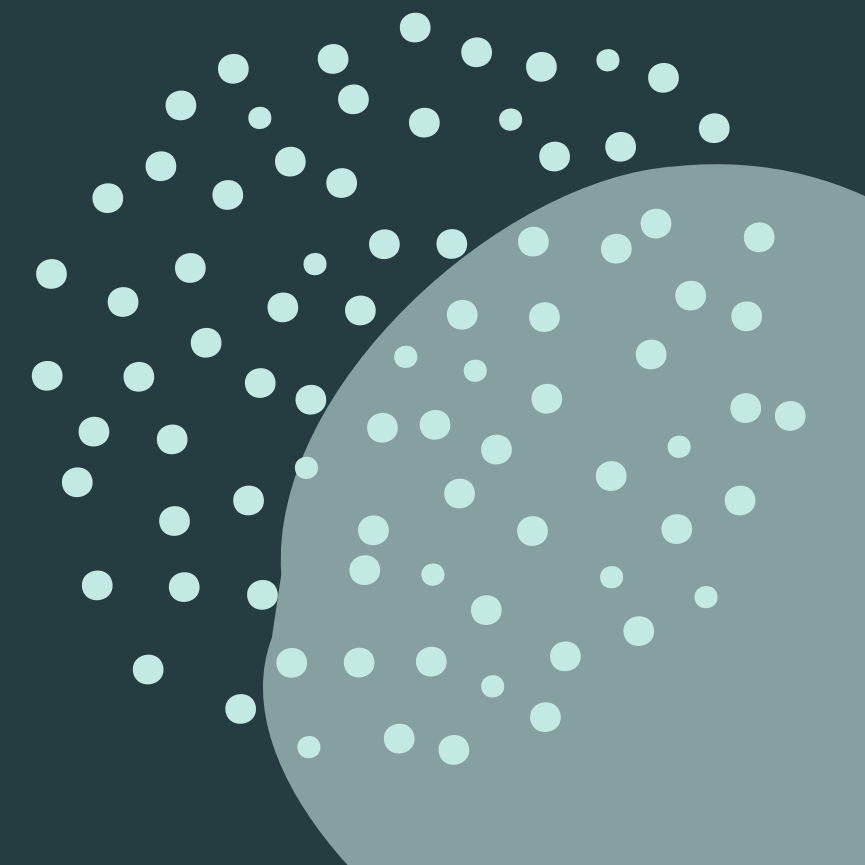


TP5: AUTOENCODERS

Agustin Spitzner

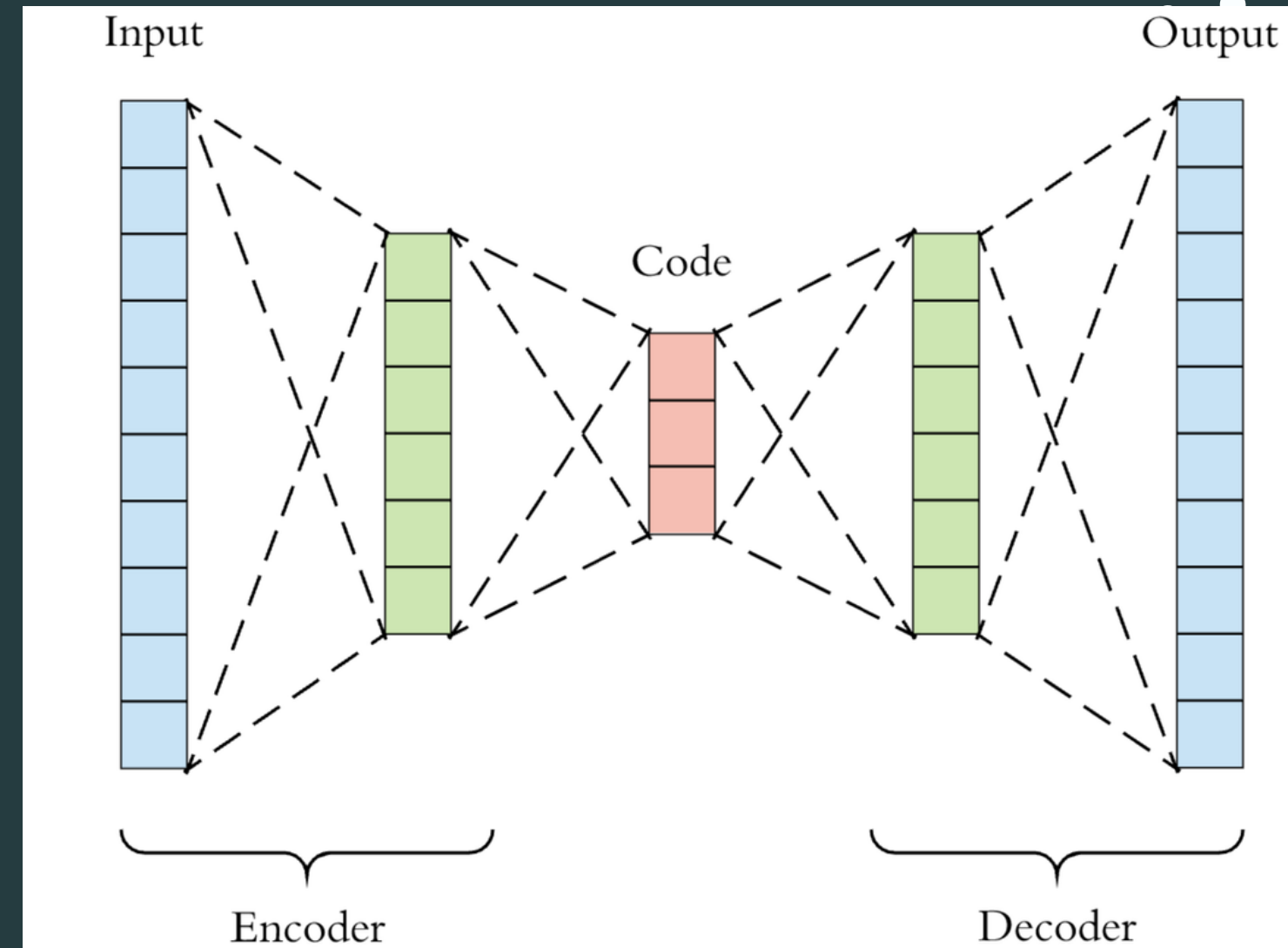
Ana Cruz

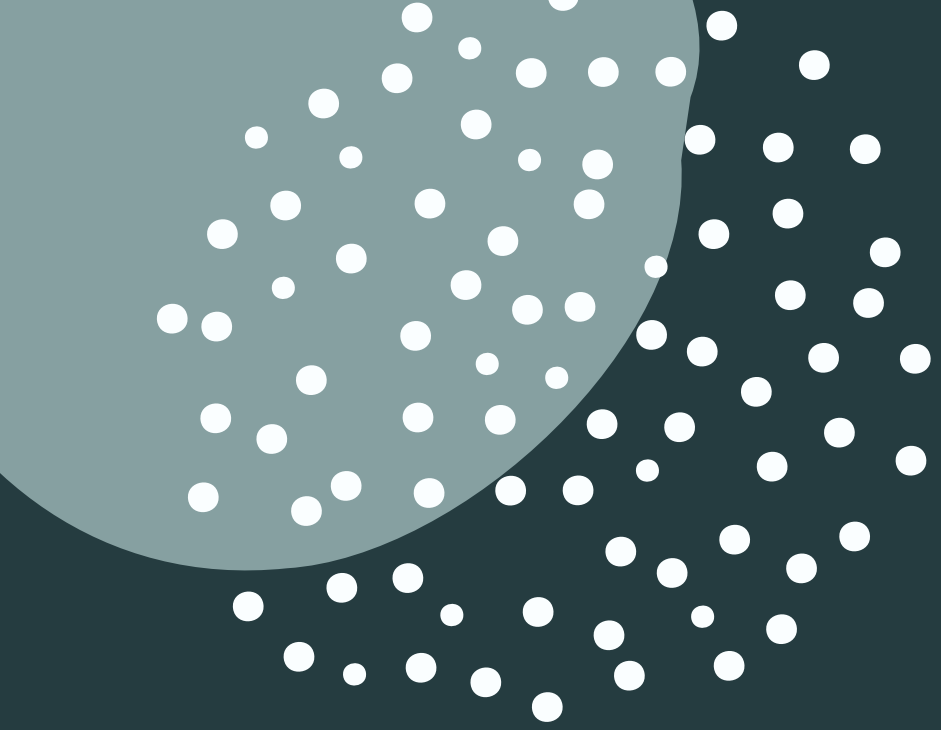
Camila Borinsky



PROBLEMA 1

Implementar un Autoencoder básico para las imágenes binarias de la lista de caracteres del archivo "font.h"





Arquitecturas para probar:

[35, 2, 35]

[35, 17, 2, 17, 35]

[35, 17, 8, 2, 8, 17, 35]

[35, 25, 17, 2, 17, 25, 35]

[35, 17, 8, 4, 2, 4, 8, 17, 35]

[35, 25, 17, 8, 4, 2, 4, 8, 17, 25 35]

ENCONTRAR UNA ARQUITECTURA

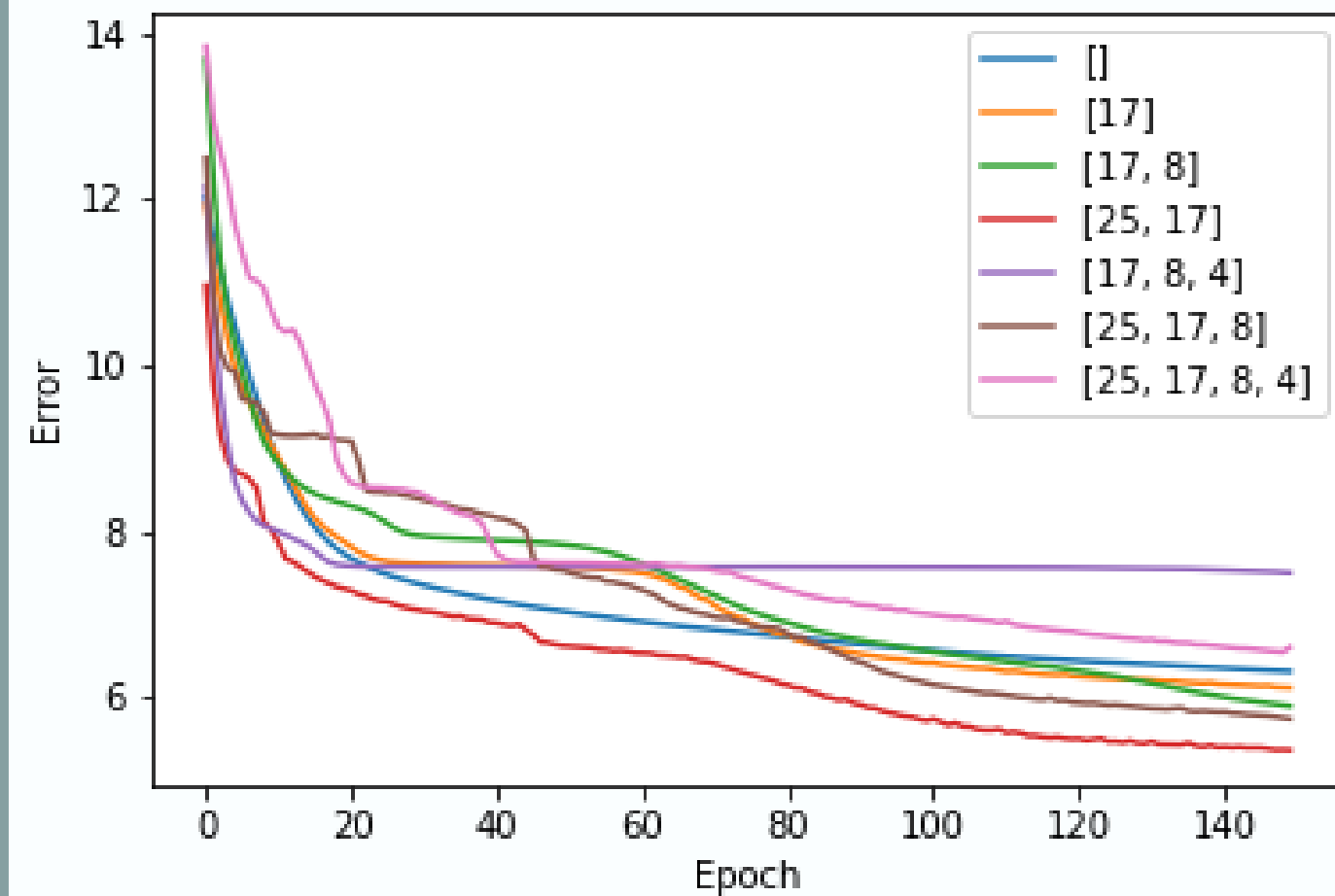
Arquitectura de red para lograr
codificar los datos a 2 dimensiones

Factores a observar:

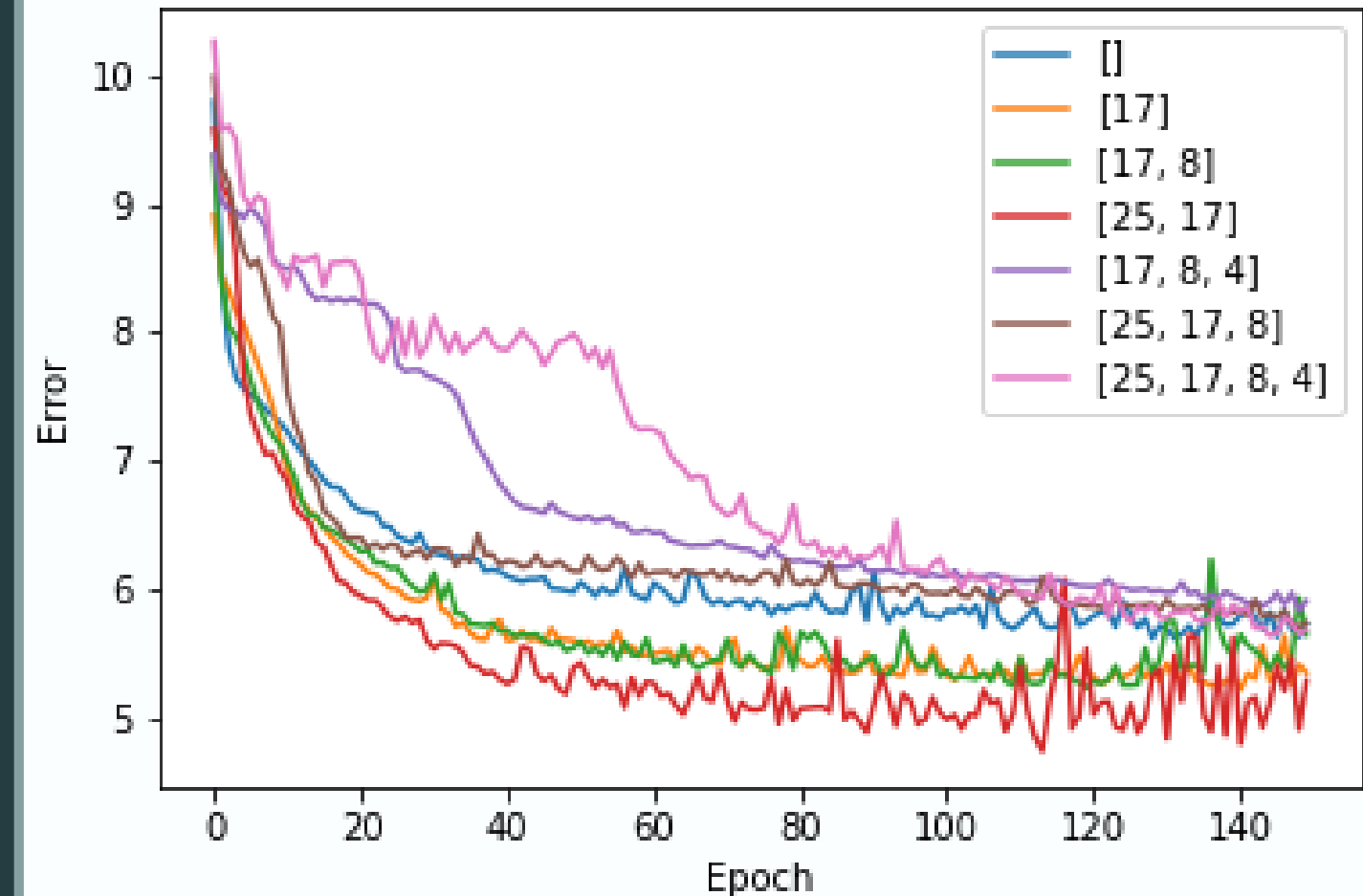
Cantidad de capas

Salto entre dimensiones de las
capas

El objetivo es que la dimensión del
latente sea 2.



