

# Trabajo práctico 3

## Redes Neuronales y Aprendizaje Profundo

Ignacio Ezequiel Cavicchioli  
Padrón 109428  
icavicchioli@fi.uba.ar

2/12/2025

### Índice

1	Introducción	2
2	Teoría sobre redes de Kohonen y SOM	2
3	Ejercicio 1	3
3.1	Consignas . . . . .	3
3.2	Desarrollo . . . . .	3
3.3	Análisis . . . . .	8
4	Ejercicio 2	9
4.1	Consignas . . . . .	9
4.2	Desarrollo . . . . .	9
4.3	Análisis . . . . .	9
5	Ejercicio 3	18
5.1	Consignas . . . . .	18
5.2	Desarrollo . . . . .	18
5.3	Análisis . . . . .	18
6	Conclusiones	18

## 1. Introducción

Este documento presenta el desarrollo de las consignas del trabajo práctico N°3 de la materia de **Redes Neuronales y Aprendizaje Profundo**. El código correspondiente fue realizado en *Jupyter notebooks, Python*, adjuntados a la entrega en formato PDF. Toda imagen o implementación requeridas para el análisis se explicitarán en el presente archivo, por lo que la lectura del código en sí queda a discreción del lector. La teoría relevante será presentada y discutida en la sección pertinente.

## 2. Teoría sobre redes de Kohonen y SOM

Las *Self-Organizing Maps* (SOM), o *redes de Kohonen*, son un tipo de red neuronal no supervisada diseñada para proyectar datos de alta dimensionalidad sobre una rejilla de menor dimensión, preservando en lo posible la topología del espacio original. Este proceso permite visualizar, agrupar y extraer estructuras latentes en los datos.

La palabra “rejilla” se refiere a la estructura informática/topológica donde viven las neuronas. Esta puede ser:

- 1D: una línea (vector de neuronas) que se cierra o no en los extremos.
- 2D: una matriz, que se puede o no cerrar en sus lados.
- 3D o más: un tensor de orden mayor.

El mecanismo de entrenamiento está inspirado en el aprendizaje hebbiano: las neuronas que responden de manera similar a un mismo estímulo se refuerzan conjuntamente. Sin embargo, la SOM incorpora además un esquema de *competitive learning*, donde sólo la neurona ganadora (y sus vecinas en la rejilla) actualizan sus pesos (o lo que corresponda según la función de vecindad). Esto genera la organización progresiva del mapa y produce un *feature mapping* explícito, ordenando los patrones según similitud.

Las SOM actúan como un método de reducción de dimensionalidad no lineal, construyendo representaciones espaciales que capturan regularidades estadísticas de los datos sin requerir etiquetas. Esto mismo se va a ver en la tercera consigna de este trabajo, donde se pasa de un espacio dimensional de N dimensiones al espacio 2D de la rejilla de neuronas, y se usa la matriz U para ratificar la existencia de *clusters*.

### 3. Ejercicio 1

#### 3.1. Consignas

Construya una red de Kohonen de 2 entradas que aprenda una distribución uniforme dentro del círculo unitario. Mostrar el mapa de preservación de topología. Probar con distribuciones uniformes dentro de otras figuras geométricas.

#### 3.2. Desarrollo

Dado lo expuesto en la teoría sobre redes de Kohonen y lo visto en clase, se espera que las redes implementadas y entrenadas puedan aprender las distribuciones uniformes que se les van a dar en este inciso. Visualmente, esto debería notarse como que las neuronas están aproximadamente equidistantes las unas de las otras, siempre dentro de los límites de la distribución.

Ahora bien, para generar muestras aleatorias uniformes en un círculo hay que pensar que se quiere la misma probabilidad de caer en cualquier parte de su área. Como el área de esta figura depende del radio, sortear un  $r$  y  $\theta$  distribuidos uniformemente entre  $(0, 1)$  y  $(0, 360^\circ)$  respectivamente generaría una distribución conjunta con mayor densidad de muestras en el centro, indicativo de mayor probabilidad.

Para balancear esto, se ajusta la distribución del radio sorteado a  $r = R\sqrt{u}$  con  $u \sim \mathcal{U}\{0, 1\}$ .

Con un  $r$  y  $\theta$  correctamente generados, se procede a hacer una conversión a coordenadas cartesianas, que van a usar las neuronas.

Las otras figuras geométricas fueron generadas de forma similar, haciendo uso de cambios de coordenadas para mantener la uniformidad.

Para entrenar la red de Kohonen utilizada en este ejercicio, se implementó una versión vectorizada del algoritmo, que es más rápida que *loopear* por todas las neuronas. La inicialización de los pesos se realizó tomando muestras aleatorias del propio conjunto de datos. Durante el entrenamiento, los pesos se modificaron mediante una función de vecindad gaussiana, de varianza linealmente decreciente con las iteraciones, que debería ayudar con la organización del mapa. La tasa de aprendizaje se mantuvo constante.

Para cada muestra, se determina la neurona ganadora (*Best Matching Unit*, BMU) y se actualizan tanto ella como sus vecinas de acuerdo con la función de vecindad. Esto se hace una cantidad de iteraciones determinada por el usuario. Para este caso, el algoritmo se ejecutó hasta ver un resultado invariante.

las imágenes de a continuación muestran las figuras generadas con las neuronas en sus ubicaciones aprendidas. Las figuras tienen 1000 muestras (puntos de datos), y hay 25 neuronas distribuidas en una matriz de  $5 \times 5$ , razón por la cual se forma una *lattice* (rejilla) de polígonos de 4 lados; cada neurona se conecta a otras 4.

