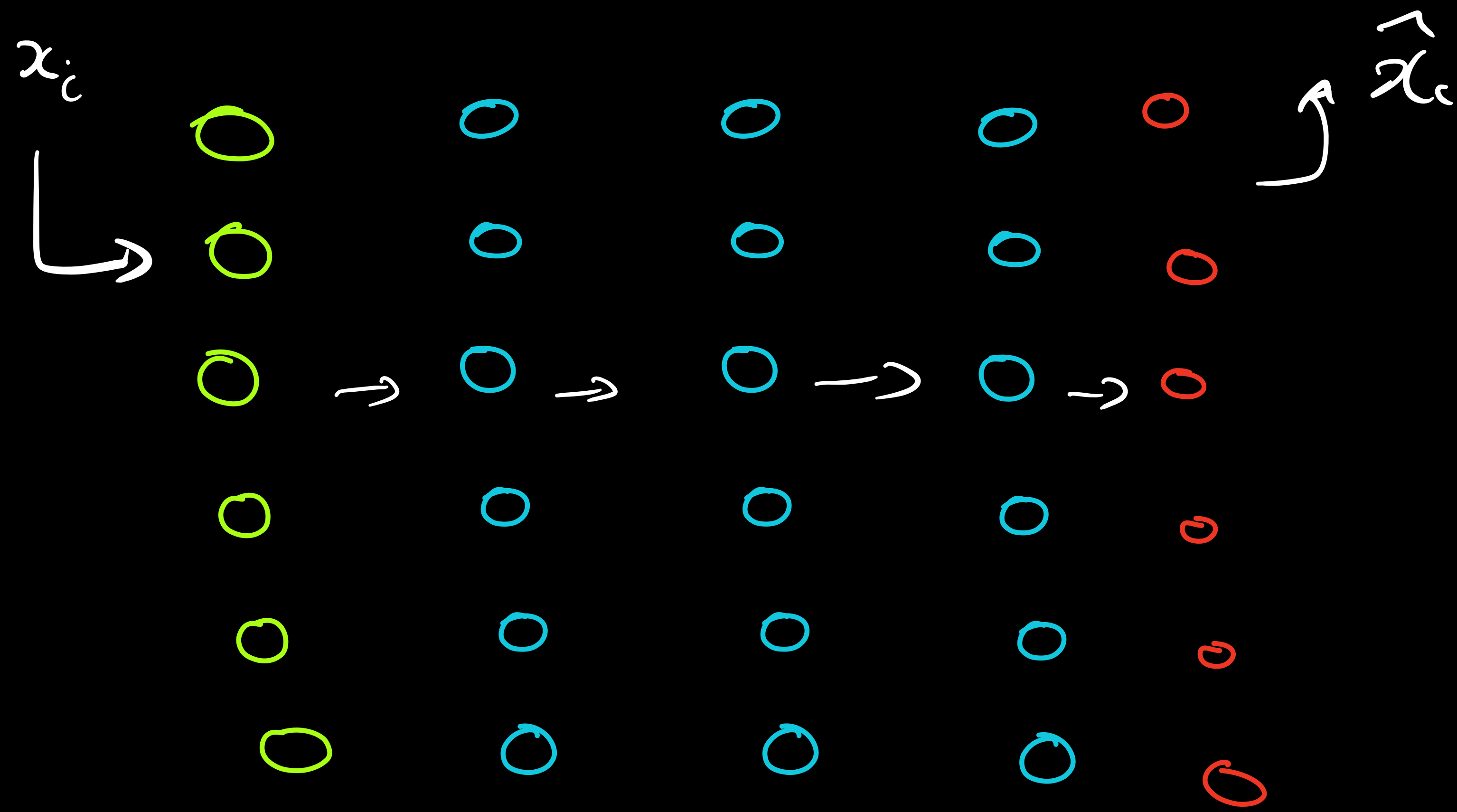
Auto Encoders

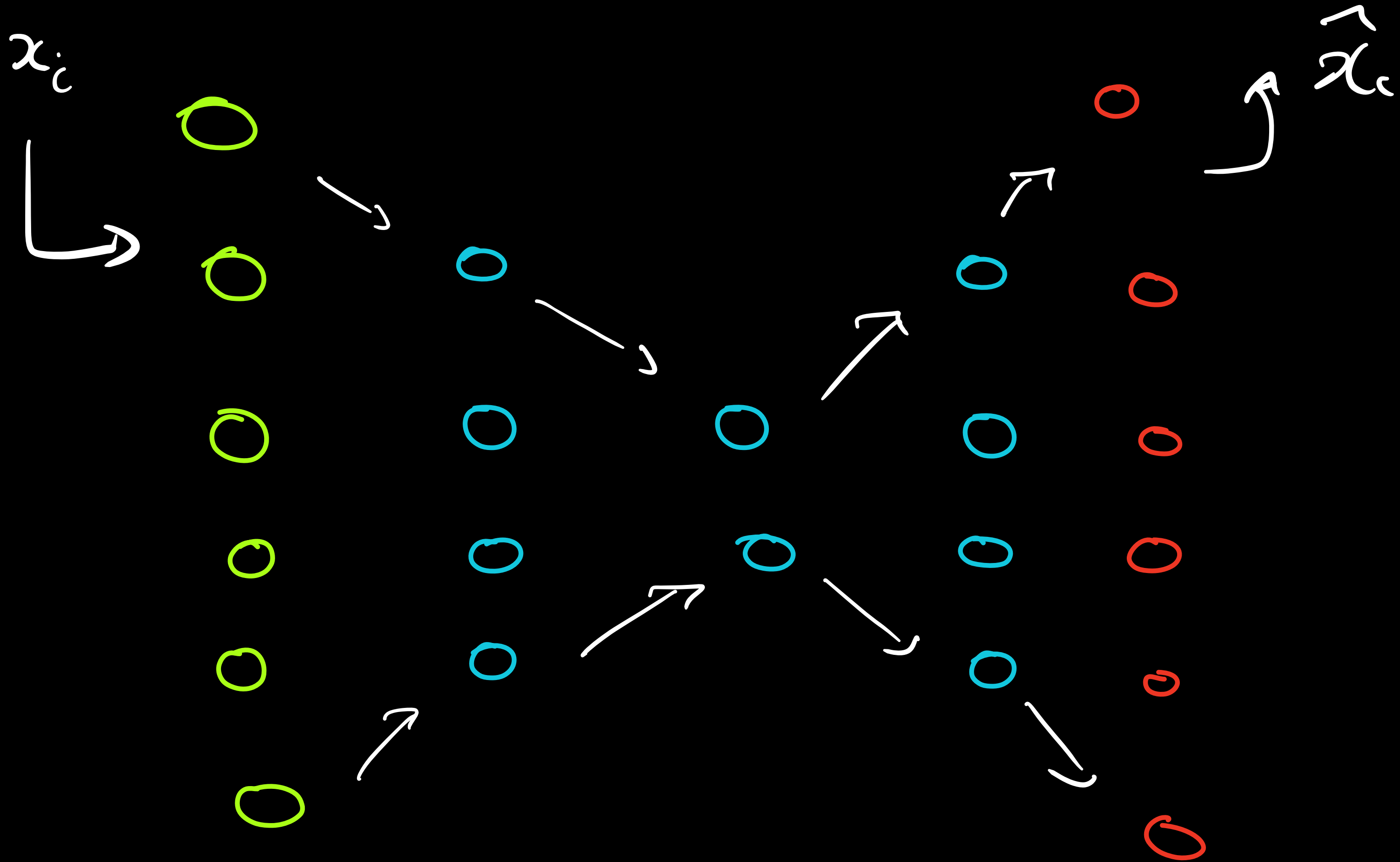
[NN]

Consider this network



I want to put x_i in and get x_i out. Do you think this is possible?

Now let's make a small edit:

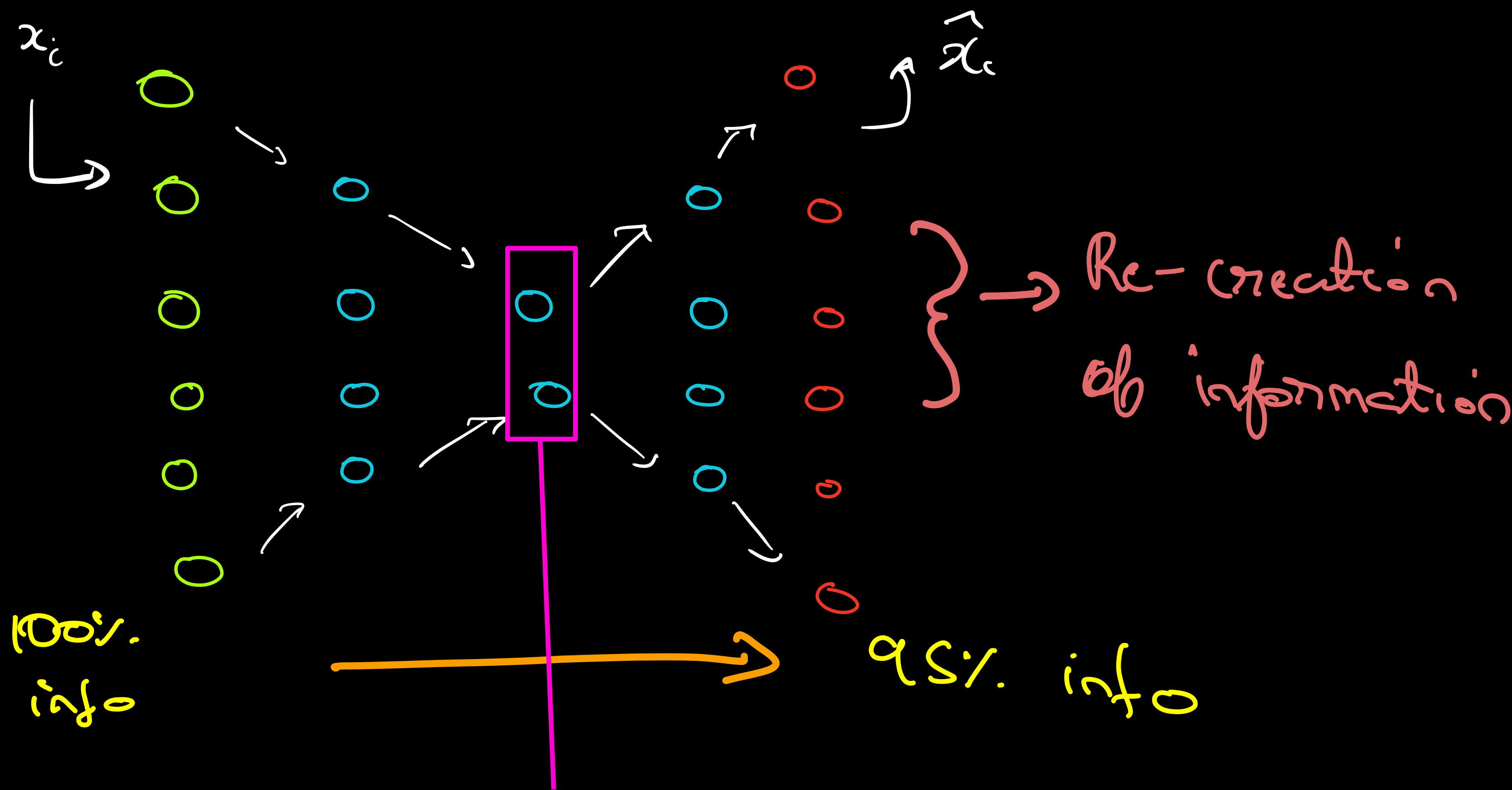


Do you think it's possible now?

