ten`	Corresponde a la solución 10 y 11 de la tarea.

Documentación de la Solución

Parte 1

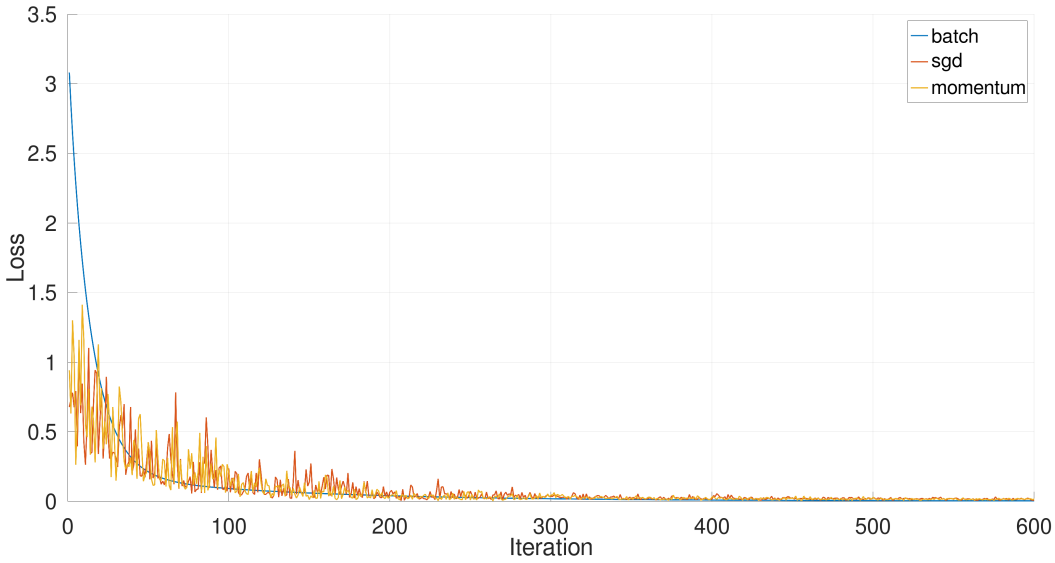
- Se agrega el update del descenso de gradiente en `optimizer`.

```
## theta update with gradient descent
function theta_new = updateGD(self,theta_current,gradient)
  theta_new = theta_current - self.alpha*gradient;
endfunction
```

- Se agrega el método de `"batch"` y `"sgd"` en `optimizer`.

```
switch (self.method)
  case "momentum"
    sampler=samplerMB;
    updater=@(tc,g) self.updateMomentum(tc,g);
  case "batch"
    sampler=samplerB;
    updater=@(tc,g) self.updateGD(tc,g);
  case "sgd"
    sampler=samplerMB;
    updater=@(tc,g) self.updateGD(tc,g);
  otherwise
    error("Method not implemented yet");
endswitch
```

- Se gráfica el error de la perdida de la regresión lineal, para los métodos `"batch"`, `"sgd"` y `"momentum"`.



Parte 2 y 3

- Se agrega la hipotesi de regresión logística.

