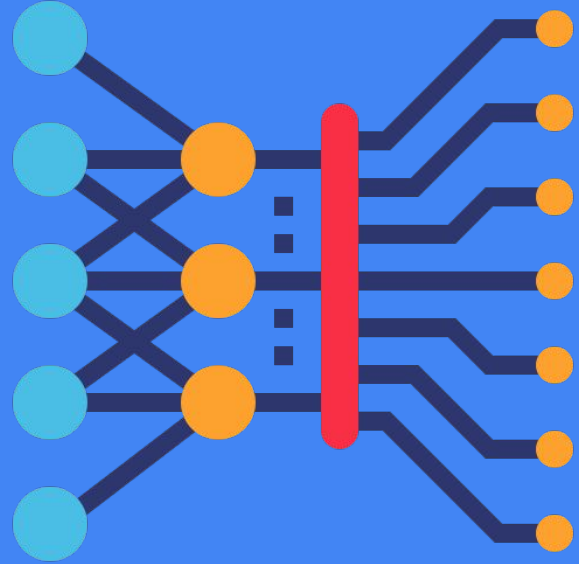


Deep Learning

Sistemas de Inteligencia Artificial
ITBA 2022 - 1C

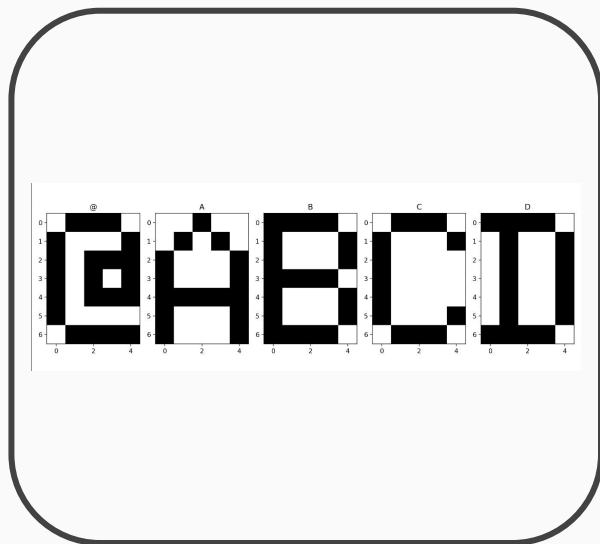
Integrantes:

- Serpe, Octavio (60076)
- Quesada, Francisco (60524)
- Arca, Gonzalo (60303)

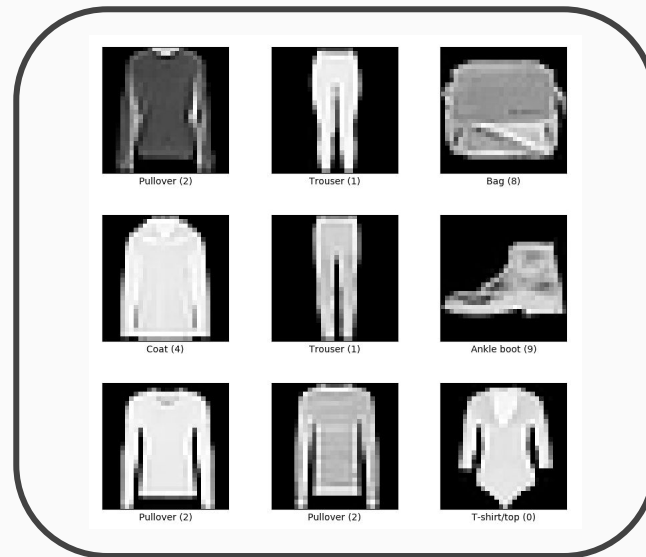


Introducción

Dado dos conjuntos de datos:



font.h



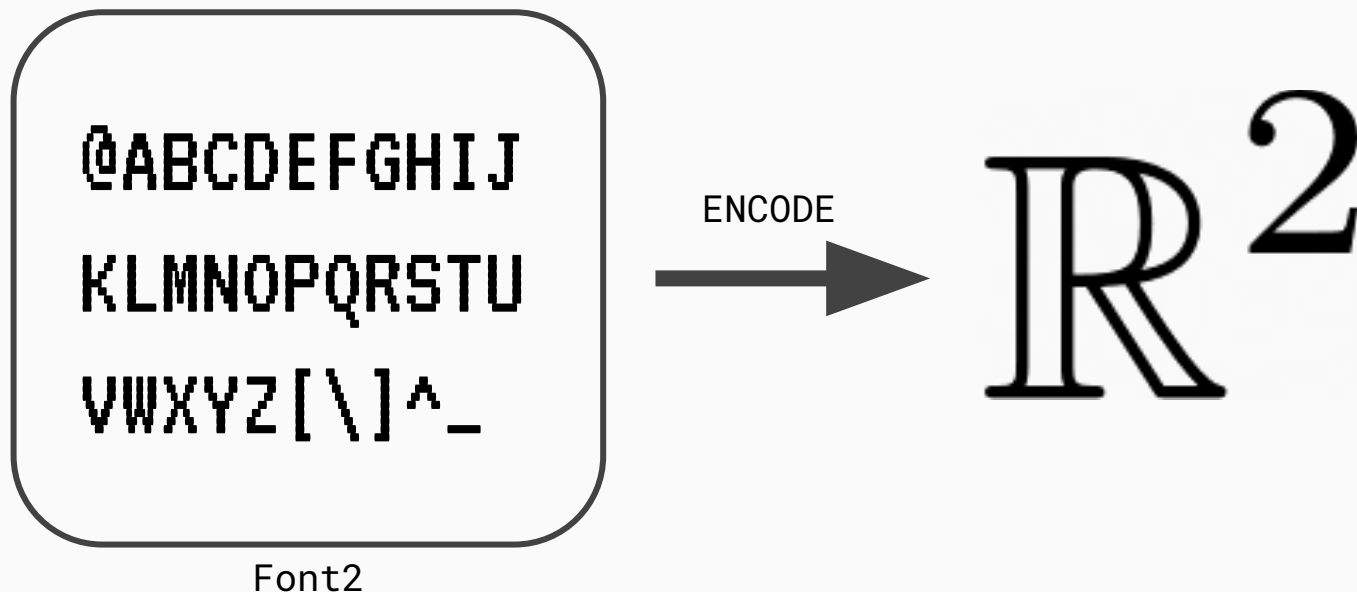
Fashion-MNIST

Aplicar el uso de **Autoencoders** para lograr **compresión**, **denoising** y **generación** de datos

Ejercicios

Problema 1.a: Autoencoder básico

Estudiar la representación en 2 dimensiones de las imágenes de 7x5 del archivo font.h haciendo uso de un Autoencoder básico



Loss function

Error cuadrático medio (MSE)

$$\text{MSE}(\mathbf{y}, f(\mathbf{x}, \mathbf{w})) = \frac{1}{N} \sum_{\mu} \|\mathbf{y}^{\mu} - f(\mathbf{x}^{\mu}, \mathbf{w})\|^2$$

Regularización

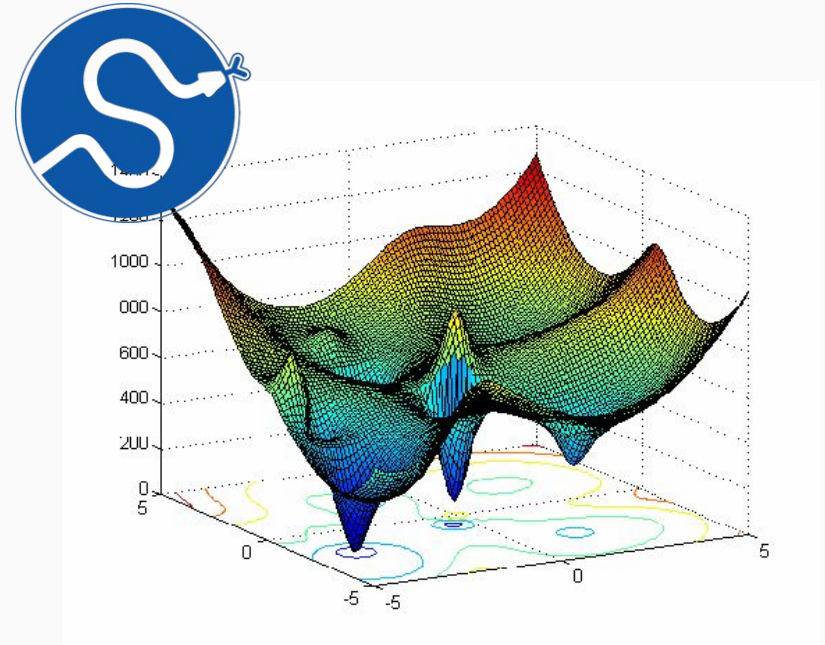
Regularización L2

$$L_{\text{reg}L2}(\mathbf{y}, \mathbf{w}, f(\mathbf{x}, \mathbf{w})) = \text{MSE}(\mathbf{y}, f(\mathbf{x}, \mathbf{w})) + \lambda \|\mathbf{w}\|_2^2$$

Técnicas de optimización utilizadas

Librería: SciPy

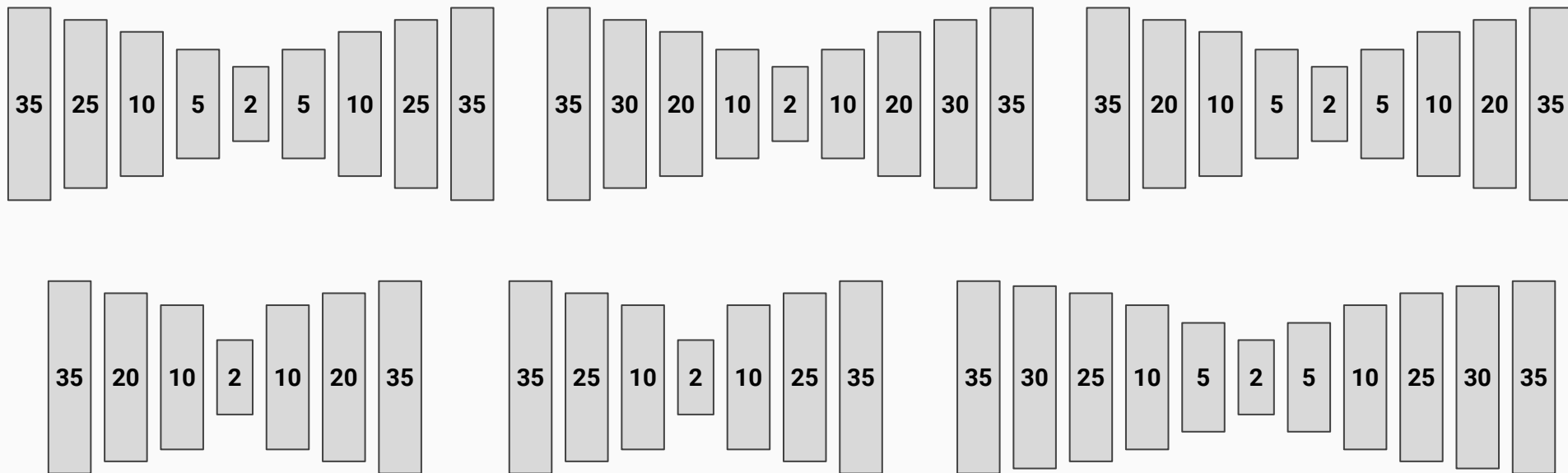
- Método de Powell
- Broyden–Fletcher–Goldfarb–Shanno (BFGS)
- Gradientes conjugados (CG)



