



## **INFORME TRABAJO PRÁCTICO ESPECIAL Nº2**

**Nivel y Área:**

72.27 – Sistemas de Inteligencia Artificial

1º Cuatrimestre 2019

**Comisión: S - Carrera: Informática**

**Fecha:** 13 de mayo de 2019

**Alumnos Expositores:**

AQUILI, Alejo Ezequiel (57432)

BASSANI, Santiago (57435)

SANGUINETI ARENA, Francisco Javier (57565)

## Informe: Trabajo Práctico Especial Nº2 – Redes Neuronales

### Introducción

Una empresa de videojuegos precisa un desarrollo que pueda simular en su plataforma terrenos de diferentes partes del mundo a partir de mediciones de latitud, longitud y altura, obtenidas de terrenos reales.

Para llevar a cabo el desarrollo se implementó una red neuronal multicapa “Feed-forward” con aprendizaje supervisado, utilizando el algoritmo de backpropagation con gradiente descendiente y mejoras sobre el mismo (momentum, eta adaptativo y weight decay), para aproximar la función de altura del terreno a simular.

De las muestras antes mencionadas se obtiene un subconjunto aleatorio de patrones de entrenamiento y de testeo para llevar adelante el entrenamiento de tipo batch de la red neuronal.

### Decisiones Importantes

En cuanto a la selección de patrones, decidió tomar los subconjuntos aleatorios(A y B) de valores de la muestra original ( $M$ ,  $M = A \cup B$ ) para generar los conjuntos de patrones de entrenamiento (A) y de testeo (B) en base a un porcentaje indicado. Para realizar la generación de los subconjuntos aleatorios se tomo una permutación aleatoria del numero de elementos de la muestra original.

Por otra parte, se decidió que para ingresar patrones de entrenamiento y de prueba utilizando puntos representativos para estos conjuntos, se haga esto mediante archivos, especificando en las configuraciones las rutas a dichos archivos.

Se decidió excluir del análisis y pruebas realizadas en este informe la funcionalidad extra implementada como son la función de activación RELU y el algoritmo de Adam como posible mejora sobre el algoritmo de backpropagation.

Finalmente se opto por usar preferentemente en las pruebas realizadas inicialización de pesos aleatoria y no FAN IN, si bien esta última es un parámetro elegible en nuestra configuración y también se utilizó como función en la capa de salida, la función lineal.

## Mejoras al algoritmo de Backpropagation

Si la tasa o velocidad de aprendizaje es muy alta, el entrenamiento progresaría con mayor velocidad, si se opta por un learning rate muy alto, el algoritmo de backpropagation podría divergir, mientras que, si se opta por un learning rate muy bajo, si bien converge hacia una solución, lo hace con mayor lentitud. Para optimizar esta situación se probaron las siguientes funciones:

Momentun: la primera mejora realizada sobre backpropagation con gradiente descendiente fue sumarle en la expresión de corrección de pesos un coeficiente llamado alfa o momentum (0.9 por defecto) que se multiplica por la corrección de los pesos de la época anterior. El objetivo de este término es homogenizar las oscilaciones en la minimización del error otorgando mayor importancia a la dirección de la pendiente. Esto resulta ideal para las situaciones donde el gradiente converge en forma lenta o las oscilaciones son considerablemente grandes.

ETA Adaptativo: esta estrategia consiste en utilizar un learning rate dinámico con el fin de optimizar el porcentaje del error que se utiliza en cada corrección de los pesos en la red. Tiene en cuenta la tendencia de crecimiento del error en las últimas K iteraciones. Su utilización ayuda a bajar el número de iteraciones necesarias y a desacoplar el resultado del aprendizaje con respecto a la inicialización del learning rate.

Weight Decay: esta estrategia consiste en penalizar el cambio de los pesos que conectan capas consecutivas derivando una tendencia de los pesos a un valor cercano a cero. Esto se utiliza para evitar el efecto de “overfitting” o memorización, como así también limitar las direcciones libres en las que se pueda mover el error. Dicha optimización tiene origen en la adición de un término ( $\frac{\lambda}{2} W^2$ ) a la función de costo que actúa como un término gaussiano con media cero.

## Definición de la Arquitectura

### Pruebas Realizadas

Se realizaron pruebas sobre distintas configuraciones o arquitecturas con distinta cantidad de neuronas por capa oculta y con distinta cantidad de capas ocultas, también se variaron los parámetros de la red como pueden ser la función de activación (tanh o exponencial), la utilización o no de las mejoras implementadas sobre backpropagation con gradiente descendiente, el learning rate de la red, la inicialización de pesos y la modalidad BATCH o INCREMENTAL en algoritmo de aprendizaje.

### Resultados obtenidos

De todas las pruebas realizadas se incluyen las más relevantes adjuntando gráficos de cada una de las arquitecturas presentadas en el Anexo del informe. Los gráficos incluidos en el anexo por cada arquitectura son el error cuadrático medio en función de las épocas durante el entrenamiento, el learning rate en función de las épocas durante el entrenamiento, el error obtenido en el conjunto de prueba luego de finalizar el entrenamiento con la superficie de error de la aproximación acompañado un heatmap con los errores (no absolutos), la función aproximada con los puntos de entrenamiento y prueba indicados sobre la superficie

aproximada y el error cometido en la aproximación, también acompañado de un heatmap.

Las 9 arquitecturas presentadas a continuación coinciden en que han obtenido mejores resultados utilizando inicialización de pesos aleatoria y corriendo en modo incremental normalizando la entrada con la norma 2 del vector de entrada y esos resultados son los adjuntos. También se utilizaron conjuntos de entrenamiento aleatorios en base al 90% de la muestra original en todos los casos dado que este porcentaje presenta mayor generalización.

ARQUITECTURA	ECM	F.A	Optimización	ETA
1 [ 2 - 1 ]	0.055591	TANH	ETA ADAPTATIVO y MOMENTUM	0.05
2 [ 2 - 3 - 1 ]	0.010669	TANH	ETA ADAPTATIVO y MOMENTUM	0.05
3 [ 2 - 15 - 1 ]	0.004429	EXP	ETA ADAPTATIVO y MOMENTUM	0.05
4 [ 2 - 25 - 15 - 1 ]	0.001238	TANH	SGD y MOMENTUM	0.005
5 [ 2 - 45 - 50 - 1 ]	0.007261	TANH	SGD y MOMENTUM	0.005
6 [ 2 - 45 - 50 - 1 ]	0.001563	EXP	SGD y MOMENTUM	0.005
7 [ 2 - 45 - 50 - 1 ]	0.001040	EXP	ETA ADAPTATIVO y MOMENTUM	0.05
8 [ 2 - 30 - 10 - 1 ]	0.000979	EXP	SGD y MOMENTUM	0.005
9 [ 2 - 30 - 10 - 1 ]	0.001781	EXP	ETA ADAPTATIVO y MOMENTUM	0.05

PARAMETROS FIJOS: Cantidad máxima de épocas: 200, Coeficiente de momentum: 0.9, Parámetros adaptativos a = 0.00003 b = 0.002 y K = 50.

Tabla 1: Tabla de las arquitecturas candidatas con cantidad máxima de épocas fija.

ARQUITECTURA	ÉPOCAS	F.A	Optimización	ETA	ECM
[ 2 - 15 - 1 ]	101	EXP	ETA ADAPTATIVO y MOMENTUM	0.05	0.009740
[ 2 - 25 - 15 - 1 ]	56	TANH	SGD y MOMENTUM	0.005	0.009810
[ 2 - 45 - 50 - 1 ]	33	EXP	SGD y MOMENTUM	0.005	0.009577
[ 2 - 30 - 10 - 1 ]	18	EXP	SGD y MOMENTUM	0.005	0.009846
[ 2 - 30 - 10 - 1 ]	73	EXP	ETA ADAPTATIVO y MOMENTUM	0.05	0.009917

PARAMETROS FIJOS: ECM <=0.01, Coeficiente de momentum: 0.9, Parámetros adaptativos a = 0.00003 b = 0.002 y K = 50.

Tabla 2: Tabla de las arquitecturas con cota del error fija.

Se eligió la arquitectura 8 de la tabla 1, dado que fue la que aprendió mejor porque dado la restricción en la cantidad de épocas fue la que mejor generalizó el problema lo cual se puede plasmar en el error cuadrático medio obtenido y en el caso en que busco alcanzar la cota de error fija fue la arquitectura que en menos cantidad de épocas lo logró. Esta arquitectura se diseño siguiendo los lineamientos del algoritmo de Tilling visto en clase, donde la cantidad de neuronas por capa oculta disminuye en medida que la cantidad de capas ocultas aumenta. Además, se midió el nivel de generalización de esta arquitectura, probando con diferentes porcentajes del conjunto de entrenamiento y se obtuvo con conjuntos de hasta el 20% (90% en las pruebas de las anteriores tablas) de la muestra original para el conjunto de patrones de entrenamiento conservando un gran poder de generalización.

## Conclusiones

Se pudo implementar una red neuronal multicapa capaz de aproximar la función altitud de un terreno dadas la longitud y latitud. Para su entrenamiento se utilizó el algoritmo de backpropagation y sus mejoras.

Entrenar a la red con una única variante de backpropagation puede ser contraproducente. Hay ciertas instancias que queda atrapada en un mínimo local y es necesario utilizar optimizaciones como momentum, parámetros adaptativos o manualmente aumentar el learning rate en algunas épocas hasta que logre salir del mismo. Así mismo, notamos que la arquitectura elegida tiene un papel importante en el nivel de detalle que logra aprender del terreno.

