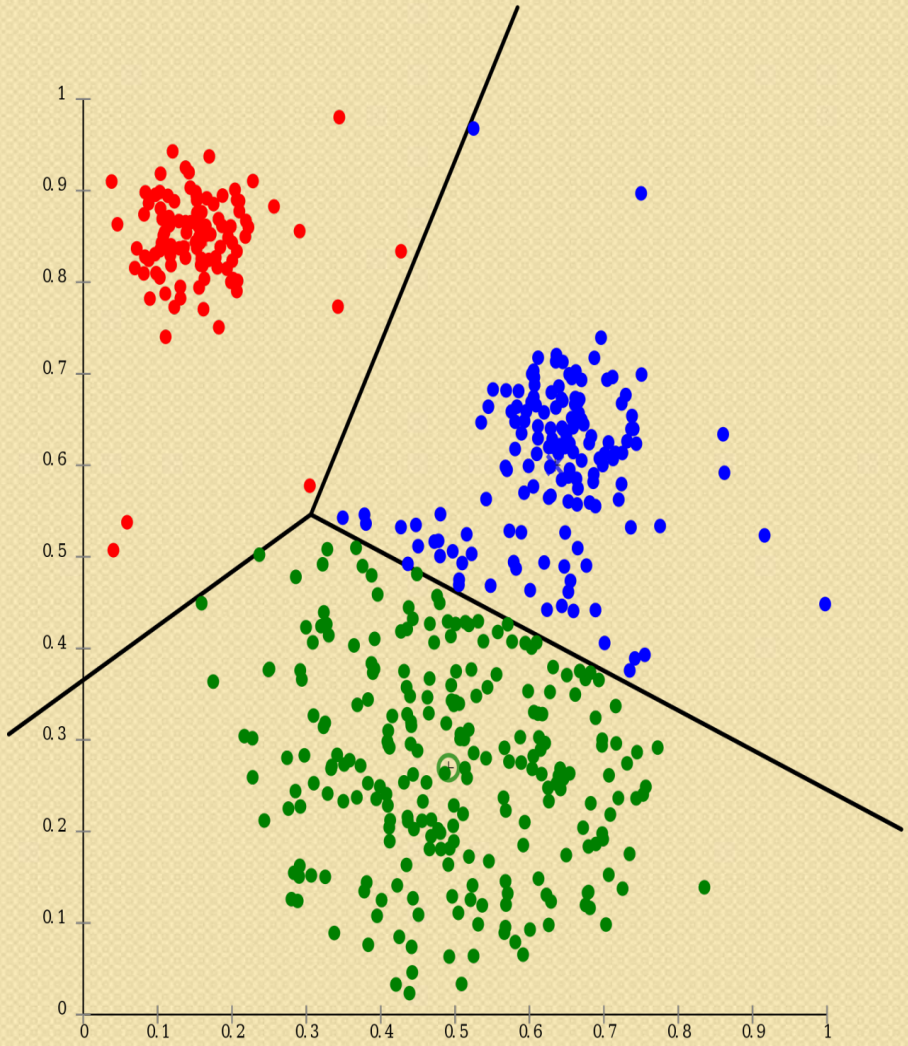
Autocodificadores (Autoencoders)