Ejercicio 1.a

Autoencoder básico

Arquitecturas candidatas

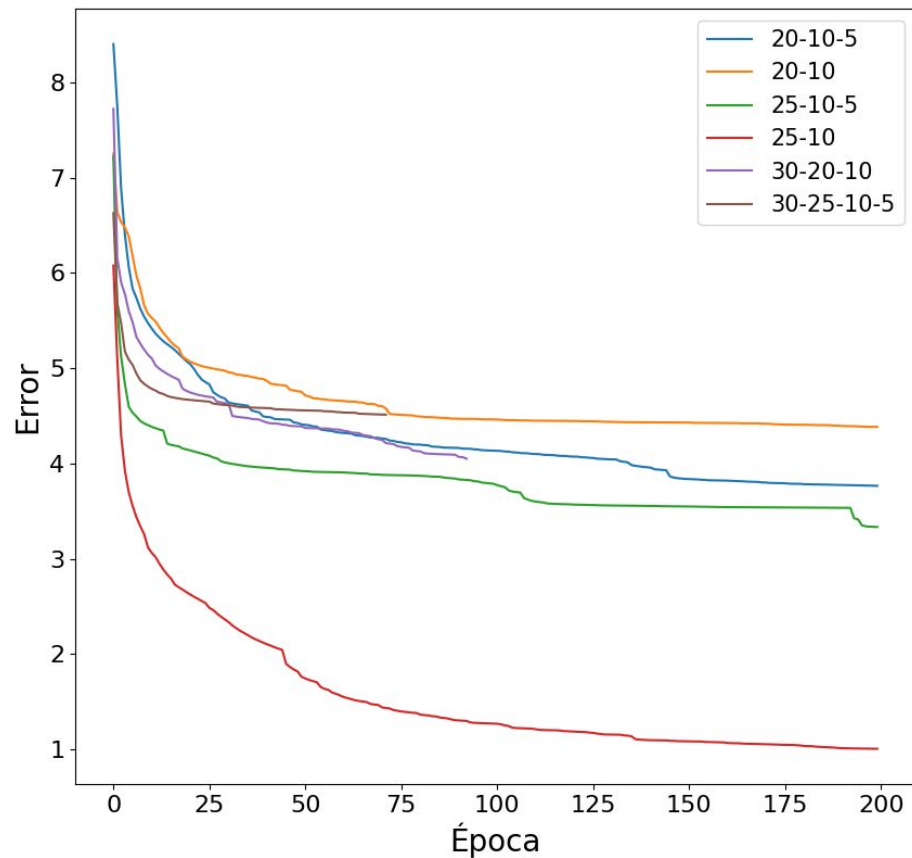


Toda capa usa **ReLU** salvo la del espacio latente, que usa la **identidad**,
y la última que utiliza **logística**

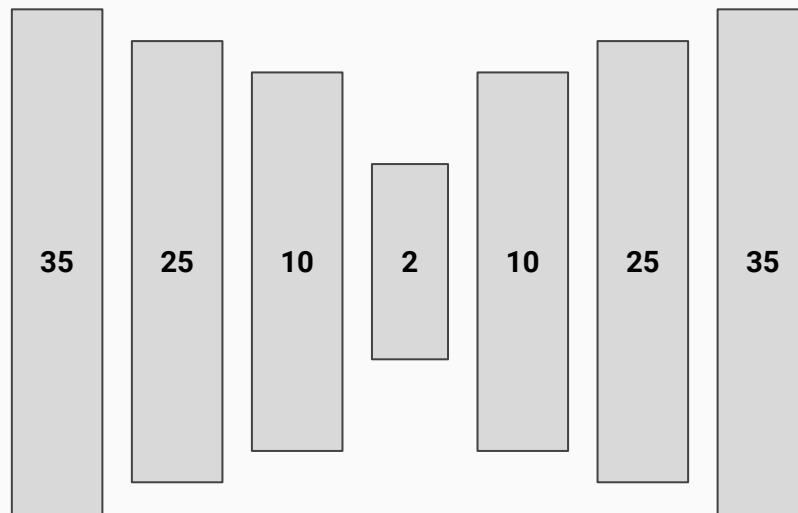
Error vs. Época

Variando arquitecturas con
dim(espacio latente) = 2

- épocas = 200
- $\lambda = 10^{-4}$
- term reg = L2
- Optimizador: Powell



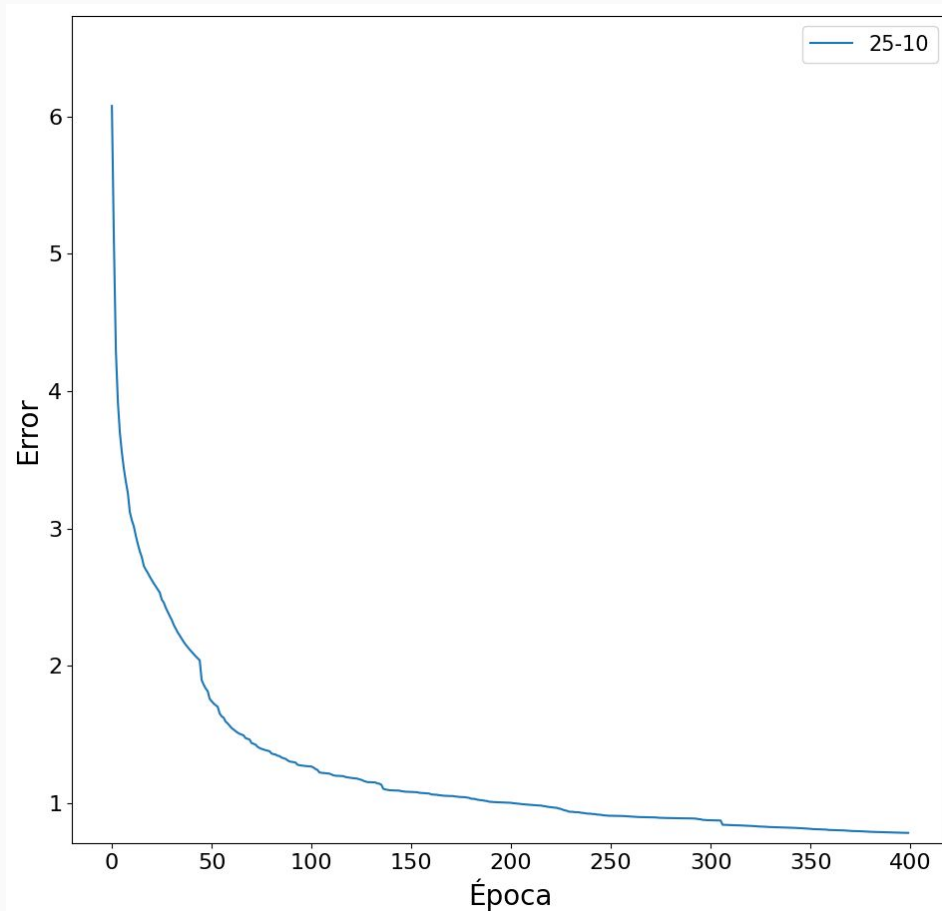
Arquitectura óptima



Resultados

Mejor arquitectura

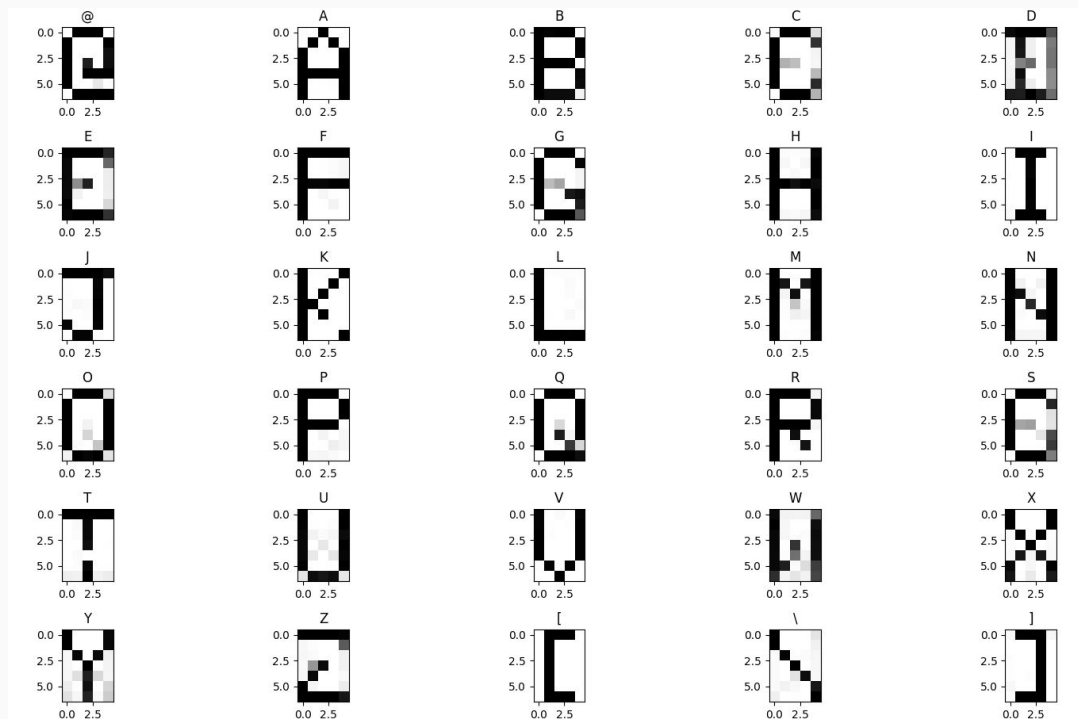
- épocas = 400
- $\lambda = 10^{-4}$
- term reg = L2
- Error final = 0.7851
- Optimizador: Powell



Resultados

Mejor arquitectura

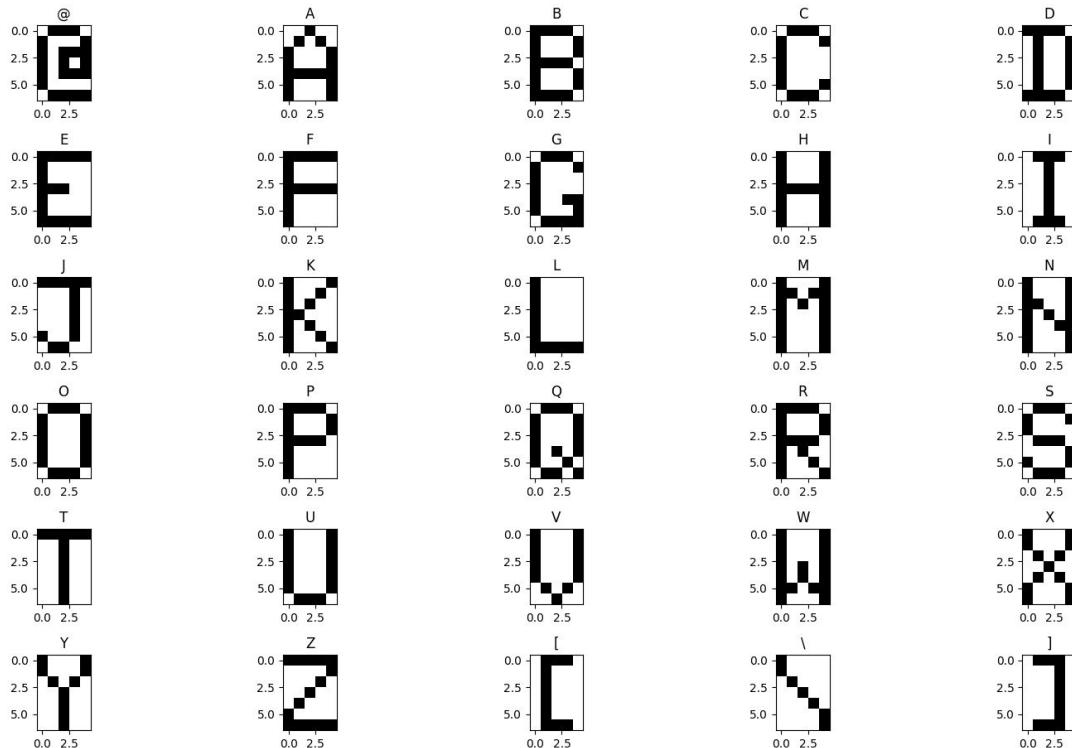
- épocas = 400
- $\lambda = 10^{-4}$
- term reg = L2
- Error final = 0.7851



Resultados

Mejor arquitectura

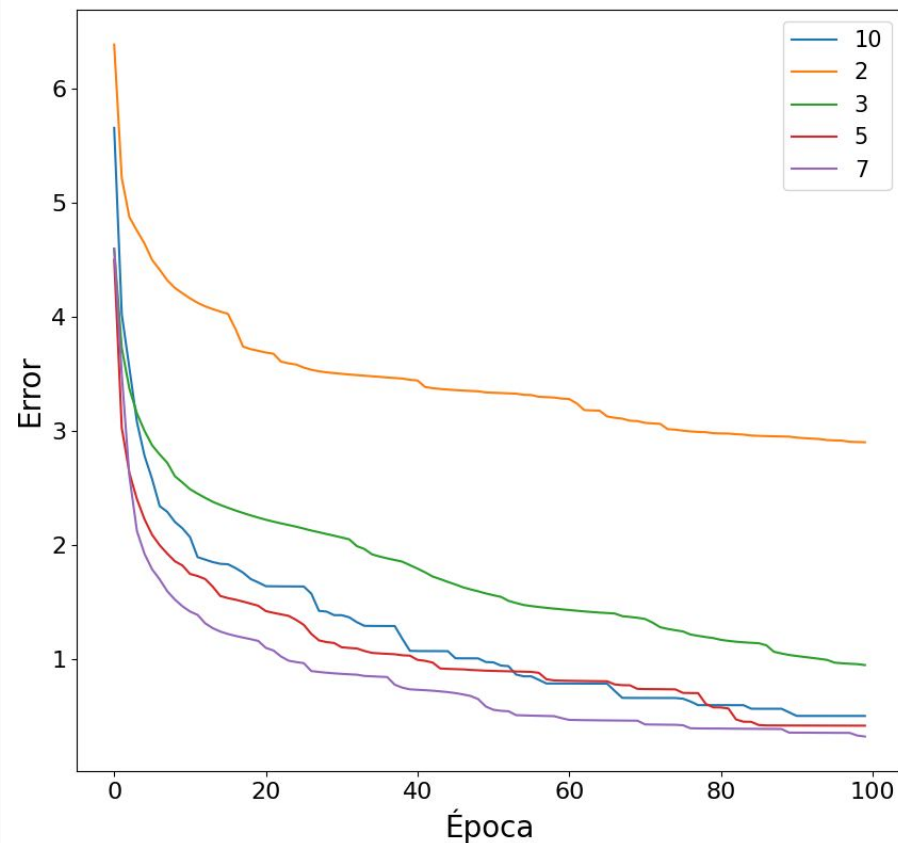
- épocas = 1000
- $\lambda = 10^{-4}$
- term reg = L2



Error vs. época

Variando **dim**(espacio latente)

- épocas = 100
- $\lambda = 10^{-4}$
- term reg = L2
- Arquitectura = 25 - 10 - x
- Optimizador: Powell



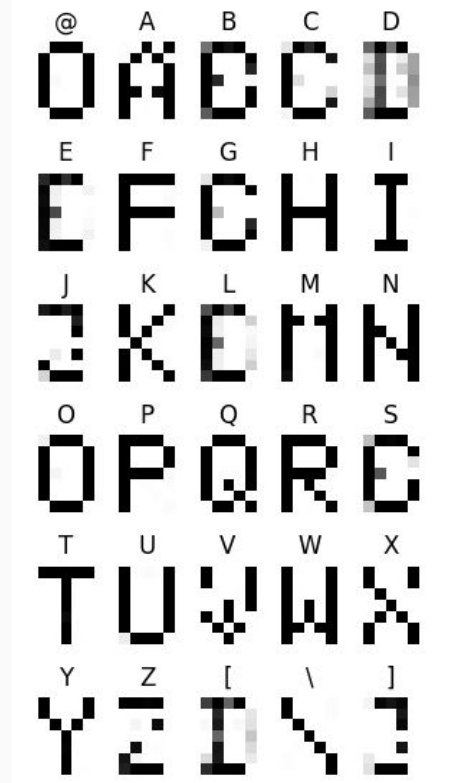
Decodificación

Resultados

$\dim(\text{latente}) = 3$,
Error ≈ 0.94



$\dim(\text{latente}) = 2$
Error ≈ 2.92



Decodificación

Resultados

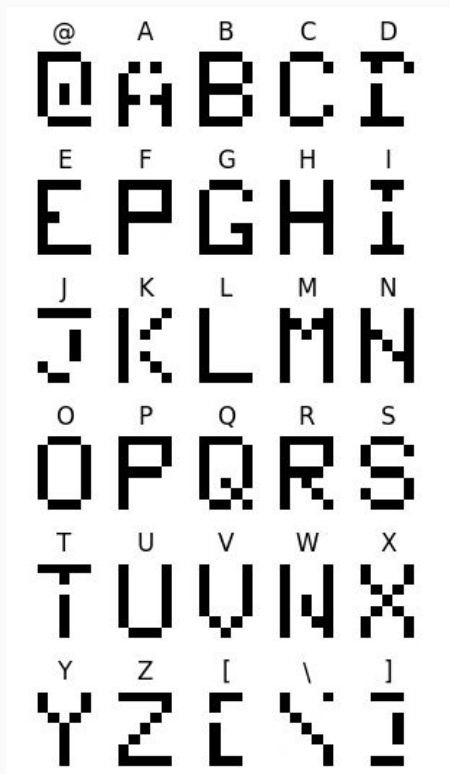
dim(latente) = 10

Error ≈ 0.503



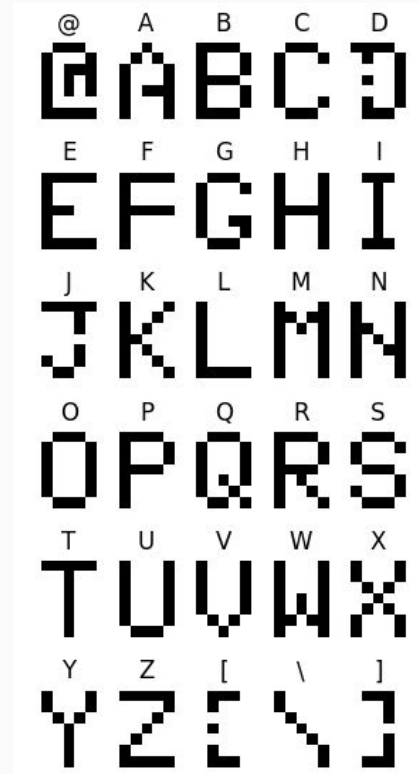
dim(latente) = 7

Error ≈ 0.32



dim(latente) = 5

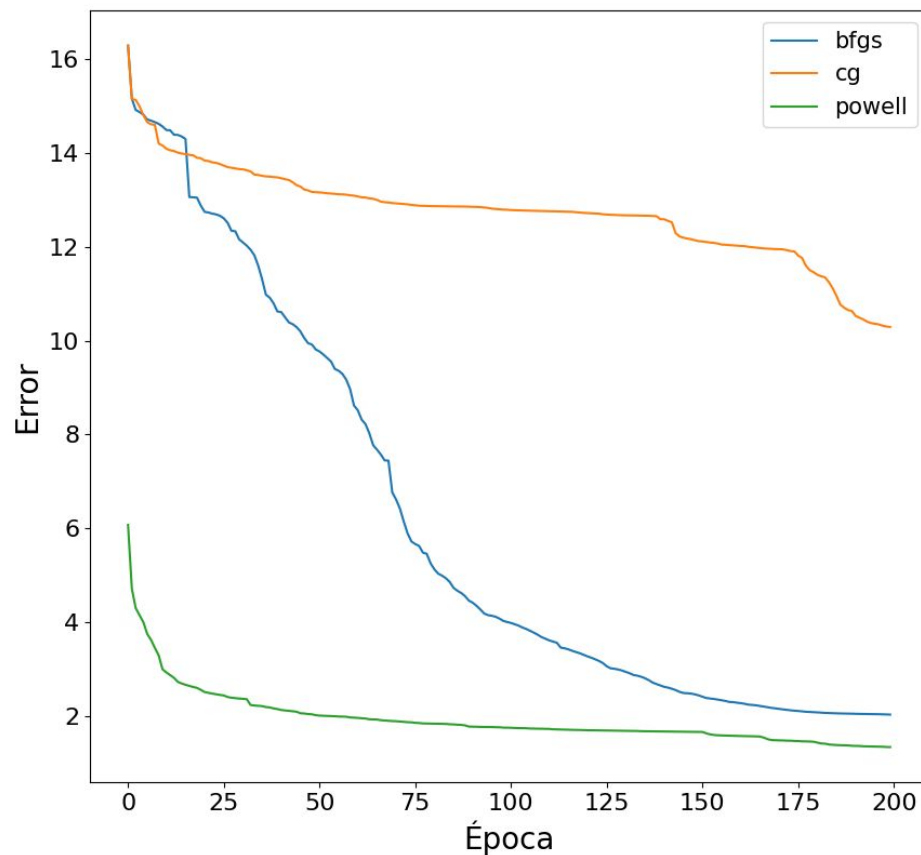
Error ≈ 0.42



Error vs. época

Variando método de *optimización*

- épocas = 200
- $\lambda = 10^{-4}$
- term reg = L2
- Arquitectura = 25 - 10 - 2



