



Sistemas de inteligencia artificial

TP5: Deep Learning

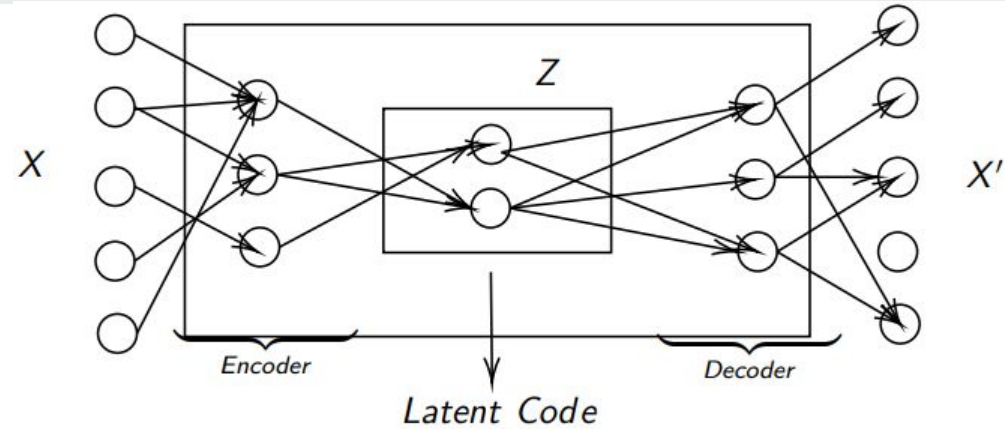
Grupo 19

Integrantes:

- Lucas Catolino
- Matias Ricarte

Ejercicio 1.a

Autoencoder

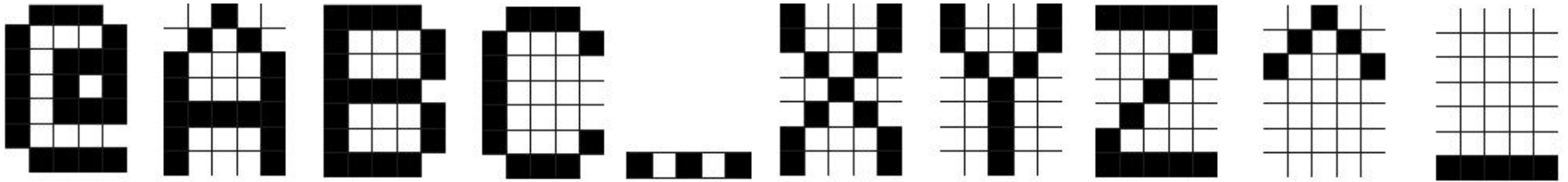


- Dos perceptrones multicapa: la salida de uno es la entrada del otro
- Arquitecturas paralelas
- Aprendizaje: función identidad $\rightarrow X = X'$
- En Z se encuentran las proyecciones en los componentes principales



Autoencoder

El problema: dado un archivo de fuentes (imágenes binarias), estudiar arquitecturas de autoencoders y tácticas de optimización, graficar el espacio latente y generar nuevas letras



Autoencoder

Optimizador: se estudiaron los resultados con y sin momentum. El momentum introduce una ponderación en las direcciones de descenso calculadas en pasos anteriores. Agrega una noción de “inercia” para suavizar el camino zigzagueante del gradiente descendiente y acelerando el descenso en direcciones similares a las anteriores



