Trabajo Práctico N°5
Deep Learning

Grupo 5

Gonzalo Baliarda
Franco Nicolás Estevez
Ezequiel Agustin Perez
Leandro Ezequiel Rodriguez
Lucas Agustín Vittor

Ejercicio 1

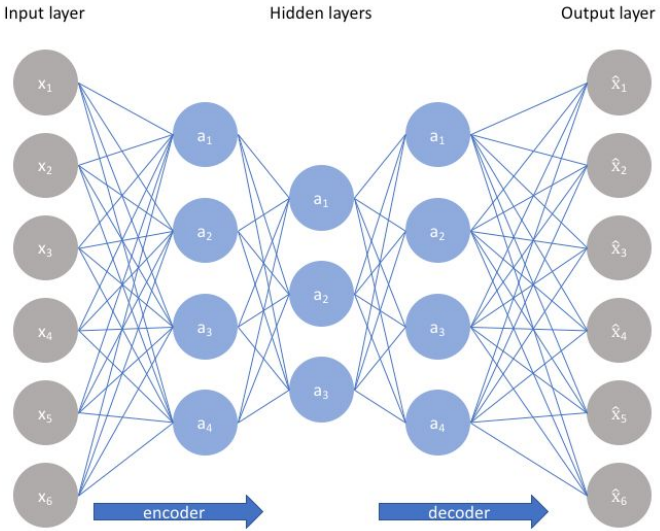
Autoencoder

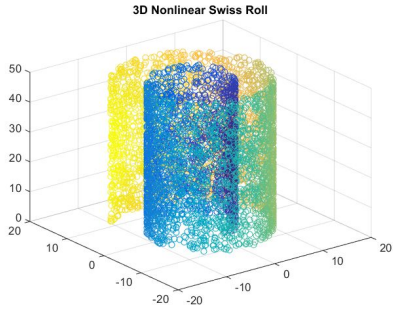
Whereas PCA attempts to discover a lower dimensional hyperplane which describes the original data, autoencoders are capable of learning nonlinear manifolds (a manifold is defined in *simple* terms as a continuous, non-intersecting surface).

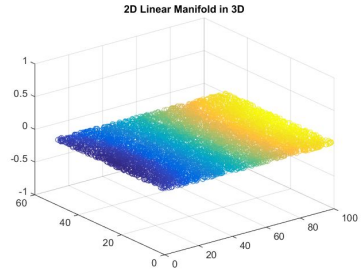
(...) Thus, our only way to ensure that the model isn't memorizing the input data is the ensure that we've sufficiently restricted the number of nodes in the hidden layer(s).

(...) By penalizing the network according to the reconstruction error, our model can learn the most important attributes of the input data and how to best reconstruct the original input from an "encoded" state. Ideally, this encoding will **learn and describe latent attributes of the input data**.

