

Sistemas de Inteligencia Artificial

TP 5 : Deep learning

GROUP 6 :

Desiree Melisa Limachi

Joseph Rouquette

Magdalena Flores Levalle

Matias Ezequiel Daneri

CONTENIDOS

01a

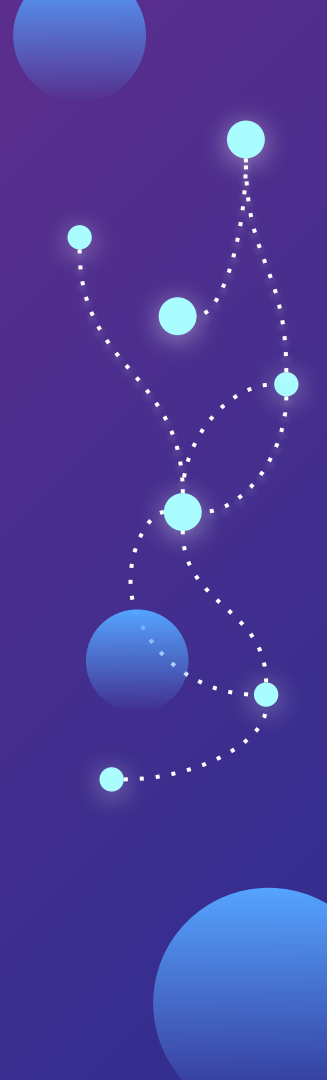
Linear
Autoencoder

01b

Denoising
Autoencoder

02

VAE



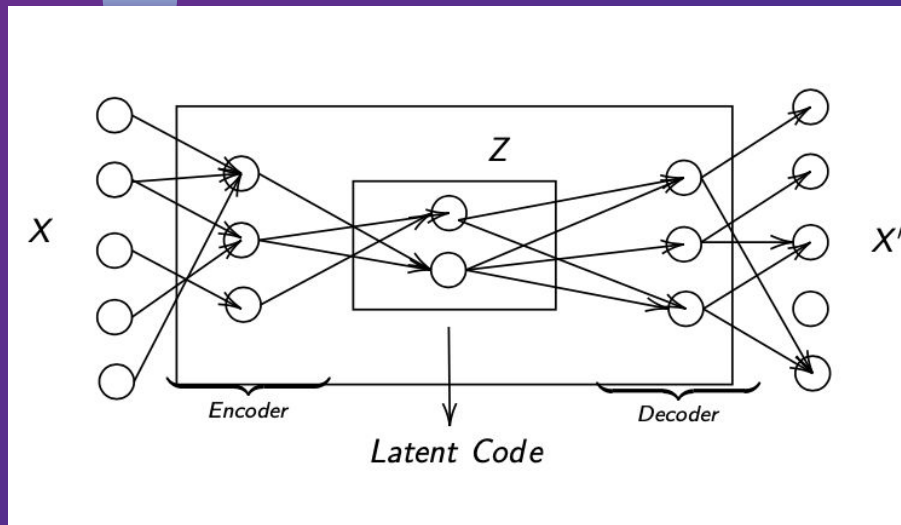


| 01a

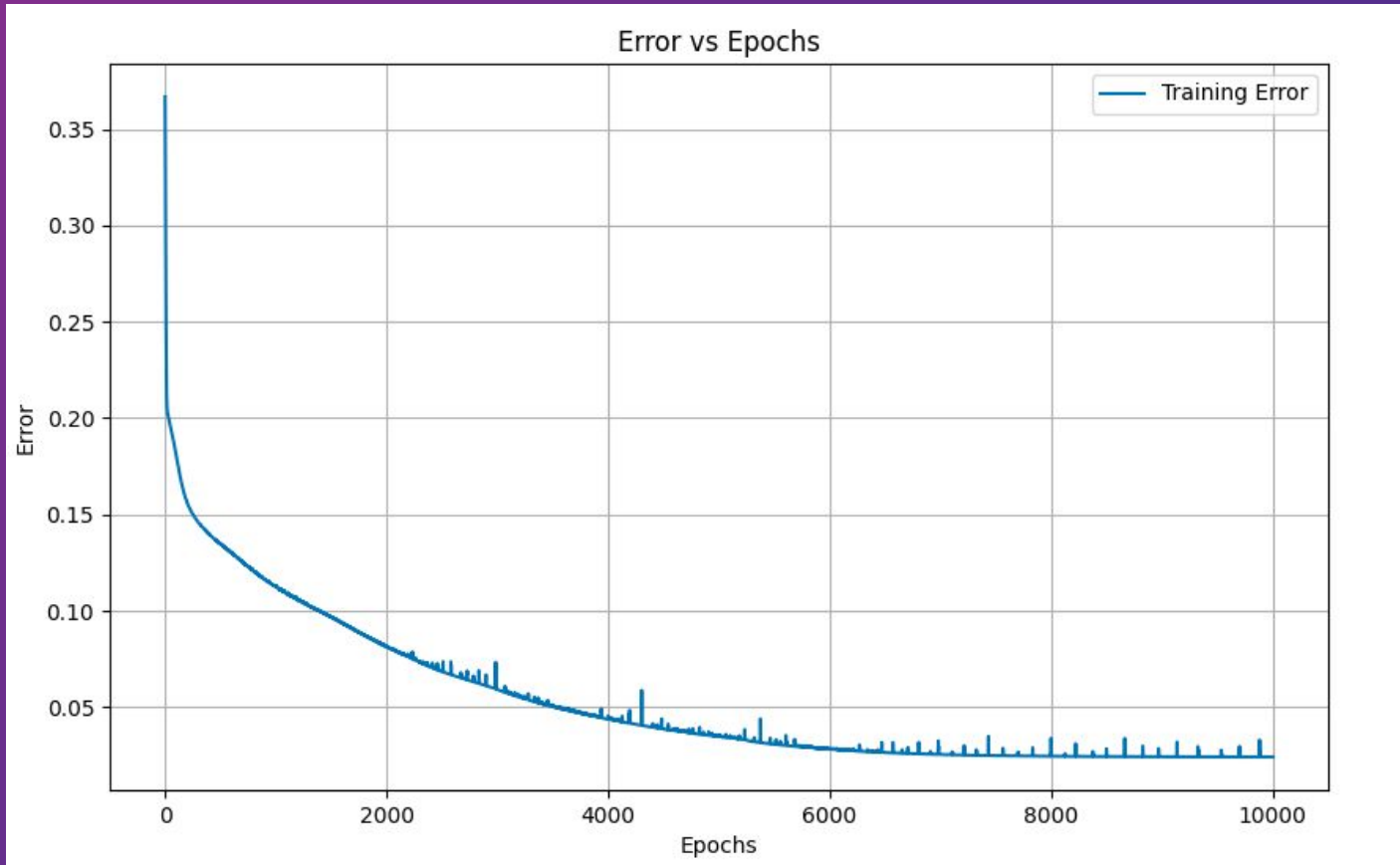
Linear Autoencoder

Arquitectura

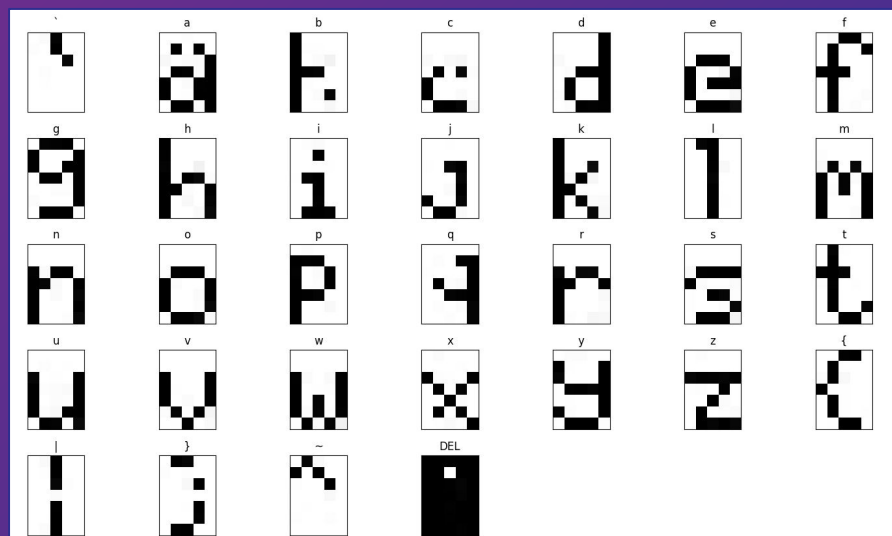
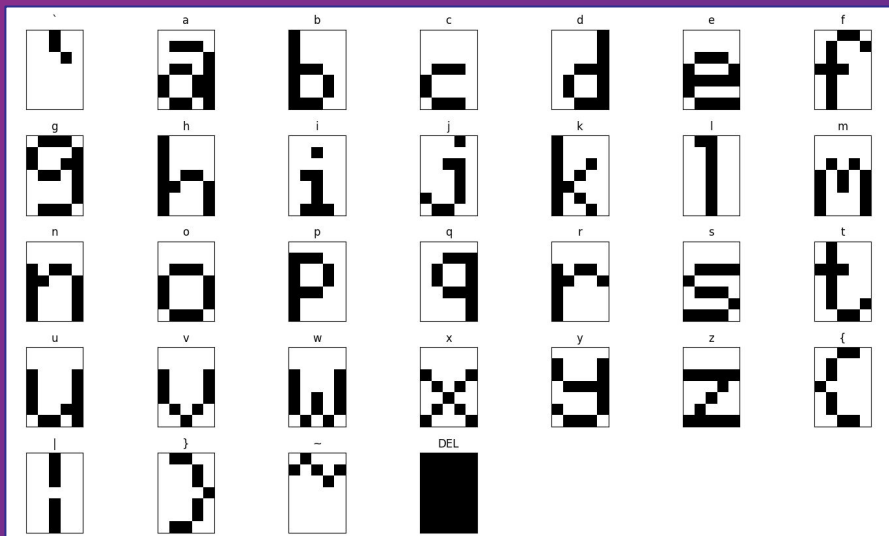