Además, queda claro que la selección de los patrones de aprendizaje es importante y que no con cualquier conjunto de valores se logra que la red llegue a un nivel de generalización satisfactorio. En el caso de estudio, el terreno presenta una diferencia de nivel importante en el centro la cual es necesaria incluirlo en el conjunto de aprendizaje para que se llegue a generalizar. La elección aleatoria de patrones de aprendizaje resultó ser una buena política de selección.

## Anexo – Gráficos de las arquitecturas candidatas

### Mediciones del terreno a aproximar

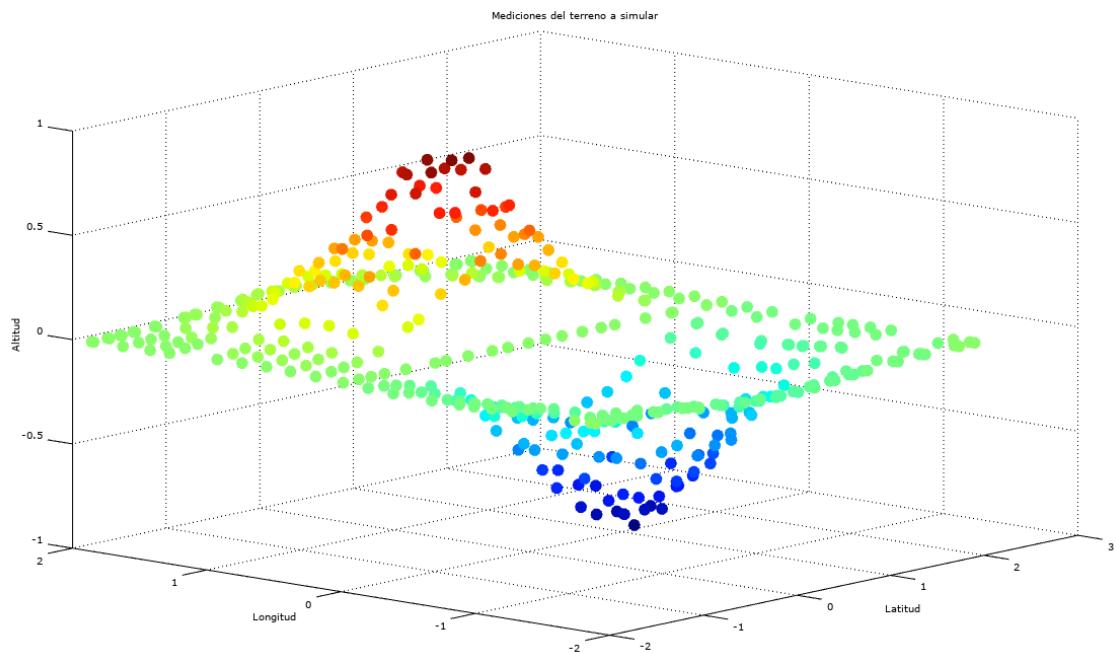
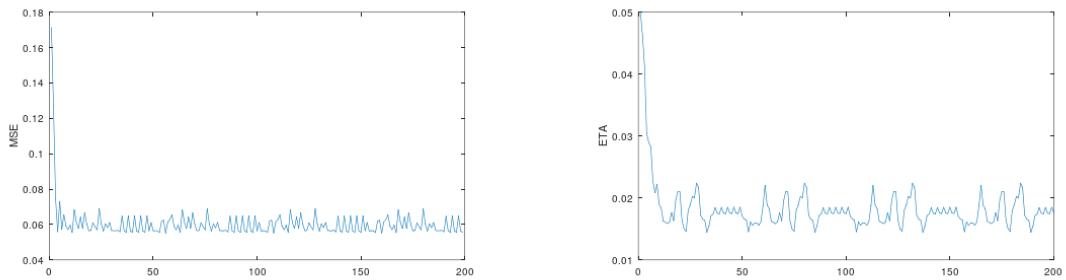


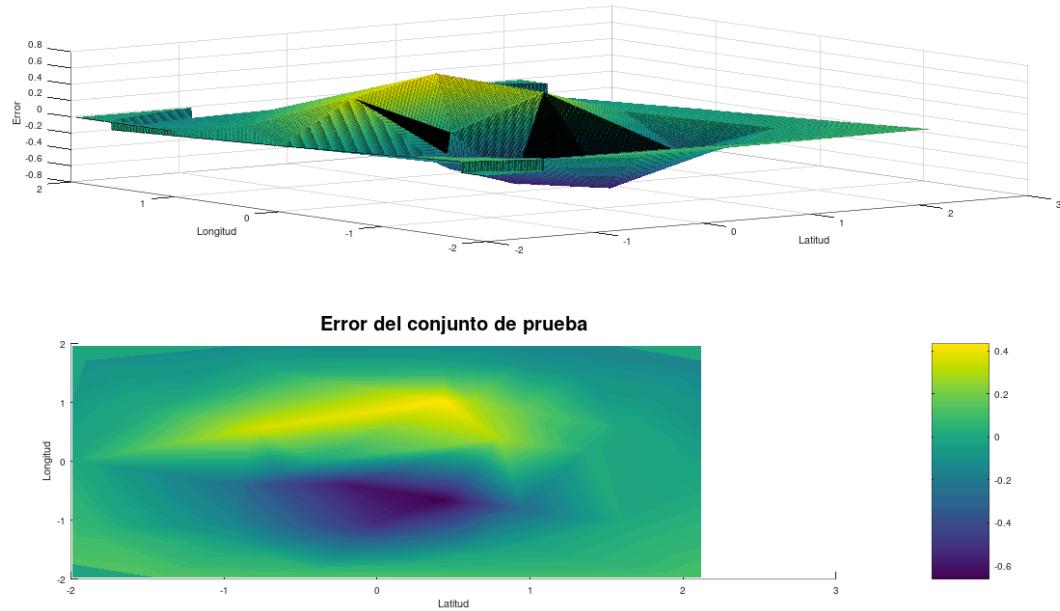
Figura 1 – Mediciones del terreno a aproximar

**Arquitectura 01: [ 2 - 1 ] (ECM = 0.055591) (TANH)**

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05, parámetros adaptativos a = 0.00003, b = 0.002 y K = 50.



*Figura 2 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento*



*Figura 3 – Error obtenido en el conjunto de prueba*

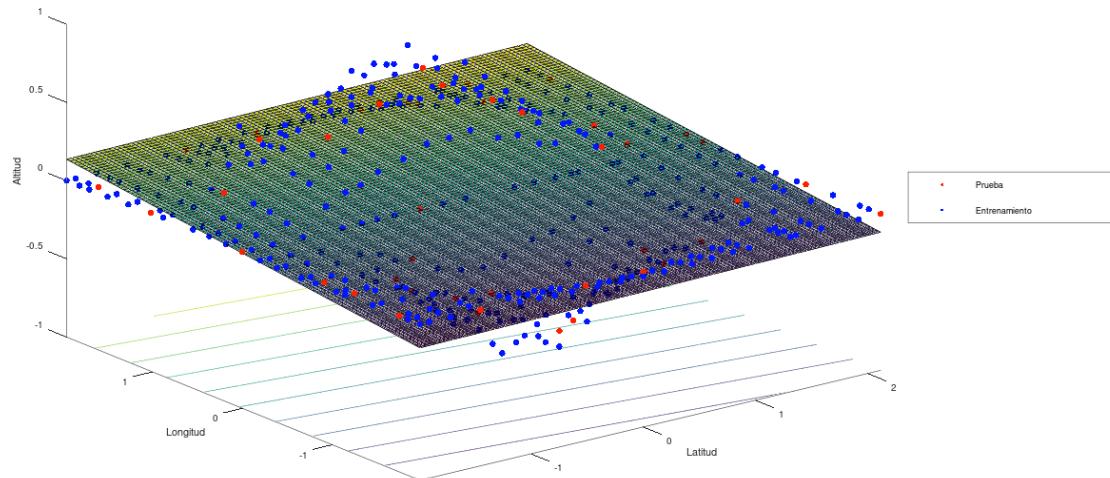
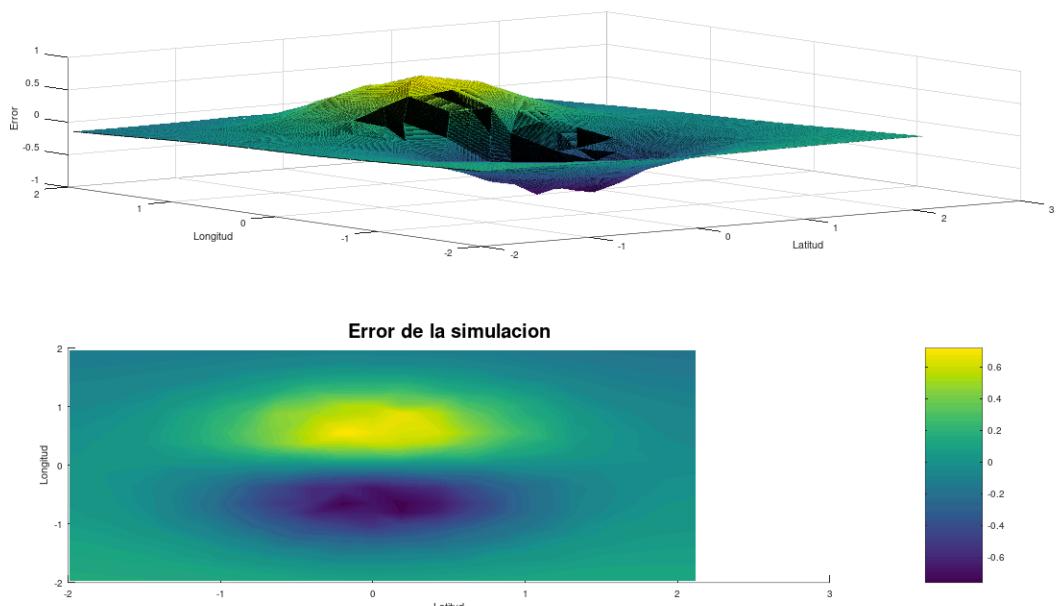
**Simulación y mediciones**

