

Aprendizaje profundo: Autoencoders

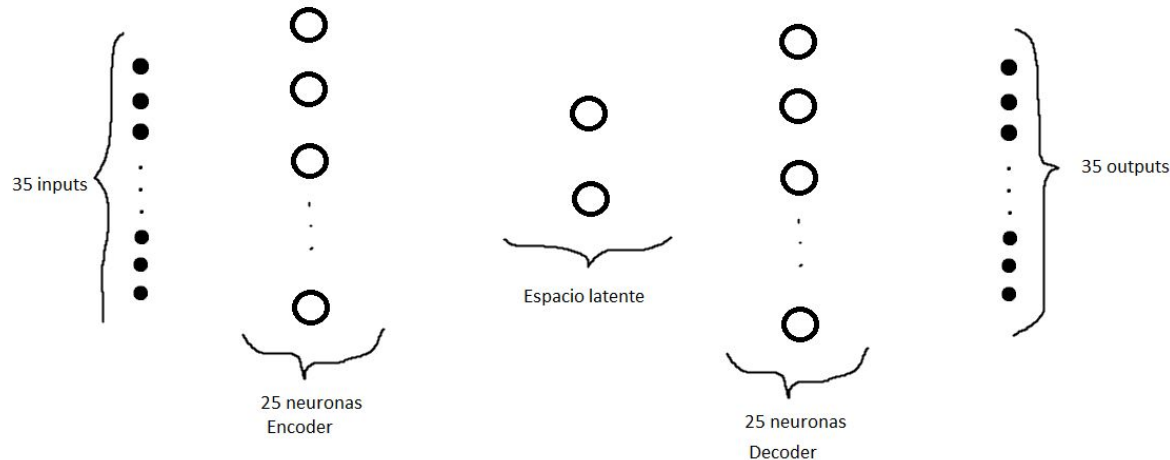
Grupo 6: Katan, Paganini

Ejercicio 1.a

Arquitectura resultante y pasos para llegar a la misma

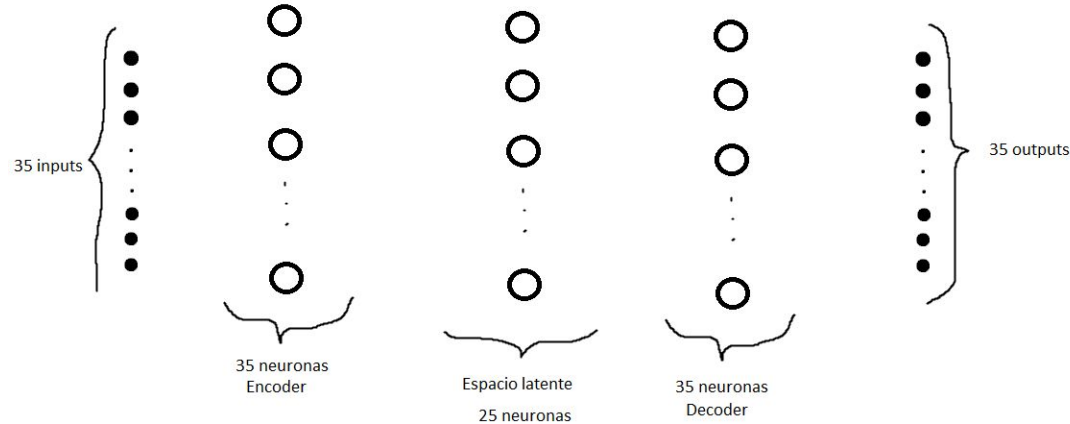
Arquitectura resultante

- Función de activación: tanh
- Error máximo: 3.3
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.85
- Aprendió un subconjunto de 10 elementos en 4003 épocas



¿Cómo llegamos a la arquitectura resultante?