Método del gradiente
descendiente





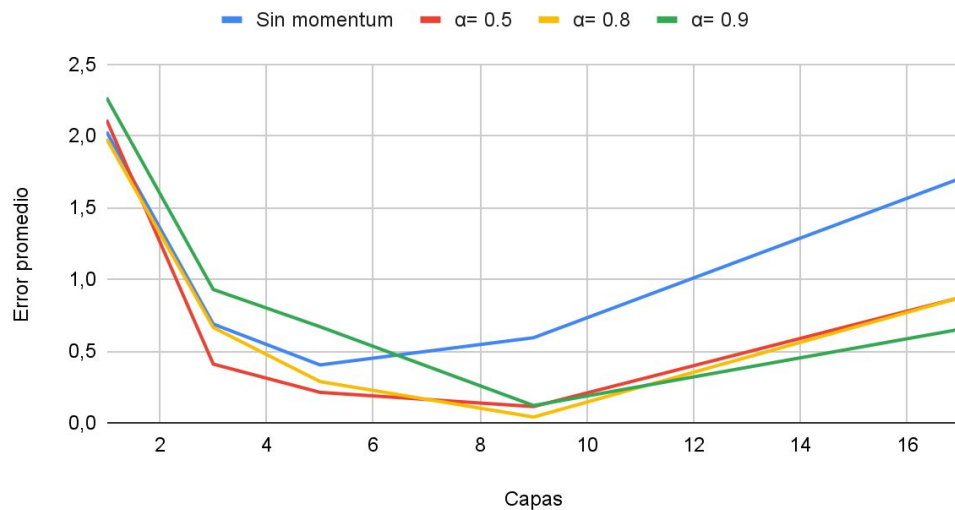
Experimentación

Se probaron distintas arquitecturas con y sin momentum

	Capas	Sin momentum	$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.9$
[2]	1	2,029090678	2,112012327	1,979408549	2,266884278
[20,2,20]	3	0,6880308877	0,4090950168	0,6640879464	0,9291257471
[25,10,2,10,25]	5	0,4032735513	0,212031022	0,287066664	0,669640356
[25,20,15,10,2,10,15,20,25]	9	0,5930309474	0,1129948061	0,039992508	0,1203184404
[31,27,23,19,15,11,7,3,2,3,7,11,15,19,23,27,31]	17	1,704542485	0,8735941077	0,8735941077	0,6525797032

Experimentación

Error promedio por capas y optimización



La mejor combinación se dió
en 9 capas con $\alpha = 0.8$

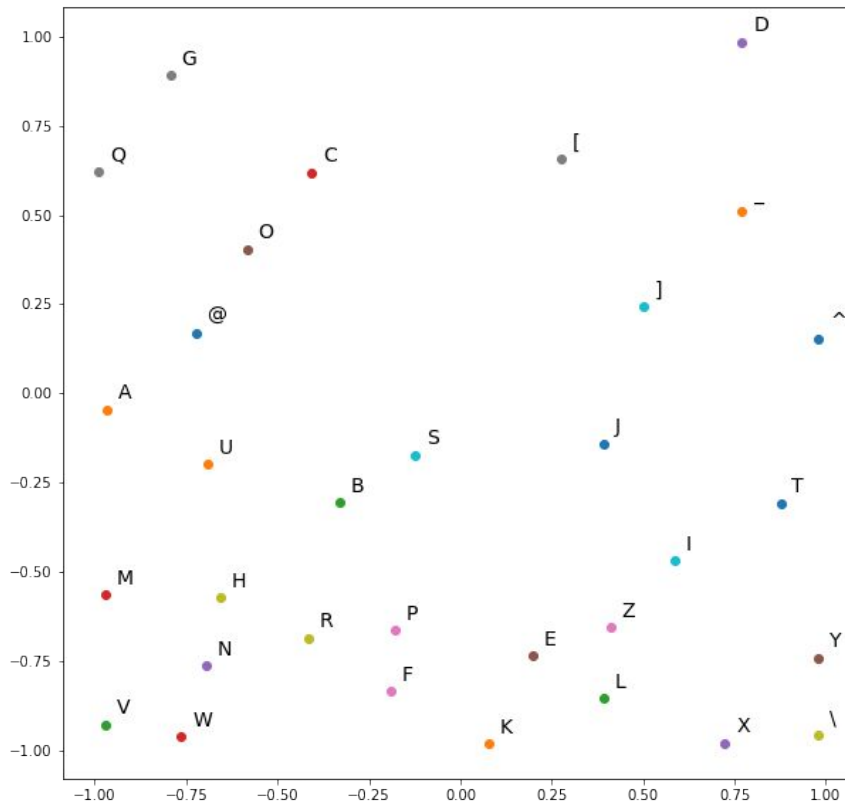
[25,20,15,10,2,10,15,20,25]

Resultados

Espacio latente

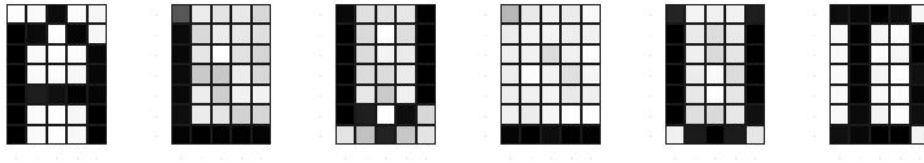
epochs	10000
momentum	0,8
beta	1
learning rate	0,01
function	$\tanh(x)$
hiddenLayers	[25, 20, 15, 10, 2, 10, 15, 20, 25]

