

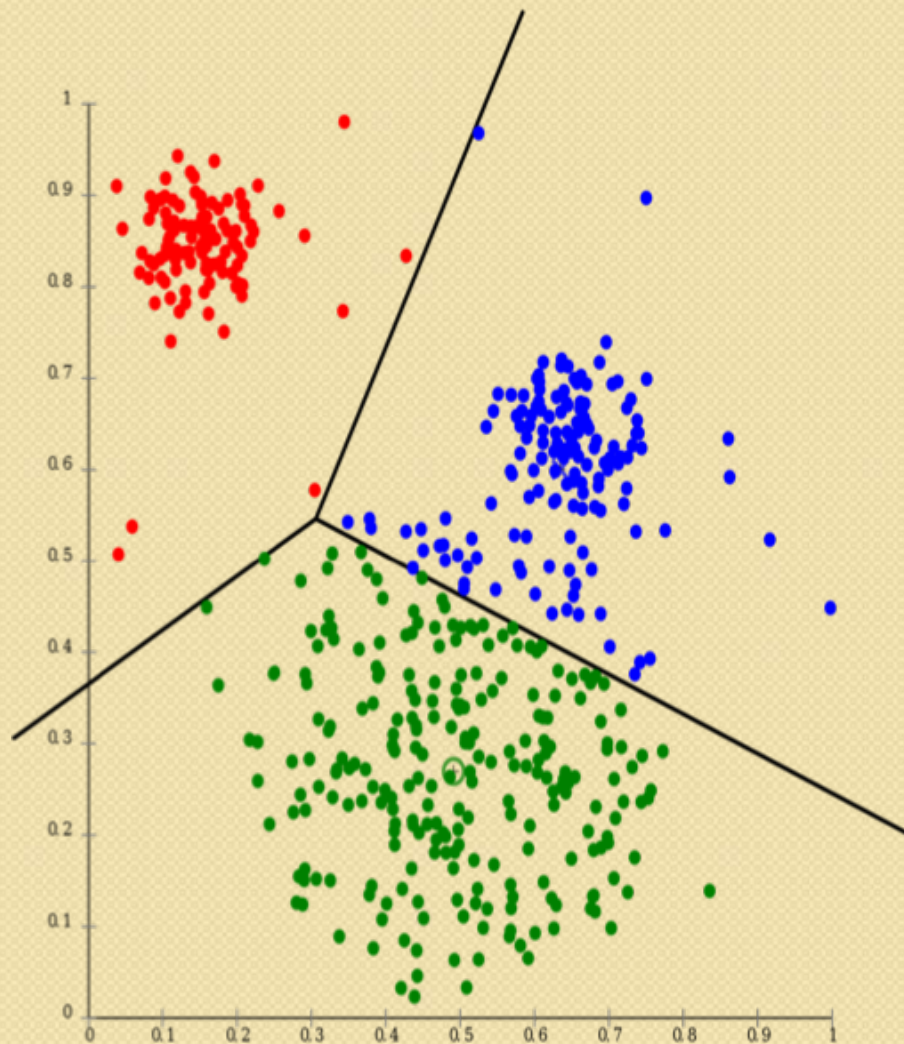
Autocodificadores (Autoencoders)

Germano C. Vasconcelos
Centro de Informática – UFPE

Conteúdo

- – Unsupervised Learning
- – Autoencoder (AE)
- – Convolutional AE
- – Regularization: Sparse
- – Denoising AE
- – Contractive AE
- – Stacked AE
- – Variational AE

Aprendizagem não supervisionada



Aprendizagem não supervisionada

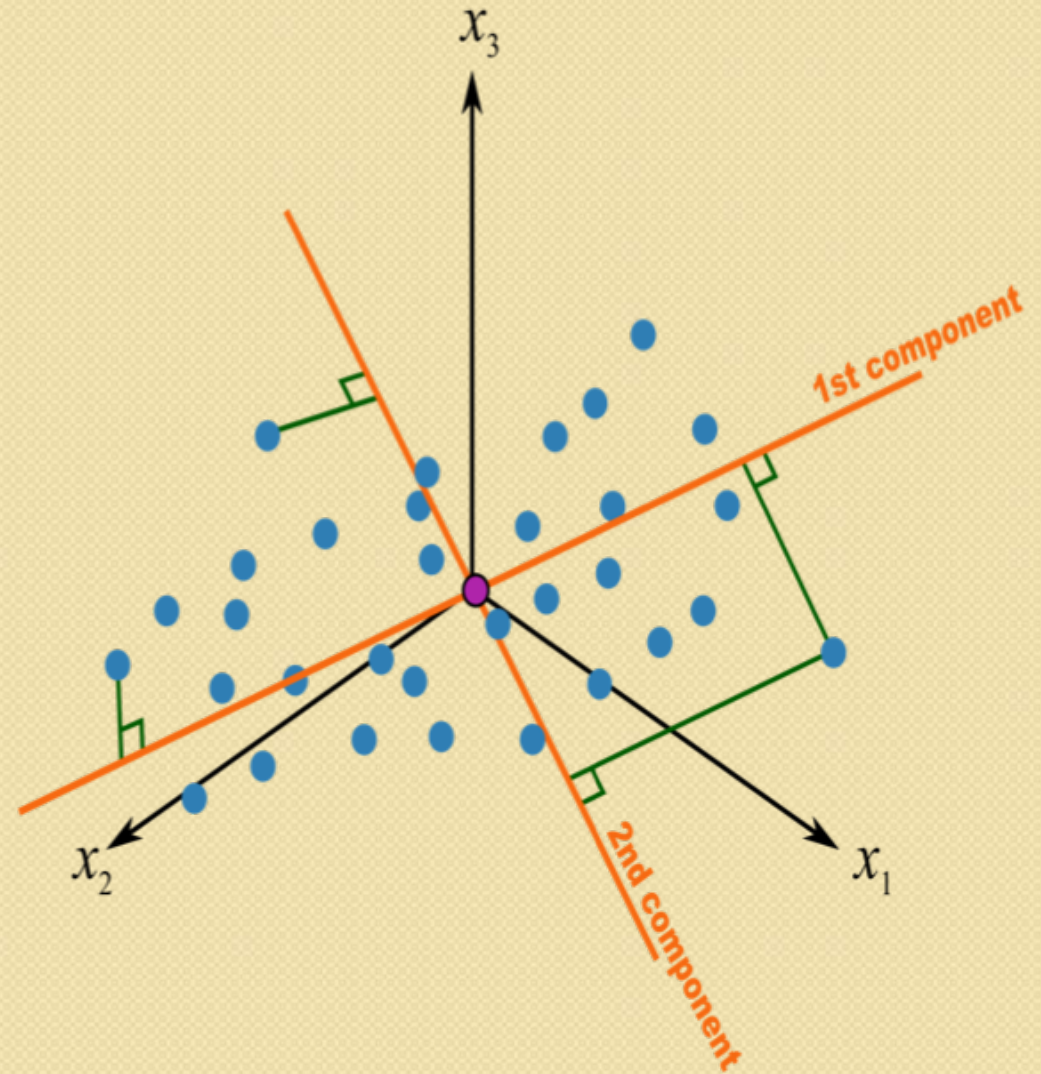
Dados: X (sem rótulos!)

Objetivo: aprender a estrutura dos dados
(aprender correlação entre as estruturas)



PCA – Análise dos Componentes Principais

- Abordagem estatística para compressão de dados e visualização
- Karl Pearson (1901)
- Limitação: componentes lineares.



Redes autocodificadoras (*autoencoders*)

- Podem aprender a estrutura do conjunto de dados de forma **não supervisionada**
 - Aprendizado de representações mais concisas
 - Características identificadas são úteis para uso posterior em aprendizagem supervisionada

Transformações realizadas

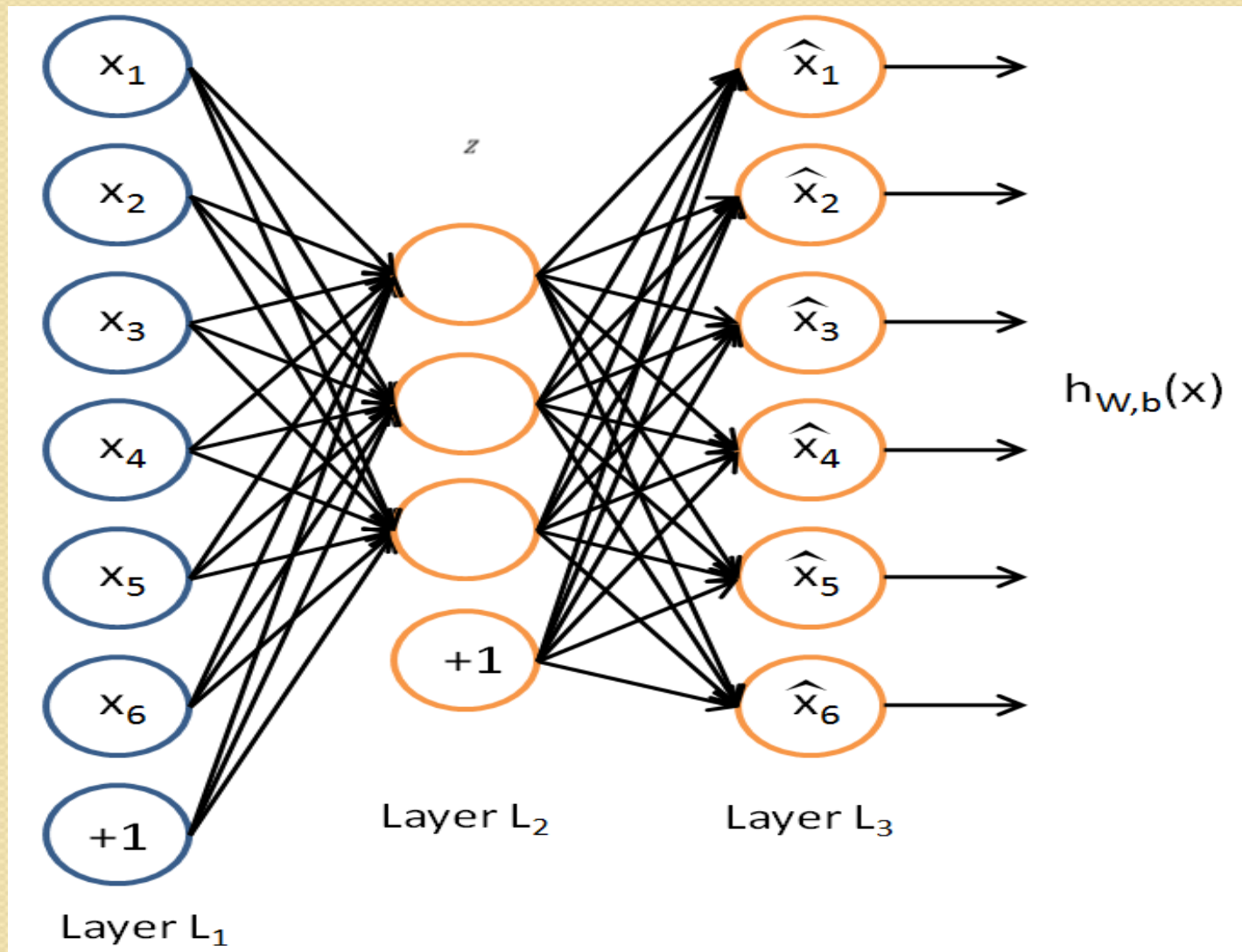
- Transformações são aplicadas na entrada de acordo com dois tipos de funções.
 - **Função de extração de características** (*encoder*) mapeia o conjunto de treinamento para uma **representação latente**.

$$h : \mathbb{R}^D \rightarrow \mathbb{R}^K$$

- **Função de reconstrução** (*decoder*) mapeia a representação produzida por h de volta para o espaço original.

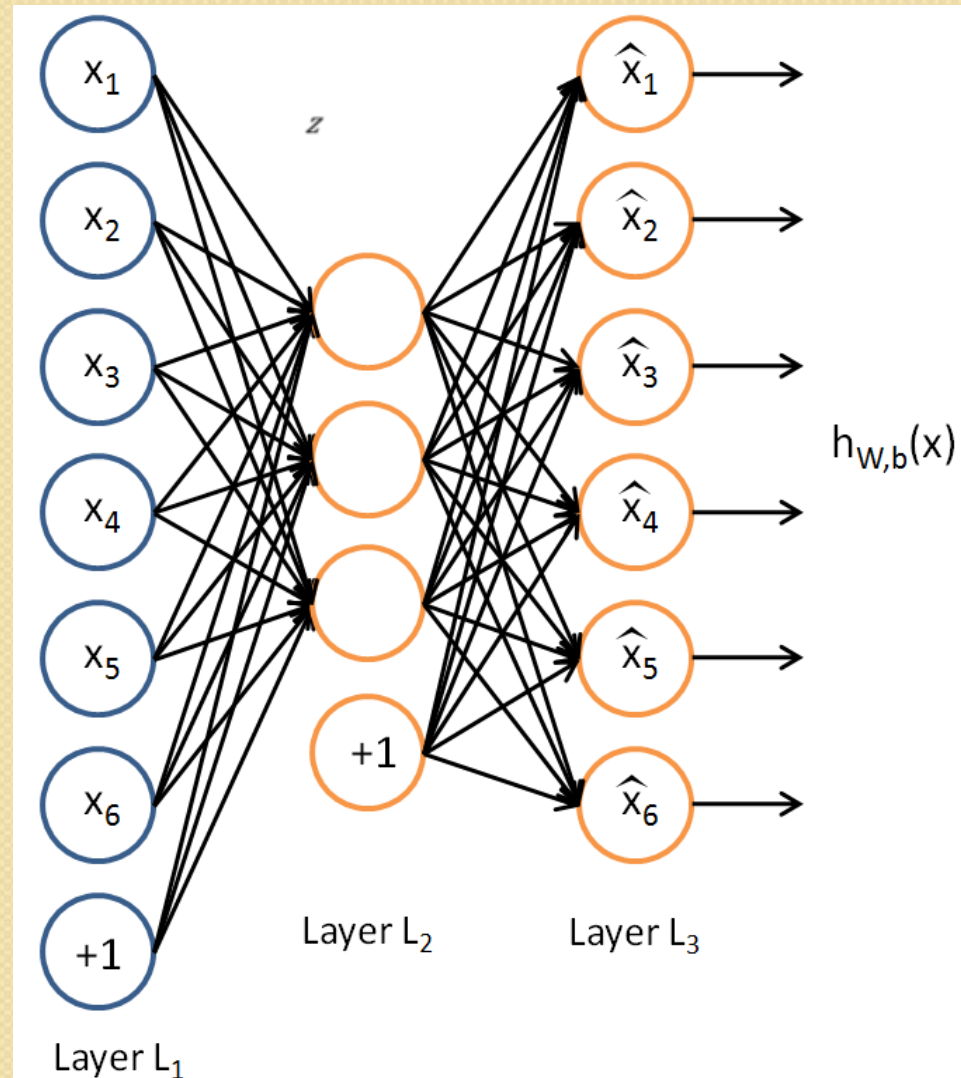
$$r : \mathbb{R}^K \rightarrow \mathbb{R}^D$$