Germano C. Vasconcelos
Centro de Informática - UFPE

Conteúdo

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

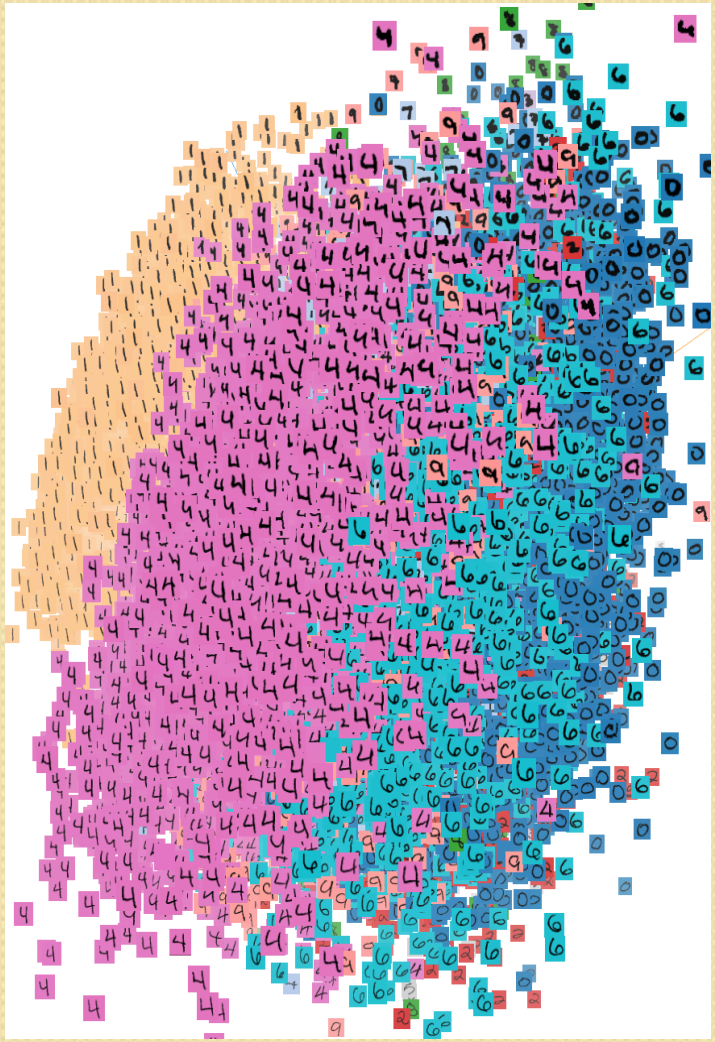
Aprendizagem não supervisionada



Aprendizagem não supervisionada

Dados: X (sem rótulos!)

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

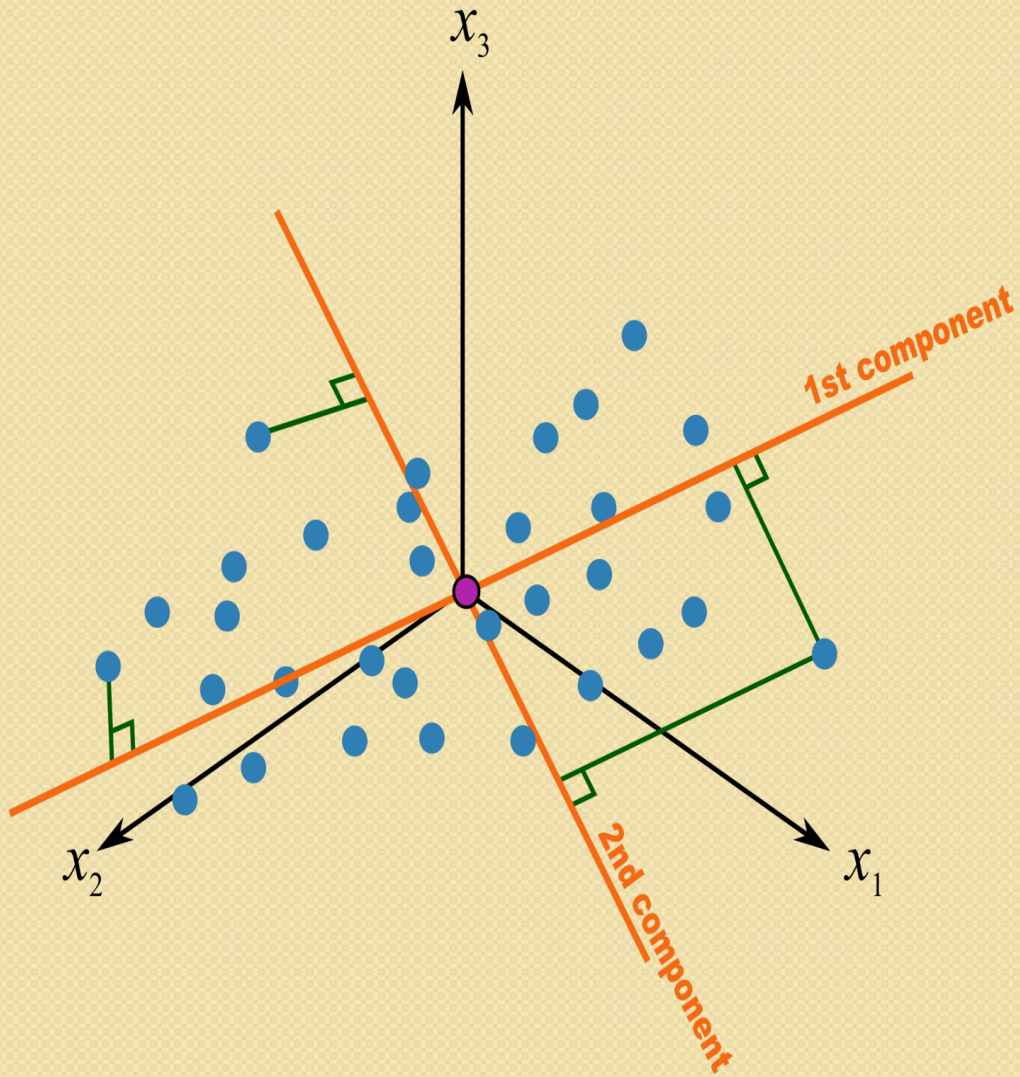


PCA – Análise dos Componentes Principais

- 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

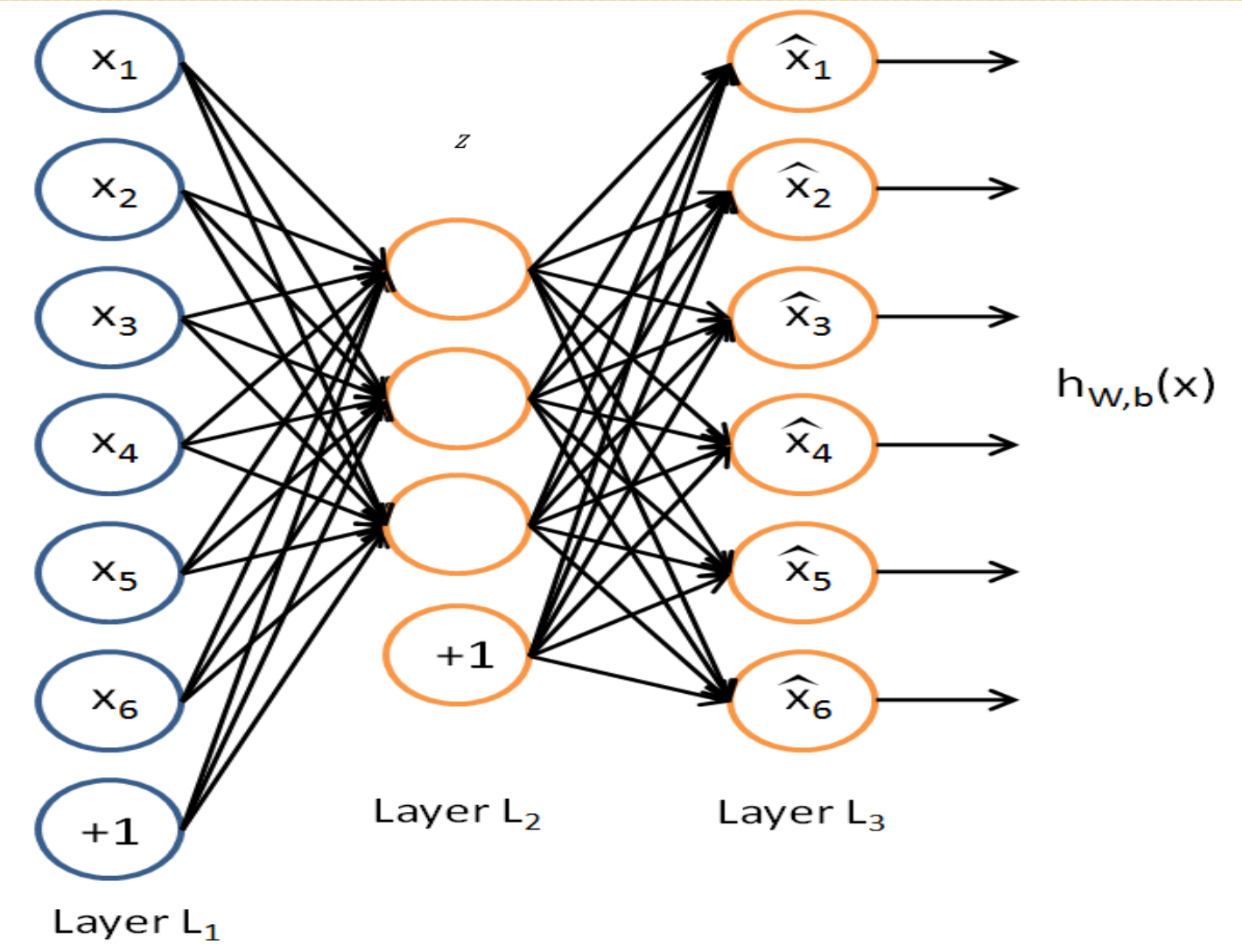
- Transformações são aplicadas na entrada de acordo com dois tipos de funções.
  - Função de extração de características (*encoder*) mapeia o conjunto de treinamento para uma **representação latente.**