[1] https://www.jeremyjordan.me/autoencoders/

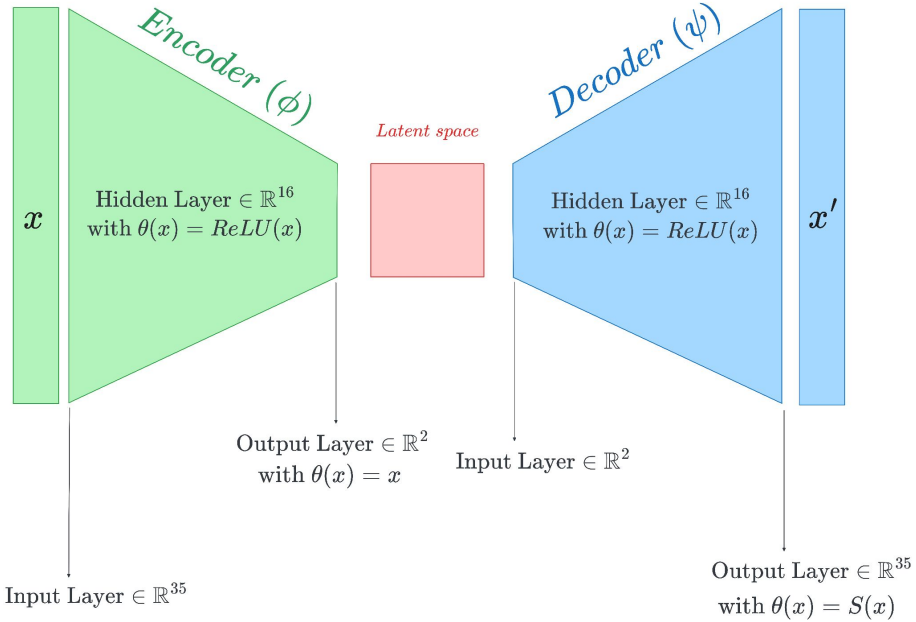




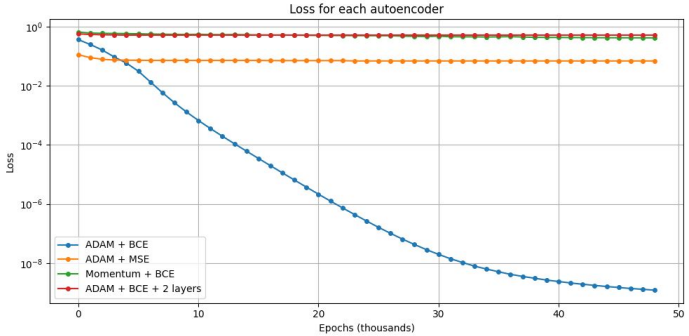


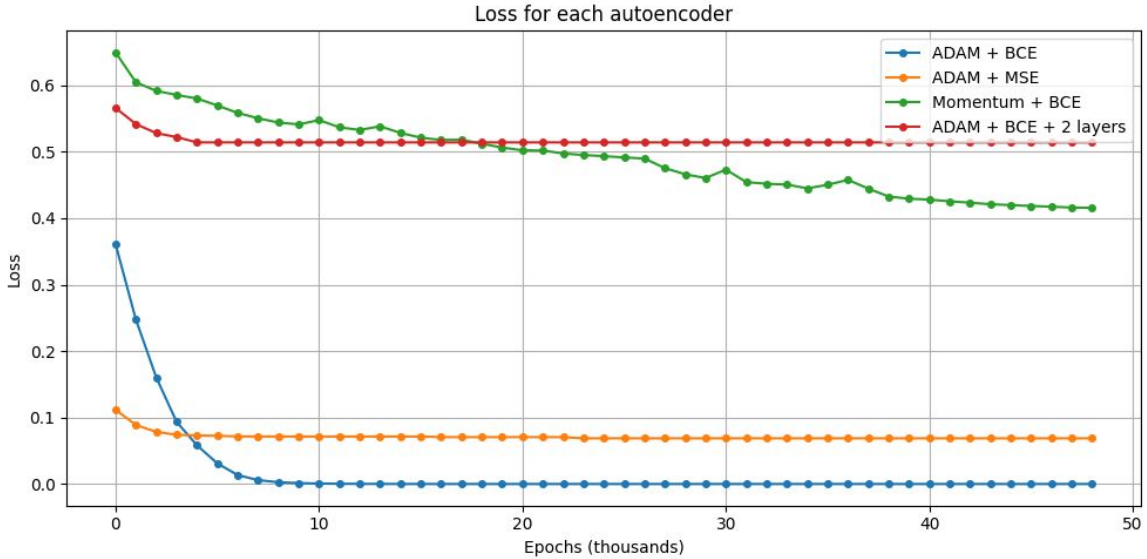
Basic Autoencoder

Arquitectura



Variaciones de arquitectura

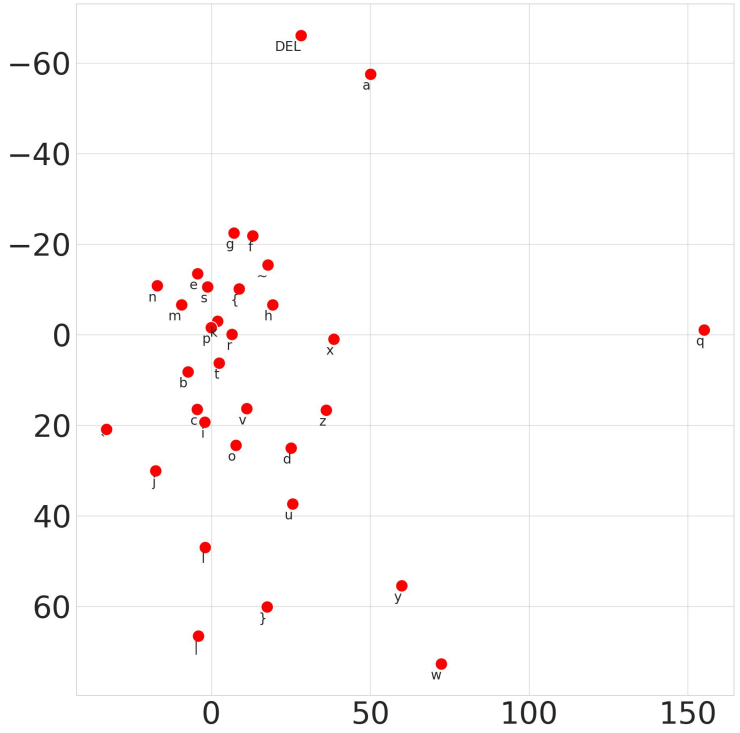