Autoencoder Tradicional



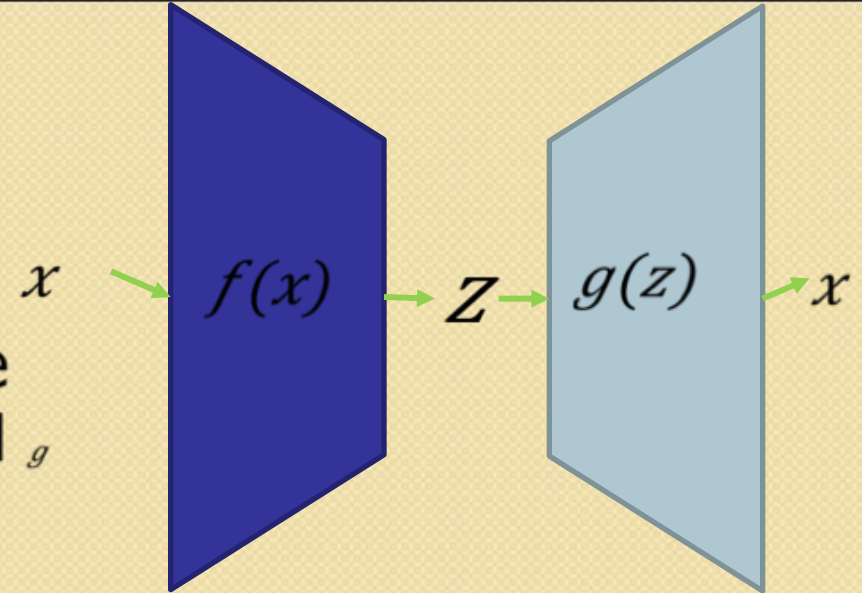
Autoencoder Tradicional (AE)

- Diferente de PCA, temos não-linearidade
- É mostrado que um AE sem funções de ativação produz capacidade semelhante a PCA



Ideia Simples

- Given data x (no labels) we would like to learn the functions f (encoder) and g (decoder) where:



and

$$f(x) = s(wx + b) = z$$

$$g(z) = s(w'z + b') = x$$

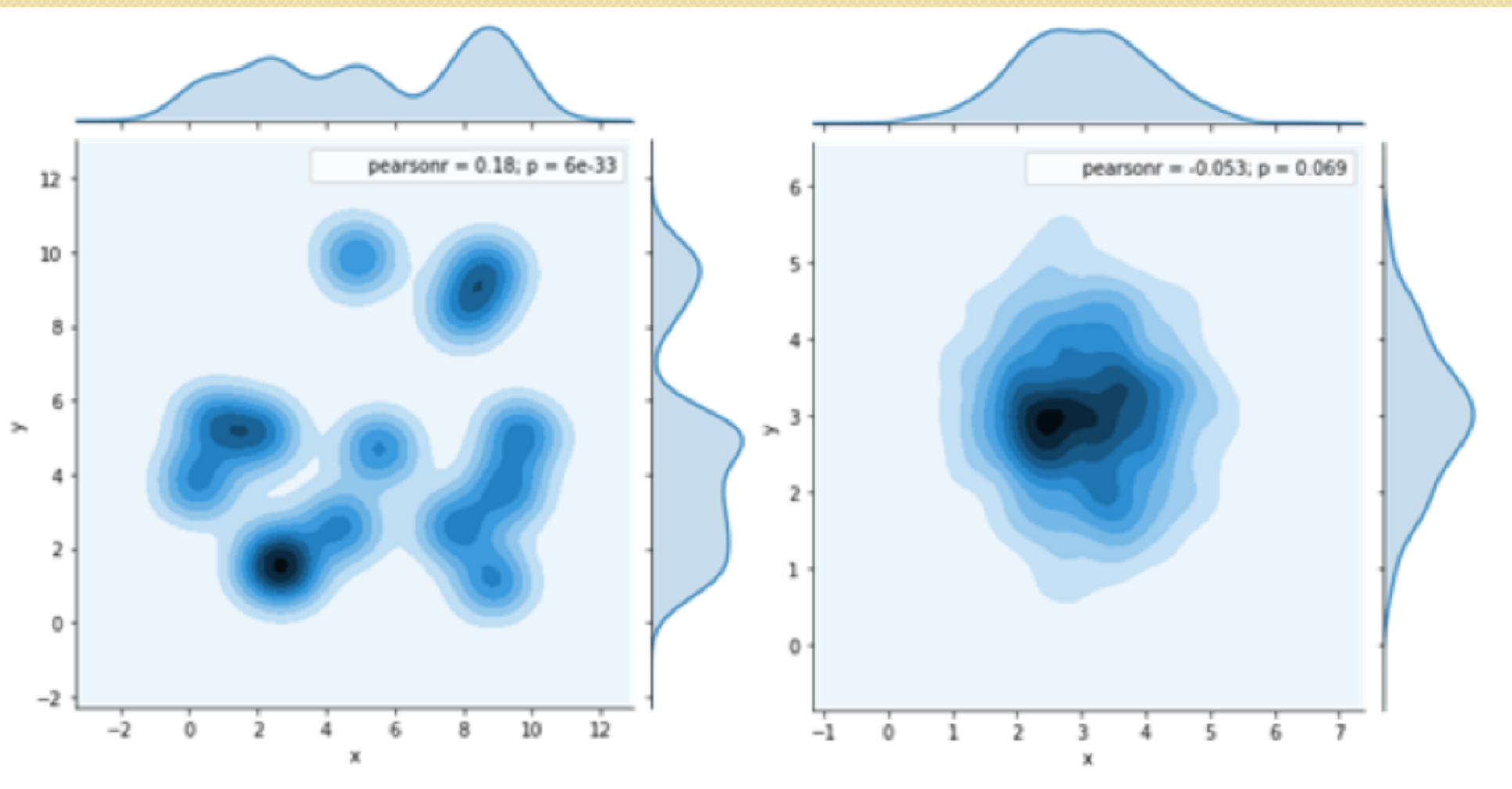
$$\text{s.t. } h(x) = g(f(x)) = x$$

(z is some **latent** representation or **code** and s is a non-linearity such as the sigmoid)

($\hat{x}$ is x 's reconstruction)

where h is an **approximation** of the identity function.

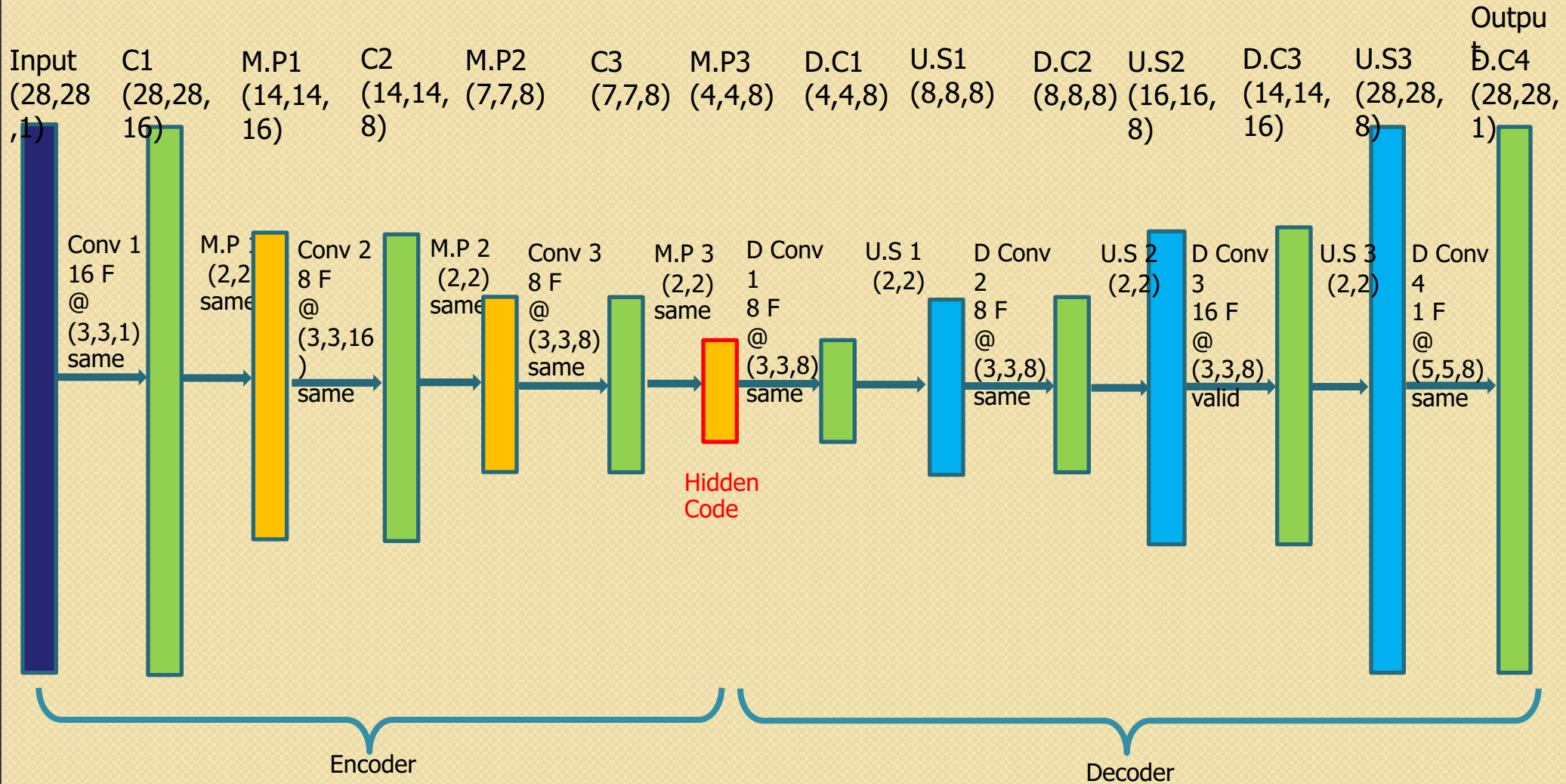
Representação



Convolutional AE

* Input values are normalized

* All of the conv layers activation functions are relu except for the last conv which is sigm



Treinamento

13

- Autoencoder é treinado para reproduzir na sua saída a própria entrada.
- Objetivo do treinamento é definir parâmetros em h e r tal que o **erro de reconstrução** seja minimizado.
- Pode ser treinada usando backprop, SGD
 - Substituindo valores-alvo pela própria entrada $\mathbf{x}$.

$$L(\mathbf{X}) = \sum_{\mathbf{x}^{(i)} \in \mathbf{X}} \ell(\mathbf{x}^{(i)}, r(h(\mathbf{x}^{(i)})))$$

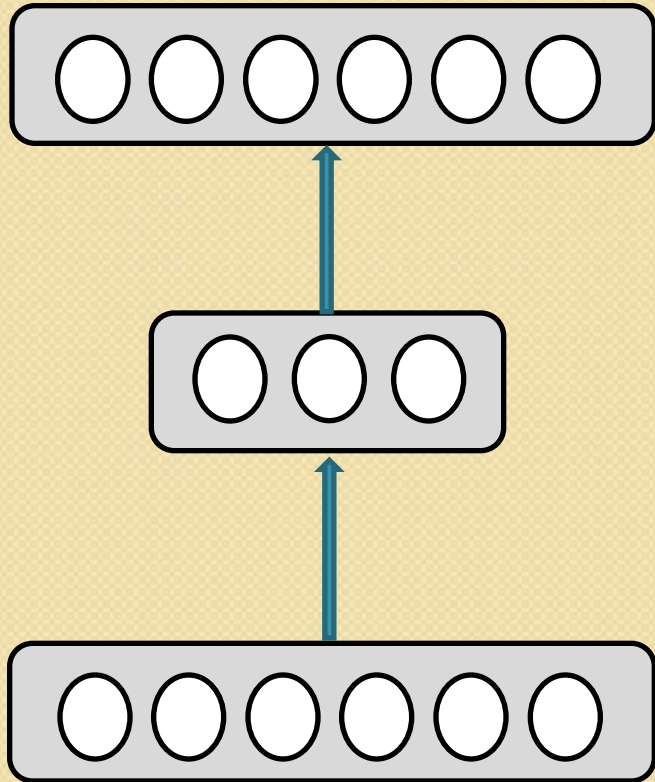
Subcompletos x Sobrecompletos (Undercomplete vs Overcomplete)

14

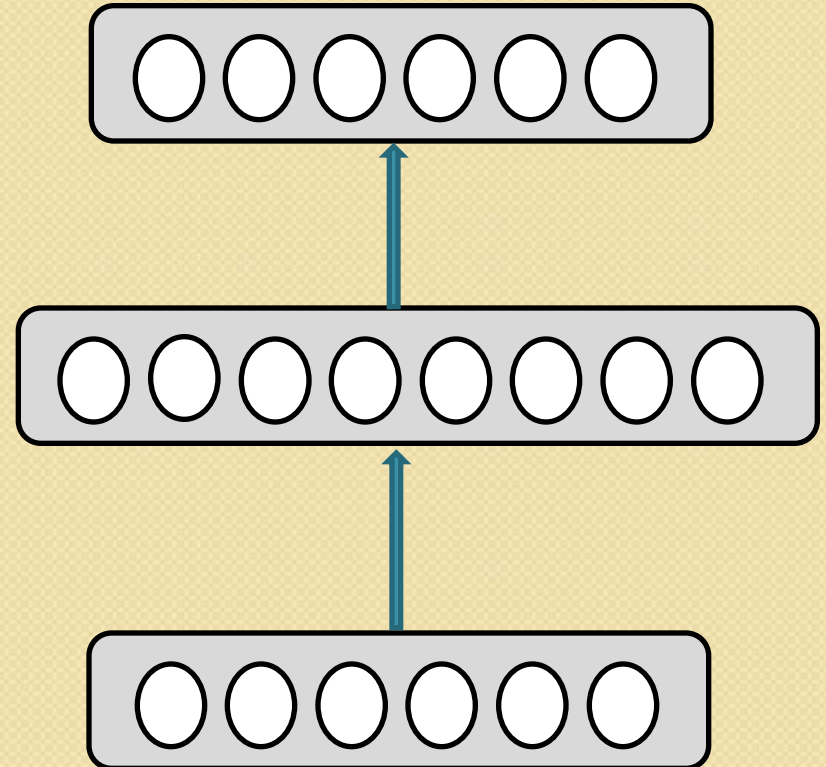
- Representação latente em uma autocodificadora tem dimensão K :
 - $K < D \rightarrow$ undercomplete autoencoder;
 - $K > D \rightarrow$ overcomplete autoencoder.
- A escolha de K determina
 1. Quantidade de unidades da camada intermediária central
 2. Tipo de informação que a autocodificadora pode aprender da distribuição de entrada.

Undercomplete AE VS Overcomplete AE

Subcompleto -
Undercomplete

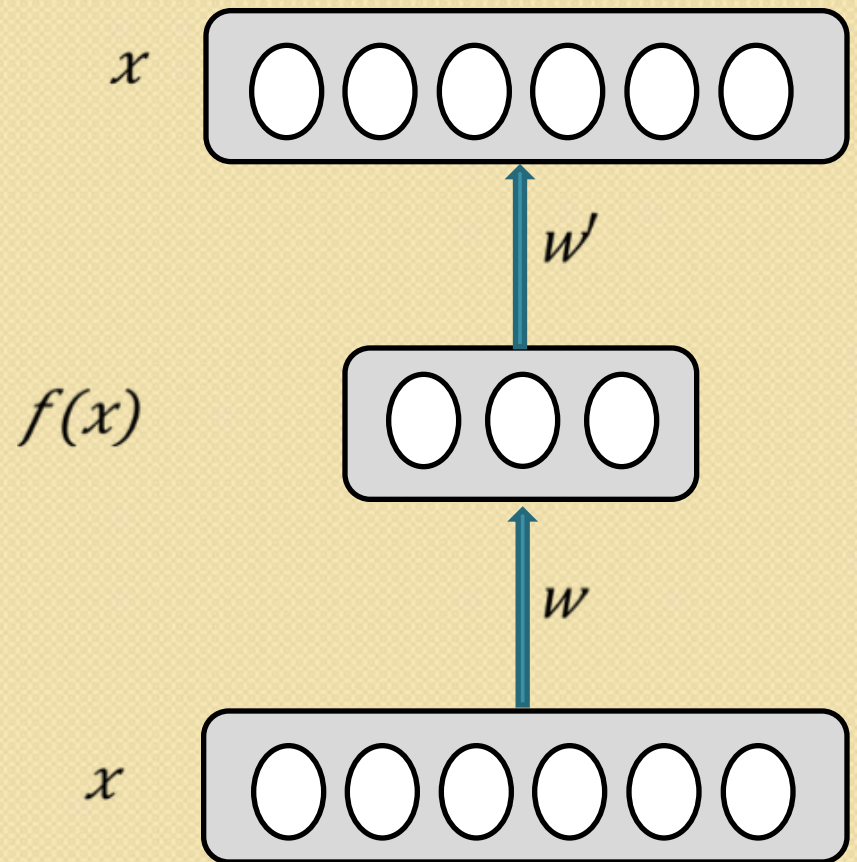


Sobrecompleto -
Overcomplete



Subcompleto AE

- Camada escondida é **Subcompleta** se menor que a de entrada
 - ☐ Comprime a entrada
 - ☐ Comprime bem apenas para conjunto de treinamento
- Nós escondidos formam
 - ☐ Boas características para distribuição de treinamento
 - ☐ Ruins para outras entradas



Caso $K < D$ (*bottleneck*)



17

- Objetivo: aprender **representação compacta** do conjunto de dados
- Exemplo:
 - Considere um conjunto de dados de treinamento T exemplos. Em cada exemplo, o esquema de codificação é de *um bit por estado*.
 - Autocodificadora deveria codificar os exemplos por meio de uma codificação binária, com $\log_2 T$ bits por exemplo.