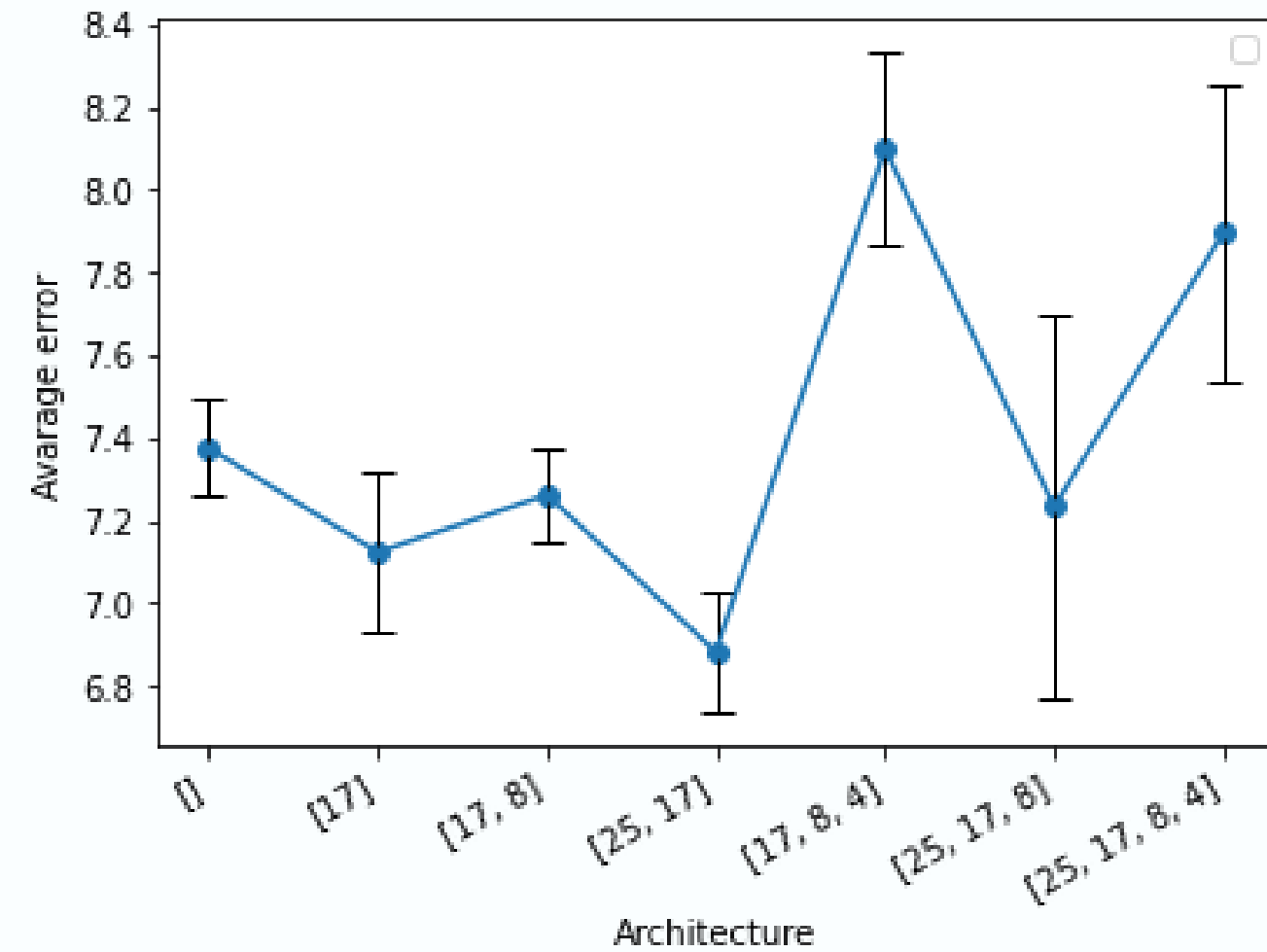
sin momentum



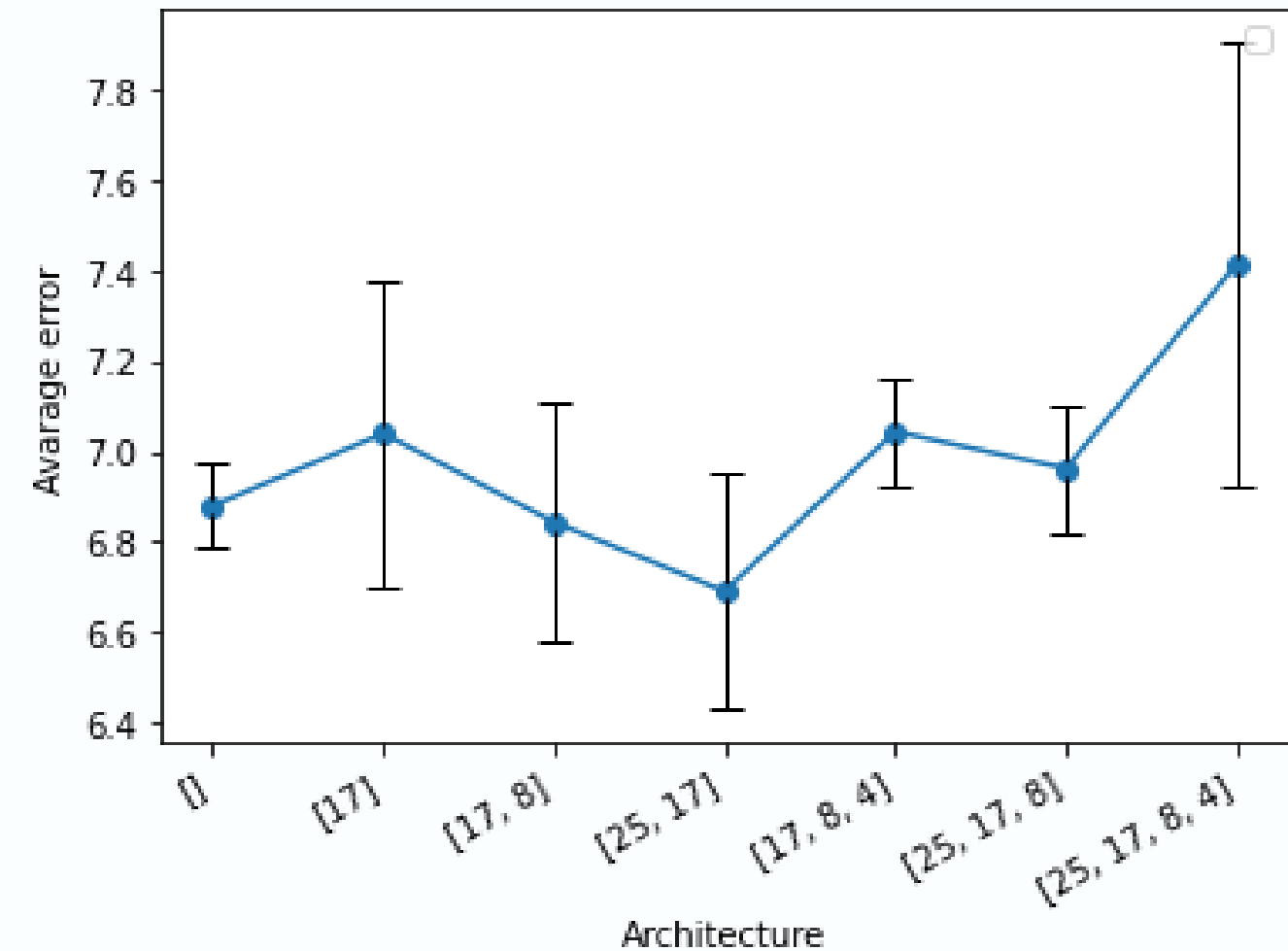
con momentum

COMPARANDO ARQUITECTURAS: ERROR VS ÉPOCAS

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones



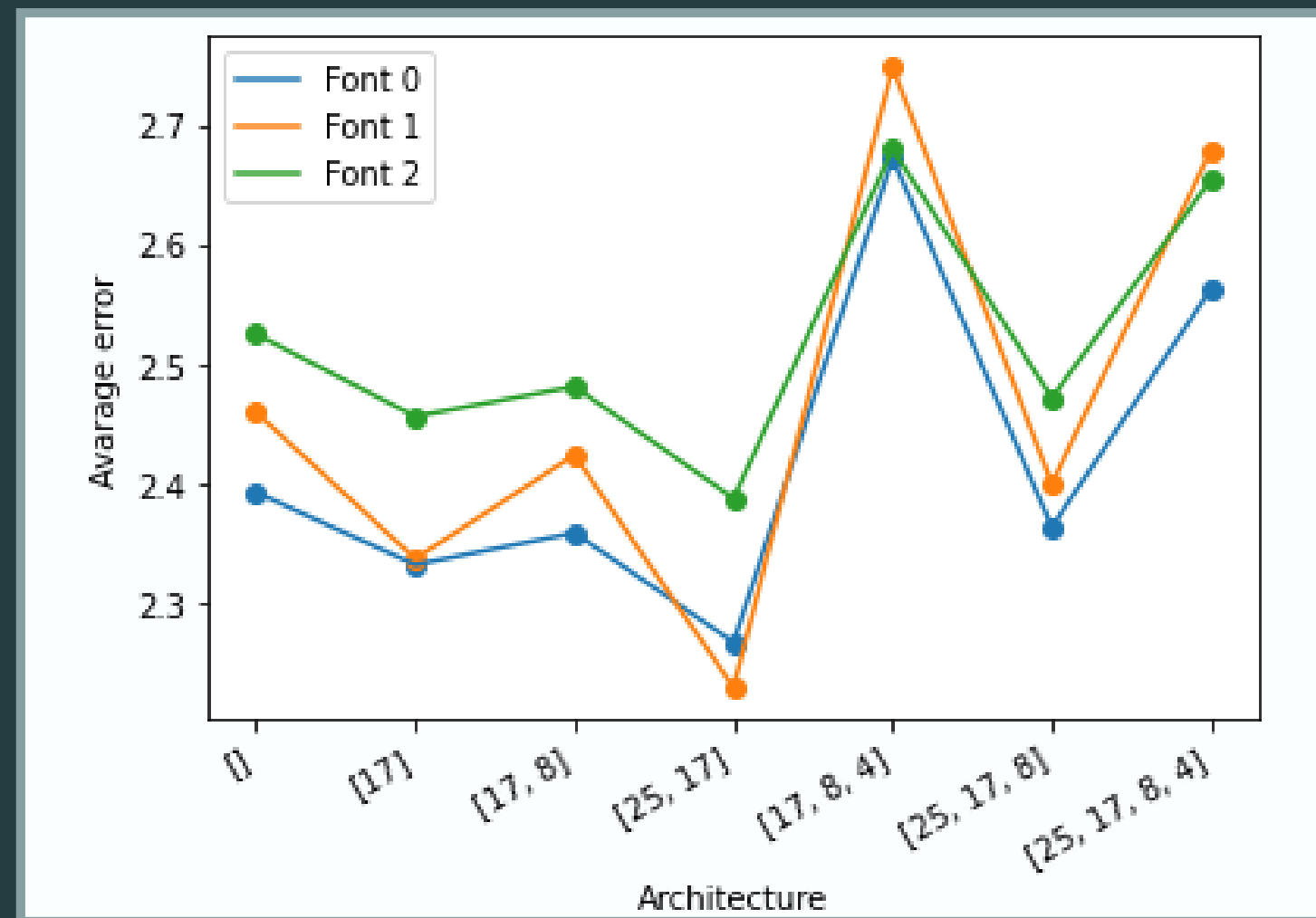
sin momentum



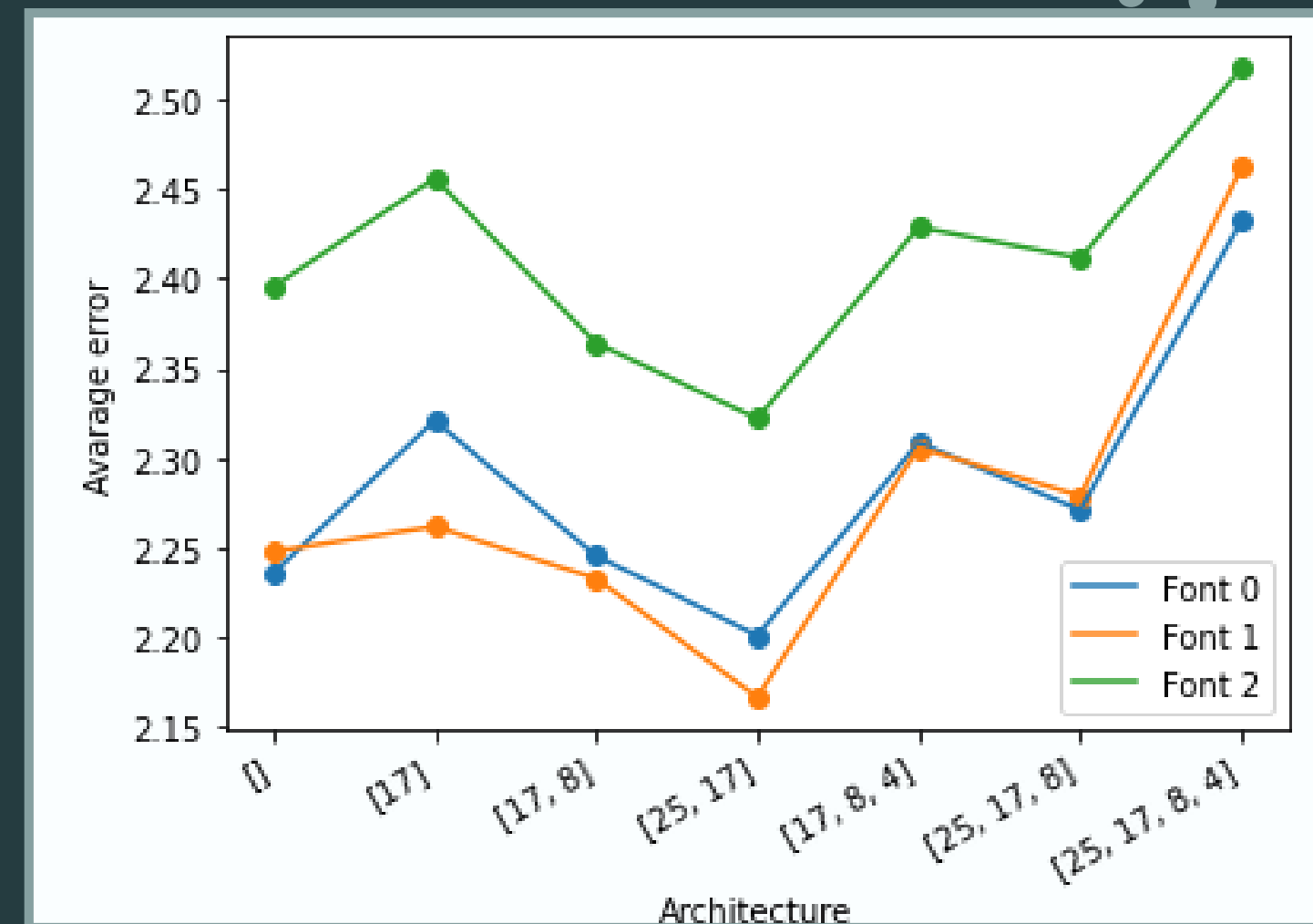