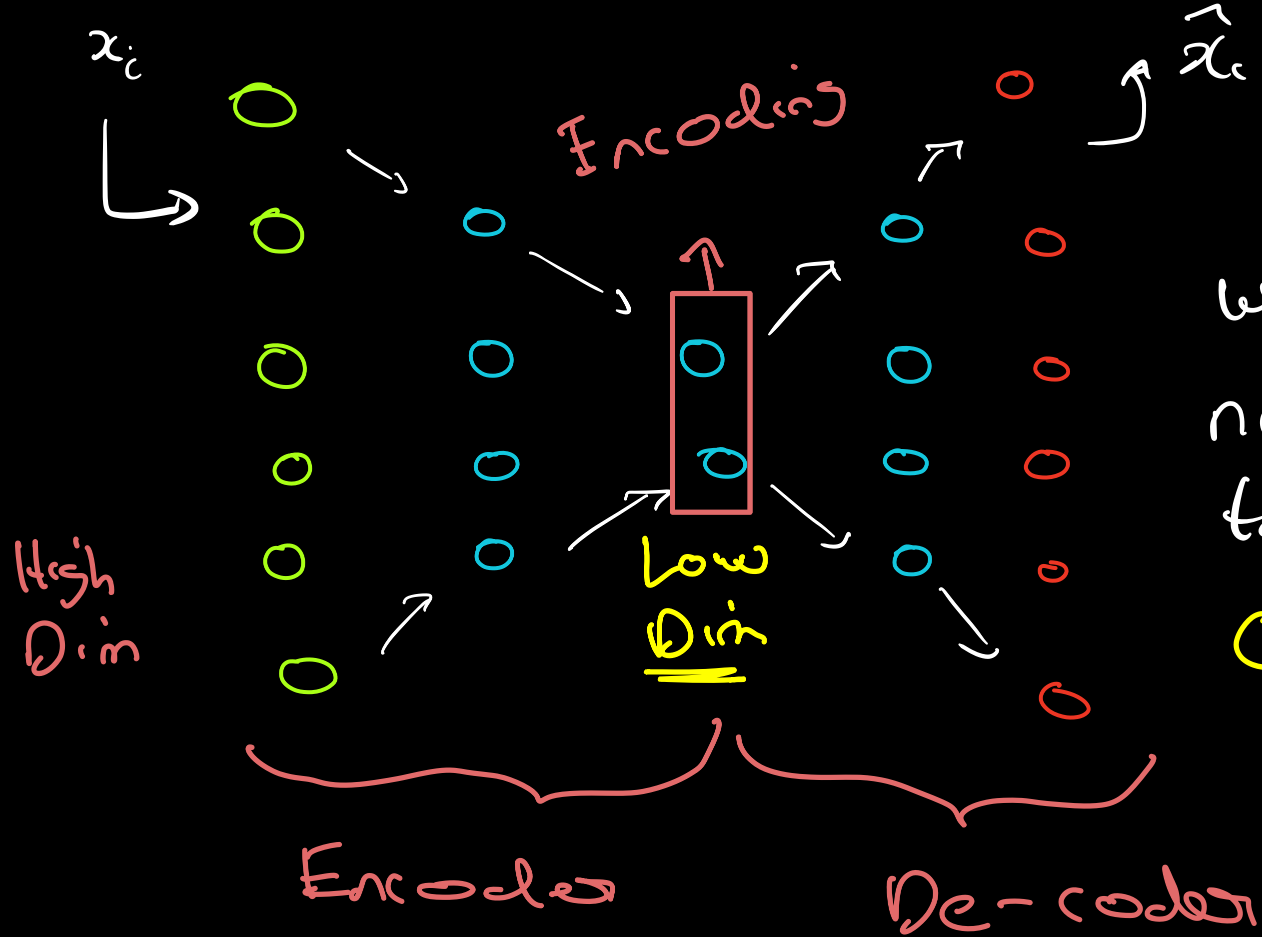
TODO → which activation?

Let's say we train this model. We will get a predicted $\hat{x}_i \sim x_i \ \forall i$.

But why did we do this?



Does this mean that this layer had ~95% of all the information? → Yes!



We can use neural network to perform Compression!!