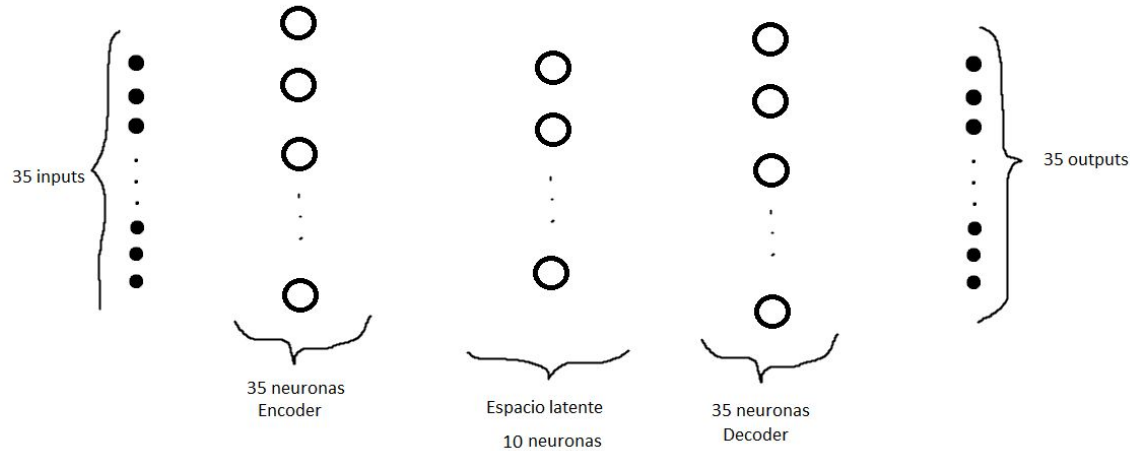
Arquitectura inicial



- Función de activación: tanh
- Error máximo: 0.5
- $\eta=0.01$
- Aprendió las 32 muestras en 2161 épocas

¿Cómo llegamos a la arquitectura resultante?

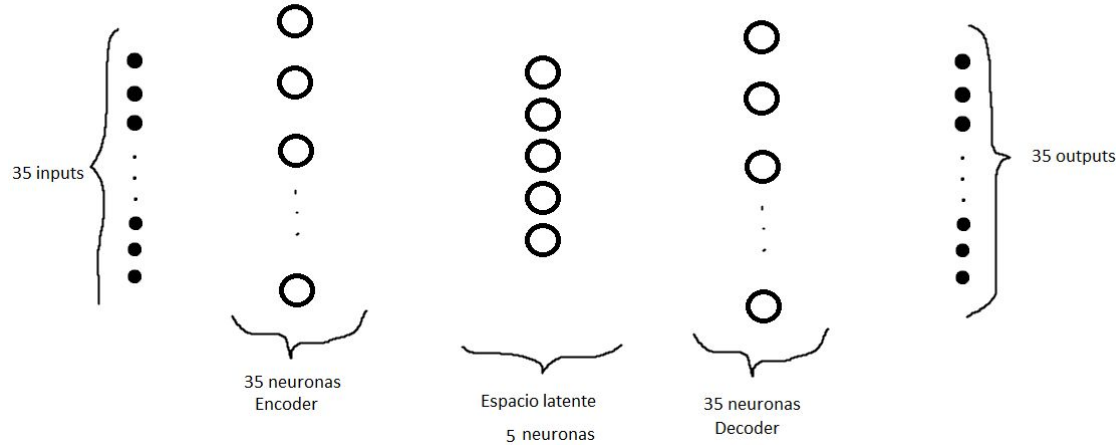
Modificando la arquitectura inicial



- Reduciendo de a 5 neuronas el espacio latente, analizando el comportamiento
- Función de activación: tanh
- Error máximo: 0.5
- $\eta=0.01$
- Aprendió las 32 muestras en 4759 épocas

¿Cómo llegamos a la arquitectura resultante?

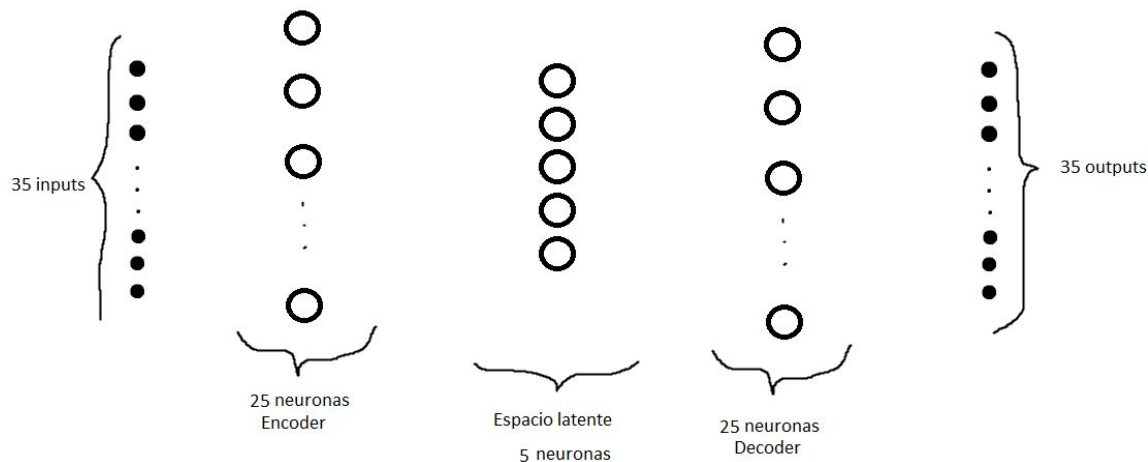
Identificamos una limitación



- Función de activación: tanh
- Error máximo: 0.5
- $\eta=0.01$
- Después de 50.000 épocas, el error no parece seguir bajando, así que cortamos la ejecución

¿Cómo llegamos a la arquitectura resultante?