Encoder : variando
Decoder : variando
Activacion : Sigmoid
Epocas : 10000
Optimizador : Adam
Learning rate : 0,001



ERROR A TRAVÉS DE LAS ÉPOCAS



VARIANDO ARQUITECTURA



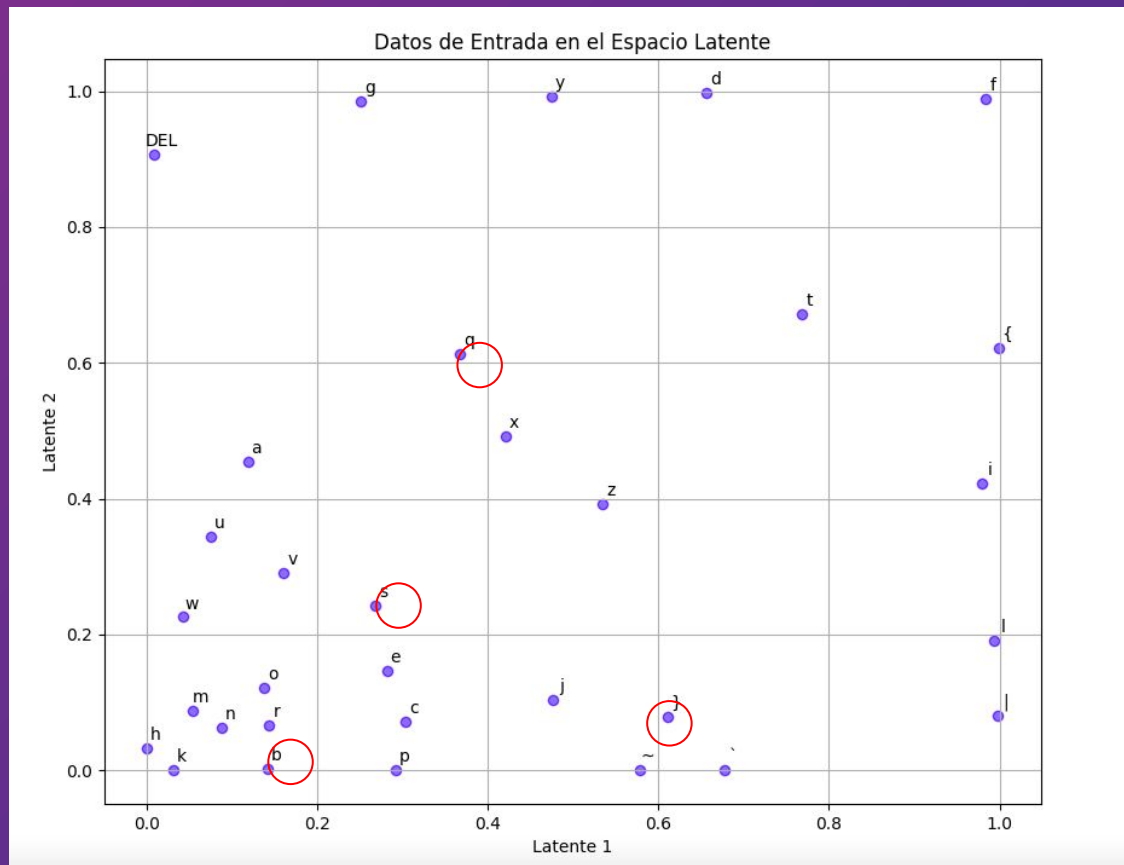
Se utilizó la arquitectura [35,20,10,2,10,20,35]

Las letras que cuentan más que 1 pixel de diferencia con la imagen del origen :

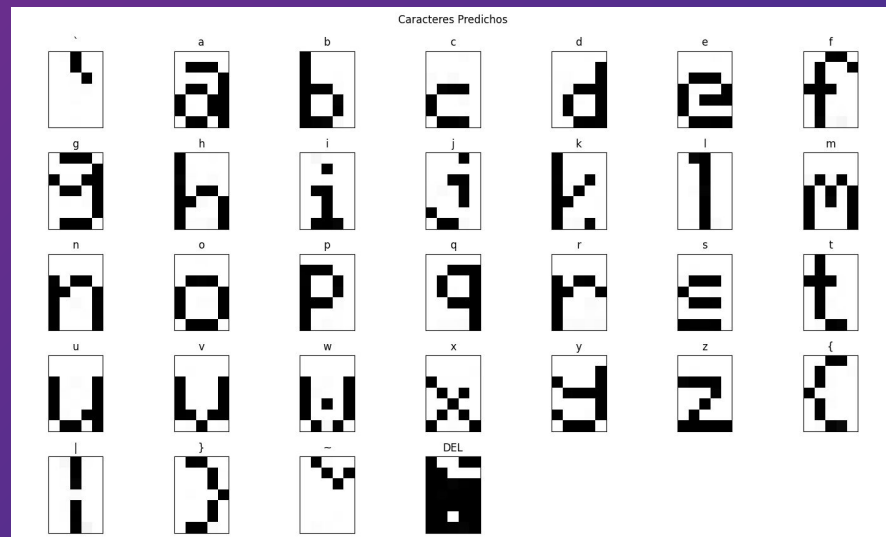
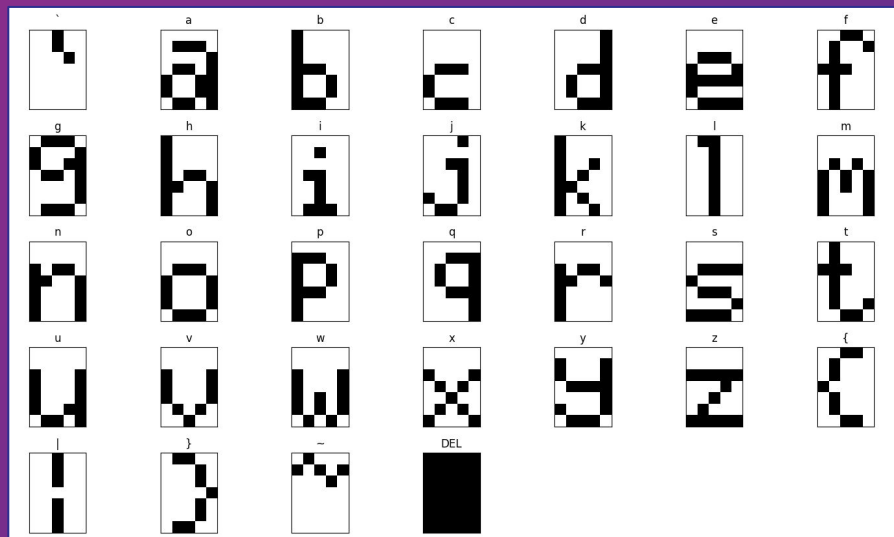
- b(3)
- q(2)
- }(2)
- s(2)