con momentum

**COMPARANDO
ARQUITECTURAS:
ERROR PROMEDIO**

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones



sin momentum

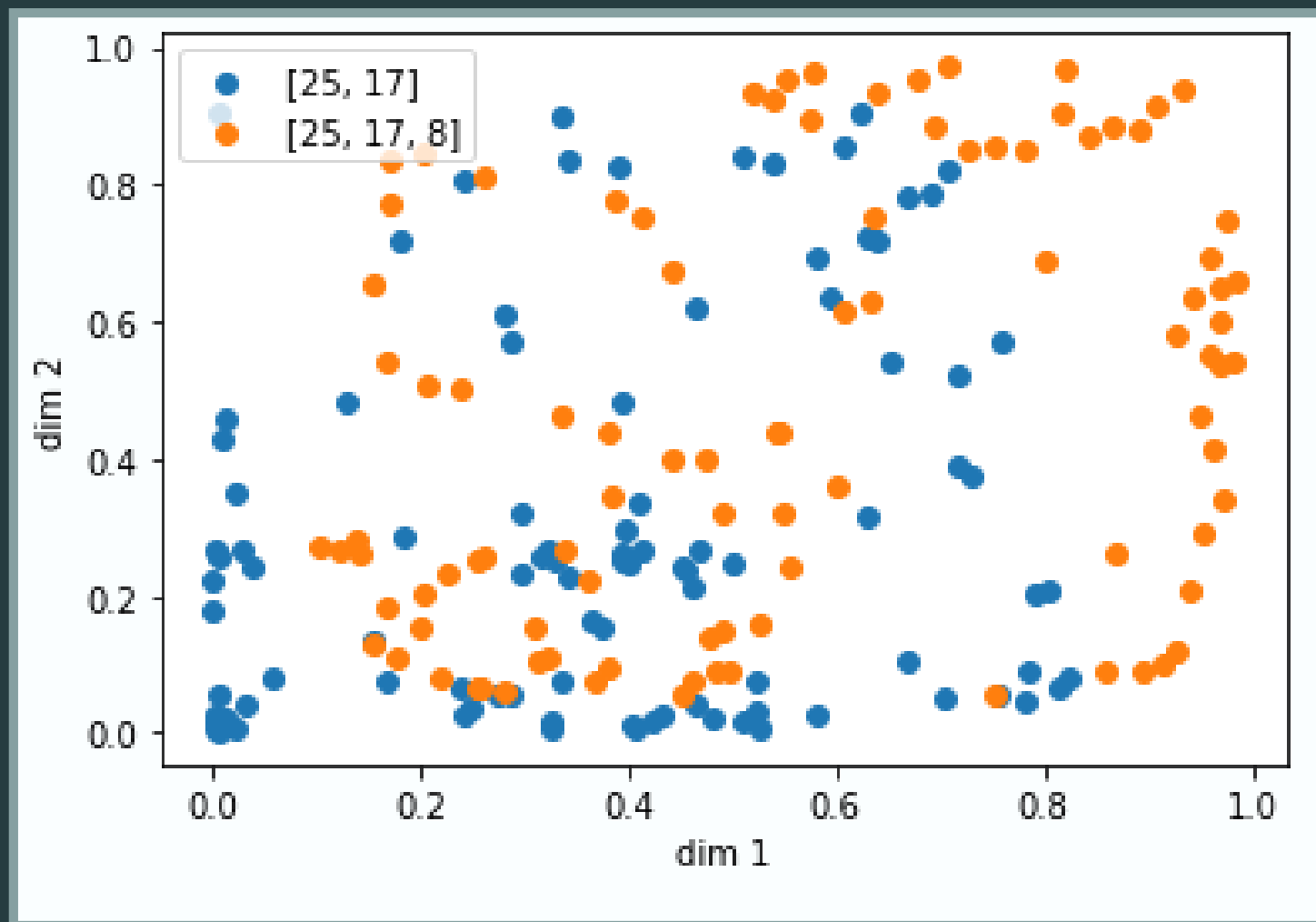


con momentum

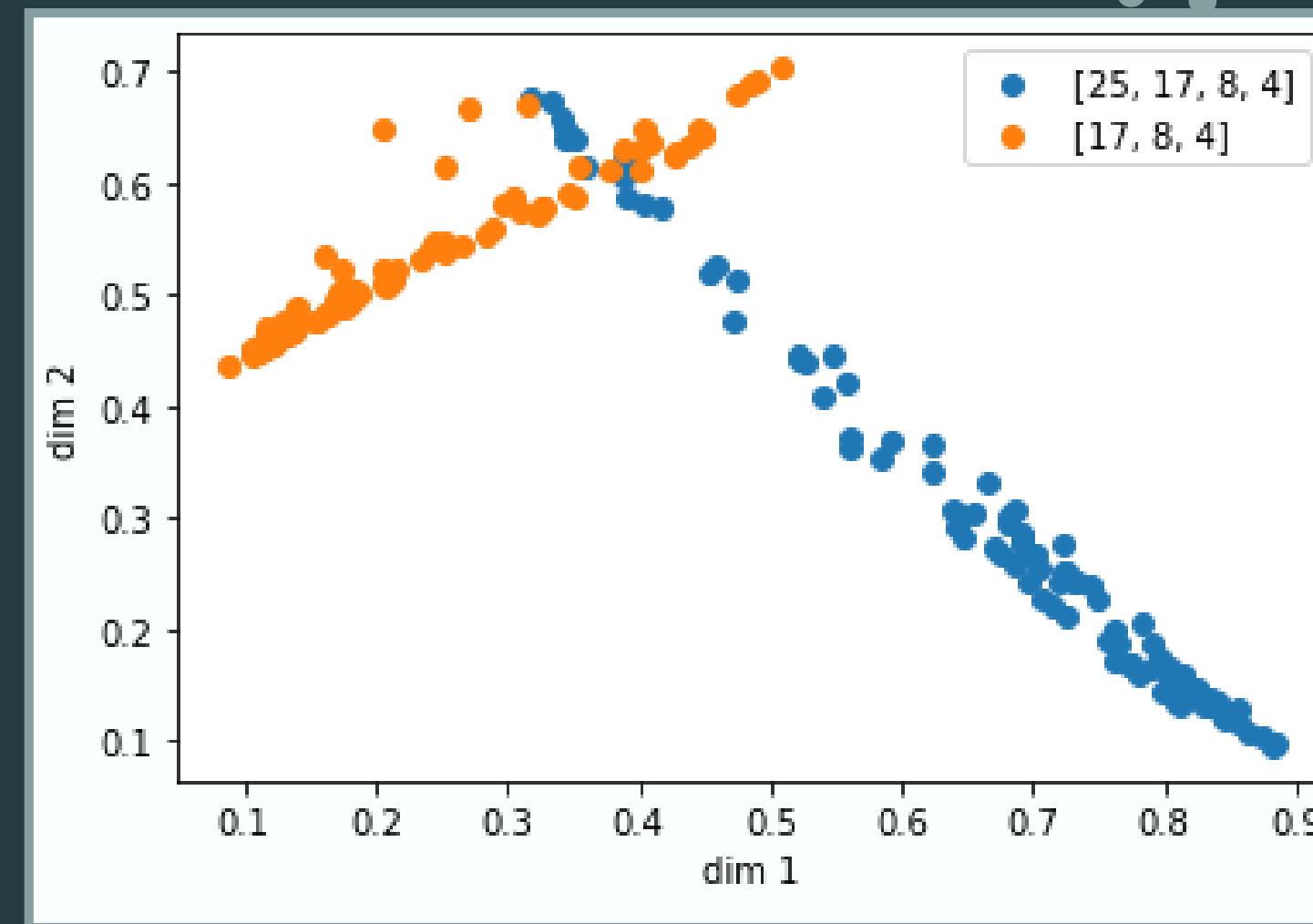
Font 0: símbolos y dígitos
Font 1: mayúsculas
Font 2: minúsculas