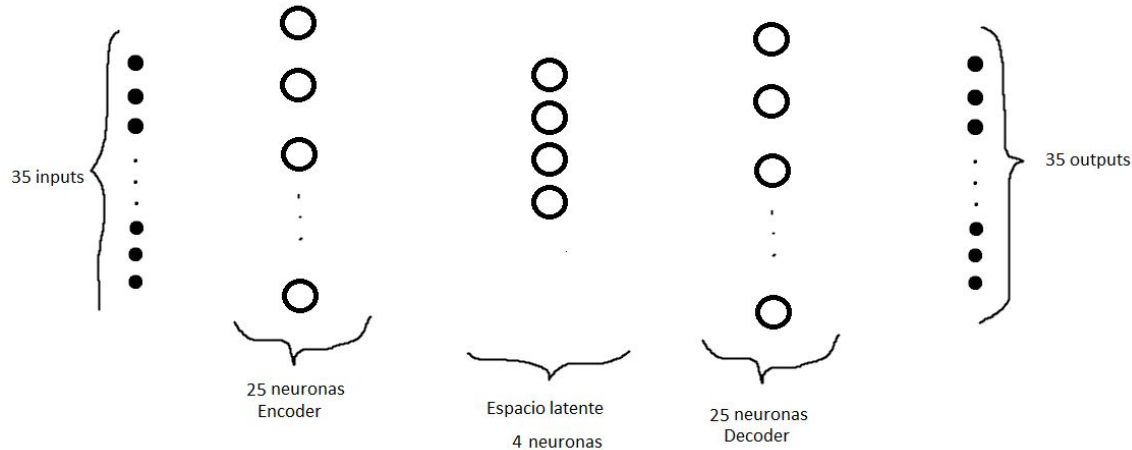
Después de varias modificaciones, logramos que aprenda con 5 neuronas en el espacio latente



- Función de activación: tanh
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.9
- error máximo: 2.0
- Aprendió 30 muestras en 33684 épocas

¿Cómo llegamos a la arquitectura resultante?

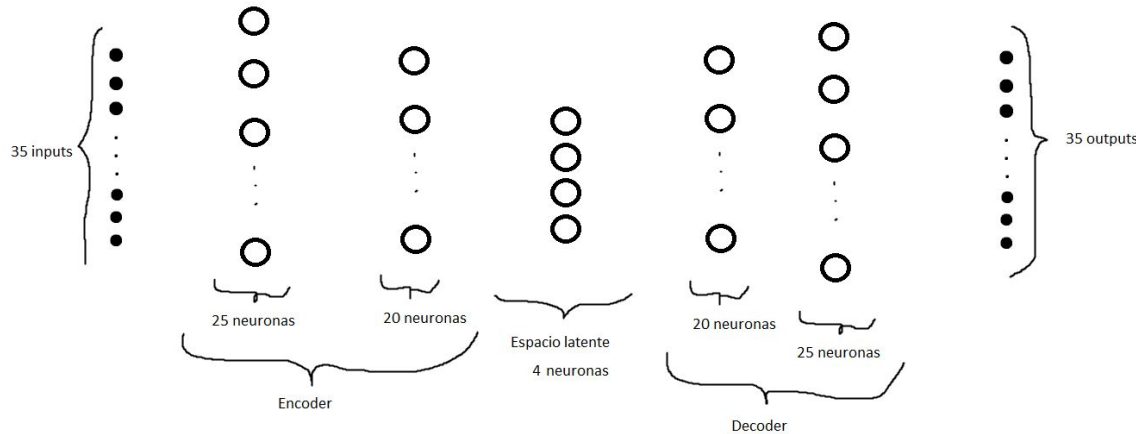
Otra limitación con 4 neuronas en el espacio latente



- Función de activación: tanh
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.9
- error máximo: 2.0
- Después de 100.000 épocas cortamos la ejecución

¿Cómo llegamos a la arquitectura resultante?