La red solo puede aprender un subconjunto de patrón (28/32)

ESPACIO LATENTE



VARIANDO ARQUITECTURA



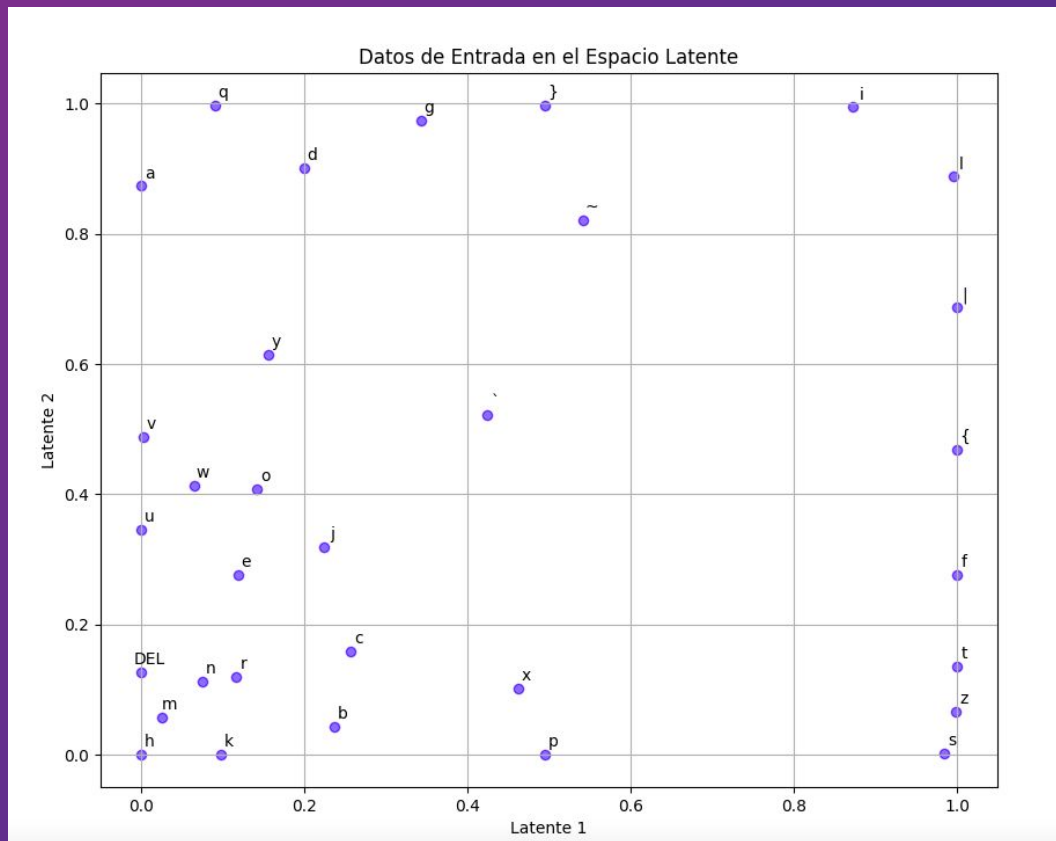
Se utilizó la arquitectura [35,25,15,2,15,25,35]

Las letras que cuentan más que 1 pixel de diferencia con la imagen del origen :

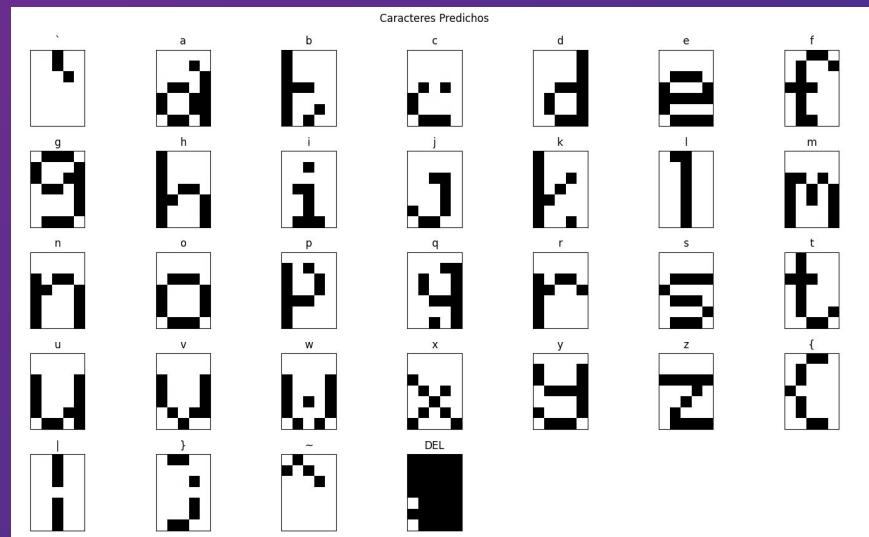
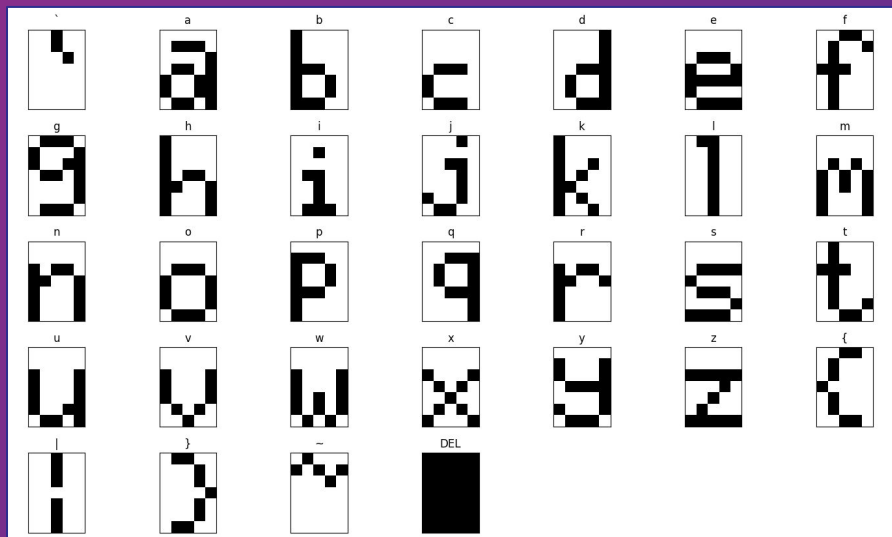
- v(2)
- DEL(6)

La red solo puede aprender un subconjunto de patrón (30/32)