**COMPARANDO
ARQUITECTURAS: ERROR
PROMEDIO POR FONT**

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones



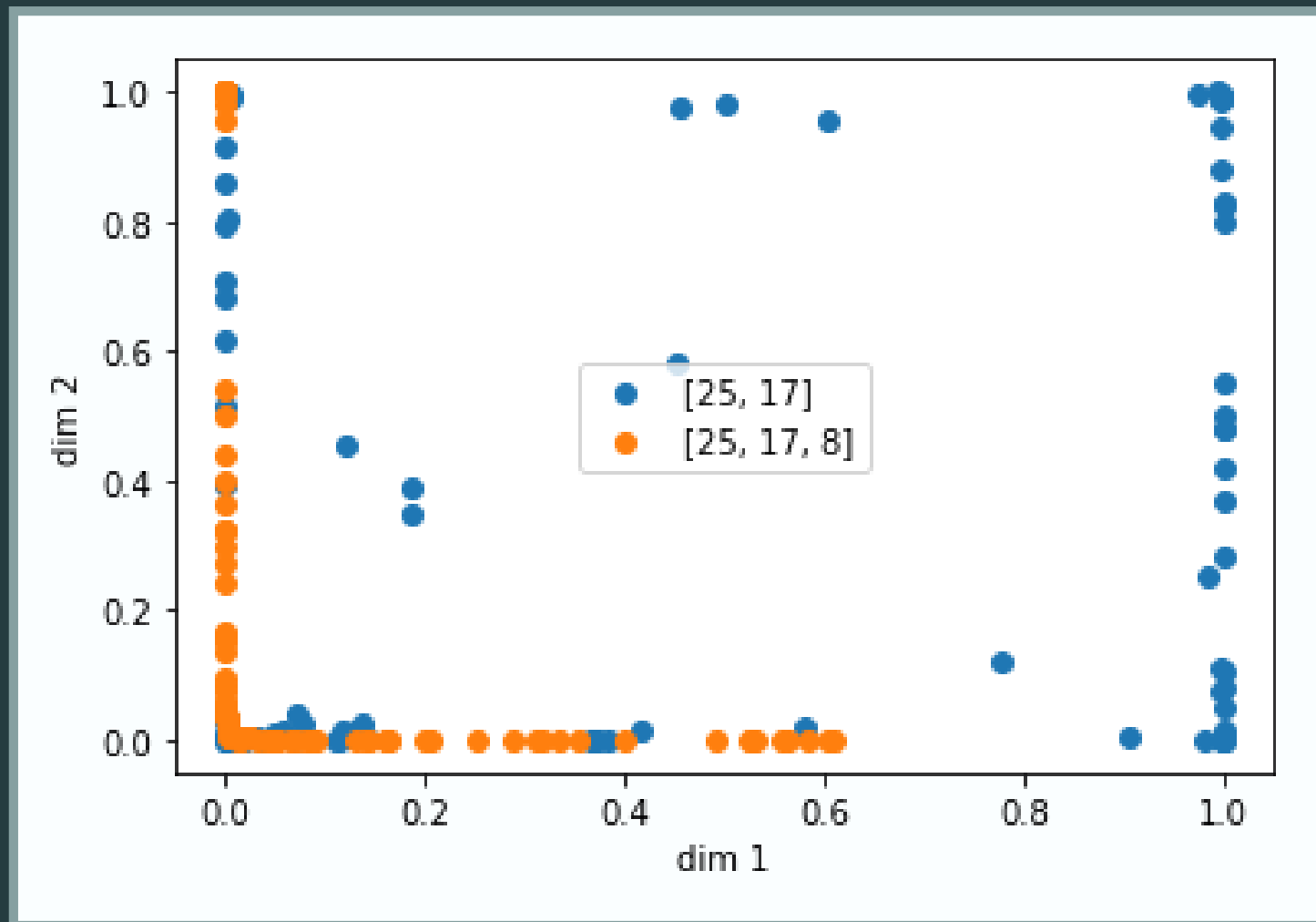
Arquitecturas con menor error



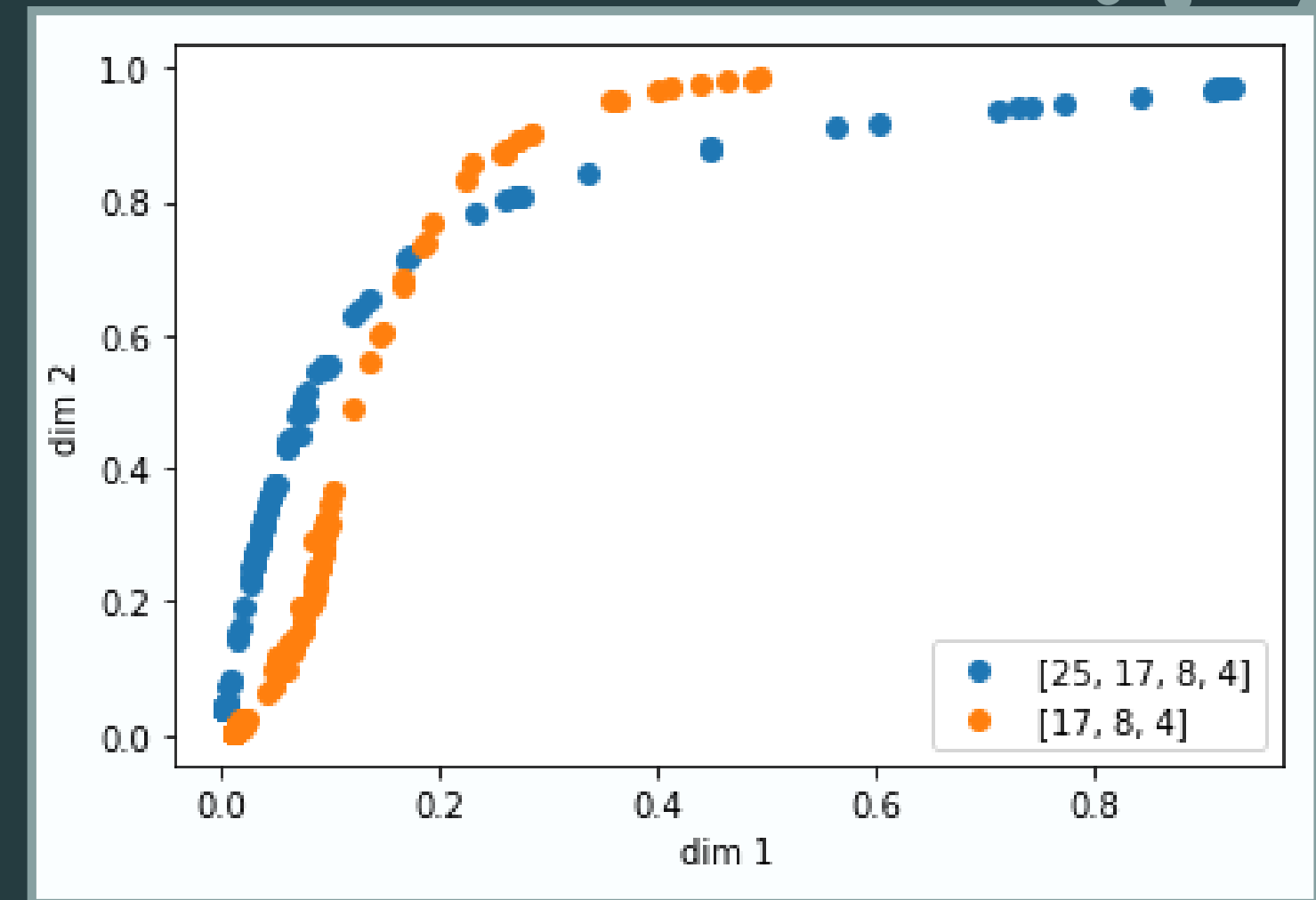
Arquitecturas con mayor error

COMPARANDO ARQUITECTURAS: ESPACIOS LATENTES (SIN MOMENTUM)

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones



Arquitecturas con menor error

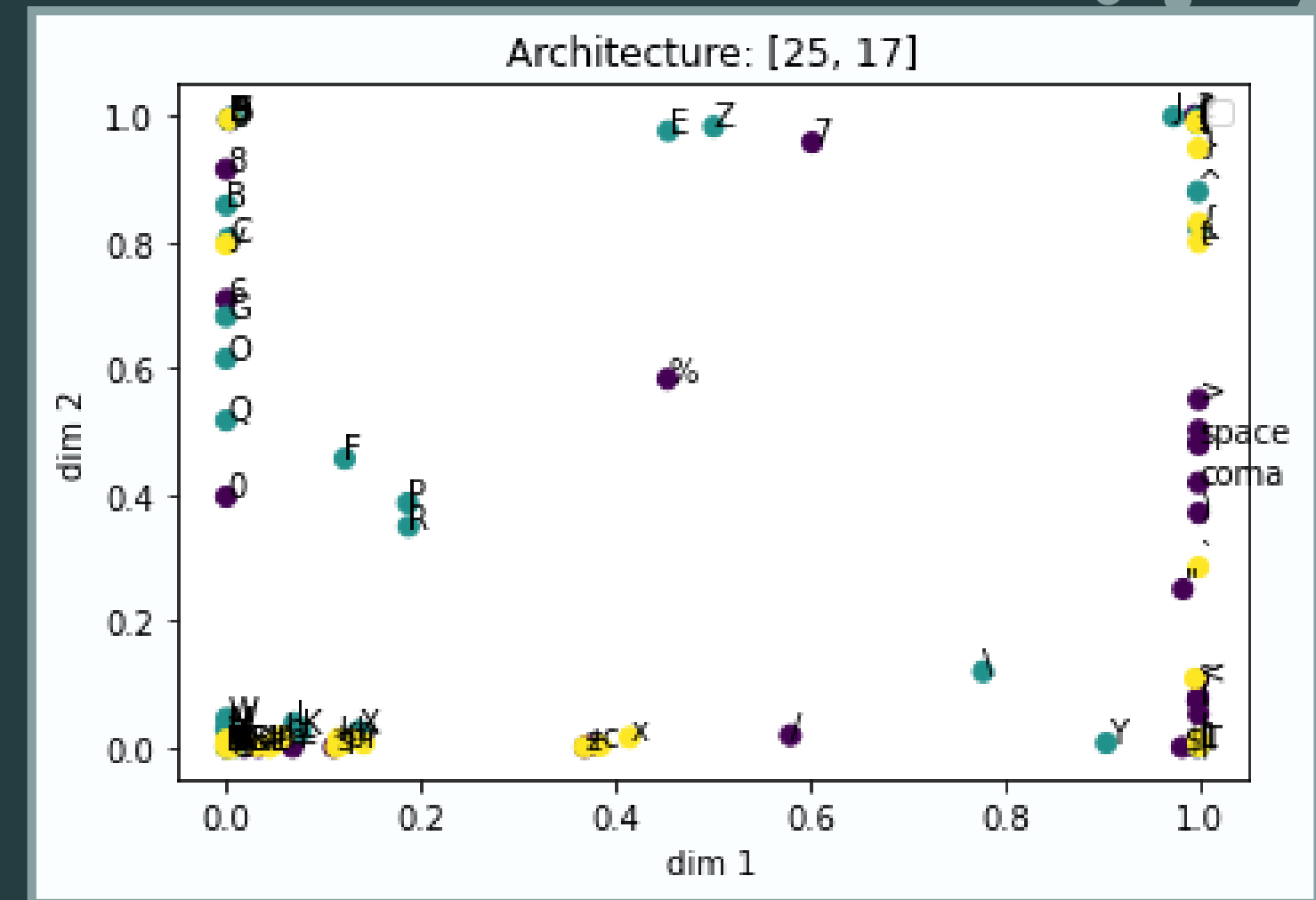
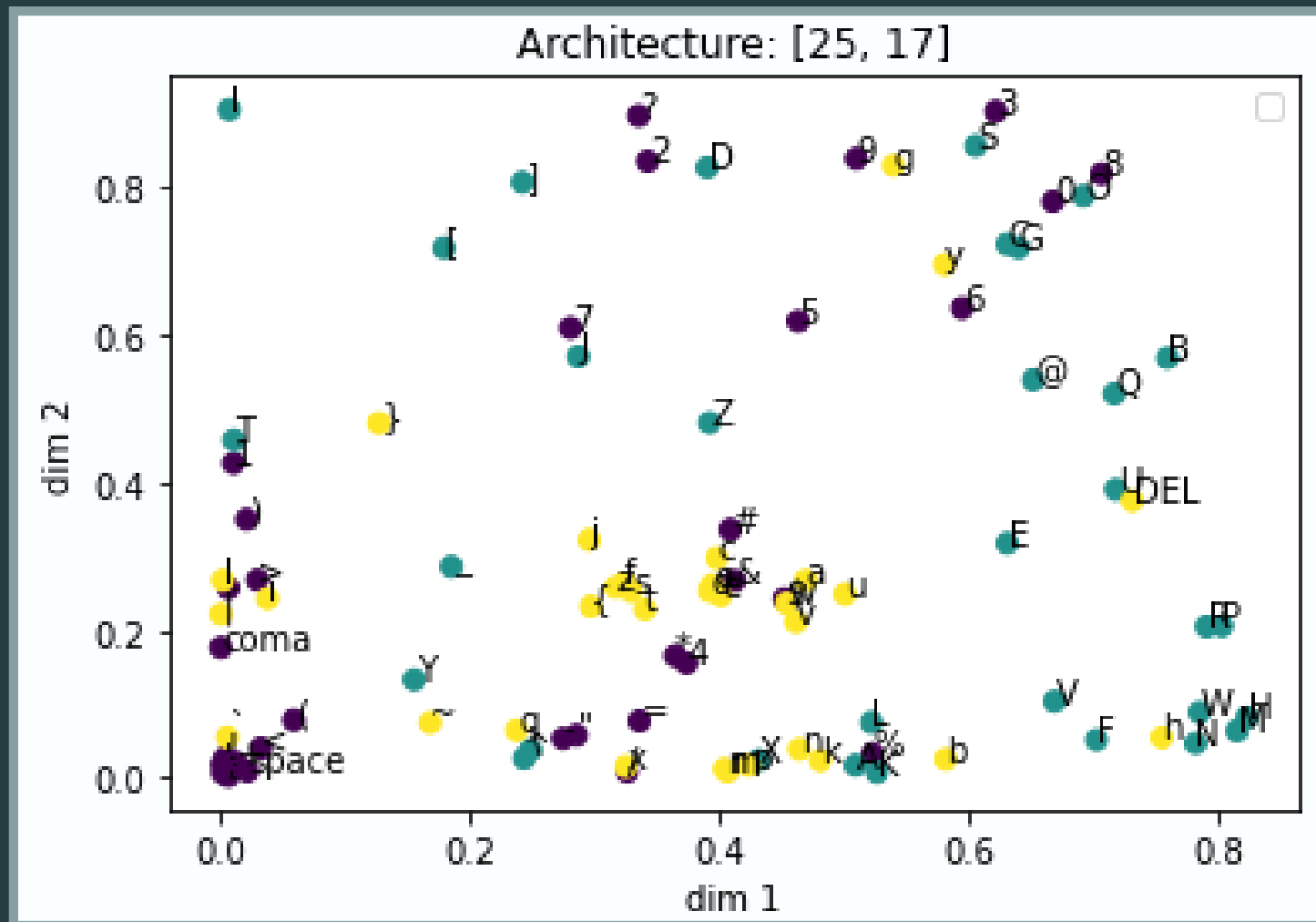


Arquitecturas con mayor error

Utilizando todos los
fonts

**COMPARANDO
ARQUITECTURAS:
ESPACIOS LATENTES (CON
MOMENTUM)**

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones

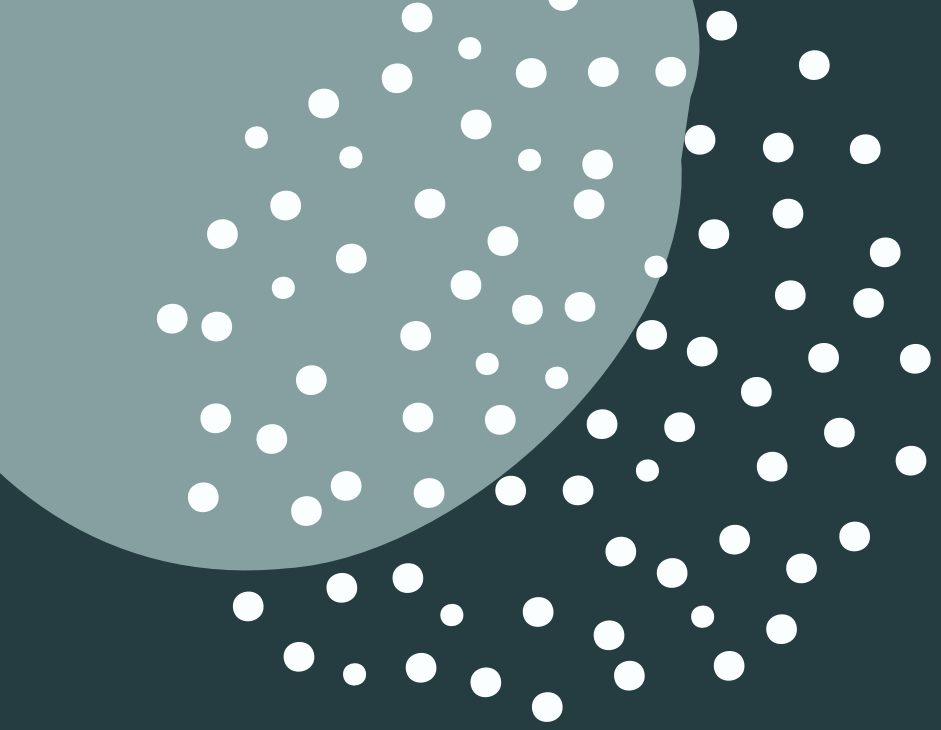


Sin momentum

Con momentum

MEJOR ARQUITECTURA: ESPACIO LATENTE

épocas: 150
tasa de aprendizaje: 0.05
Promedio 5 ejecuciones



Con y sin momentum la comparación de arquitecturas quedó similar.

Con momentum se obtuvo error más bajo.