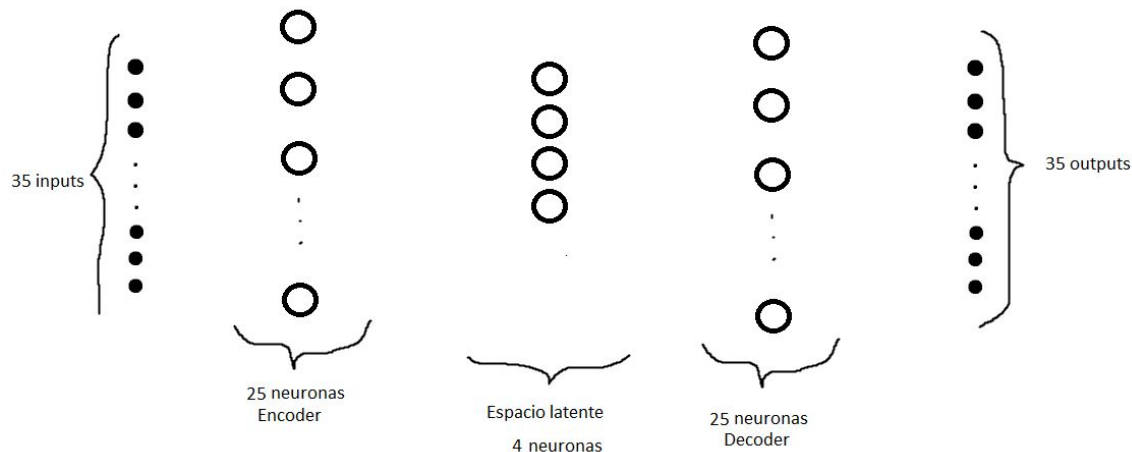
Experimentamos con más capas sin resultados



- Función de activación: tanh
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.9
- error máximo: 2.0
- Después de 100.000 épocas cortamos la ejecución

¿Cómo llegamos a la arquitectura resultante?

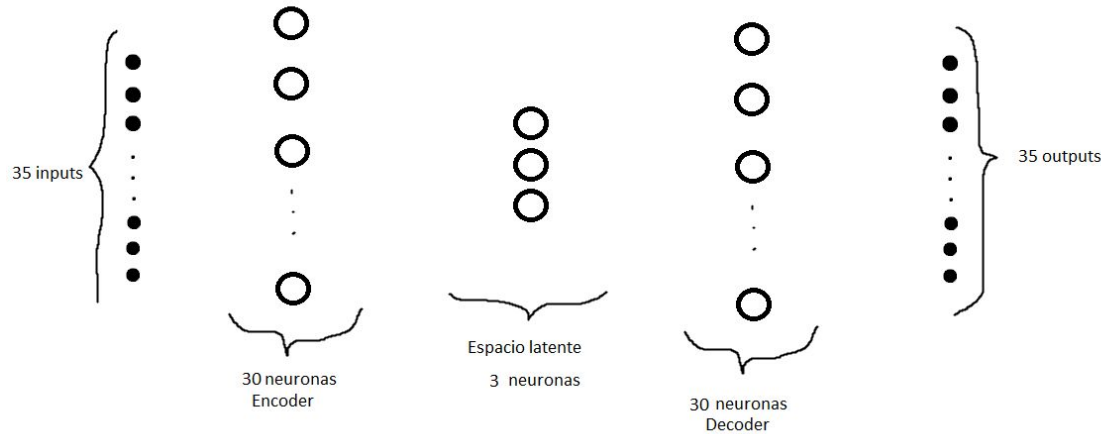
Logramos que aprenda con 4 neuronas en el espacio latente
Reduciendo la cantidad de muestras



- Función de activación: tanh
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.9
- error máximo: 2.0
- Después de 86.221 épocas aprende 20 muestras

¿Cómo llegamos a la arquitectura resultante?

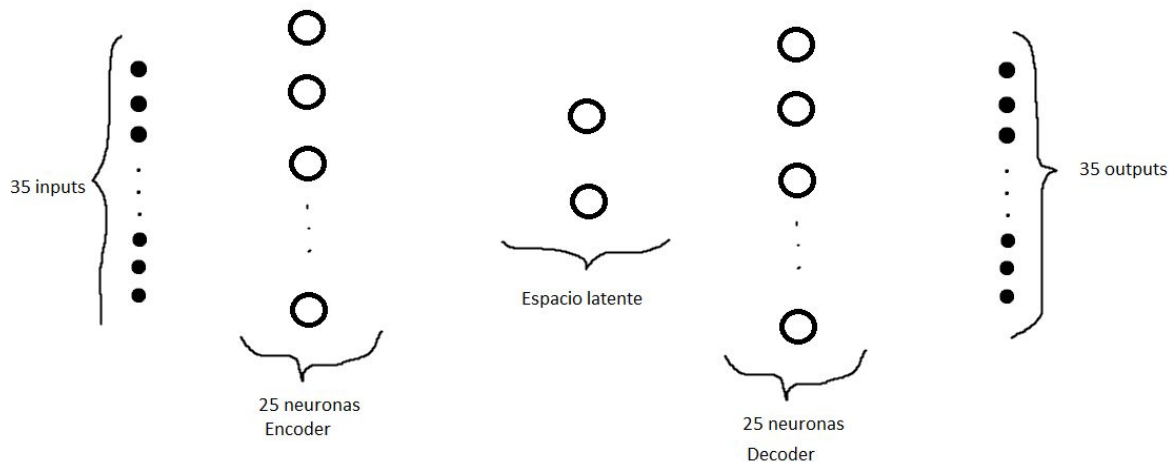
Aprendiendo con 3 neuronas en el espacio latente
Reduciendo muestras y cambiando la red



