UNIVERSIDAD NACIONAL DE INGENIER´IA FACULTAD DE CIENCIAS

INTELIGENCIA ARTIFICIAL



Informe 1 : REDES NEURONALES

Alumno:

Julca Carhuaz Danny Michael 20111320H

Profesor:

Moran C´ardenas Antonio

**´Indice**

1. [Redes Neuronales 1](#_TOC_250006)
   1. [Entrenamiento Batch 1](#_TOC_250005)
      1. [Funci´on Sigmoidea 2 2](#_TOC_250004)
      2. [Funci´on Gaussiana 5](#_TOC_250003)
   2. [Entrenamiento Patron 8](#_TOC_250002)
      1. [Funci´on Sigmoidea 2 10](#_TOC_250001)
      2. [Funci´on Gaussiana 13](#_TOC_250000)

# Redes Neuronales

En el presente informe se analiza los algoritmos de aprendizaje Batch y Patron. La red neuronal presenta una sola capa oculta.

Las variaciones se realizan en las funciones de sigmoidea y gaussiana, variando el parametro de aprendizaje (eta) y el nu´mero de neuronas en la u´nica capa oculta (nm).

Todas las pruebas con con bias.

## Entrenamiento Batch

*% E n t re n a m ie n t o Batch con b i a s*

**c l e a r** ;

**c l c** ;

**c l o s e a l l** ; a = 3 ;

b = 4 ;

x = 4 : 0 . 1 : 4 ;

−

x = x ’ ;

N = **length** ( x ) ; yb = a x + b ;

∗

yb = yb + 0 . 7 5 **randn** (N, 1 ) ; ne = 1 ;

∗

nm = 5 ;

b i a s = **input** ( ’ B i as : S I = 1 : ’ ) ;

**i f** ( b i a s == 1 ) ne = ne +1;

x = [ x **ones** (N, 1 ) ] ;

**end**

v = 0 . 1 5 **randn** ( ne ,nm ) ; w = 0 . 1 5 **randn** (nm, 1 ) ; e t a = **input** ( ’ e t a : ’ )

∗

∗

**f o r** i t e r = 1 : 2 0 0 0

dJdv = 0 ; dJdw = 0 ;

**f o r** k = 1 :N

i n = ( x ( k , : ) ) ’ ; m = v ’ i n ;

∗

n = 2 . 0 . / ( 1 + **exp**( m) ) 1 ; *% S ig m o id e a 2*

− −

*% n = e x p ( m. ˆ 2 ) ; % Gaussiana*

−

out = w’ n ;

∗

y ( k , 1 ) = out ;

e r = out yb ( k , 1 ) ; **error** ( k , 1 ) = e r ; dndm = ( 1 − n . ∗ n ) / 2 ;

−

*% dndm =* −*2.0*∗*(n .* ∗*m) ;*

**end**

dJdw = dJdw + e r . ∗ n ;

dJdv = dJdv + e r . ∗ i n ∗ (w. ∗ dndm ) ’ ;

*% w = w* − *e t a* ∗ *dJdw/ nx ;*

*% v = v* − *e t a* ∗ *dJdv / nx ;*

**end**

w = w e t a dJdw/N; v = v e t a dJdv /N;

JJ = 0 . 5 **sum**( **error** . **error** ) J ( i t e r , 1 ) = JJ ;

∗ ∗

− ∗

− ∗

**f i g u r e** ( 1 ) ;

**plot** ( x ( : , 1 ) , y , x ( : , 1 ) , yb , ’ ’ ) ;

∗

**f i g u r e** ( 2 ) ;

**plot** ( J ) ;

### Funci´on Sigmoidea 2

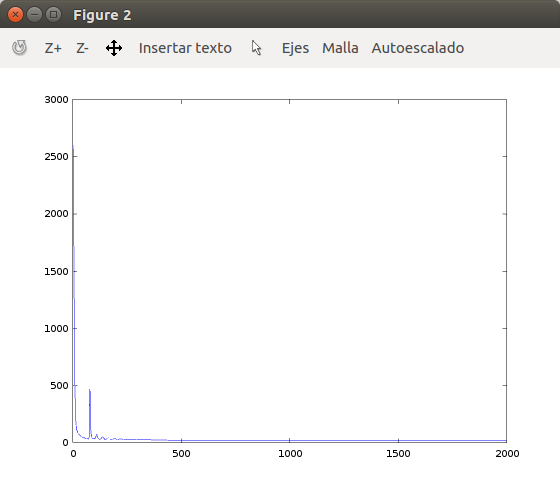
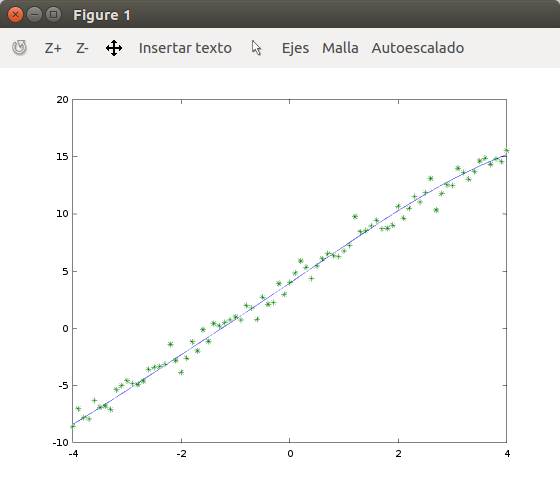


Figura 1: Usando nm 5, eta 0.1

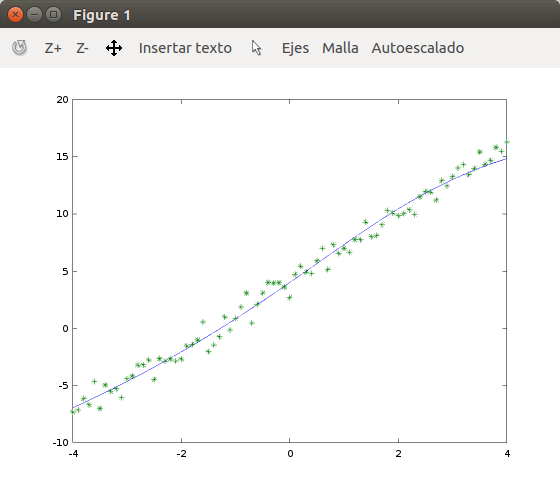
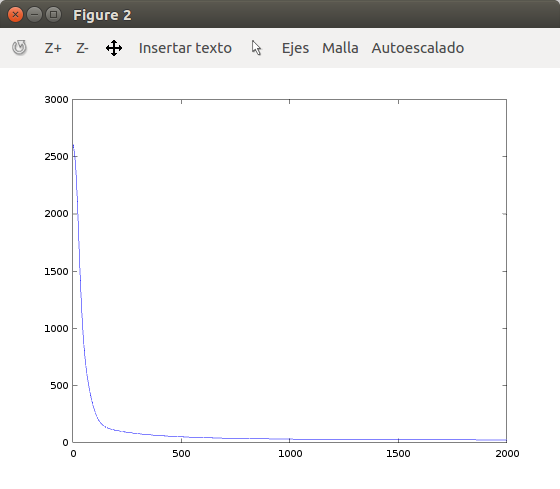
 