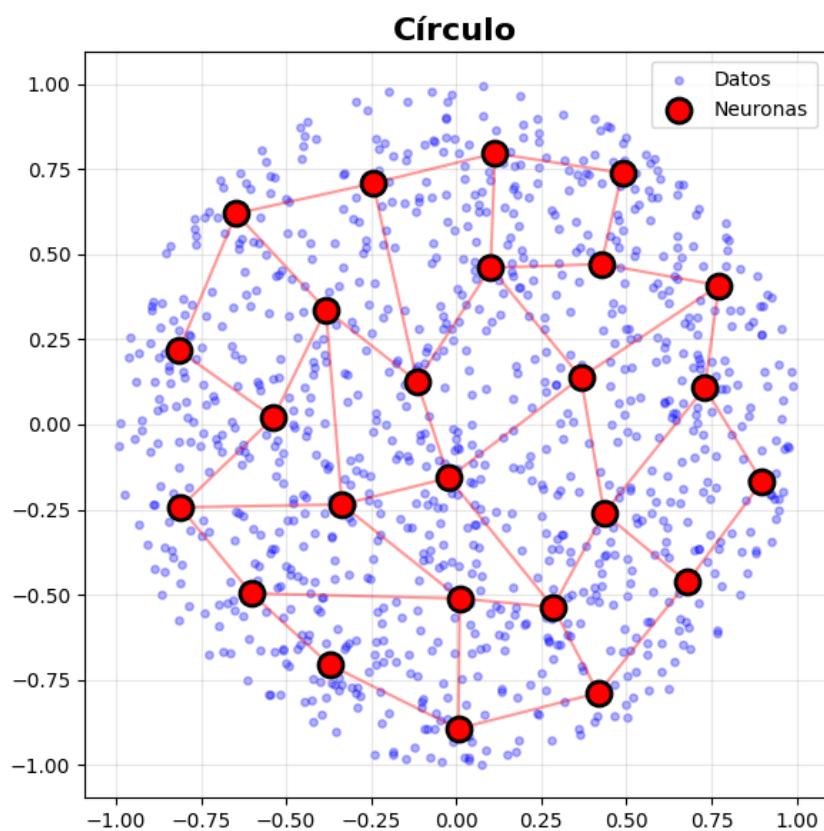


Figura 1: Distribución de datos con forma de círculo y posiciones finales de las neuronas de la red luego del entrenamiento.

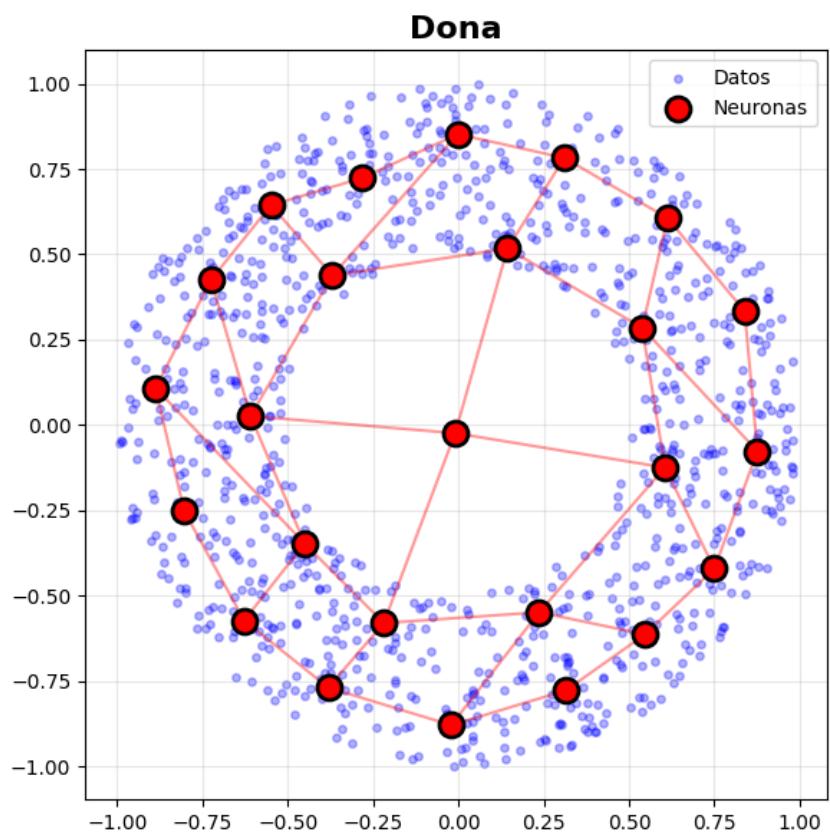


Figura 2: Distribución de datos con forma de dona y posiciones finales de las neuronas de la red luego del entrenamiento.

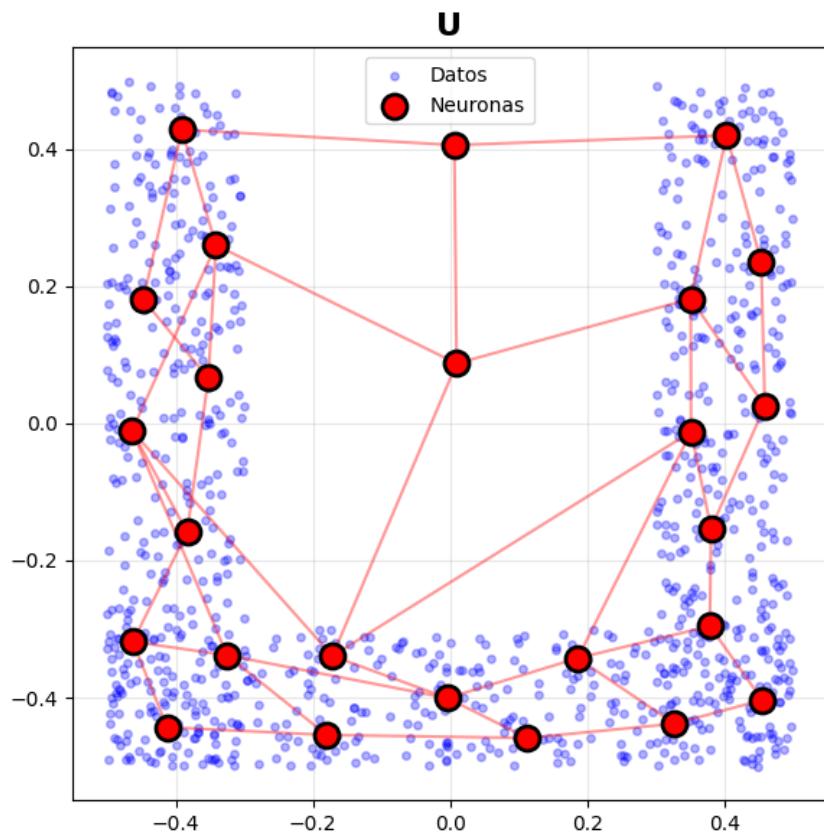


Figura 3: Distribución de datos con forma de U y posiciones finales de las neuronas de la red luego del entrenamiento.