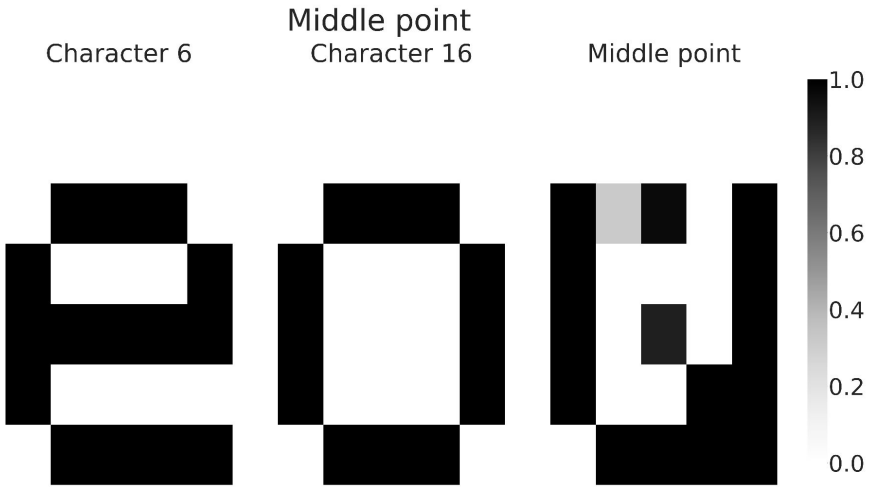

```
learning rate: 0.001
epochs: 50000
momentum_rate: 0.9
adam_optimization: {
  beta1: 0.9
  beta2: 0.999
  epsilon: 1e-8
}
```

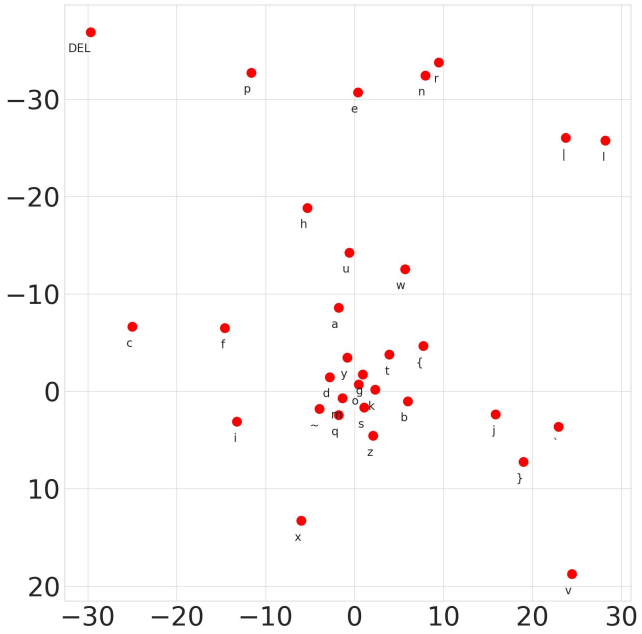
1 capa oculta => [16]
2 capas ocultas => [22, 10]