Figura 2: Usando nm 5, eta 0.01

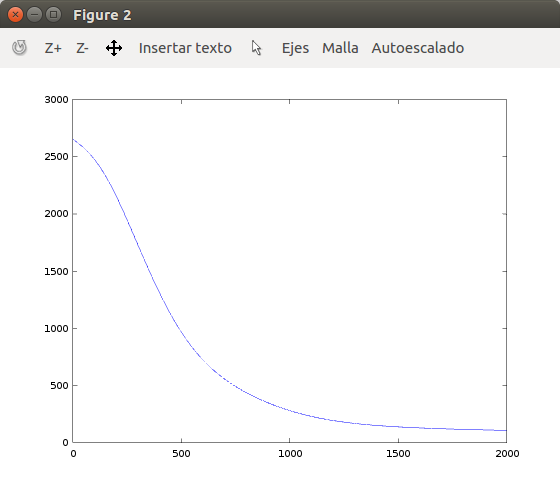
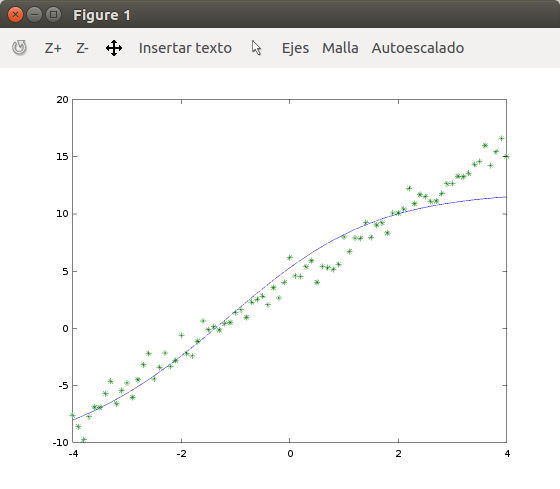


Figura 3: Usando nm 5, eta 0.001

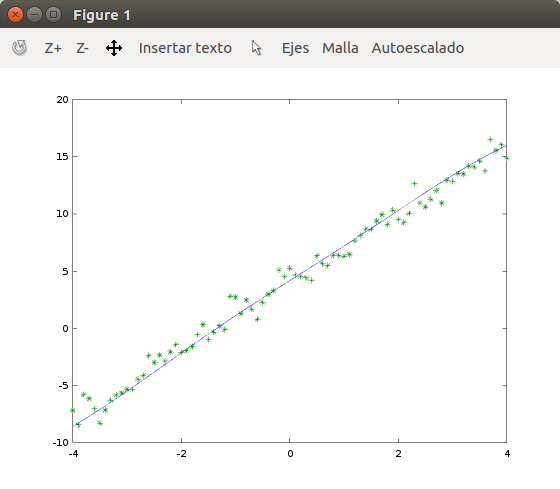
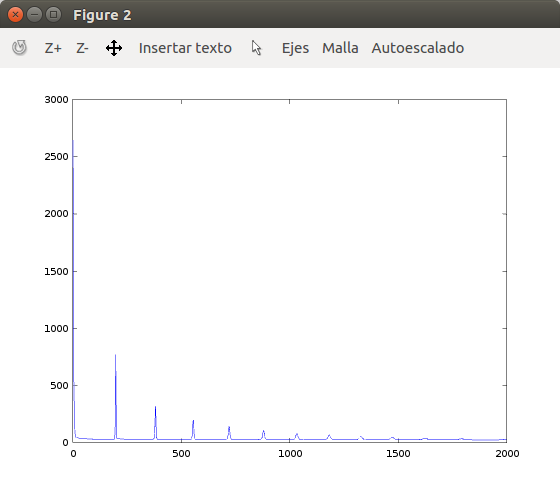
 

Figura 4: Usando nm 25, eta 0.1

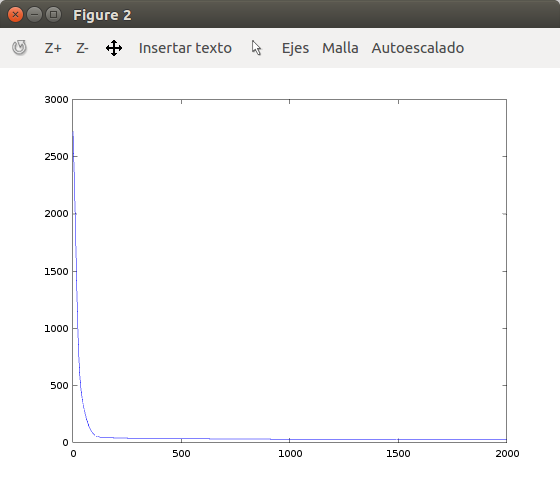
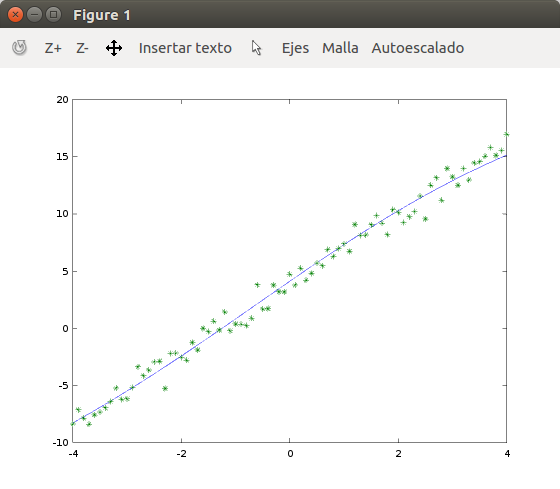