Figura 4 – Aproximación de la altitud

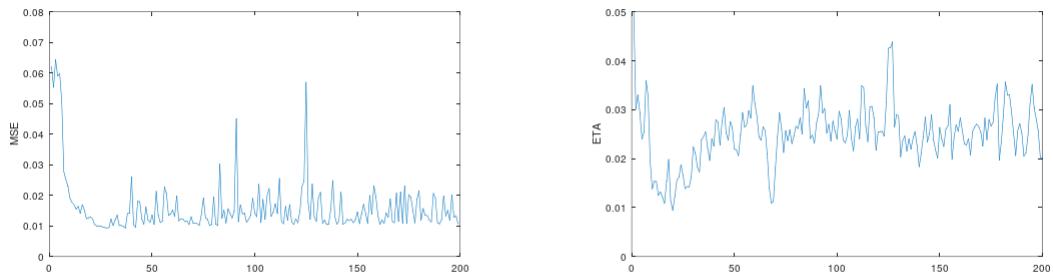


Error de la simulación

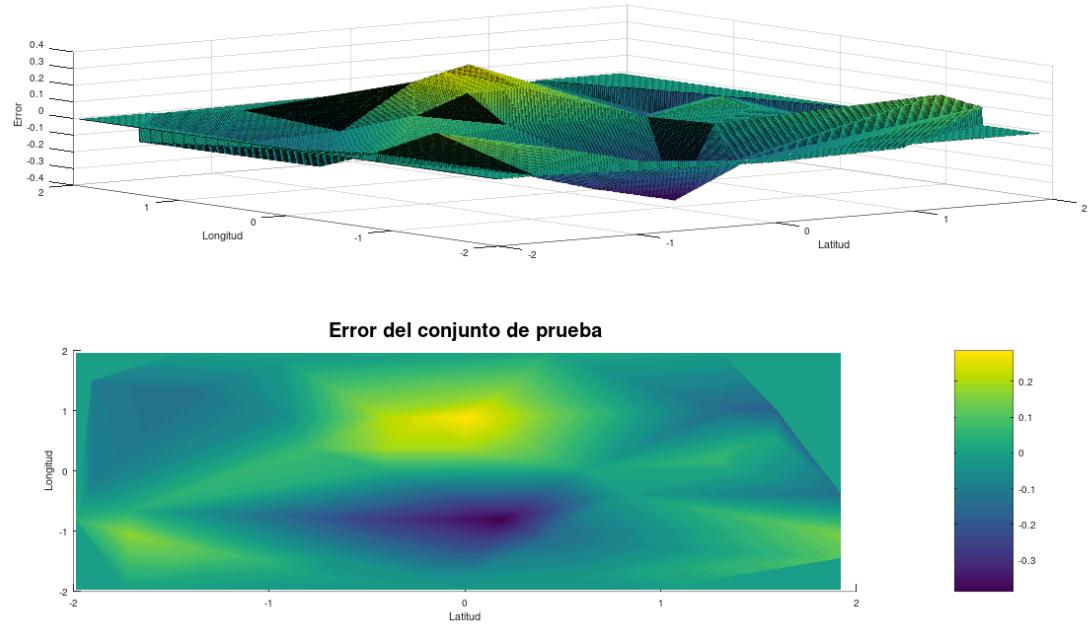
Figura 5 – Error obtenido en la aproximación

**Arquitectura 02: [ 2 - 3 - 1 ] (ECM = 0.010669) (TANH)**

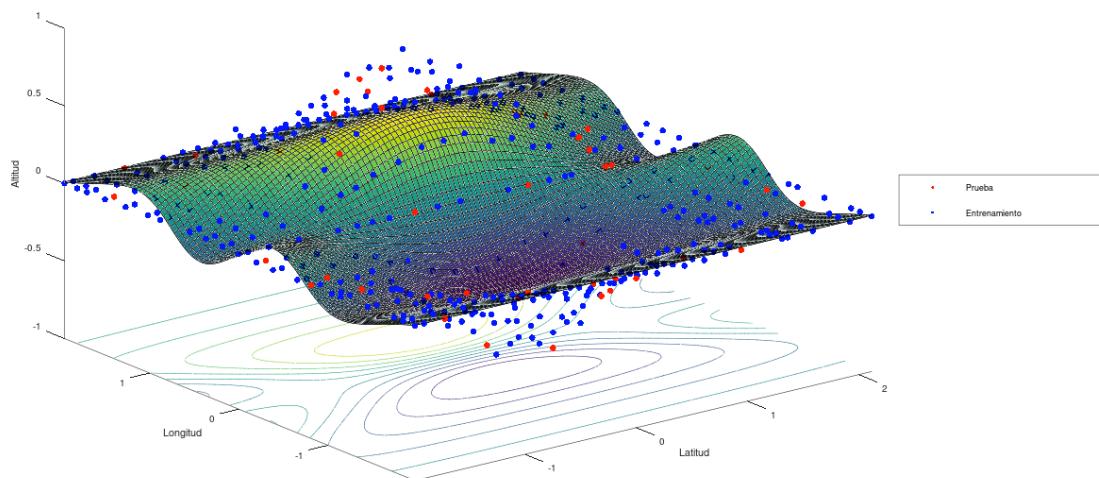
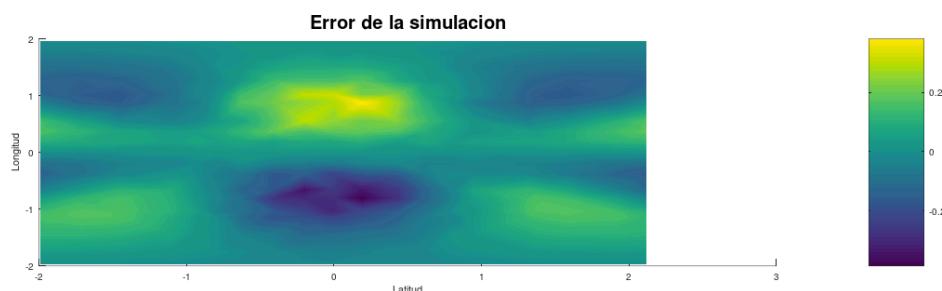
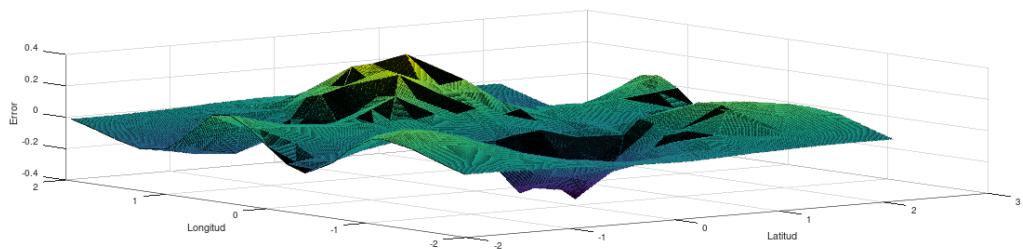
PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05, parámetros adaptativos a = 0.00003, b = 0.002 y K = 50.



*Figura 6 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento*

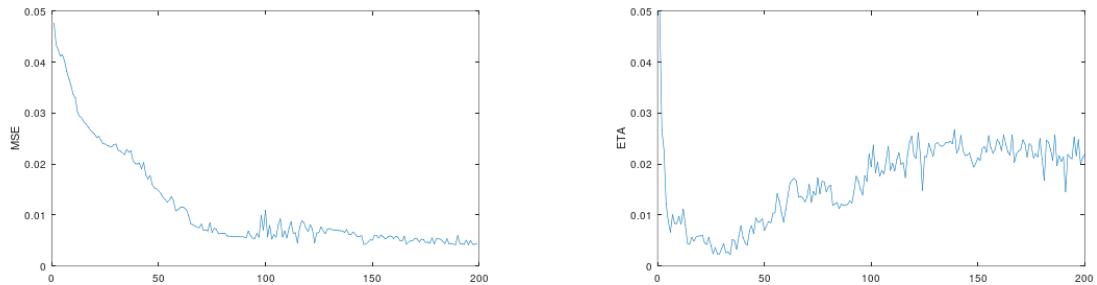


*Figura 7 – Error obtenido en el conjunto de prueba*

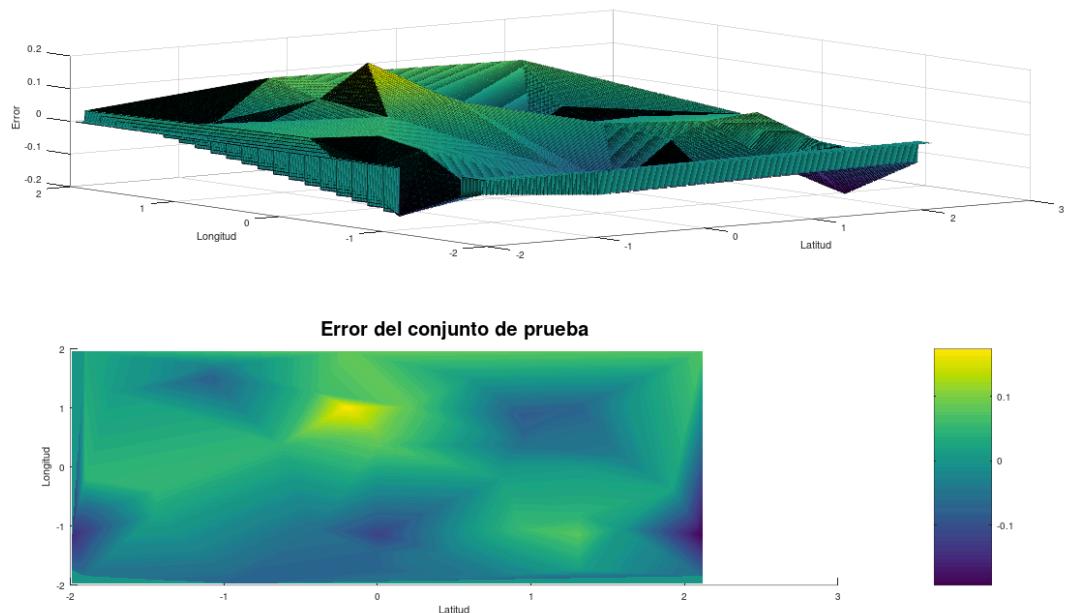
**Simulacion y mediciones***Figura 8 – Aproximación de la altitud**Figura 9 – Error obtenido en la aproximación*