Espacio latente



Capacidad de Generación



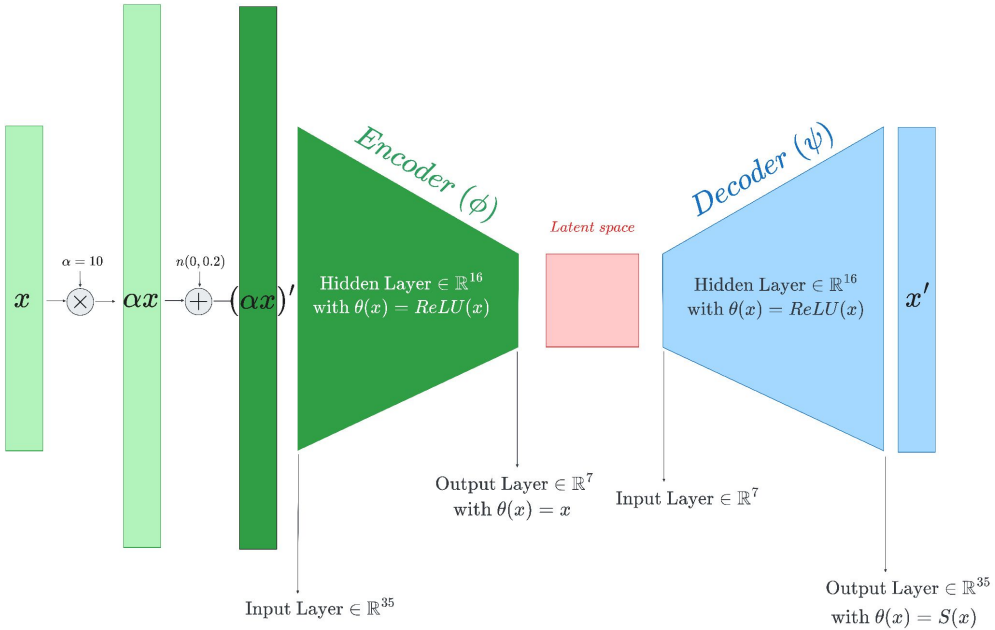


Conclusiones

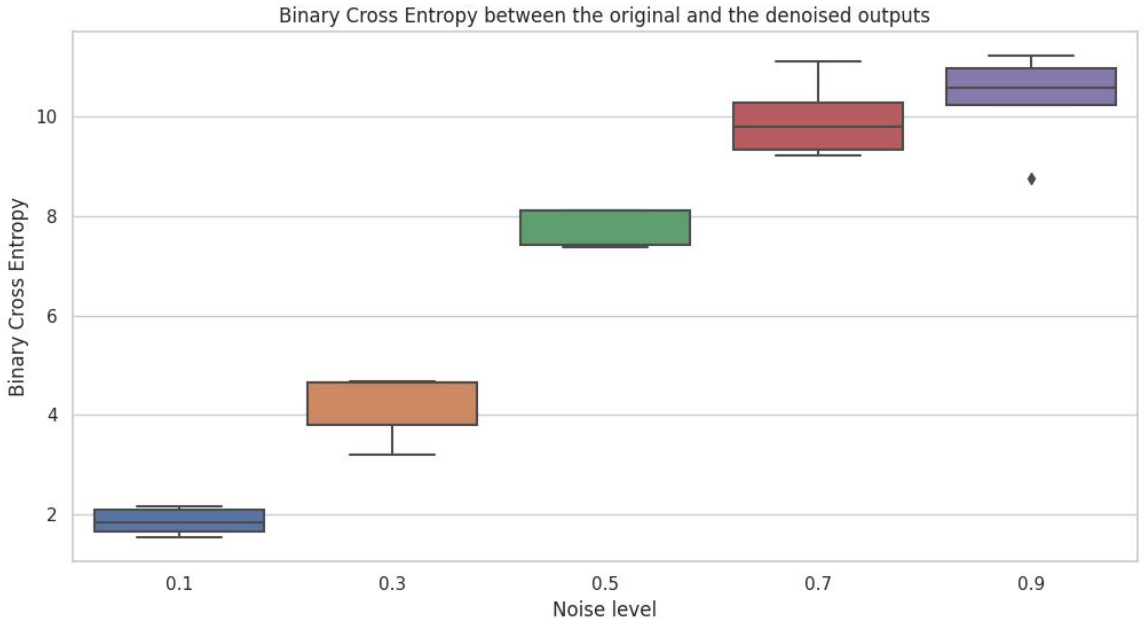
- La arquitectura del autoencoder impacta fuertemente en su desempeño.
- ADAM es ampliamente superior a Momentum.
- BCE es superior a MCE como función de pérdida.
- El cambio de BCE a MSE impacta menos que el de ADAM a Momentum.
- La capacidad generativa del Autoencoder básico no es muy buena.