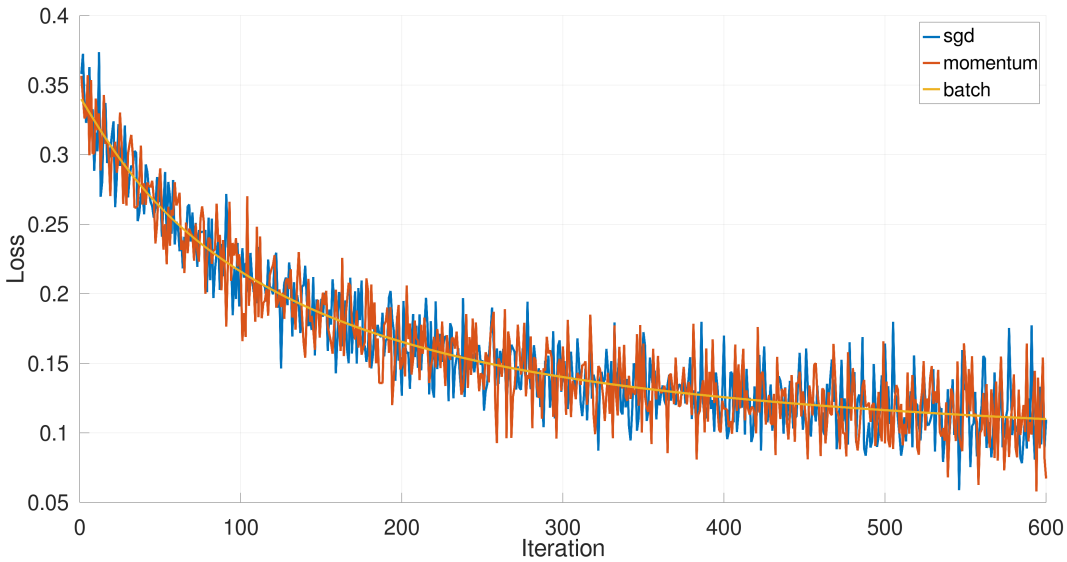
```
% Hypothesis function used in logistic regression
function h=logreg_hyp(theta,X)
  assert(columns(theta)==1)
  assert(columns(X)==rows(theta))
  h = 1 ./ (1 + exp(-(X*theta)));
endfunction
```

- Se agrega la perdida de la regresión logística.

```
% Loss function used in logistic regression
function err=logreg_loss(theta,X,y)
  assert(rows(y)==rows(X))

  ## residuals
  r=y-logreg_hyp(theta,X);
  err=1/rows(y)*(r'*r); # OLS
endfunction
```

Parte 4, 5, 6

- Se optimiza la regresión logística con los metodos de `"batch"`, `"sgd"` y `"momentum"`. Y se gráfica el error de la perdida.



- Se calcula el error emprico de la regresión logística.

```
function [total_errors, percentage_error]= logreg_empirical_error(theta,X,y)
  assert(rows(y)==rows(X))
  h = round(logreg_hyp(theta,X));
  total_errors =  sum((y - h) != 0);;
 percentage_error = 100*total_errors/rows(y);
endfunction      You, 19 hours ago • feat: add emperical error
```

- Se obtiene el resultado del error emprico de la regresión logística, para cada de los métodos de optimización.

```
Probando método 'sgd'.
100% ==============================================================================
theta =

 -0.116149
 -0.451153
 -1.238336
 -0.072030
 -1.021229

Training error: 34 / 266 (12.782 %)
Test error: 4 / 67 (5.97015 %)
```

```
Probando método 'momentum'.
100% ==============================================================================
theta =

 -0.112242
 -0.450608
 -1.242790
 -0.078451
 -1.021036

Training error: 34 / 266 (12.782 %)
Test error: 5 / 67 (7.46269 %)
```

```
Probando método 'batch'.
100% ==============================================================================
theta =

 -0.116575
 -0.454416
 -1.238997
 -0.071902
 -1.019511

Training error: 34 / 266 (12.782 %)
Test error: 4 / 67 (5.97015 %)
```

- Se observa que el de menor error empírico fue el `"sgd"` y el `"batch"` .

¿Cuáles son las features más importantes para la regresión logística?

Si ordenamos de mayor magnitud a menor el valor de la componente de θ en el caso del método `"sgd"` , observamos que:

Componente de θ	Magnitud

Componente de θ	Magnitud
θ_3 "Culmen Depth (mm)"	1.238336
θ_5 "Body Mass (g)"	1.021229
θ_2 "Culmen Length (mm)"	0.451153
θ_1 "Bias"	0.116149
θ_4 "Flipper Length (mm)"	0.072030

Entonces vemos que las features más relevantes para la regresión logística para determinar el sexo son la profundidad del pico, el peso y la longitud del pico.