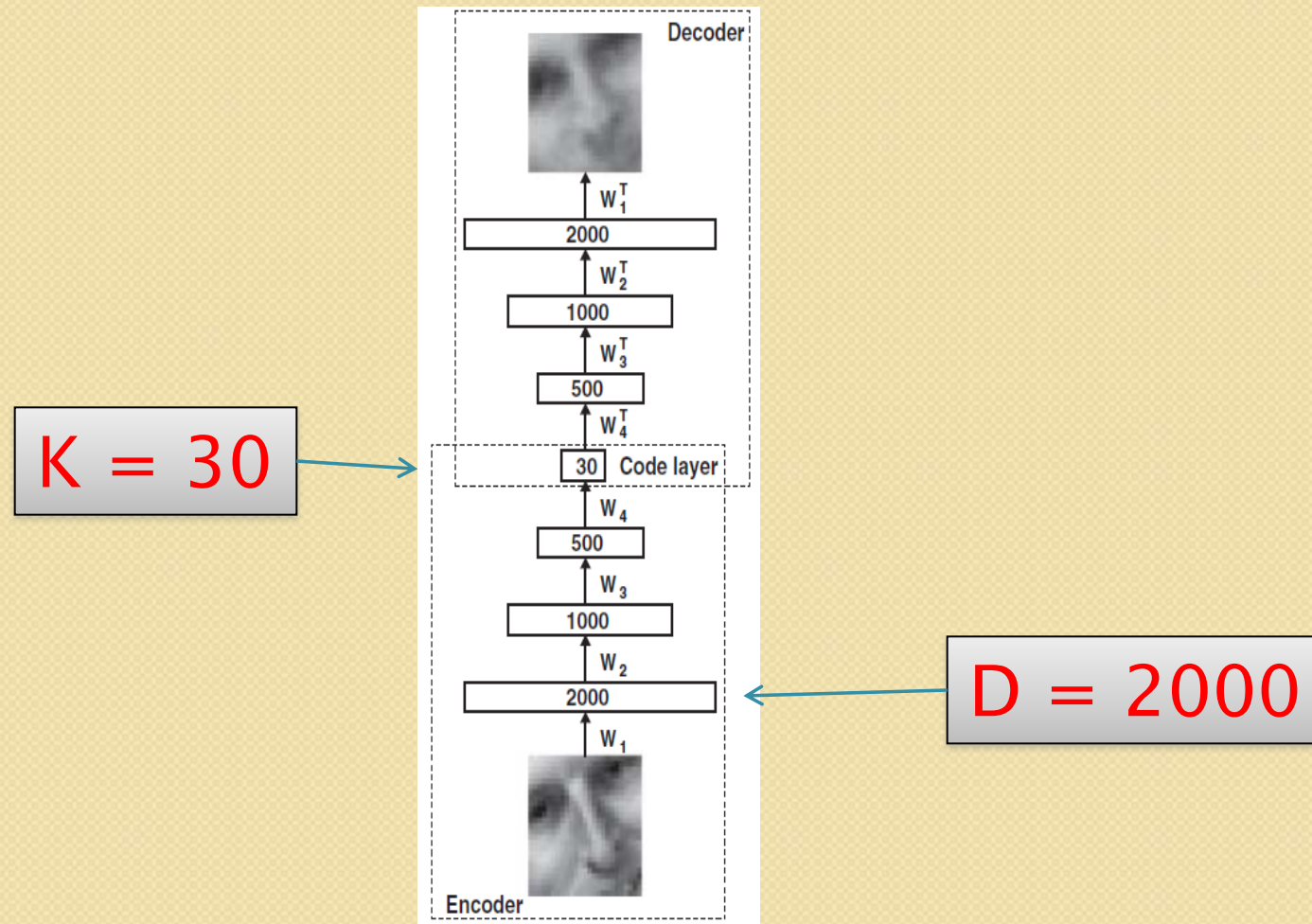
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

one-hot encoding

Caso $K < D$ (*bottleneck*)



18



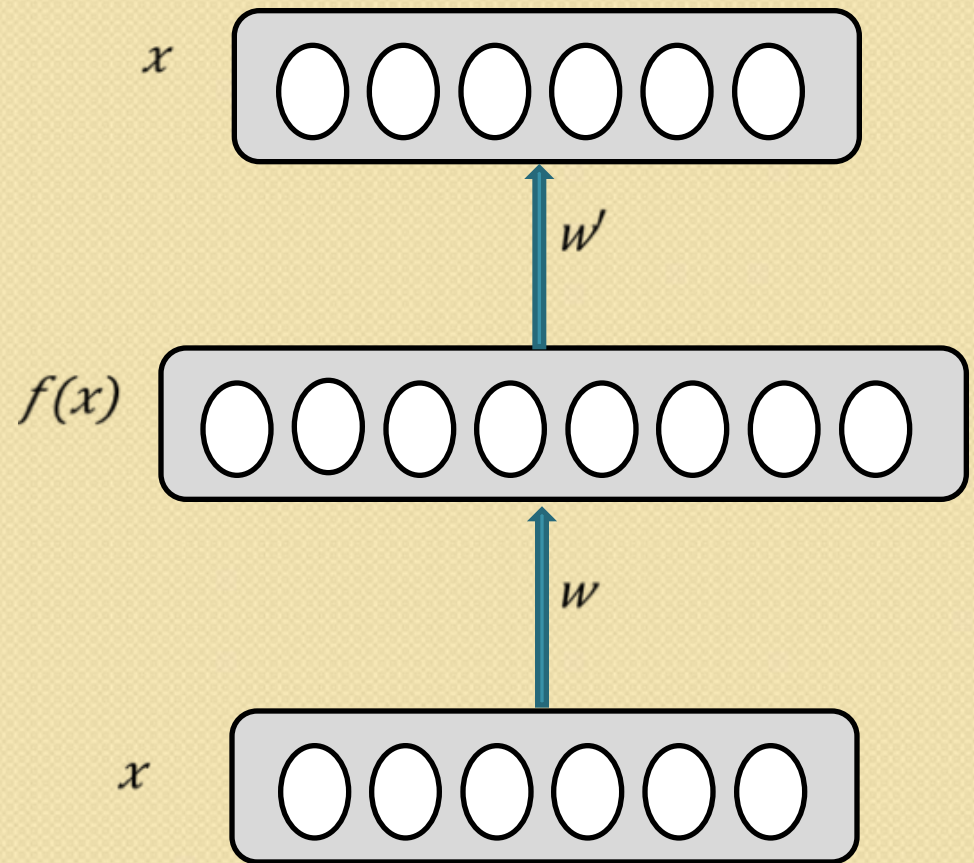
Caso $K > D$

19

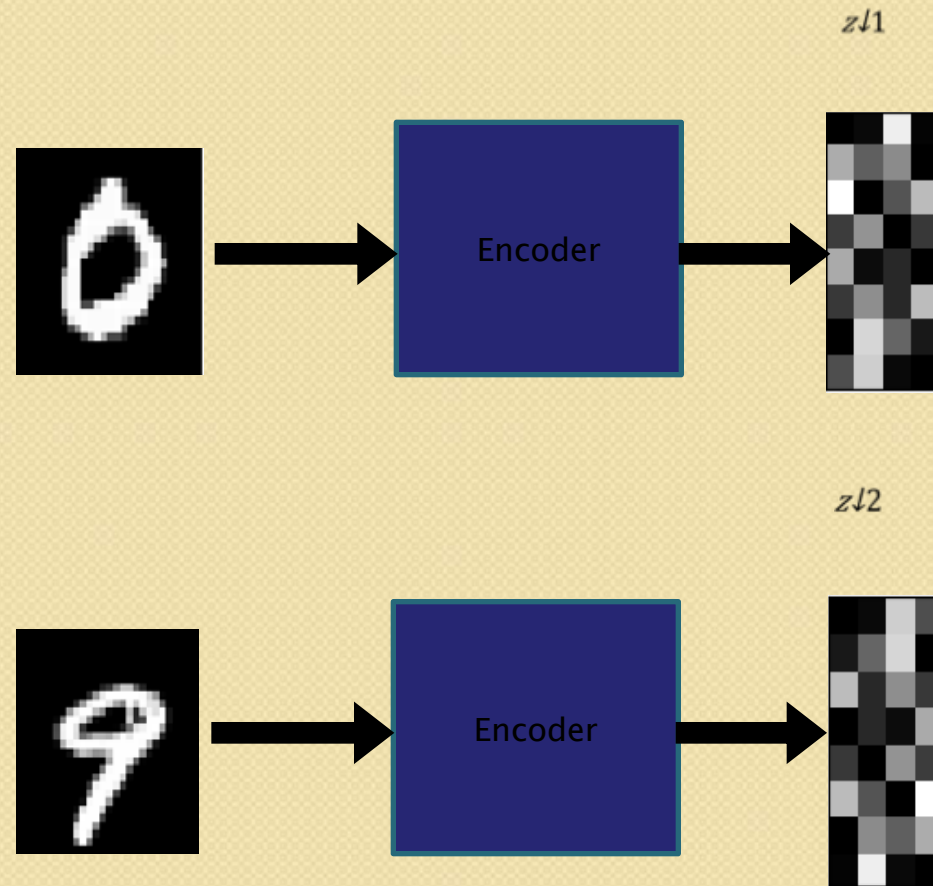
- Objetivo: encontrar características da entrada para posteriormente serem apresentadas a um classificador linear (SVM, k-NN, etc).
- Problema potencial no treinamento: autocodificadora apenas copia os D bits da entrada para D unidades na camada central.
 - aprende a função identidade (i.e., $f(x) = x$).
 - deixando de usar $K-D$ unidades nessa camada.

Sobrecompleto AE

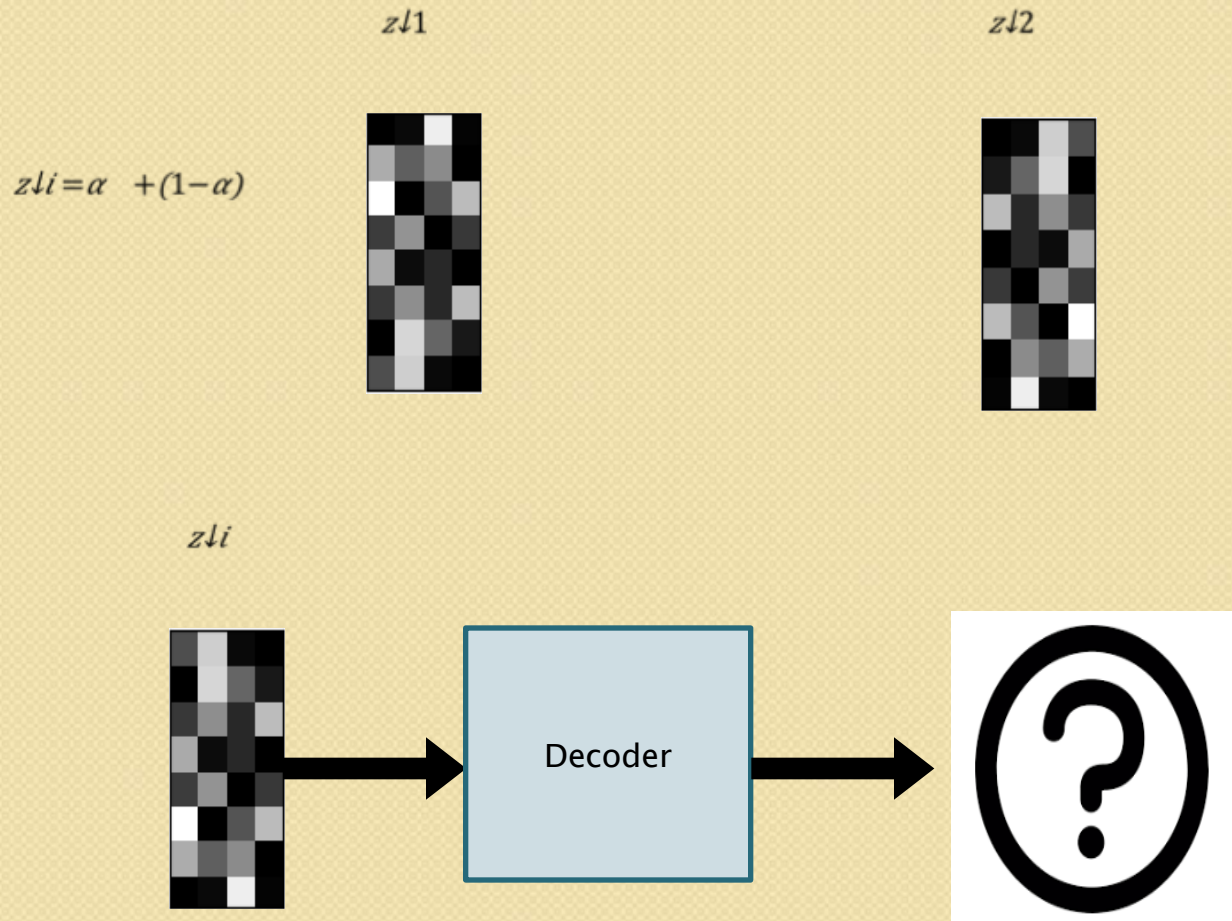
- Camada escondida é **Sobrecompleta** se maior que a de entrada
 - ❑ Sem compressão na camada intermediária.
 - ❑ Cada unidade escondida pode criar uma cópia diferente do componente de entrada
- Adicionar dimensões é útil para treinar um classificador linear.
- Um código em dimensão maior ajuda a modelar distribuições mais complexas
- Não há garantias que unidades intermediárias consigam extrair boas características



Simple interpolation no espaço latente – keras



Simple interpolation no espaço latente – keras



Convolutional AE – Keras example results

- 50 epochs.
- 88% accuracy on validation set.