Denoising Autoencoder

Arquitectura

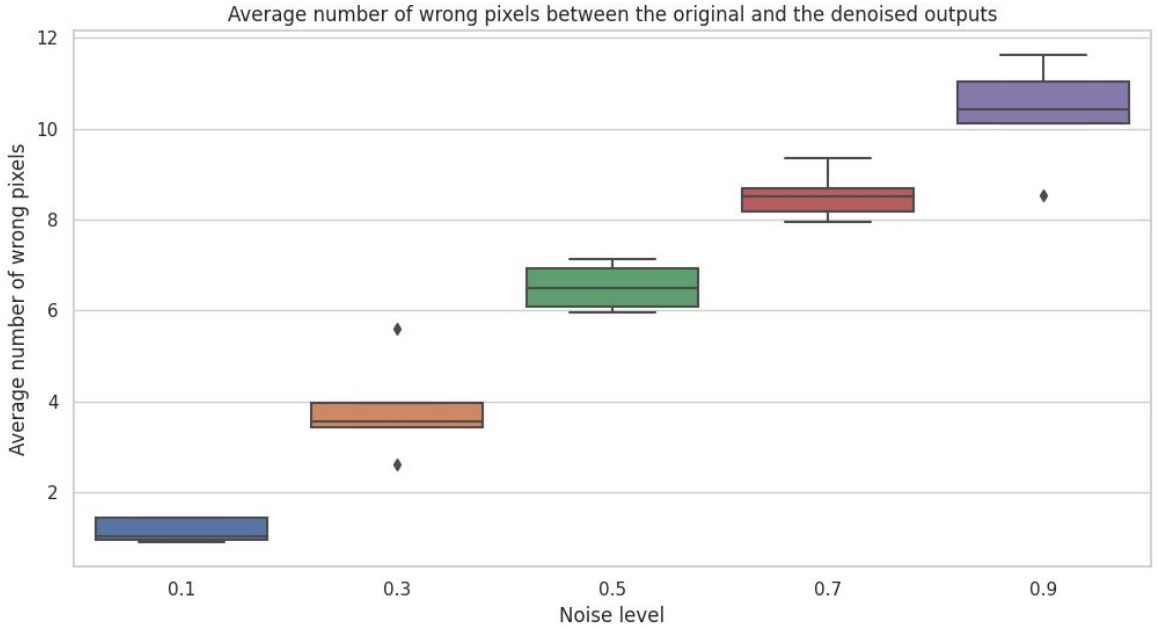


Pérdida vs Nivel de Ruido



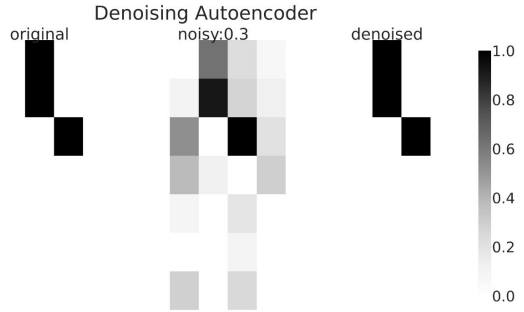

```
learning rate: 0.001
epochs: 50000
denoising autoencoder: {
  train noise: 0.2
  predict rounds: 5
  data augmentation factor: 10
}
adam_optimization: {
  beta1: 0.9
  beta2: 0.999
  epsilon: 1e-8
}
```