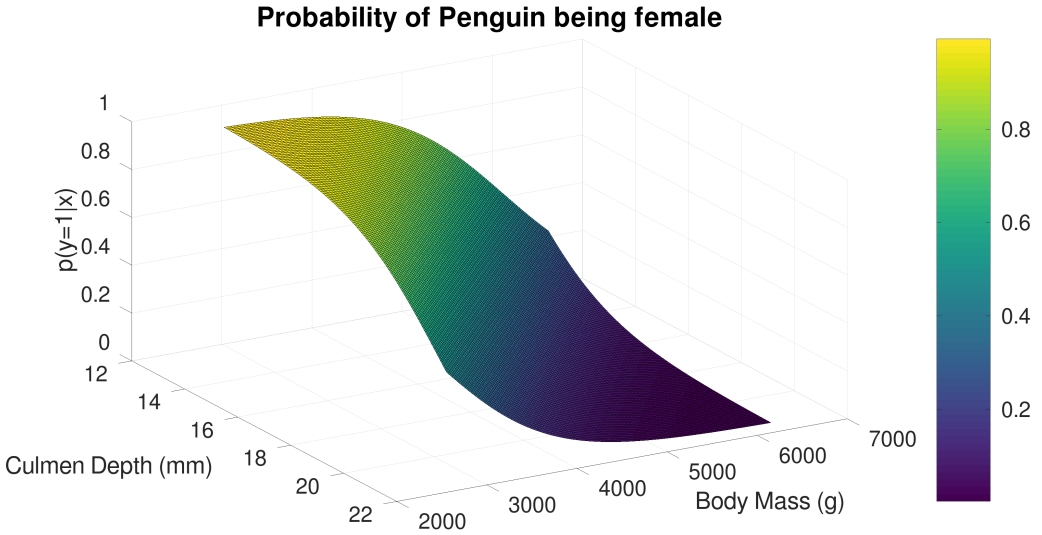
Porque estas en teoría son las features más relevantes para la regresión logística. Esto es porque entre mayor sea la magnitud de del theta esta feature tiene mayor peso en el calculo de la hipótesis de regresión logística.

Parte 7, 8, 9

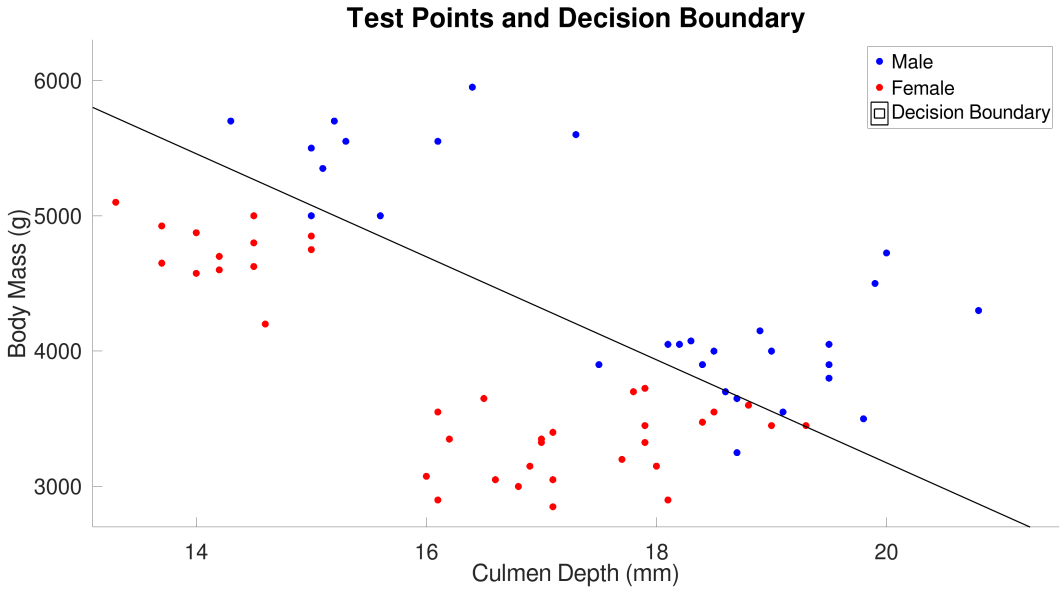
- Para encontrar de forma empírica la features más relevantes para la regresión logística para conocer el sexo de los pingüinos, se calcula el error empírico para dos pares de features. Utilizamos el método `"sgd"` para la regresión logística, para la minización del error.

```
##### PART SEVEN #####

Combination:    1   2
100% ==============================================================================
Test error: 25 / 67 (37.3134 %)
Combination:    1   3
100% ==============================================================================
Test error: 26 / 67 (38.806 %)
Combination:    1   4
100% ==============================================================================
Test error: 25 / 67 (37.3134 %)
Combination:    1   5
100% ==============================================================================
Test error: 28 / 67 (41.791 %)
Combination:    2   3
100% ==============================================================================
Test error: 14 / 67 (20.8955 %)
Combination:    2   4
100% ==============================================================================
Test error: 26 / 67 (38.806 %)
Combination:    2   5
100% ==============================================================================
Test error: 27 / 67 (40.2985 %)
Combination:    3   4
100% ==============================================================================
Test error: 9 / 67 (13.4328 %)
Combination:    3   5
100% ==============================================================================
Test error: 6 / 67 (8.95522 %)
Combination:    4   5
100% ==============================================================================
Test error: 28 / 67 (41.791 %)


Best combination are:

Combination:
  3   5
[Culmen Depth (mm), Body Mass (g)]
percentage error: 8.95522 %
############
```