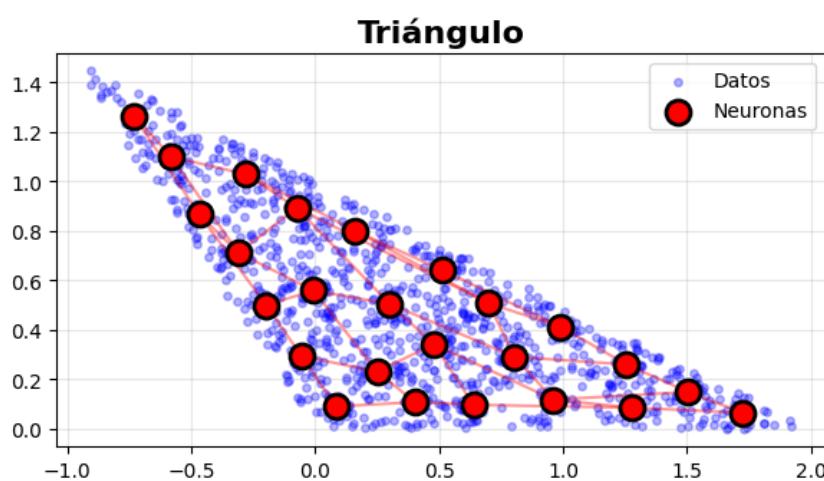


Figura 4: Distribución de datos con forma de triángulo y posiciones finales de las neuronas de la red luego del entrenamiento.

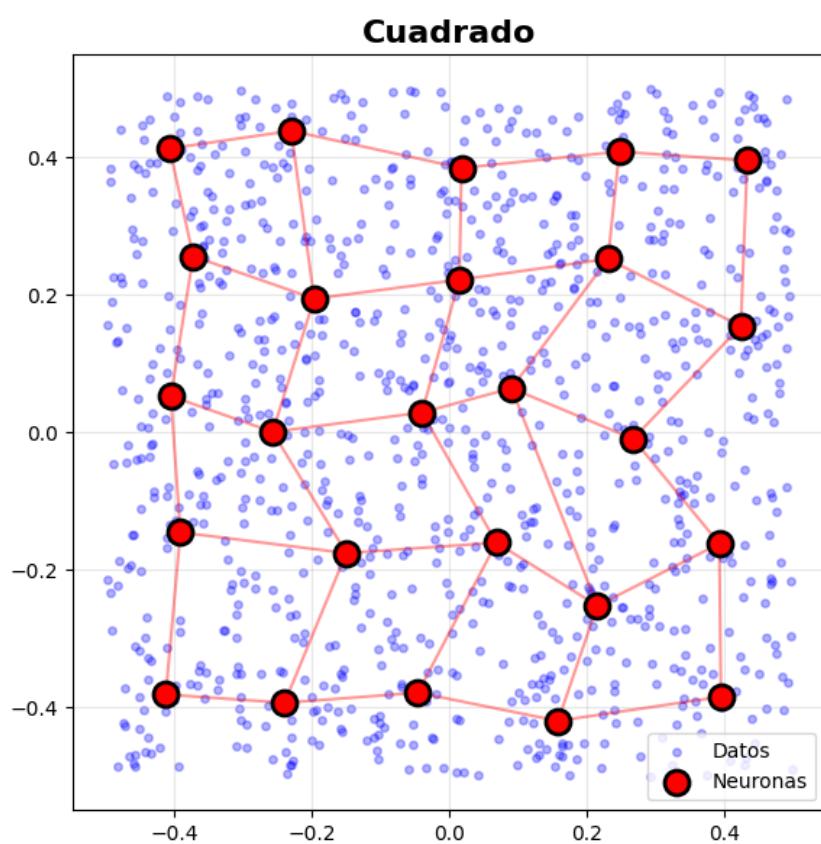


Figura 5: Distribución de datos con forma de cuadrado y posiciones finales de las neuronas de la red luego del entrenamiento.

### 3.3. Análisis

Se observa que la red implementada es capaz de aprender considerablemente bien las distribuciones de las figuras 1, 4 y 5, que son el círculo, triángulo y cuadrado respectivamente. La posición de las neuronas es aproximadamente equidistante las unas de las otras, respondiendo a la distribución de datos subyacente. También se pudieron generar los mapas de preservación de topología.

En las formas U y dona (fig. 2 y 3), las neuronas aprendieron la distribución pero hay algunas unidades que quedaron por fuera de las figuras: en la U quedaron 2 neuronas mientras que en la dona, 1. Se considera que este efecto tiene su origen en que esas neuronas nunca ganan, y son movidas por todas las otras neuronas con una “fuerza” inversamente proporcional a la distancia. Por esto mismo hay neuronas en el centro de la dona (todas las neuronas dentro de la dona tiran con la misma fuerza por estar a la misma distancia), y en la U tenemos neuronas más arriba o abajo (las neuronas de los laterales tiran hacia si mismas y contrarrestan los efectos de las de la parte de abajo de la U).

En las otras geometrías esto no sucede.

Para entender esto se puede ver la imagen 6, que tiene algunos de los vectores de fuerza imaginaria en verde. En ella se va a notar con claridad lo enunciado más arriba.