Autoencoder com filtragem de ruído

24

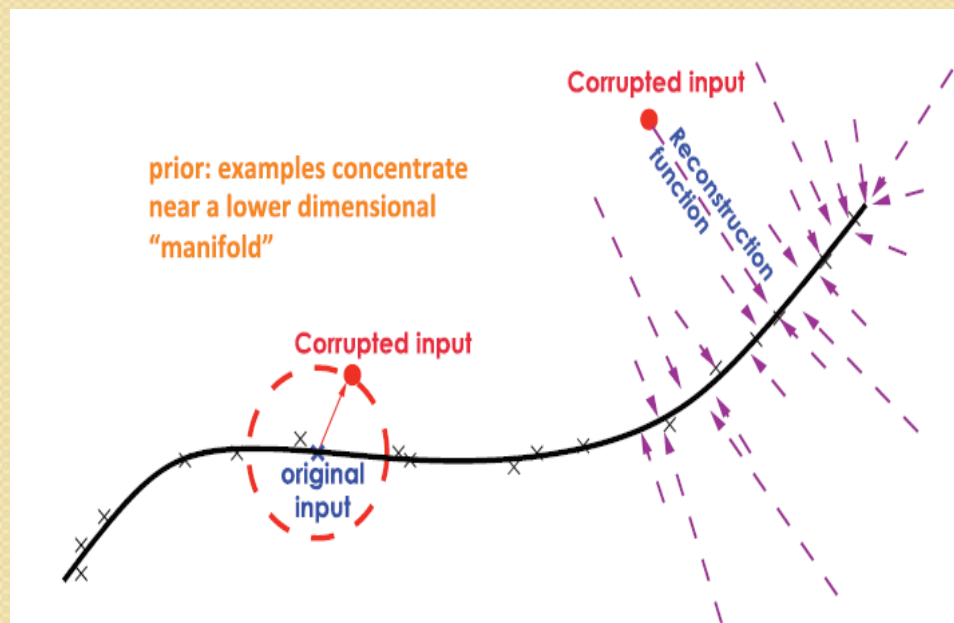
(*denoising autoencoder*)

- Ideia básica: representação aprendida seja **robusta a ruídos** nos dados de entrada.
- Aplicar processo probabilístico em cada exemplo de treinamento **x** antes de apresentá-lo à rede.
- Alternativas
 - a) com probabilidade p , atribuir zero a cada componente de **x**.
 - b) perturbar cada componente de **x** por meio de um ruído gaussiano aditivo.

Autoencoder com filtragem de ruído

(*denoising autoencoder*)

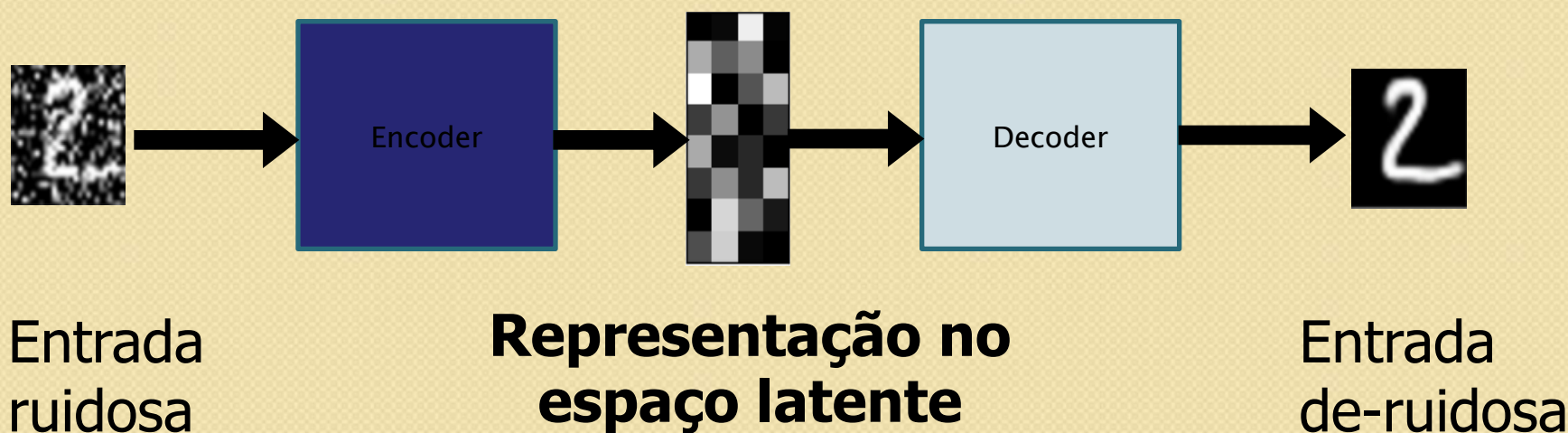
25



Denoising Autoencoders

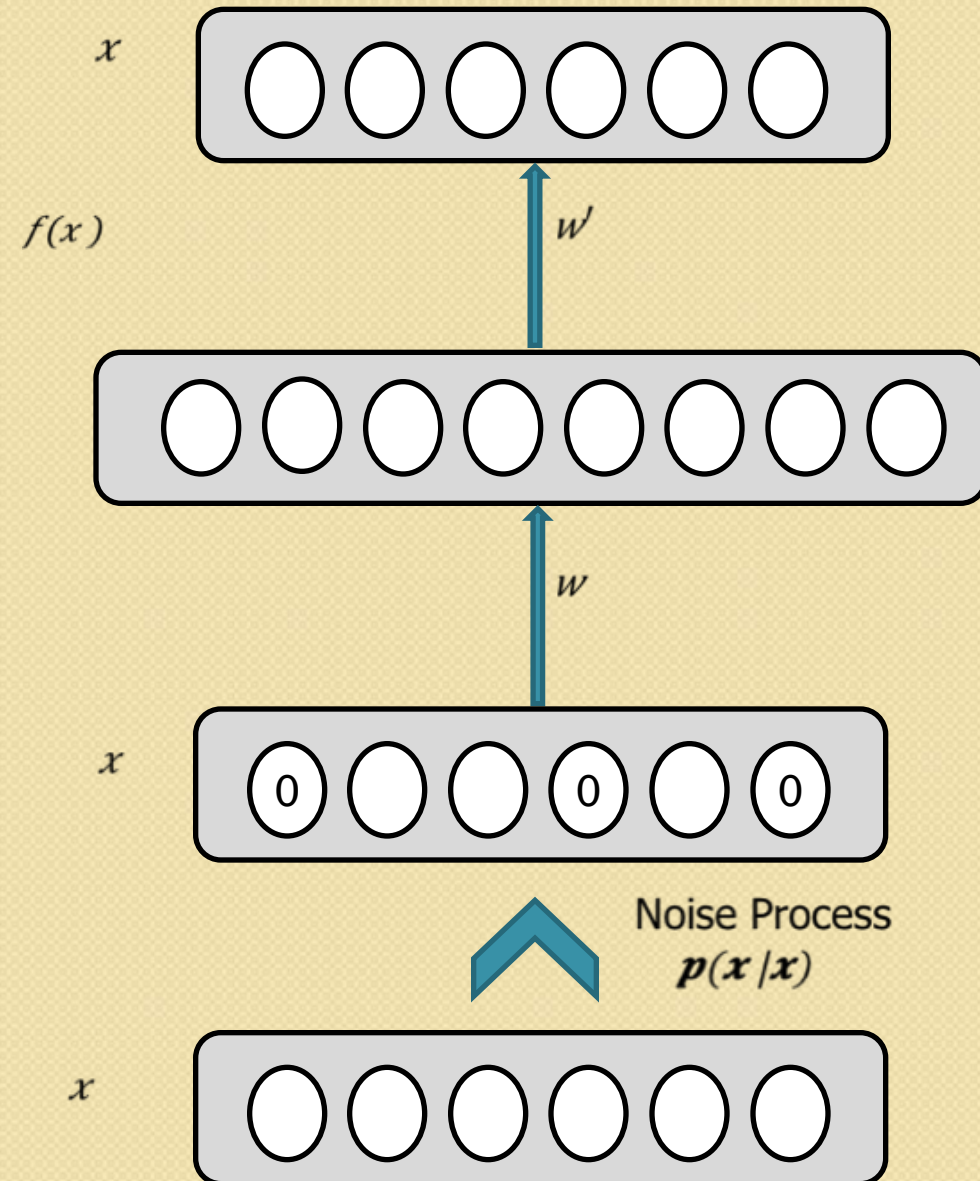
- Codificar a entrada sem “copiar” a função identidade
- Desfazer o efeito do processo estocástico de corrupção aplicado na entrada.

Modelo mais robusto a ruídos



Denoising Autoencoders

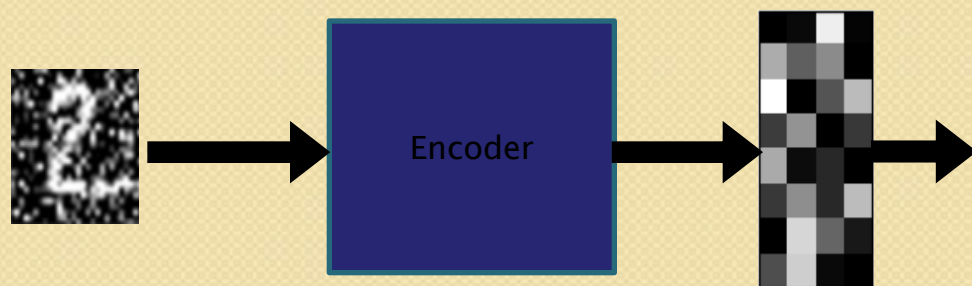
- Reconstruction $\hat{x}$ computed from the corrupted input x .
- Loss function compares $\hat{x}$ reconstruction with the noiseless x .
- ❖ The autoencoder cannot fully trust each feature of x independently so it must learn the correlations of x 's features.
- ❖ Based on those relations we can predict a more 'not prone to changes' model.
- We are forcing the hidden layer to learn a generalized structure of the data.



Denoising Autoencoders

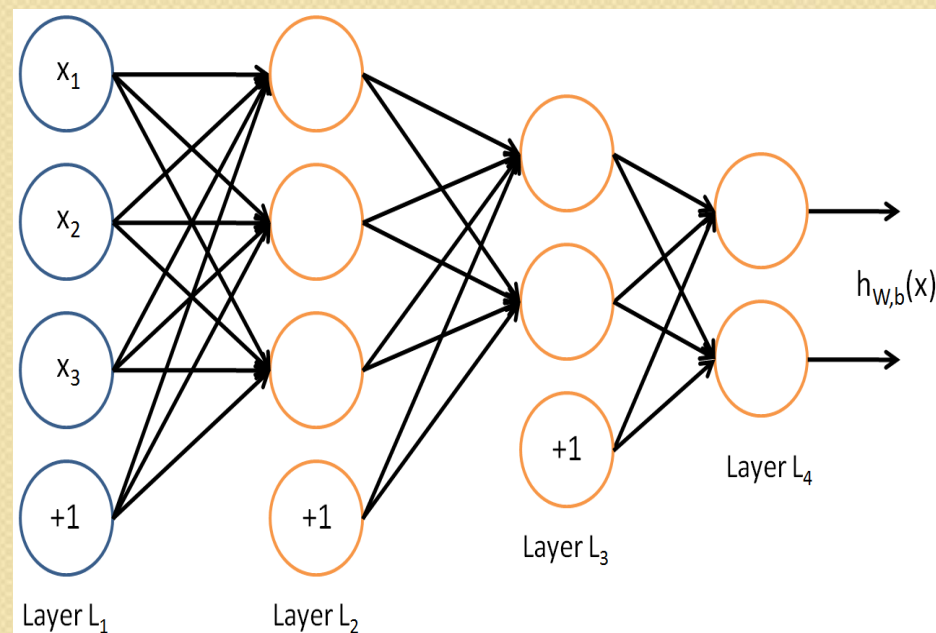
Caso de uso:

- Extrair representações robustas para um classificador



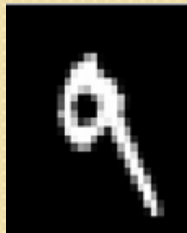
Entrada
ruidosa

**Representação no
espaço latente**



Processo – Denoising Autoencoders

Taken some
input x



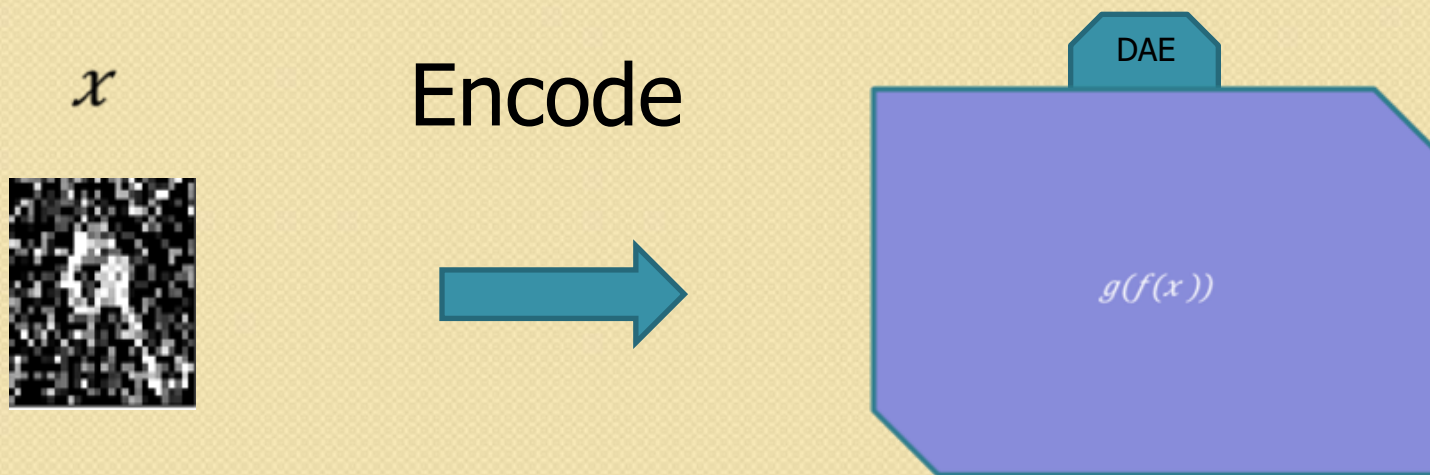
Apply
Noise



x'

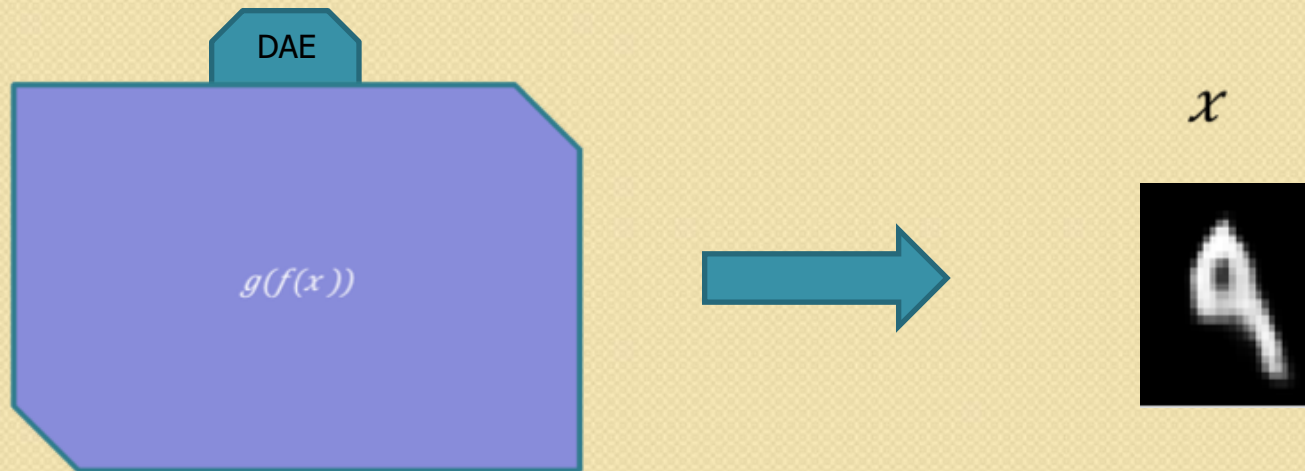


Processo Denoising Autoencoders

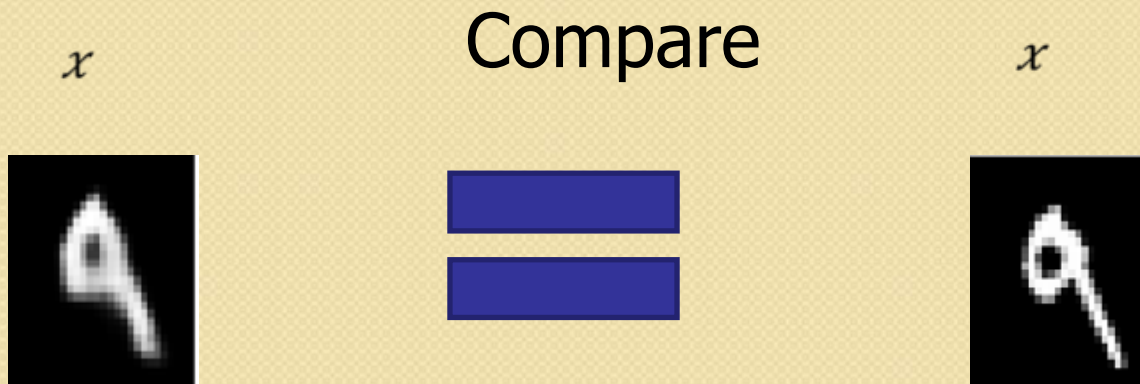


Processo – Denoising Autoencoders

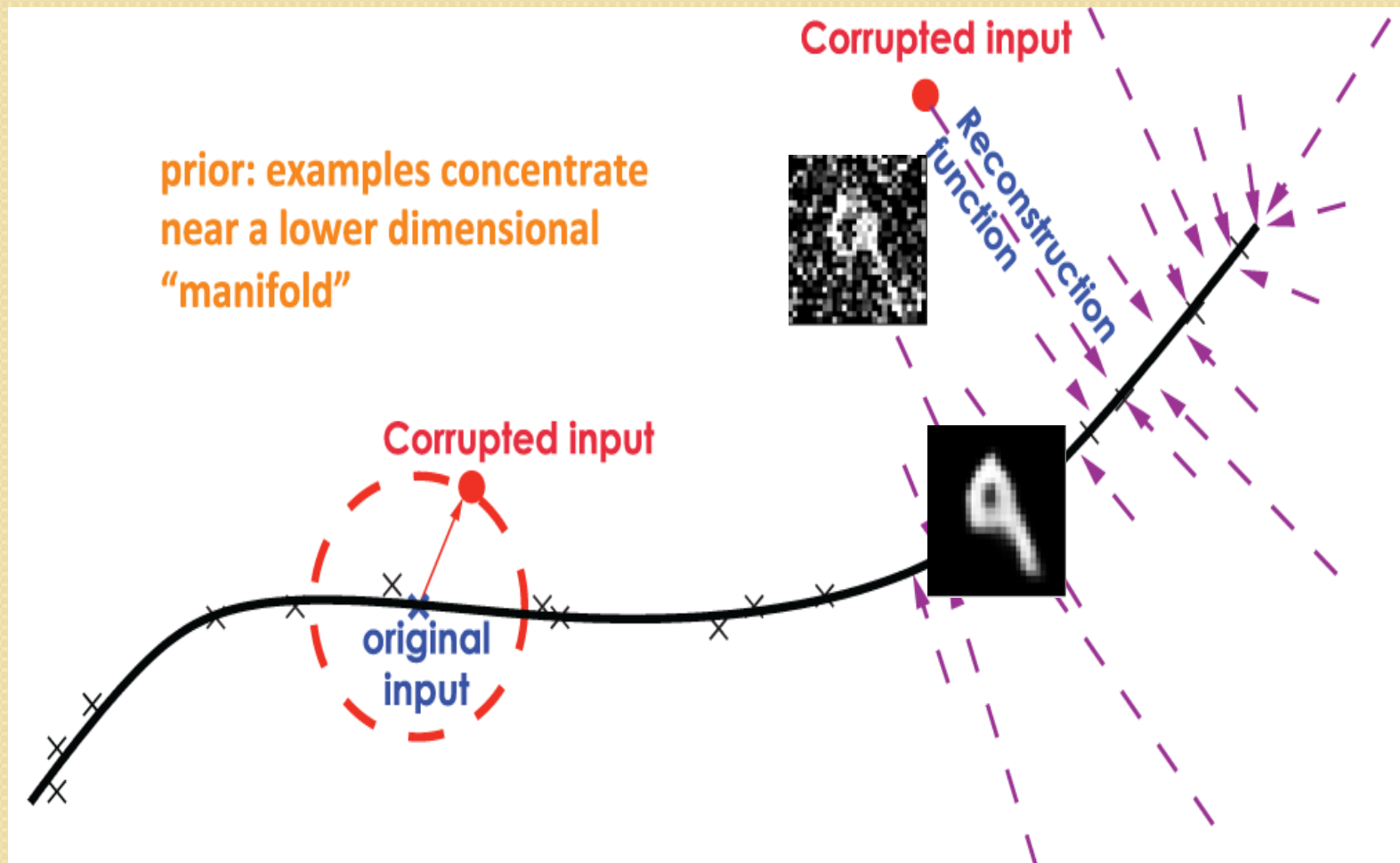
Decode



Processo – Denoising Autoencoders

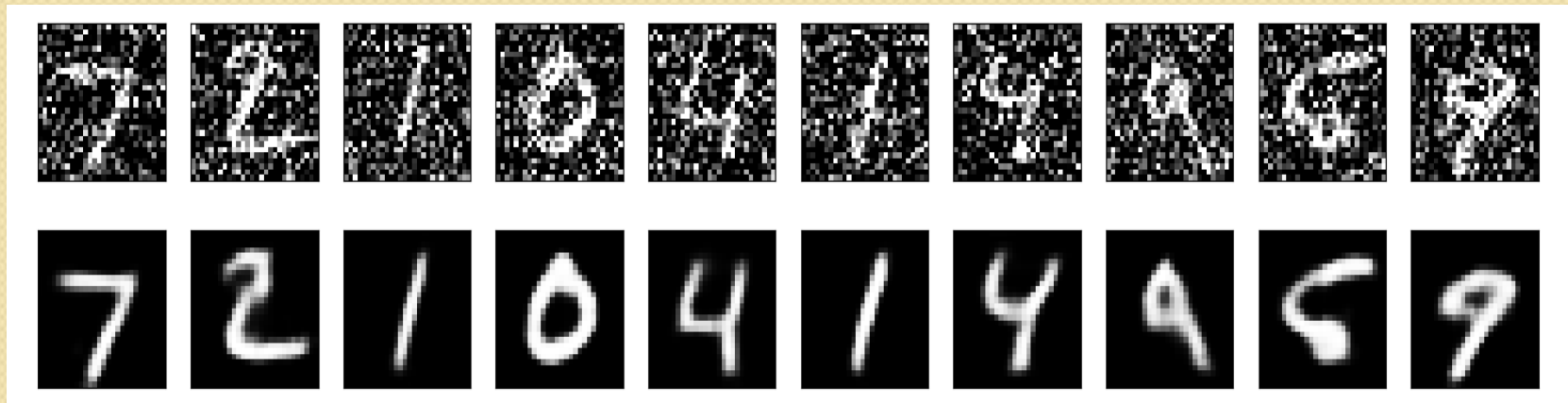


Processo – Denoising Autoencoders



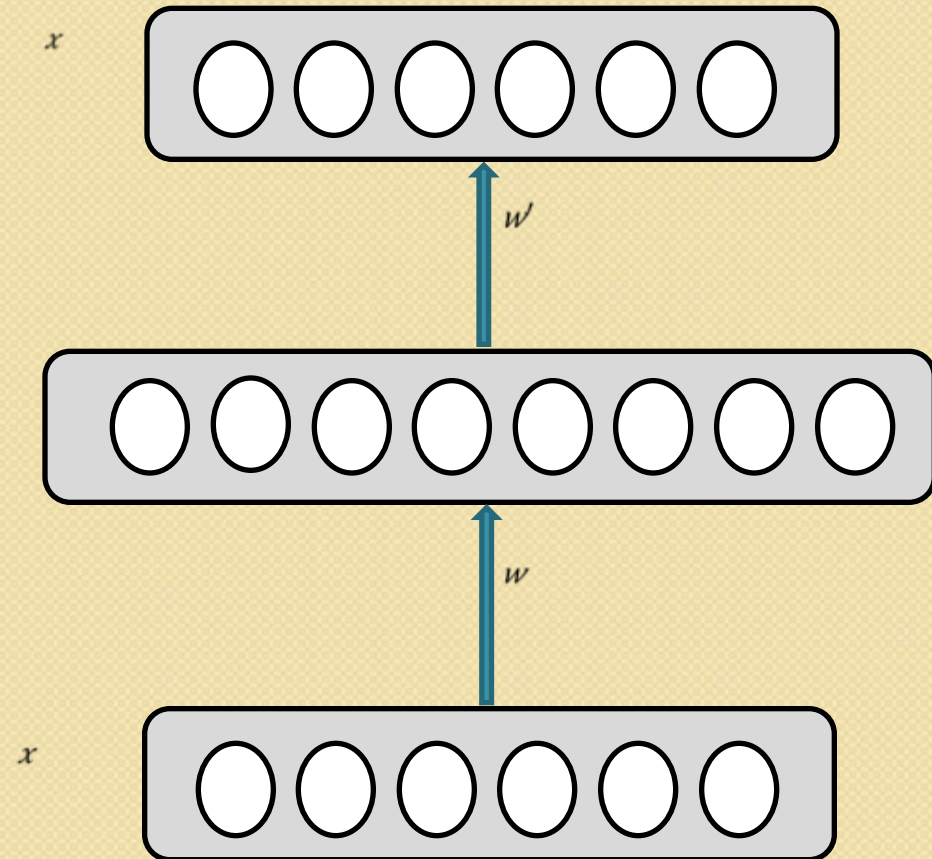
Denoising convolutional AE – keras

- 50 epochs.
- Noise factor 0.5
- 92% accuracy on validation set.



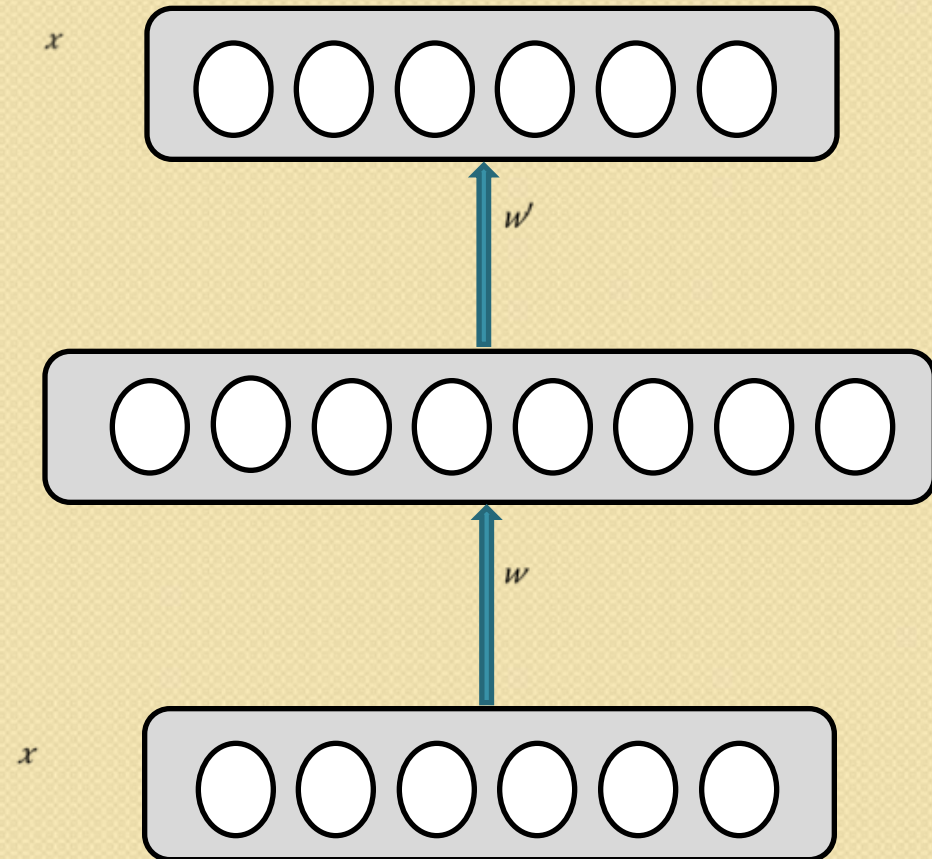
Contractive autoencoders

- Evitar características indesejadas
 - Adicionar um termo de regularização $\Omega(x)$ à função de perda para limitar as representações na camada escondida



Contractive autoencoders

- – Idea: extrair características que representam apenas variações no conjunto de treinamento. Ficar invariante a quaisquer outras variações
- Pontos próximos entre si no espaço de entrada devem manter essa propriedade no espaço latente



Autoencoder contrativa

(*contractive autoencoder*)

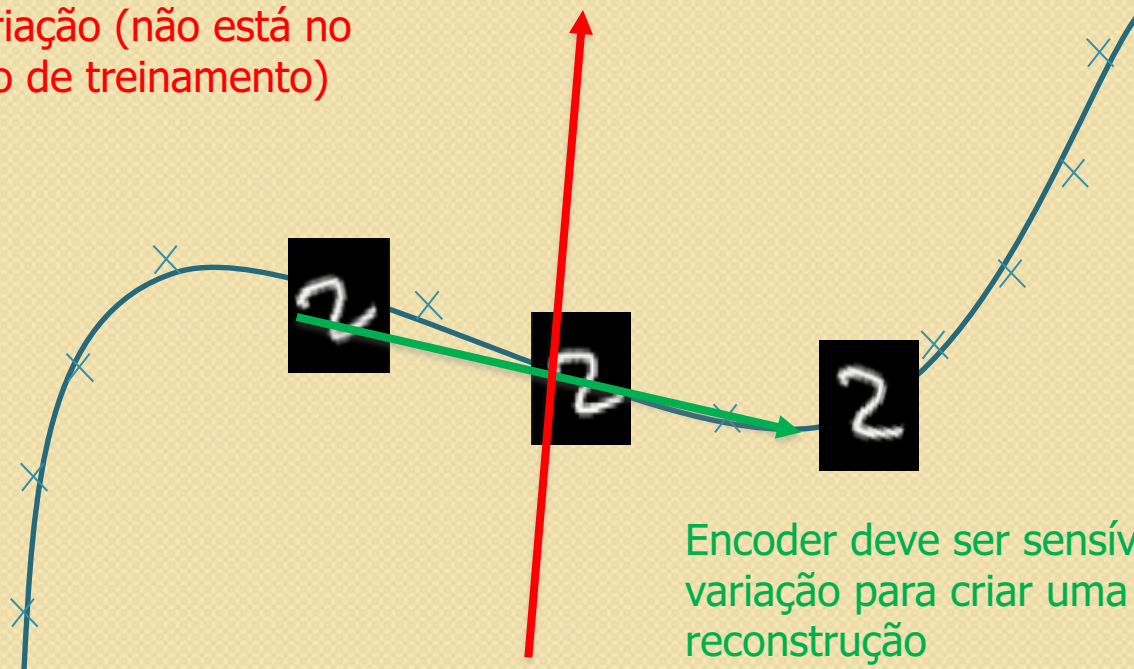
37

- Ideia básica: adicionar **penalização** à função de perda para penalizar representações indesejadas.

$$\underbrace{- \sum_{k=1}^d [x_k \log z_k + (1 - x_k) \log(1 - z_k)]}_{\text{"mantenha boas representações"}} + \underbrace{\left\| \frac{\partial h(x)}{\partial x} \right\|^2}_{\text{"descarte todas as representações"}} \leftarrow \text{penalização} = \text{"mantenha apenas boas representações"}$$