La red logró aprender todo el conjunto de entrada



Resultados

Decodificación a partir de los valores en el espacio latente del conjunto de datos original



La red logró aprender todo el set de datos con mínimos errores

epochs	10000
momentum	0,8
beta	1
learning rate	0,01
function	tanh(x)
hiddenLayers	[25, 20, 15, 10, 2, 10, 15, 20, 25]

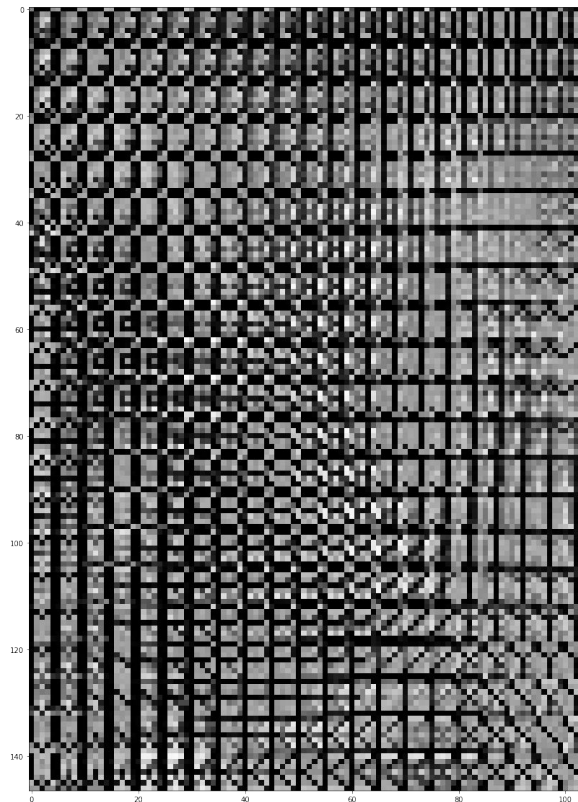
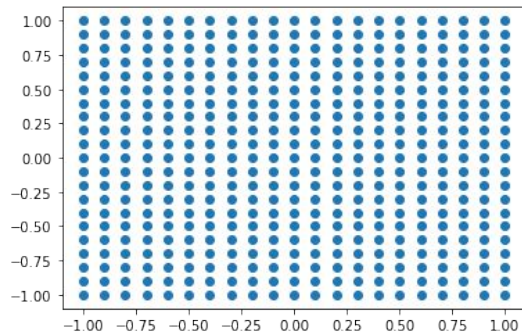


