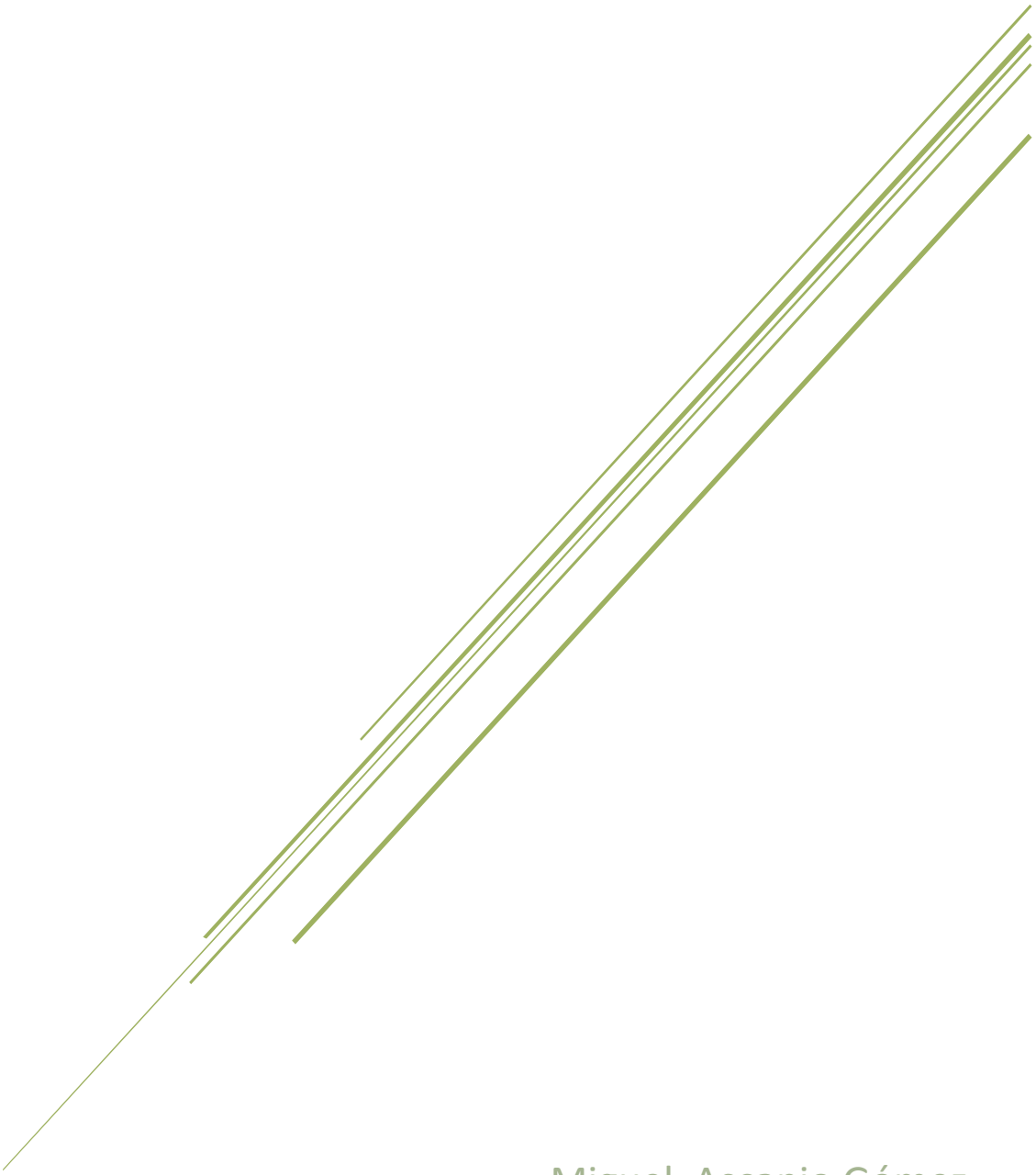
APRENDIZAJE AUTOMÁTICO

Práctica 3



Miguel Ascanio Gómez

Esther Ávila Benito

En esta práctica se ha implementado en Octave un método para el reconocimiento de imágenes de números manuscritas mediante el método de regresión logística.