### Arquitectura 03: [ 2 - 15 - 1 ] (ECM = 0.004429) (EXP)

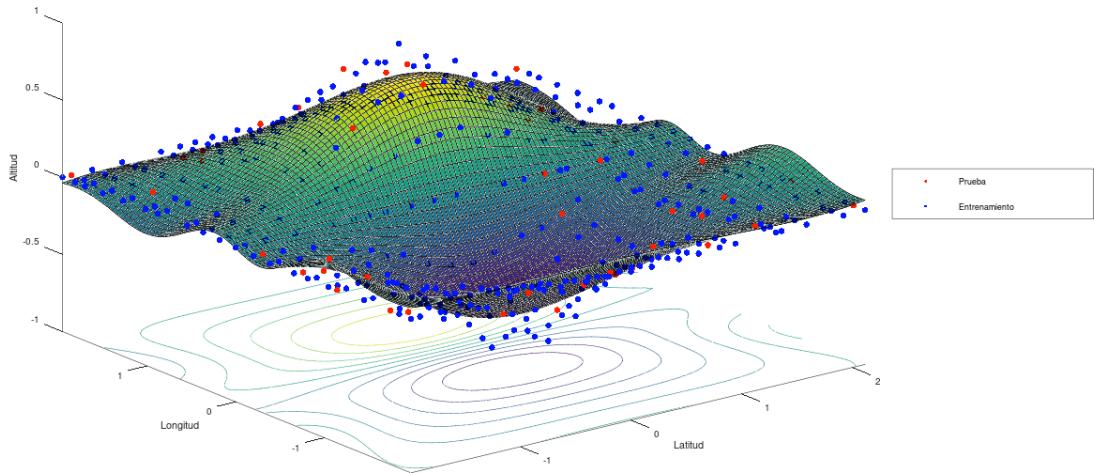
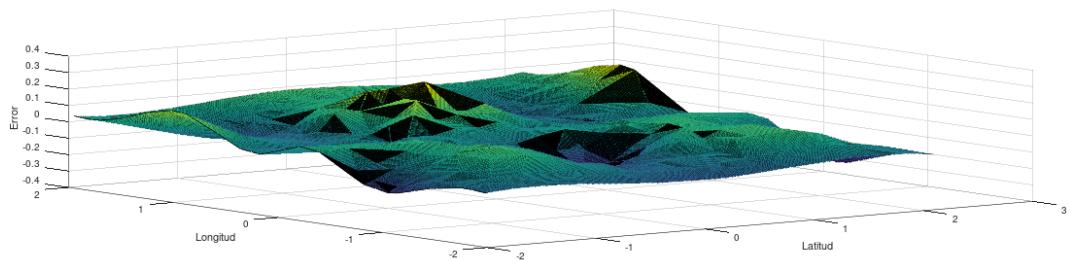
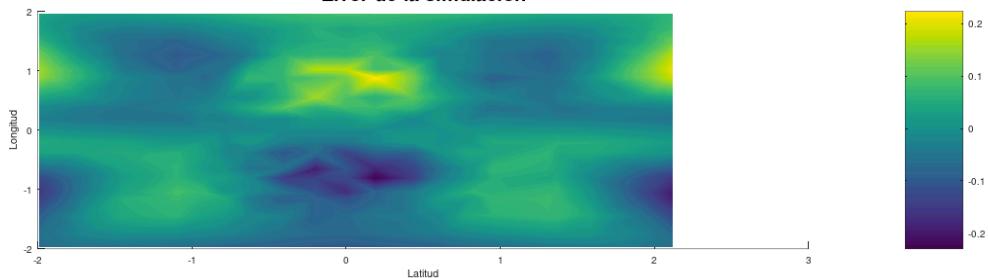
PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.005.



*Figura 10 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento*



*Figura 11 – Error obtenido en el conjunto de prueba*

**Simulacion y mediciones***Figura 12 – Aproximación de la altitud***Error de la simulacion***Figura 13 – Error obtenido en la aproximación*

### Arquitectura 04: [ 2 - 25 - 15 - 1 ] (ECM = 0.001238) (TANH)

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05, parámetros adaptativos a = 0.00003, b = 0.002 y K = 50.

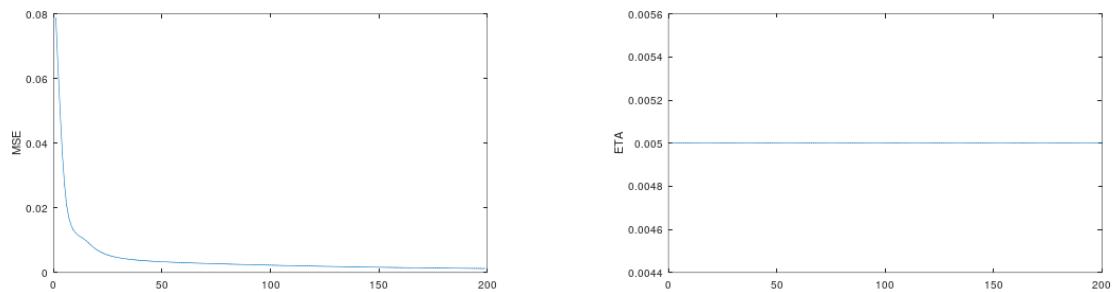


Figura 14 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

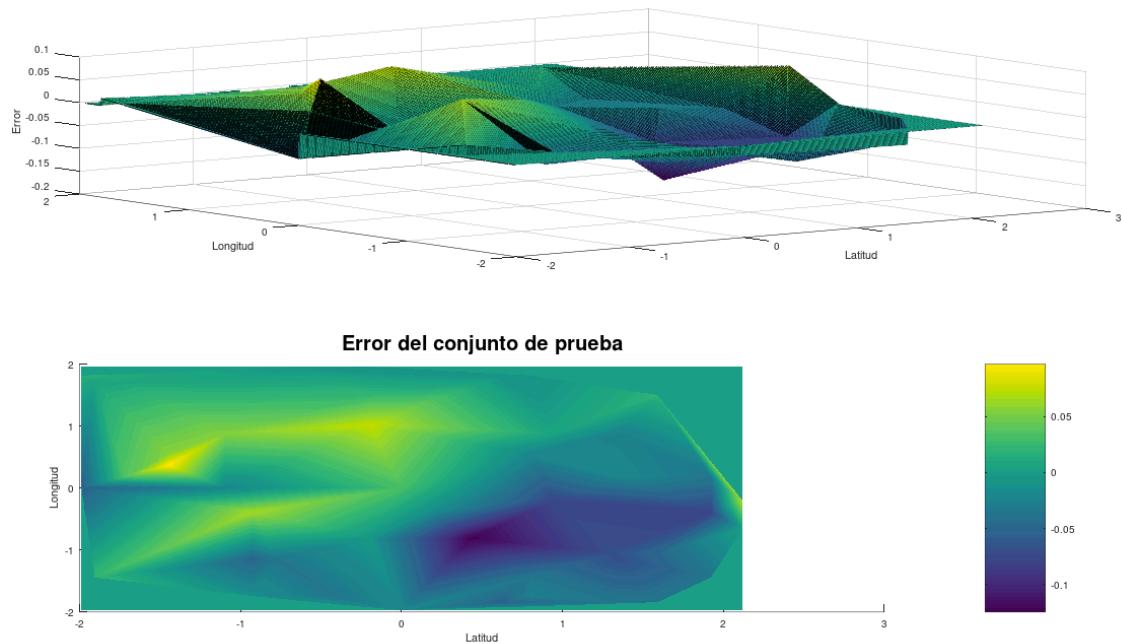
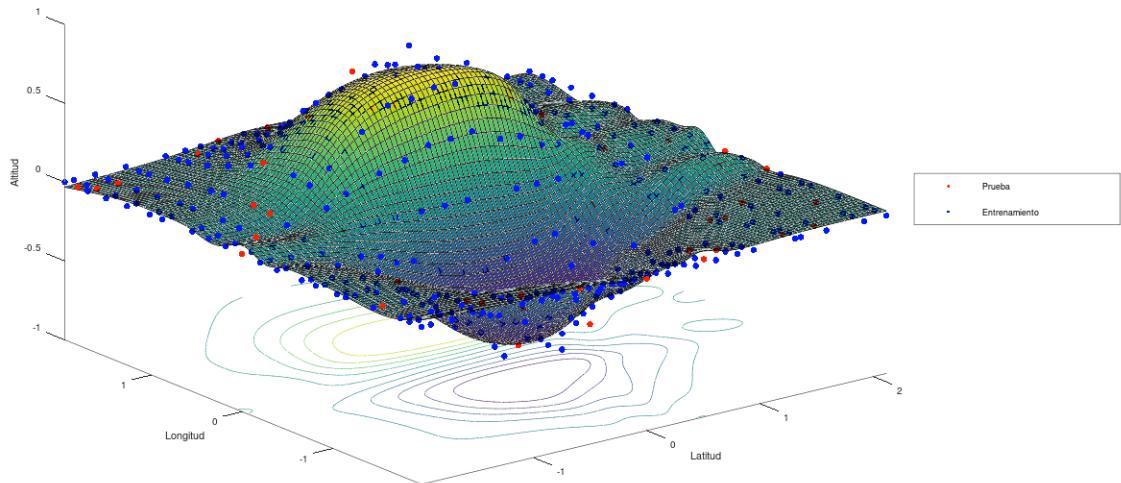
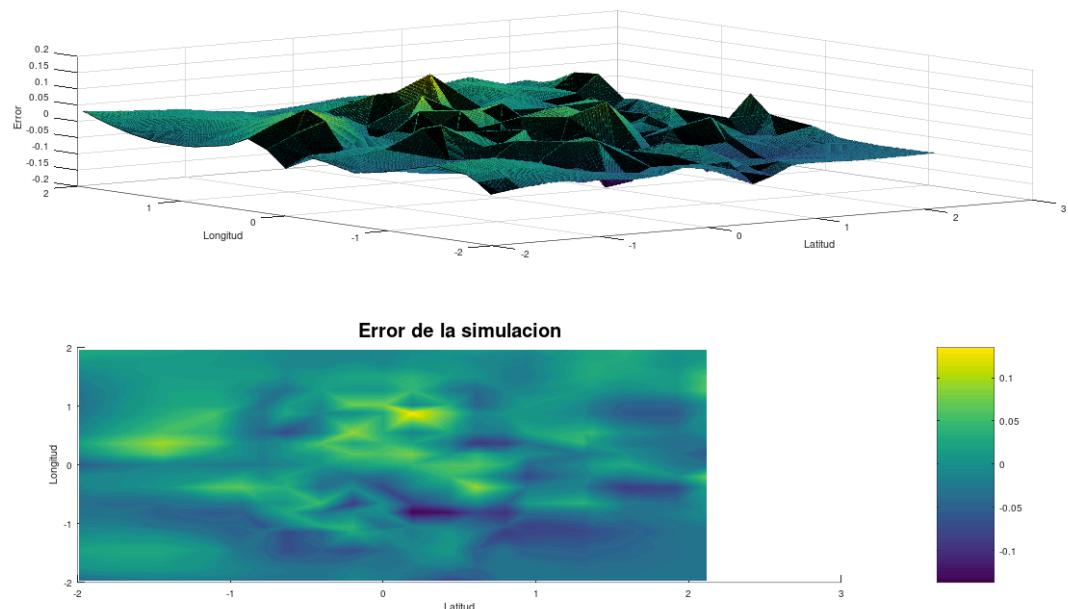