ESPACIO LATENTE



VARIANDO ARQUITECTURA



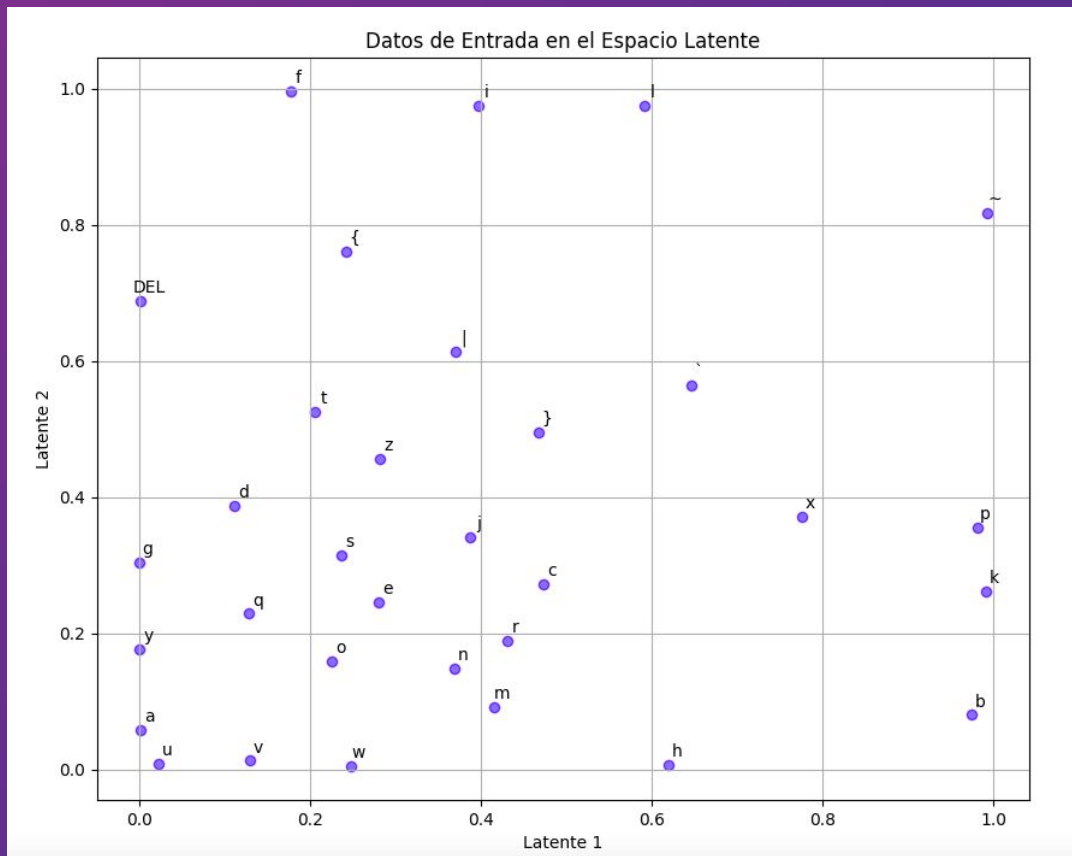
Se utilizó la arquitectura [35,30,20,10,2,10,20,30,35]

Las letras que cuentan más que 1 pixel de diferencia con la imagen del origen :

- a(2)
- q(2)
- b(2)
- DEL(2)

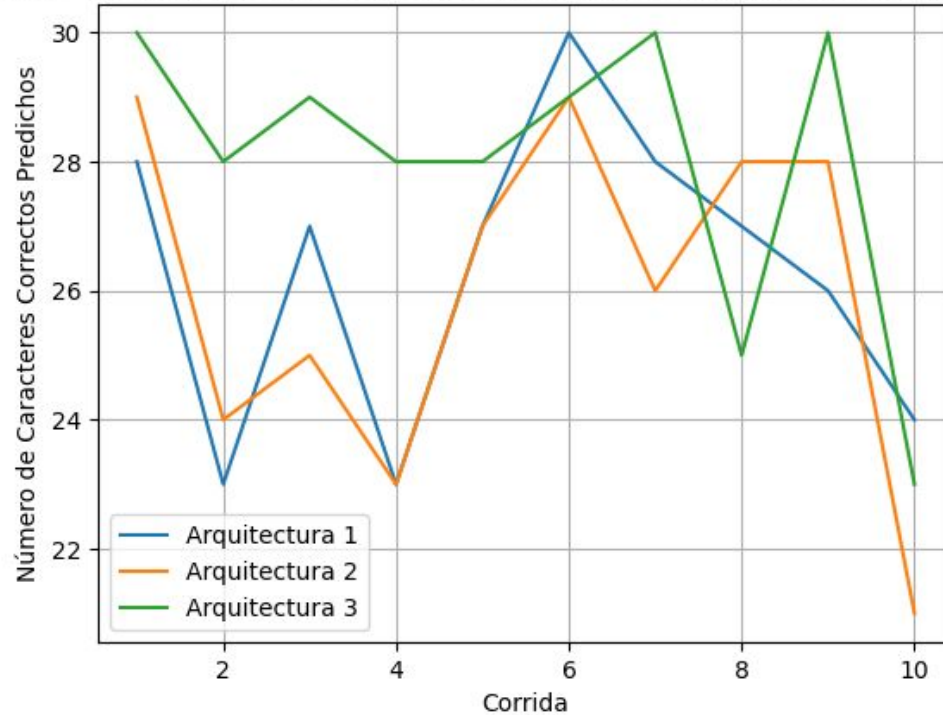
La red solo puede aprender un subconjunto de patrón (28/32)

ESPACIO LATENTE

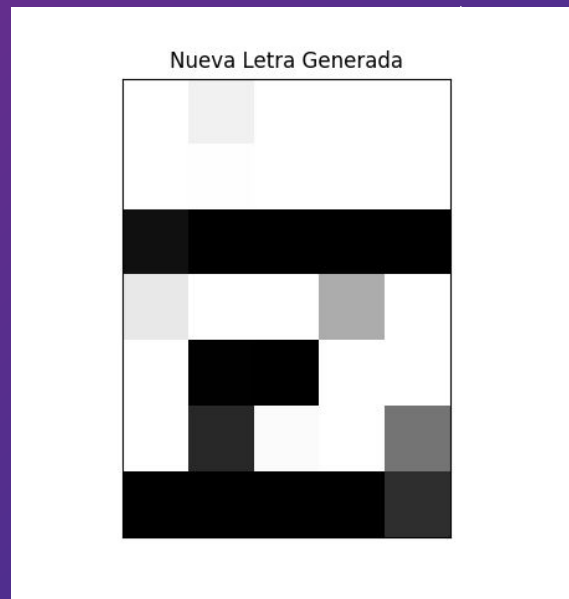
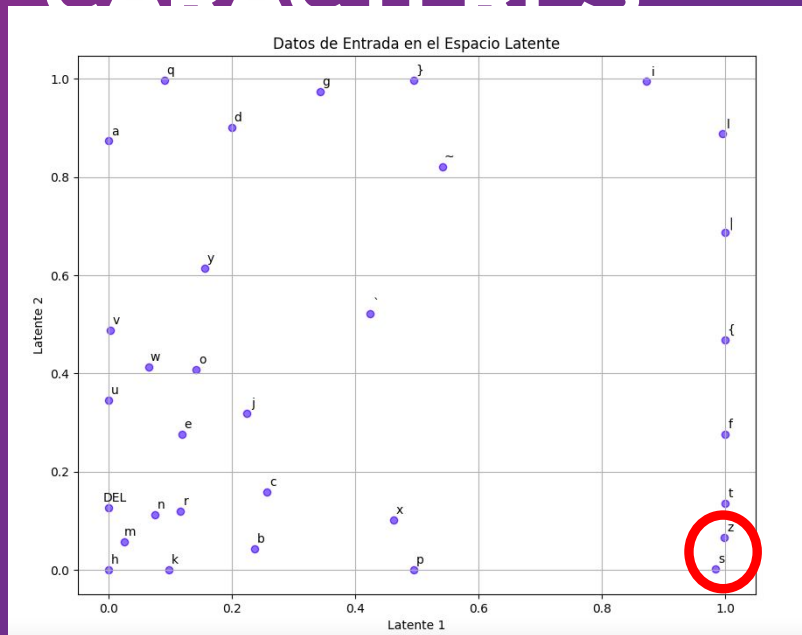


Comparación de cada arquitectura

Número de Caracteres Correctos Predichos por Corrida para Cada Arquitectura



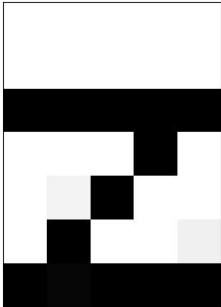
GENERANDO NUEVOS CARACTERES



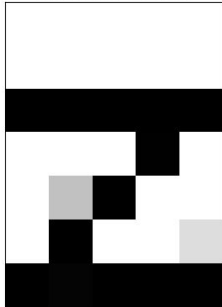
Tomamos un punto en el espacio latente y lo pasamos por el decodificador para generar una nueva letra → Variación entre la 'S' y la 'Z'.