Encoding ~ Embedding ~ Latent Features

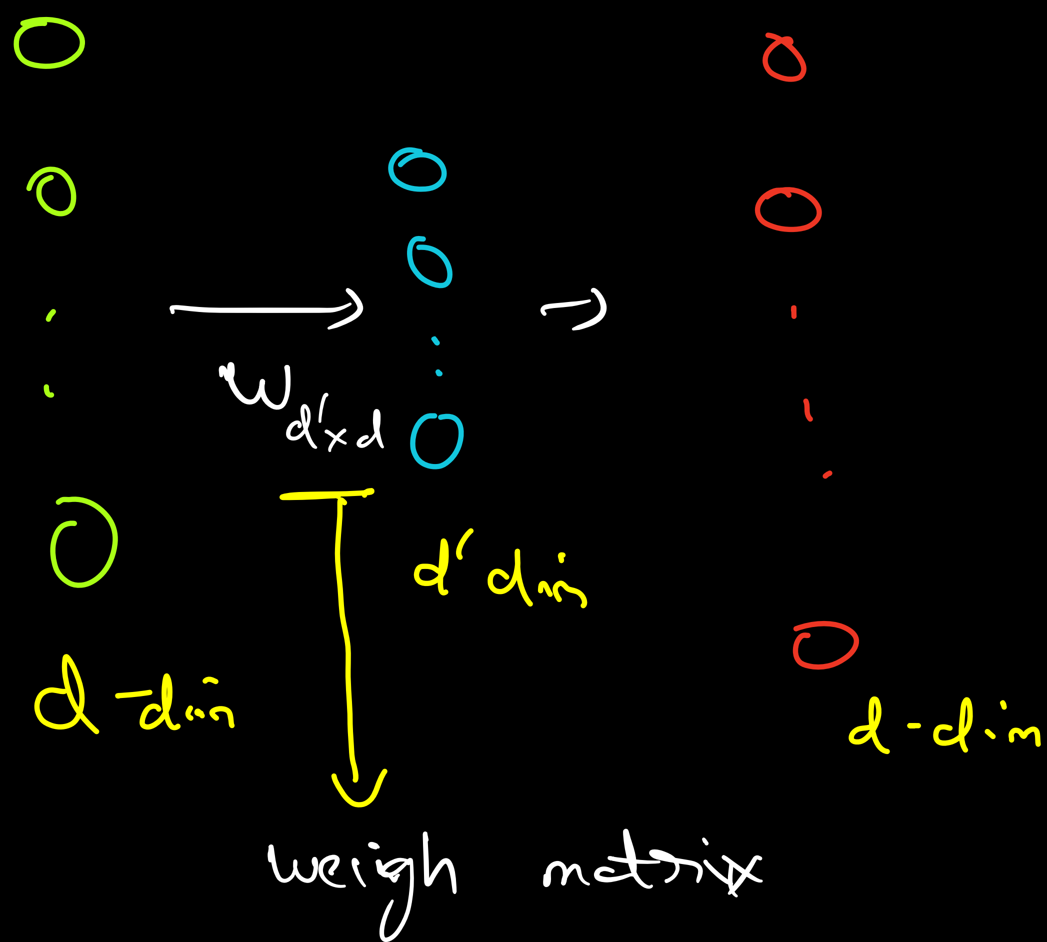
Application of AE

- → Compression / Dim reduction
- → Visualisation
- → De-noising
- → Embeddings [Feature Eng (Automatic)]
  - → Recommender Systems
  - → Clustering
  - → Search , etc.

→ code

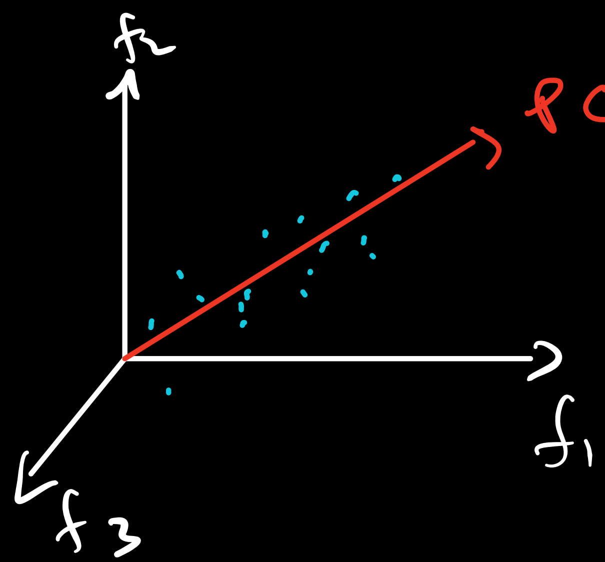
Dim / Red

Can we create a PCA equivalent network?



Note that a principal component is

a linear combination of input axes!



$PC = \alpha f_1 + \beta f_2 + \gamma f_3$

$\downarrow$

linear combination

Hence, if we use a linear activation func^n and make biases $= 0$.

[loosely speaking]

So a single layer Linear auto encoder can simulate PCA.

Also, we may add constraints to make weigh matrix colums $\perp$ (e. Vectors).