Figura 15 – Error obtenido en el conjunto de prueba

**Simulación y mediciones***Figura 16 – Aproximación de la altitud**Figura 17 – Error obtenido en la aproximación*

## Arquitectura 05: [ 2 - 45 - 50 - 1 ] (ECM = 0.007261) (TANH)

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05.

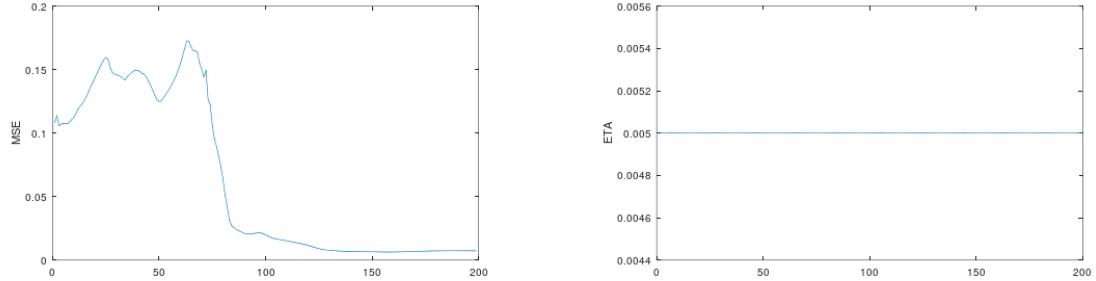


Figura 18 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

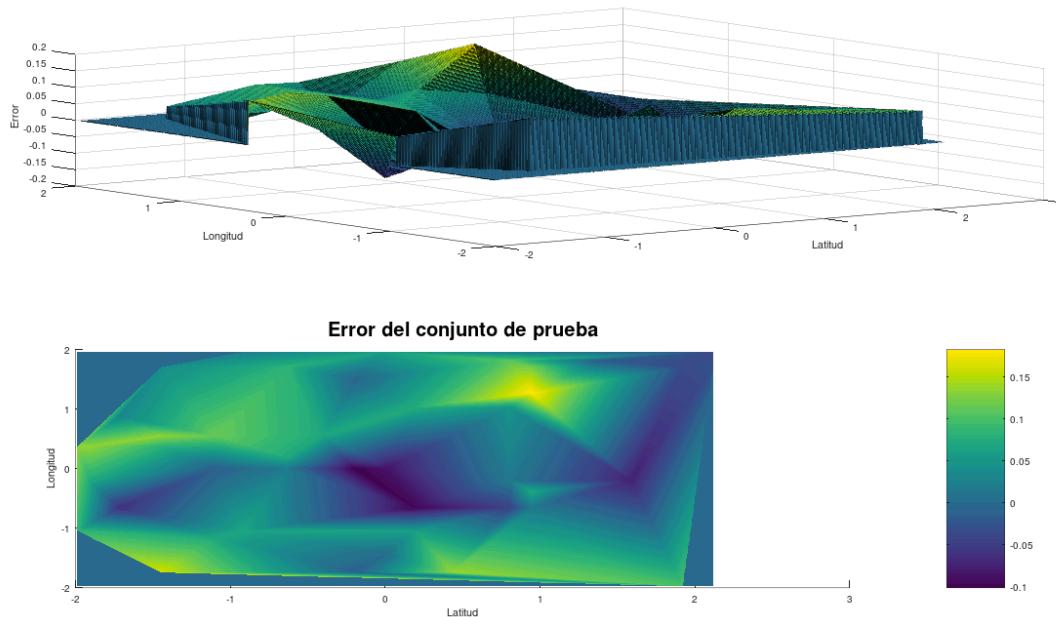
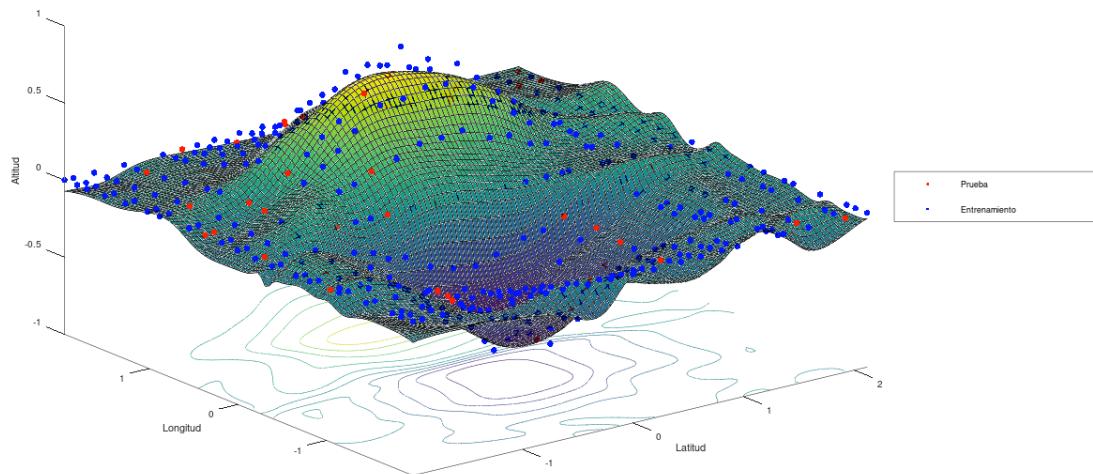
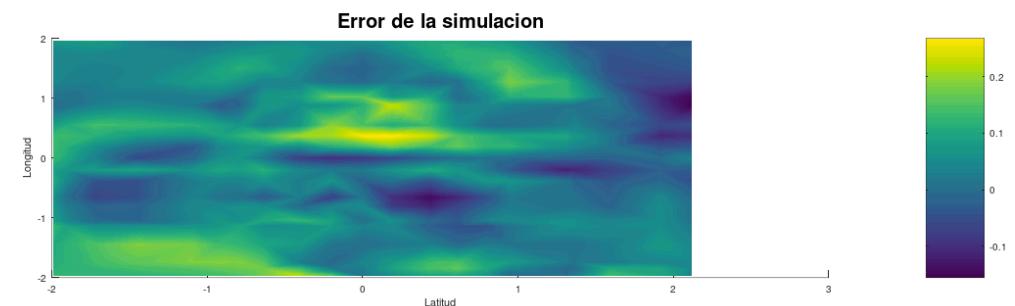
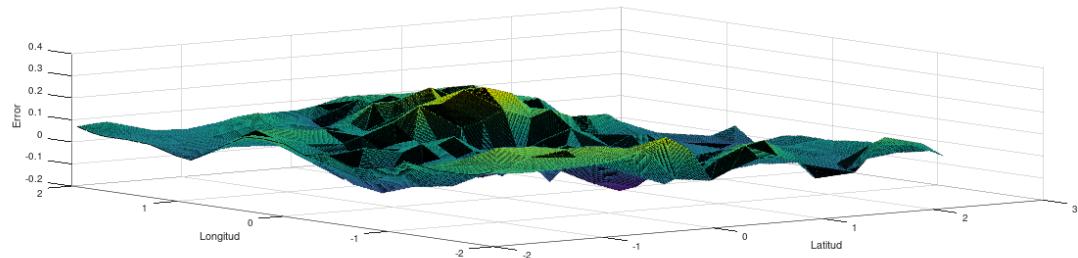


Figura 19 – Error obtenido en el conjunto de prueba

**Simulacion y mediciones***Figura 20 – Aproximación de la altitud**Figura 21 – Error obtenido en la aproximación*

### Arquitectura 06: [ 2 - 45 - 50 - 1 ] (ECM = 0.001563) (EXP)

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05.