- Función de activación: tanh
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.9
- error máximo: 3.0
- Después de 60.428 épocas aprende 15 muestras

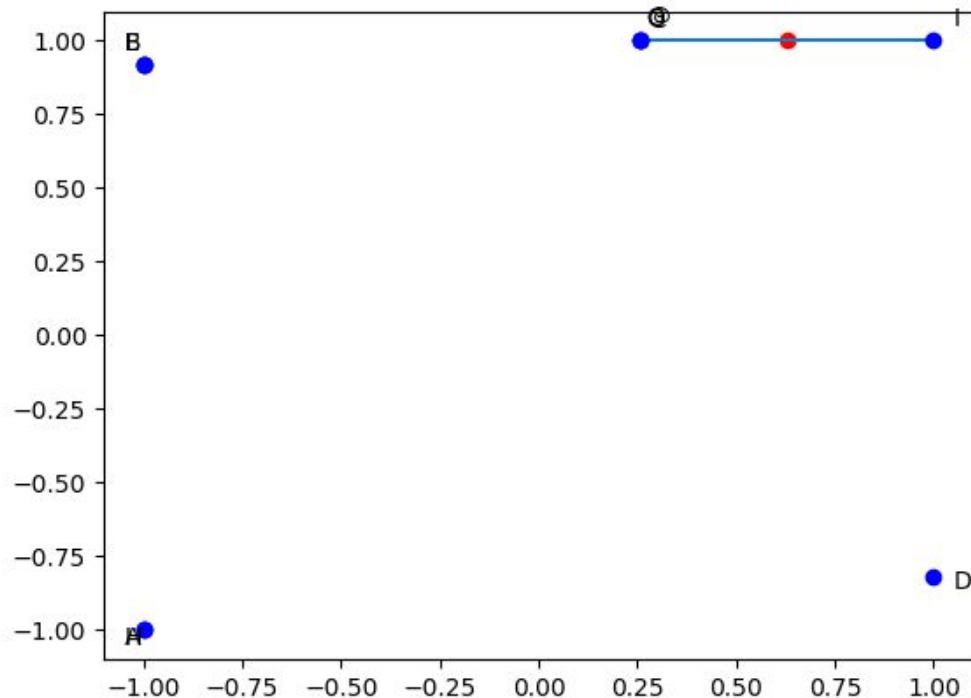
¿Cómo llegamos a la arquitectura resultante?

Cambiando parámetros y reduciendo muestras, llegamos a la arquitectura final de 2 neuronas en el espacio latente



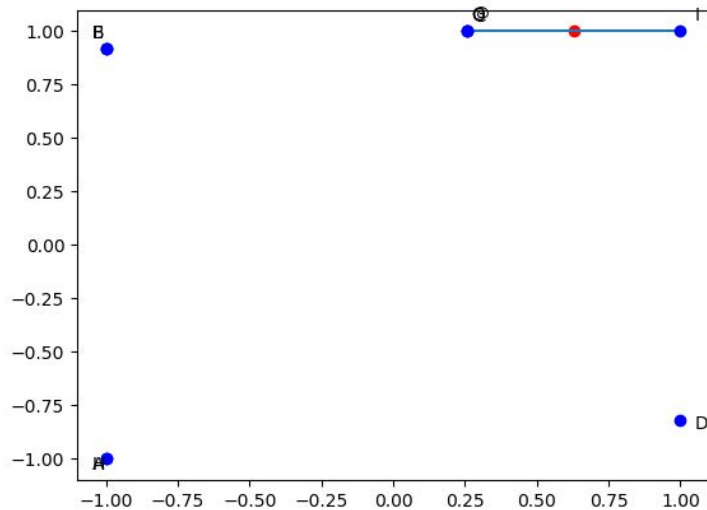
- Función de activación: tanh
- Error máximo: 3.3
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.85
- Aprendió un subconjunto de 10 elementos en 4003 épocas