Resultados

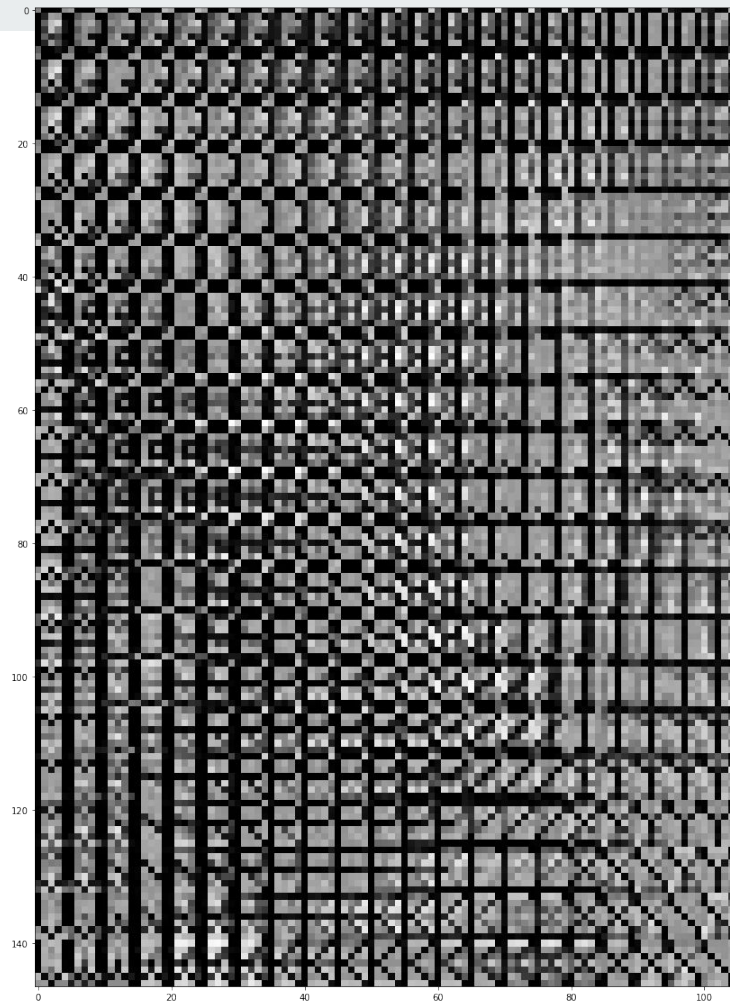
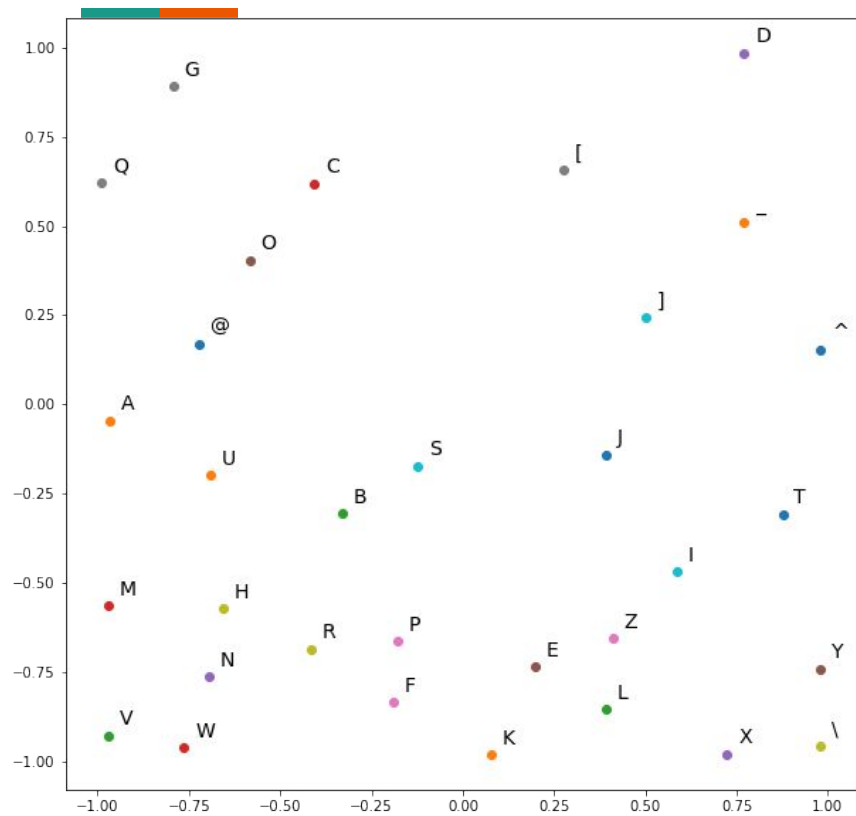
epochs	10000
momentum	0,8
beta	1
learning rate	0,01
function	$\tanh(x)$
hiddenLayers	[25, 20, 15, 10, 2, 10, 15, 20, 25]

Decodificación a partir de los valores en el espacio latente elegidos de manera uniforme

Se usó la red para generar nuevas muestras que no pertenecieran al conjunto de entrenamiento



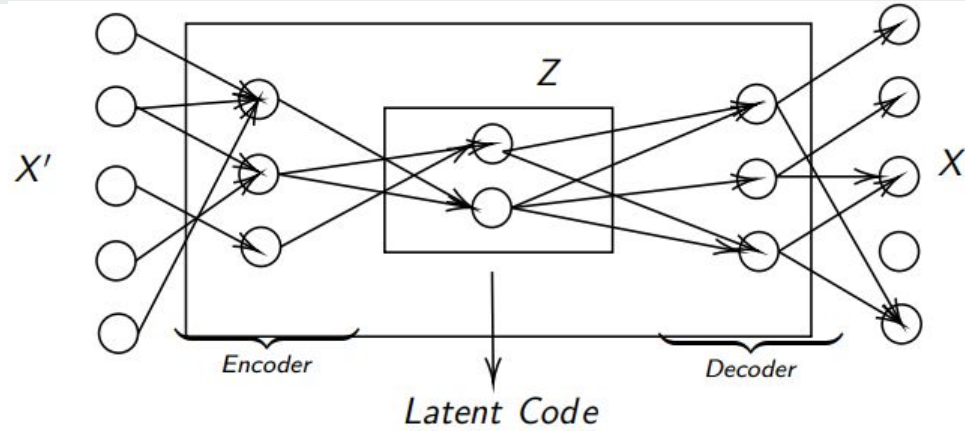
Resultados



Ejercicio 1.b

Denoising Autoencoder

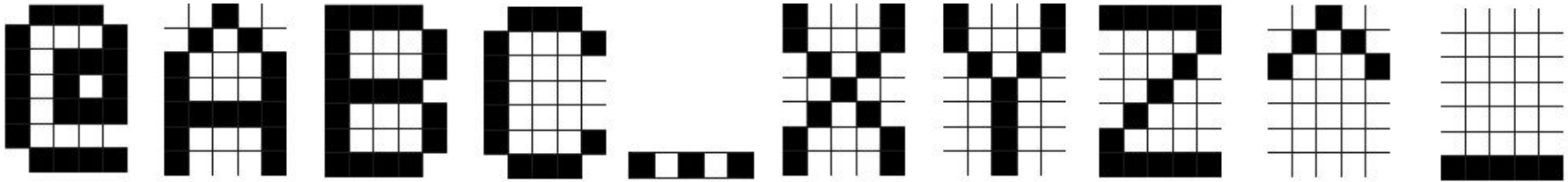
- Se entrena con una entrada ruidosa, comparando con la entrada sin ruido
- Se simula ruido al conjunto de entrada
- Busca preservar información más allá del ruido





Denoising Autoencoder

El problema: dado un archivo de fuentes (imágenes binarias), implementar una arquitectura conveniente, distorsionar la entrada y estudiar la capacidad de eliminación del ruido

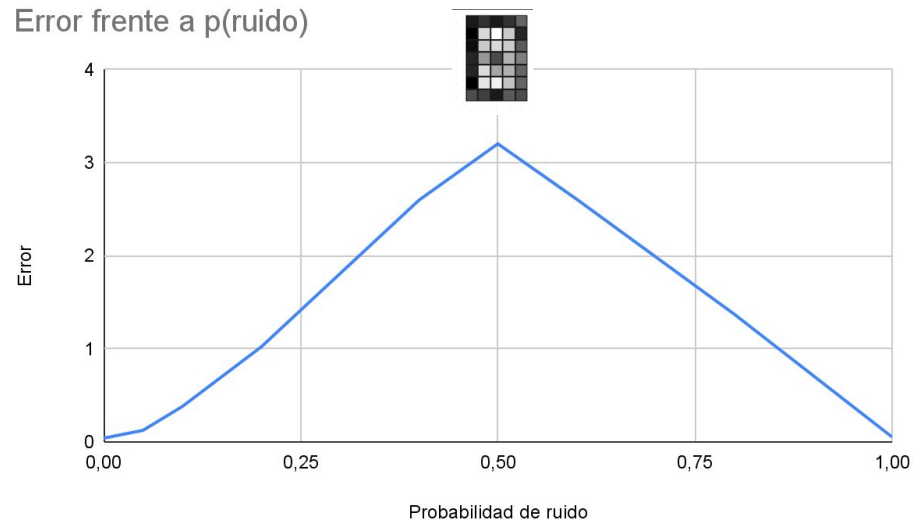


Resultados

epochs	10000
momentum	0,8
beta	1
learning rate	0,01
function	tanh(x)
hiddenLayers	[25, 20, 15, 10, 2, 10, 15, 20, 25]

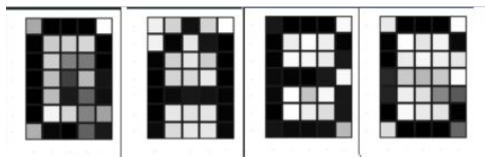
Ruido	Error
0	0,039992508
0,05	0,1241927836
0,1	0,3815621126
0,2	1,021471619
0,4	2,597748449
0,5	3,202972687
0,6	2,607369799
0,8	1,371351545
1	0,05282446101

Error frente a p(ruido)

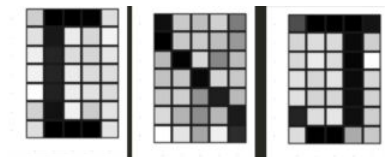


Resultados

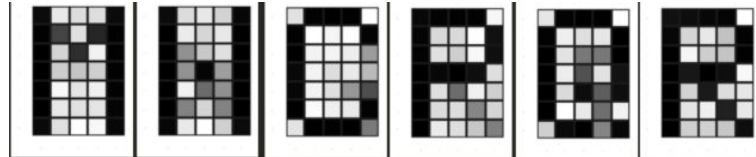
@, a, b, c



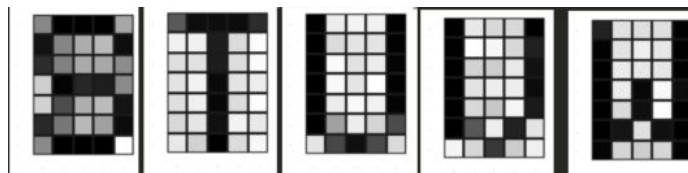
[, \,]