Las fonts se comportan similar a pesar de contener símbolos distintos

ENCONTRAR UNA ARQUITECTURA

Conclusiones

Mejor arquitectura:
[35, 25, 17, 2, 17, 25, 35]

Peor arquitectura:

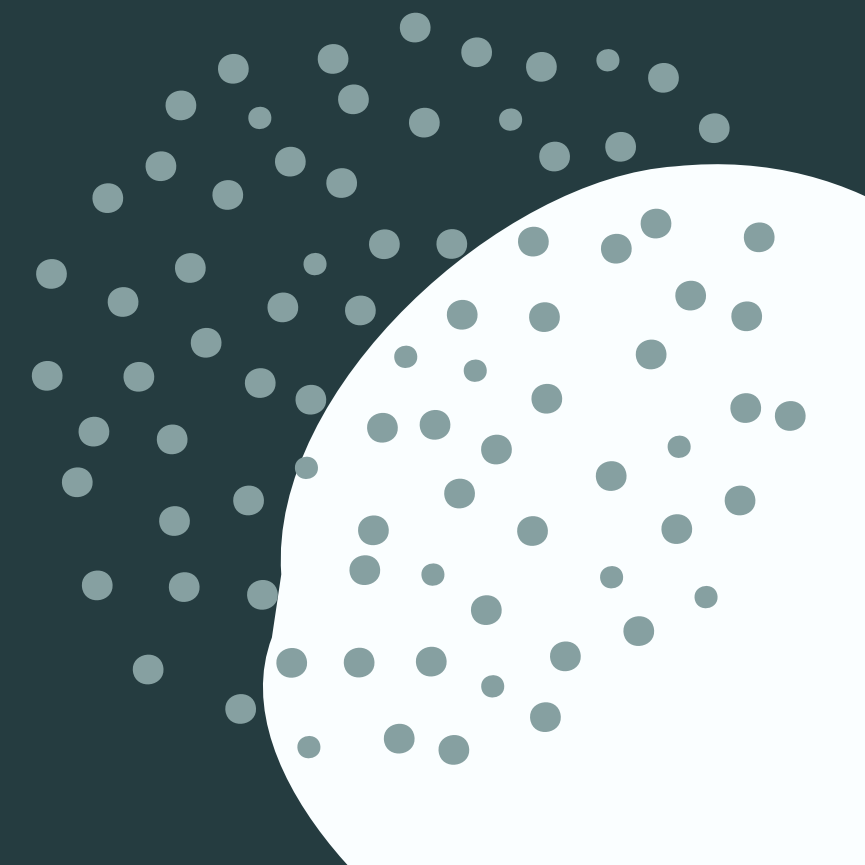
- con momentum: [35, 17, 8, 4, 2, 4, 8, 17, 35]
- sin momentum: [35, 25, 17, 8, 4, 2, 4, 8, 17, 35]

ENCONTRAR UN OPTIMIZADOR

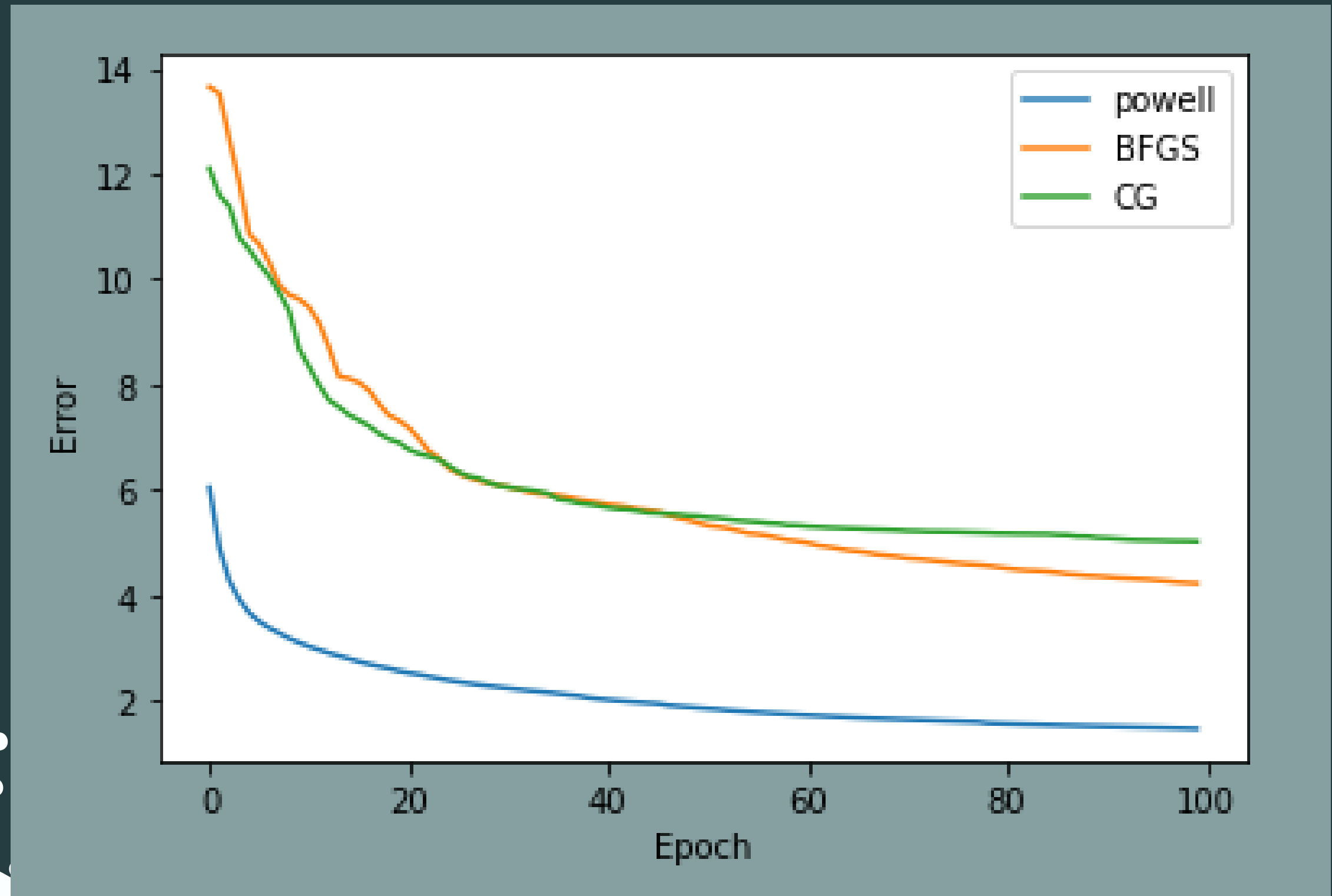
Usamos la librería scipy para evaluar distintos optimizadores

Optimizadores para probar:

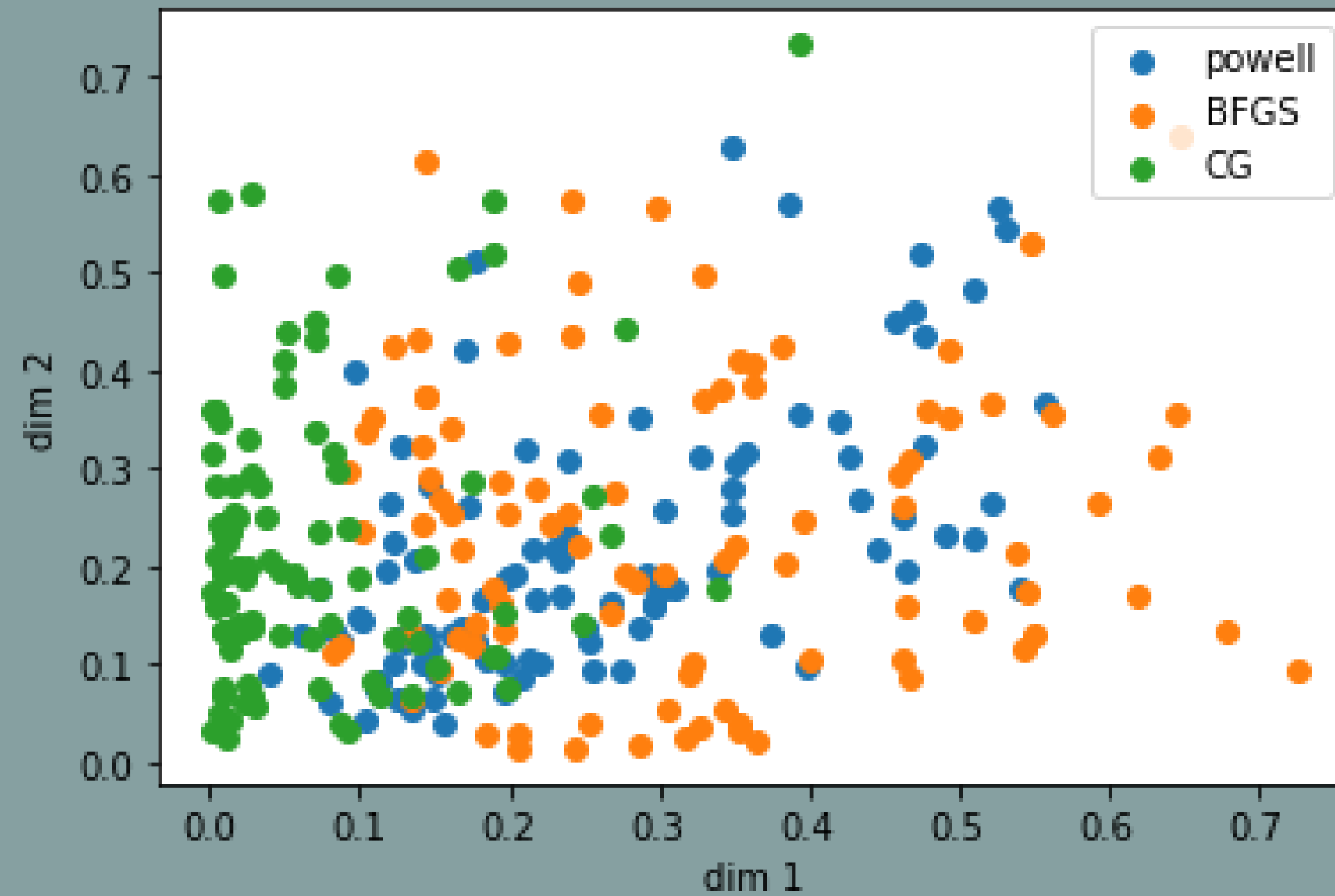
- Powell: Metodo de Powell Modificado
- BFGS: Método de BFGS (Broyden et al). Utiliza solo la primera derivada. Devuelve una aproximación del inverso del Hessiano
- GC: método de gradientes conjugadas. Utiliza solo la primera derivada