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

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

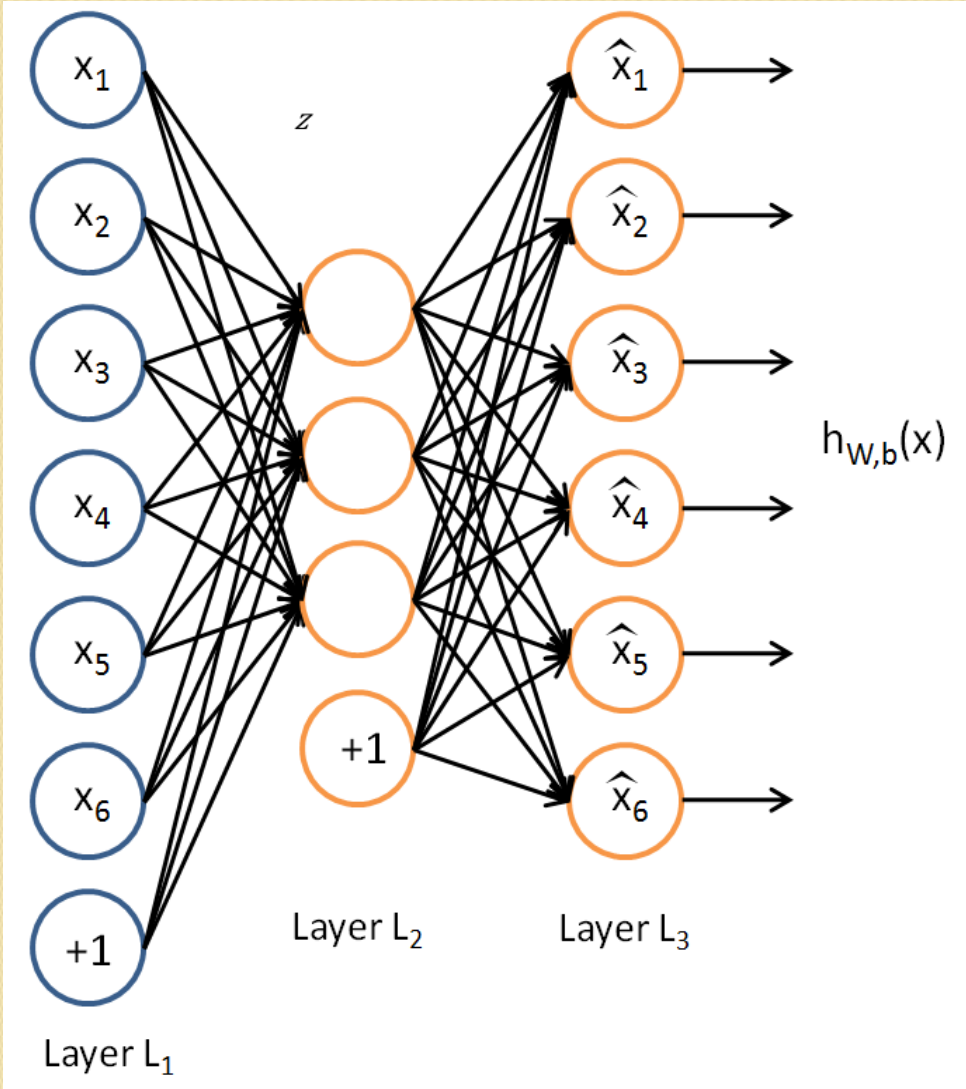
$$r : \mathbb{R}^K \rightarrow \mathbb{R}^D$$

Autoencoder Tradicional

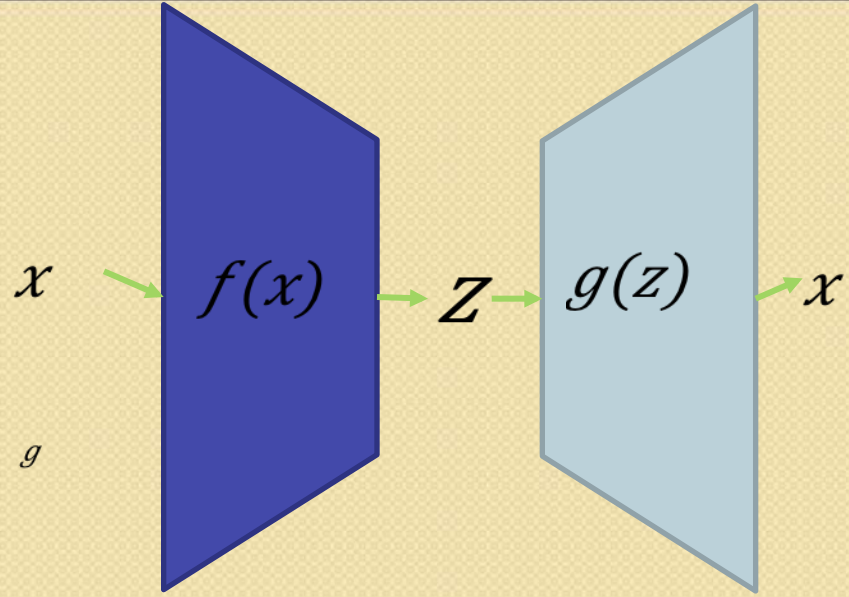


Autoencoder Tradicional (AE)

