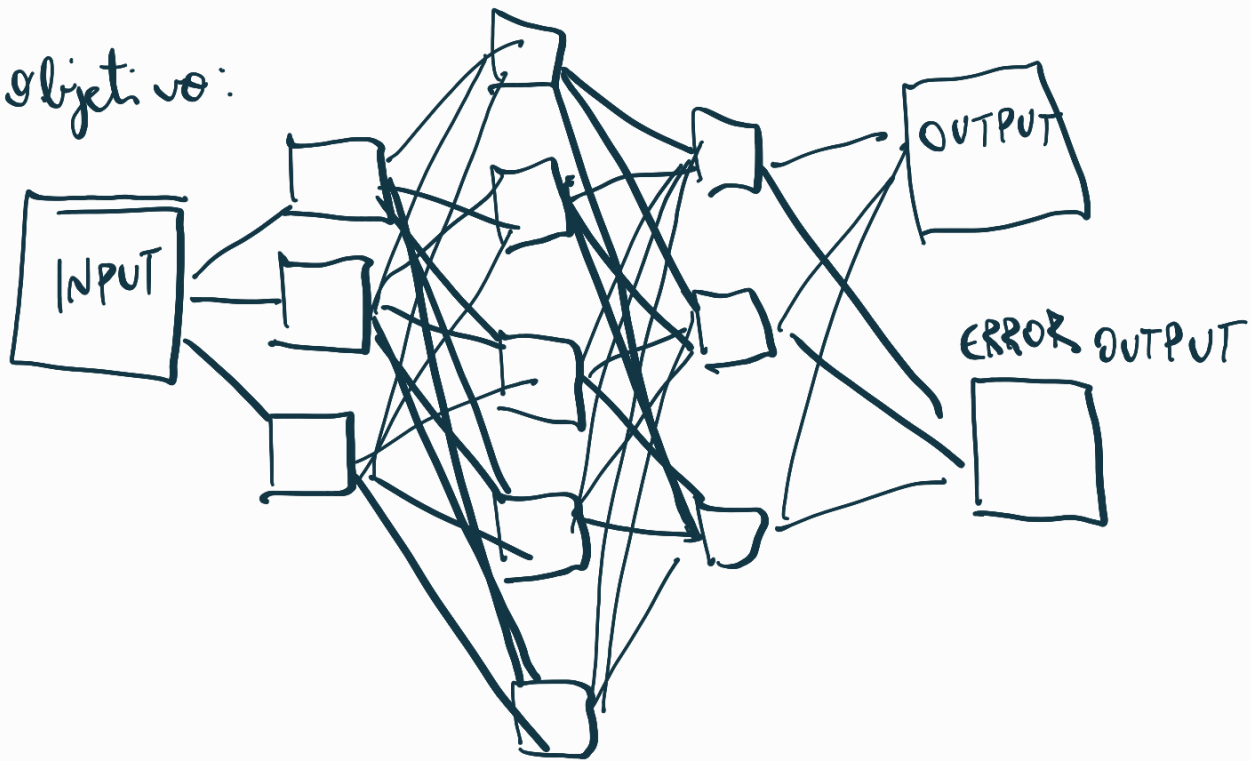


# Rede Neural.

objetivo:



## - Construção de simples dataset:

Construindo com numpy simples base de dados, no qual:

1	1	0	0	1	0
0	0	0	0	0	1

← Test e evaluation Dataset.

temos padrões de "pixels" 2x2 ou seja 4 pixels por img.

