

## Laboratorio 3

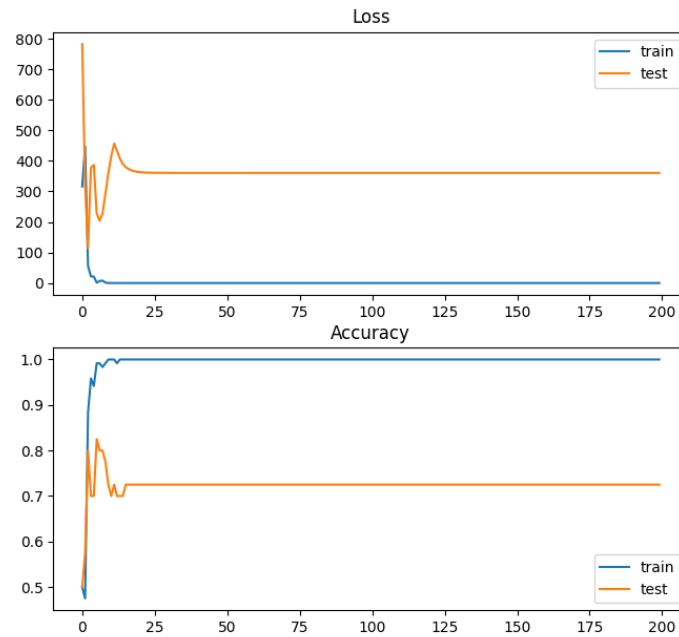


Figura 1. Evolución de la pérdida y la exactitud del modelo en función del número de epochs.

```
Matriz confusion  
tf.Tensor(  
[[11  9]  
 [ 2 18]], shape=(2, 2), dtype=int32)
```

Figura 2. Matriz de confusión

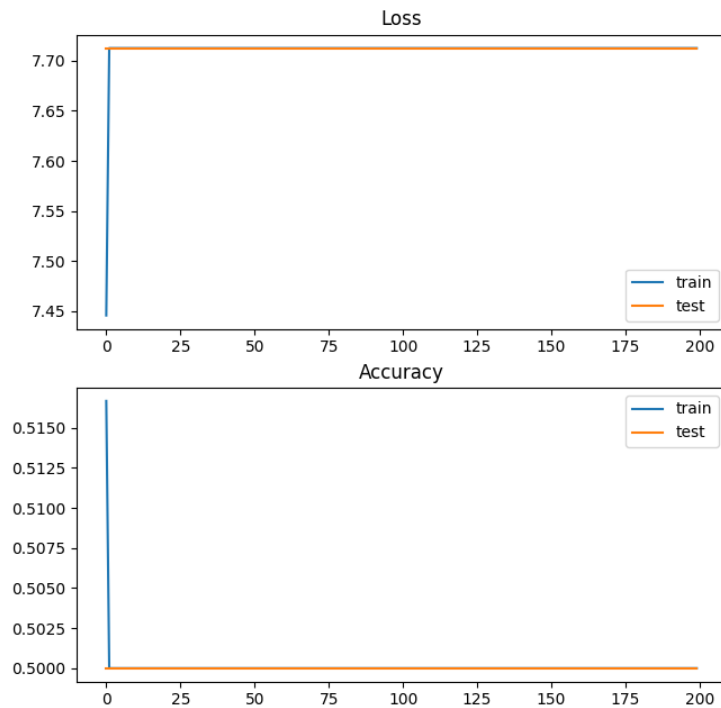


Figura 3. Evolución de la pérdida y la exactitud del modelo en función del número de epochs con función de activación lineal.

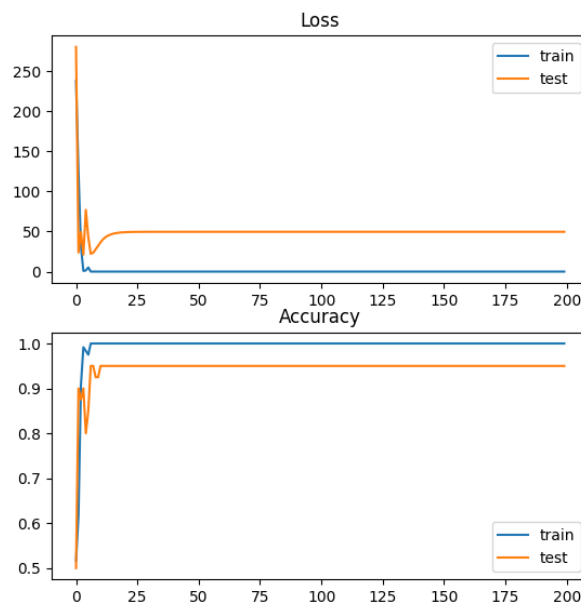


Figura 4. Evolución de la pérdida y la exactitud del modelo en función del número de epochs con función de activación SGD y cross validation

```
Matriz confusion
tf.Tensor(
[[20  0]
 [ 2 18]], shape=(2, 2), dtype=int32)
```

Figura 5. Matriz de confusión para hiperparametros seteado

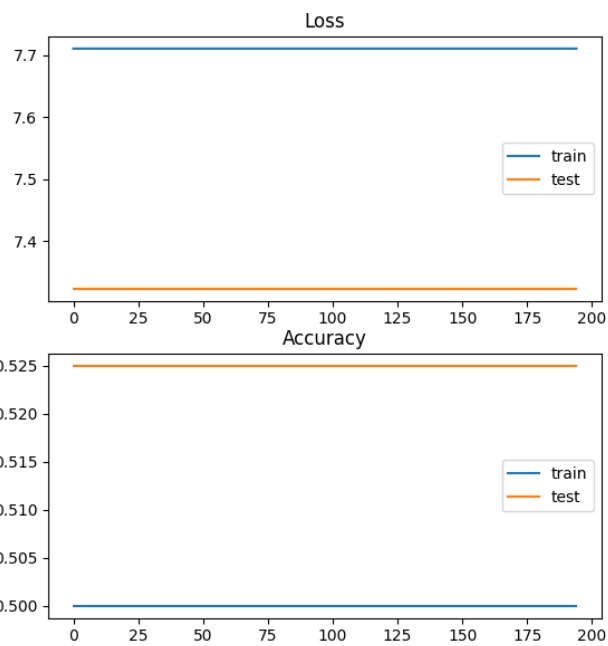


Figura 6. Evolución de la pérdida y la exactitud del modelo en función del número de epochs con función de activación lineal y cross validation

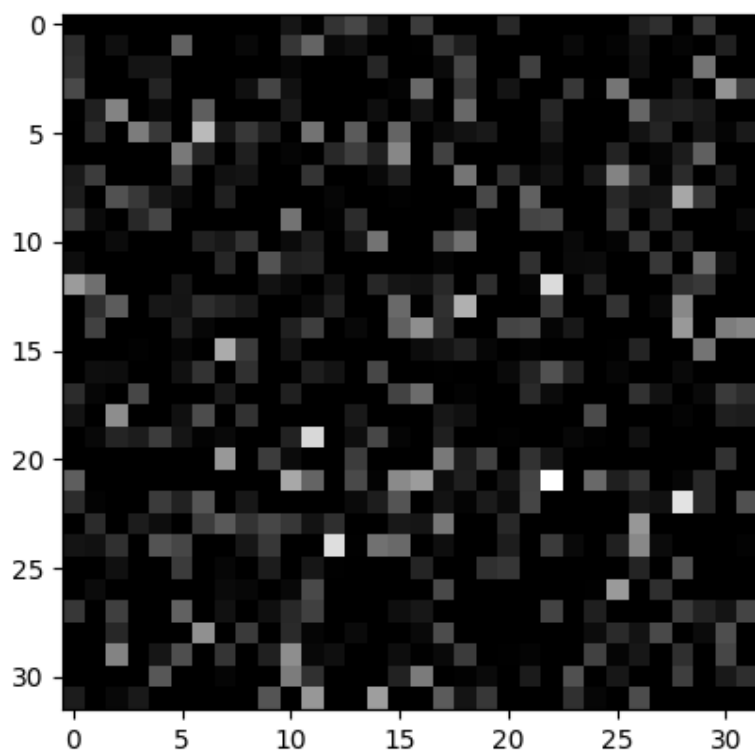


Figura 7. Imagen de pesos para clasificación de perros y gatos.