- 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:

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

and

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

s.t $h(x)=g(f(x))=x$

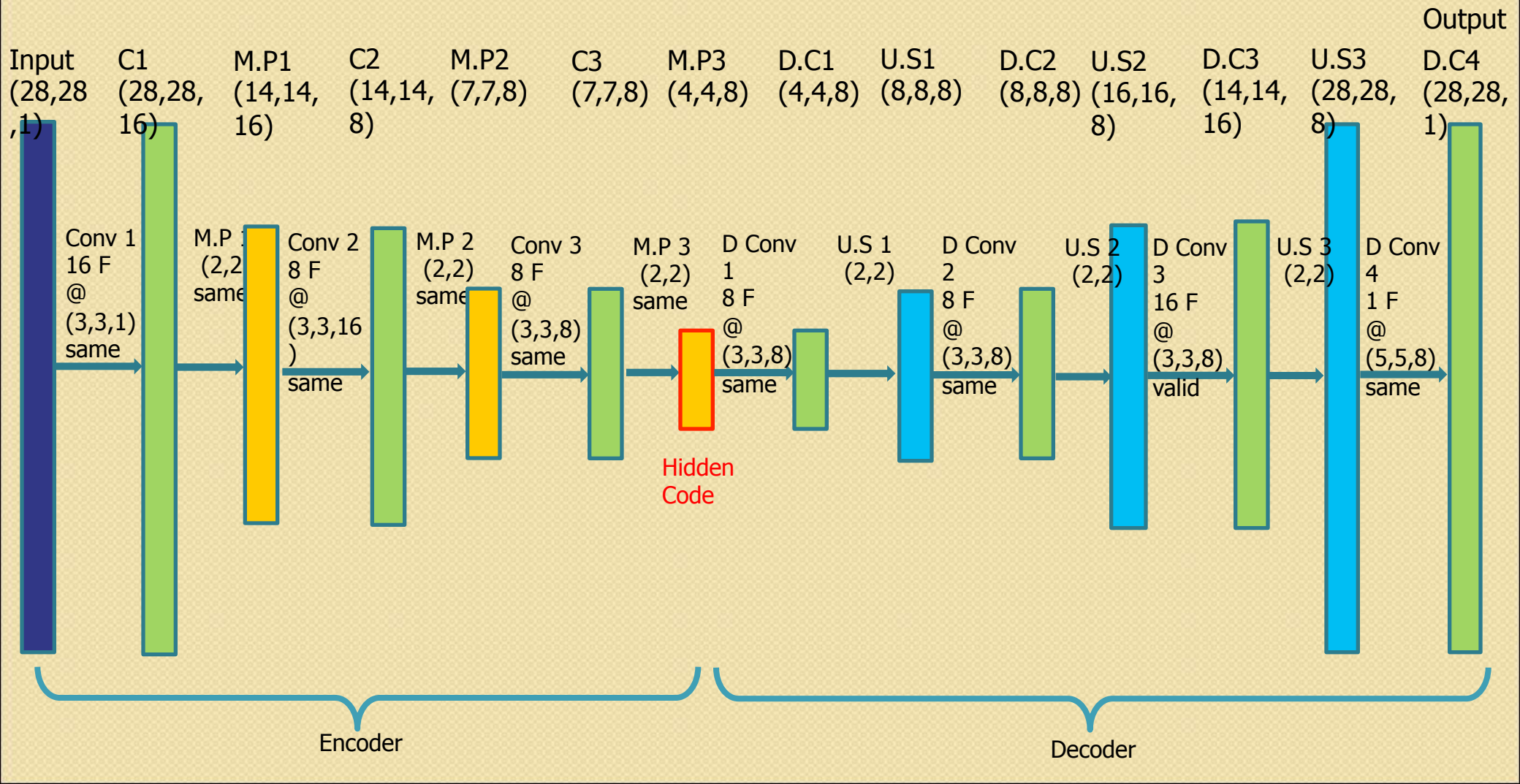
where h is an **approximation** of the identity function.

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

(x is x's reconstruction)

Convolutional AE

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



Treinamento

- 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 **x**.