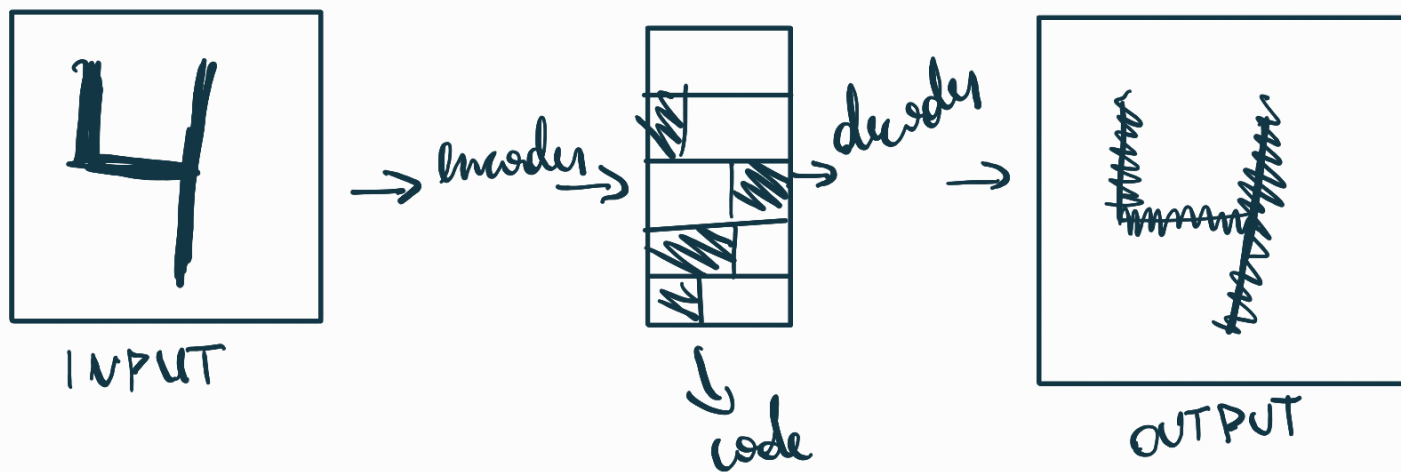
## - Construindo Autoencoder

**Encoder:** - comprime representação do input  
- escolhe features mais importantes  
- reduz dimensões.

**Decoder:**

- tenta recriar o input com o

... e o maior número de bits possível.



O código consegue extrair as features mais importantes para que imagem possa ser reconstruída.

### Normalizer:

Assurar que os valores estejam em um intervalo desejado, alterando sua escala

lento que vamos usar:

usando intervalo  $[0, 1]$

transformamos números de forma

$$(value - offset) / scale - 0.5$$



valor  
mínimo



valor máximo

valor mínimo.

Assim nosso input vai para  $-0.5$  e  $0.5$

### Denormalizer:

Volta o valor para o original.

$$(value - offset) / scale - 0.5 = x$$

$$(value - offset) / scale = x + 0.5.$$

$$(value - offset) = (x + 0.5) * scale$$

$$value = ((x + 0.5) * scale) + offset.$$

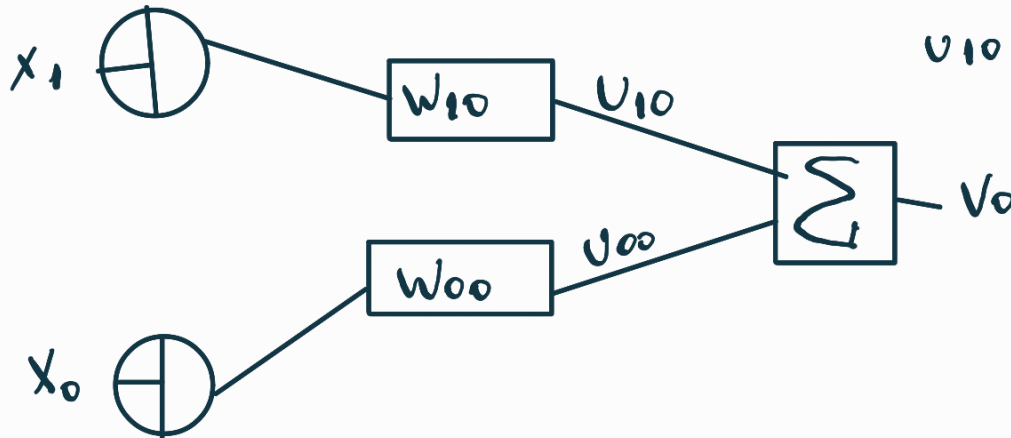
Layers:

$$y = mx + b$$

$$u_{00} = x_0 w_{00}$$

$$u_{10} = x_1 w_{10}$$

$$v_0 = u_{00} + u_{10}$$



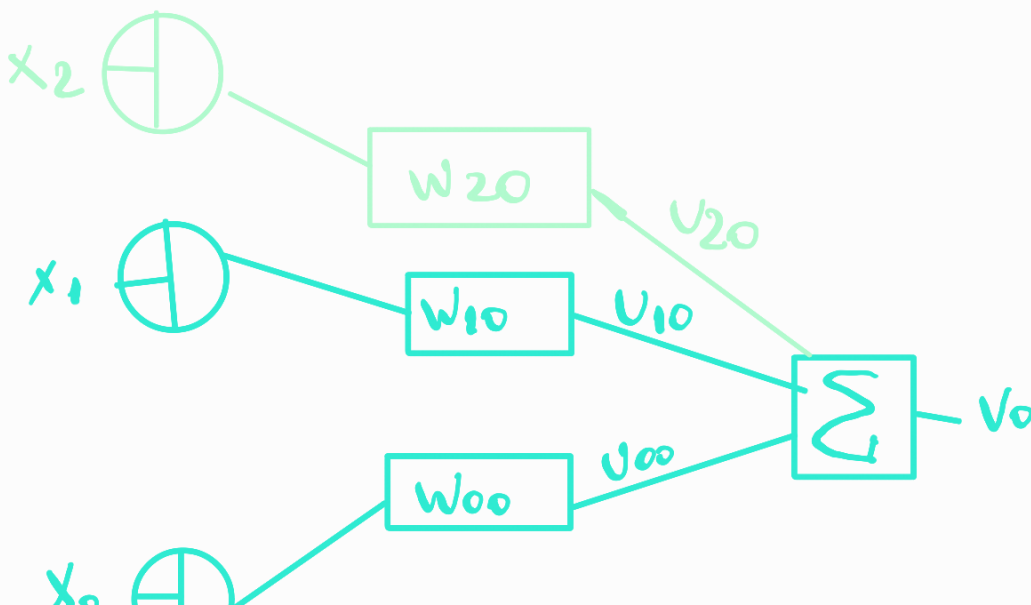
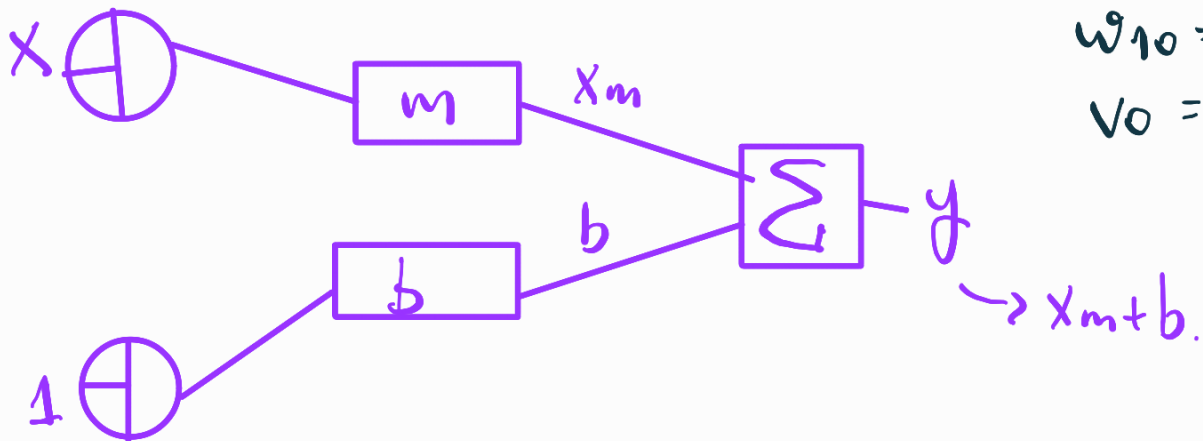
$$x_0 = 1$$

$$x_1 = x$$

$$w_{00} = b$$

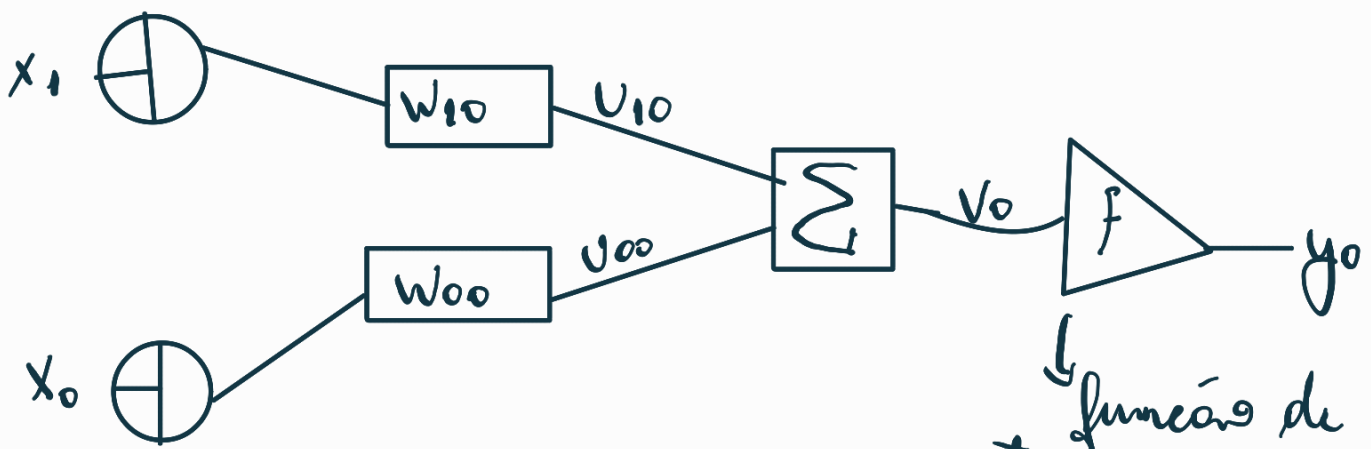
$$w_{10} = m$$

$$v_0 = y$$



$$u_{i0} = x_i w_{i0}$$

$$v_0 = \sum_i u_{i0}$$



↓ função de ativação para tornar o efeito linear

ex:  $u_{10} = x_{i0} w_{i0}$

$$v_0 = \sum_i u_{i0} \quad f(e) = \frac{1}{1 + e^{-e}} \rightarrow \text{sigmoid.}$$

$$y_0 = f(v_0)$$

logo comportamento vai de:

