



# TP5: Deep Learning

## Grupo 2

Tomás Álvarez Escalante (60127)

Alejo Francisco Caeiro (60692)

Lucas Agustín Ferreiro (61595)

Román Gómez Kiss (61003)

# TABLA DE CONTENIDOS

01

Linear  
Autoencoder

Ejercicio 1.a

02

Denoising  
Autoencoder

Ejercicio 1.b

03

Variational  
Autoencoder

Ejercicio 2

...

The background features a light blue-grey color with various geometric shapes and lines. In the top-left, there is a green plus sign. A dark blue line with small circles at its ends runs diagonally from the top-left towards the center. Another dark blue line runs vertically on the right side, with a red and white striped rectangular block positioned next to it. In the bottom-left, there are four red 'x' marks arranged in a small cluster. The overall aesthetic is clean and modern, resembling a technical or architectural drawing.

01

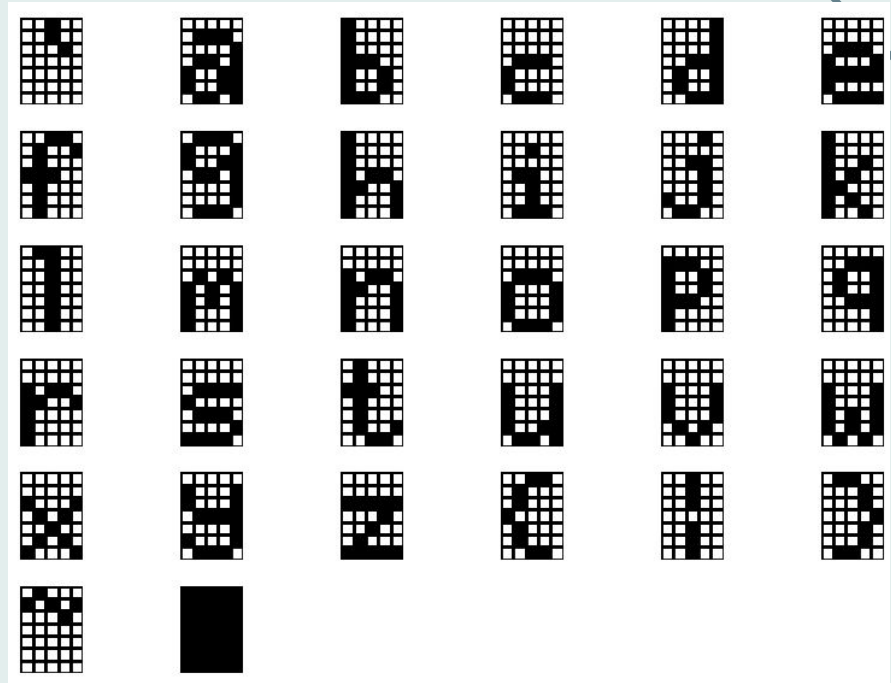
# Linear Autoencoder

# Problema

Se tiene un conjunto de 32 imágenes binarias de tamaño 7x5 para representar ciertos caracteres.

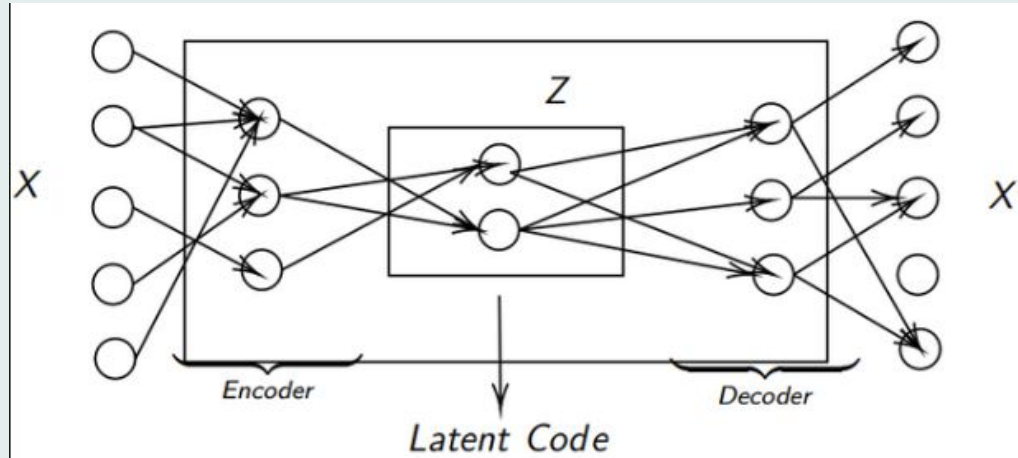
Se busca representar estos datos mediante un autoencoder con un espacio latente de dos dimensiones y con un error máximo de 1 pixel.

También se busca mostrar la capacidad de la red de generar un nuevo carácter que no pertenece al conjunto de entrenamiento.



# Autoencoder

Son redes neuronales de aprendizaje no supervisado cuyo objetivo principal es reducir la dimensionalidad. Están conformados por dos perceptrones multicapa, donde la salida de la primera red se conecta con la entrada de la segunda red, la cual tiene la distribución invertida de neuronas en las capas y como salida tiene la misma dimensión que la entrada de la primera red. La idea es minimizar  $\|X-X'\|^2$

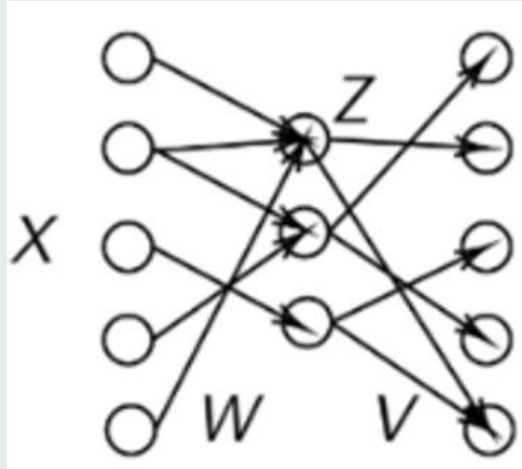


$Z$  nos permite tener una nueva codificación de  $X$  en una dimensionalidad más pequeña.

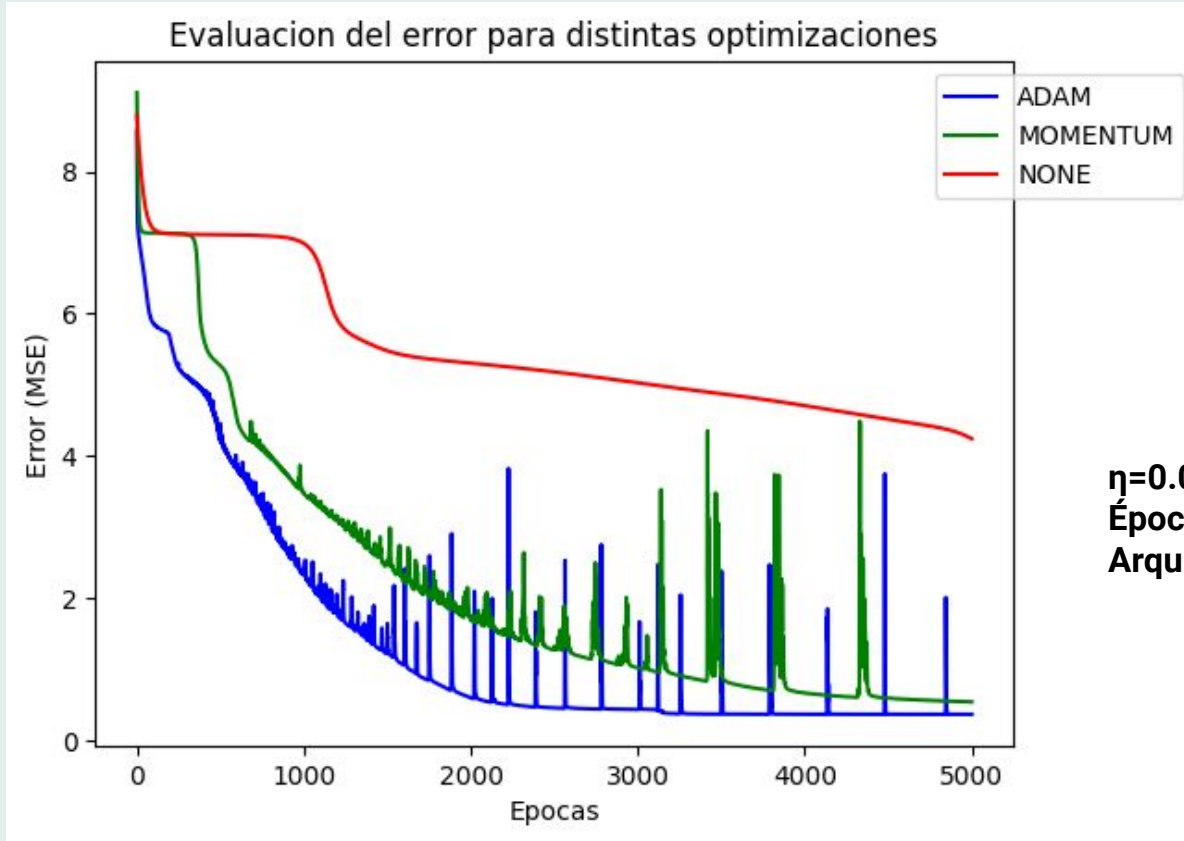
# Linear Autoencoder

Los perceptrones involucrados son lineales; una vez que el autoencoder aprende minimiza  $\|X - ZV^T\|$

La salida del espacio latente  $Z$  del autoencoder lineal son las salidas de las proyecciones de los datos en los componentes principales de PCA.