Error y tiempo

Variando método de *optimización*

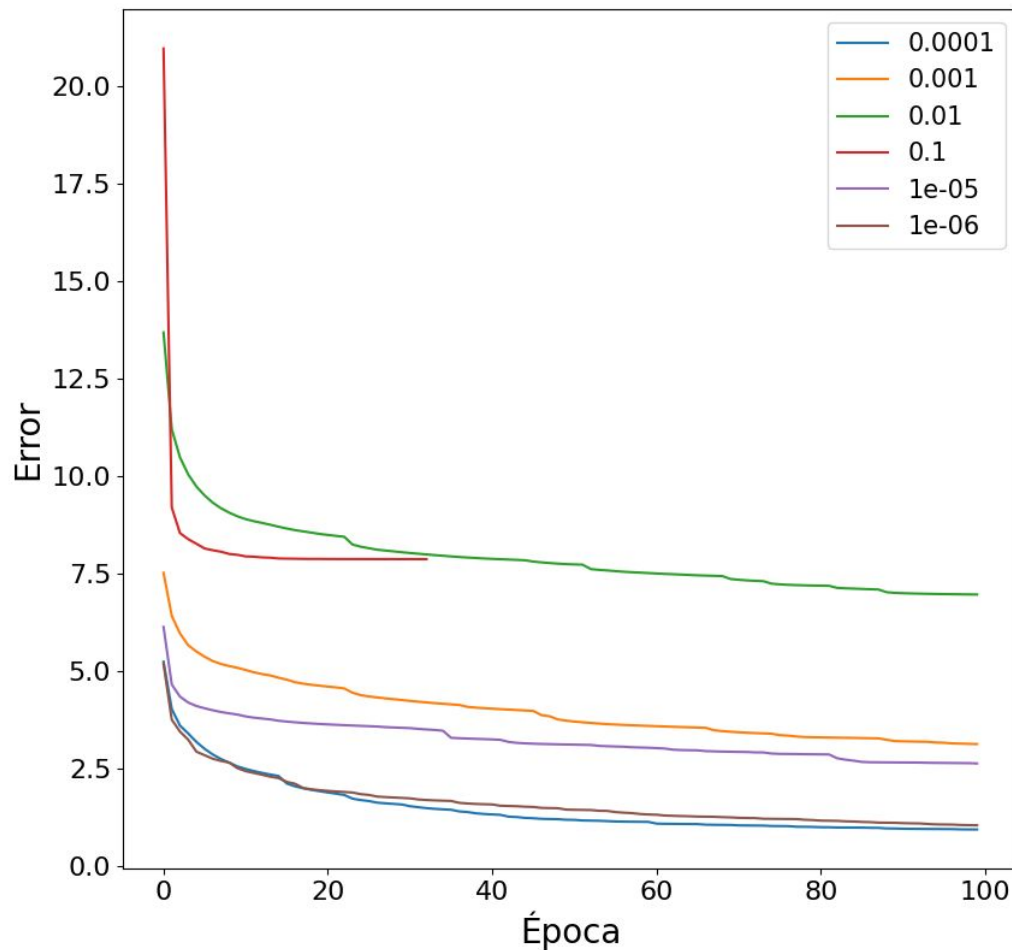
- épocas = 200
- $\lambda = 10^{-4}$
- term reg = L2
- Arquitectura = 25 - 10 - 2

Método	Tiempo de ejecución (s)	Error
BFGS	121.91	2.03
Gradientes conjugados	83.61	10.3
Powell	1163.34	1.34

Error vs. época

Variando término **regularizador**

- épocas = 100
- term reg = L2
- Arquitectura = 25 - 10 - 2
-



Espacio latente

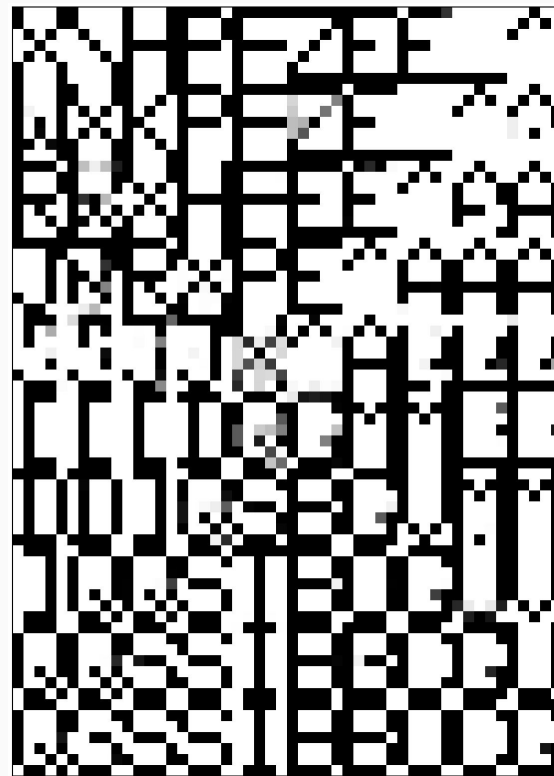
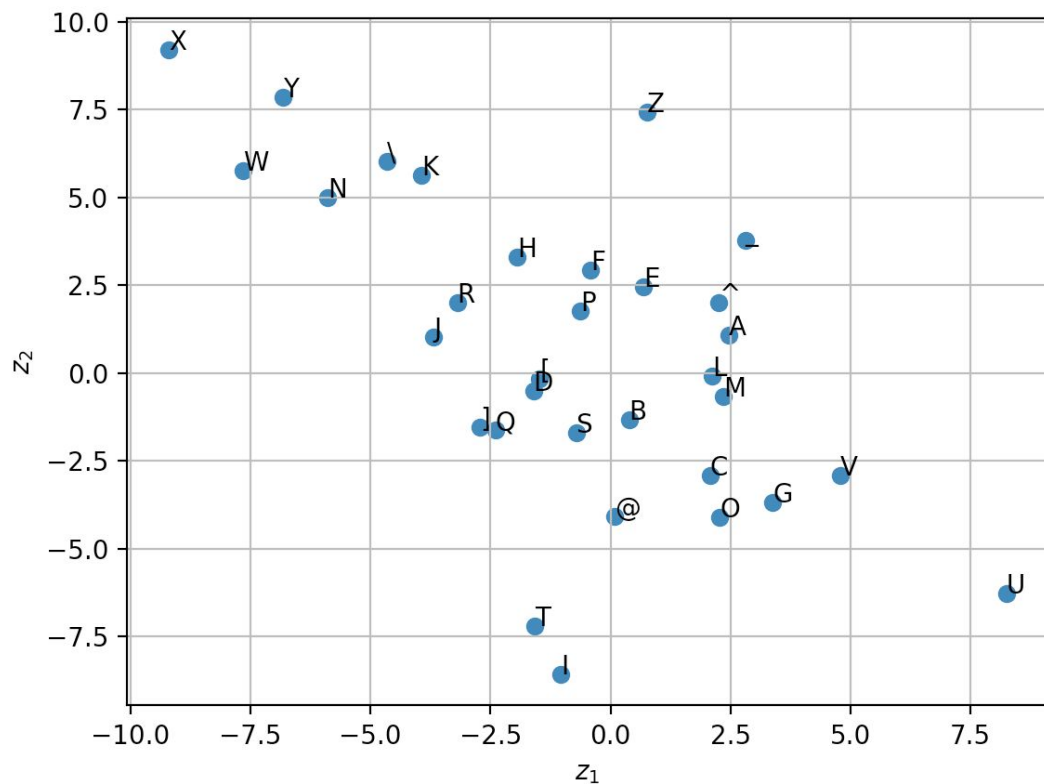
Creación de nuevas letras