Contractive autoencoders

Encoder não deve ser sensível a essa variação (não está no conjunto de treinamento)



Encoder deve ser sensível a essa variação para criar uma boa reconstrução

Autoencoder esperso

(sparse autoencoder)

39

- Objetivo: apenas pequena quantidade de unidades da camada oculta seja ativada para cada padrão de entrada.
- Esparsidade pode ser obtida
 - Termos adicionais na função de perda durante o treinamento
 - Forçar apenas as k unidades mais ativas e tornando todas as demais unidades iguais a zero

Autoencoder esparso

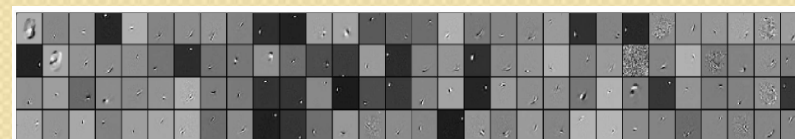
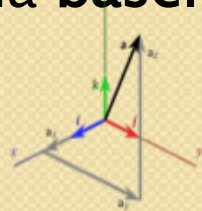
(sparse autoencoder)

- Motivação biológica: visual córtex (V1)

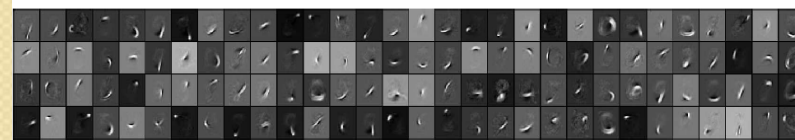


$$7 = 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 1 \begin{matrix} \text{[digit 7]} \end{matrix} + 0.8 \begin{matrix} \text{[digit 7]} \end{matrix} + 0.8 \begin{matrix} \text{[digit 7]} \end{matrix}$$

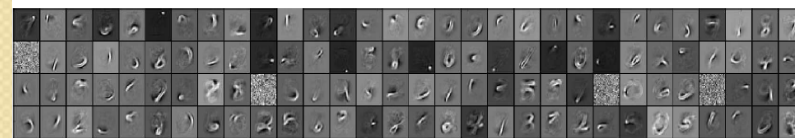
Exemplo: imagens de 28x28 pixels podem ser representadas por uma qtd. pequena de **códigos** a partir de uma **base**.



(a) $k = 70$



(b) $k = 40$



(c) $k = 25$

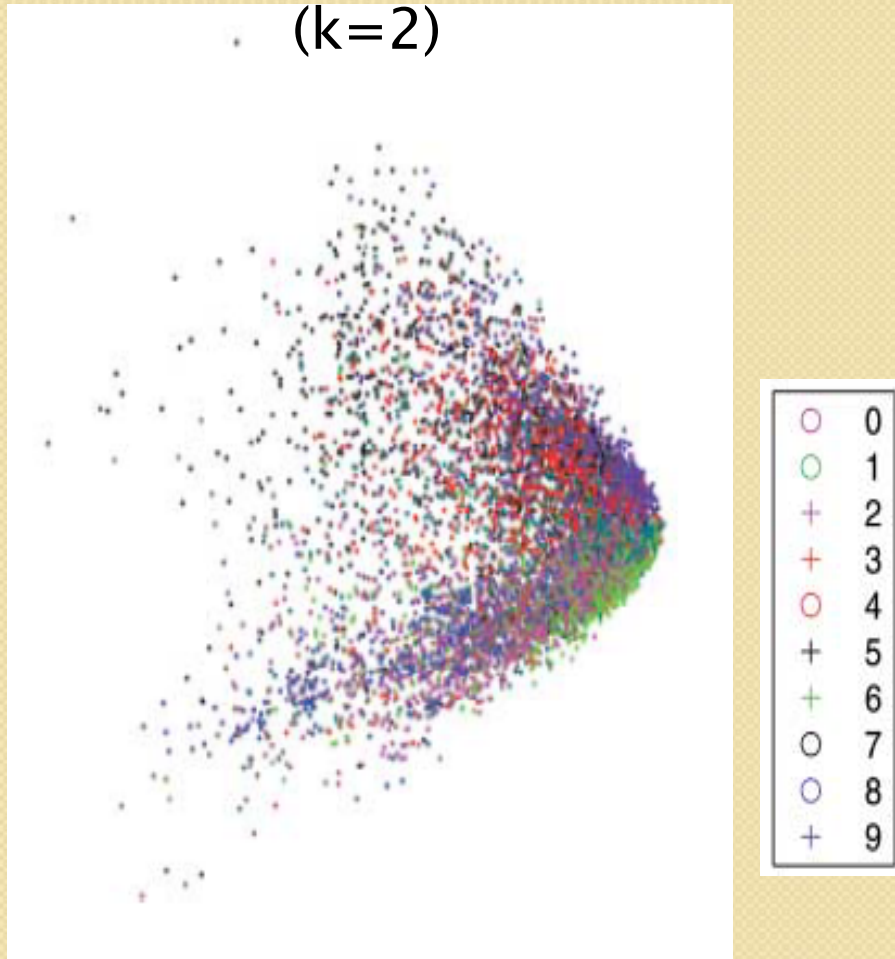


(d) $k = 10$

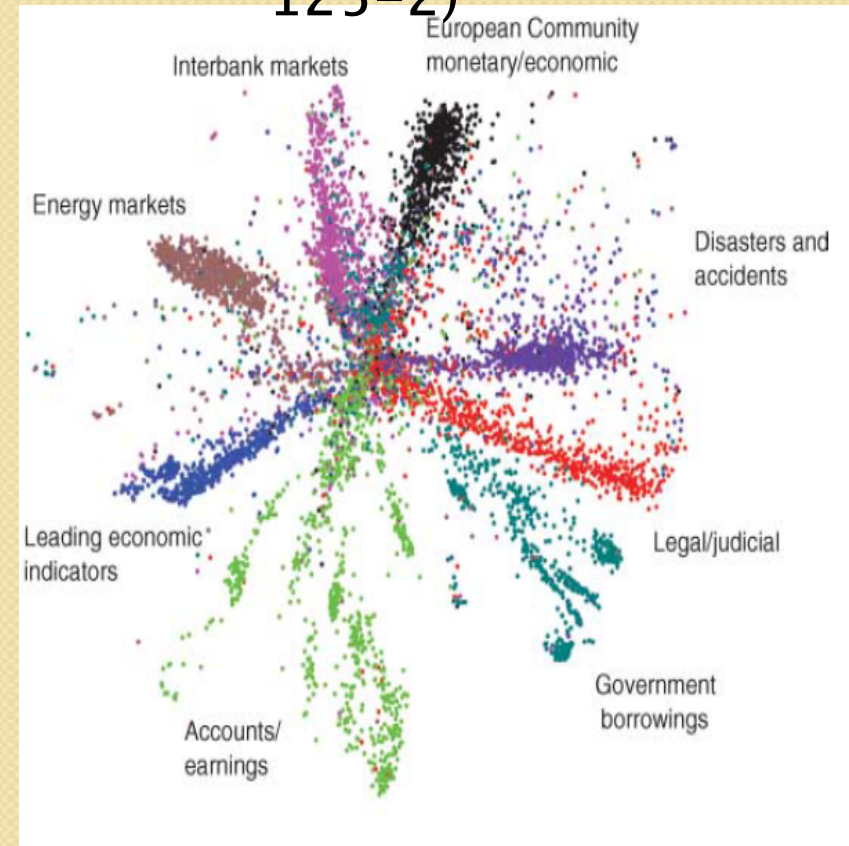
Aplicações (I) – redução de dimensionalidade

41

PCA
(k=2)



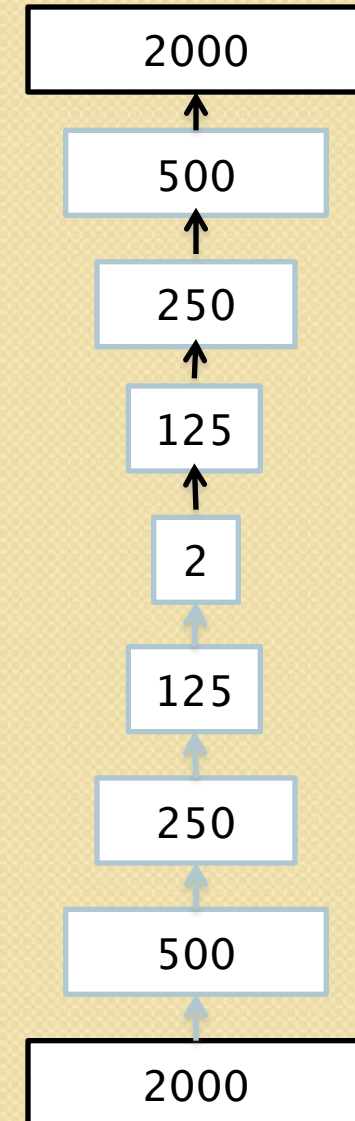
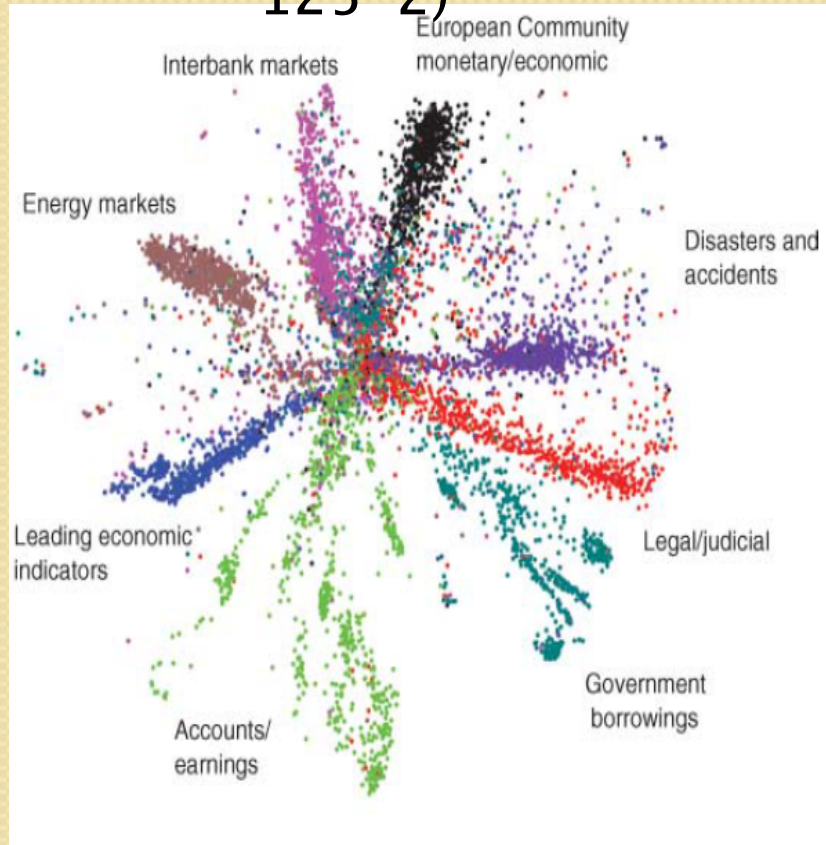
Autocodificadora
(2000-500-250-
125-2)



Aplicações (I) – redução de dimensionalidade

42

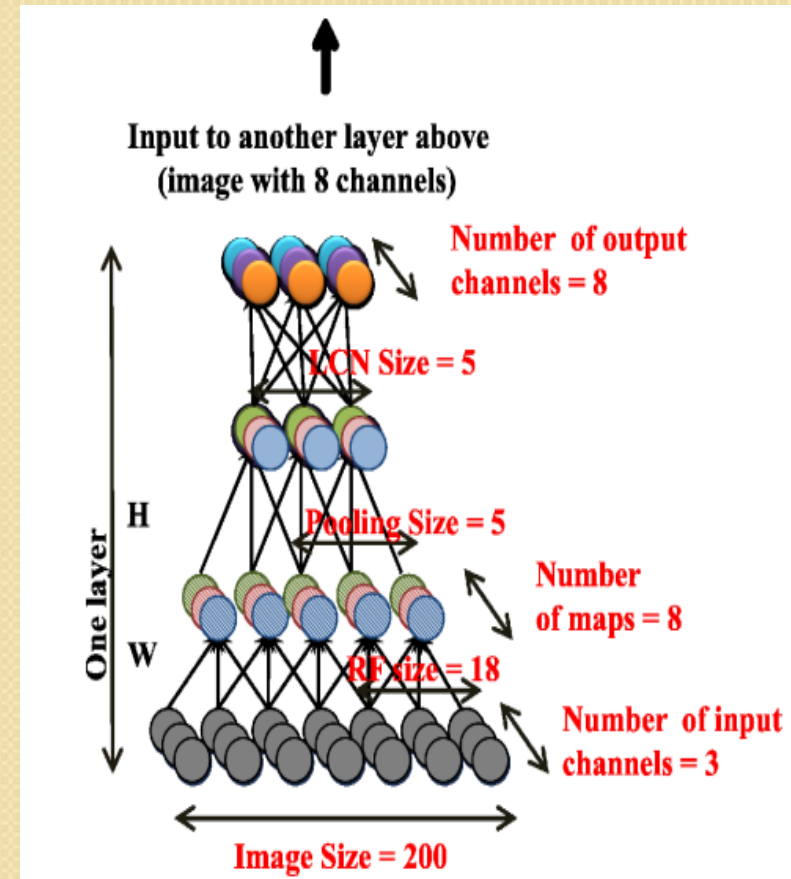
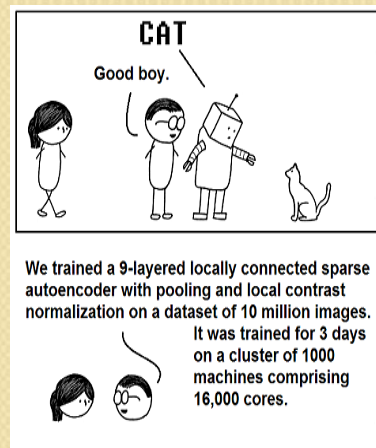
Autocodificadora
(2000–500–250–
125–2)



Aplicações (II) – aprendizado de conceitos

43

- 10M vídeos do YouTube
 - 1 frame por vídeo (200x200)
- A rede aprendeu conceitos de face humana e de face de gatos



Ideia em si é simples, yet powerfull!