# Optimizaciones



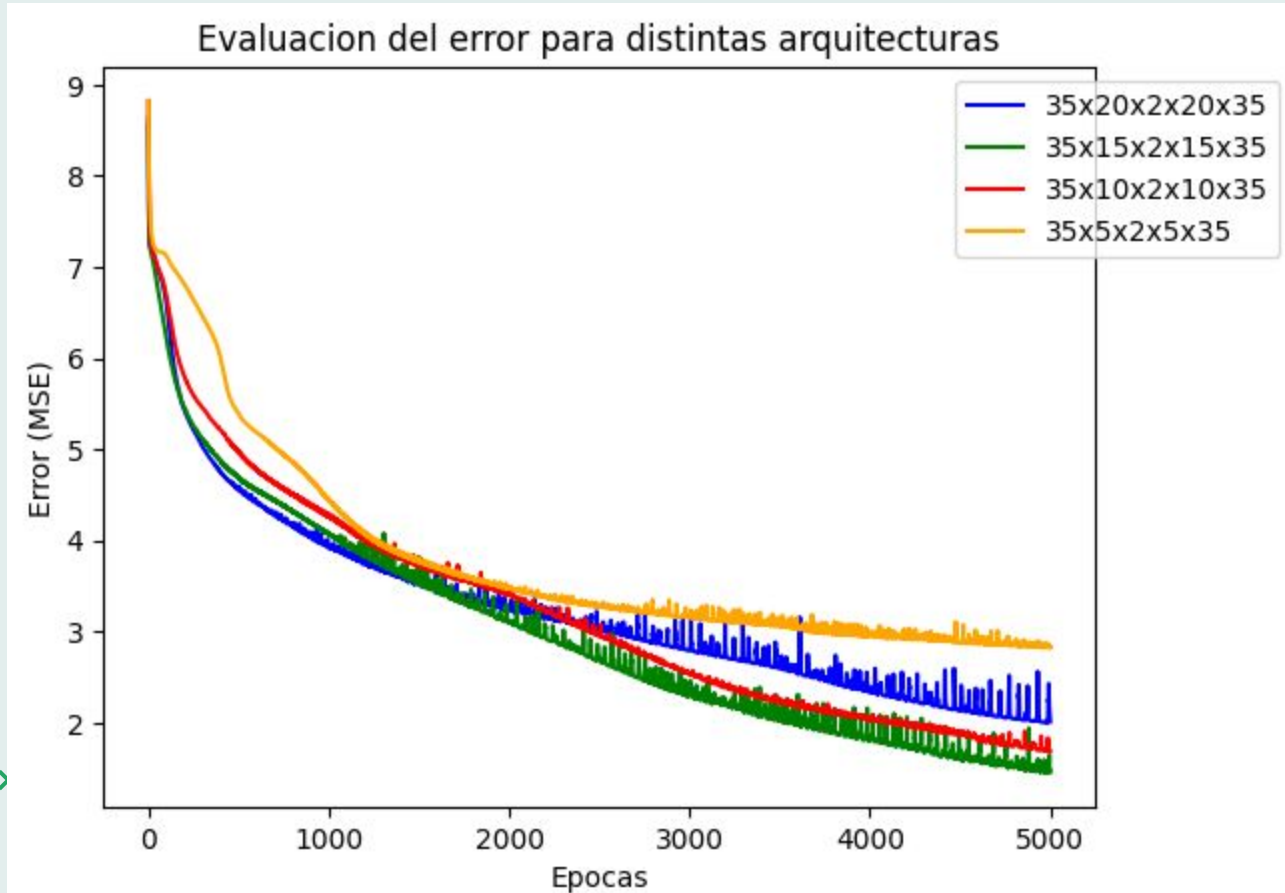
$\eta=0.001$

Épocas: 5000

Arquitectura: 35-10-2-10-35

```
"alpha": 0.8,  
"beta1": 0.9,  
"beta2": 0.999,  
"epsilon": 1e-8,
```