REGRESIÓN LOGÍSTICA

VISUALIZACIÓN DE LOS DATOS

```
load("ex3data1.mat");
m = size(X, 1);

# Mediante la función displayData mostramos en pantalla una muestra aleatoria de 100 ejemplos de entrenamiento

rand_indices = randperm(m);
sel = X(rand_indices(1:100), :);

displayData(sel);
```

VECTORIAZACIÓN DE LA FUNCIÓN DE COSTE Y DEL GRADIENTE

Mediante la siguiente función obtenemos el gradiente y el coste de la regresión logística con el fin de que éste último se vaya reduciendo. Esta función es equivalente a la que se ha utilizado en prácticas anteriores, con la diferencia de que es una versión vectorizada.

```
function [J, grad] = lrCostFunction(theta, X, Y, lambda)
        warning("off", "Octave:broadcast");

        m = length(X(:,1));
        valoresH = (theta' * X')';
        h = sigmoide(valoresH);
        # Sumandos
        a = (0.-Y) .* log(h) .- ((1 .- Y) .* log(1 .- h));
        # Theta cuadrado
        t = theta .^ 2;

        J = (1/m) * sum(a) + ((lambda / m / 2) * sum(t));

        # Sumandos
        a = (h .- Y) .* X;
        # Se pone a cero para no incluir el primero
        theta(1) = 0;

        grad = (1/m * sum(a)) .- ((lambda/m) .* theta)';

        warning("on", "Octave:broadcast");

endfunction
```

CLASIFICACIÓN DE UNO FRENTE A TODOS

Con la función *oneVsAll* se compara cada ejemplo de entrenamiento con todas las etiquetas posibles con el fin de encontrar una función (*fminunc*) que indica con la mayor precisión posible si dicho ejemplo es equivalente a la etiqueta con la que se compara.

```
function [all_theta] = oneVsAll(X, y, num_etiquetas, lambda)

	initial_theta = zeros(length(X(1,:)), 1);
	for c = 1:num_etiquetas
		options = optimset("GradObj", "on", "MaxIter", 50);
		all_theta(:,c) = fminunc(@(t)(lrCostFunction(t, X, (y == c), lambda)), initial_theta,
options);
	endfor

endfunction
```

PRUEBA DE PRECISIÓN

```
load("ex3data1.mat");
m = size(X, 1);

num_etiquetas = 10;

theta = oneVsAll(X, y, num_etiquetas, 1);

display("Porcentaje de aciertos: ");
prueba1(theta, X, y)

# Porcentaje de aciertos

function porcentaje = prueba1(theta, X, y)
	m = length(X(:,1));

	valoresH = (theta' * X')';
	h = sigmoide(valoresH);


	mm = max(h')';
	aciertos = 0;
	for i = 1:m
		if (find(mm(i) == h(i,:)) == y(i))
			aciertos++;
		endif
	endfor

	porcentaje = aciertos / m * 100;


endfunction
```

El resultado es de 96,6% de elementos de entrenamiento bien clasificados.

RED NEURONAL

En esta parte se comparan los resultados obtenidos para un conjunto de ejemplos de entrenamiento con los resultados esperados con el fin de determinar la precisión sobre los ejemplos. La precisión obtenida es de 97.5 %.

```
function h = red(Theta1, Theta2, X)

a1 = [1; X];

z2 = Theta1 * a1;
a2 = sigmoide(z2);
a2 = [1; a2];

z3 = Theta2 * a2;
h = sigmoide(z3);

endfunction

--------------------------------------------------------------------------------

load("ex3data1.mat")
load('ex3weights.mat');

aciertos = 0;

m = size(X, 1);
for i = 1:m
# Calcular los aciertos mientras se calculan los resultados de la red
        resultado = red(Theta1, Theta2, X(i, :)');
        mx = max(resultado);
        if(find(resultado == mx) == y(i))
                aciertos++;
        endif
endfor
disp("Porcentaje de aciertos:")
aciertos / m * 100
```