M, N, O, P, Q, R



s, t, u, v, w

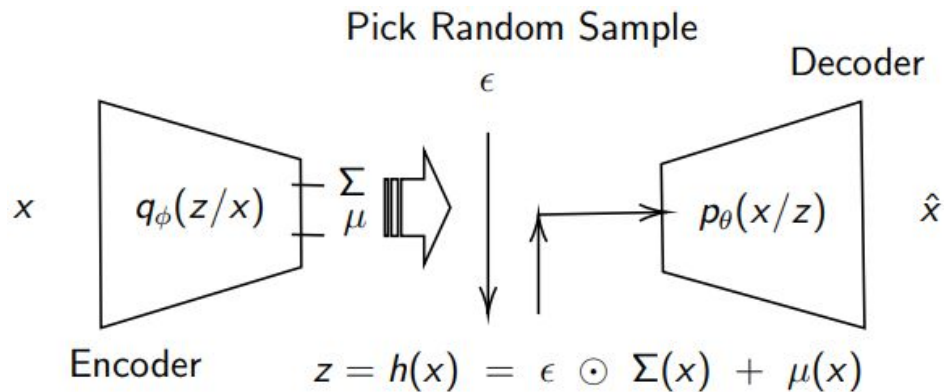


Error: 0.05

Ejercicio 2

Autoencoder Variacional Simple

- El espacio latente se representa como una distribución (generalmente Gaussiana)
- Se basa en un estimador Bayesiano: cuál es la probabilidad de un suceso dada una ocurrencia anterior
- Se aprenden los parámetros para definir la distribución: media y desvío estándar
- Se sampean datos de la “distribución latente”, se los alimenta al decoder y se obtienen datos que parecen ser del set original



Autoencoder Variacional Simple

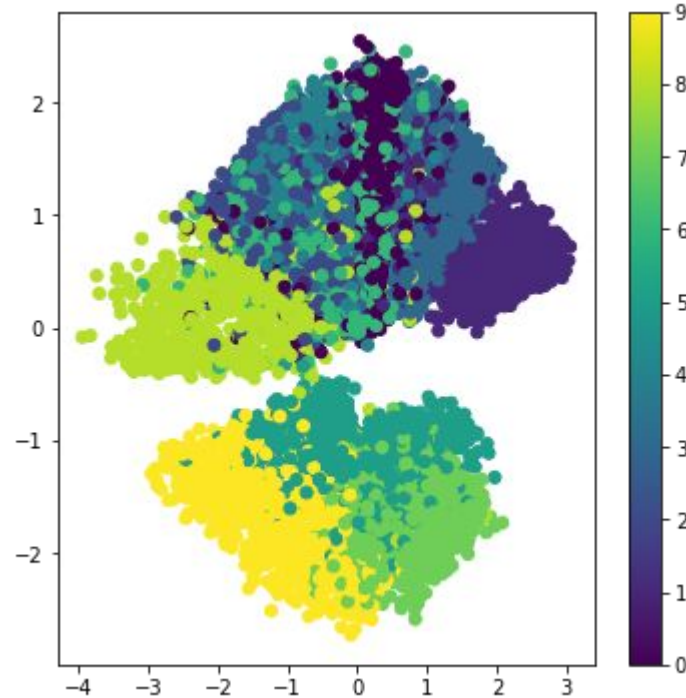
El problema: dada la capacidad generativa de la red, elegir un conjunto de datos y generar una nueva muestra