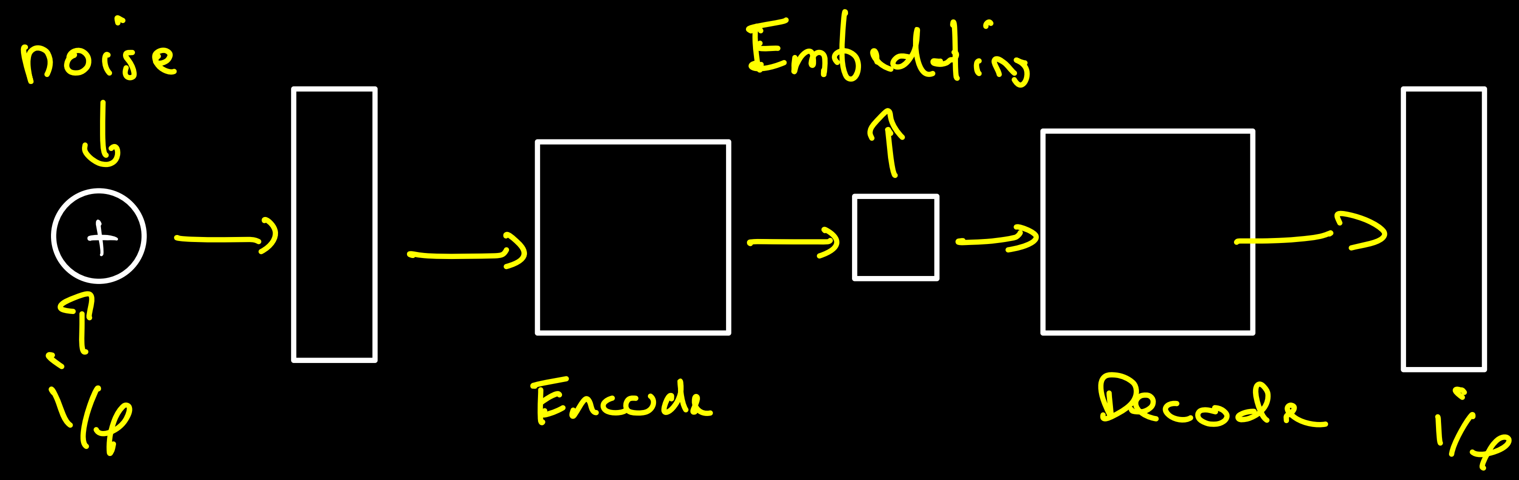
→ There is no need for encoder and decoder to be symmetric. But it is common.

Denoising Auto Encoder (DAE)

→ Sometimes auto encoders can overfit and learn something called an "Identity fcn"

This means that output = input

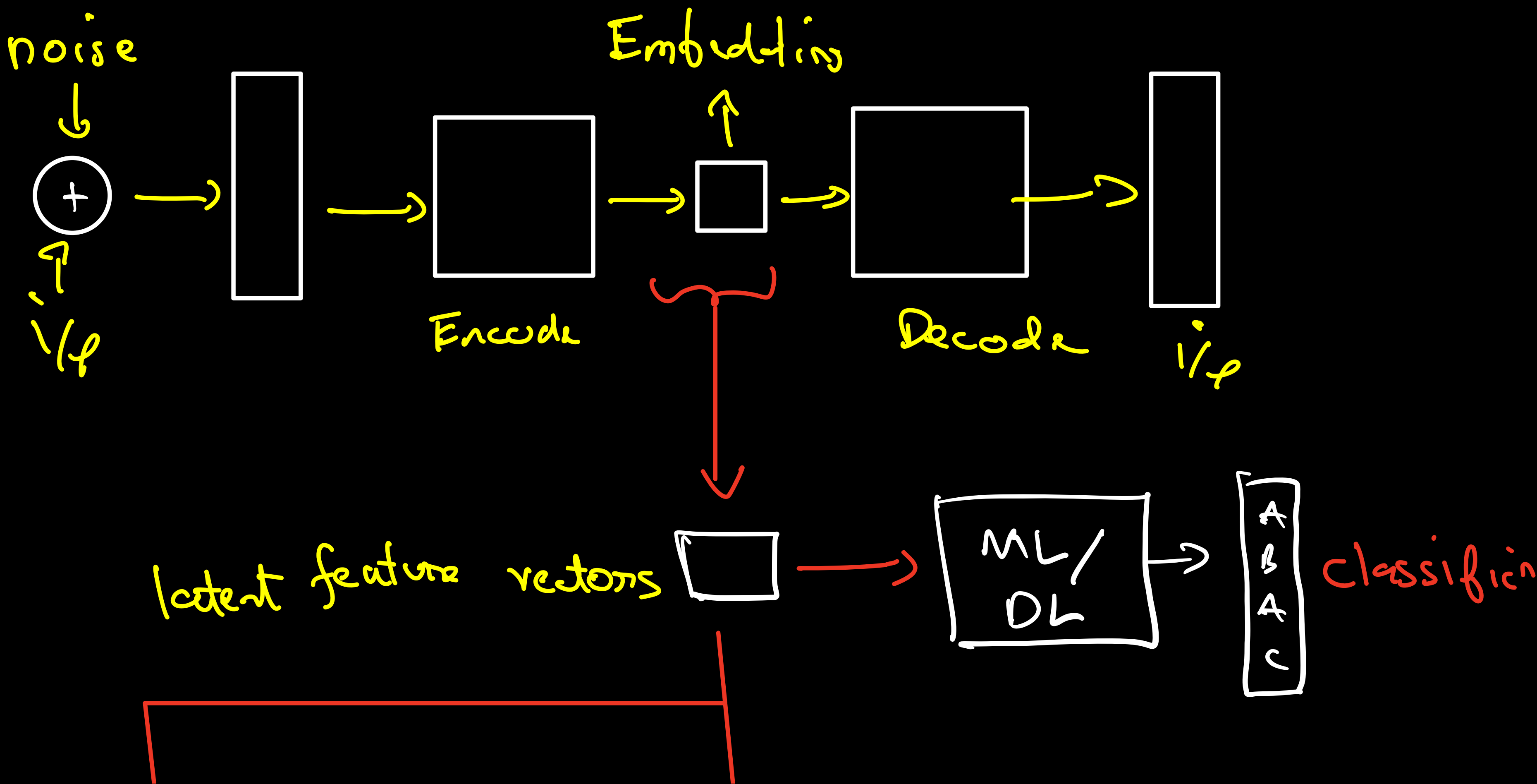
↳ useless.