# Variando la cantidad de nodos de la capa intermedia

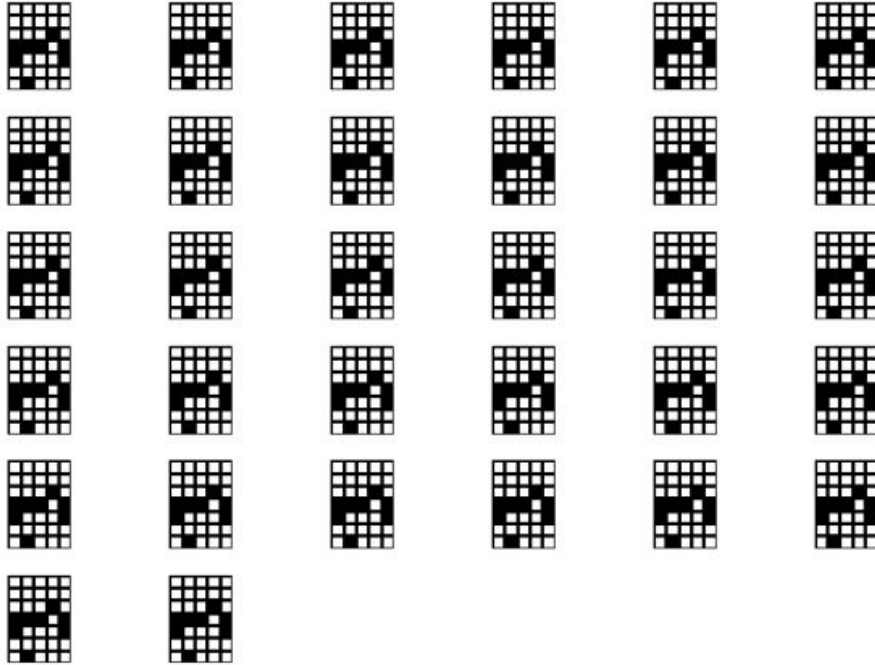




# Variando el learning rate



## Predicted



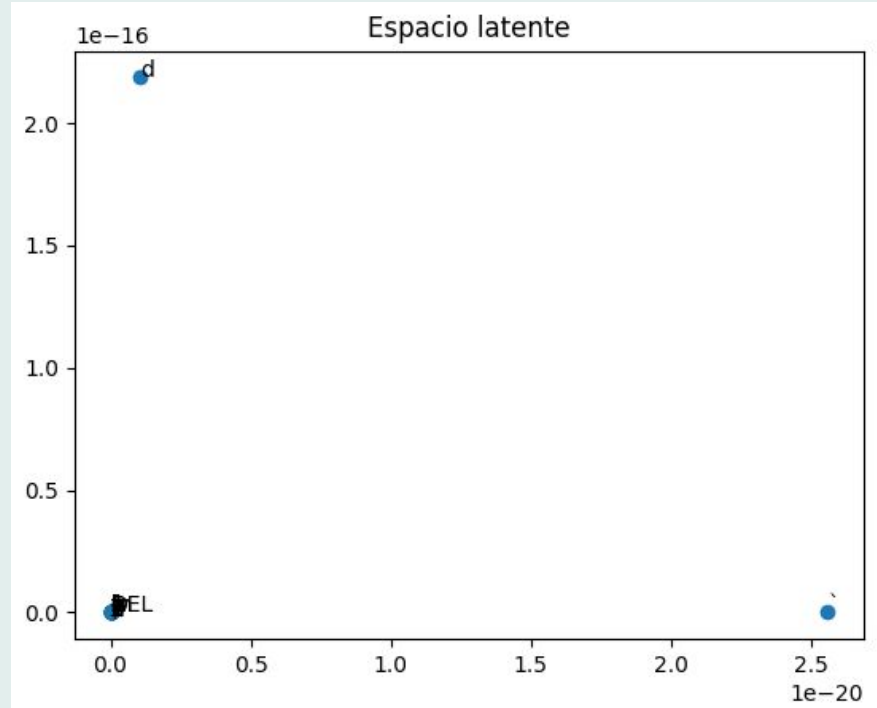
x x  
x x

$\eta=0.1$

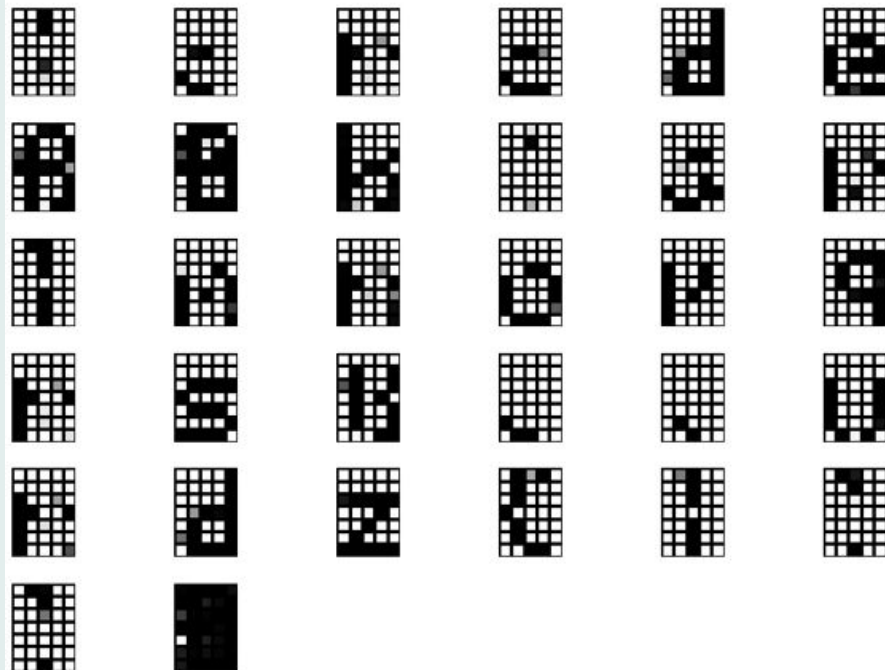
Épocas: 20000

Arquitectura: 35-10-2-10-35

MSE: 12.34375



## Predicted



x x  
x x

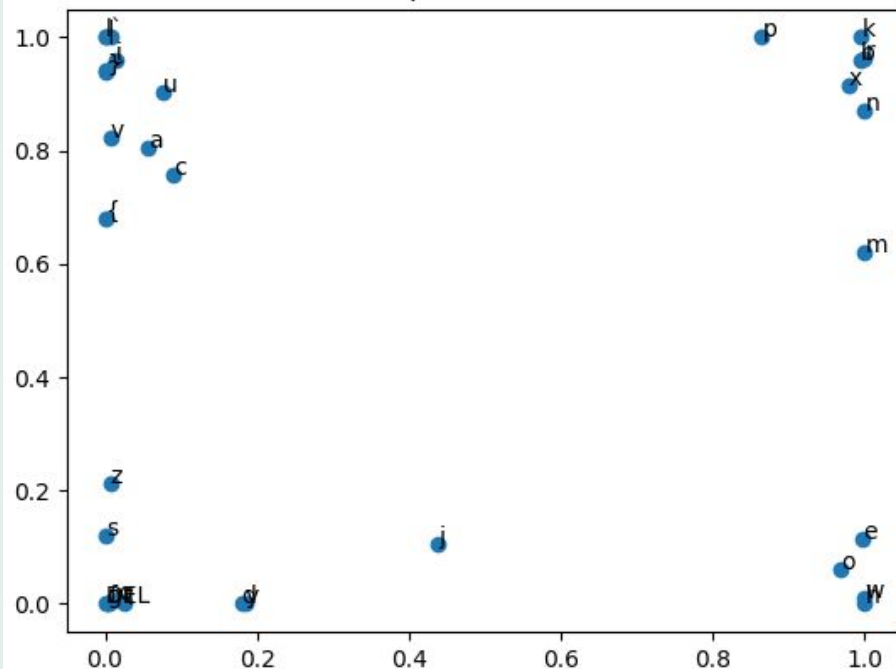
$\eta=0.01$

Épocas: 20000

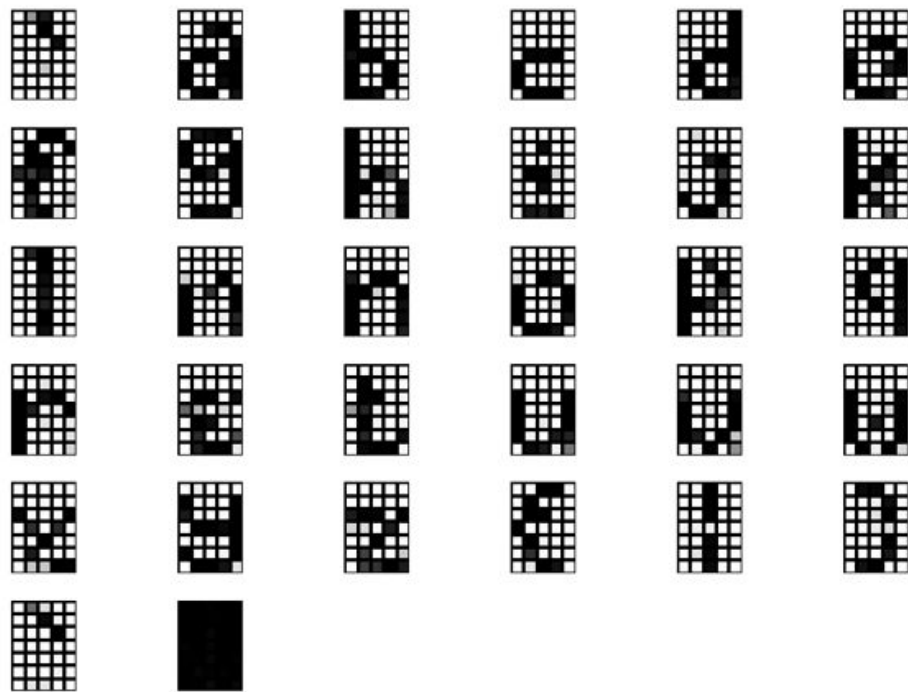
Arquitectura: 35-10-2-10-35

MSE: 4.30261

## Espacio latente



## Predicted



× ×  
× ×

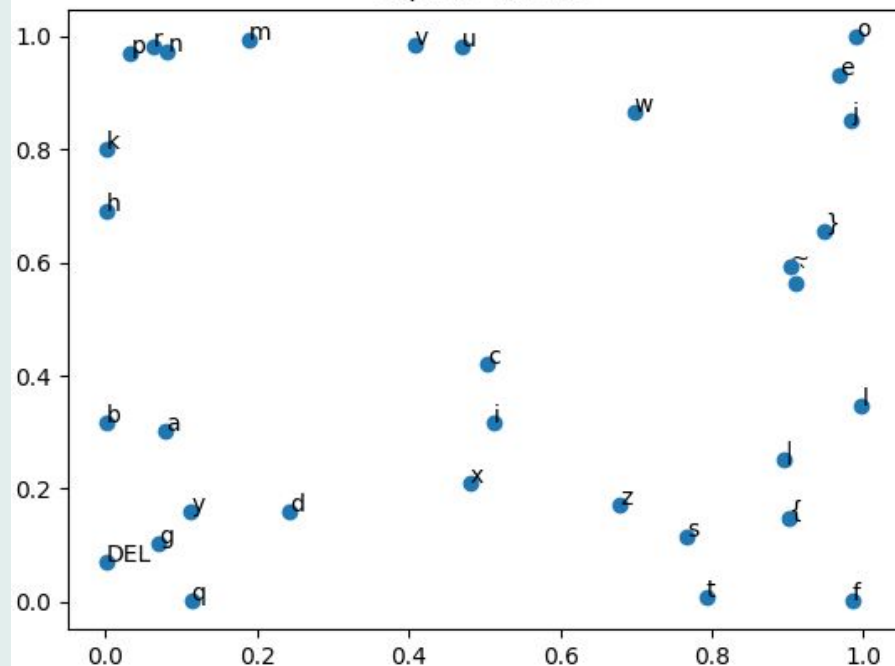
$\eta=0.001$

Épocas: 20000

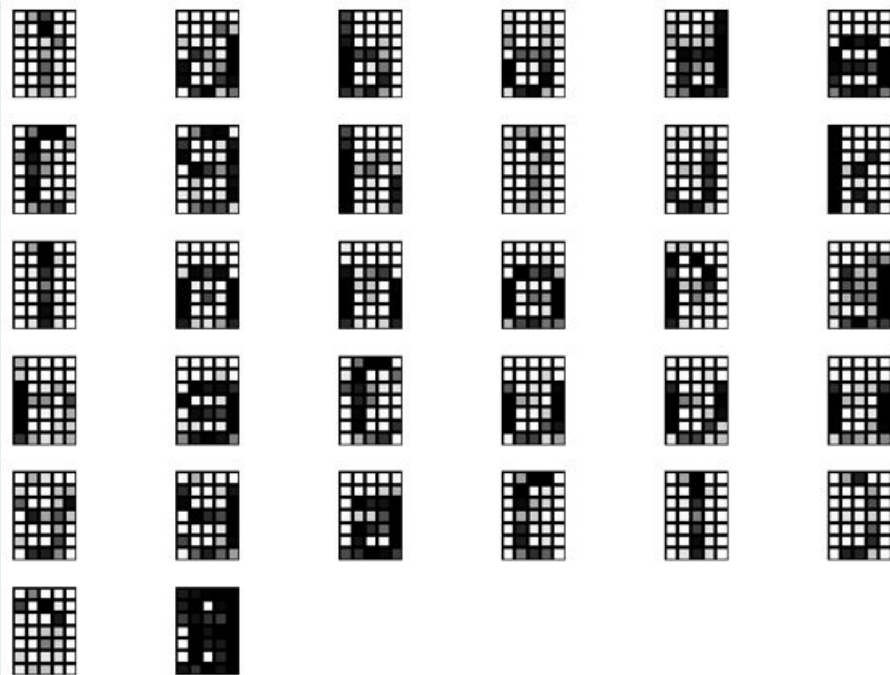
Arquitectura: 35-10-2-10-35

MSE: 1.15914

## Espacio latente



## Predicted

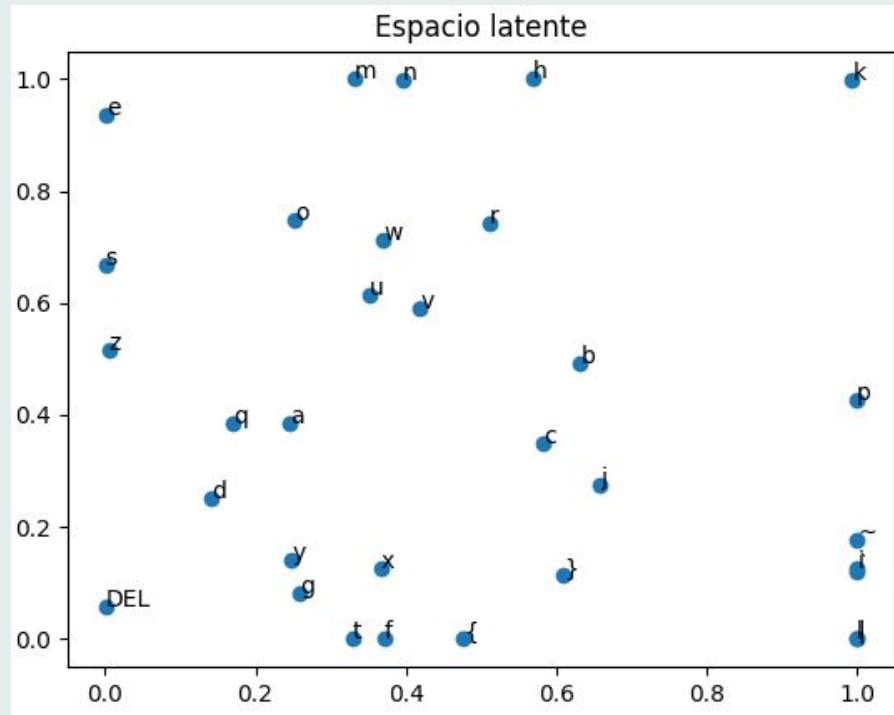


x x  
x x

$\eta=0.0001$   
Épocas: 20000

Arquitectura: 35-10-2-10-35

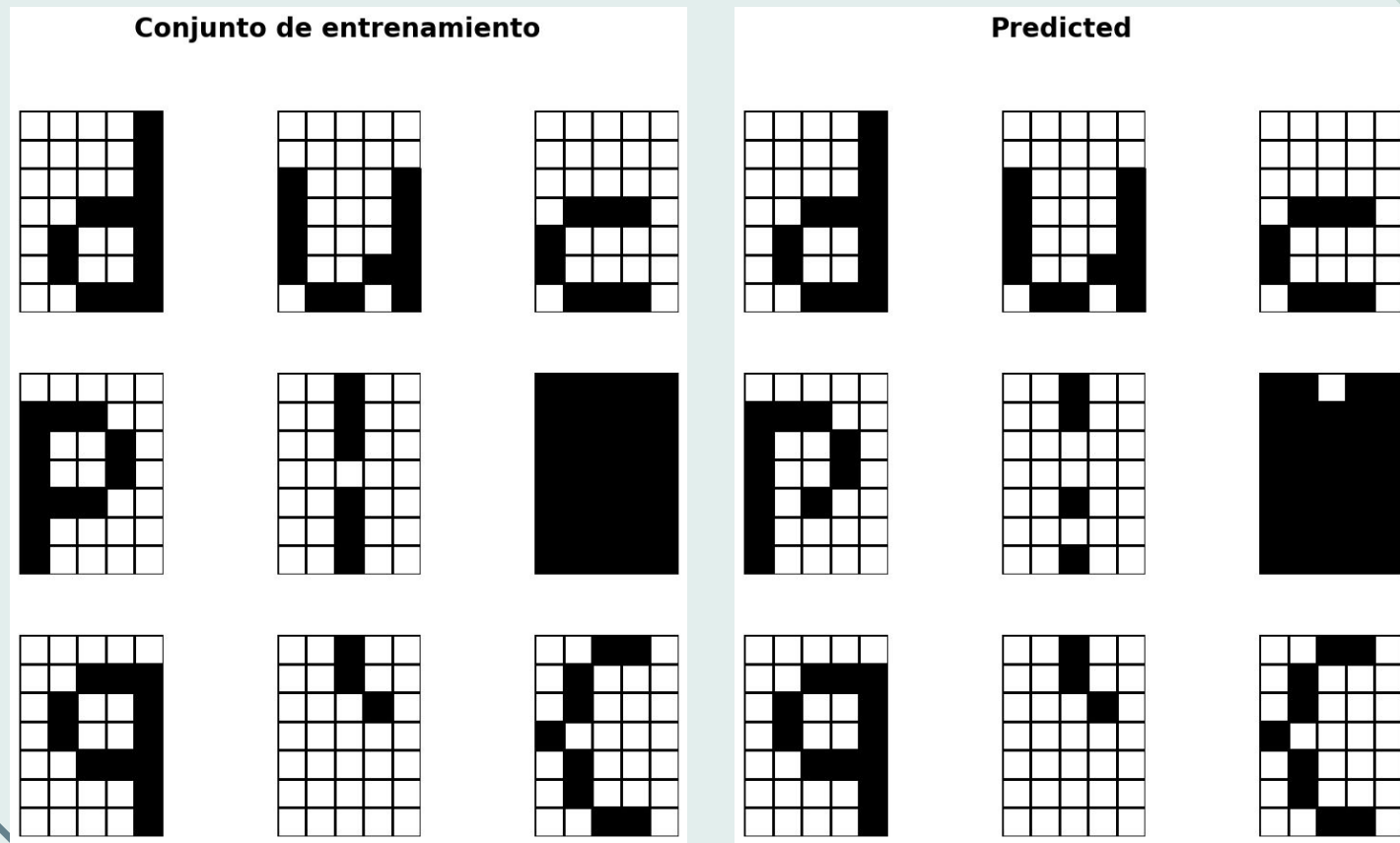
MSE: 2.63081



# Testeando con subconjuntos

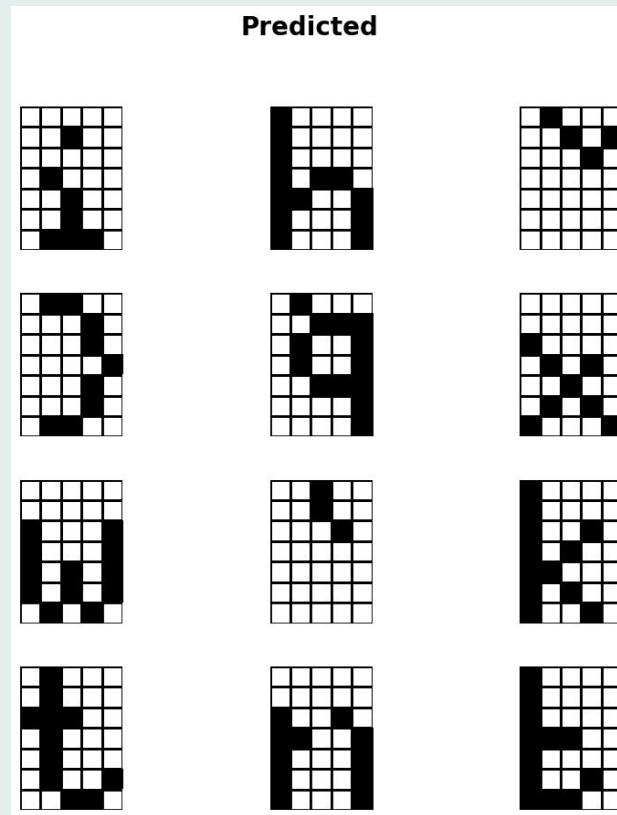
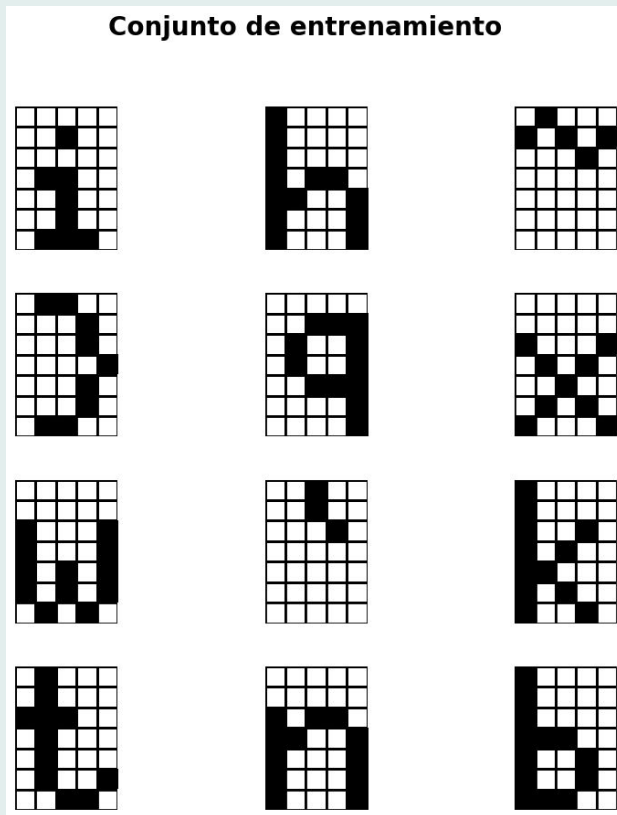


# Subconjunto del 30% de los datos



$\eta=0.001$  Épocas: 10000 Arquitectura: 35-15-2-15-35 MSE: 0.43291

# Subconjunto del 40% de los datos



× ×

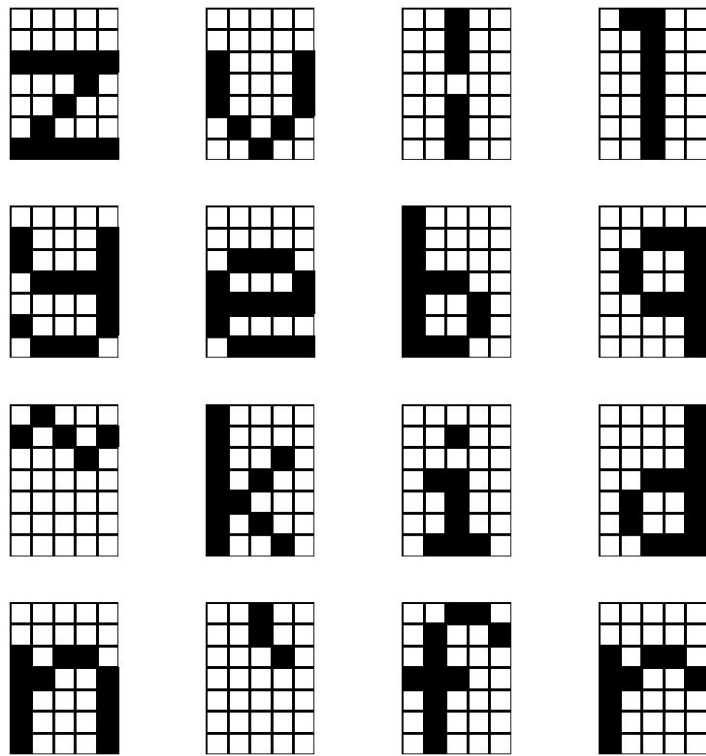
$\eta=0.001$  Épocas: 15000 Arquitectura: 35-15-2-15-35 MSE: 0.50001



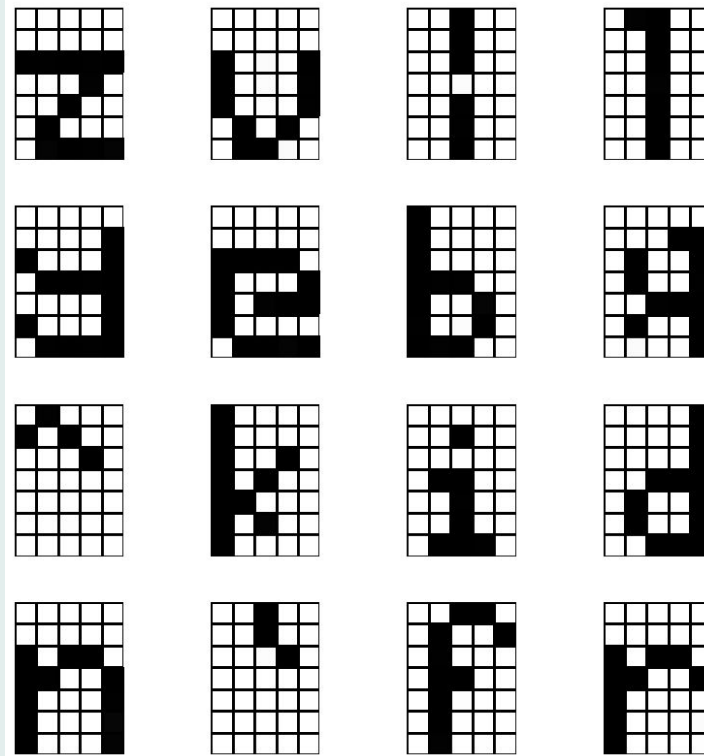


# Subconjunto del 50% de los datos

Conjunto de entrenamiento

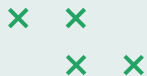
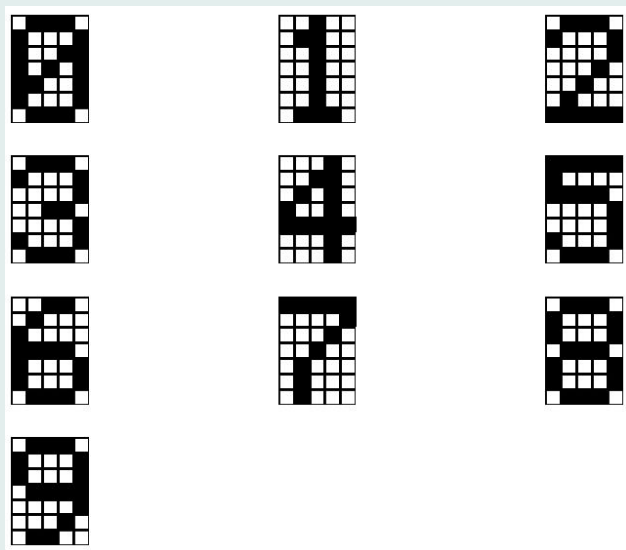


Predicted

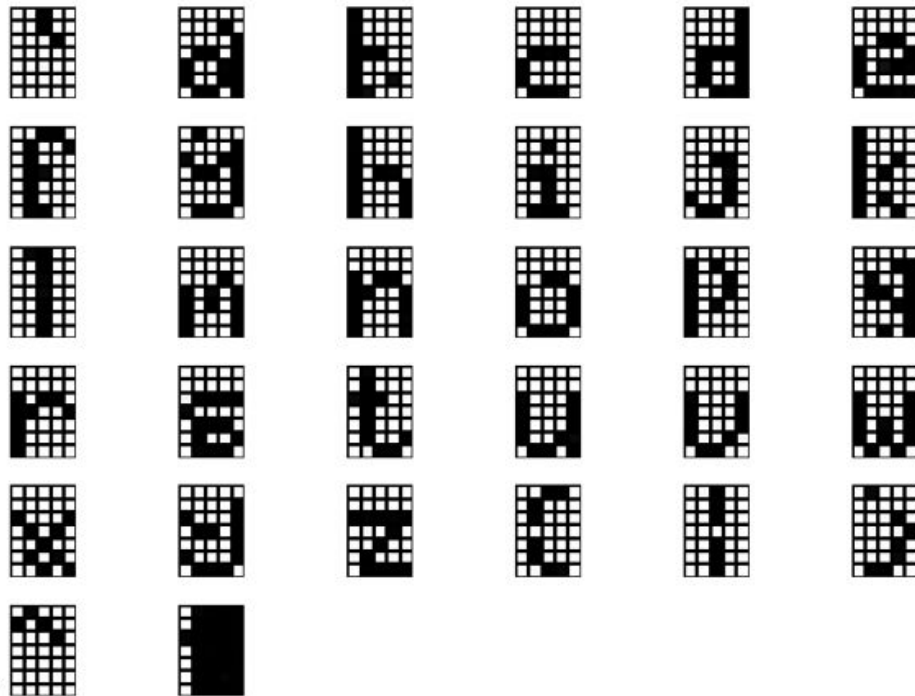


$\eta=0.001$  Épocas: 15000 Arquitectura: 35-15-2-15-35 MSE: 0.68778

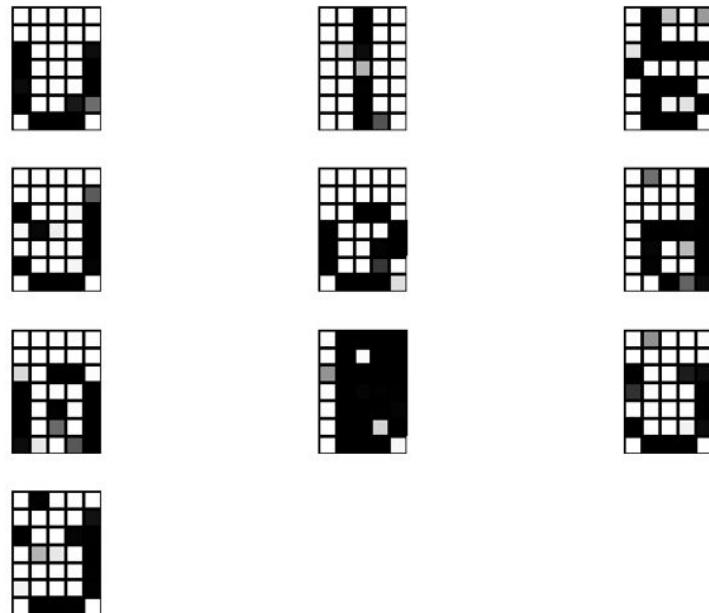
# Capacidad de generar nuevos caracteres



**Predicted**



**Predicted**



× ×  
× ×

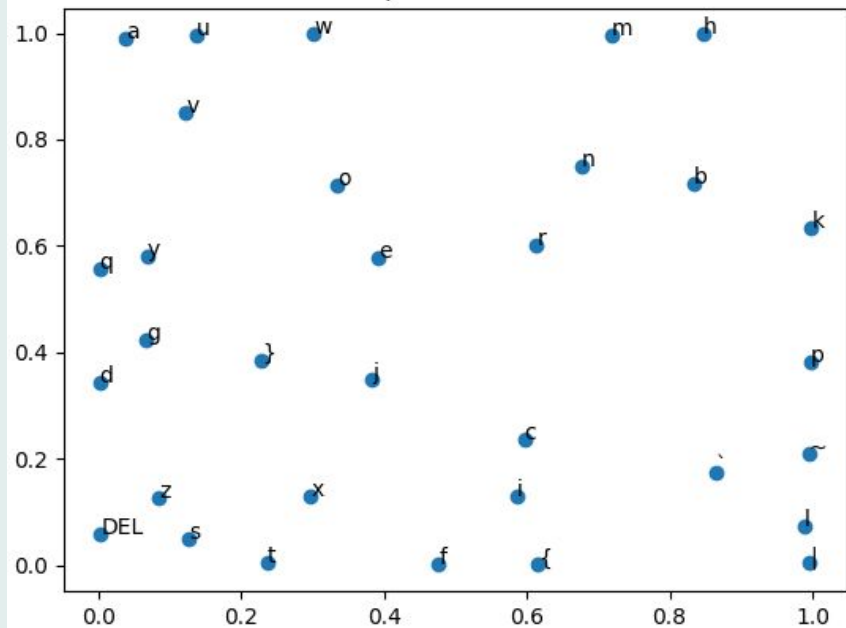
$\eta=0.001$

Épocas: 15000

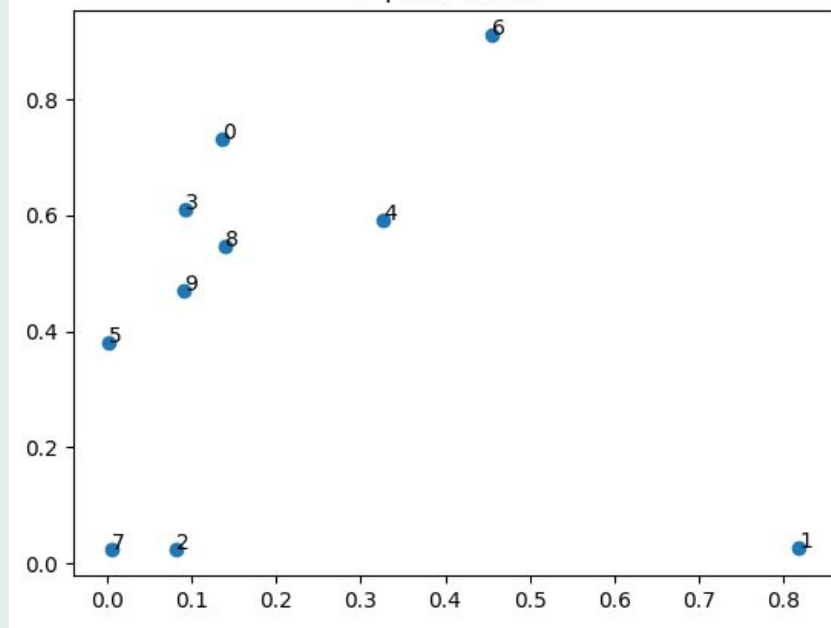
Arquitectura: 35-15-2-15-35

MSE: 1.34577

Espacio latente



Espacio latente



× ×  
× ×

# Conclusiones

- Cuanto más grande es el conjunto de entrenamiento, mayor es el error obtenido, es decir, será más difícil de aprender dicho conjunto por el autoencoder.
- Al analizar la capa latente, se puede observar que letras similares tienen codificaciones similares (lemma PCA).
- No existe un número de capas ideal.
- Tampoco existe un número de neuronas por capa ideal.
- Se realizaron diferentes pruebas para poder obtener el mejor resultado posible.
- El método de optimización más eficiente es ADAM.



The background features a light blue-grey color with abstract geometric shapes in a slightly darker shade of blue-grey. These shapes are connected by thin, dark blue lines that form a network-like pattern. In the top-left corner, there is a green plus sign. In the bottom-left corner, there are four red 'x' marks arranged in a small cluster. On the right side, there is a vertical red bar with diagonal white stripes.

02

# Denoising Autoencoder

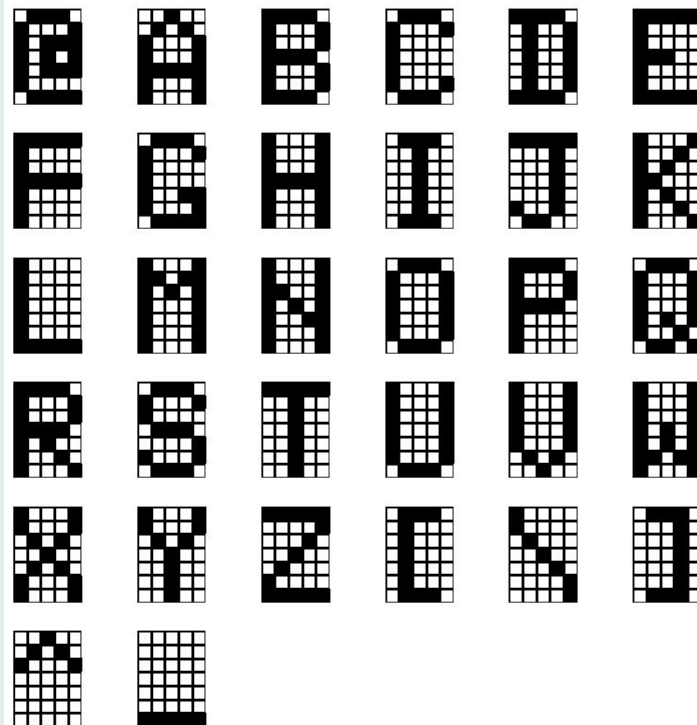
# Problema

Se busca estudiar la capacidad del autoencoder de eliminar ruido sobre los caracteres del conjunto de entrenamiento. Para este problema se decidió utilizar el siguiente conjunto de caracteres:

Para realizar esto se utiliza el concepto de un Denoising Autoencoder.

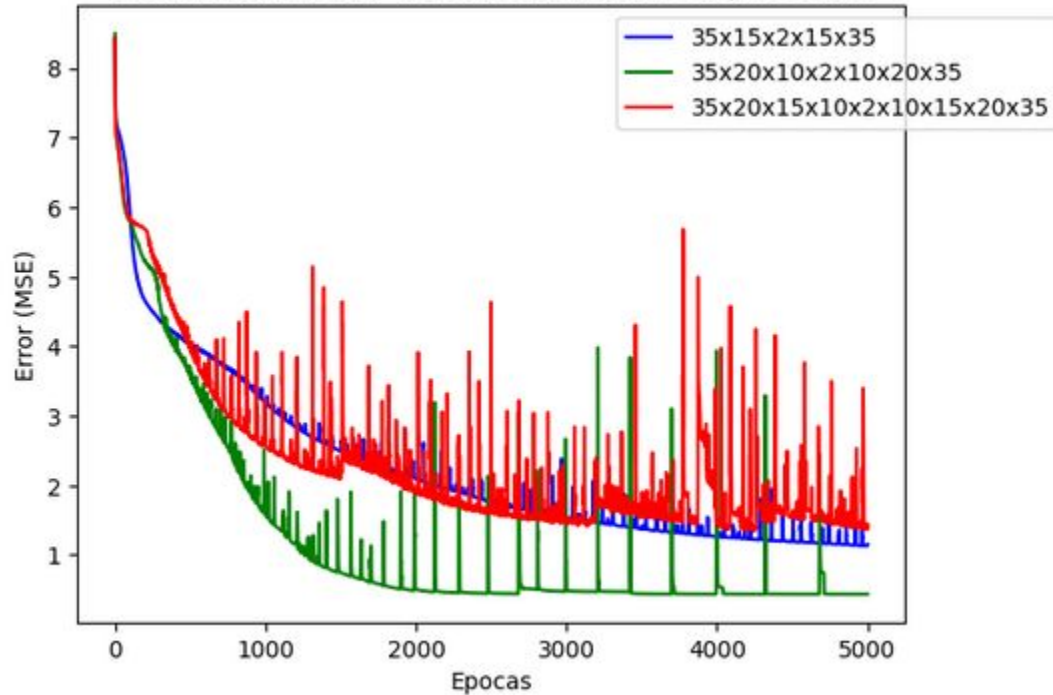
La mutación o ruido aplicado consiste en elegir aleatoriamente N caracteres a ser mutados, donde se varía el valor de cada píxel por un número delta con probabilidad uniforme entre (0,0.5).