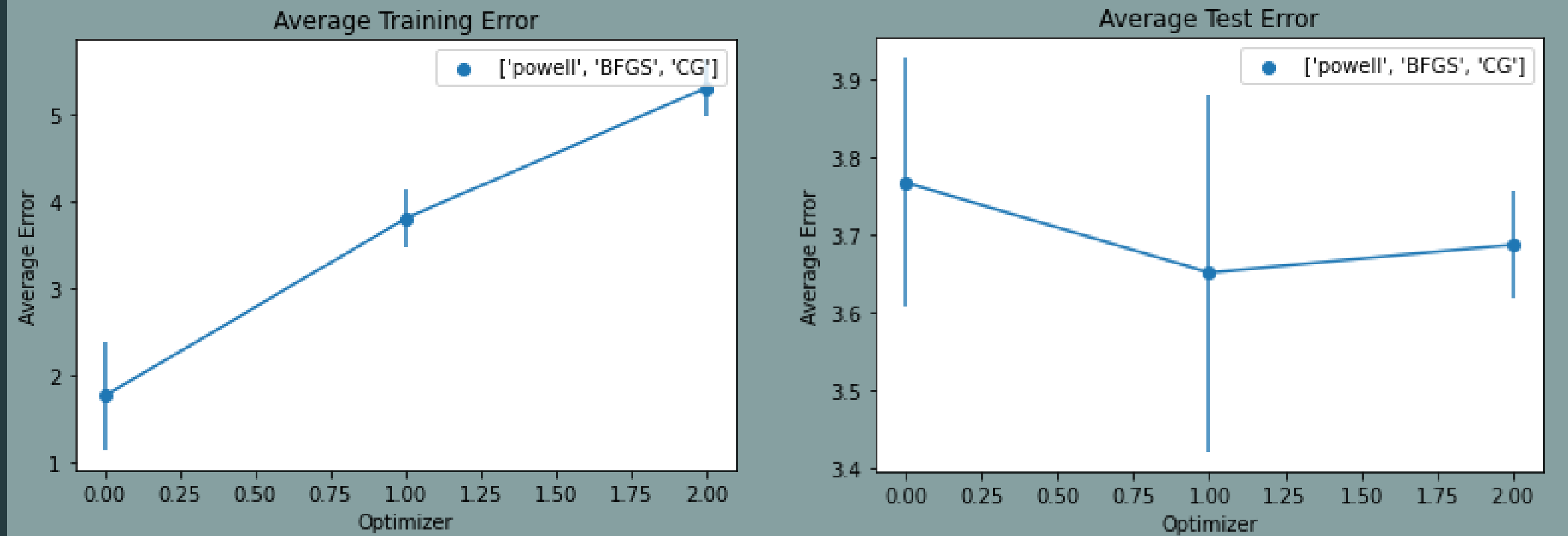
Comparación de métodos de optimización



Capas latentes

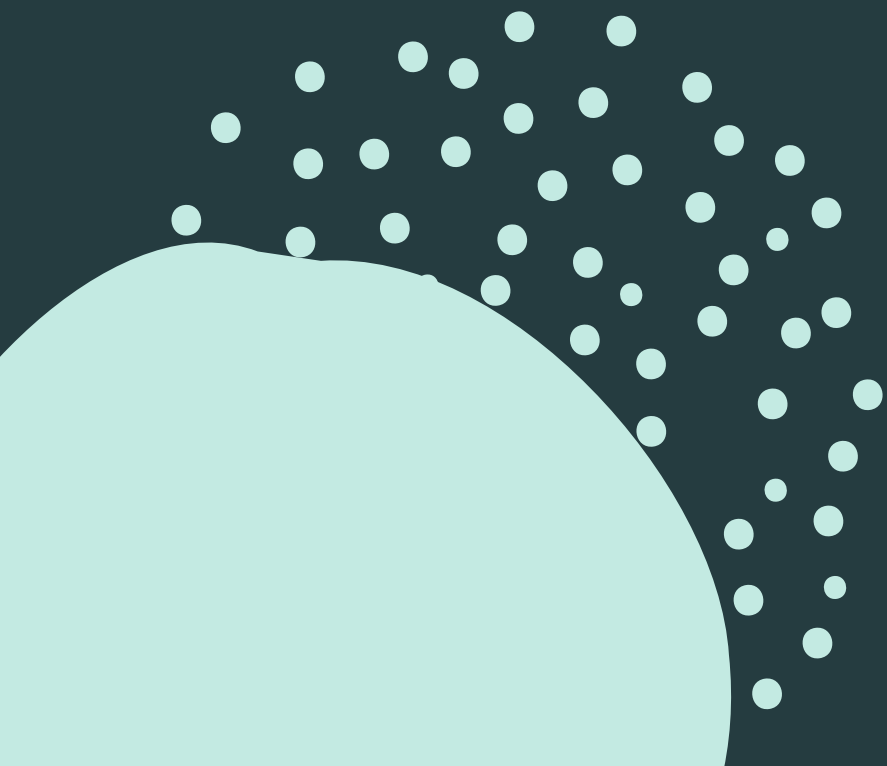


Error en training y testing



GENERAR CONTENIDO

Exploramos el espacio latente



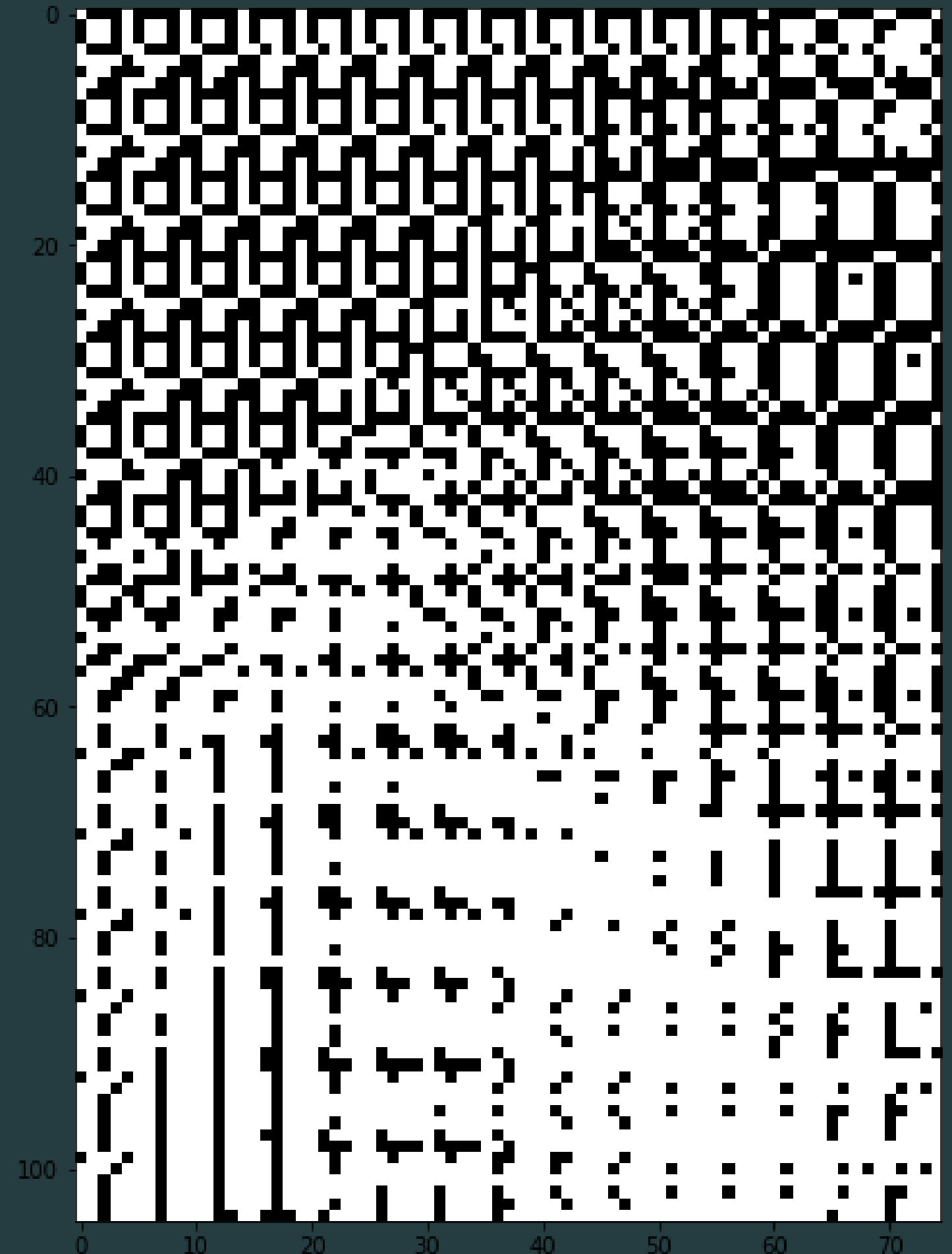
MUESTRA DE ENTRENAMIENTO: 50%

optimizador: Powell

arquitectura: [35, 25, 17, 2, 17, 25, 35]

iteraciones del optimizador: 20

error de entrenamiento: 3.6054



MUESTRA DE ENTRENAMIENTO: 30%

optimizador: Powell

arquitectura: [35, 25, 17, 2, 17, 25, 35]

iteraciones del optimizador: 20

error de entrenamiento: 0.4575



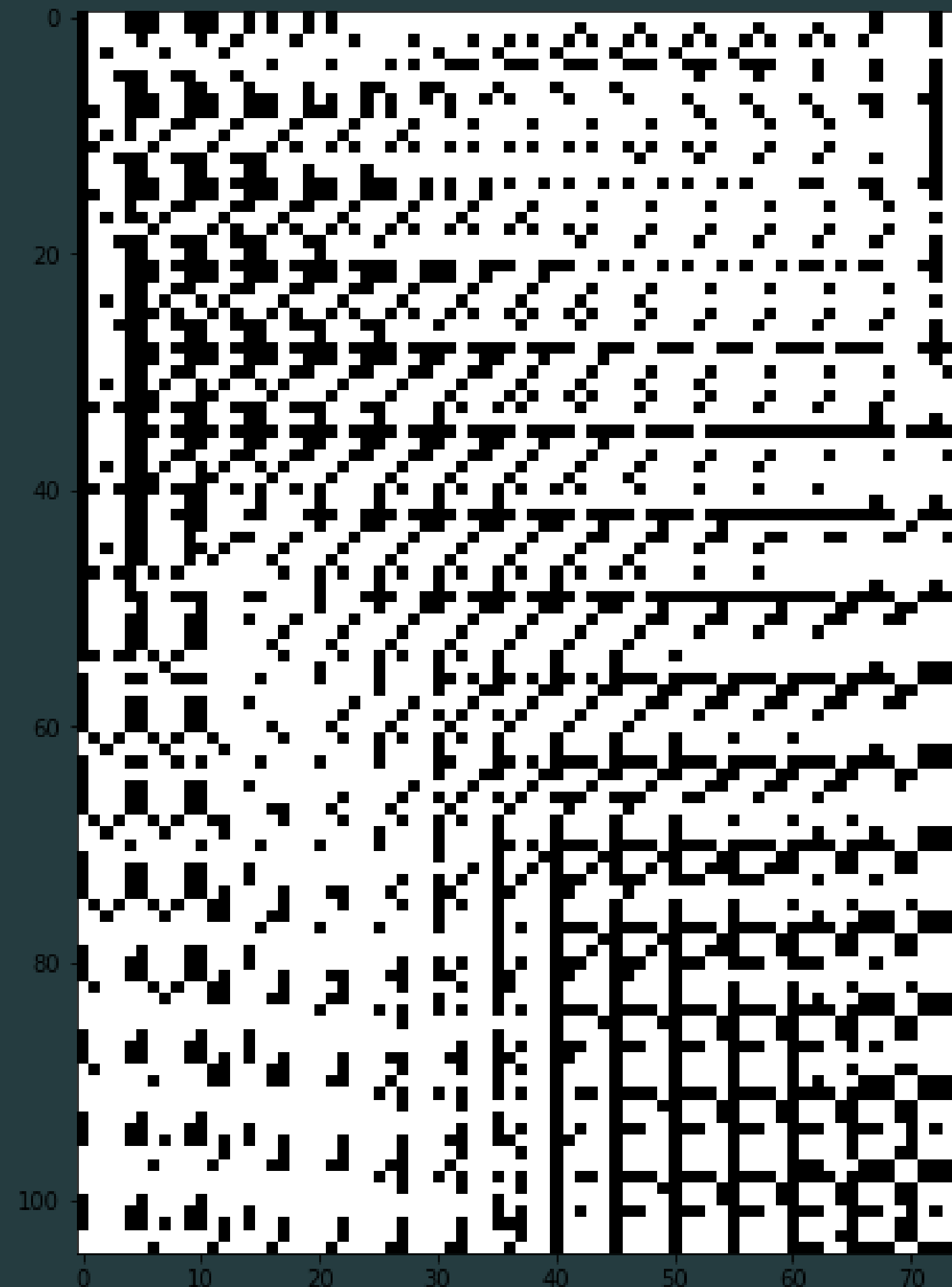
MUESTRA DE ENTRENAMIENTO: 70%

optimizador: Powell

arquitectura: [35, 25, 17, 2, 17, 25, 35]

iteraciones del optimizador: 20

error de entrenamiento: 1.8284





Arquitectura

[35, 25, 17, 5, 17, 25, 35]

Optimizador

Powell

Max Iteraciones: 20

Ruido

$p = 0.1$

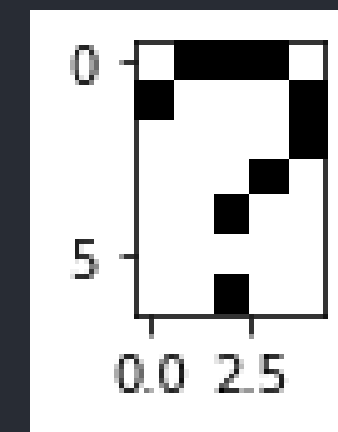
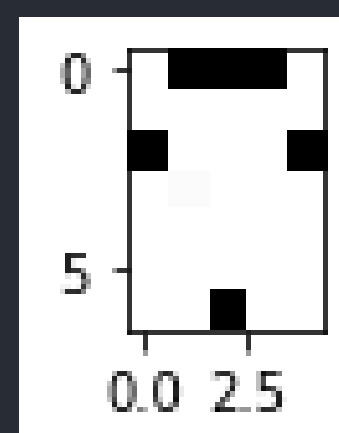
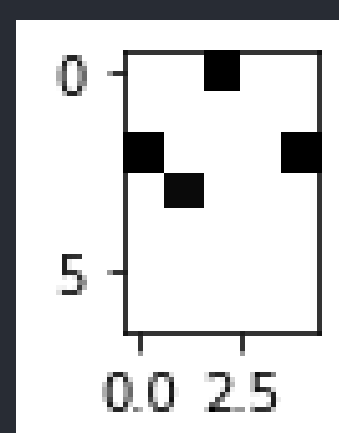
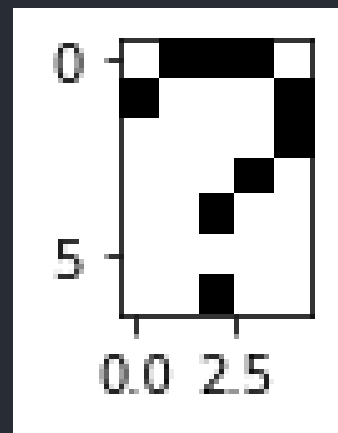
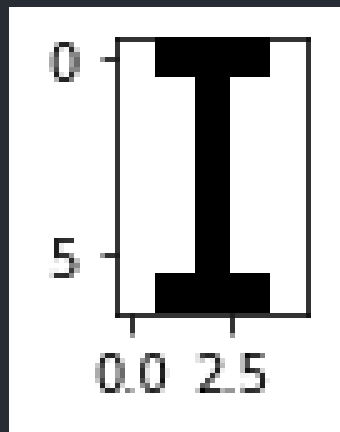
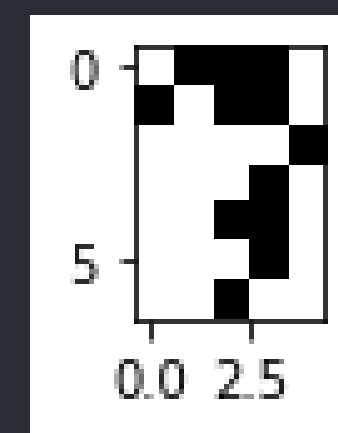
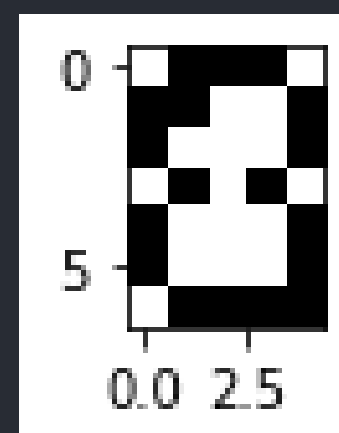
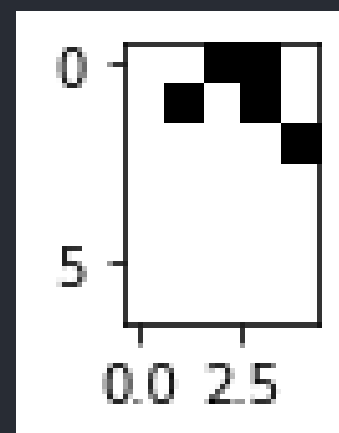
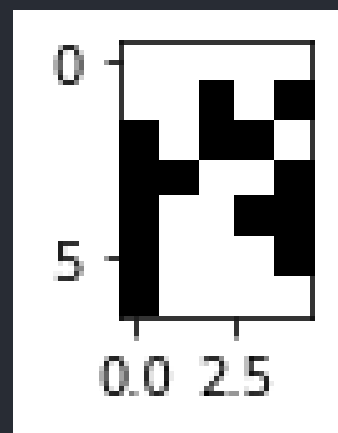
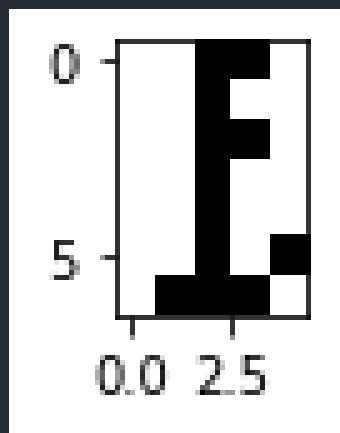
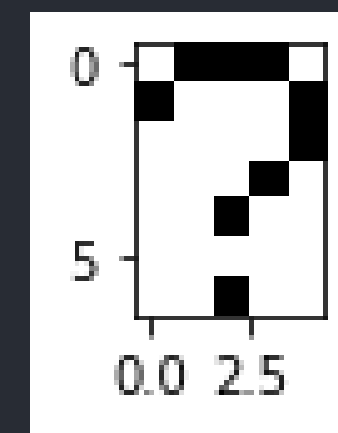
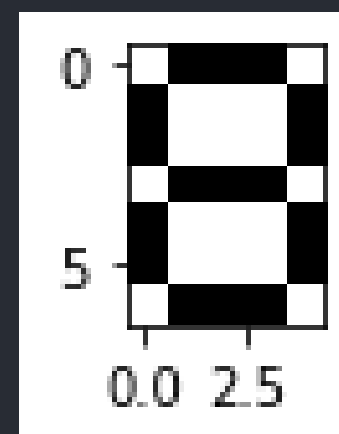
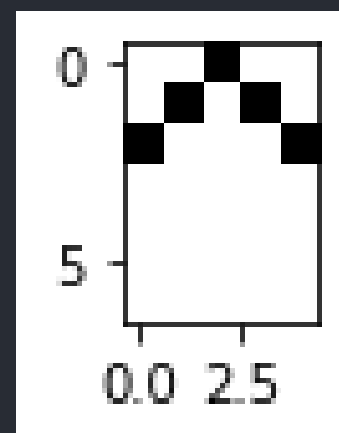
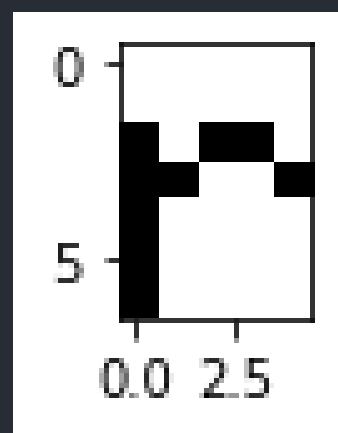
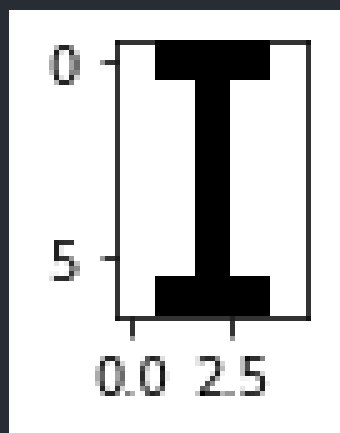
muestras con ruido por caracter: 4

Entrenamiento

2 caracteres de cada font



DENOISING AUTOENCODER



Error: 1E-54

Error: 4.24

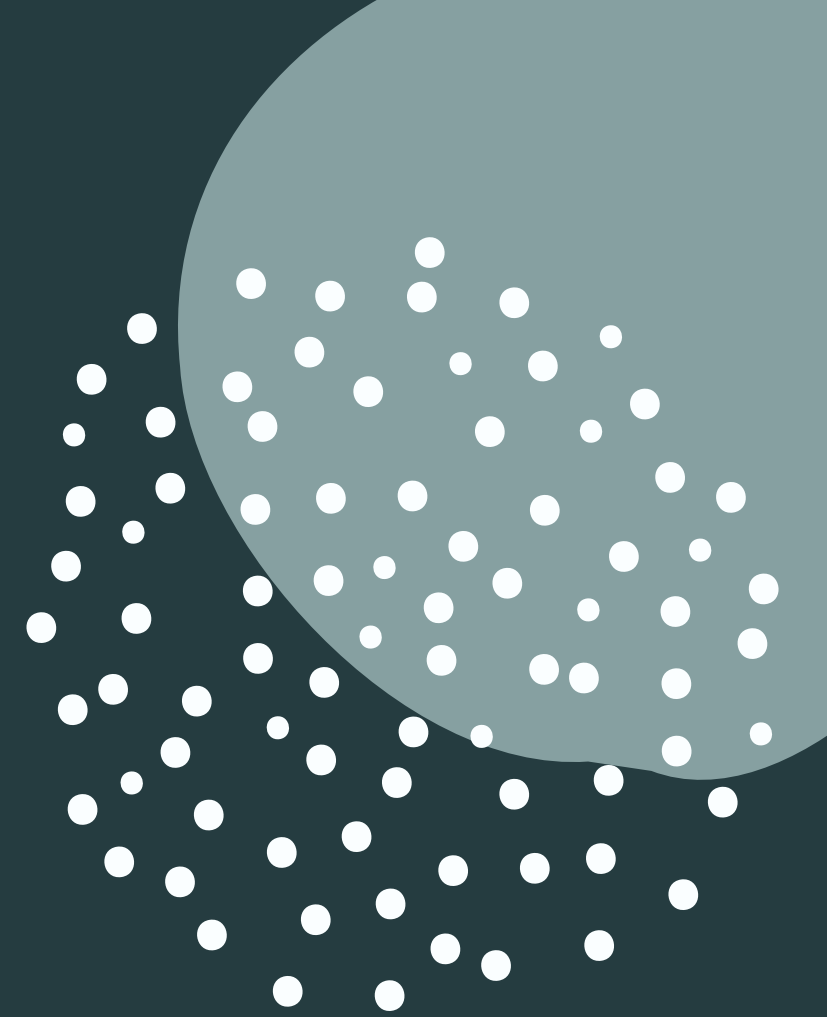
Error 1.71

Error 3.31

Error 3E-5

PROBLEMA 2

Dada la capacidad generativa del autoencoder,
elegir (o construir) un conjunto de datos
e intentar utilizar el Autoencoder para generar una
nueva muestra que ustedes puedan
juzgar que pertenece al conjunto de datos que le
presentaron al autoencoder



VAE

- Utilizamos la librería keras
- Usamos una capa intermedia tanto para el encoder como para el decoder
- Dimensión de capa intermedia de 256
- Función de activación ReLU
- Se le define la función de pérdida a partir de la cross entropy entre los labels (valores de los pixels) y la salida del decoder

Model: "encoder"

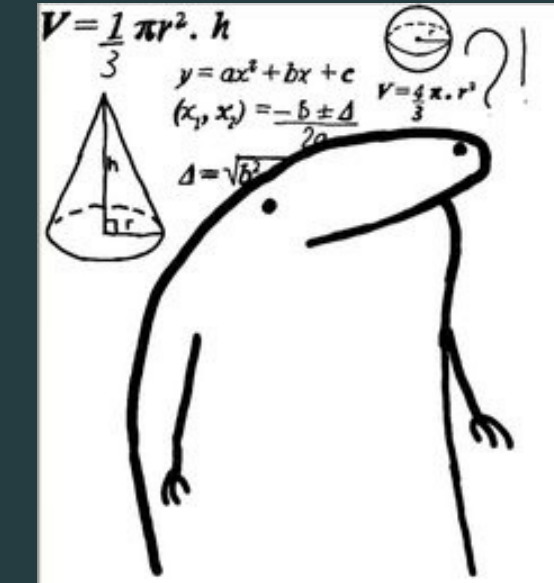
Layer (type)	Output Shape	Param #	Connected to
=====			
input (InputLayer)	[(None, 6400)]	0	[]
encoding (Dense)	(None, 256)	1638656	['input[0][0]']
mean (Dense)	(None, 2)	514	['encoding[0][0]']
log-variance (Dense)	(None, 2)	514	['encoding[0][0]']
lambda_17 (Lambda)	(None, 2)	0	['mean[0][0]', 'log-variance[0][0]']

Model: "decoder"

Layer (type)	Output Shape	Param #
=====		
decoder_input (InputLayer)	[(None, 2)]	0
decoder_h (Dense)	(None, 256)	768
flat_decoded (Dense)	(None, 6400)	1644800

DATASET

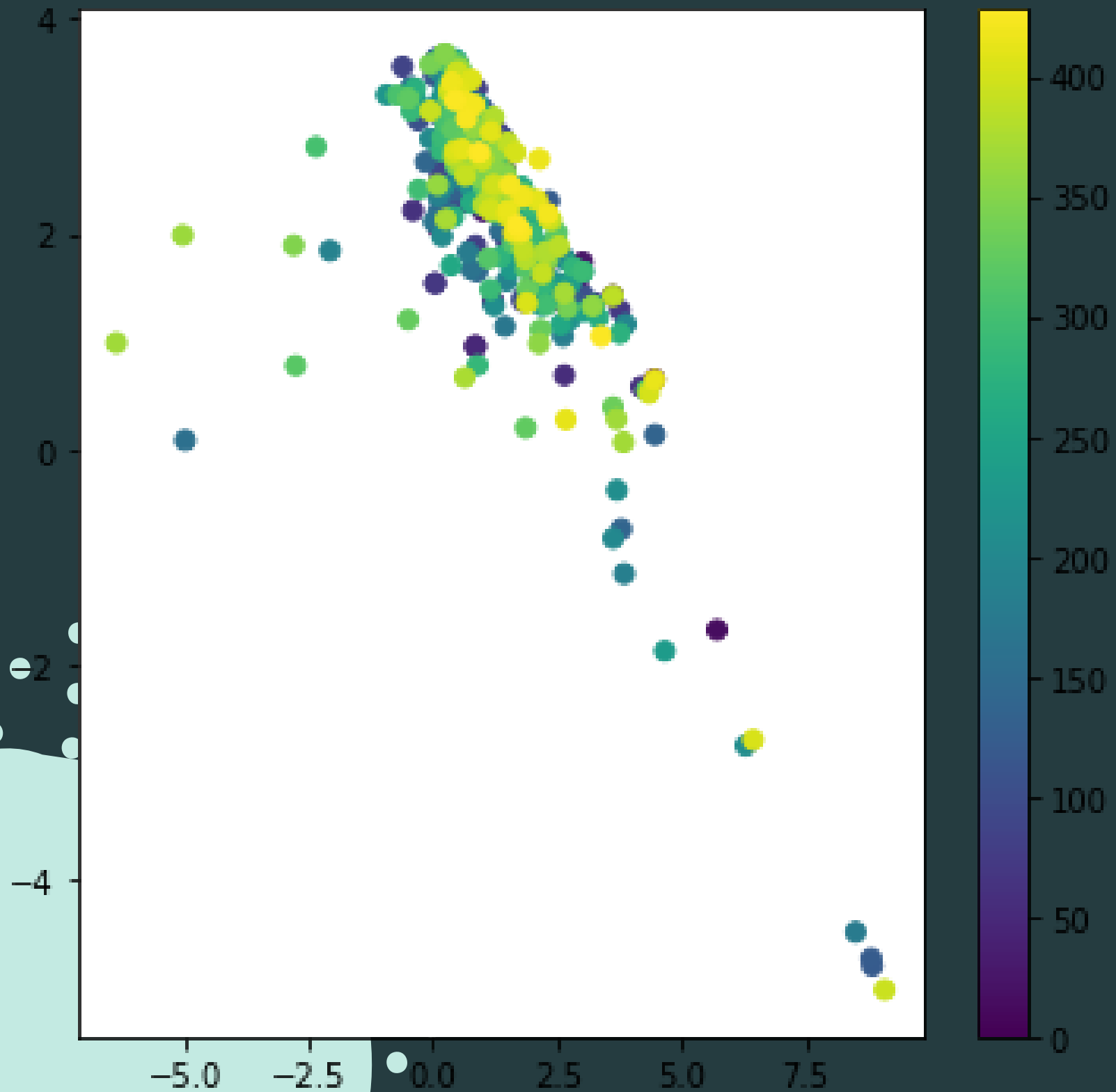
- Creamos un dataset a partir de imagenes de Flork
- Un total de 426 imágenes
- Las ajustamos a un tamaño de 80x80 pixeles



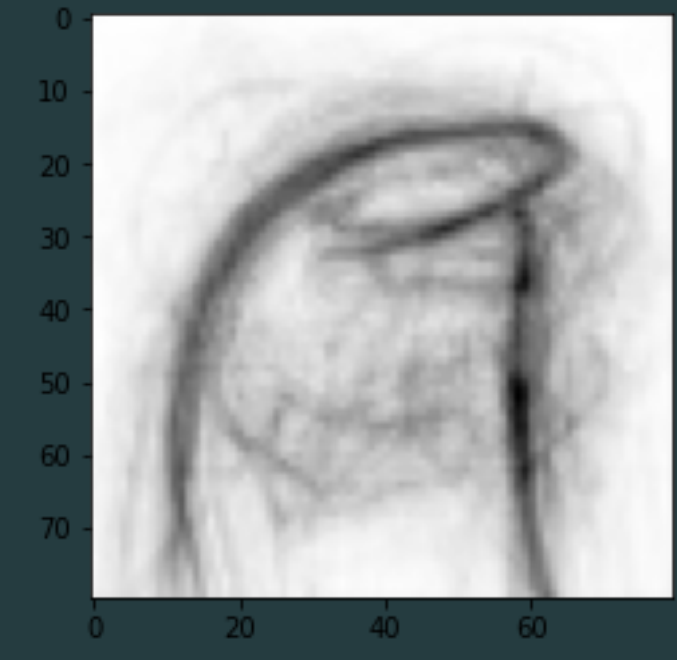
MAPA 2D DEL ESPACIO LATENTE

Épocas: 50

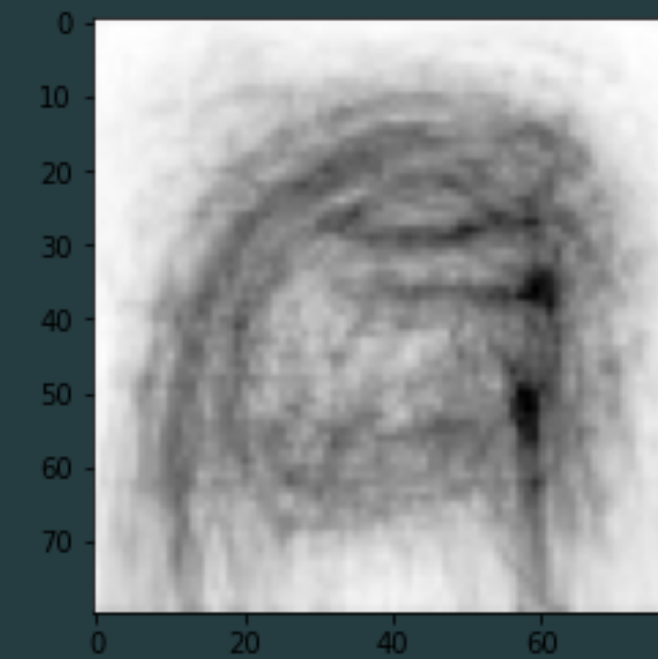
Capas Intermedias: 1 de dimensión 256



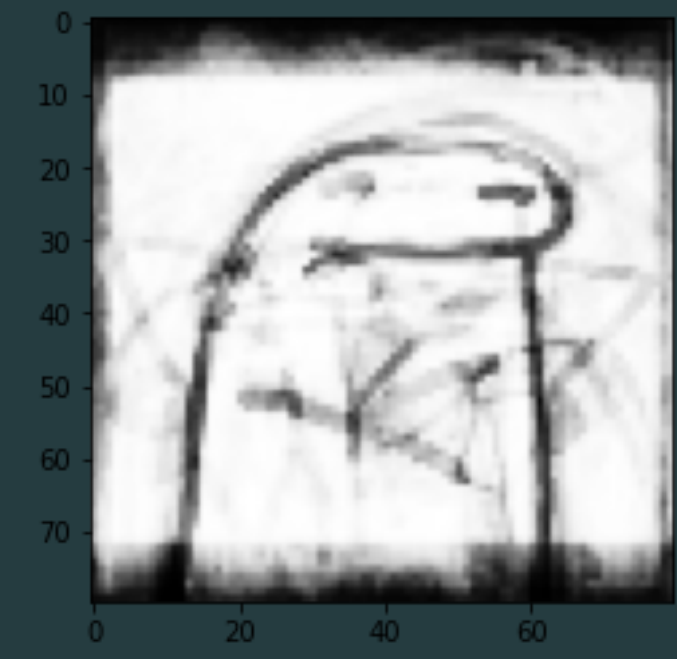
[2.5,-2]



[2.5,1]



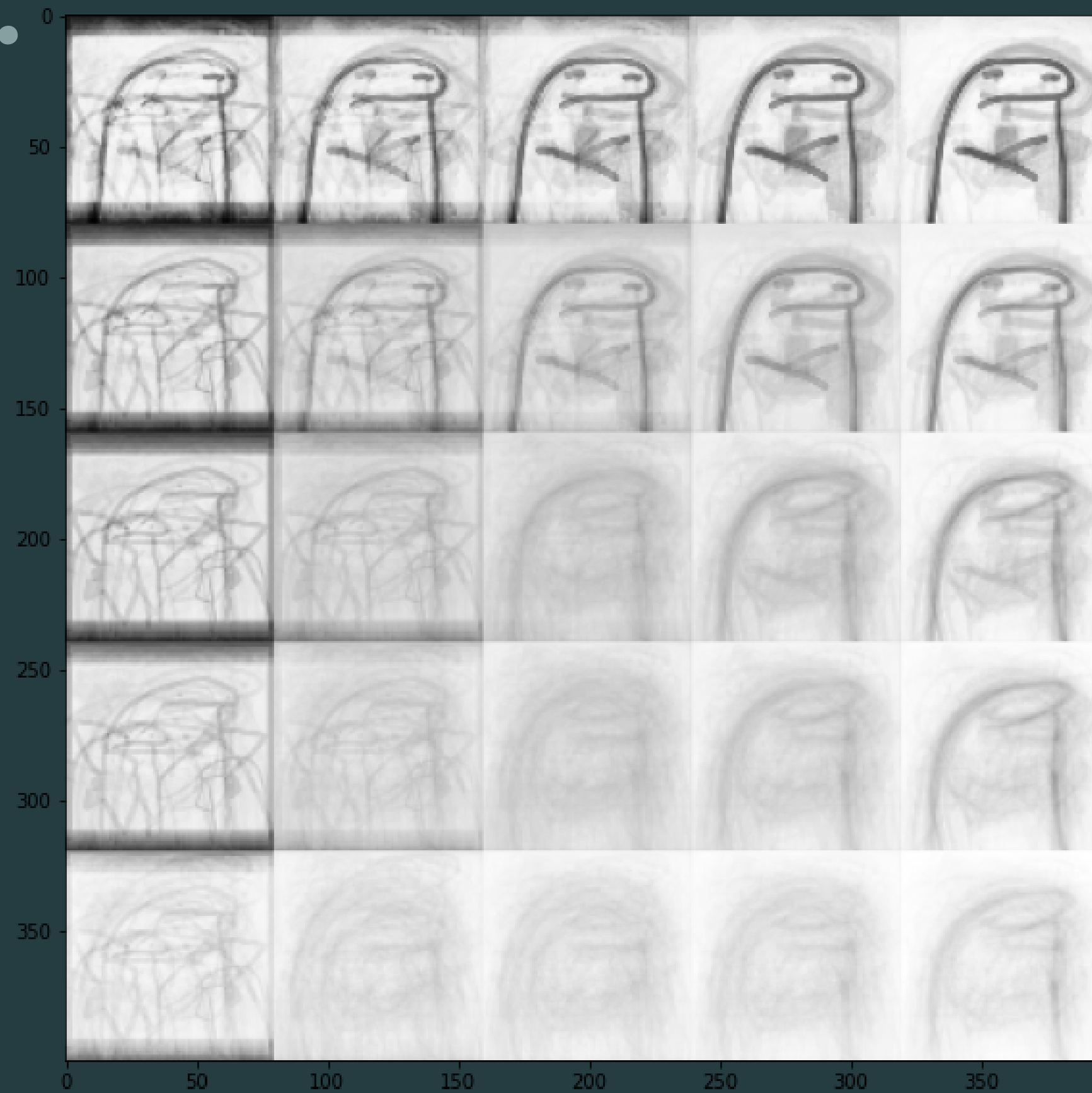
[1,2]



[-2.5,-4]

Épocas: 50

Capas Intermedias: 1 de dimensión 256

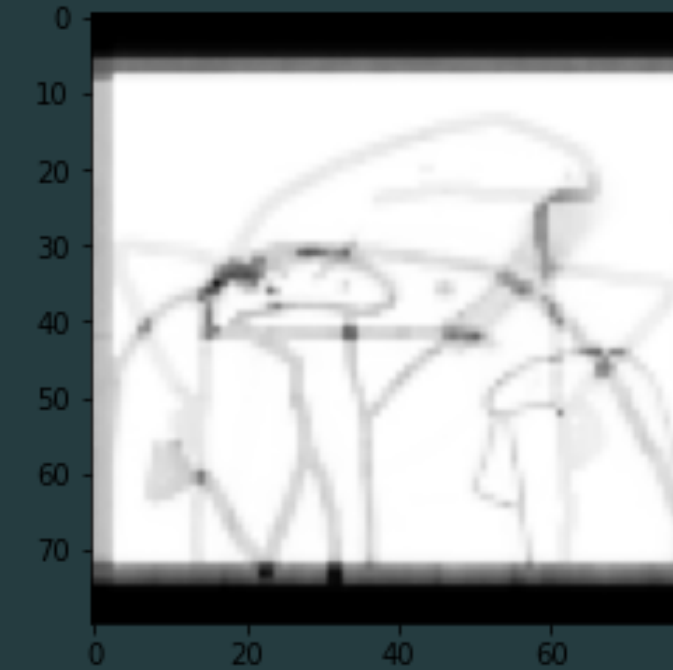
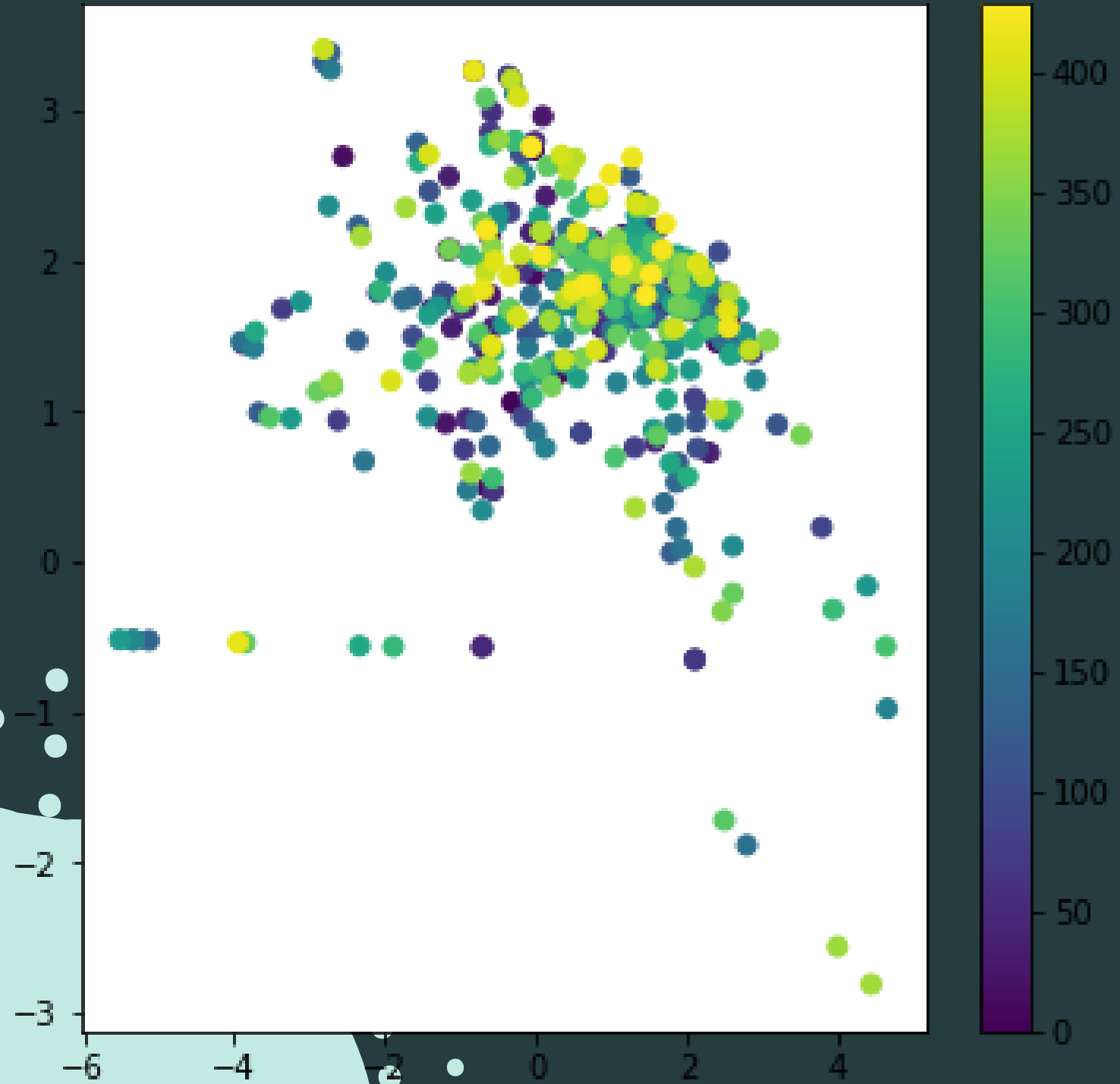


Valores en X = [-1.6448
-5.9776e-01 -1.3914e-16
5.9776e-01 1.6448]

Valores en Y = [-1.6448
-5.9776e-01 -1.3914e-16
5.9776e-01 1.6448]

Épocas: 200

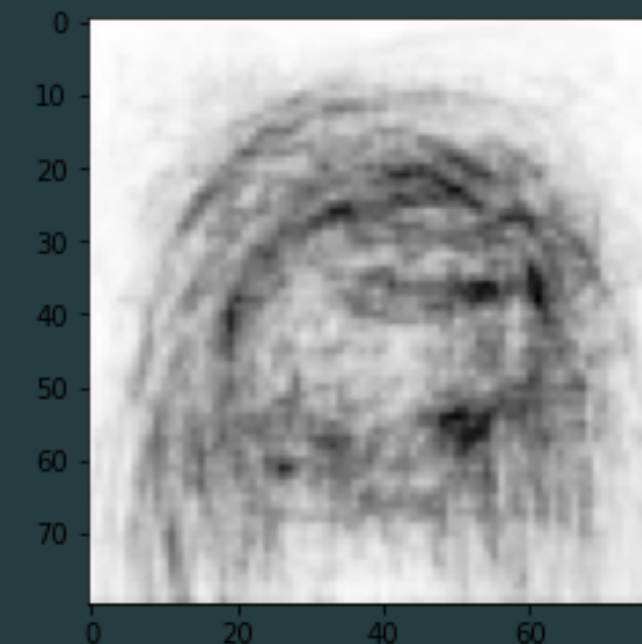
Capas Intermedias: 1 de dimensión 640 y
otra de 256



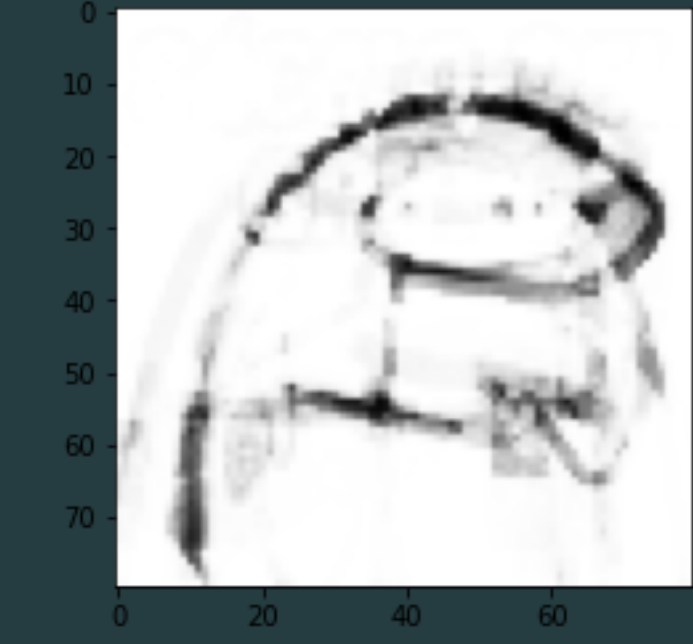
[4,-2]



[-4,-1]



[1,2]



[-4,-2.5]

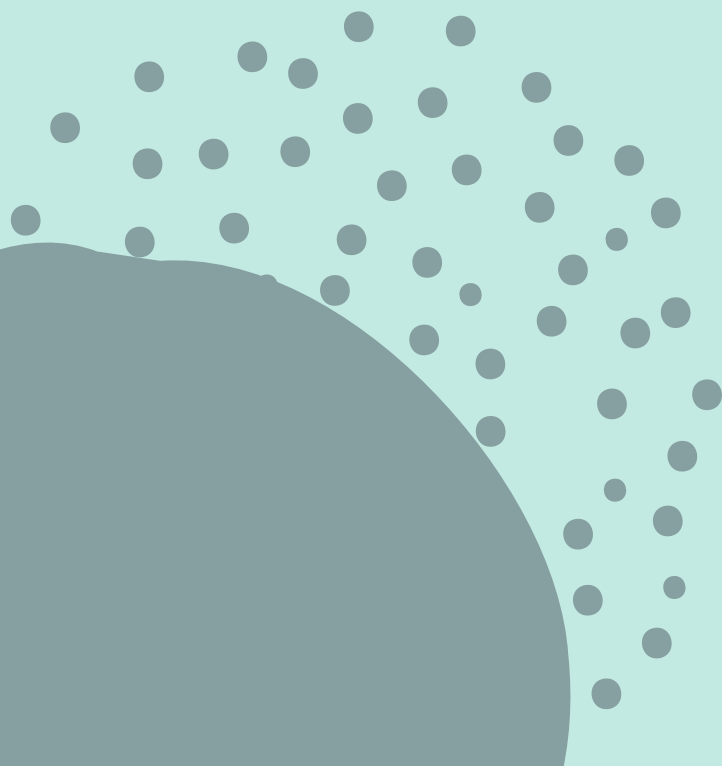
Capas Intermedias: 1 de dimensión 640 y otra de 256

Valores en Y = [-1.6448
-5.9776e-01 -1.3914e-16
5.9776e-01 1.6448]



CONCLUSIONES VAE

- Las imágenes generadas a partir de puntos en zonas de alta densidad tienden a ser borrosas
- Las imágenes generadas en zonas de menor densidad tienden a ser similares a la entrada mas cercana
- A medida que se toman puntos mas alejados de todos los puntos de entrada, la imagen va perdiendo detalle



¡MUCHAS GRACIAS!