This reduces overfitting and ensures that the "encoding" is useful.

→ Further → this model is useful in the real world because in reality most images are noisy.

Feature Extraction and Intro to Tranfer Learning



Similarity Compute
- → cosine
- → euclidean
- → etc.

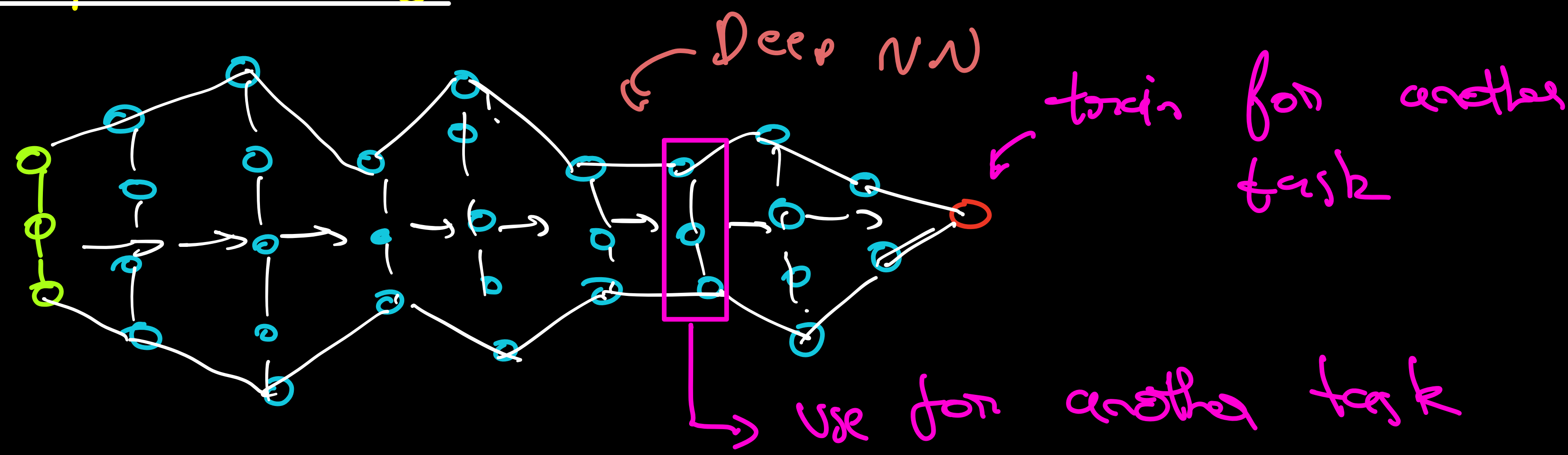
Search   Recommendat^n

$\boxed{\text{Unsupervised learning}}$

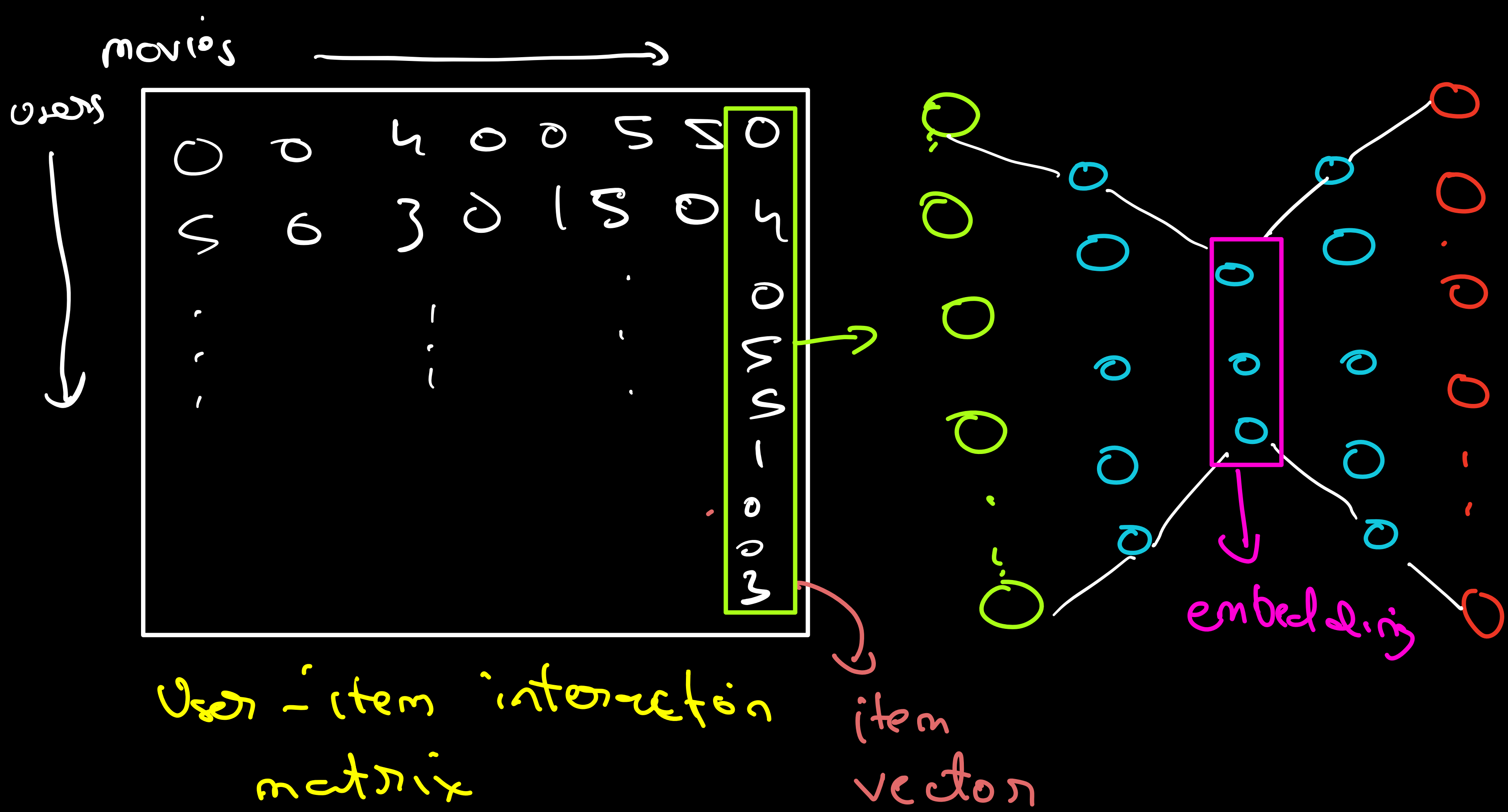
↓

Clustering

Transfer learning



Collaborative Filtering with AE



→ It is advised to only compare the non-zero values in the loss.