Figura 5: Usando nm 25, eta 0.01

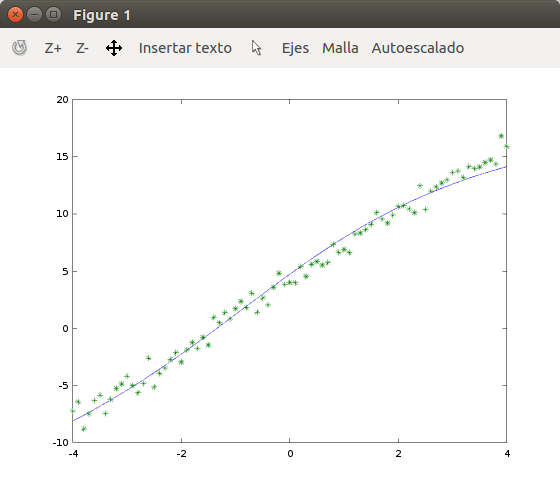
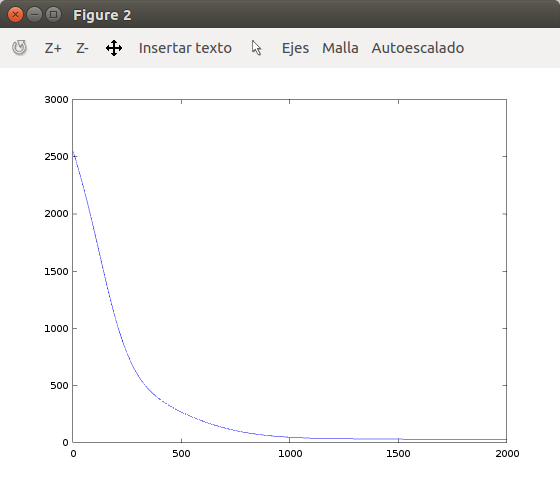
 

Figura 6: Usando nm 25, eta 0.001

### Funci´on Gaussiana

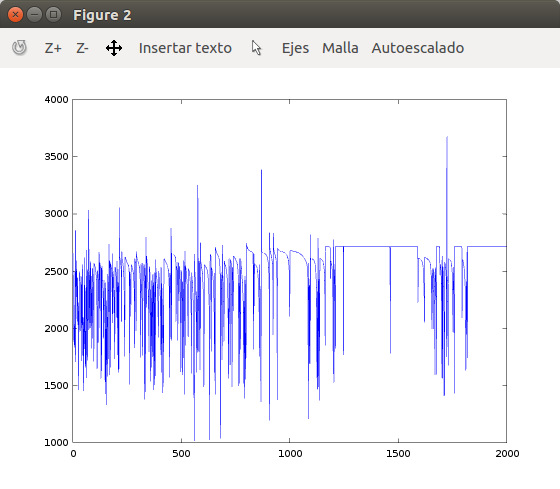
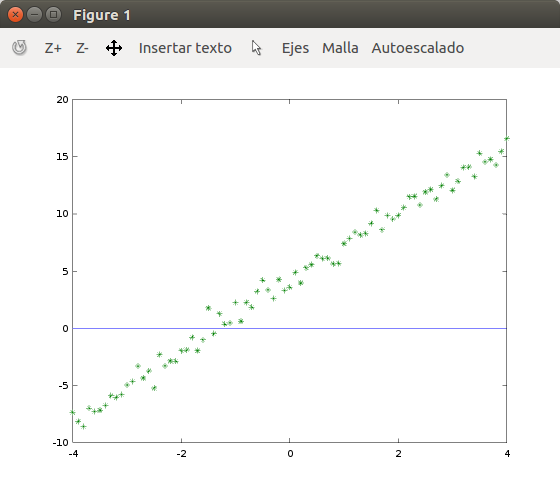


Figura 7: Usando nm 5, eta 0.1

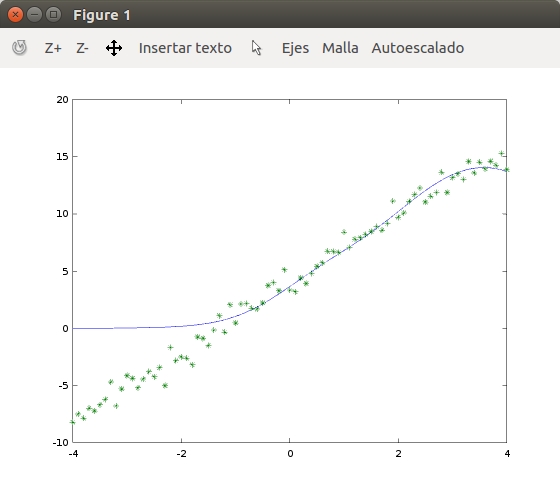
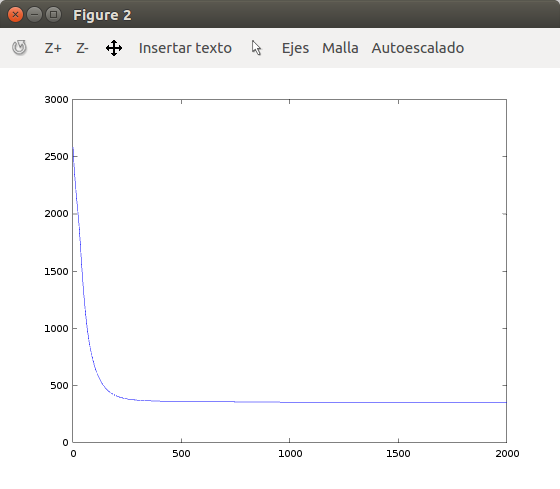
 