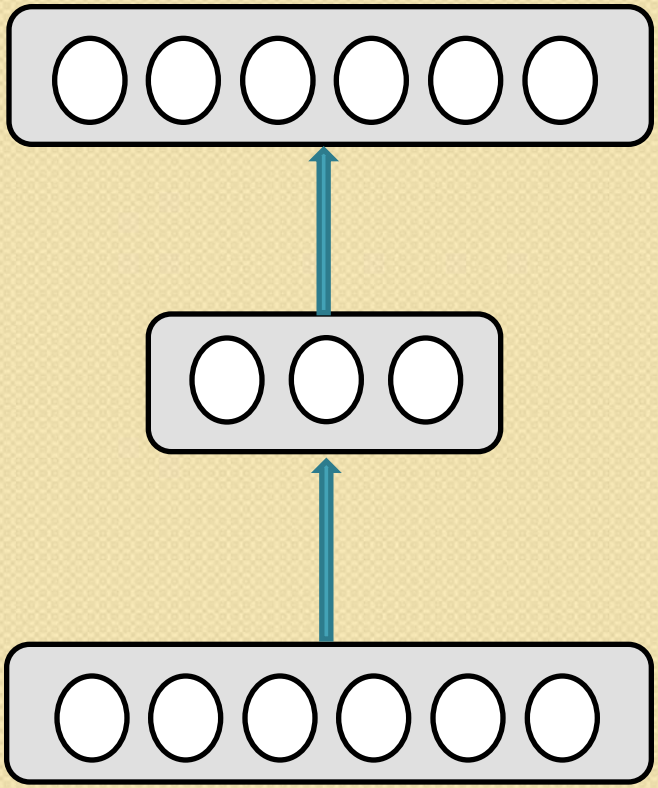
$$L(\mathbf{X}) = \sum_{\mathbf{x}^{(i)} \in \mathbf{X}} \ell(\mathbf{x}^{(i)}, r(h(\mathbf{x}^{(i)})))$$

Subcompletos x Sobrecompletos (Undercomplete vs Overcomplete)

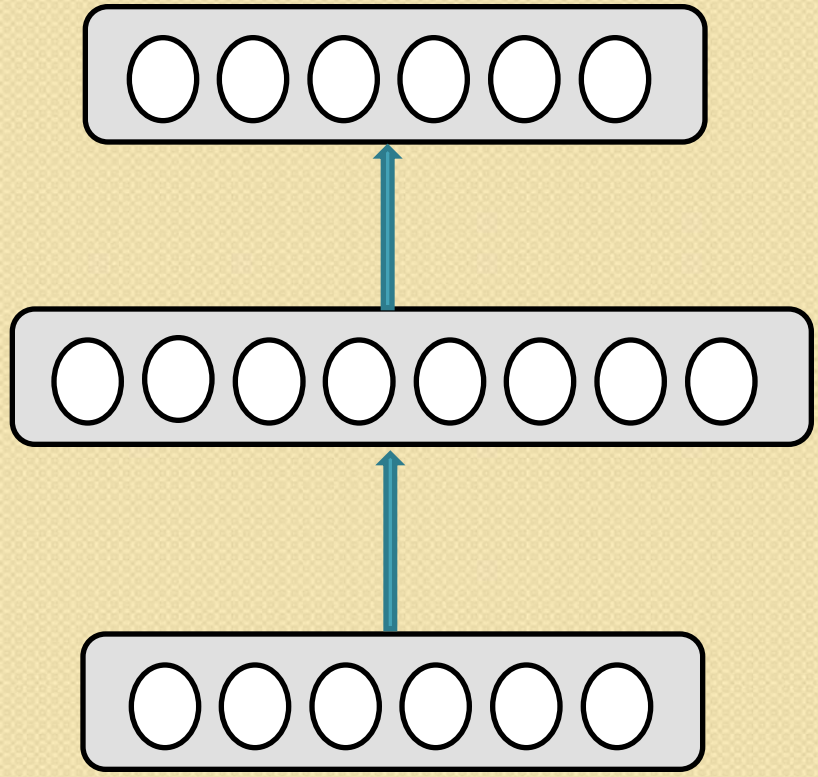
- Representação latente em uma autocodificadora tem dimensão K:
  - K < D → undercomplete autoencoder;
  - K > D → overcomplete autoencoder.
- A escolha de K determina
  1. Quantidade de unidades da camada intermediária central
  2. Tipo de informação a autocodificadora pode aprender da distribuição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