- Épocas = 1000

- $\lambda = 10^{-4}$

- Term reg = L2

- Arquitectura = 25 - 10 - 2



Espacio latente

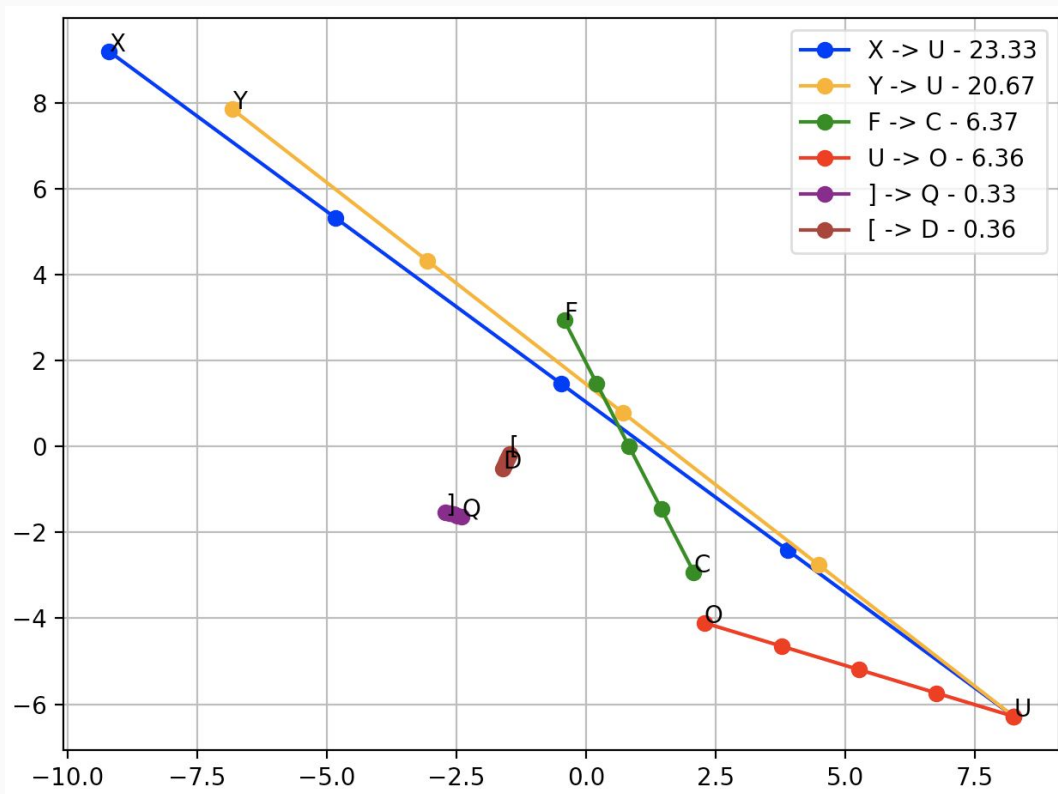
Creación de nuevas letras

- Épocas = 1000

- $\lambda = 10^{-4}$

- Term reg = L2

- Arquitectura = 25 - 10 - 2



X -> U X -> U X -> U X -> U X -> U

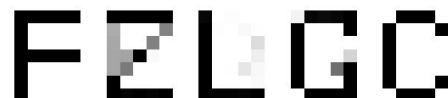


Más
lejanos

Y -> U Y -> U Y -> U Y -> U Y -> U



F -> C F -> C F -> C F -> C F -> C



Inter-
medios

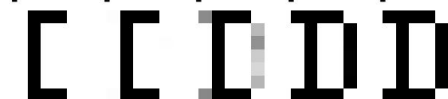
U -> O U -> O U -> O U -> O U -> O



] -> Q] -> Q] -> Q] -> Q] -> Q



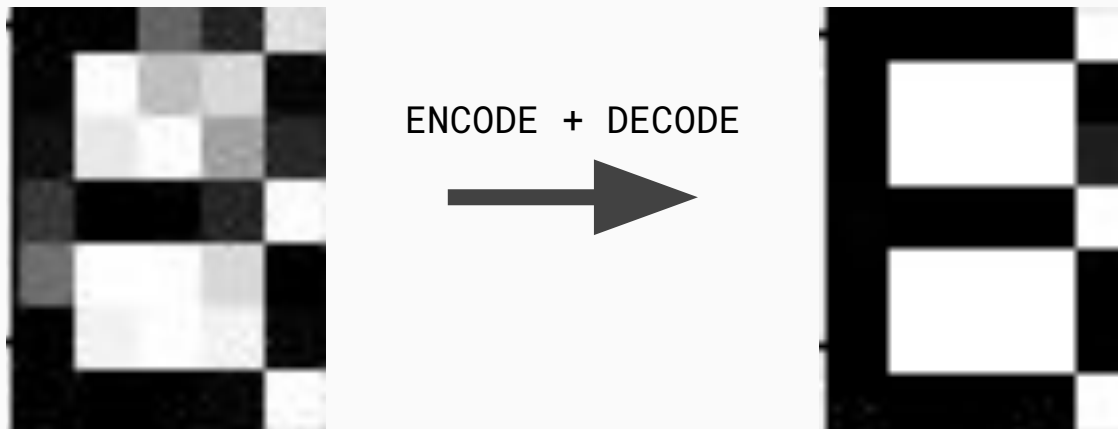
[-> D [-> D [-> D [-> D [-> D



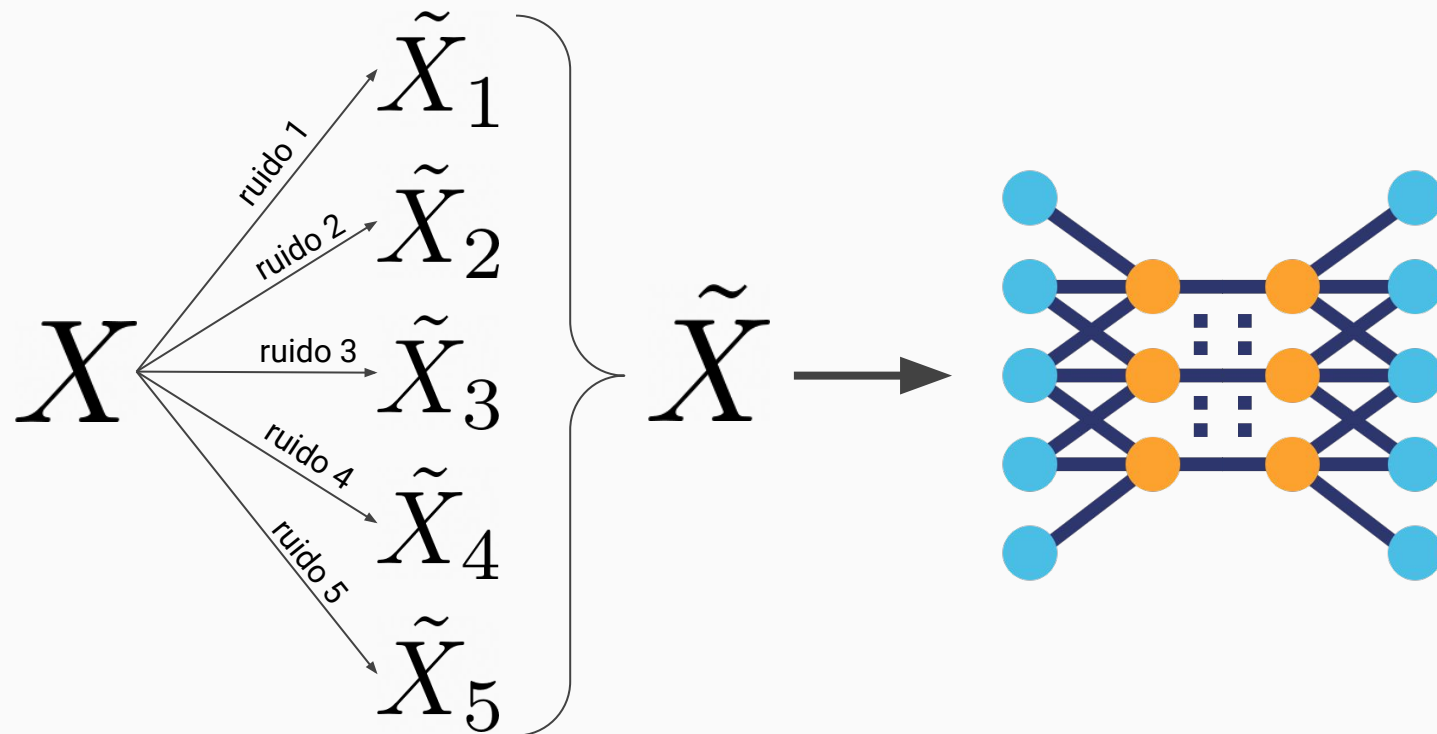
Más
cercanos

Problema 1.b: Denoising Autoencoder

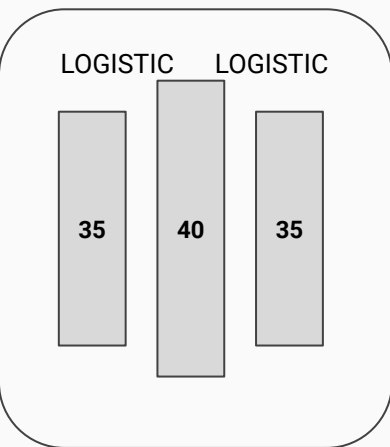
Distorsionar las **entradas** en diferentes niveles y **estudiar la capacidad** del Autoencoder de **eliminar el ruido**.



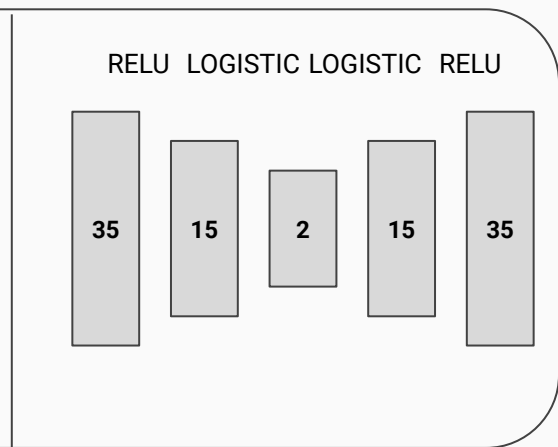
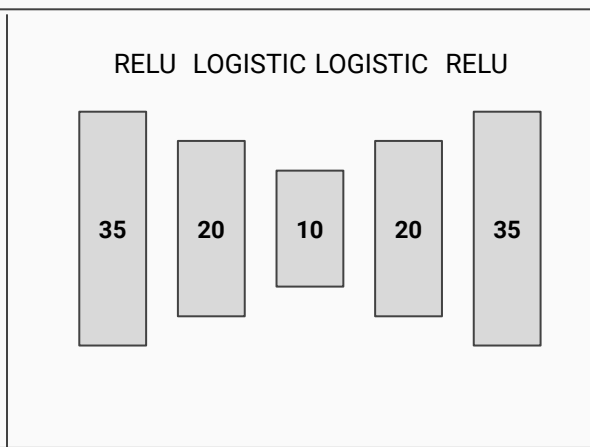
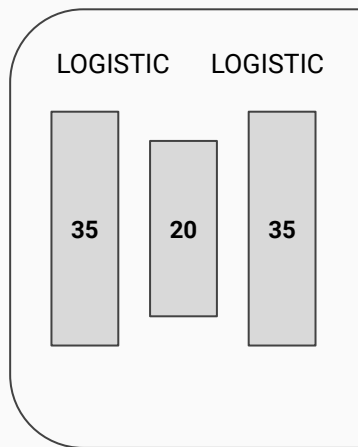
Método de entrenamiento



Arquitecturas candidatas



Overcomplete

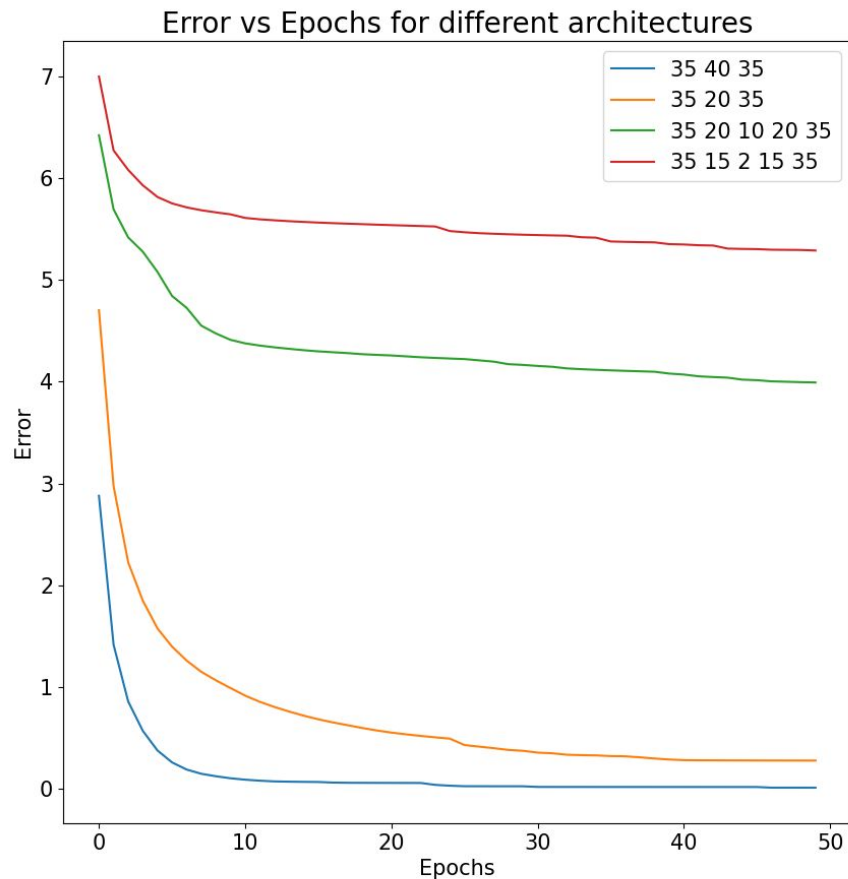


Undercomplete

Error vs. época

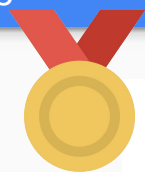
Variando arquitecturas

- $P(\text{ruido}) = 0.5$
- ruido gaussiano
- épocas = 50
- optimizador: Powell
- $\lambda = 10^{-6}$