Conjunto de entrenamiento



# Arquitecturas

Evaluación del error de DAE con distintas arquitecturas



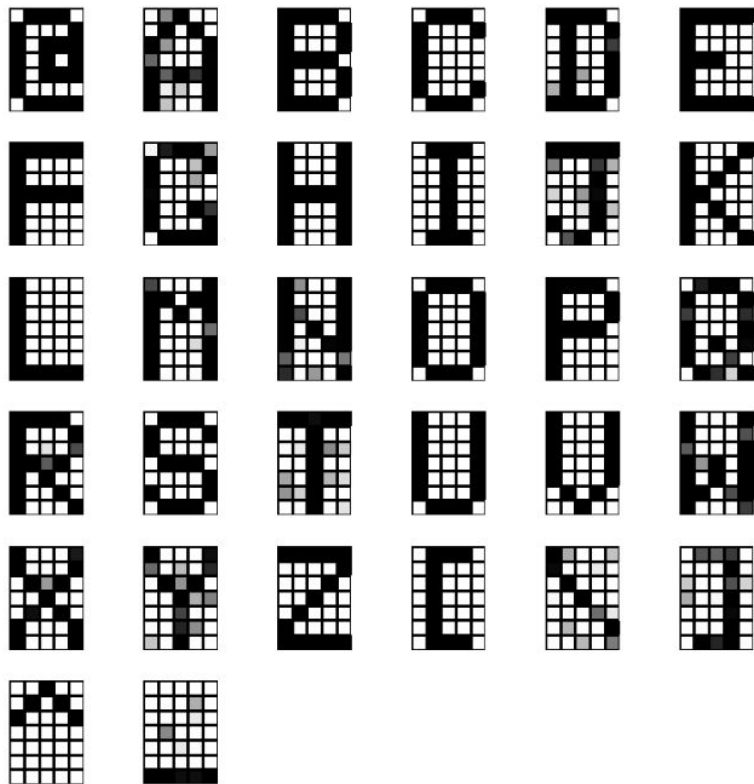


# Variando la probabilidad de mutación

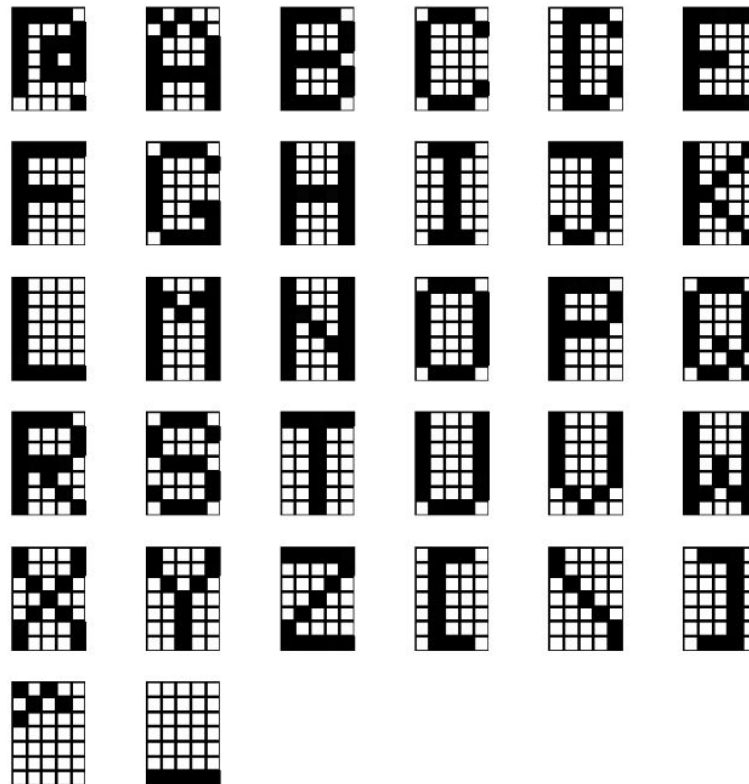


# Probabilidad 0.2 de mutar cada pixel

Conjunto de entrenamiento con 15 caracteres mutados

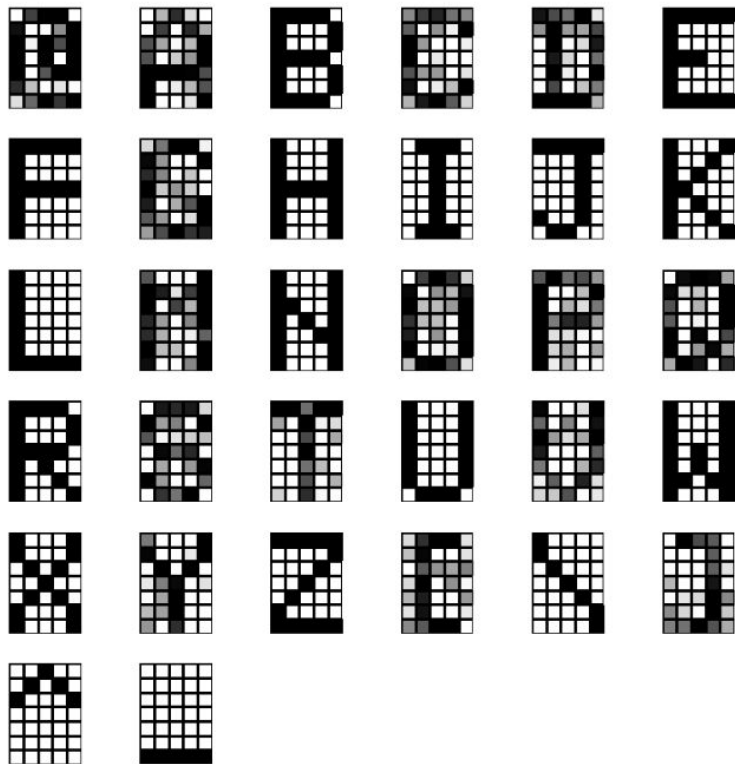


Eliminacion del ruido

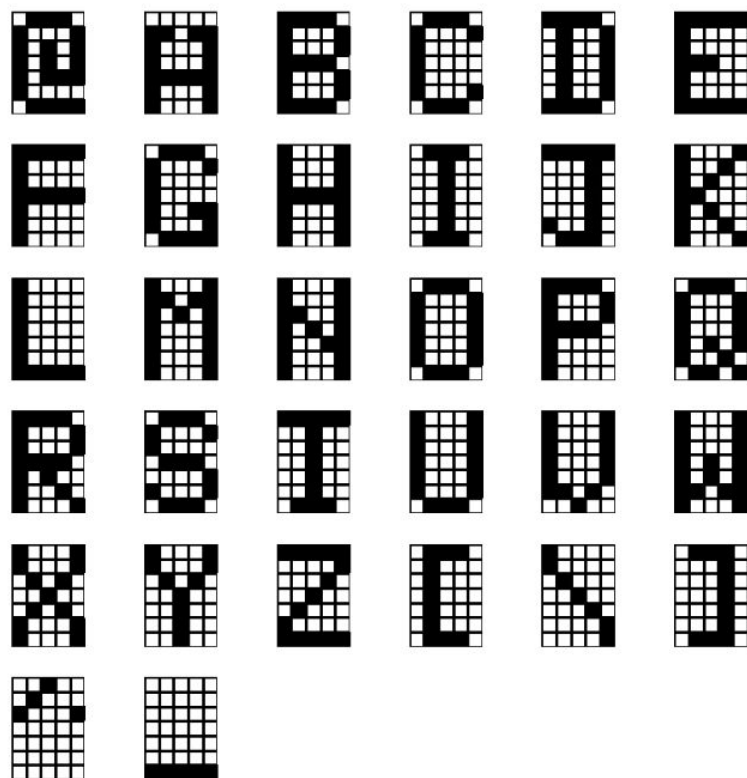


# Probabilidad 0.5 de mutar cada pixel

Conjunto de entrenamiento con 15 caracteres mutados

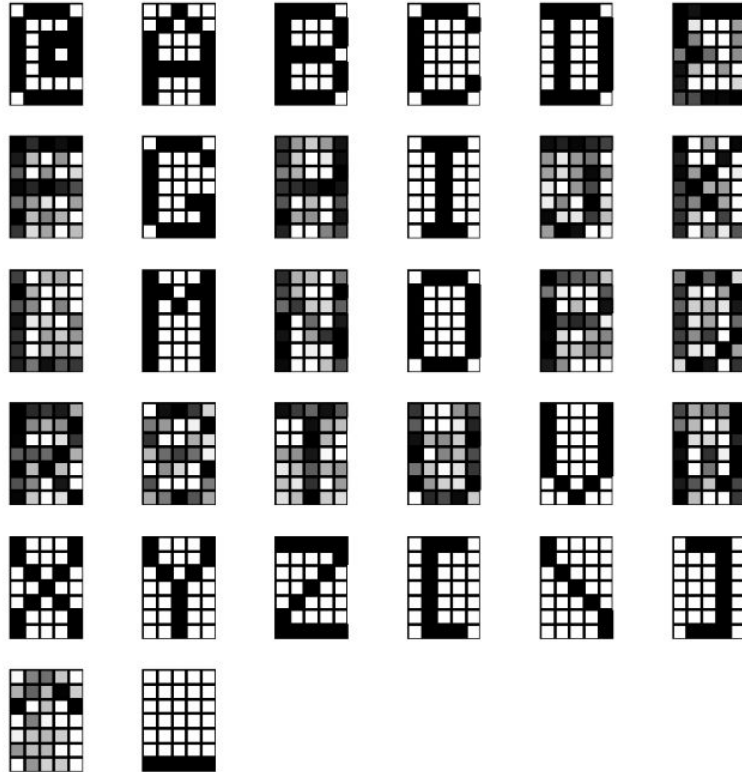


Eliminacion del ruido

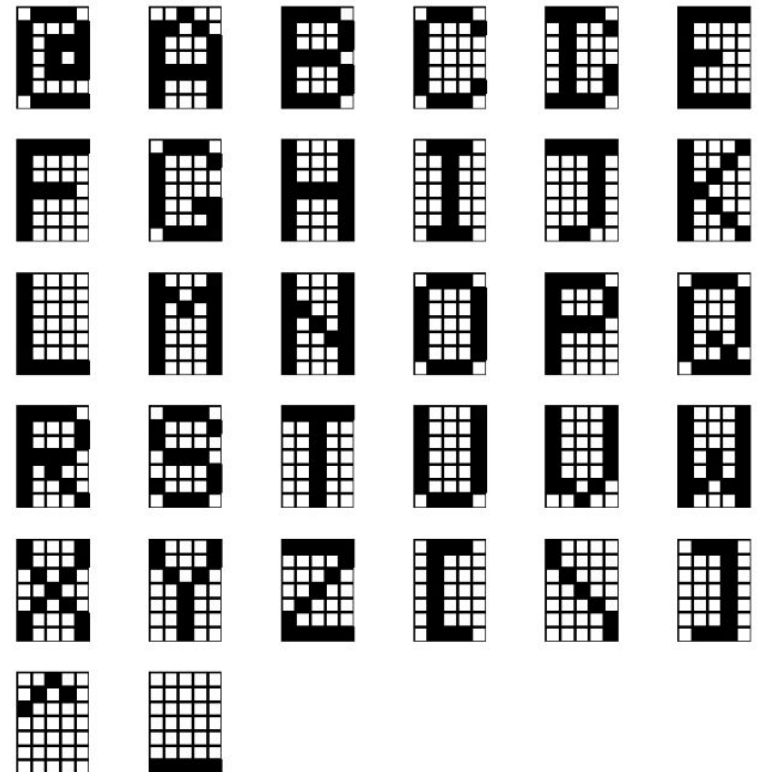


# Probabilidad 0.7 de mutar cada pixel

Conjunto de entrenamiento con 15 caracteres mutados

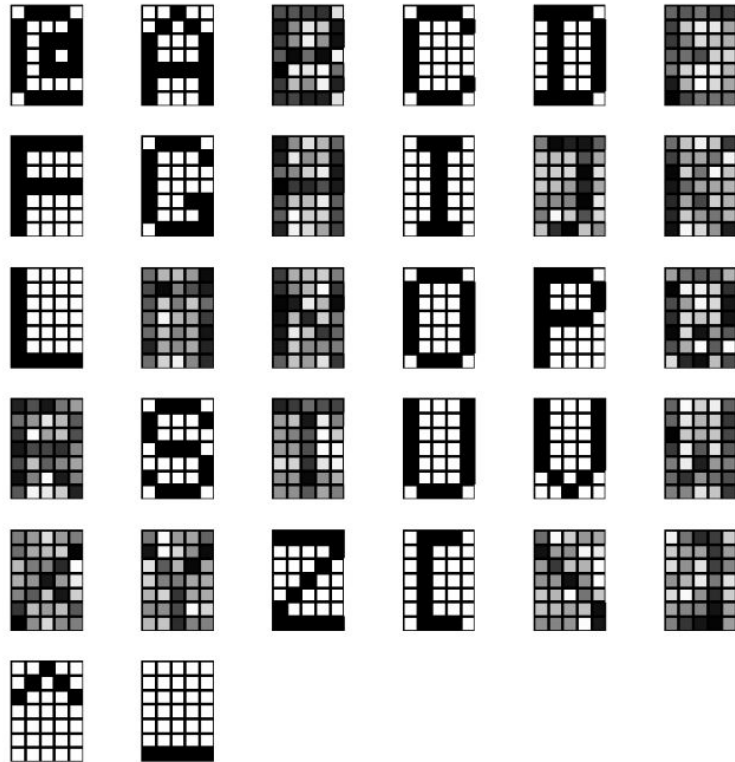


Eliminacion del ruido

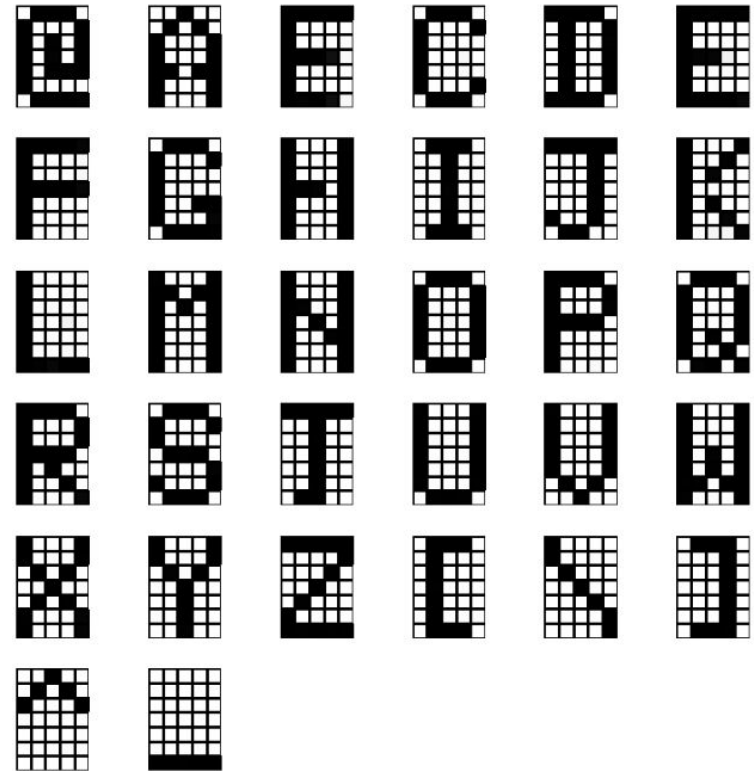


# Probabilidad 1 de mutar cada pixel

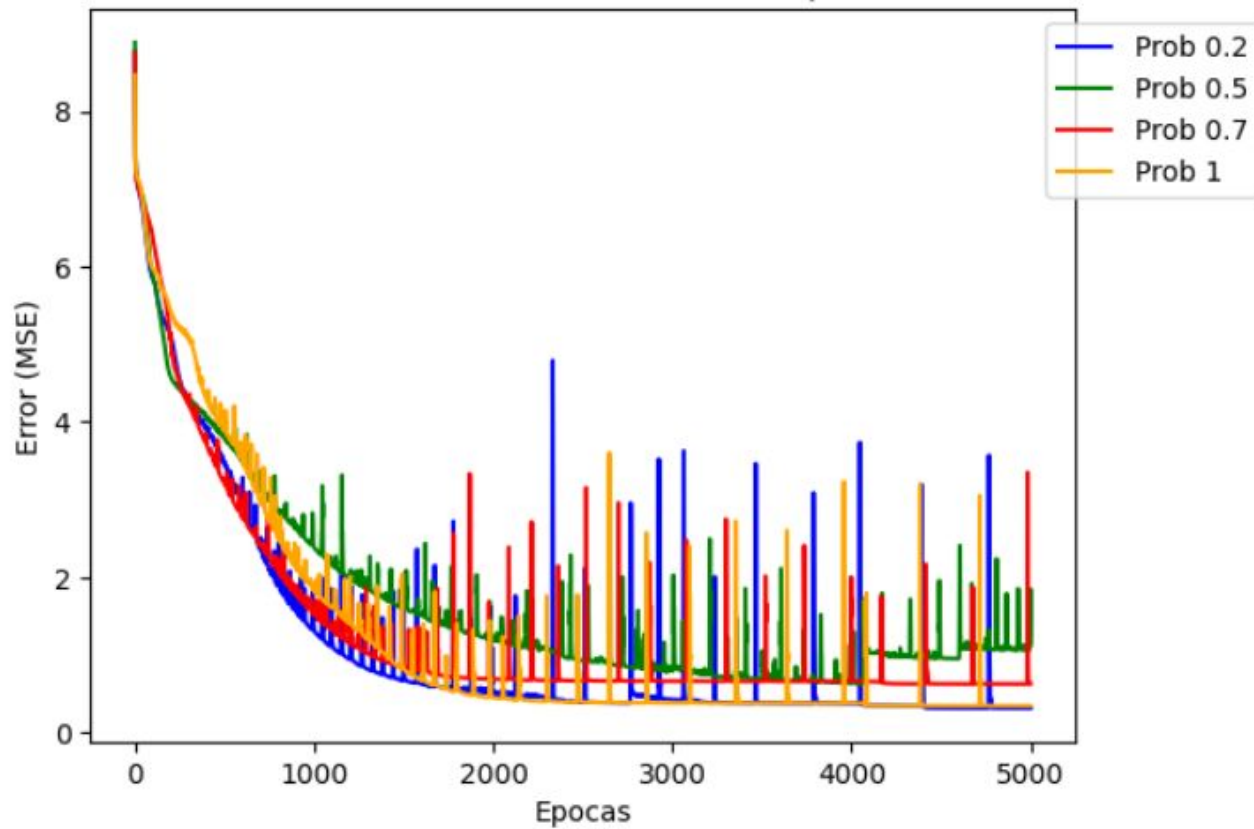
Conjunto de entrenamiento con 15 caracteres mutados



Eliminacion del ruido

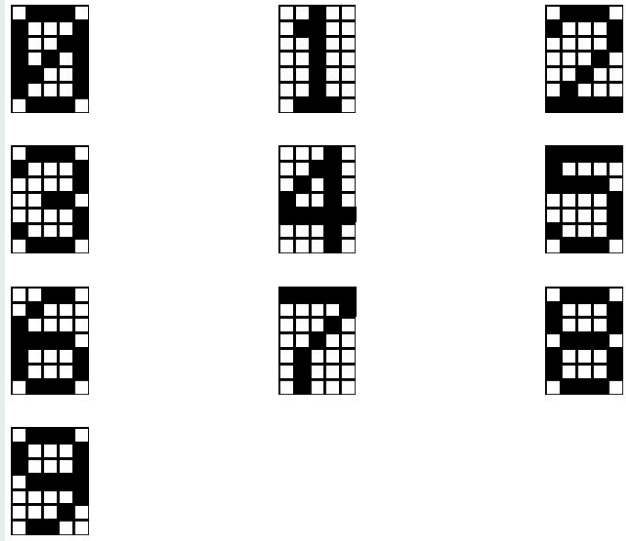


Evaluacion del error de DAE con distintas probabilidades



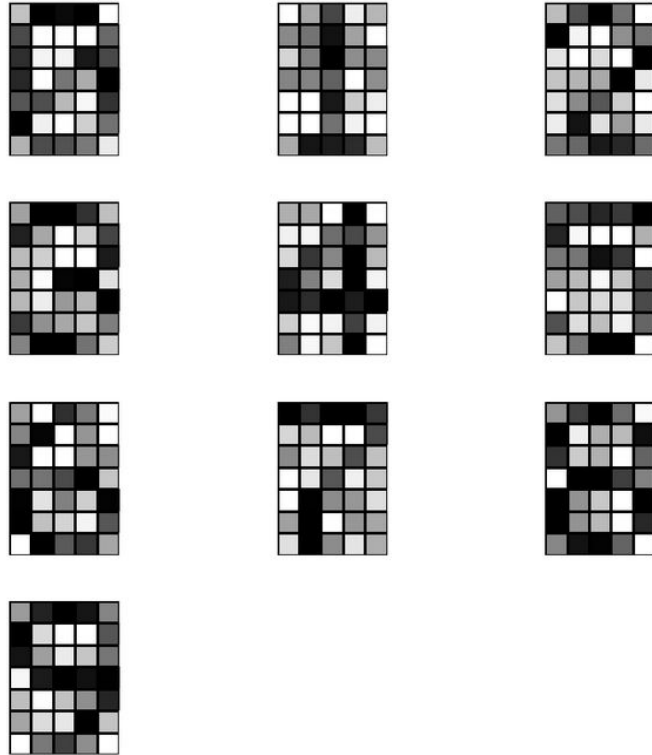
x x

# Variando la probabilidad de mutación con un nuevo conjunto de datos

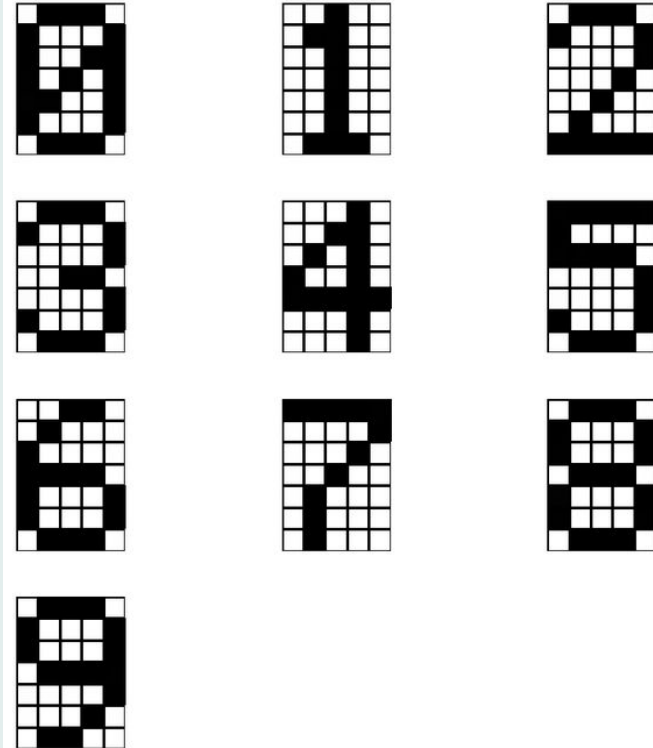


# Probabilidad 0.7 de mutar cada pixel

Conjunto de entrenamiento