- Obtuvo que θ_3 "Culmen Depth (mm)" y θ_5 "Body Mass (g)" son las features más relevantes. Esto concuerda con el análisis del punto 4.
- Se dibuja la superficie generada por la regresión logística de las features "Culmen Depth (mm)" y "Body Mass (g)".



- Se gráfica la fontera de decisión de las features "Culmen Depth (mm)" y "Body Mass (g)".



Parte 10 y 11

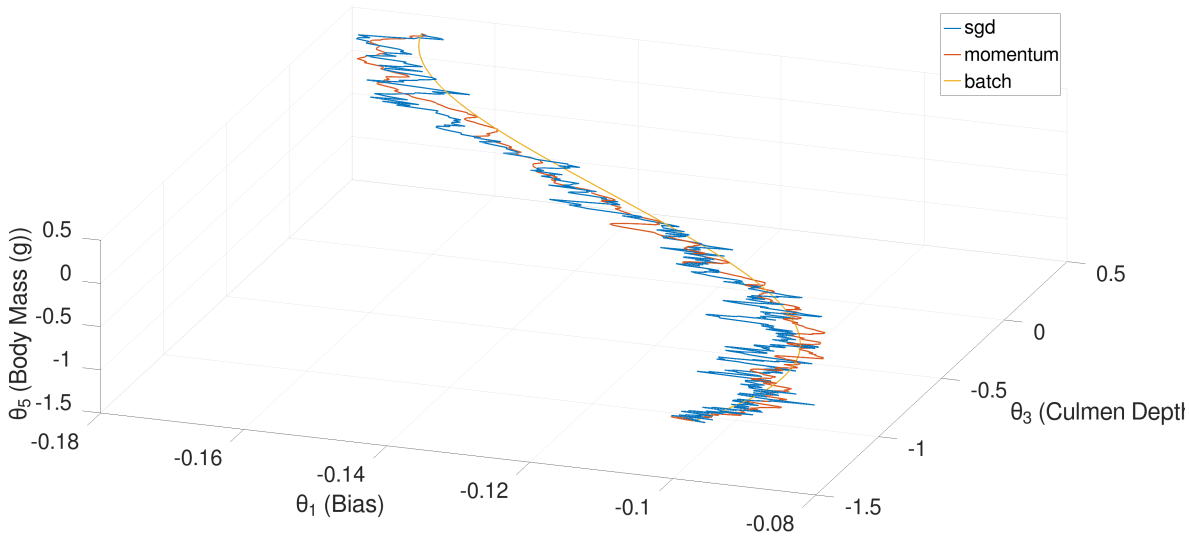
- Se vuelve al calcular el error empírico de la regresión logística pero para 3 features.

```
##### PART TEN #####

Training Combination:   1  2  3
100% ==========================================================================
Training Combination:   1  2  4
100% ==========================================================================
Training Combination:   1  2  5
100% ==========================================================================
Training Combination:   1  3  4
100% ==========================================================================
Training Combination:   1  3  5
100% ==========================================================================
Training Combination:   1  4  5
100% ==========================================================================
Training Combination:   2  3  4
100% ==========================================================================
Training Combination:   2  3  5
100% ==========================================================================
Training Combination:   2  4  5
100% ==========================================================================
Training Combination:   3  4  5
100% ==========================================================================


The most important feature combination are:

Combination:
  1  3  5
[Bias, Culmen Depth (mm), Body Mass (g)]
percentage error: 7.46269 %
############
```