Resultados - Arquitecturas

Denoising Autoencoder



35 40 35

QQQ
AAA
BBB
CCO
DDD
EEE
FFF
GGG
HHH
III

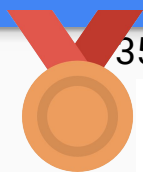
Error ≈ 0.008



35 20 35

QQQ
AAA
BBB
CCO
DDD
EEE
FFF
GGG
HHH
III

Error ≈ 0.3



35 20 10 20 35

QQQ
AAA
BBB
CCO
DDD
EEE
FFF
GGG
HHH
III

Error ≈ 4

35 10 2 10 35

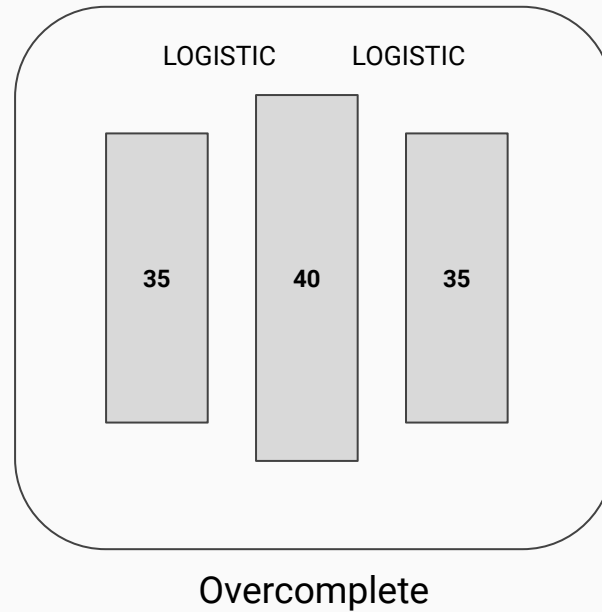
QQQ
AAA
BBB
CCO
DDD
EEE
FFF
GGG
HHH
III

Error ≈ 5.3

Ejercicio 1.b

Denoising Autoencoder

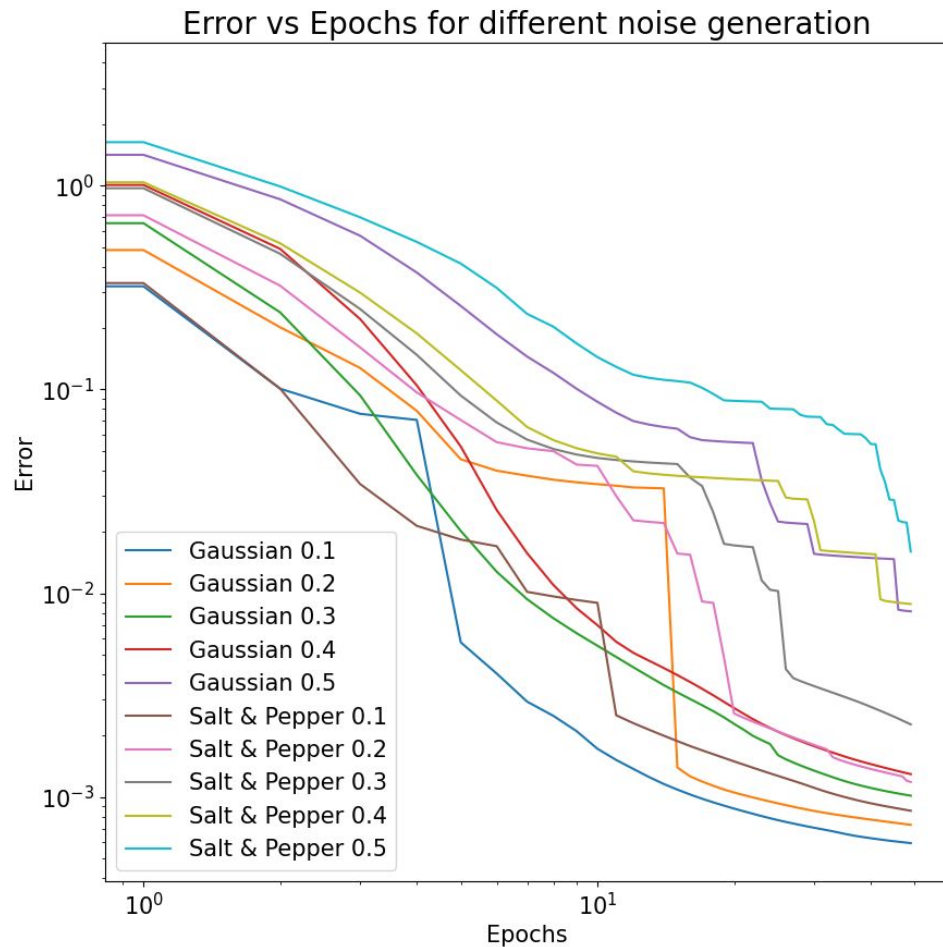
Arquitectura óptima



Error vs. época

Variando *tipo* y *cantidad de ruido*

- Arquitectura 35-40-35
- épocas = 50
- optimizador: Powell
- $\lambda = 10^{-6}$



Denoising: Ruido gaussiano

Resultados

0.1



Error ≈ 0.0005

0.2



Error ≈ 0.0007

0.3



Error ≈ 0.0010

0.4



Error ≈ 0.013

0.5



Error ≈ 0.0082

Denoising: Ruido salt & pepper

Resultados

0.1



Q Q Q
A A A
B B B
C C C
D D D
E E E
F F F
G G G
H H H
I I I

Error ≈ 0.0008

0.2



Q Q Q
A A A
B B B
C C C
D D D
E E E
F F F
G G G
H H H
I I I

Error ≈ 0.001

0.3



Q Q Q
A A A
B B B
C C C
D D D
E E E
F F F
G G G
H H H
I I I

Error ≈ 0.002

0.4



Q Q Q
A A A
B B B
C C C
D D D
E E E
F F F
G G G
H H H
I I I

Error ≈ 0.008

0.5

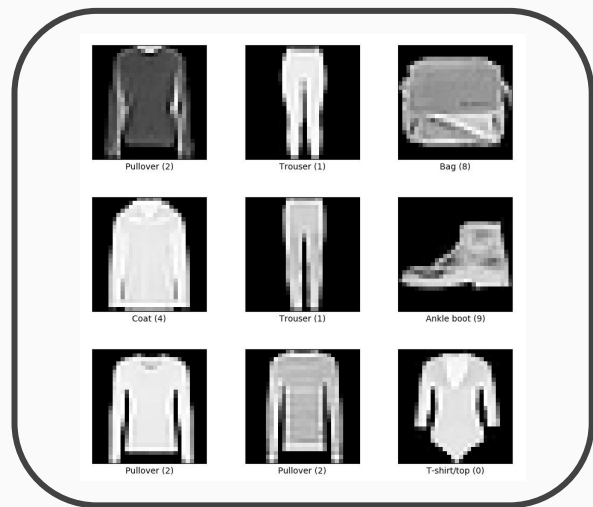


Q Q Q
A A A
B B B
C C C
D D D
E E E
F F F
G G G
H H H
I I I

Error ≈ 0.02

Problema 2: Generación de muestras mediante Autoencoders

Utilizar el **Autoencoder** para **generar** una **nueva muestra** que **aparenta pertenecer** al **conjunto de datos** presentados al Autoencoder



Fashion-MNIST

AUTOENCODER



Librerías utilizadas

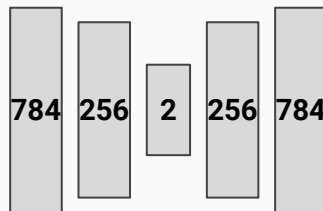
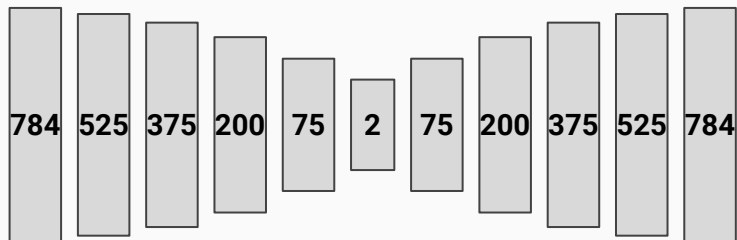
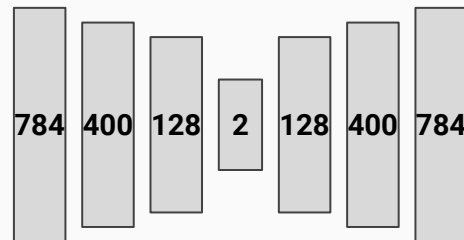
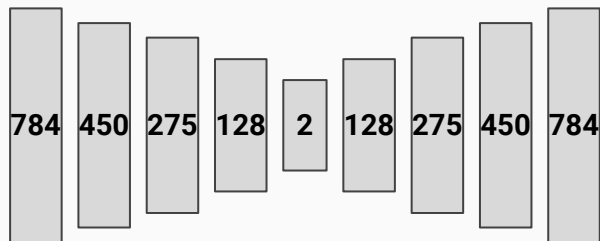


+



Ejercicio 2: Autoencoders generativos

Posibles arquitecturas:



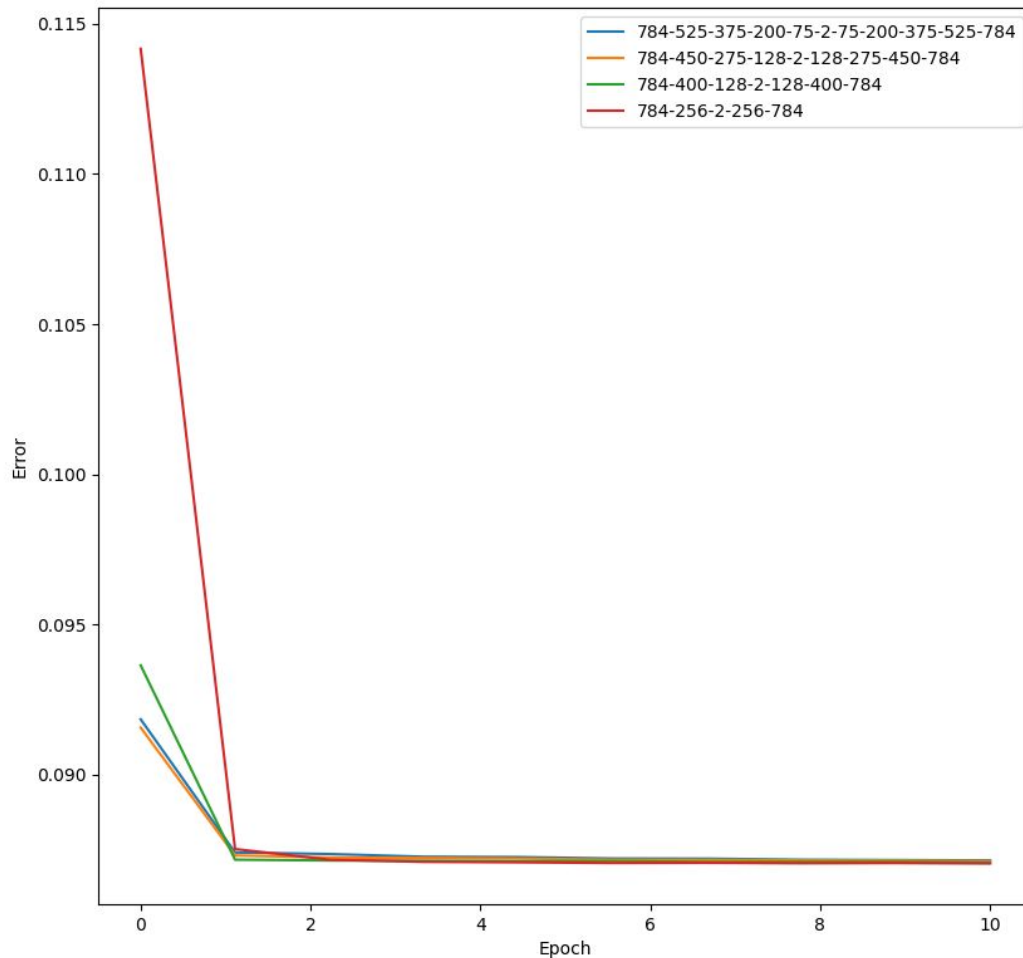
Toda capa usa **ReLU** salvo la del espacio latente y la última que usan **logística**.