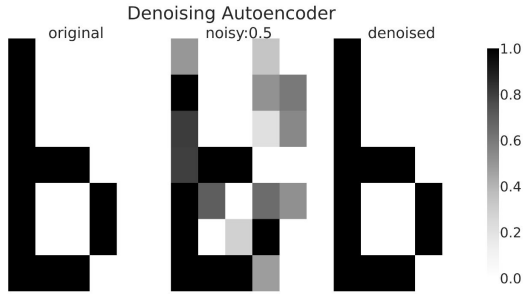
Promedio de píxeles diferentes vs Ruido

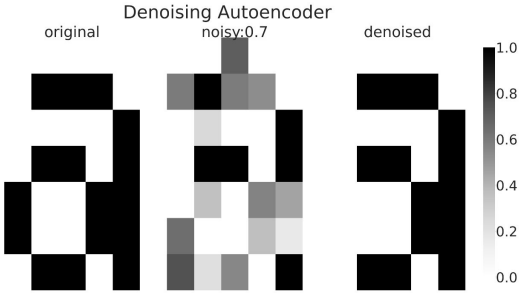


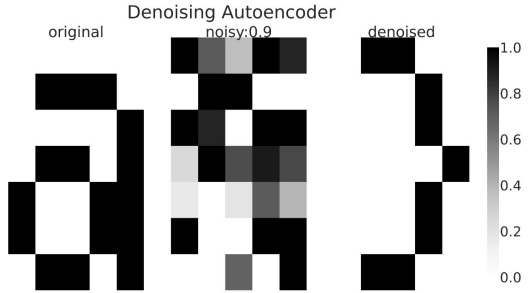

```
learning rate: 0.001
epochs: 50000
denoising autoencoder: {
  train noise: 0.2
  predict rounds: 5
  data augmentation factor: 10
}
adam_optimization: {
  beta1: 0.9
  beta2: 0.999
  epsilon: 1e-8
}
```

Capacidad de denoising









Conclusiones

- La arquitectura del autoencoder es importante, pero saber definir el training set puede mejorar dramáticamente la performance.

- Hay que encontrar la tupla de hiperparametros *(learning_rate*, *epochs*, *train_niose*, *hidden_nodes*, *laten_dim*, *data_augmentation_factor*, *optimization_algorithm)* que minimice la loss function.

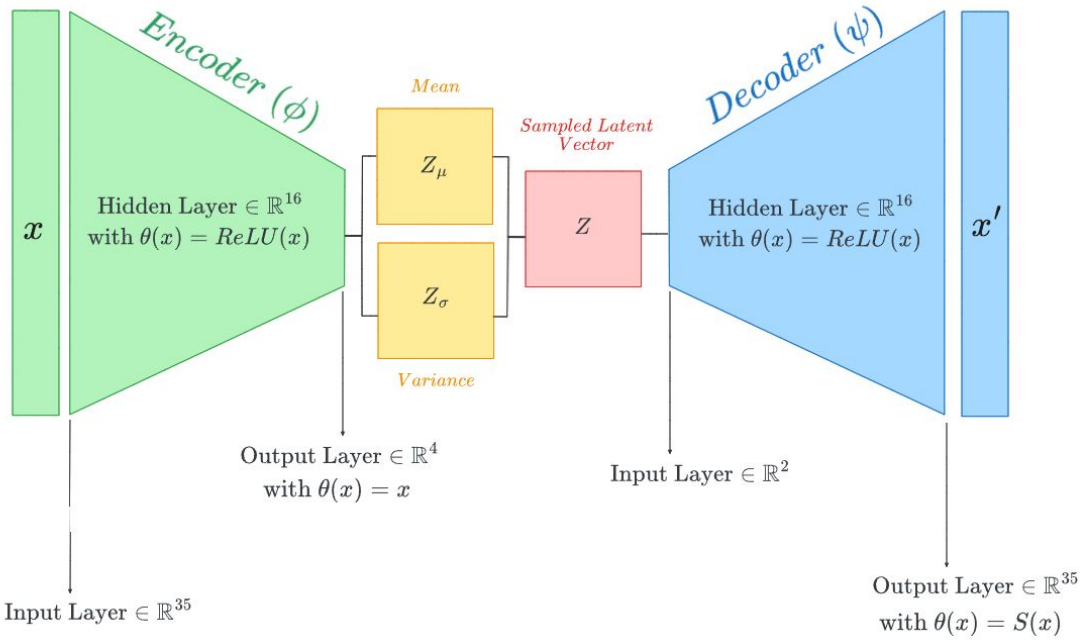
- Si bien tenemos un gráfico de la loss function a varios noise levels, es importante tener ejemplos del output del modelo para entender cualitativamente si performana bien o no.