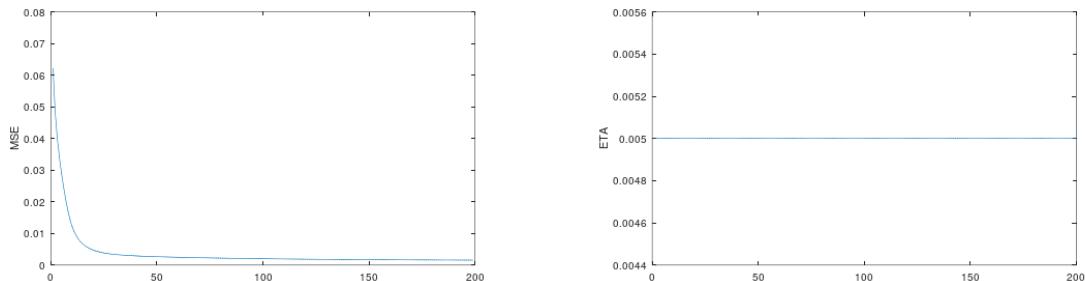


Figura 22 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

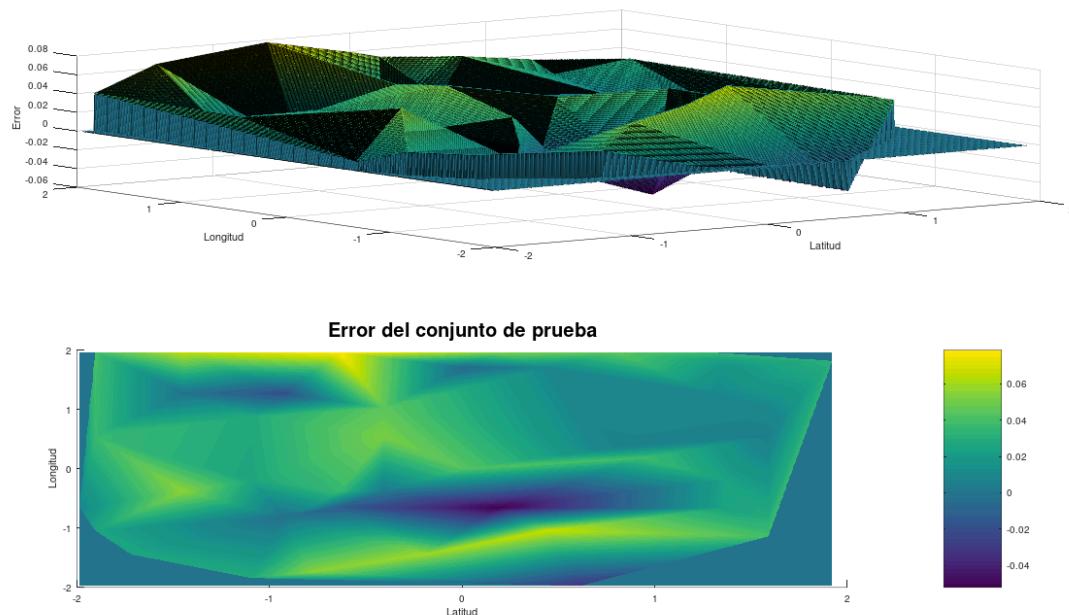


Figura 23 – Error obtenido en el conjunto de prueba

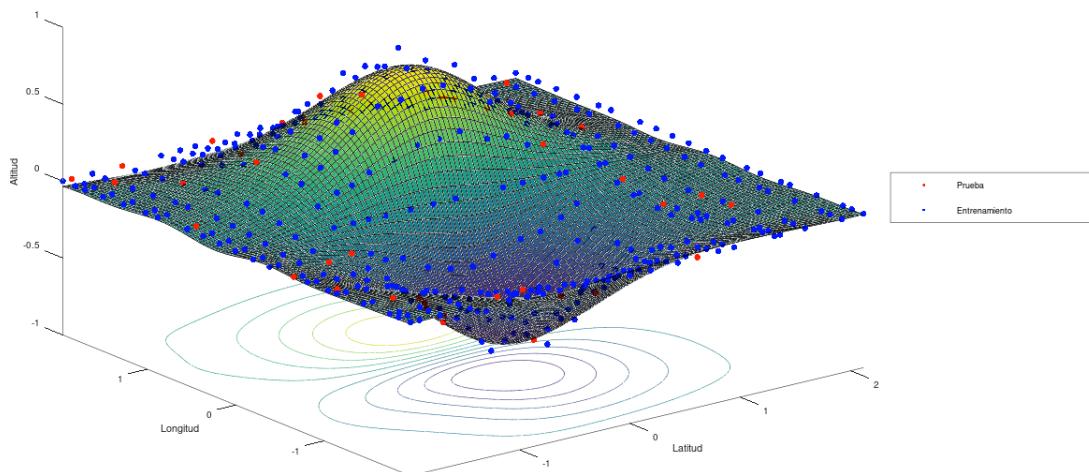
**Simulacion y mediciones**

Figura 24 – Aproximación de la altitud

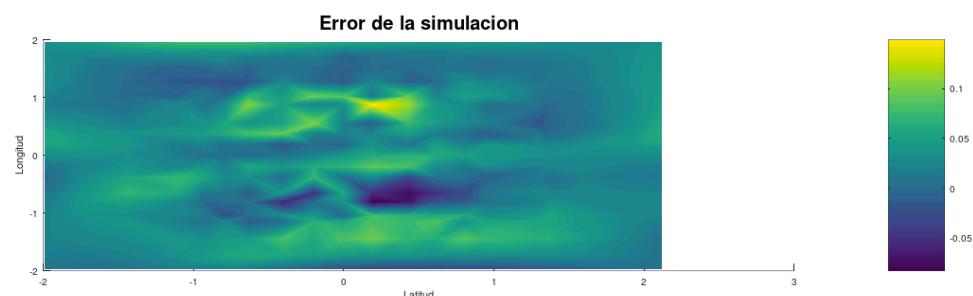
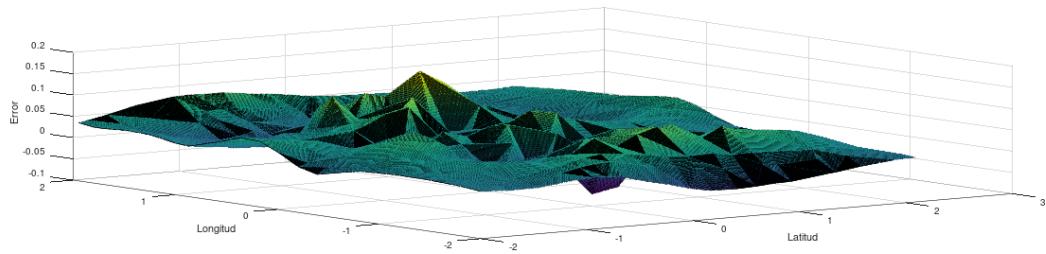


Figura 25 – Error obtenido en la aproximación

### Arquitectura 07: [ 2 - 45 - 50 - 1 ] (ECM = 0.001040) (EXP)

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05, parámetros adaptativos a = 0.00003, b = 0.002 y K = 50.