GENERANDO NUEVOS CARACTERES

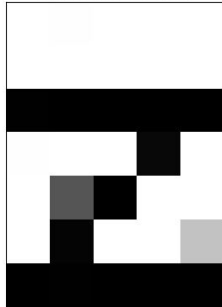
Nueva Letra Generada



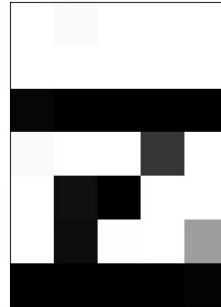
Nueva Letra Generada



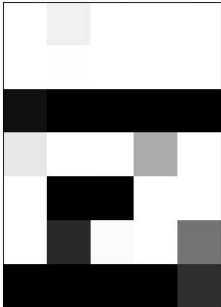
Nueva Letra Generada



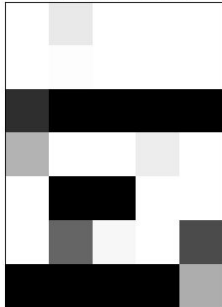
Nueva Letra Generada



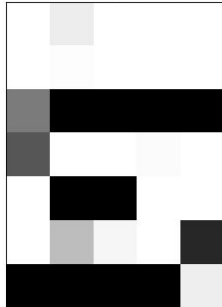
Nueva Letra Generada



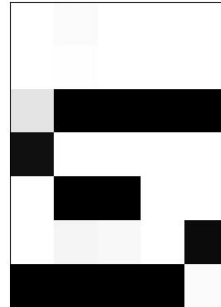
Nueva Letra Generada



Nueva Letra Generada



Nueva Letra Generada



Conclusiones

● Variación según la arquitectura.

Dos arquitecturas mantuvieron una cantidad de predicciones correctas similares entre sí, mientras que la última destacó por encima de las anteriores. Por ende, la arquitectura elegida es esencial para un buen entrenamiento del autoencoder.

● Compresión

Utilizando una capa latente de dos neuronas pudimos comprimir los caracteres con un error mínimo. Pocos fueron los casos donde se obtuvieron más de un carácter con más de un pixel erróneo.

● Capacidad de creación

En base a caracteres ya aprendidos, el espacio latente es una forma de crear nuevas variaciones inexistentes de los caracteres.

A partir de un punto específico, podemos decodificarlo y nos brindará una representación de letras aprendidas.

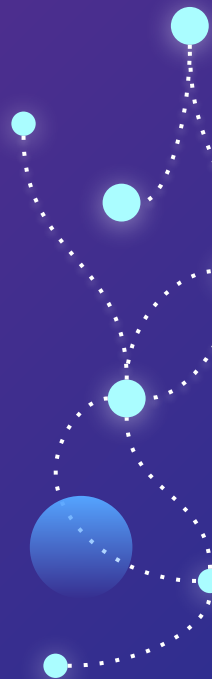


| 01b

Denoising Autoencoder

IDEAR NUEVA ARQUITECTURA

La idea entonces es en base de una muestra con ruido y la muestra original, entrenar el autoencoder para obtener la muestra original a partir de la que lleva ruido.



Arquitectura

Encoder : [35,30,20,10,2]

Decoder : [2,10,20,30,35]