Ejercicio 2

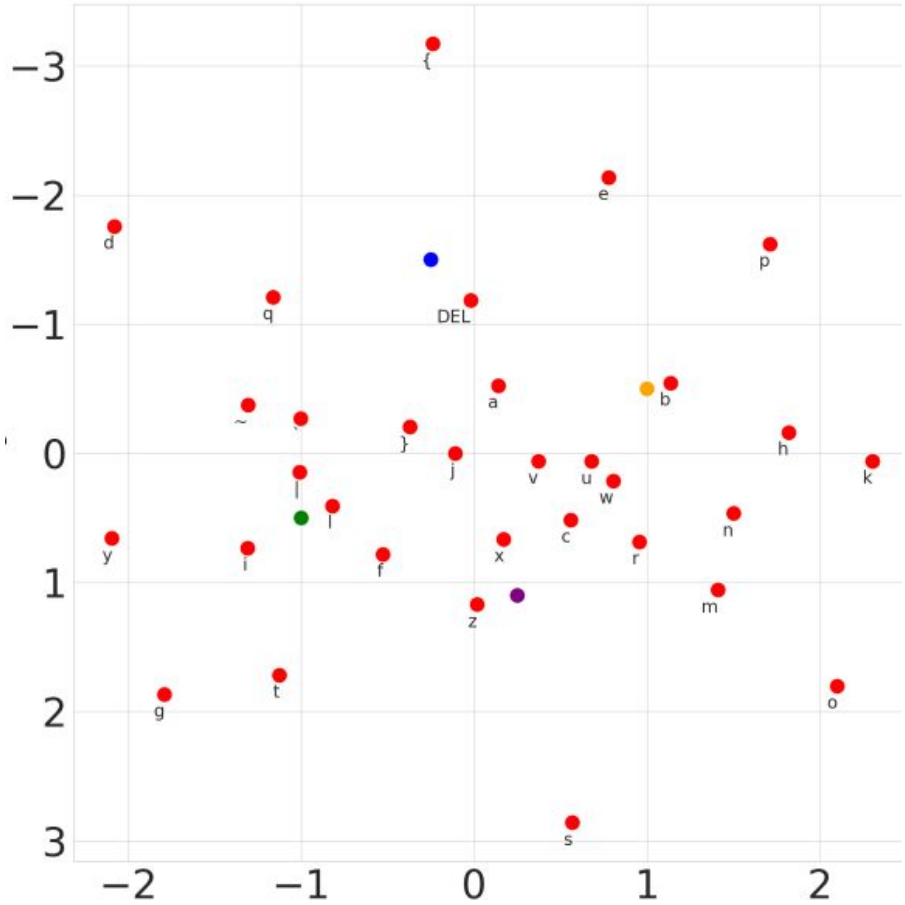
Variational Autoencoder

Arquitectura

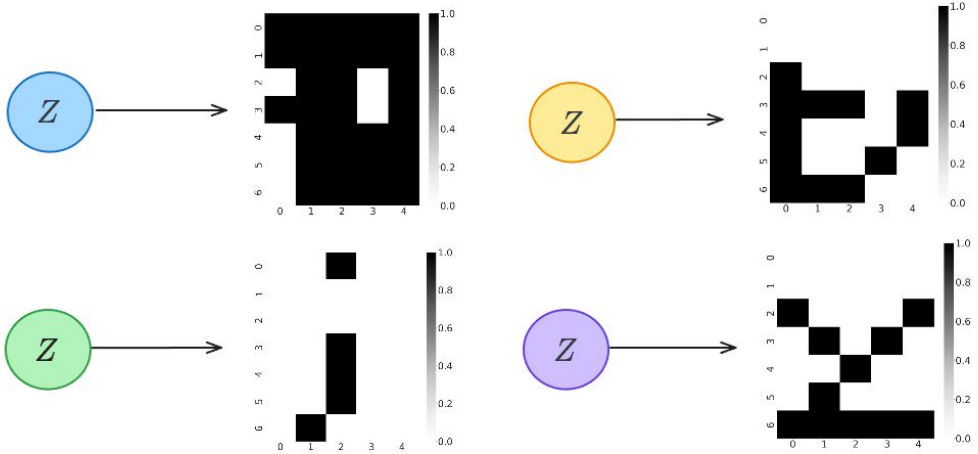
- **Optimizador**: ADAM.
- **Función de pérdida**: BCE y divergencia KL.