Figura 8: Usando nm 5, eta 0.01

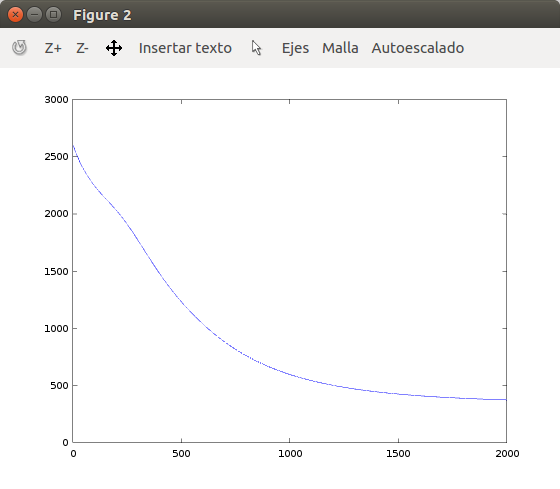
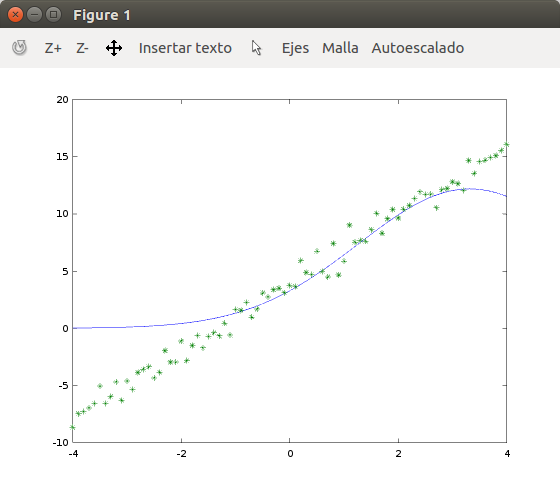


Figura 9: Usando nm 5, eta 0.001

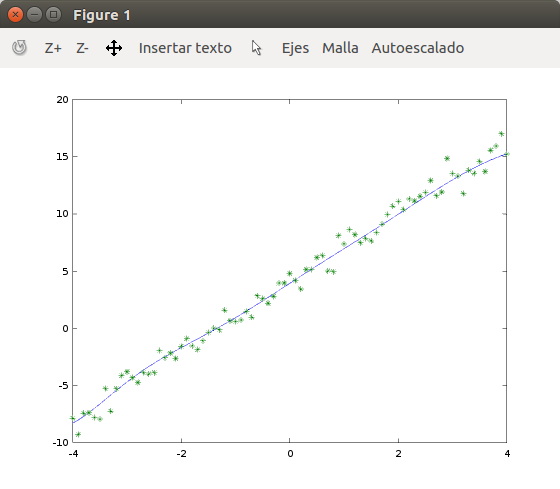
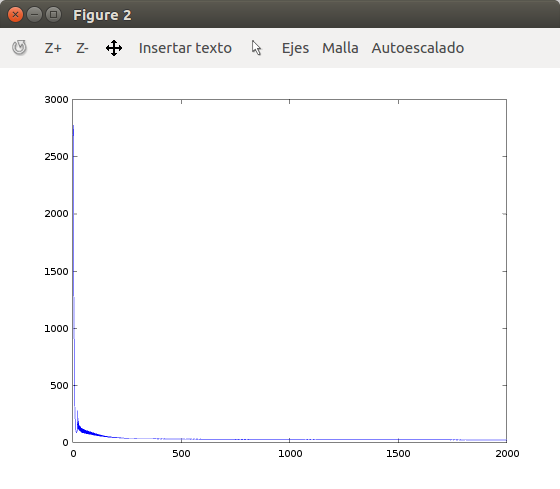
 

Figura 10: Usando nm 25, eta 0.1

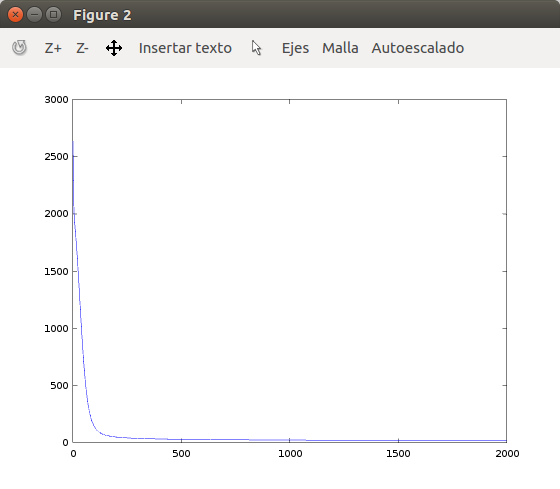
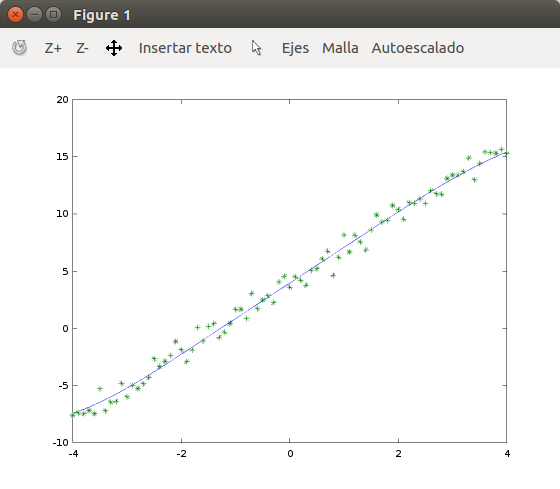


Figura 11: Usando nm 25, eta 0.01

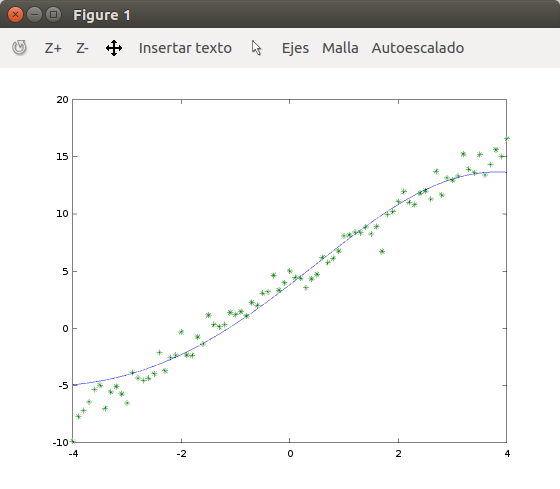
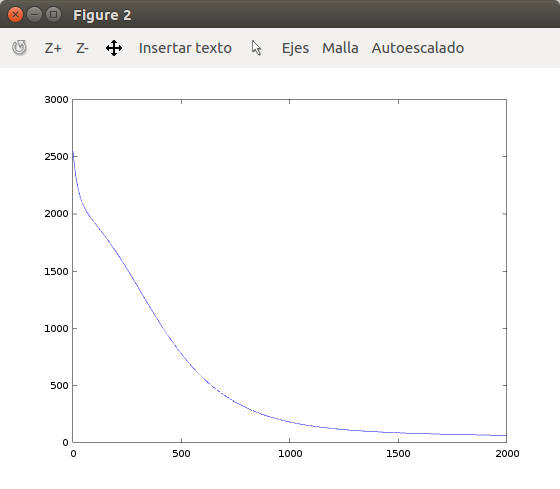
 