Com restrições colocadas na rede (limitar # de neurônios ou usar regularização) a rede pode descobrir a estrutura dos dados



Tenta capturar a distribuição dos dados (específica para cada conjunto de dados)

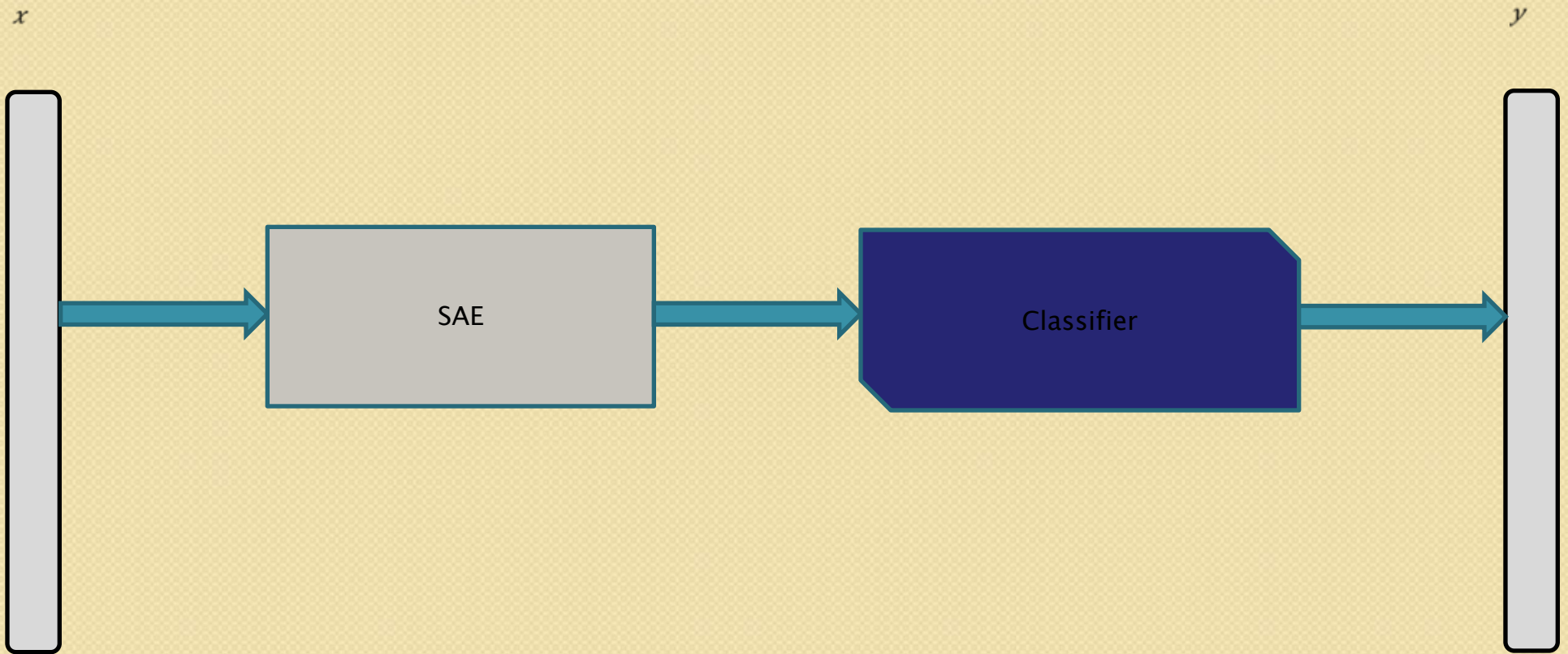
Stacked AE

- Usar a qualidade de extração de características do AE
- Construir um classificador supervisionado profundo
- Benefícios: modelo já é treinado de maneira mais “smart”
- Usar essa técnica supervisionada traz conceitualmente um treinamento com mais dados não rotulados

Stacked AE (SAE)

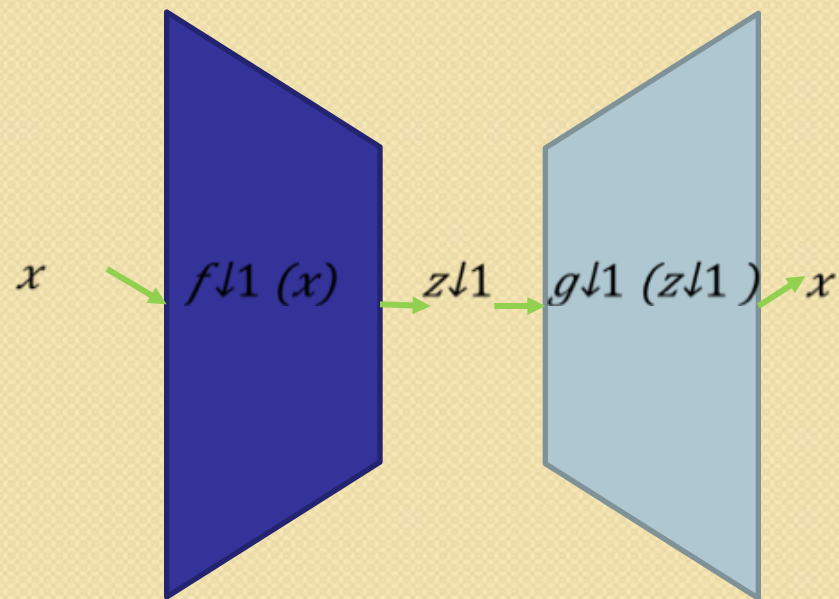
- Consiste de 2 etapas:
 - 1. Treinar cada camada AE uma após a outra
 - 2. Acoplar qualquer classificador no final (MLP/ SVM / RBF / SGB)

Stacked AE



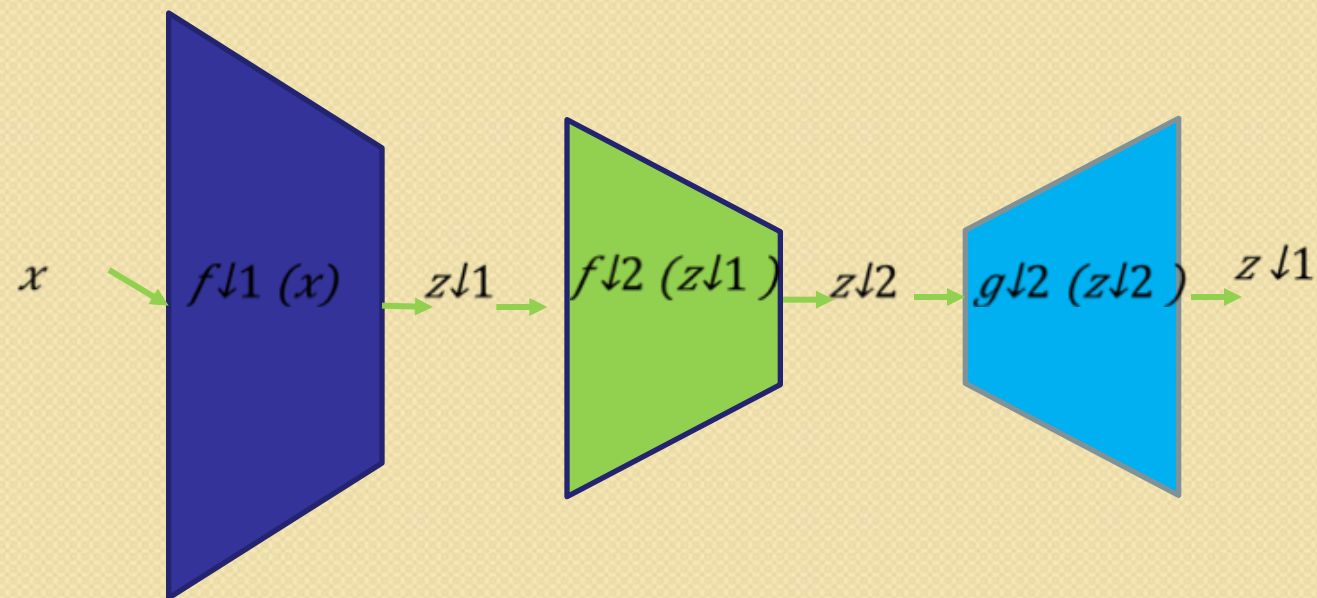
Stacked AE – treinamento

First Layer Training (AE 1)



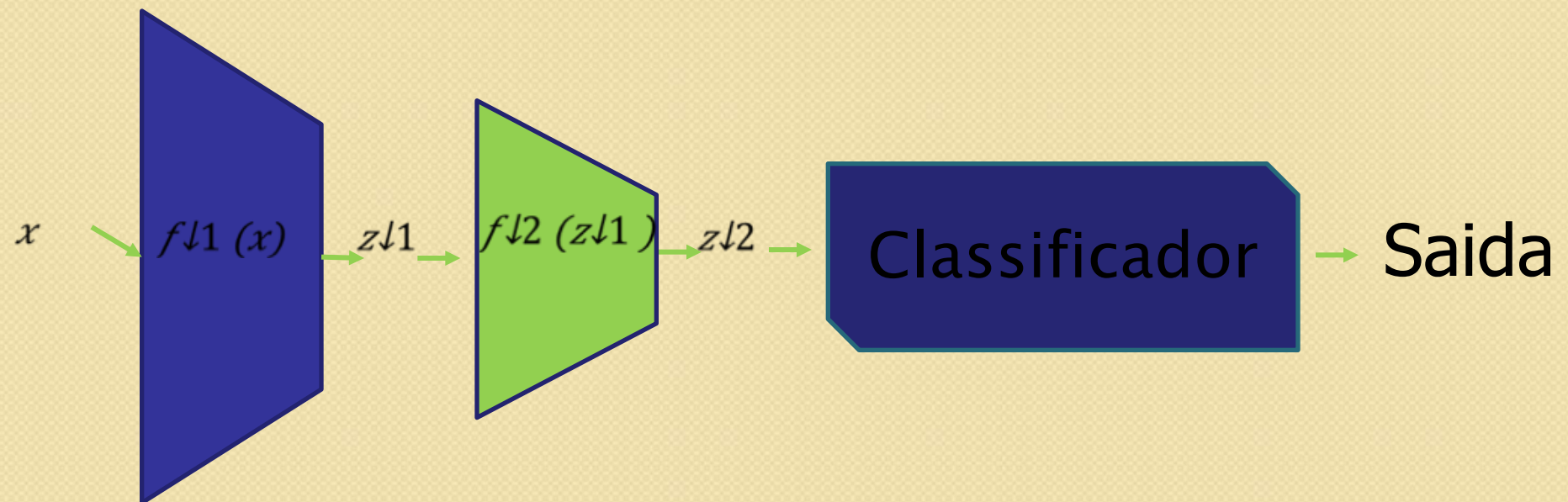
Stacked AE – treinamento

Second Layer Training (AE 2)



Stacked AE – treinamento

Adicione um classificador



Qual autoencoder?

- Ambos denoising AE e contractive AE funcionam bem!

Autoencoder padrão: exemplo

- <https://cs.stanford.edu/people/karpathy/convnetjs/demo/autoencoder.html> – By Andrej Karpathy

Autoencoder padrão: 3 exemplos tensorflow

- <https://www.tensorflow.org/tutorials/generative/autoencoder>
- <https://www.tensorflow.org/tutorials/generative/autoencoder?hl=pt-br#:~:text=An%20autoencoder%20is%20a%20special,representation%20back%20to%20an%20image.>

Autoencoder padrão

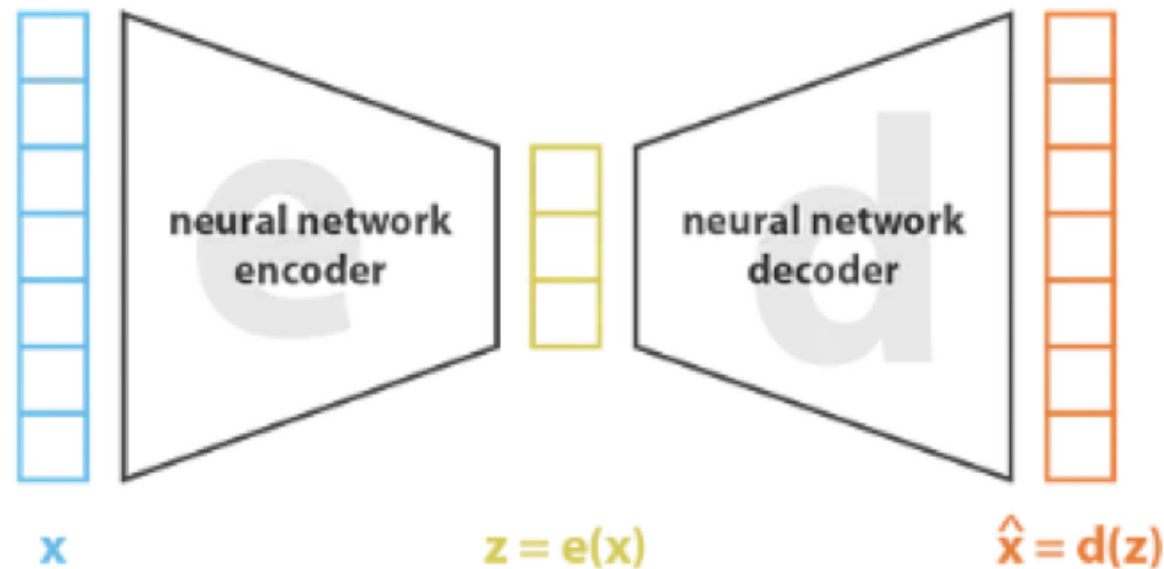
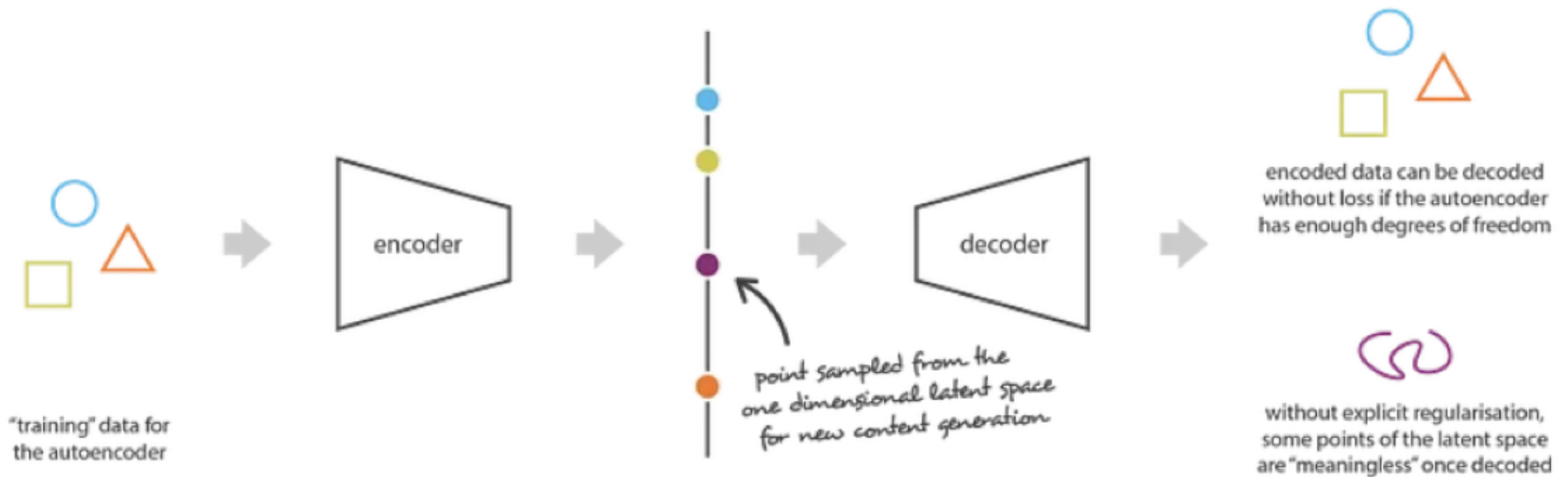
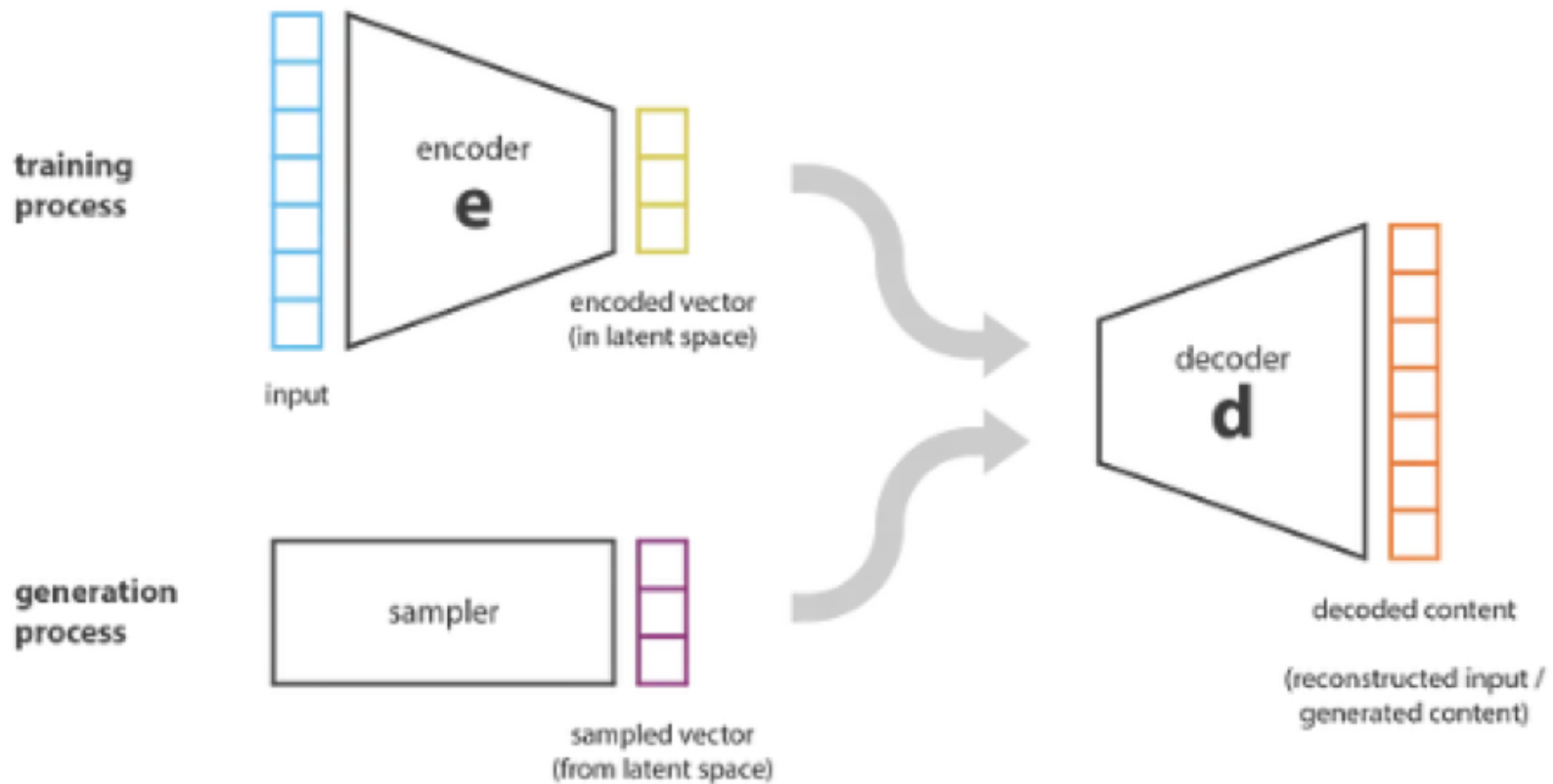