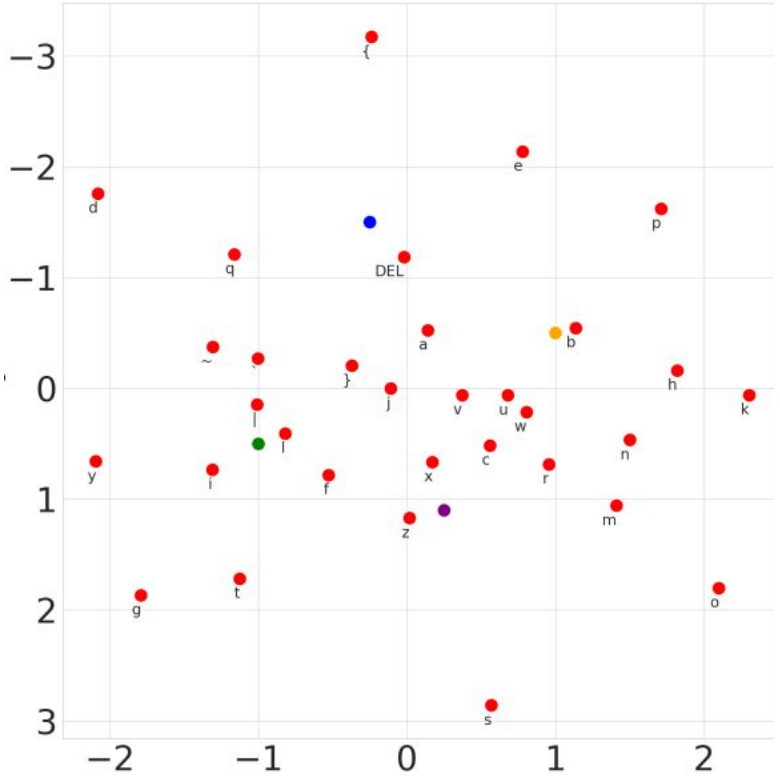


Espacio Latente



Espacio Latente

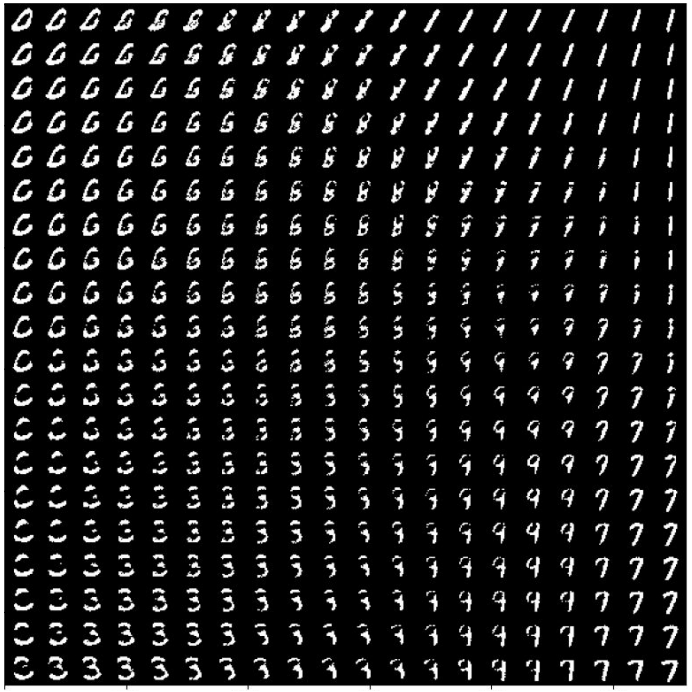


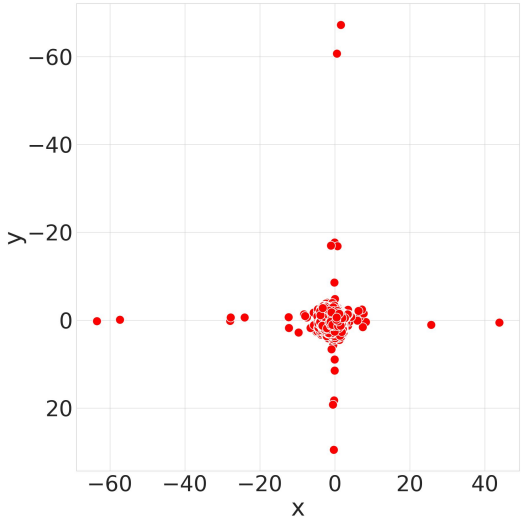


Recorrido del Espacio Latente

MNIST

- Dataset 60000
- Capa oculta 256
- Capa entrada 784
- Capa latente 2
- Tamaño batch 100
- 50 épocas





Conclusiones

- El espacio latente es mucho más compacto que en el Autoencoder básico.
- La capacidad generativa del VAE es mucho mejor.
- Para datasets muy grandes, es necesario entrenar usando la técnica de mini-batch.

Gracias