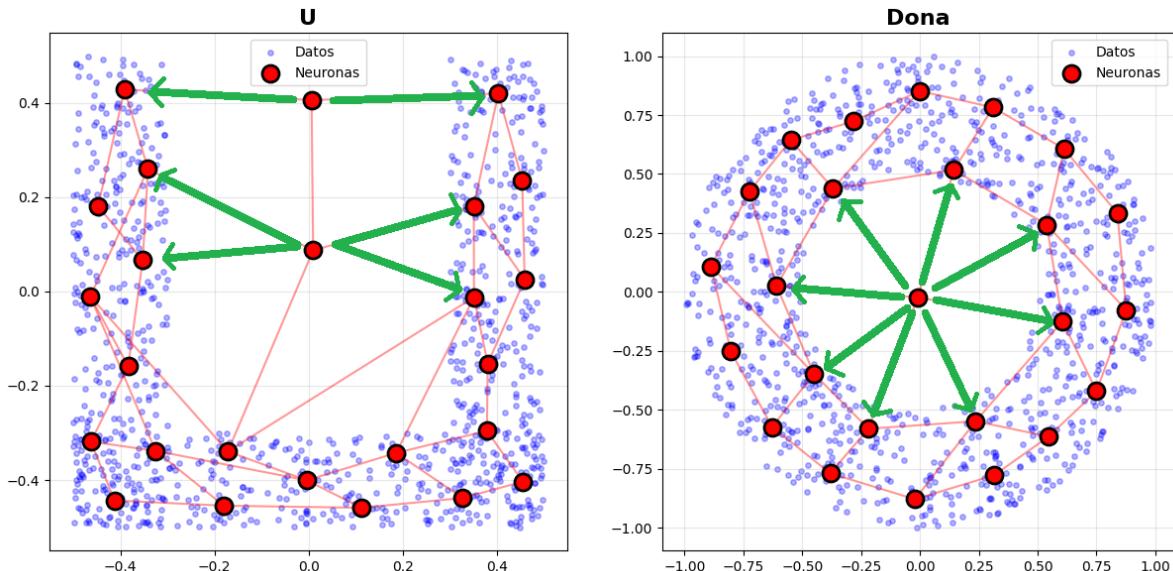


Figura 6

Otro detalle interesante es que, si se observa con detenimiento, las neuronas están sobre los puntos de mayor densidad de muestras, indicando que:

1. El aprendizaje está funcionando porque las neuronas se acercan a donde ganan más, que coincide con las zonas donde hay mayor concentración de muestras.
2. Las distribuciones generadas no fueron perfectamente uniformes: la presencia de cúmulos locales de mayor densidad hace que algunas neuronas se agrupen en esas regiones.

## 4. Ejercicio 2

### 4.1. Consignas

Resuelva (aproximadamente) el “Traveling salesman problem” para 200 ciudades con una red de Kohonen.

### 4.2. Desarrollo

Para este ejercicio se usó un *dataset* de 312 ciudades estadounidenses, a partir del cual se crearon sets de menos muestras para ir mostrando la resolución del problema del “Traveling salesman problem” de menor a mayor complejidad.

La

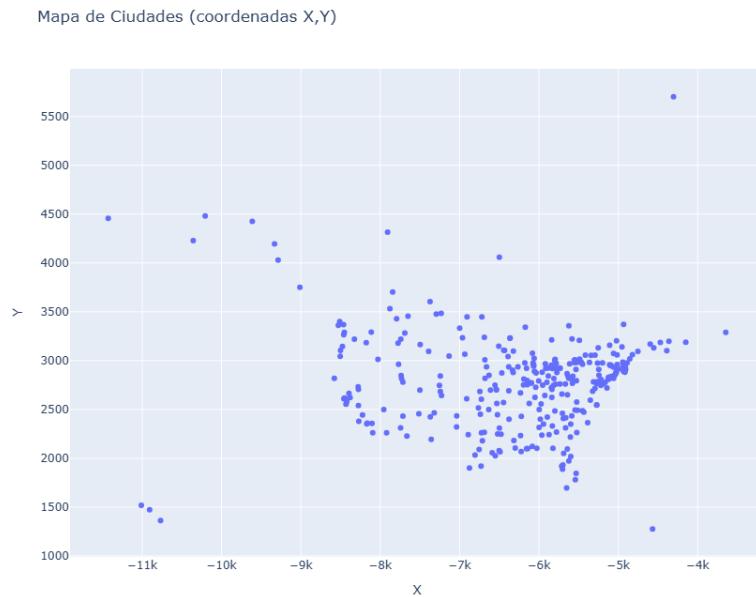


Figura 7: Todas las muestras de ciudades de USA del dataset

### 4.3. Análisis

Kohonen 1D 20 ciudades

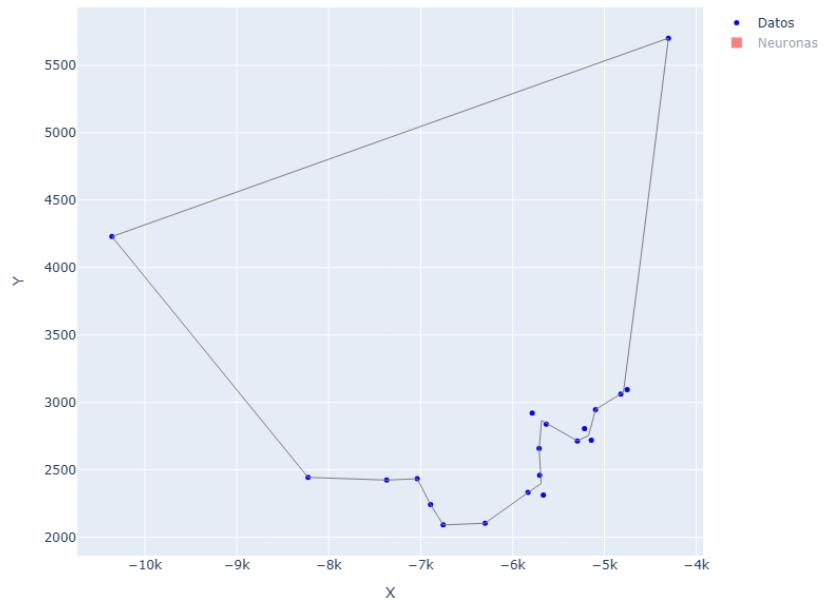


Figura 8: 20 ciudades con 20 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 20 ciudades