Autoencoder básico

Error vs. época

Variando arquitecturas

- épocas = 10
- optimizador: RMSprop

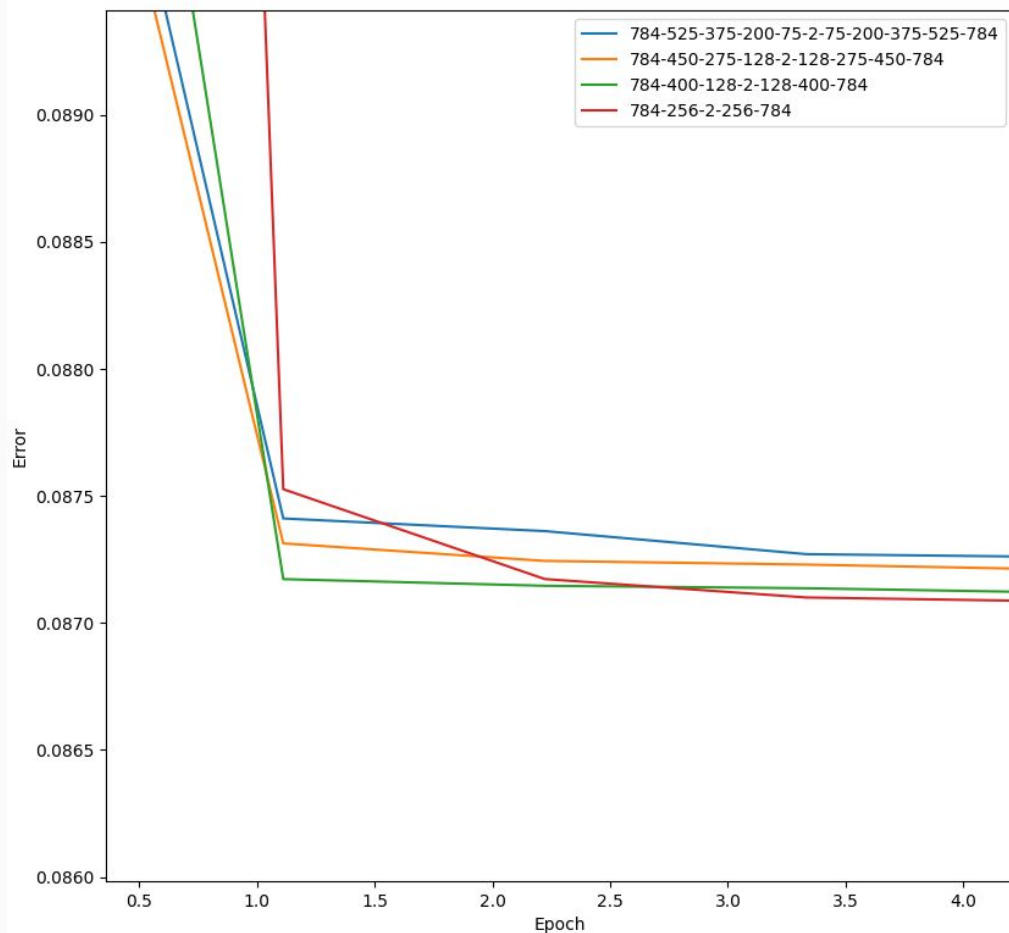


Autoencoder básico

Error vs. época

Variando arquitecturas

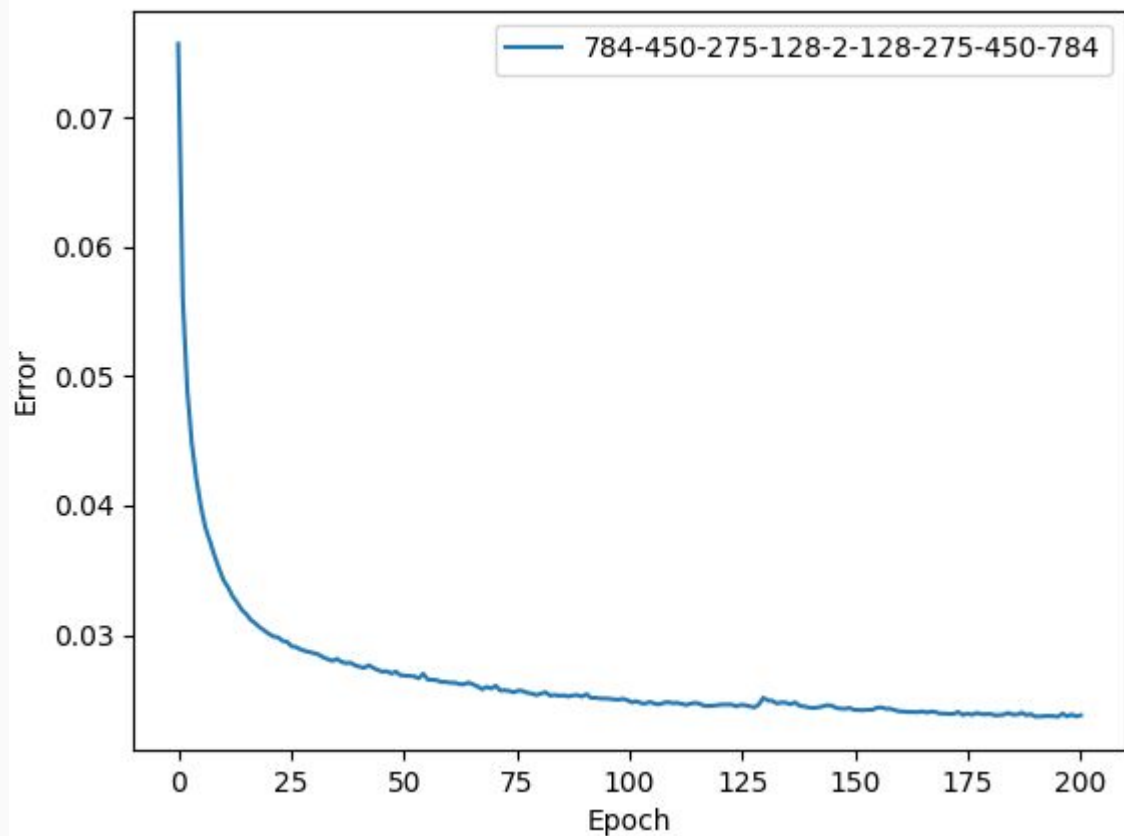
- épocas = 10
- optimizador: RMSprop



Autoencoder básico

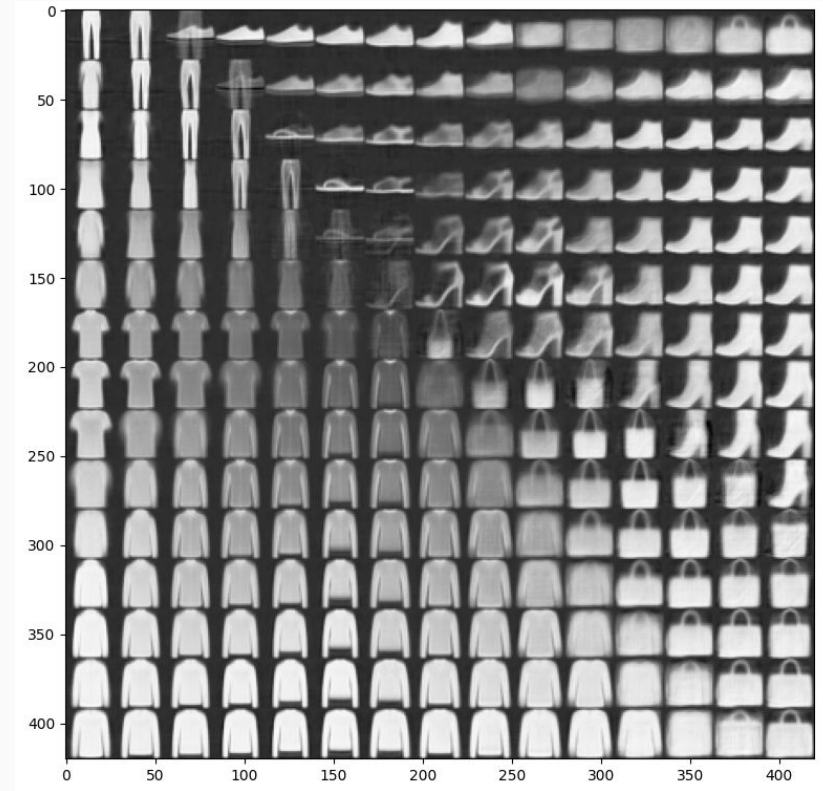
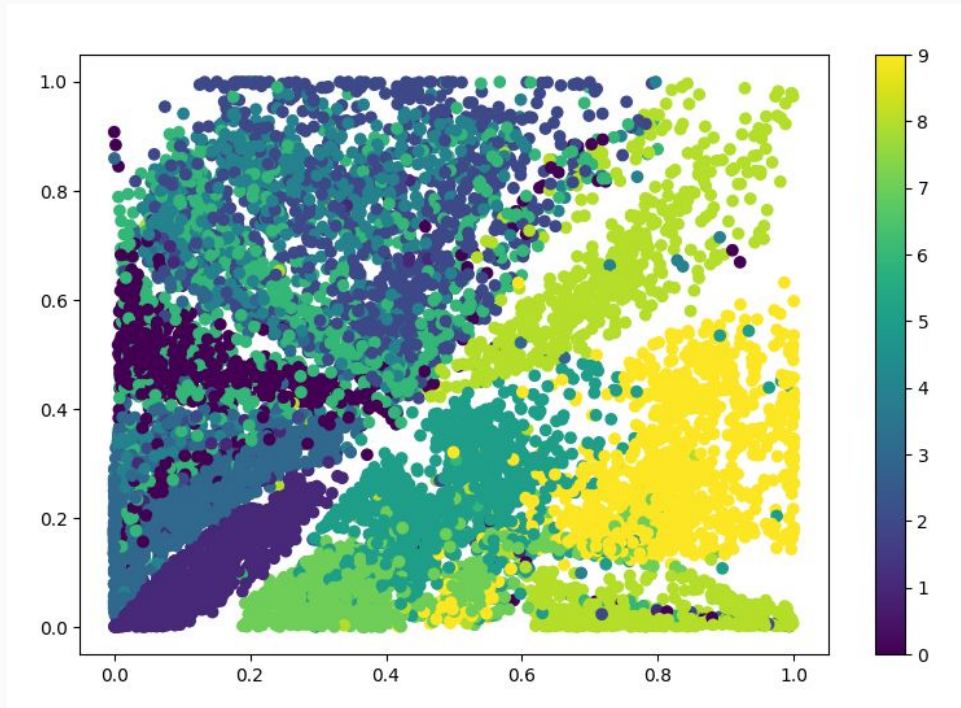
Error vs. época

- épocas = 200
- optimizador: RMSprop



Espacio latente

Representación

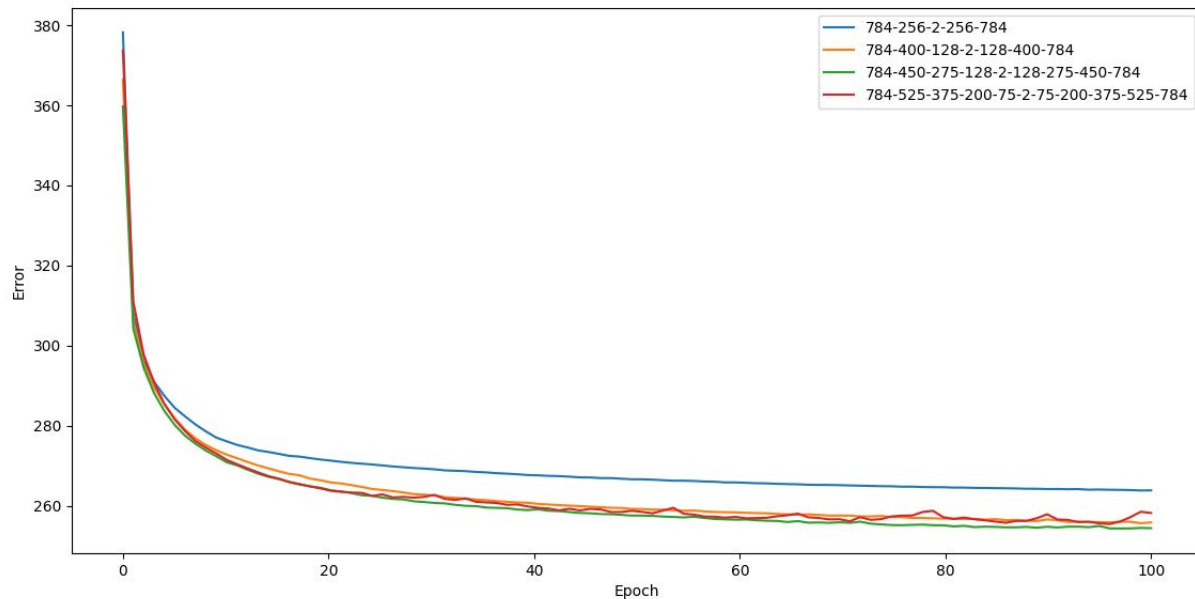


VAE

Error vs. época

Variando *arquitecturas*

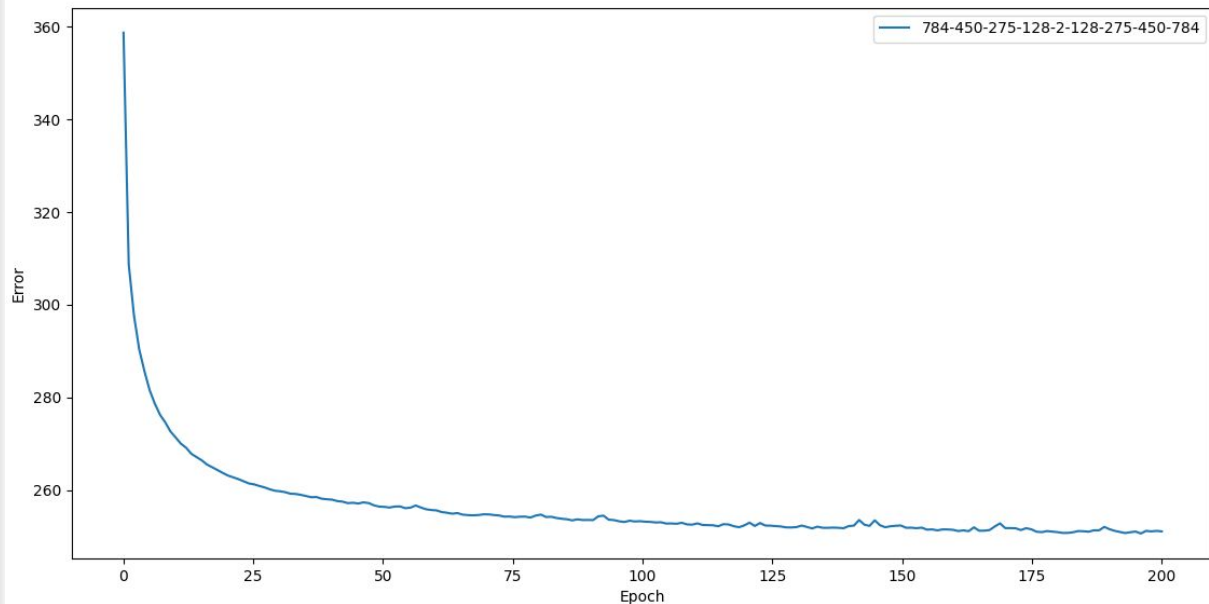
- épocas = 100
- optimizador: RMSprop



VAE

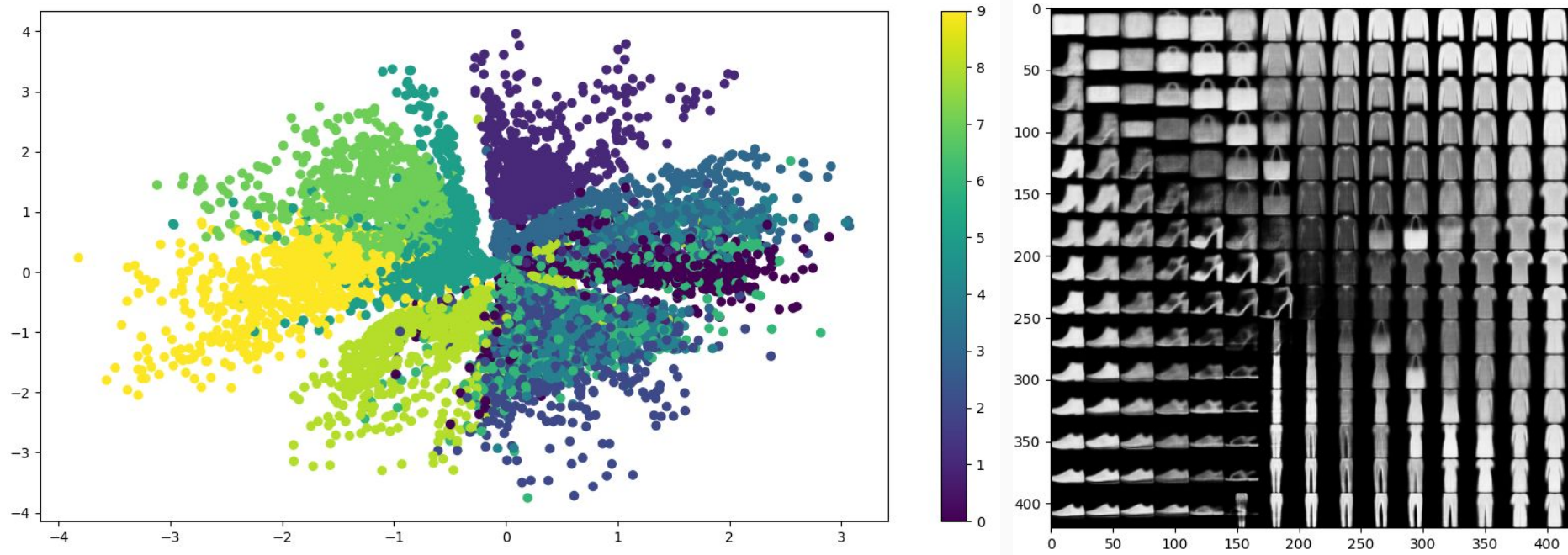
Error vs. época

- épocas = 200
- optimizador: RMSprop



VAE - Espacio latente

Representación



Conclusiones

- A menor dimensión del espacio latente, mayor compresión y error.
- Agregar capas no garantiza un error menor.
- A mayor cantidad y tamaño de capas, mayor es el tiempo de cómputo.
- Para reducción de ruido resulta más útil tener una arquitectura *overcomplete*
- A mayor ruido, mayor error en el aprendizaje del autoencoder

- El ruido gaussiano es más fácil de filtrar y aprender que el salt & pepper
- Es importante considerar el tiempo de ejecución y la precisión a la hora de elegir un método de optimización.
- Para generar nuevos datos con características similares conviene utilizar un Autoencoder variacional, en lugar de un Autoencoder básico

Gracias por su atención