Figura 12: Usando nm 25, eta 0.001

## 1.2. Entrenamiento Patron

*% E n t re n a m ie n t o Patron con b i a s*

**c l e a r** ;

**c l c** ;

**c l o s e a l l** ;

**disp** ( ’ H e l l o ’ ) ;

a = 1 ; b = 2 ;

x = 2 : 0 . 0 5 : 3 ;

−

x = x ’ ;

N = **length** ( x ) ;

yb = a∗x + b + 0 . 2 ∗ **randn** (N, 1 ) ;

ne = 1 ; nm = 5 ;

b i a s = **input** ( ’ B i as : S I = 1 : ’ ) ;

**i f** ( b i a s == 1 ) ne = ne +1;

x = [ x **ones** (N, 1 ) ] ;

**end**

v = 0 . 2 5 ∗ **randn** ( ne ,nm ) ; w = 0 . 2 5 ∗ **randn** (nm, 1 ) ;

J o l d = 1 e15 ;

e t a = **input** ( ’ e t a : ’ ) ;

**f o r** i t e r = 1 : 5 0 0 0

w11 ( i t e r , 1 ) = w ( 1 , 1 ) ; dJdv = 0 ; dJdw = 0 ; **f o r** k = 1 :N

i n = ( x ( k , : ) ) ’ ; m = v ’ i n ;

∗

*%n = 1 . 0 . / ( 1 + e x p ( m) ) ; % S ig m o id e a 1*

−

n = 2 . 0 . / ( 1 + **exp**( m) ) 1 ; *% S ig m o id e a 2*

− −

*% n = e x p ( m. ˆ 2 ) ; % Gaussiana*

−

out = w’ n ;

∗

y ( k , 1 ) = out ;

e r = out yb ( k , 1 ) ;

−

**error** ( k , 1 ) = e r ;

*%dndm = n.*∗*(1* − *n ) ; % S ig m o id e a 1*

dndm = ( 1 − n . ∗ n ) / 2 ; *% S ig m o id e a 2*

*% dndm =* −*2.0*∗*(n .* ∗*m) ; % Gaussiana*

dJdw = 0∗dJdw + e r . ∗ n ;

dJdv = 0∗ dJdv + e r . ∗ i n ∗ (w. ∗ dndm ) ’ ; w = w − e t a ∗dJdw ;

v = v − e t a ∗ dJdv ;

**end**

*% w = w e t a dJdw/N;*

− ∗

*% v = v e t a dJdv /N;*

− ∗

JJ = 0 . 5 **sum**( **error** . **error** ) dJ = **abs** ( JJ J o l d ) ;

−

∗ ∗

dJpor = **sqrt** ( dJ/ JJ ) 1 0 0 ;

∗

**i f** ( dJpor *<* 0 . 7 5 ) *% P o r c e n t u a l*

**break** ;

**end**

**end**

J ( i t e r , 1 ) = JJ ; J o l d = JJ ;

**f i g u r e** ( 1 ) ;

**plot** ( x ( : , 1 ) , y , x ( : , 1 ) , yb , ’ ’ ) ;

∗

**f i g u r e** ( 2 ) ;

**subplot** ( 2 , 1 , 1 ) ; **plot** ( J ) ;

**subplot** ( 2 , 1 , 2 ) ; **plot** ( w11 ) ;

### Funci´on Sigmoidea 2

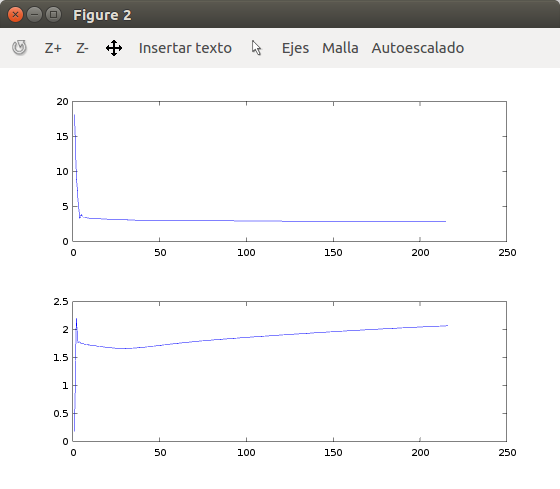
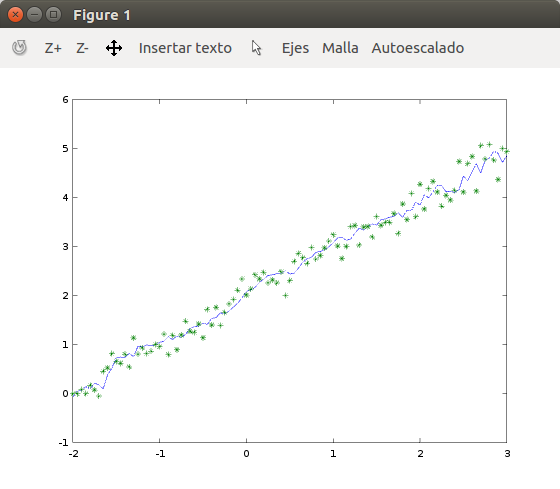


Figura 13: Usando nm 5, eta 0.1

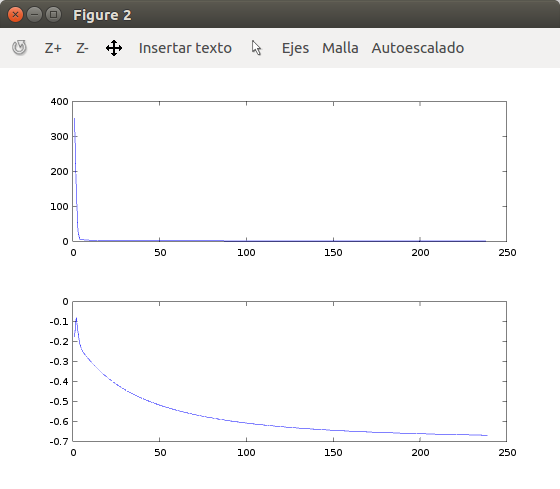
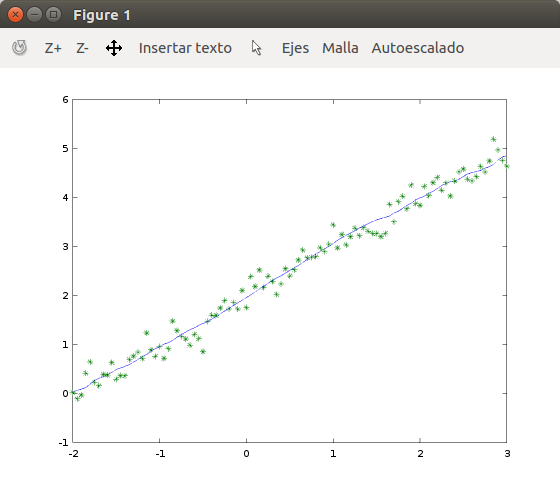