## Comparación de Trayectorias

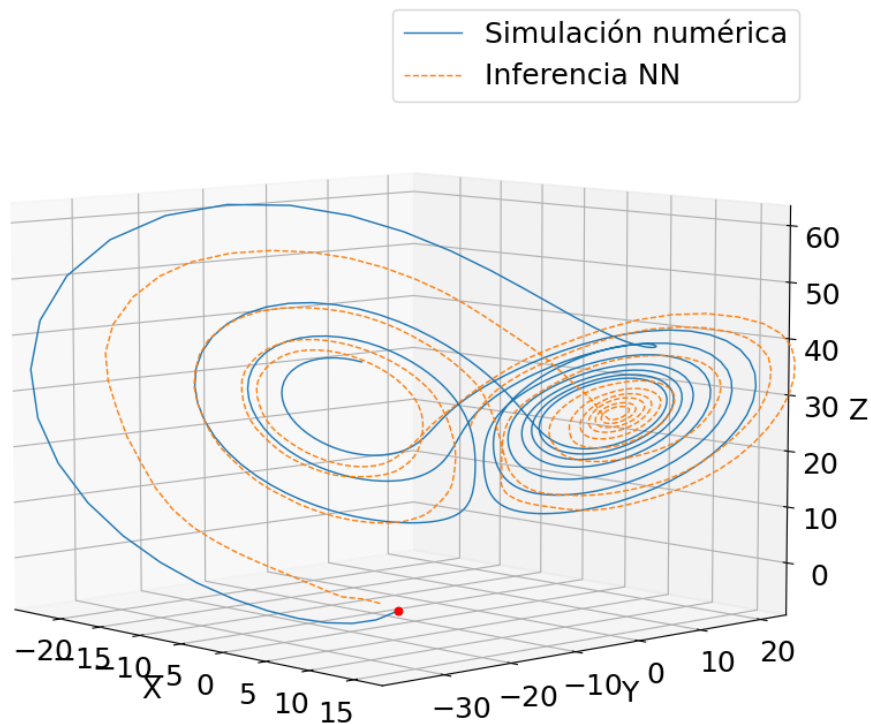


Figura 8. Trayectorias del sistema del sistema de LORENZ por método numérico versus predicción red neuronal, montado.