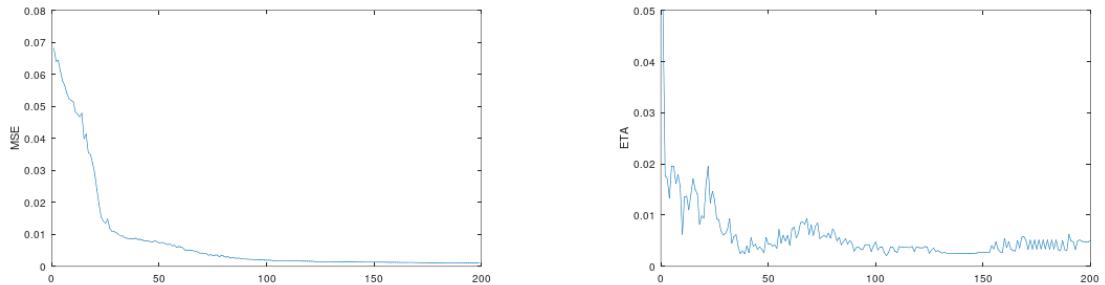


Figura 26 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

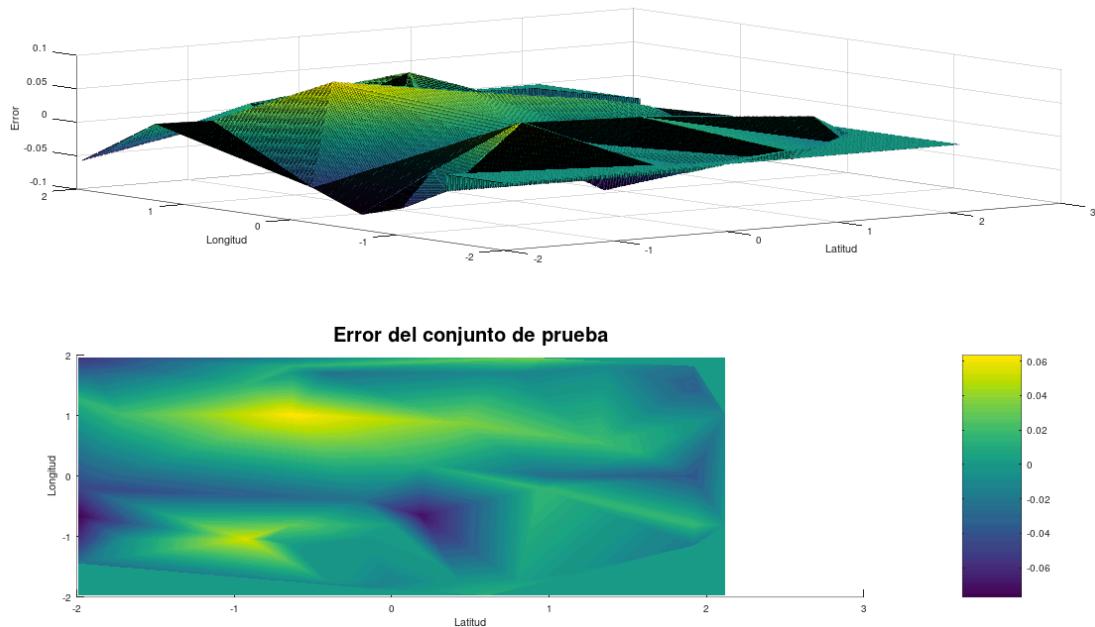
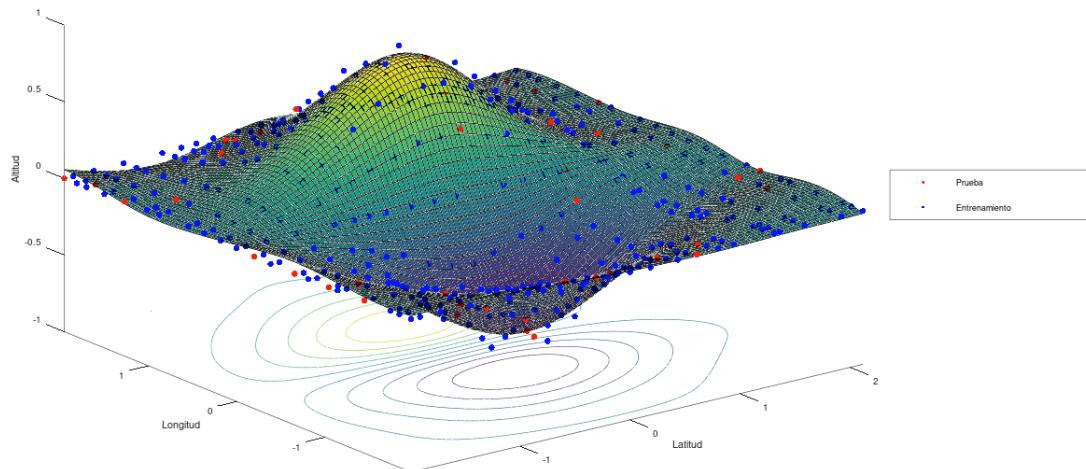
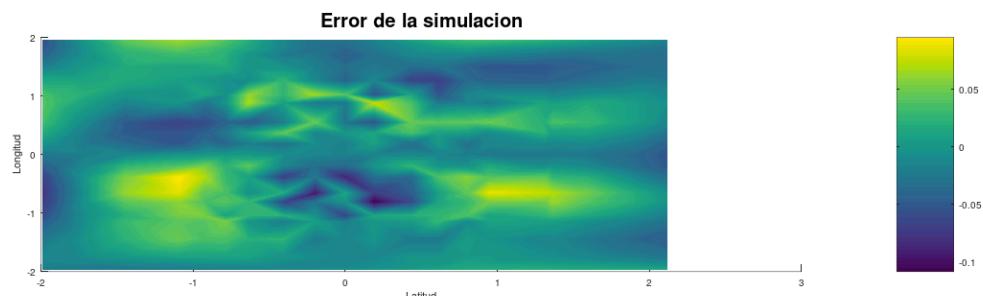
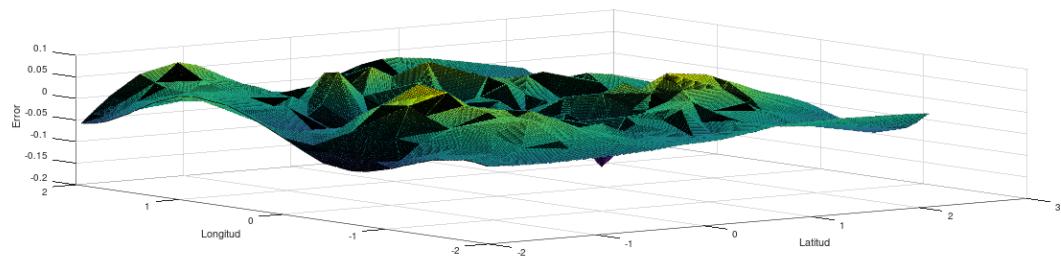


Figura 27 – Error obtenido en el conjunto de prueba

**Simulacion y mediciones***Figura 28 – Aproximación de la altitud**Figura 29 – Error obtenido en la aproximación*

## Arquitectura 08: [ 2 - 30 - 10 - 1 ] (ECM = 0.000979) (EXP)

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05.

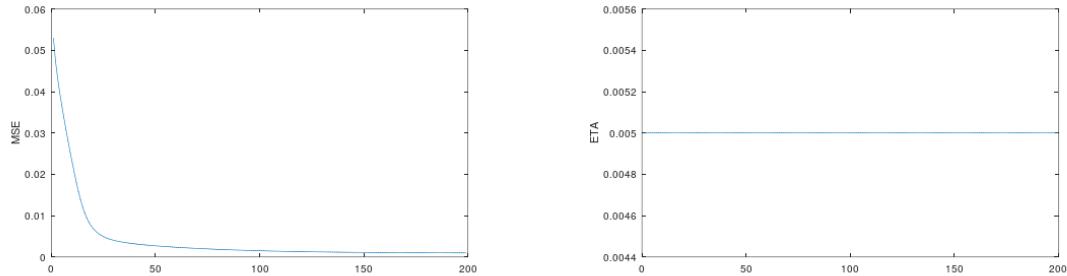


Figura 30 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

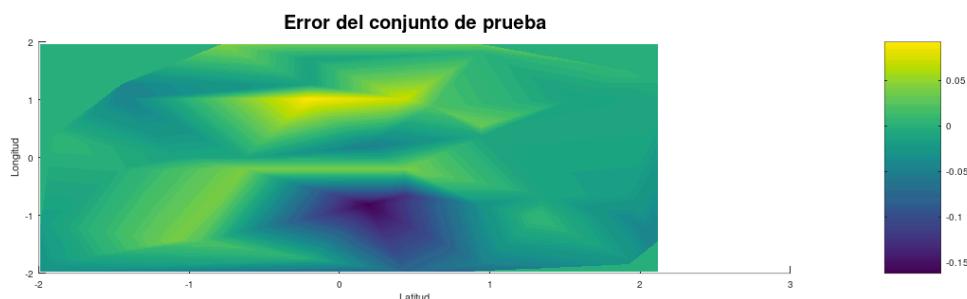
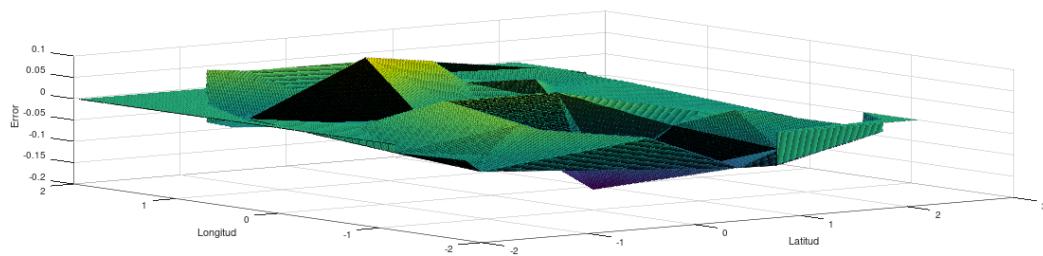
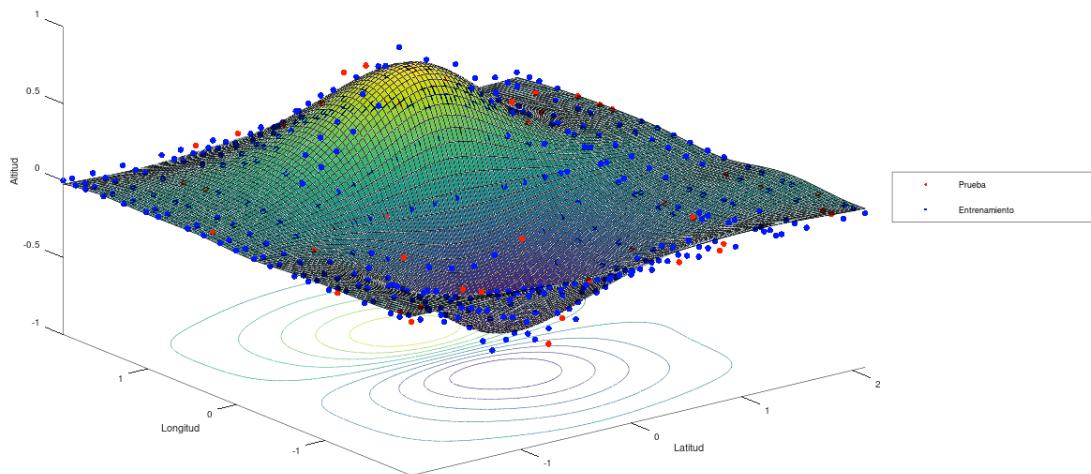
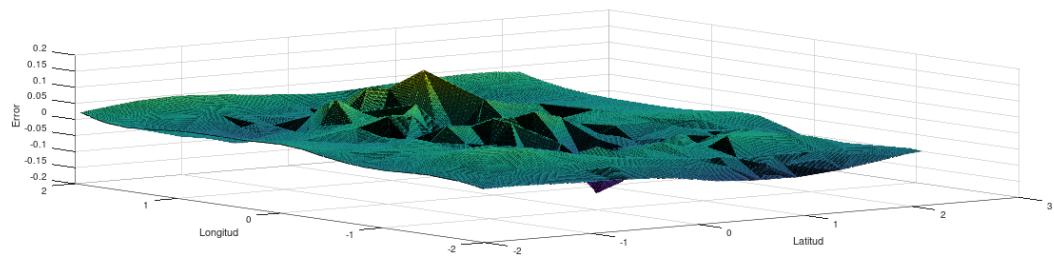
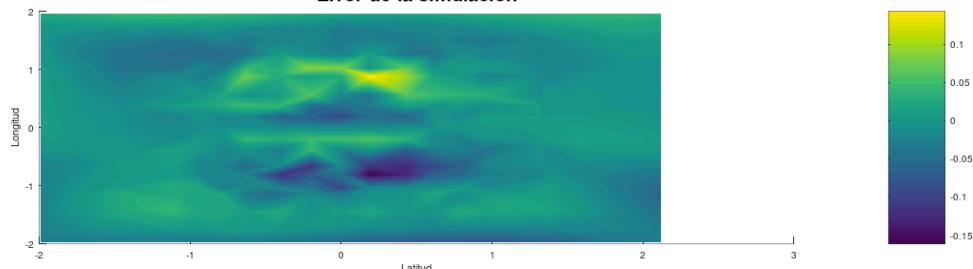