Figura 14: Usando nm 5, eta 0.01

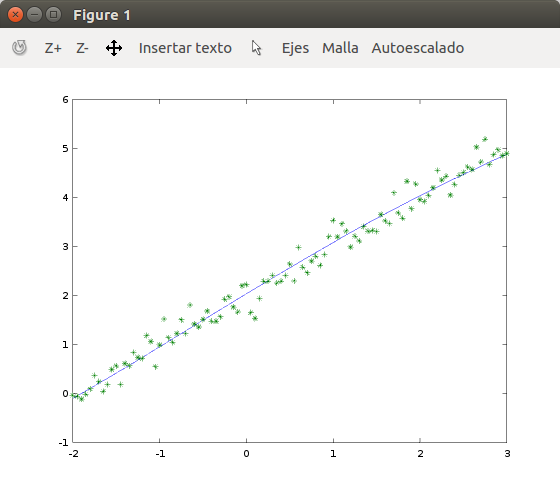
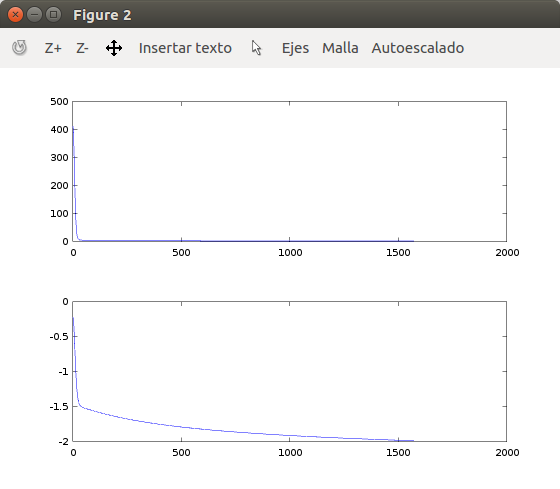
 

Figura 15: Usando nm 5, eta 0.001

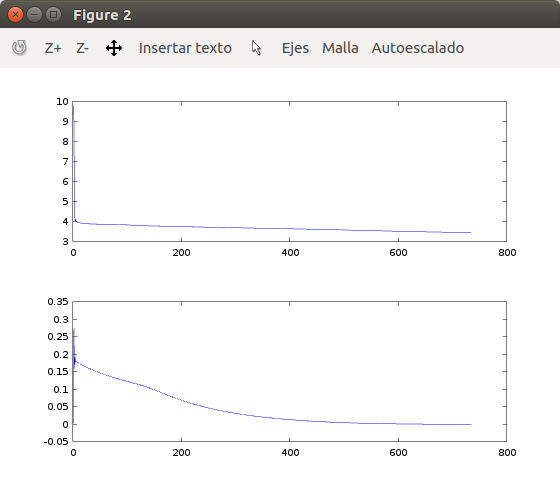
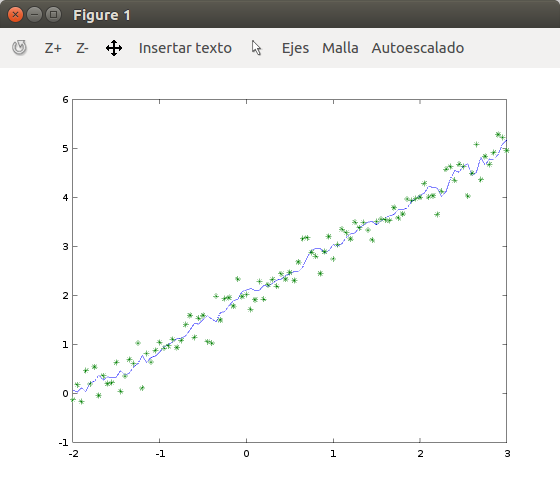


Figura 16: Usando nm 25, eta 0.1

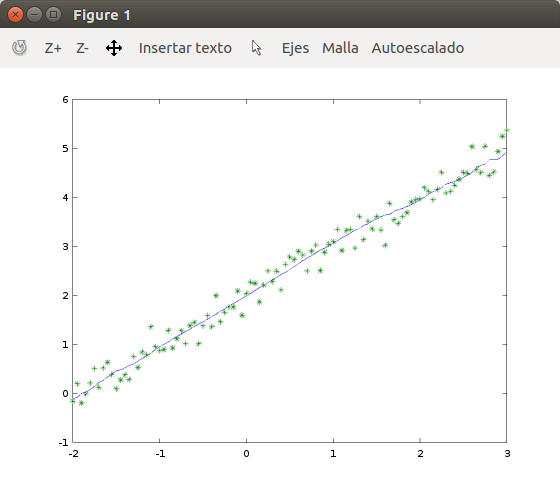
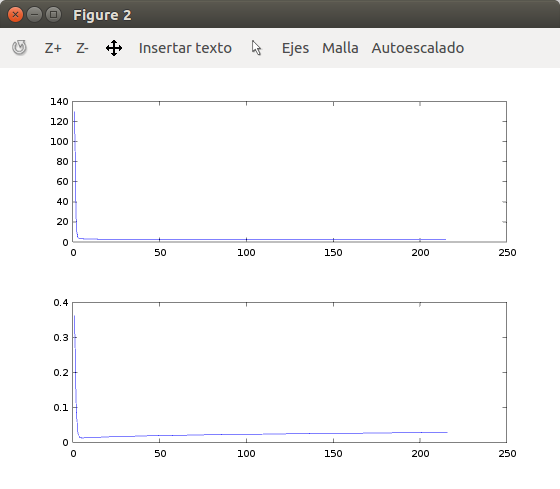
 