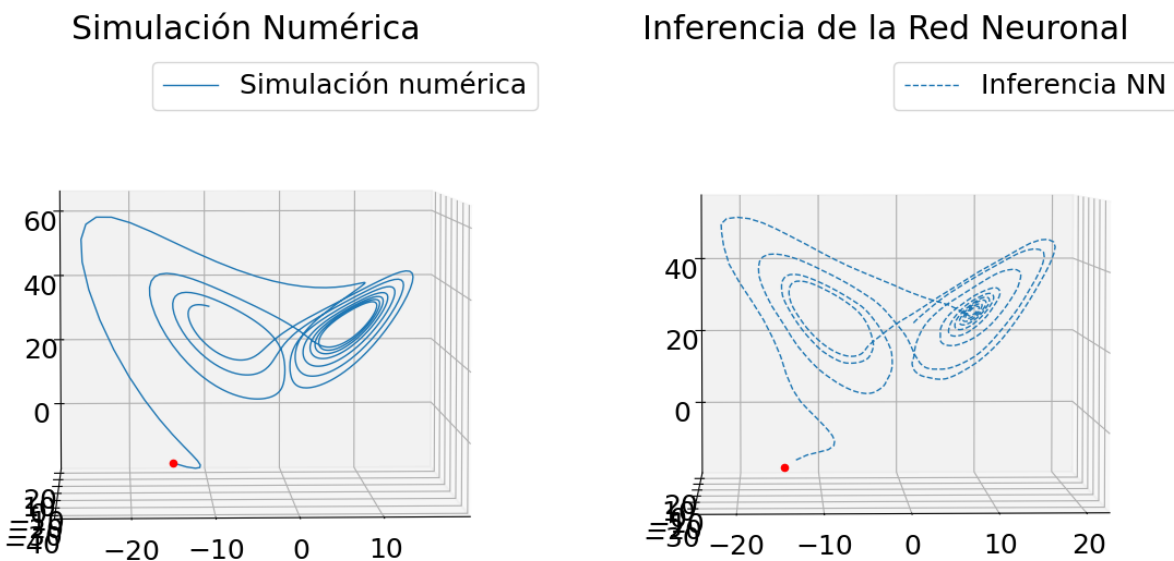


Figura 9. Trayectorias del sistema del sistema de LORENZ por método numérico versus predicción red neuronal, individual.

## PARTE I

¿Existe una discrepancia entre la exactitud del modelo para la data de entrenamiento y la de validación?

Si, esto puede deberse a varios factores. Uno de ellos podría ser el hecho que no se seleccionó de forma aleatoria los datos de entrenamiento, otro de estos factores puede ser el hiperparámetro.

¿Según la evolución de la pérdida, cree que el modelo obtenido le hace un overfit a la data de entrenamiento?

Sí, ya que hay una gran diferencia entre el train y test.

¿Qué le dice la matriz de confusión con respecto al rendimiento del modelo en el set de validación?

Se estima que el 75% de las predicciones sean correctas, ya que generará datos erróneos como se puede observar en la Figura 2.

¿Qué ocurre si cambia la función de activación del modelo a una lineal?

El modelo se pierde ya que hay poco overfit como se observa en la Figura 3, además que el 50% de las veces acertara.

¿Cuál es el valor de los hiper-parámetros que mejor le funcionaron?

activation = 'sigmoid'

learning\_rate = 0.008

momentum = 0.92

epochs = 195

¿Existe una discrepancia entre la exactitud del modelo para la data de entrenamiento y la de validación?

Sí, pero esta es muy mínima. Se puede observar en la Figura 4 y 5.

¿Según la evolución de la pérdida, cree que el modelo obtenido le hace un overfit a la data de entrenamiento?

Sí, y este al no ser muy grande es beneficioso.

¿Qué le dice la matriz de confusión con respecto al rendimiento del modelo en el set de validación?

Es muy eficiente siendo que el 95% de las veces acertara

¿Qué ocurre si cambia la función de activación del modelo a una lineal?

Existe demasiado overfit, por lo que nuevamente el perceptrón no será preciso. Se puede observar el resultado en la Figura 6.

¿Qué puede decir sobre el modelo con base en esta imagen?

La forma en la que logra realizar la clasificación es muy abstracta

¿Qué factores son determinantes para el modelo al momento de efectuar la clasificación entre gatos y perros?

Por lo que se logra percibir, influye la forma de la trompa, orejas y contorno de la cara.

## PARTE II

¿Cuál fue la arquitectura final que empleó para resolver el modelo (especifique)? ¿Qué tan exacto fue el modelo entrenado?

¿Qué tan cara, computacionalmente hablando, es la evaluación de la red neuronal entrenada en comparación a la evaluación del campo vectorial del sistema Lorenz?  
¿Cómo puede establecer que el modelo no está presentando overfitting?

El entrenamiento del modelo llevó un tiempo aproximado de 6 minutos, a pesar de ello no es tan preciso, esto puede deberse a varios factores, siendo uno de ellos la función de activación los cuales para esta red neuronal se usaron 4 capas de 128 neuronas cada una siendo las funciones de activación “relu”, “elu”, “relu”, “elu”, respectivamente. Lo que definitivamente sesga la decisión por el tipo de signo de las funciones, a pesar de ello se obtuvo las siguientes características del modelo:

- Loss: 0.4218
- Accuracy: 0.9966

Observando la Figura 8 y 9 se puede inferir que si existe overfitting.