Figura 31 – Error obtenido en el conjunto de prueba

**Simulacion y mediciones***Figura 32 – Aproximación de la altitud**Error de la simulacion**Figura 33 – Error obtenido en la aproximación*

**Arquitectura 09: [ 2 - 30 - 10 - 1 ] (ECM = 0.001781) (EXP)**

PARAMETROS: Cantidad máxima de épocas: 200, coeficiente de momentum: 0.9, learning rate 0.05, parámetros adaptativos a = 0.00003, b = 0.002 y K = 50.

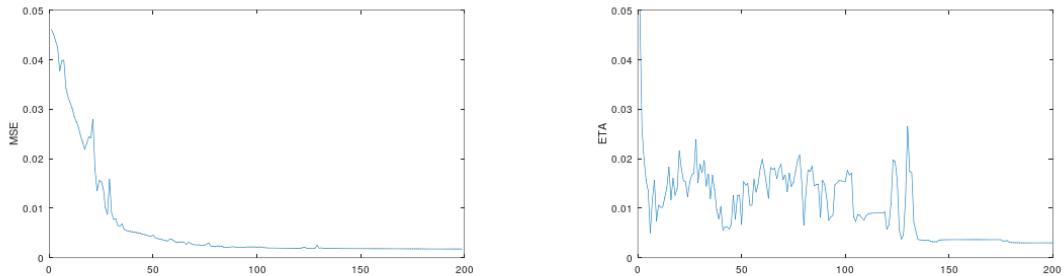


Figura 34 – Error cuadrático medio y eta en función de las épocas durante el entrenamiento

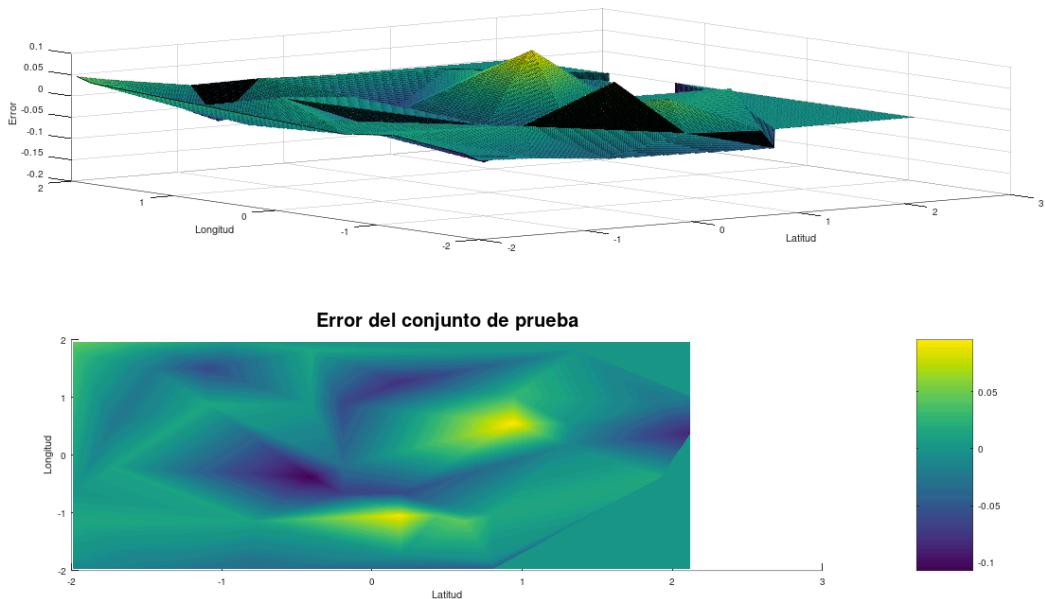


Figura 35 – Error obtenido en el conjunto de prueba

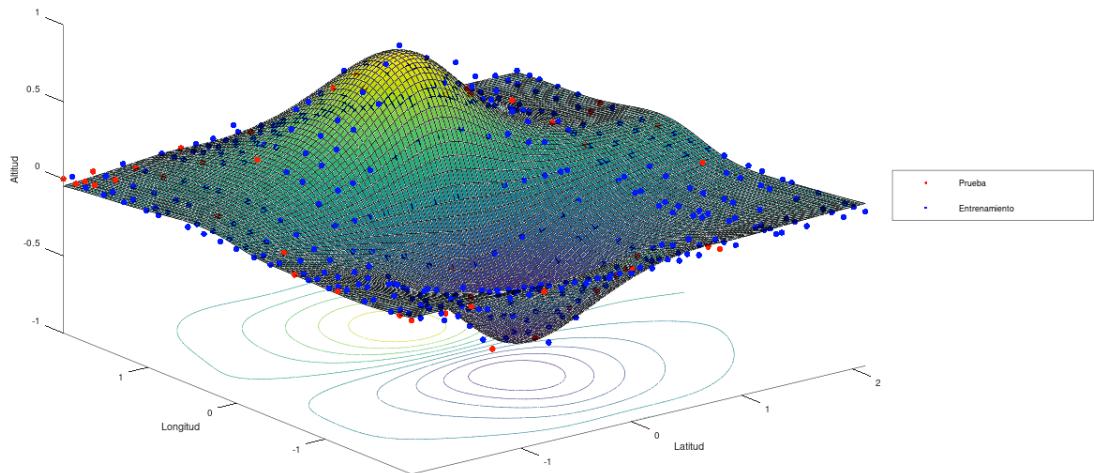
**Simulación y mediciones**

Figura 36 – Aproximación de la altitud

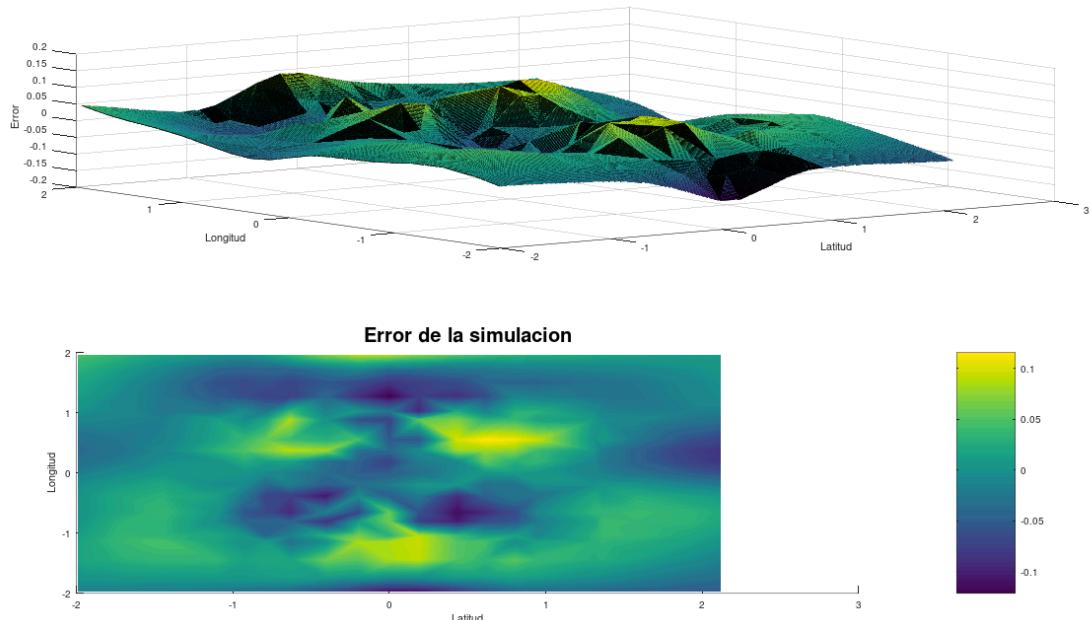


Figura 37 – Error obtenido en la aproximación