Figura 17: Usando nm 25, eta 0.01

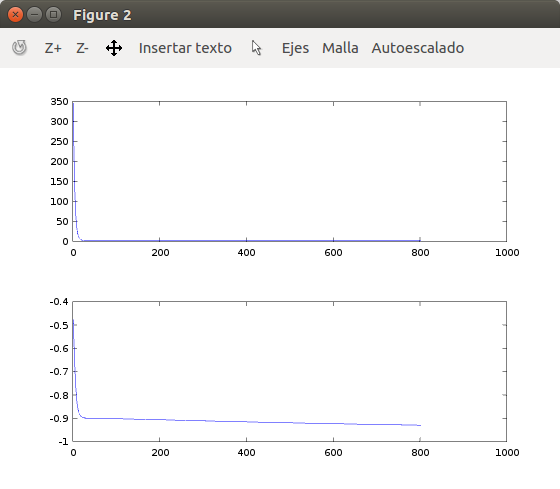
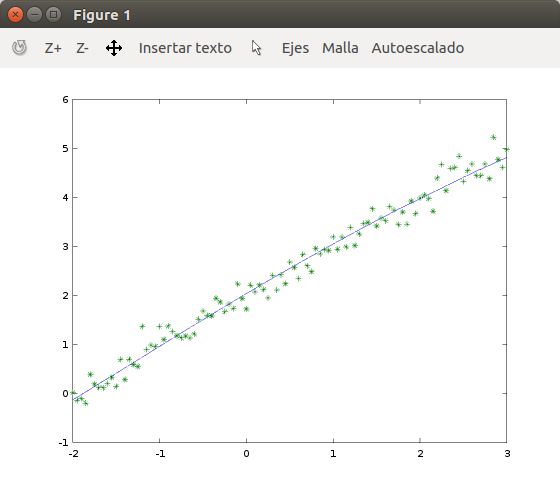


Figura 18: Usando nm 25, eta 0.001

### Funci´on Gaussiana

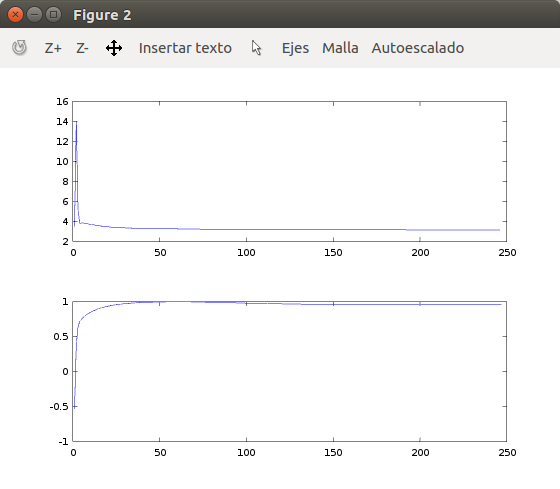
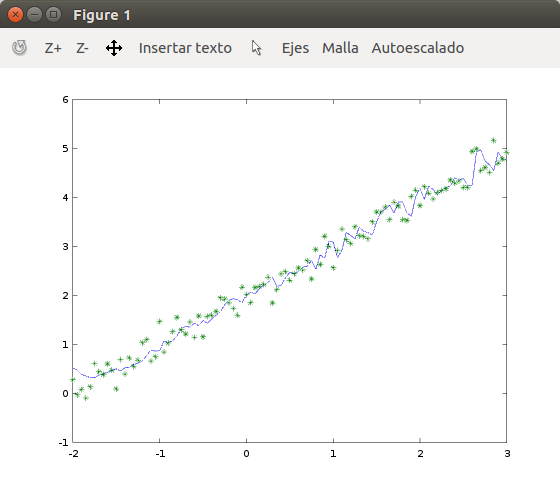


Figura 19: Usando nm 5, eta 0.1

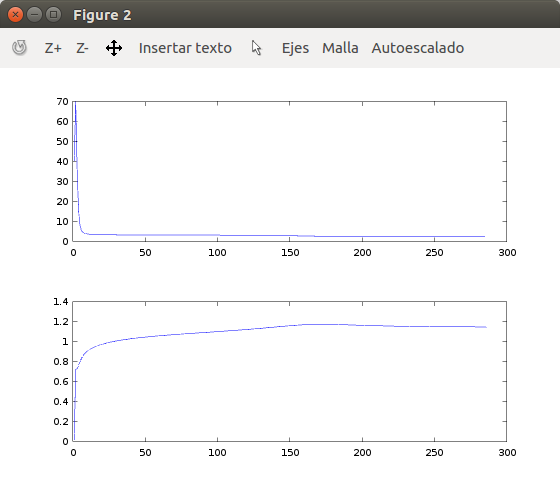
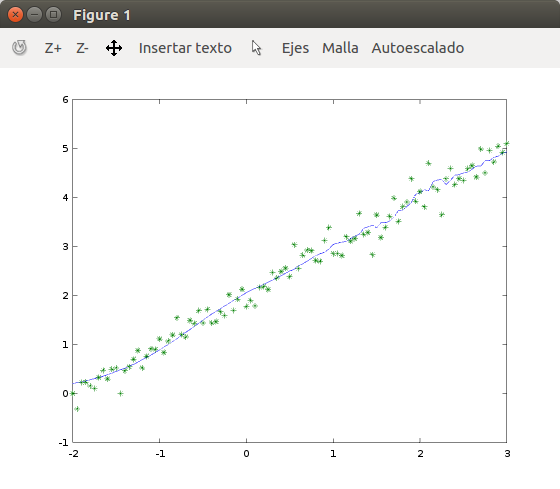


Figura 20: Usando nm 5, eta 0.01

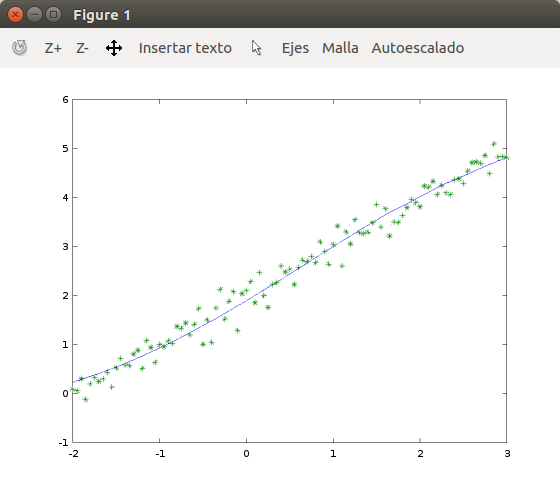
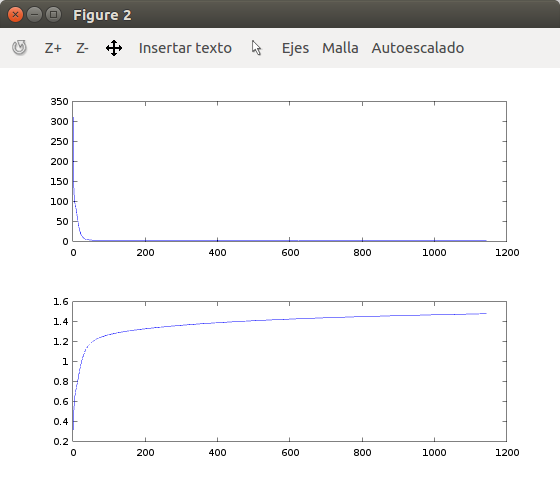
 