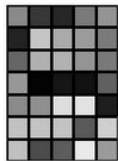
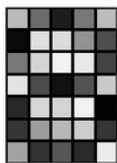
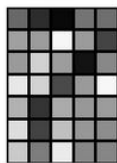
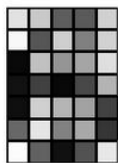
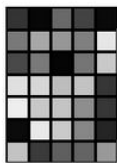
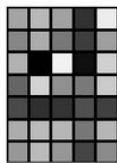
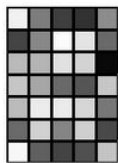
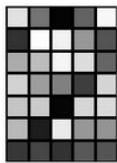
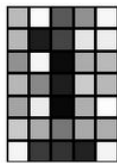
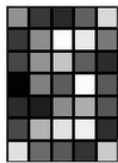
Eliminacion del ruido



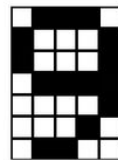
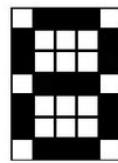
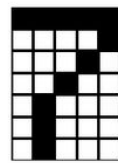
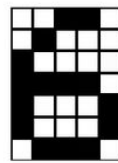
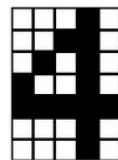
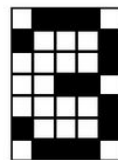
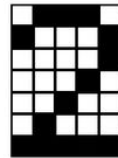
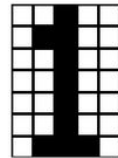


# Probabilidad 1 de mutar cada pixel

Conjunto de entrenamiento



Eliminacion del ruido



# Conclusiones

- El DAE es capaz de asociar el valor ruidoso al valor original en la gran mayoría de los casos.
- A mayor cantidad de datos del conjunto de entrenamiento, el DAE tiene mayores dificultades para eliminar el ruido.
- A mayor cantidad de capas, se asocian mejor las letras ruidosas. Pero además el tiempo de aprendizaje del DAE aumenta considerablemente.
- A mayor ruido, más le cuesta asociar las entradas a la red.



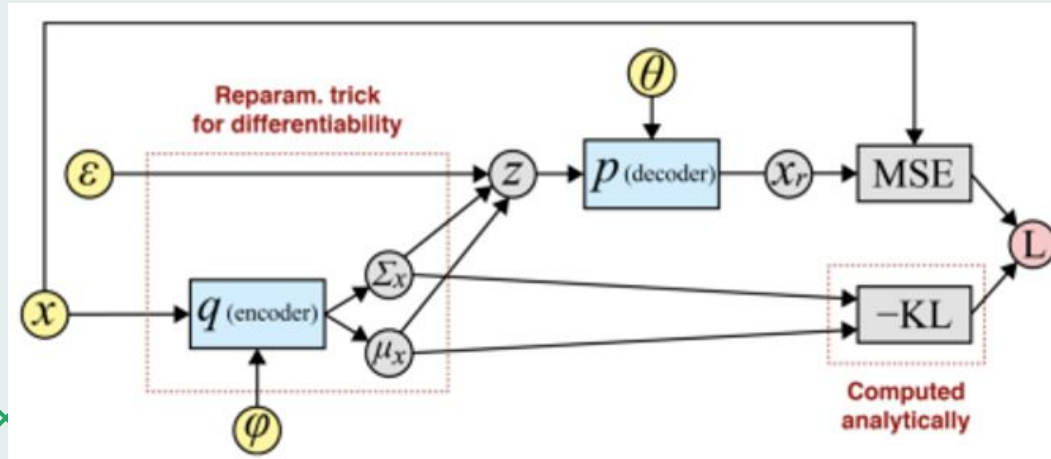
The background features a light blue-grey color with abstract geometric patterns. On the left, there are green and dark blue shapes with a green plus sign and a blue line with a circle. On the right, there are green and dark blue shapes with a red and white striped vertical bar. At the bottom left, there are four red 'x' marks.

03

# Variational Autoencoder

# Variational Autoencoder

Un autoencoder variacional provee una forma probabilística de describir una observación en el espacio latente. En vez de construir un autoencoder que emita una solo valor para describir cada atributo del estado latente, el VAE describe una distribución probabilística para cada atributo del estado latente. De esta forma, nos podemos mover dentro de un vector de representación para generar nuevas muestras a la salida del decodificador.



# Diseño del VAE basado en Keras

- Las capas convolucionales usan un kernel de 3x3 con stride 2 y padding SAME.
- Las activaciones intermedias se realizan con ReLU y la de la capa de salida del decoder con sigmoidea.
- Se analizan en cada época las métricas de:
  - Reconstruction\_loss: diferencia entre  $X$  y  $X'$ .
  - K<sub>L</sub>\_loss: pérdida en base a la convergencia KL, la cual busca minimizar la diferencia entre las funciones de distribución.
  - Total\_loss: suma de los dos anteriores.

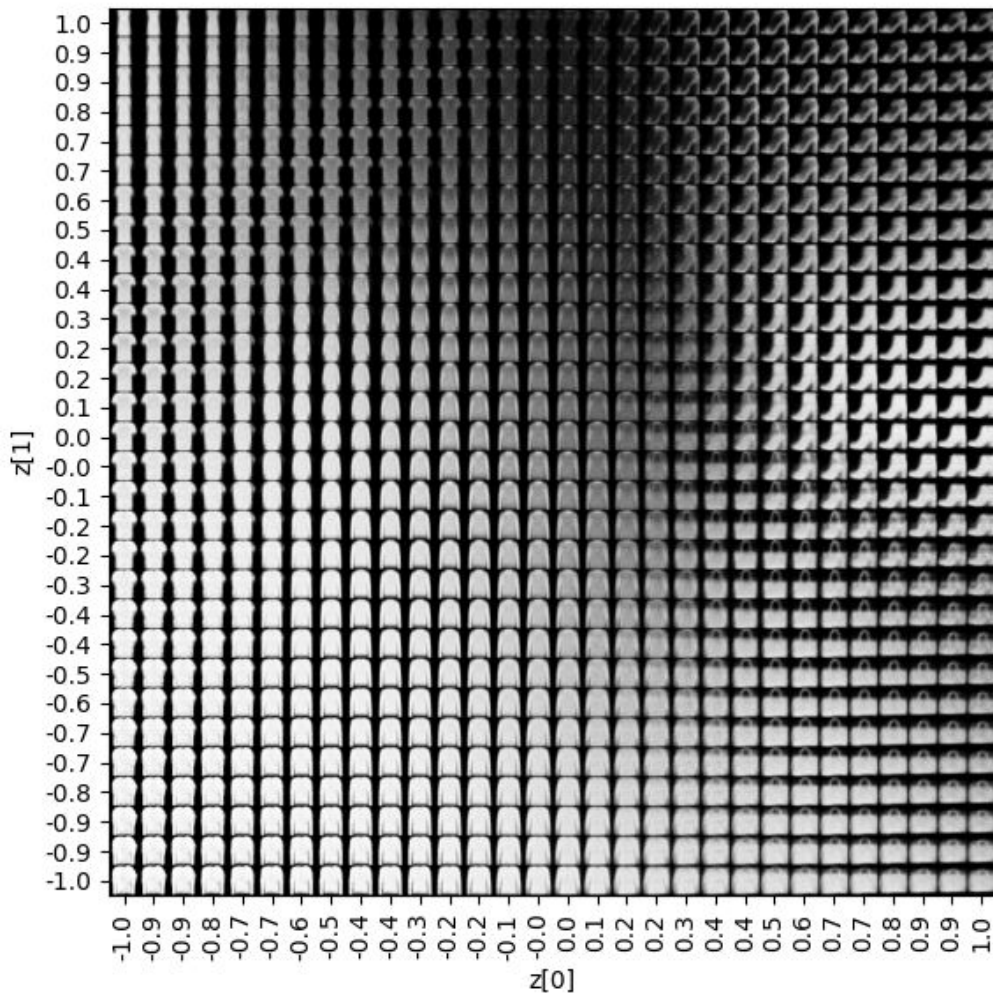
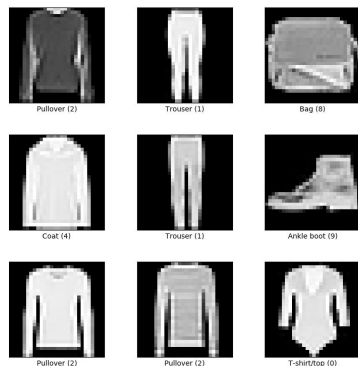
# Fashion MNIST

Conjunto de datos de Keras

Set de 60,000 muestras para entrenamiento

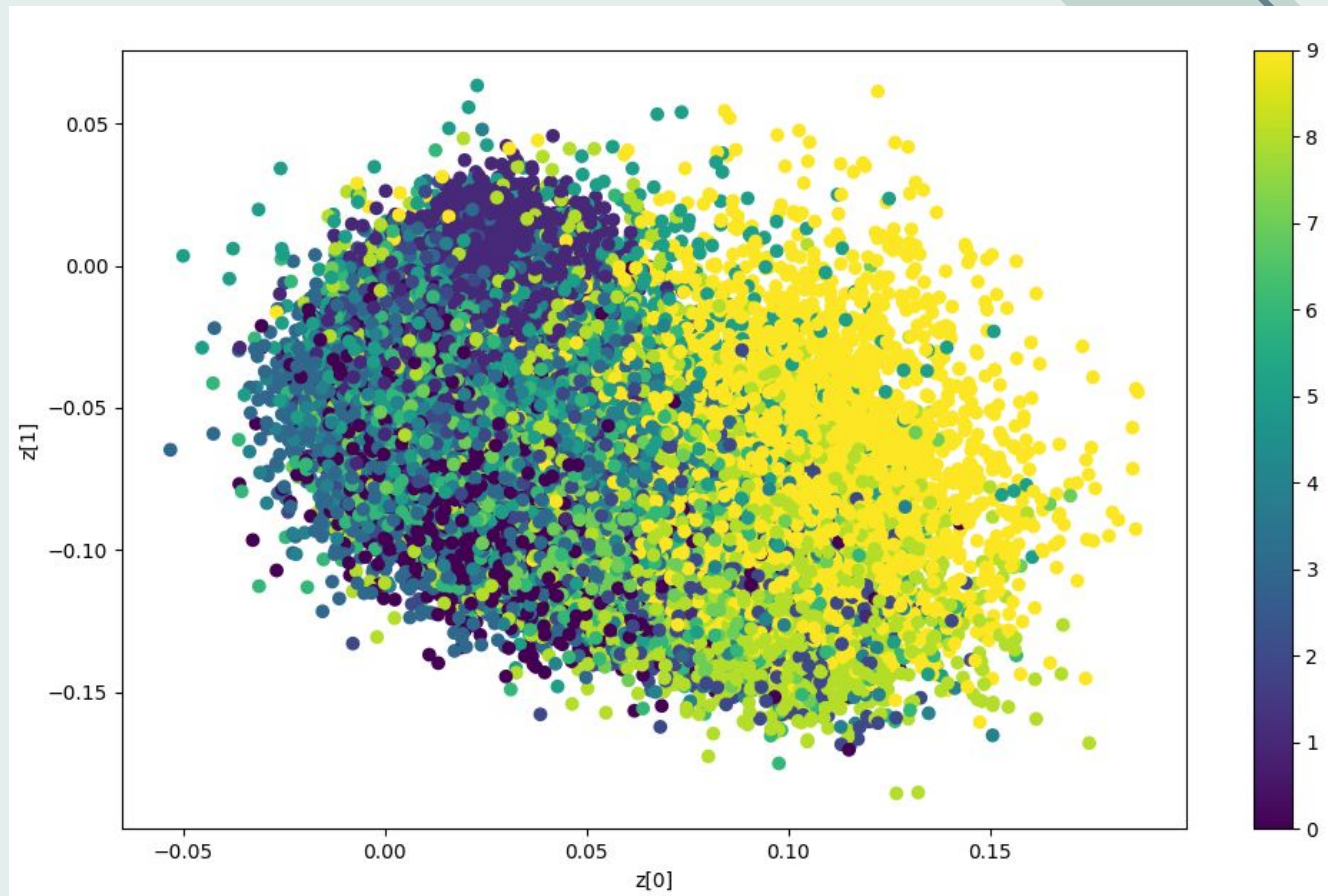
Set de 10,000 muestras para testeo

El VAE nos permite explorar salidas del autoencoder que no necesariamente tienen una entrada asociada, lo que con un buen conjunto de datos permite generar nueva data.



Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

✕ ✕  
 ✕ ✕



¡Gracias!