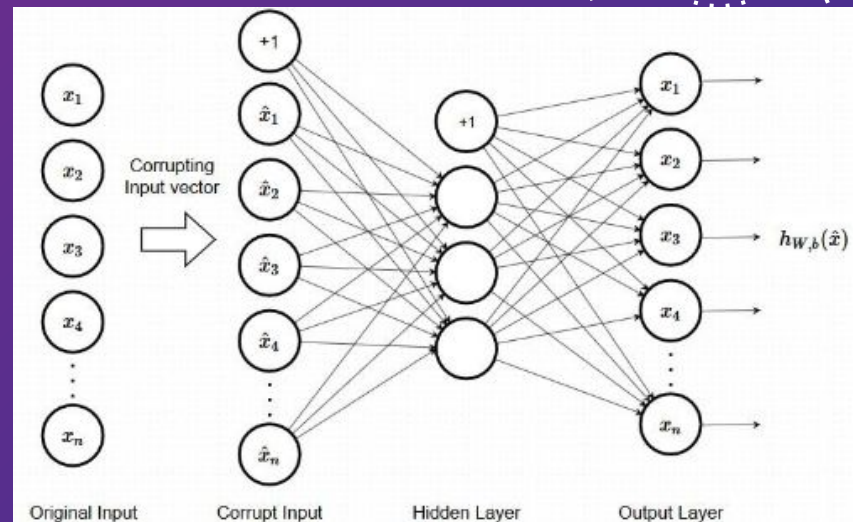
Activación : Sigmoid

Épocas : 10000

Optimizador : Adam

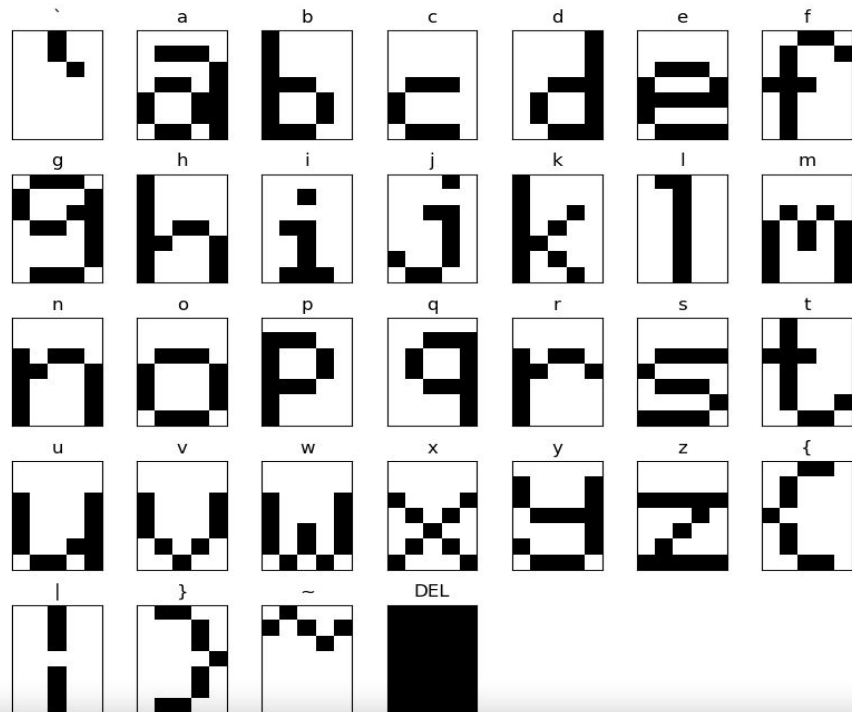
Learning rate : 0,001

Ruido : 10%

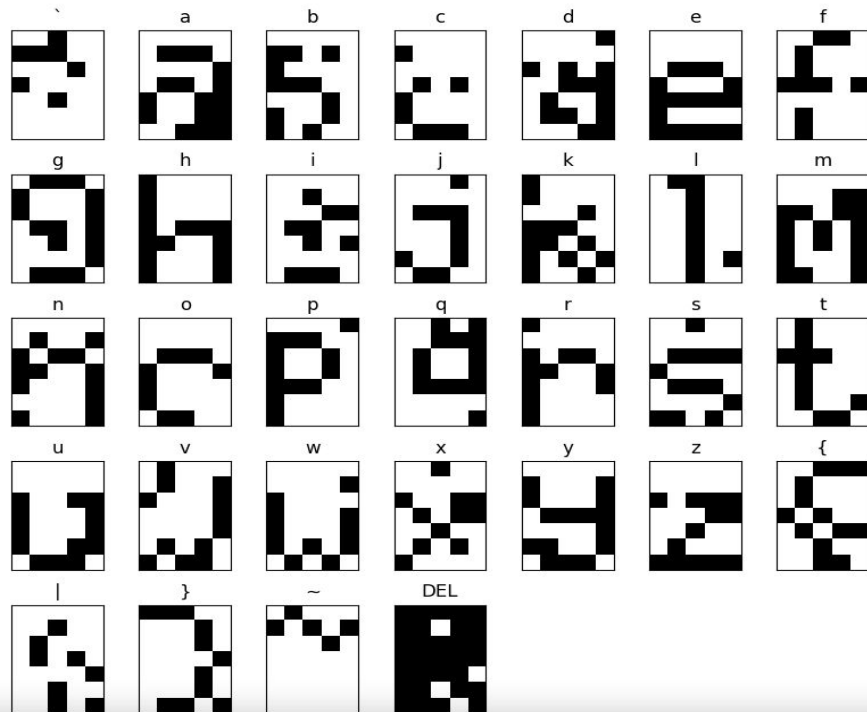


RUIDO : Salt & Pepper

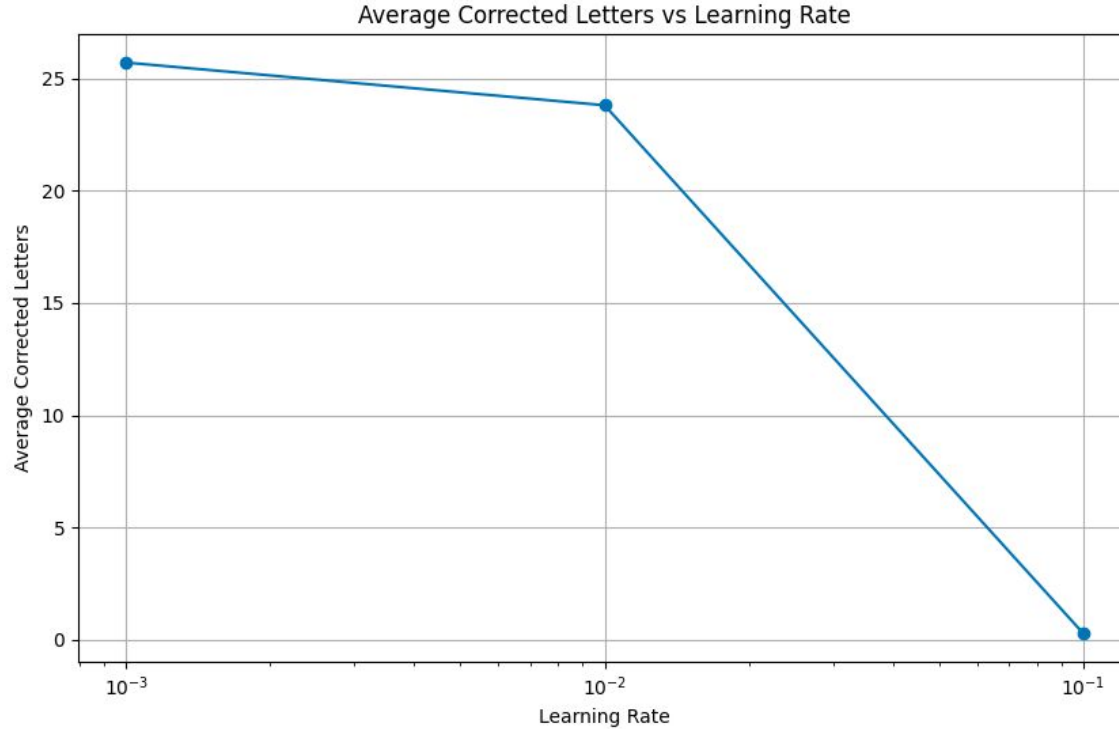
Caracteres Originales



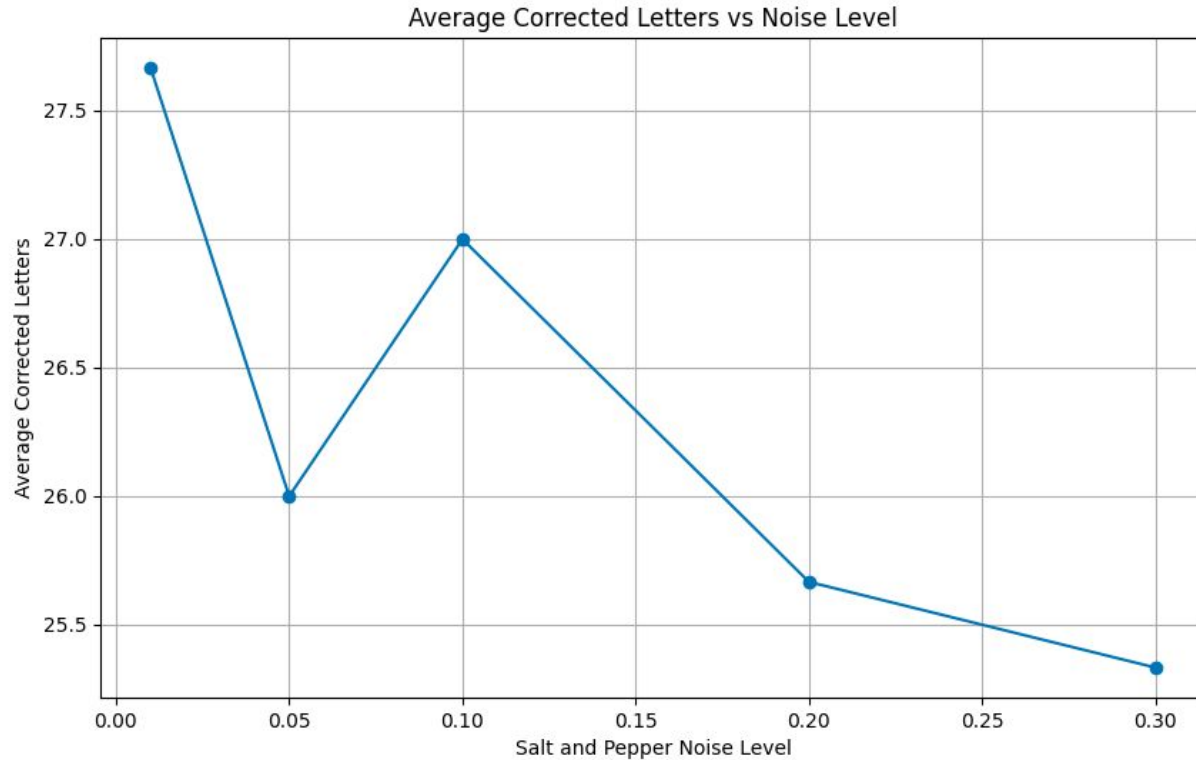
Caracteres con ruido



VARIANDO EL LEARNING RATE

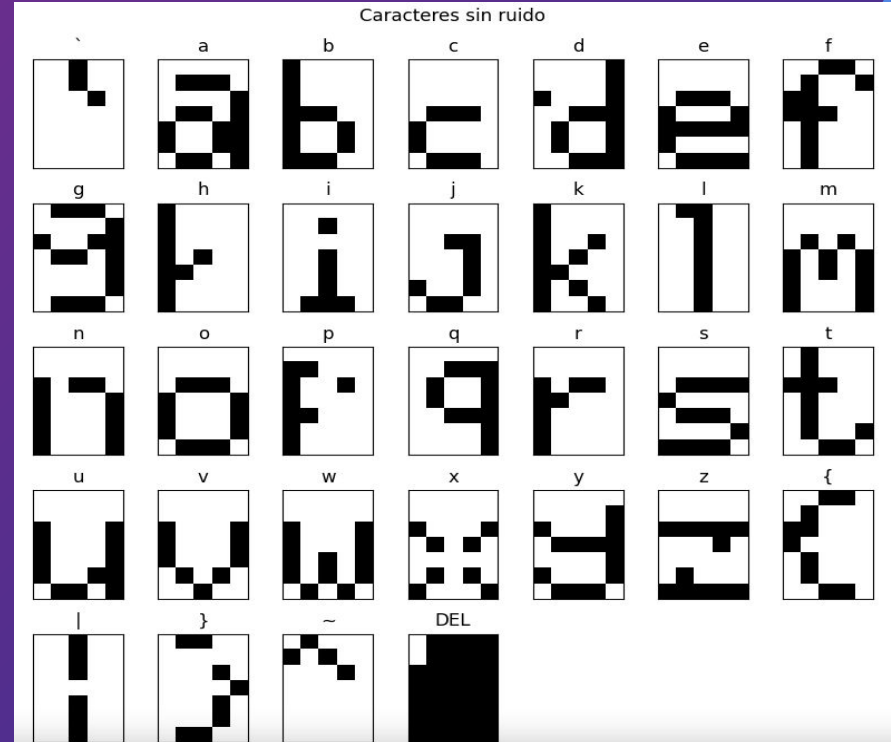
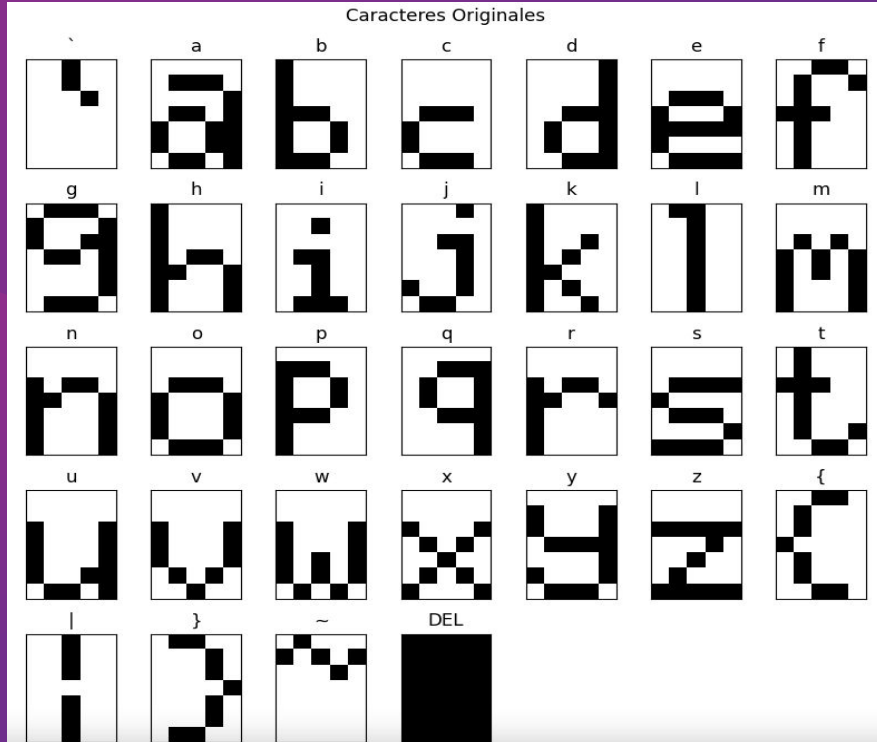


VARIANDO EL RUIDO



Promedio de
3 corridas

Eliminación del ruido



Conclusiones

● Ruido

Más allá de 30 de ruido, la capacidad de nuestra Autoencoder Denoising para divulgar imágenes de caracteres se cuestiona.

● Efectividad de eliminación del ruido

En general, es un muy buen disyuntor de datos simple como este.