-

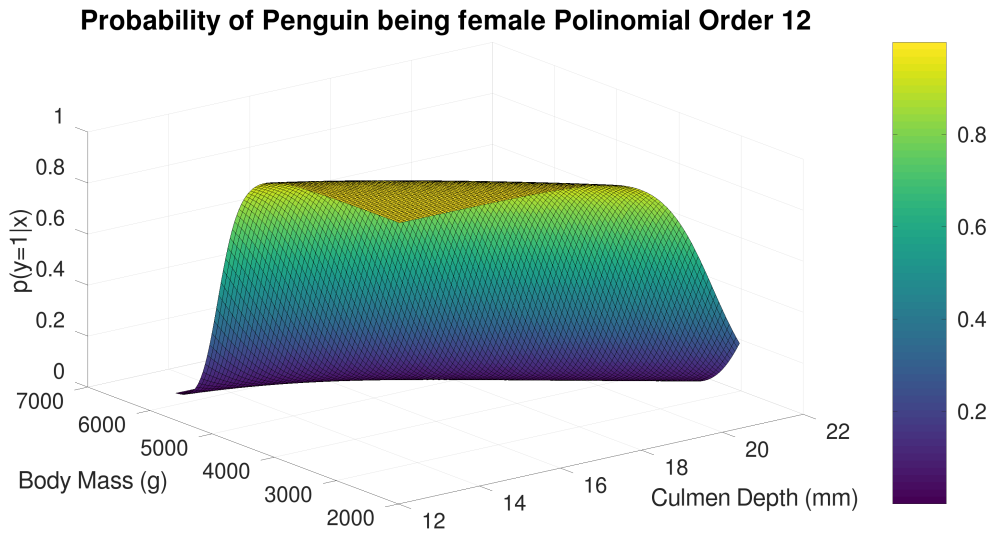
Se obtiene que las 3 features más relevantes para la regresión logística son θ_3 "Culmen Depth (mm)", θ_5 "Body Mass (g)" y θ_1 "Bias". Utilizamos el método `"sgd"` para la regresión logística, para la minización del error.

- Se desconoce el porque dio un menor error, con "Bias" en vez que con "Culmen Length (mm)".
- Se gráfica el perdida de la regresión logística vs la componentes del θ.



Parte 12 (Extra Points)

- Se usa el resultado de la parte 6, para el uso de las features más relevantes para la regresión logística.
- Se dibuja la superficie generada por la regresión logística de las features "Culmen Depth (mm)" y "Body Mass (g)". Pero con orden 12 de la matriz de diseño.



- Se gráfica la fontera de decisión de las features "Culmen Depth (mm)" y "Body Mass (g)". Pero con orden 12 de la matriz de diseño.