Figura 21: Usando nm 5, eta 0.001

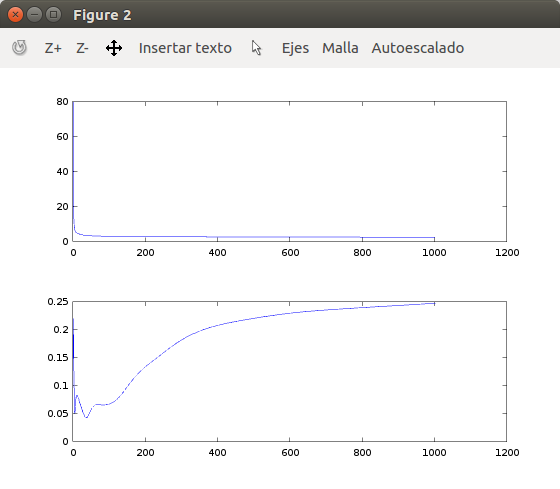
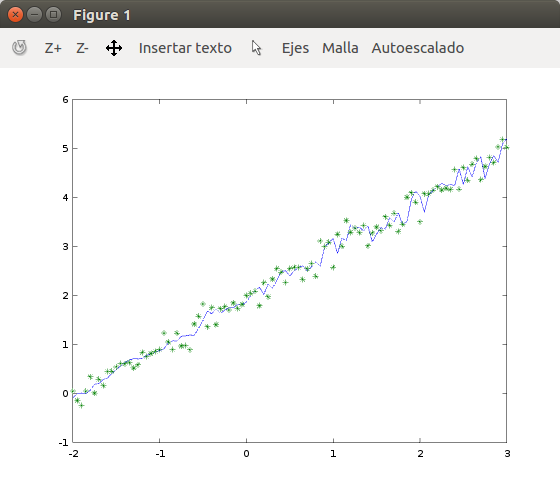


Figura 22: Usando nm 25, eta 0.1

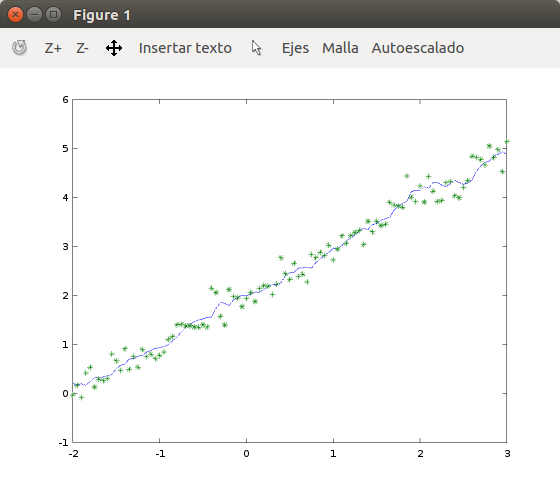
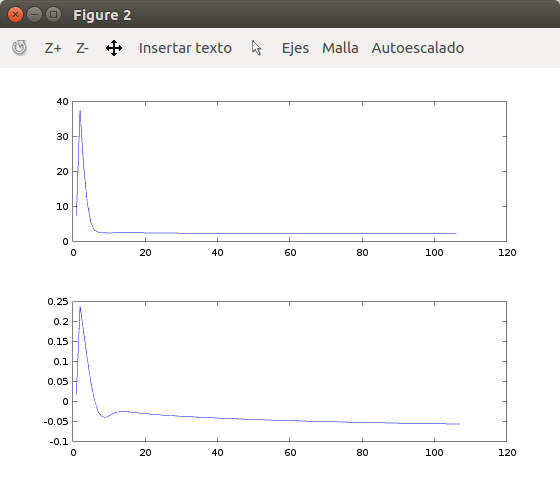
 

Figura 23: Usando nm 25, eta 0.01

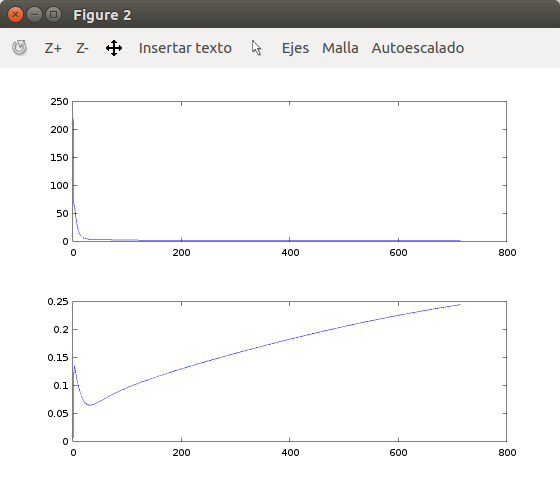
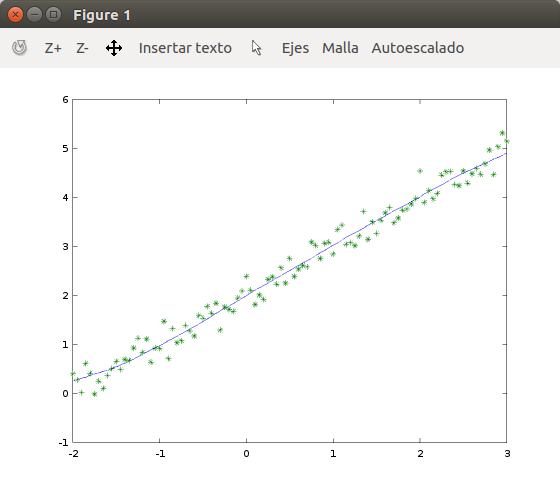


Figura 24: Usando nm 25, eta 0.001