● Learning rate

Un learning rate demasiado grande hace completamente inútil nuestro autoencoder



| 02

Variational Autoencoder

Arquitectura

Encoder : variando

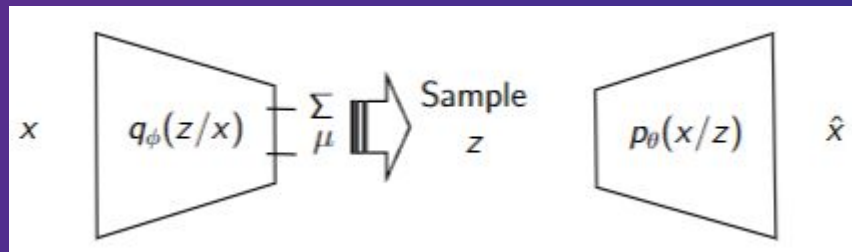
Decoder : variando

Activacion : ReLU

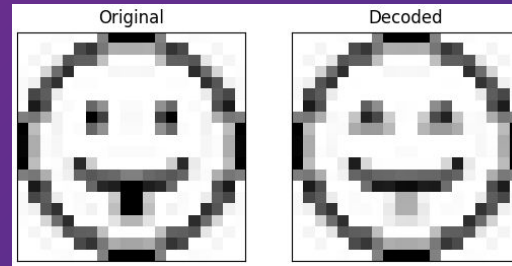
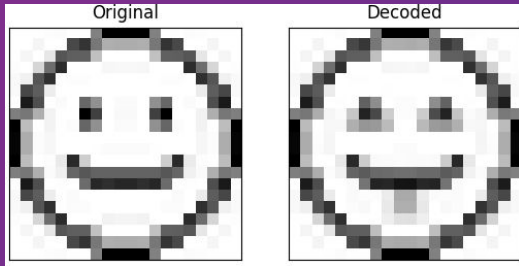
Epocas : 1000

Optimizador : Adam

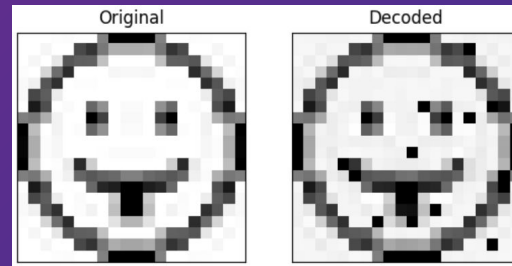
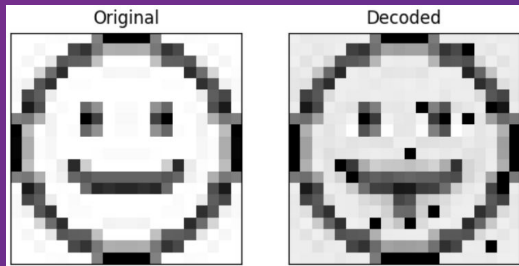
Learning rate : 0,001



VARIANDO ARQUITECTURA



ReLU +
Sigmoid



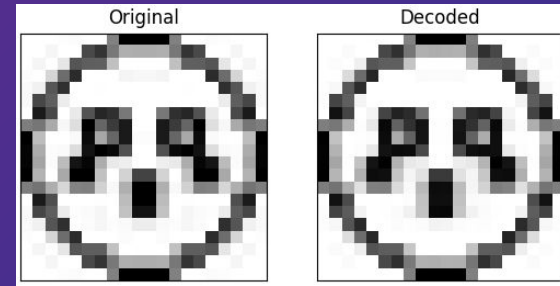
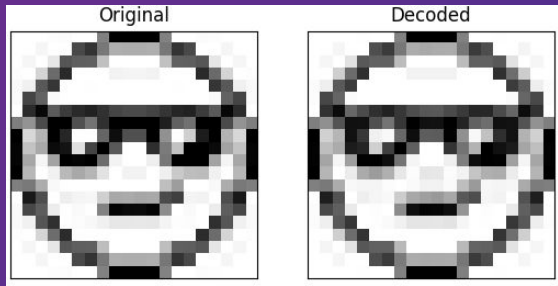
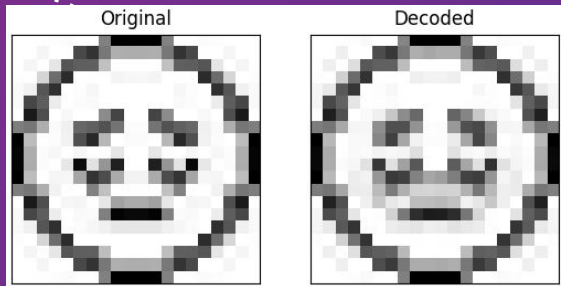
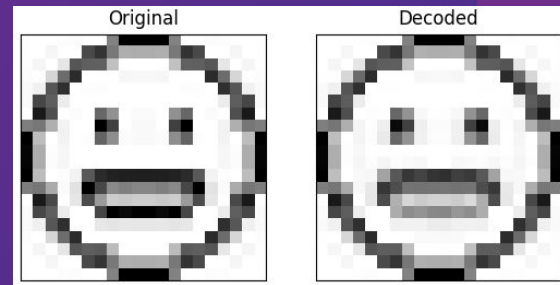
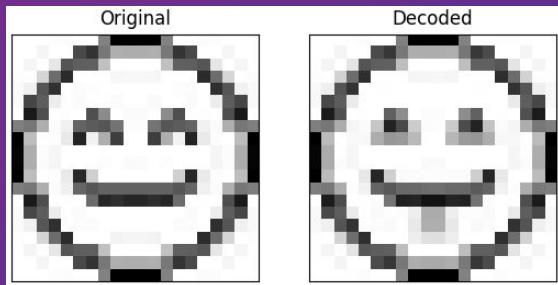
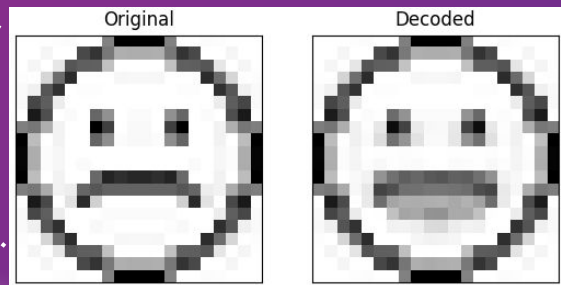
ReLU