Undercomplete AE VS Overcomplete AE

Subcompleto -
Undercomplete

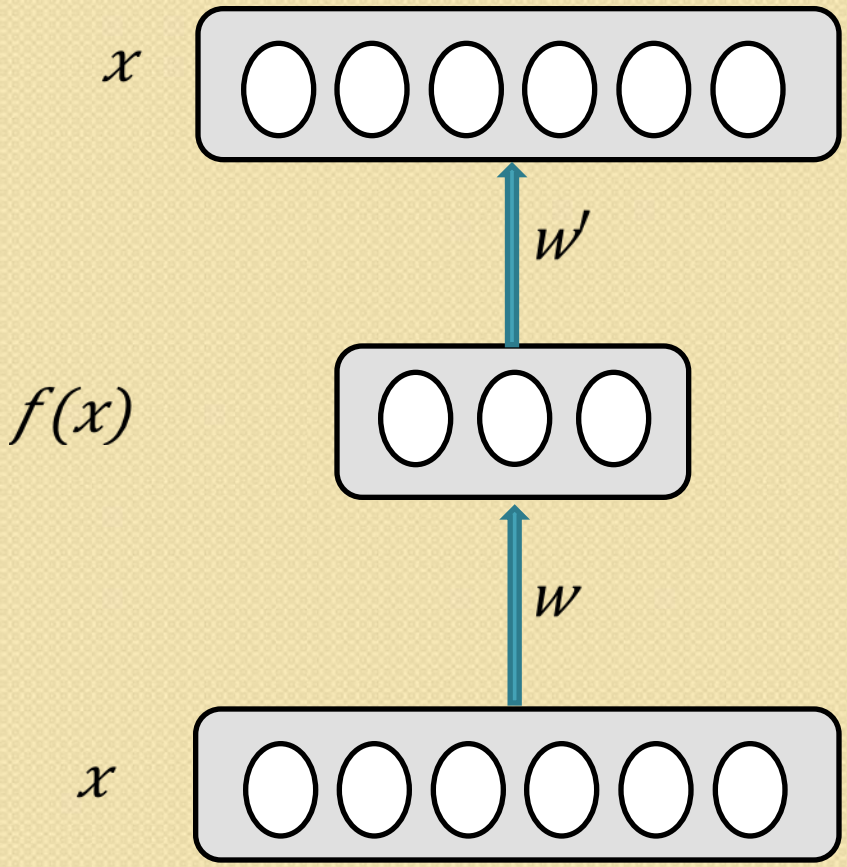


Sobrecompleto -
Overcomplete



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



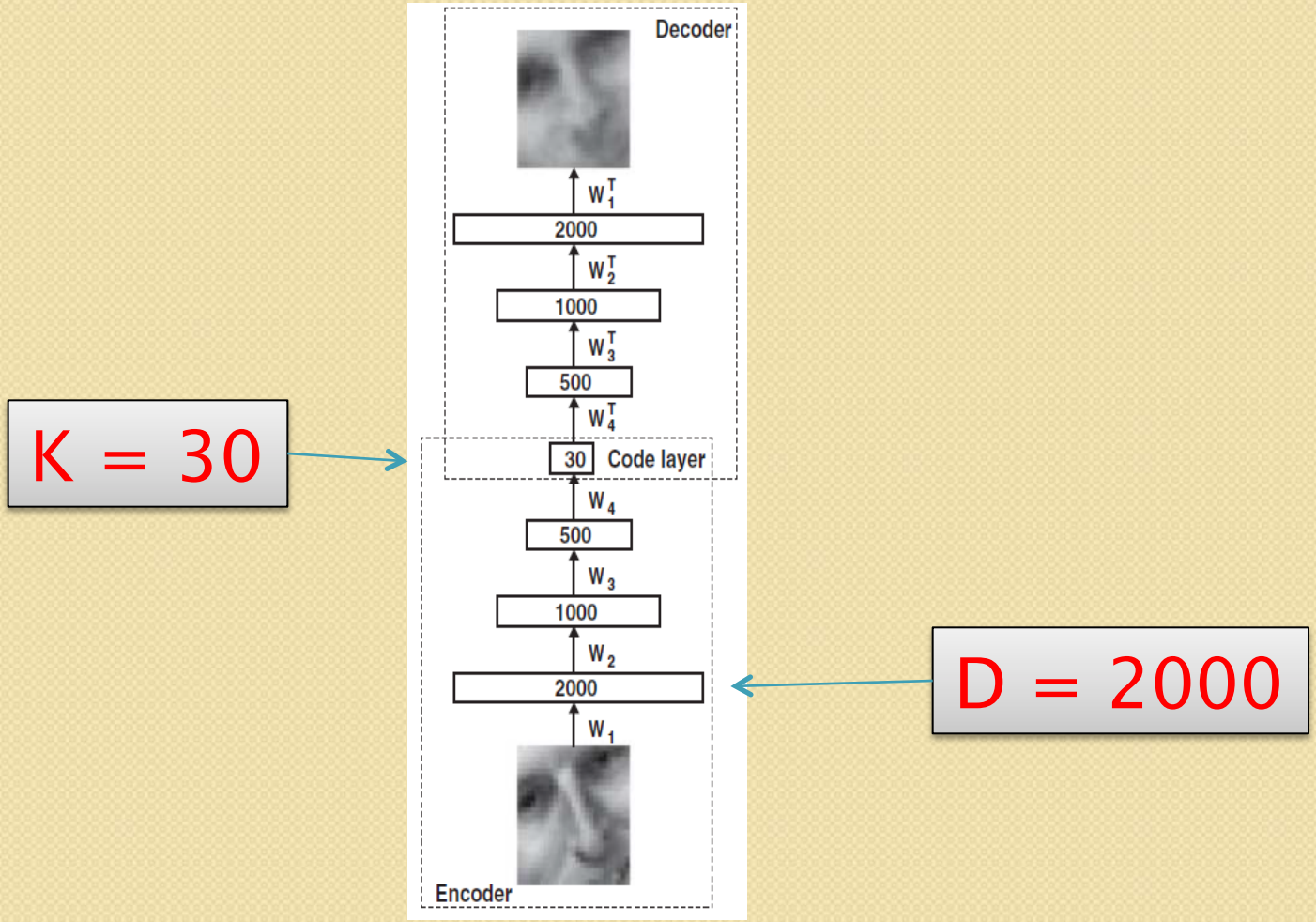
Caso K < D (*bottleneck*)