- Data set: fashion mnist
- Librerías: Keras, Tensorflow



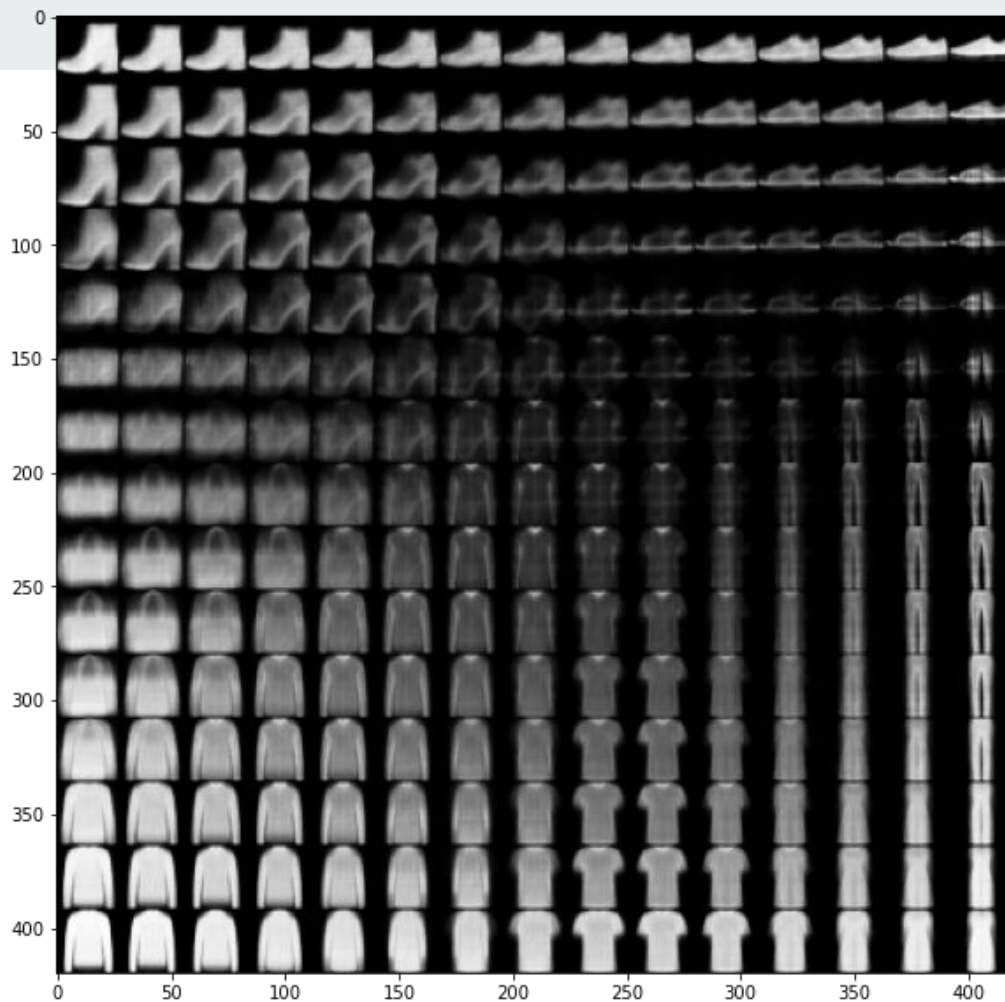
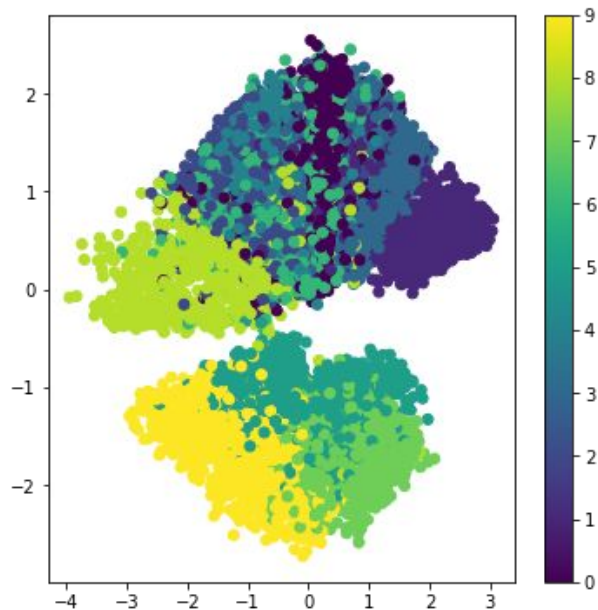
Resultados

- Epochs: 50
- Dimensiones intermedias: 256
- Espacio latente: 2
- Training set: 60000
- Testing set: 10000