Datos de entrada en el espacio latente



```
Letter positions in latent space:  
@: (0.25911897606381284, 1.0), index: 0  
A: (-0.9961891776363734, -1.0), index: 1  
B: (-1.0, 0.9189357411736235), index: 2  
C: (0.25911897606381284, 1.0), index: 3  
D: (1.0, -0.8195594125921941), index: 4  
E: (-1.0, 0.9189357411736235), index: 5  
F: (-1.0, 0.9189357411736235), index: 6  
G: (0.25911897606381284, 1.0), index: 7  
H: (-1.0, -1.0), index: 8  
I: (1.0, 1.0), index: 9
```

Generando una nueva letra



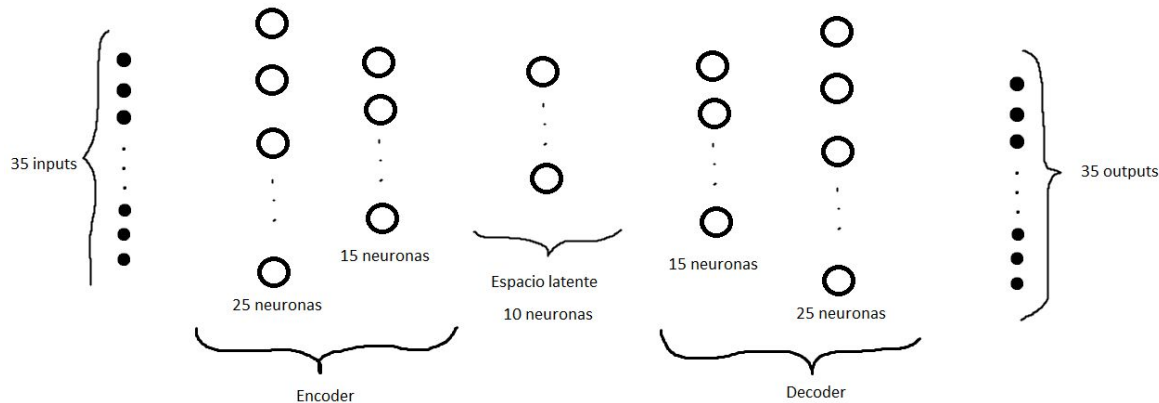
```
Point A: 0
Point B: 9
Interpolated letter:
###
#  #
#  #
#  #
#  #
#  #
###
```

Ejercicio 1.b

Denoising Autoencoder

Denoising Autoencoder: Resultados

Arquitectura



- Función de activación: tanh
- Error máximo: 0.7
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.85
- Utilizando un subconjunto de 25 elementos

Denoising Autoencoder: Resultados

Conjunto de entrenamiento, entradas de la red

Para el conjunto de entrenamiento para las entradas de la red, se tomaron las letras originales y se alteraron una cantidad de 5 bits aleatorios de las mismas

Original font representation:

```
#  
# #  
# #  
# #  
#####  
# #  
# #
```

Original font representation:

```
# #  
# #  
# #  
#  
# #  
# #  
# #
```

Noisy font representation:

```
# #  
# #  
# #  
# #  
### #  
# # #  
#
```

Noisy font representation:

```
#  
#  
#  
#  
# ##  
# #  
# # #
```

Denoising Autoencoder: Resultados

Conjunto de entrenamiento, entradas de la red

Original font representation:

```
####  
#  #  
#  #  
####  
# #  
# #  
# #
```

Original font representation:

```
###  
#  #  
#  
###  
  #  
#  #  
###
```

Noisy font representation:

```
####  
#  #  
#  #  
  ##  
# #  
  ##  
#
```

Noisy font representation:

```
##  
#  
#  
####  
  #  
#  #  
####
```

Denoising Autoencoder: Resultados

Conjunto de entrenamiento, entradas de la red

Original font representation:

```
#  #  
#  #  
#  #  
##  
#  #  
#  #  
#  #
```

Noisy font representation:

```
#  #  
#  
#  #  
  ##  
#  #  
#  #  
#
```

Original font representation:

```
#####  
#  
#  
#####  
#  
#  
#
```

Noisy font representation:

```
#####  
  
#  #  
  ##  
#  
#  #  
#
```

Denoising Autoencoder: Resultados

Pruebas alterando 5 bits, salidas de la red

Letter without noise:

```
###  
#  #  
#  
###  
  #  
#  #  
###
```

Letter with noise:

```
###  
#  #  
#  #  
###  
# ##  
# # #  
###
```

Letter without noise:

```
####  
#  #  
#  #  
####  
#  #  
#  #  
#  #
```

Letter with noise:

```
##  
#  #  
#  ##  
####  
  #  
#  ##  
#  #
```

Network output after denoising letter:

```
### |  
#  #  
#  
###  
  #  
#  #  
###
```

Network output after denoising letter:

```
####  
#  #  
#  #  
####  
#  #  
#  #  
#  #
```

Denoising Autoencoder: Resultados

Pruebas alterando 5 bits, salidas de la red

```
Letter without noise:  
#  #  
#  #  
#  #  
#  #  
#  #  
#  #  
#
```

```
Letter with noise:  
#  ##  
#  #  
#  ##  
#  #  
#  #  
#  #  
##
```

```
Letter without noise:  
###  
#  
#  
#  
#  
#  
###
```

```
Letter with noise:  
###  
#  
#  
##  
# #  
###
```

```
Network output after denoising letter:  
#  #  
#  #  
#  #  
#  #  
#  #  
#  #  
#
```

```
Network output after denoising letter:  
###  
#  
#  
#  
#  
#  
###
```

Denoising Autoencoder: Resultados

Pruebas alterando 5 bits, salidas de la red

Letter without noise:

```
####  
#  #  
#  #  
####  
#  
#  
#
```

Letter with noise:

```
####  
    #  
#  ##  
####  
#  
## #  
#
```

Letter without noise:

```
#  #  
#  #  
#  #  
# # #  
# # #  
## ##  
#  #
```

Letter with noise:

```
#  #  
#  
#  #  
# # #  
## ##  
#
```

Network output after denoising letter:

```
####  
#  #  
#  #  
####  
# #  
# #  
#  #
```

Network output after denoising letter:

```
#  #  
#  #  
#  #  
#####  
#  #  
#  #  
#  #
```

Denoising Autoencoder: Resultados

Pruebas alterando 7 bits, salidas de la red

Letter without noise:

```
#  #  
#  #  
#  #  
# # #  
# # #  
## ##  
#  #
```

Letter with noise:

```
#  #  
    #  
#  #  
#  #  
### #  
#  #  
    #
```

Letter without noise:

```
####  
#  #  
#  #  
####  
#  
#  
#
```

Letter with noise:

```
####  
  
# ###  
####  
#  
  
# #
```

Network output after denoising letter:

```
#  #  
#  #  
#  #  
# # #  
# # #  
## ##  
#  #
```

Network output after denoising letter:

```
####  
#  #  
#  #  
####  
#  
#  
#
```

Denoising Autoencoder: Resultados

Pruebas alterando 7 bits, salidas de la red

Letter without noise:

```
#  
#  
#  
#  
#  
#  
#####
```

Letter with noise:

```
###  
  
#  
##  
# # #  
#  
### #
```

Letter without noise:

```
#####  
#  
#  
###  
#  
#  
#####
```

Letter with noise:

```
# #  
# #  
##  
###  
# #  
# #  
#####
```

Network output after denoising letter:

```
####  
# #  
# #  
####  
# #  
# #  
# #
```

Network output after denoising letter:

```
####  
# #  
#  
###  
#  
# #  
####
```


Denoising Autoencoder: Resultados

Pruebas alterando 10 bits, salidas de la red

Letter without noise:

#####

#

#

#

#

#

#

Letter with noise:

#####

#

#

###

##

Letter without noise:

###

#

#

#

#

#

#

Letter with noise:

##

###

#

#

#

#####

#

Network output after denoising letter:

###

#

#

#

#

#

###

Network output after denoising letter:

###

#

#

#

#

#

#

Denoising Autoencoder: Resultados

Pruebas alterando 10 bits, salidas de la red

Letter without noise:

```
####  
#  #  
#  #  
####  
#  #  
#  #  
#  #
```

Letter with noise:

```
  ##  
# ###  
# # #  
#####  
# ##  
#  #  
#  #
```

Letter without noise:

```
####  
#  #  
#  #  
####  
#  
#  
#
```

Letter with noise:

```
####  
### #  
# ##  
# ###  
# # #  
#  #
```

Network output after denoising letter:

```
#  #  
#  #  
# ##  
##  #  
#  #  
#  #  
#  #
```

Network output after denoising letter:

```
#  #  
#  #  
#  #  
# # #  
# # #  
## ##  
#  #
```