Se utilizó la arquitectura [400, 300, 200, 100, 2, 100, 200, 300, 400]

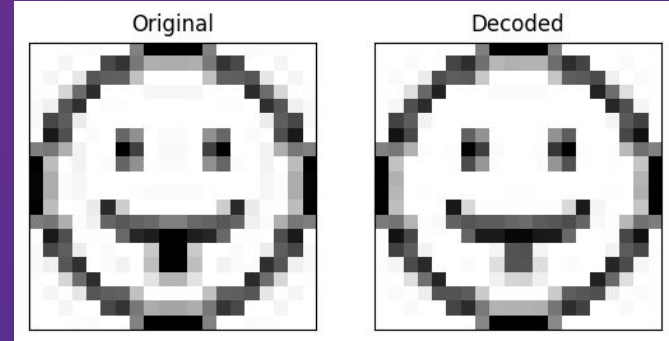
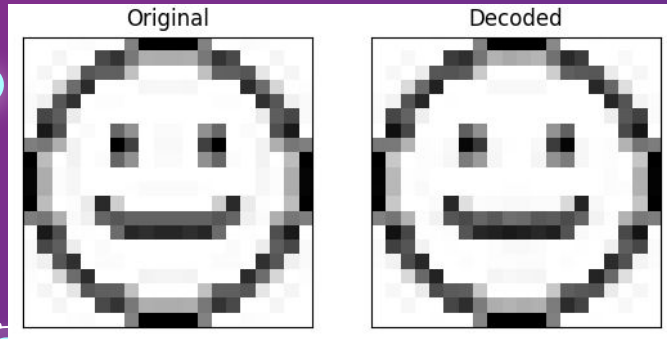
Los emojis se repiten bastante y tienen un mayor número de píxeles incorrectos

La red puede aprender un subconjunto de patrón (4/8)

VARIANDO ARQUITECTURA



VARIANDO ARQUITECTURA

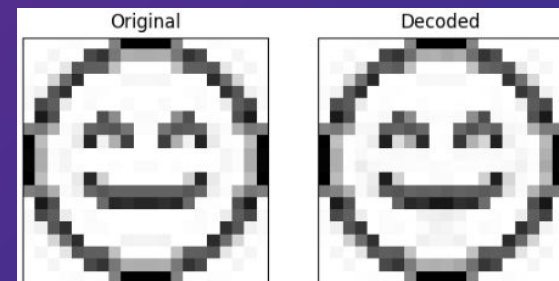
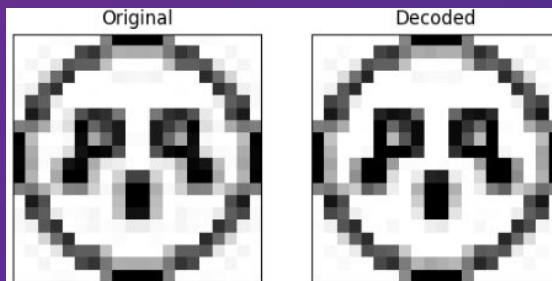
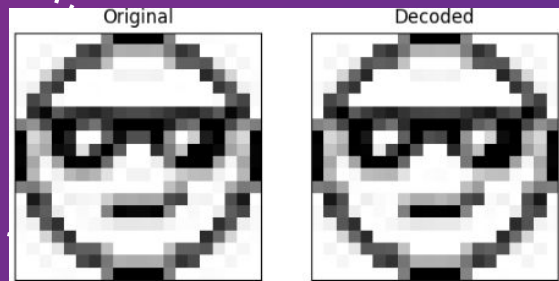
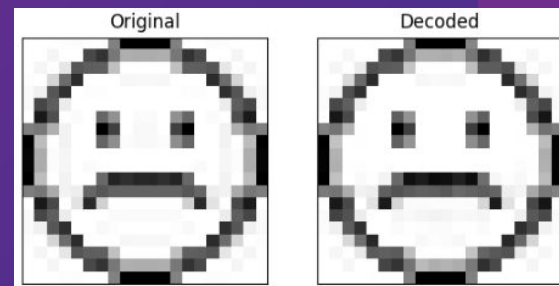
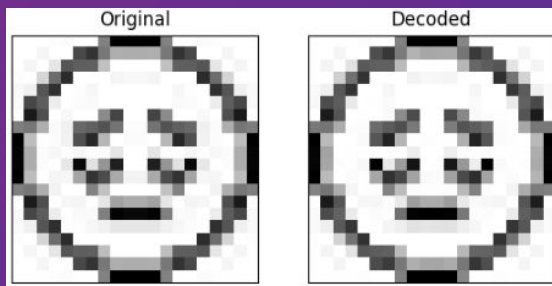
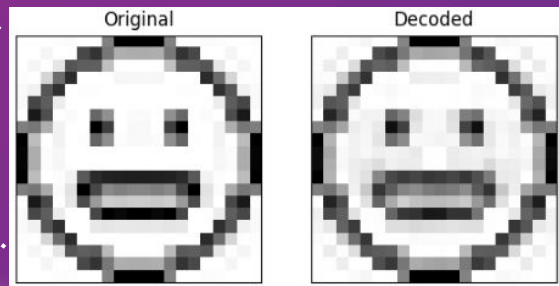


Se utilizó la arquitectura [400,300,200,20,200,300,400]

Los emojis presentan mínimos píxeles diferentes a la imagen original, pero no son tenidos en cuenta para ver una imagen diferente

La red puede aprender un subconjunto de patrón (8/8)

VARIANDO ARQUITECTURA



GENERANDO NUEVOS EMOJIS



Se generan 10 imágenes intermedias que muestran una transición suave de un emoji a otro.

Utilizamos la interpolación ponderada de las imágenes latentes y el decodificador del VAE para crear estas transiciones, resultando en nuevas combinaciones visuales de emojis.

Conclusiones

● Espacio latente

En este caso, tener un espacio latente más grande al lineal/denoising es necesario para poder tener una mejor variabilidad de las imágenes, ya que podría generar las mismas

● Normalizar grises

Podemos ver en la primer arquitectura un fondo gris, no permitiendo tener todo en la misma escala, esto se realizó utilizando en la última capa la función de activación Sigmoid

The background is a gradient of purple and blue. It features several decorative elements: dotted white lines that curve and loop across the frame, and various circles. Some circles are solid light blue, while others are white with a light blue outline. There are also larger, semi-transparent blue and purple circles scattered throughout. The overall aesthetic is modern and geometric.

GRACIAS!