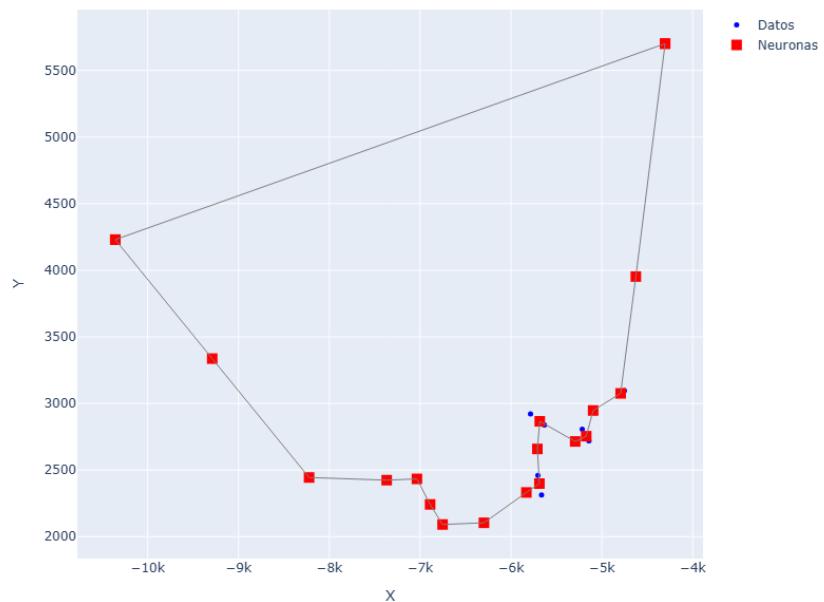


Figura 9: 20 ciudades con 20 neuronas - puede haber overfitting: neuronas sobrantes o neuronas que representan varias ciudades

Kohonen 1D 20 ciudades

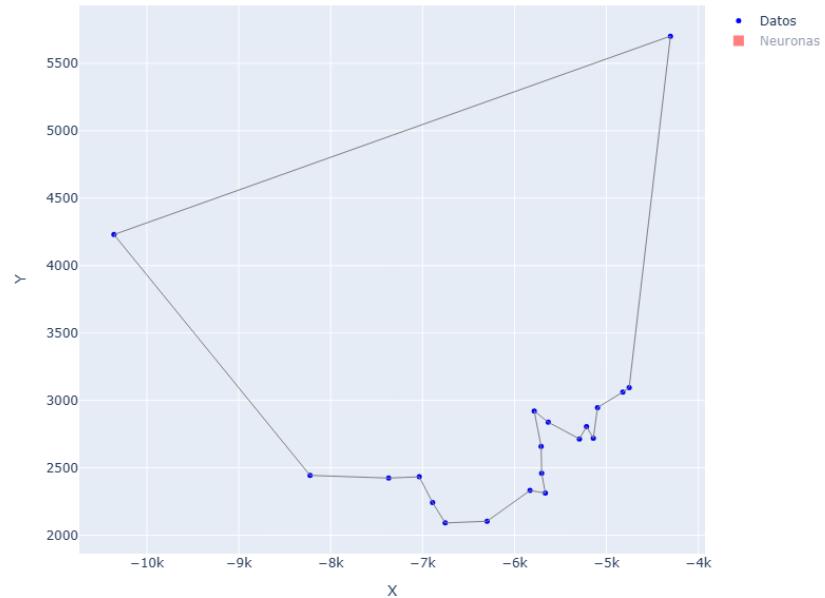


Figura 10: 20 ciudades 40 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 20 ciudades

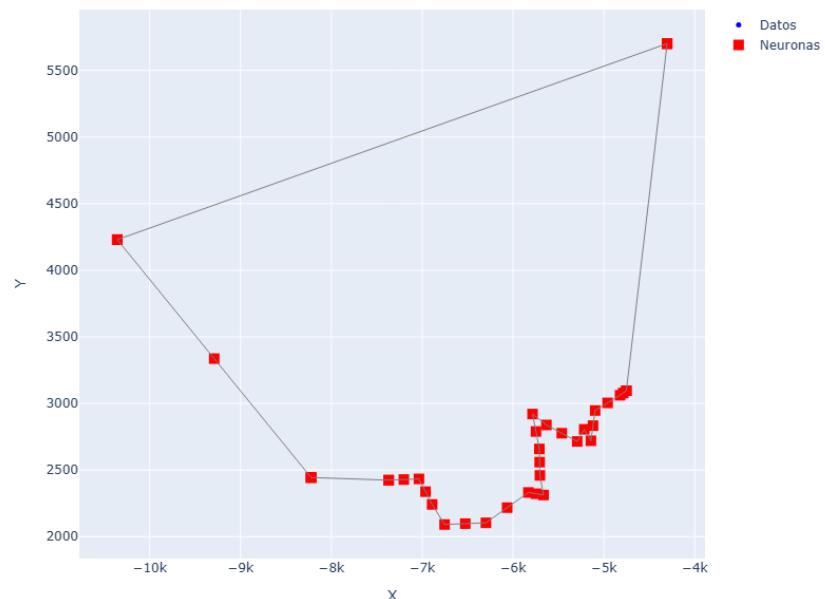


Figura 11: 20 ciudades 40 neuronas

Kohonen 1D 50 ciudades

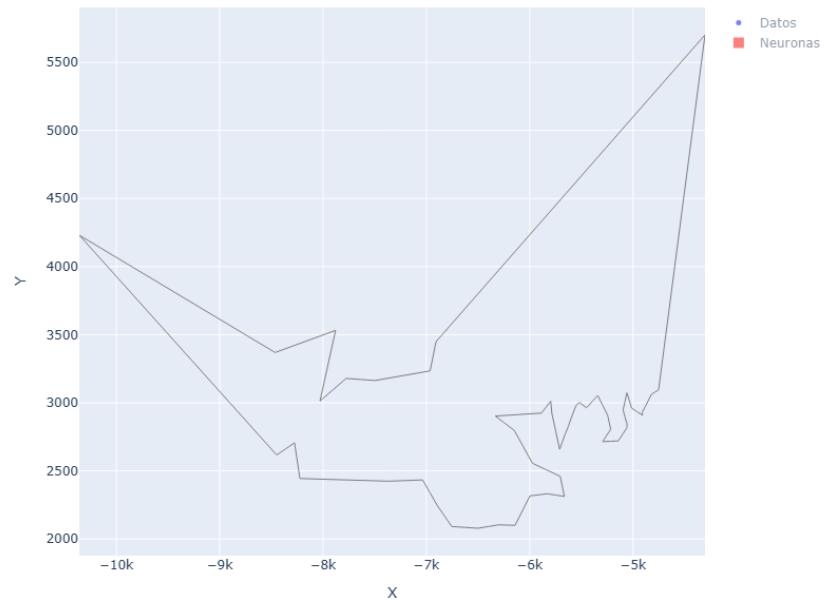


Figura 12: 50 ciudades 100 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 50 ciudades