Illustration of an autoencoder with its loss function.

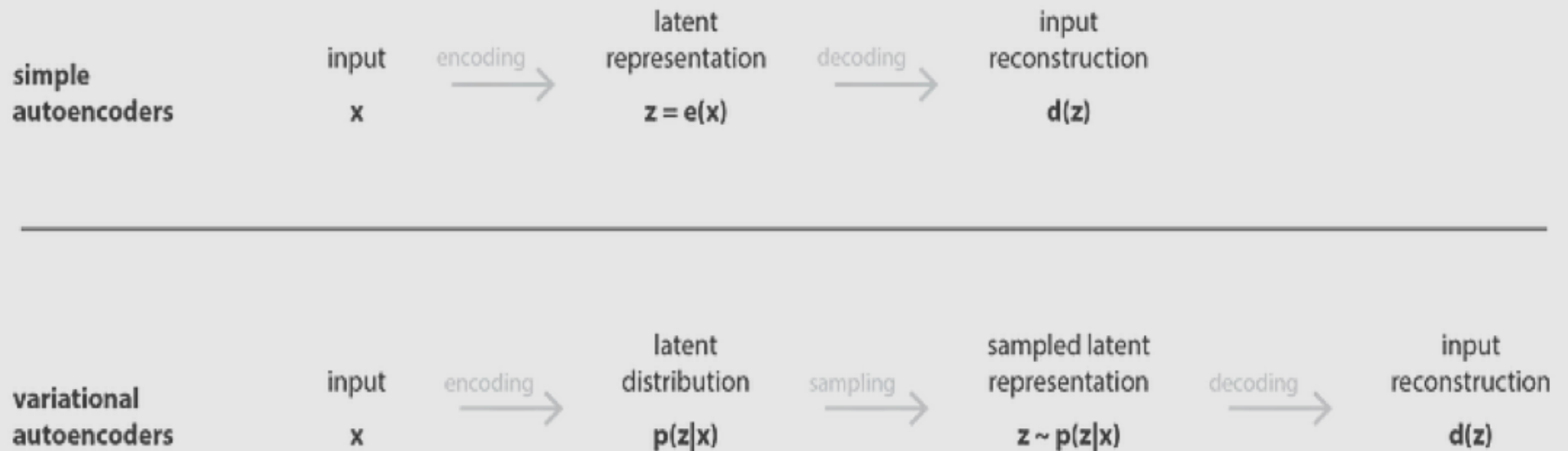
Autoencoder padrão



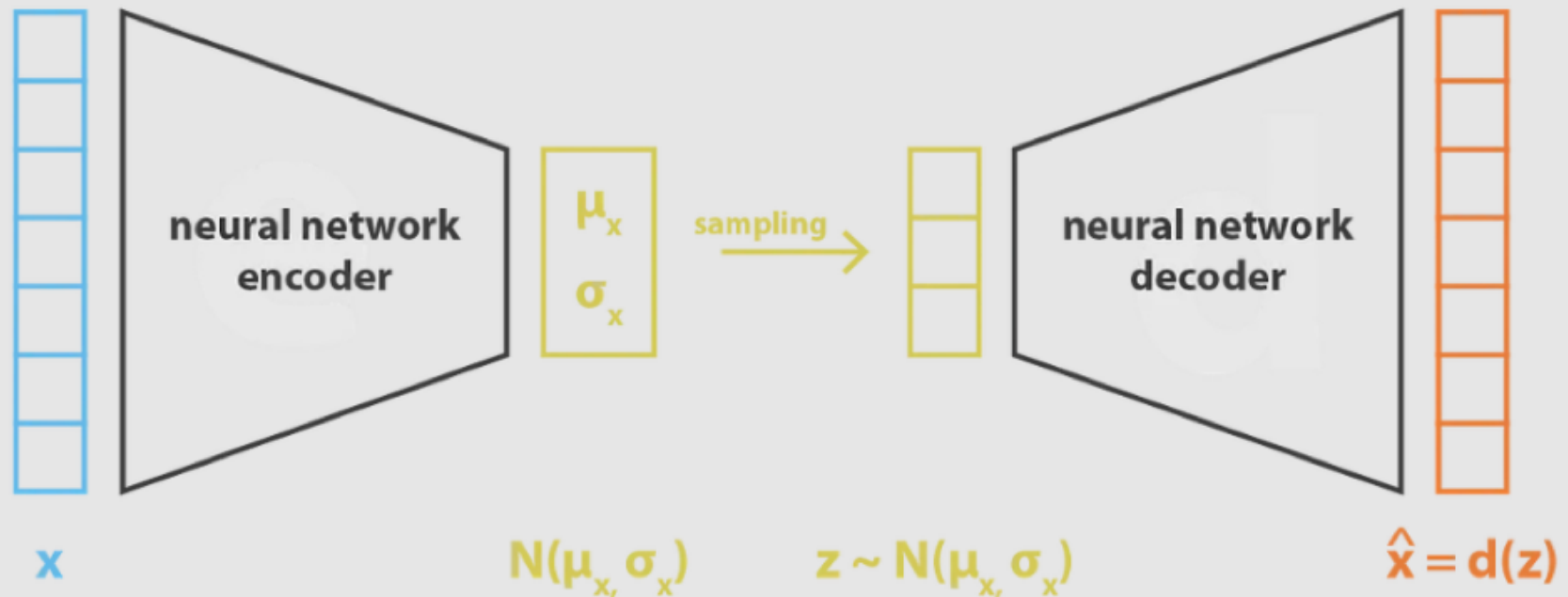
Autoencoder variacional



Variational Autoencoder

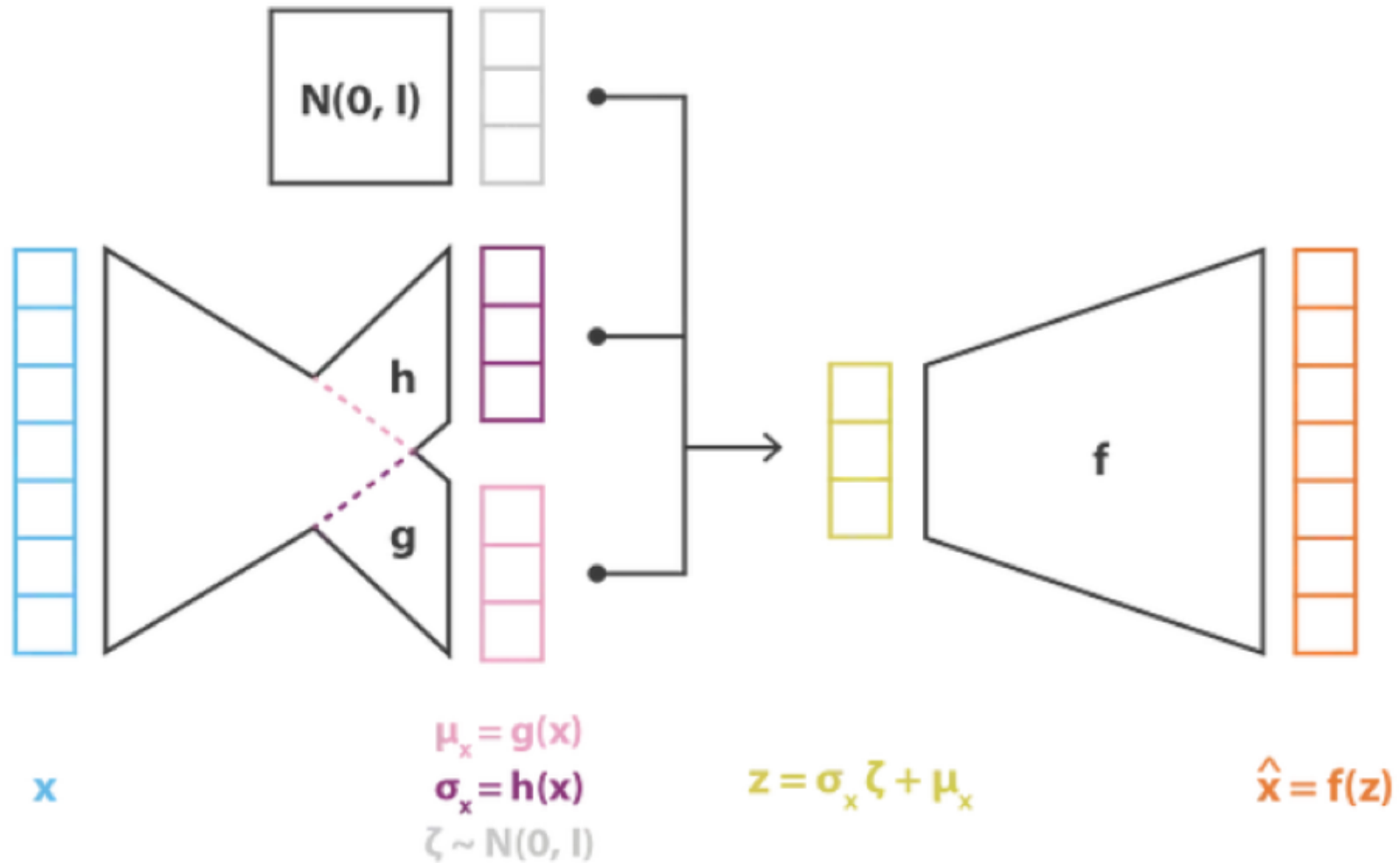


Variational Autoencoder



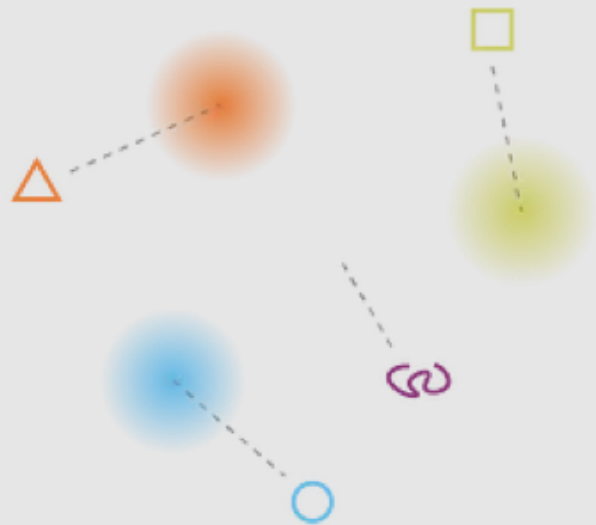
$$\text{loss} = ||x - \hat{x}||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

Variational Autoencoder

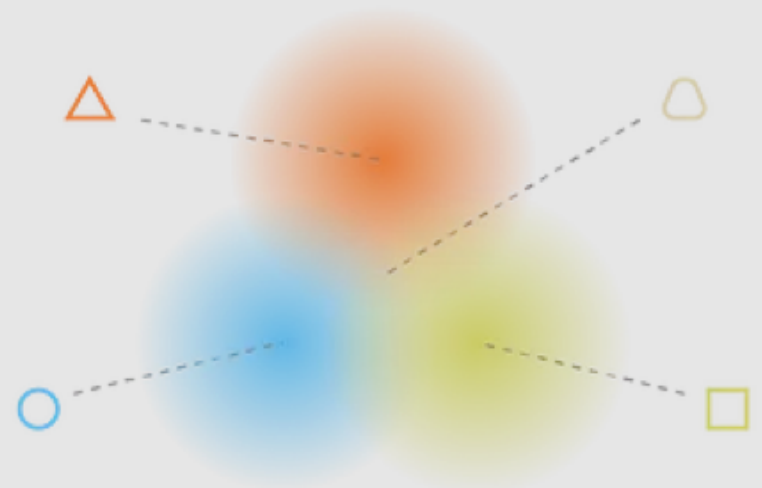


$$\text{loss} = C \|x - \hat{x}\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = C \|x - f(z)\|^2 + \text{KL}[N(g(x), h(x)), N(0, I)]$$

Variational Autoencoder



what can happen without regularisation



what we want to obtain with regularisation

Variational Autoencoder : Videos complementares bem explicativos

- <https://youtu.be/SSXDkfiPs7c?si=QKvjIXA6W89HbeMz> (denoising and variational autoencoder)
- <https://www.youtube.com/watch?v=vy8q-WnHa9A> (reparametrization trick)
- <https://www.scribbr.com/statistics/standard-normal-distribution/#:~:text=The%20standard%20normal%20distribution%2C%20also,the%20standard%20deviation%20is%201.> (normal distribution)

Variational Autoencoder: Explicação com matemática mais detalhada

- <https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>
- <https://synthesis.ai/2023/02/07/generative-ai-i-variational-autoencoders/>
- <https://youtu.be/9zKuYvjFFS8?si=rImBci-DInKV9I8g>

Referências Autoencoder

1. <https://arxiv.org/pdf/1206.5538.pdf>
2. <http://www.deeplearningbook.org/contents/autoencoders.html>
3. <http://deeplearning.net/tutorial/dA.html>
4. <http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/>
5. <http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf>

Variational Autoencoder: exemplos

- <https://www.siares.com/projects/variational-autoencoder>
- <https://www.mathworks.com/help/deeplearning/ug/train-a-variational-autoencoder-vae-to-generate-images.html>