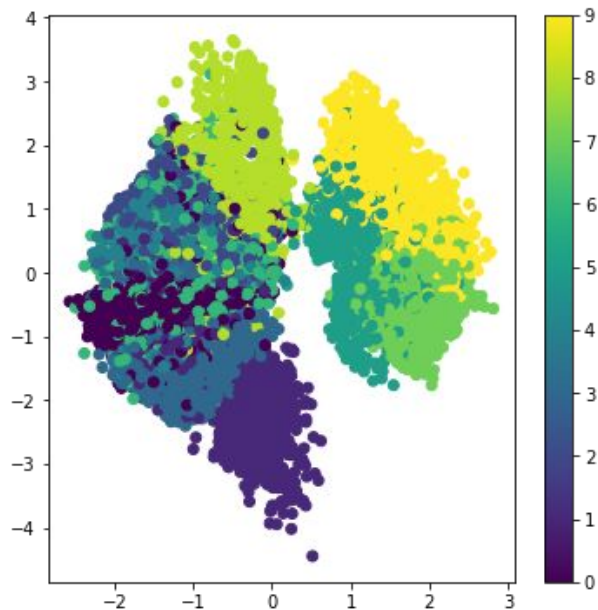


Label	0	1	2	3	4	5	6	7	8	9
Description	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot

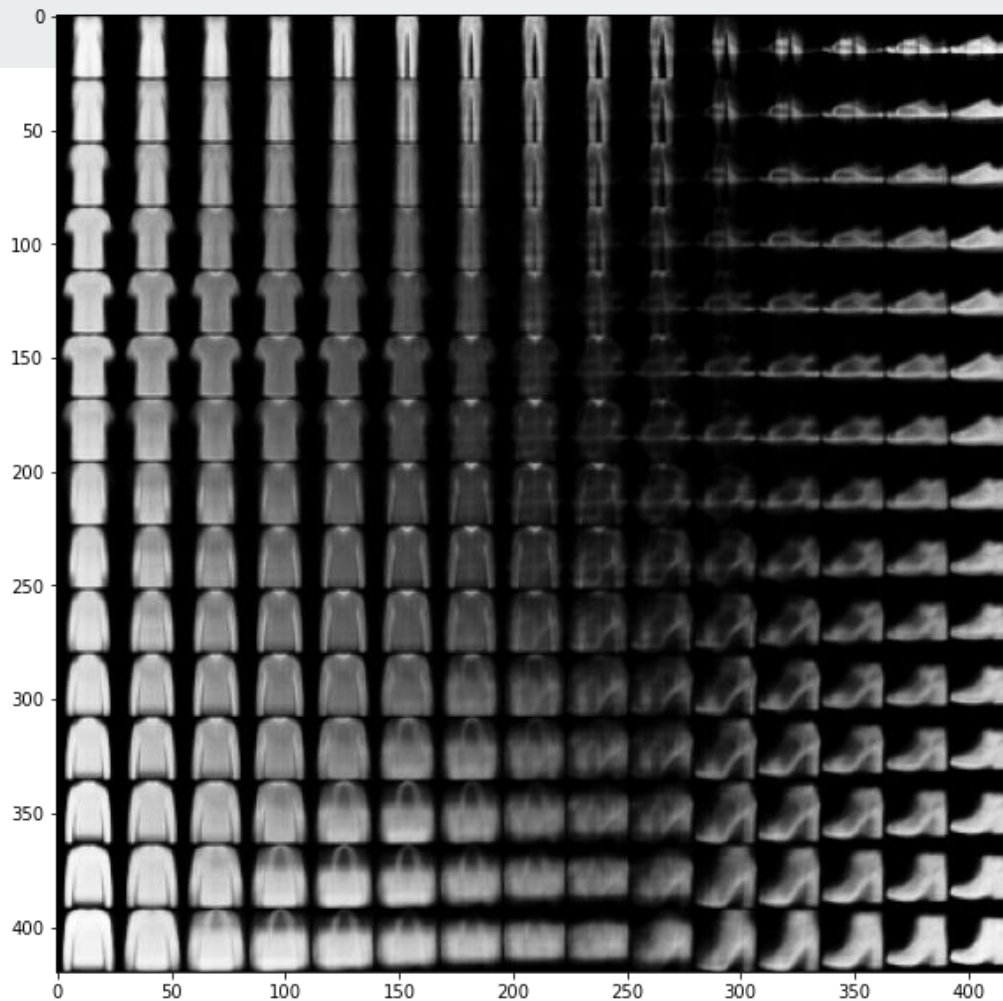
Resultados



Resultados



Dimensiones intermedias: 375





Muchas gracias

Grupo 19

Integrantes:

- Lucas Catolino
- Matias Ricarte