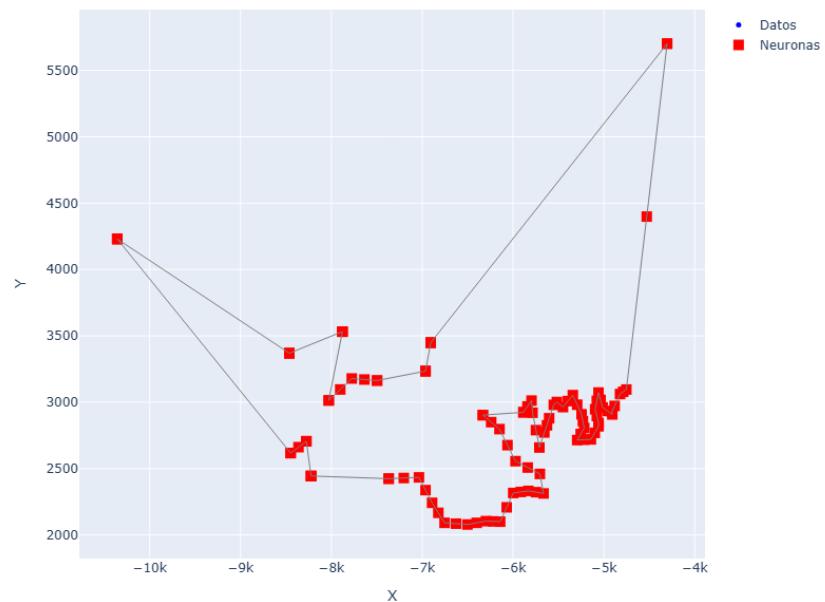


Figura 13: 50 ciudades 100 neuronas

Kohonen 1D 100 ciudades

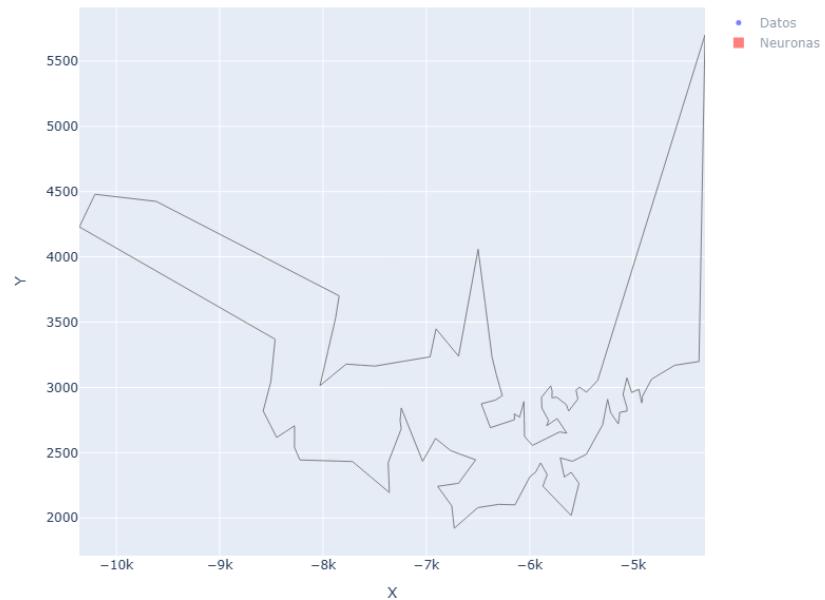


Figura 14: 100 ciudades 200 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 100 ciudades