Denoising Autoencoder: Resultados

Pruebas alterando 15 bits, salidas de la red

Letter without noise:

```
# #  
# #  
# #  
# #  
# #  
# #  
#
```

Letter with noise:

```
## #  
#  
##  
### #  
#  
## ##  
##
```

Letter without noise:

```
###  
# #  
#  
#  
# ##  
# #  
####
```

Letter with noise:

```
# ##  
#  
# #  
#  
##  
# #  
##
```

Network output after denoising letter:

```
###  
# #  
#  
###  
#  
# #  
###
```

Network output after denoising letter:

```
###  
# #  
# #  
#  
# #  
# #  
####
```

Ejercicio 2

Generando una nueva muestra

Conjunto de datos usado

Bitmaps de dígitos del 0 al 9 de 7x5

```
###  
#  #  
#  ##  
# # #  
##  #  
#  #  
###  
Digit: 0
```

```
#  
##  
#  
#  
#  
#  
###  
Digit: 1
```

```
###  
#  #  
#  #  
#  
#  
#  
#####  
Digit: 2
```

```
###  
#  #  
#  #  
##  
#  
#  #  
###  
Digit: 3
```

```
#  
##  
# #  
# #  
#####  
#  
#  
Digit: 4
```

```
#####  
#  
####  
#  
#  
# #  
###  
Digit: 5
```

```
##  
#  
#  
####  
# #  
# #  
###  
Digit: 6
```

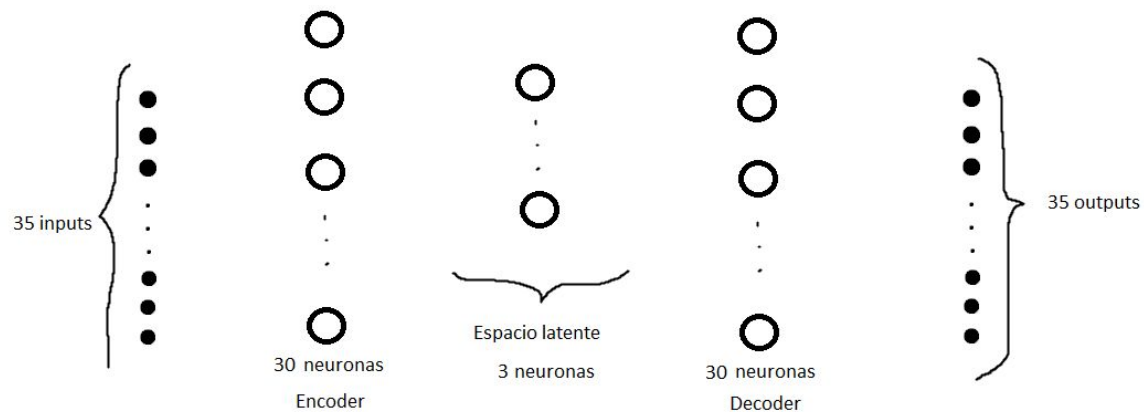
```
#####  
#  
#  
#  
#  
#  
#  
Digit: 7
```

```
###  
#  #  
#  #  
###  
#  #  
#  #  
###  
Digit: 8
```

```
###  
#  #  
#  #  
####  
#  
#  
##  
Digit: 9
```

Autoencoder

Arquitectura



- Función de activación: tanh
- Error máximo: 0.3
- eta adaptativo con etaBase=0.01 y deltaEta=0.005 cada 5 épocas constantes
- momentum con alpha=0.85

Resultados

Generando un nuevo dígito interpolando dos dígitos existentes en el espacio latente

Puntos a interpolar en el espacio latente

Point A digit:

```
###  
#  #  
#  ##  
# # #  
##  #  
#  #  
###
```

Point B digit:

```
###  
#  #  
    #  
    ##  
    #  
#  #  
###
```

Resultado con un factor de interpolación de 0.5

Interpolation factor: 0.5

```
###  
#  #  
#  #  
#####  
#  #  
#  #  
###
```

Resultados

Utilizando distintos factores de interpolación, vemos cómo se va “transformando” la nueva muestra desde un punto hacia el otro

Puntos a interpolar en el espacio latente

Point A digit:

```
###  
#  #  
#  ##  
#  #  #  
##  #  
#  #  
###
```

Point B digit:

```
###  
#  #  
#  #  
#  #  
##  
#  #  
#  #  
###
```

Muestras generadas

Interpolation factor: 0.1

```
###  
#  #  
#  ##  
#  #  #  
##  #  
#  #  
###
```

Interpolation factor: 0.5

```
###  
#  #  
#  #  
#####  
#  #  
#  #  
###
```

Interpolation factor: 0.9

```
###  
#  #  
#  #  
##  
#  #  
#  #  
###
```