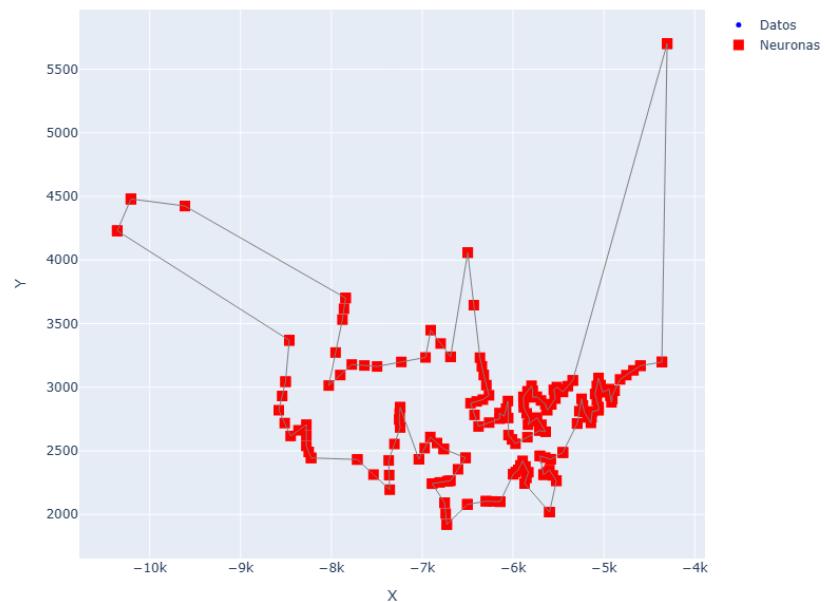


Figura 15: 100 ciudades 200 neuronas

Kohonen 1D 200 ciudades

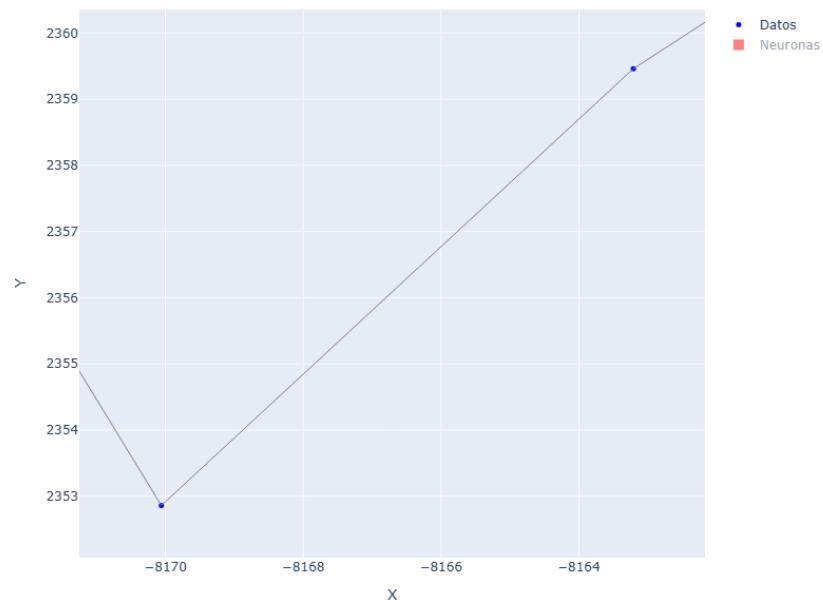


Figura 16: 200 ciudades 400 neuronas - dos ciudades representadas por tres neuronas, una queda en el medio

Kohonen 1D 200 ciudades

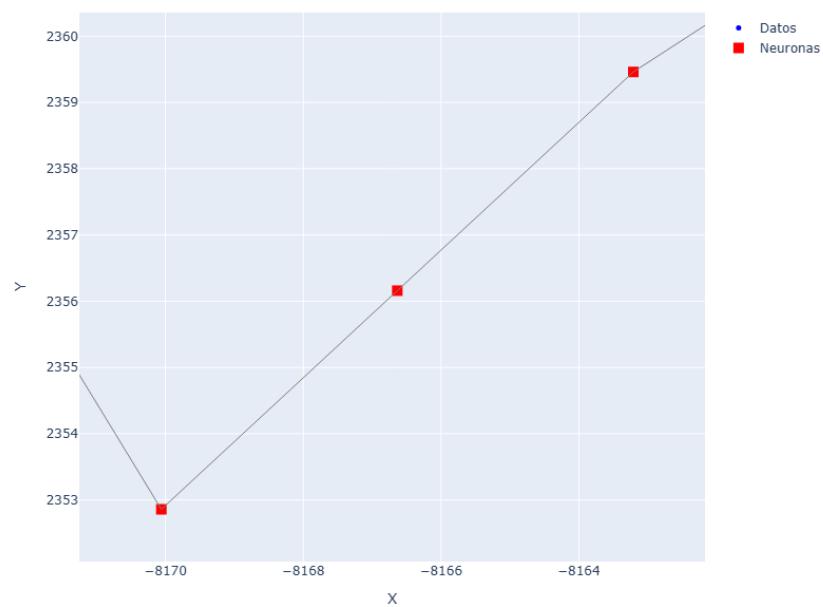


Figura 17: 200 ciudades 400 neuronas - se ve la neurona intermedia entre dos ciudades cercanas

Kohonen 1D 200 ciudades

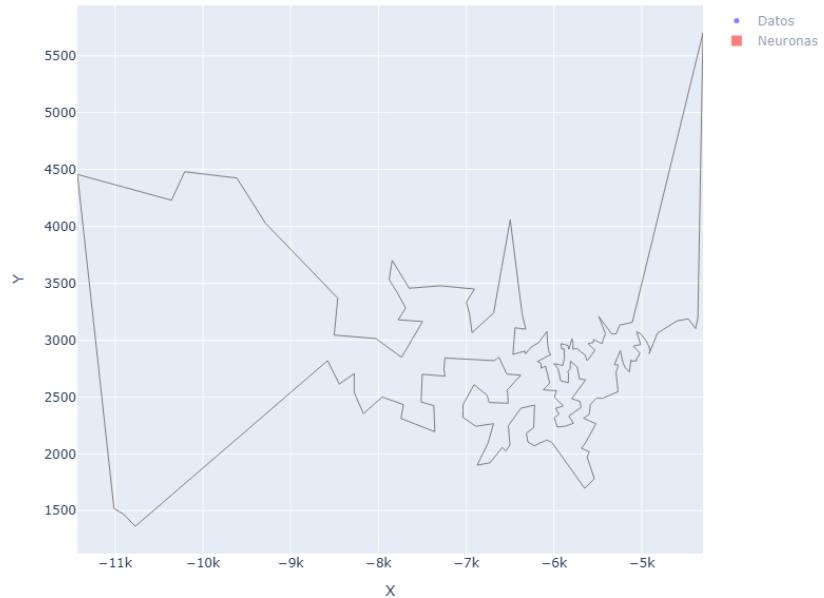


Figura 18: 200 ciudades 400 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 200 ciudades

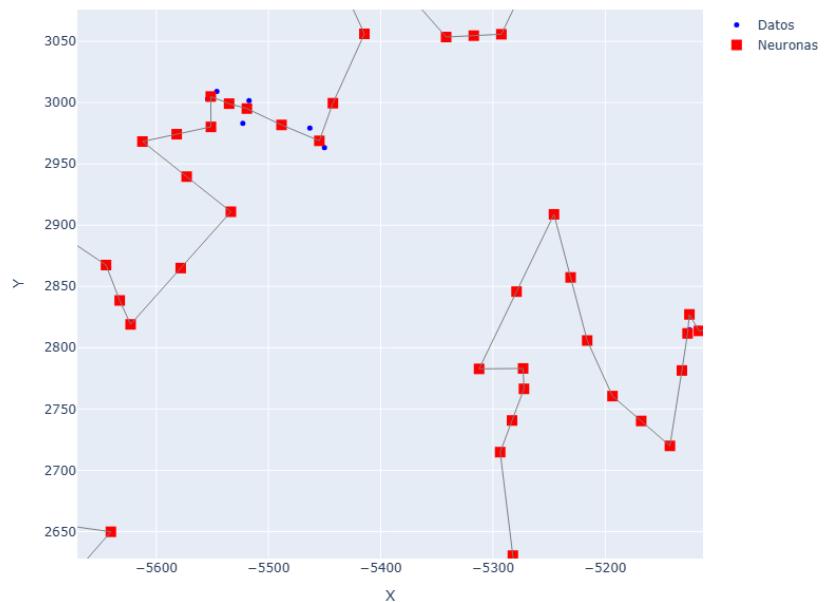


Figura 19: 200 ciudades 400 neuronas - zoom para más detalle

Kohonen 1D 200 ciudades

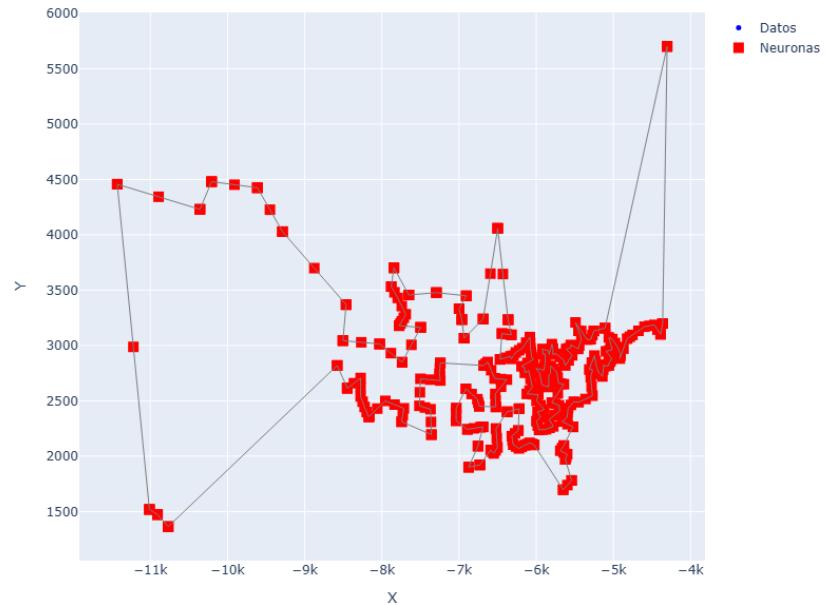


Figura 20: 200 ciudades 400 neuronas

Kohonen 1D 312 ciudades

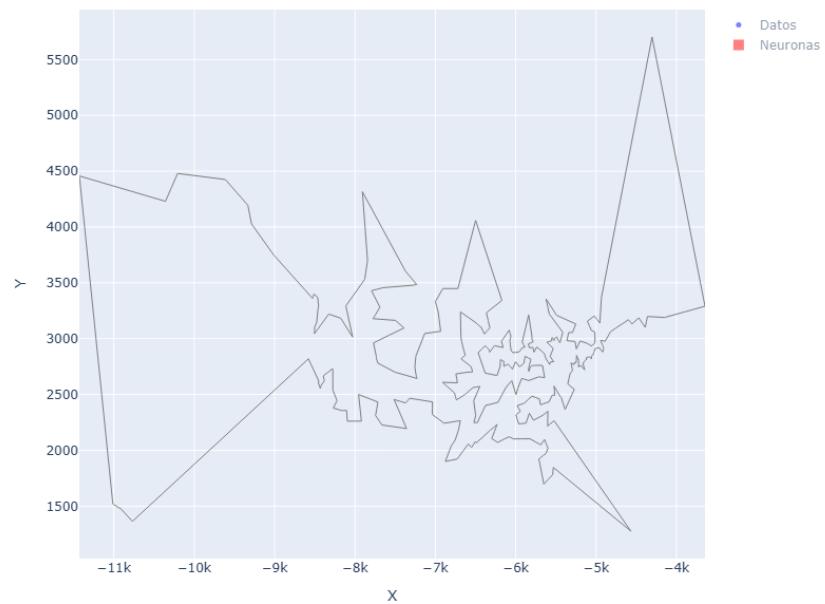


Figura 21: 312 ciudades 624 neuronas - el camino sin las neuronas o ciudades

Kohonen 1D 312 ciudades

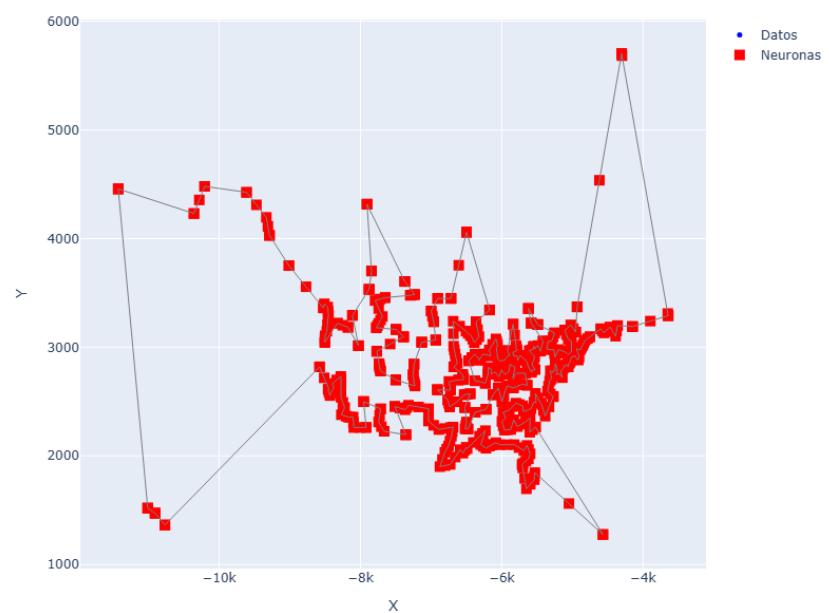


Figura 22: 312 ciudades 624 neuronas

## 5. Ejercicio 3

En el campus encontrará el archivo “datos\_para\_clustering.mat” que contiene una matriz de datos de 500 mediciones de una variable de 100 dimensiones.

- a) Utilice una red de Kohonen para reducir la dimensionalidad de los datos.
- b) Verifique la presencia de clusters, e indique cuantos puede visualizar, haciendo uso de la matriz U.

5.1. Consignas

5.2. Desarrollo

5.3. Análisis

## 6. Conclusiones

Cada consigna tiene su análisis, así que no es necesario en detalle sobre algún ejercicio particular. Considero que el trabajo sirvió como una introducción extremadamente completa los temas de redes neuronales basadas en perceptrones, métodos de optimización y redes convolucionales. A mi entender, las consignas fueron resueltas de forma satisfactoria, ahondando en algunos temas no requeridos pero que hacen al entendimiento de la materia.