- 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*)



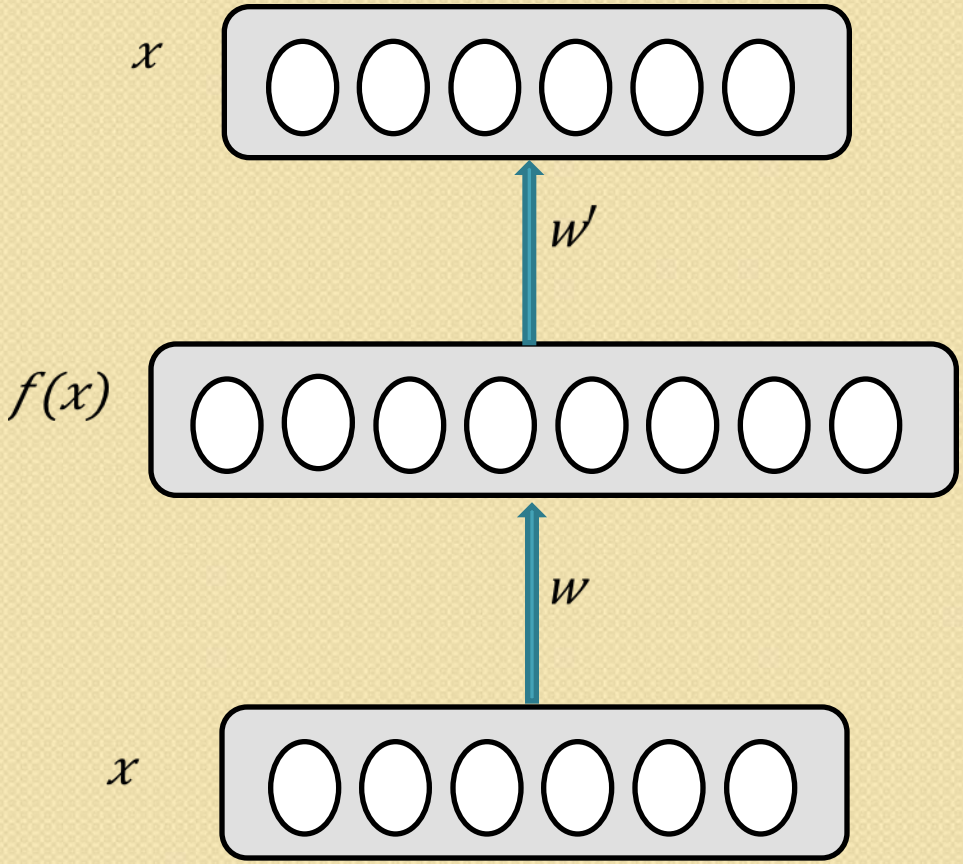
Reducing the Dimensionality of Data with Neural Networks (Science, 2006)

Caso K > D

- 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.

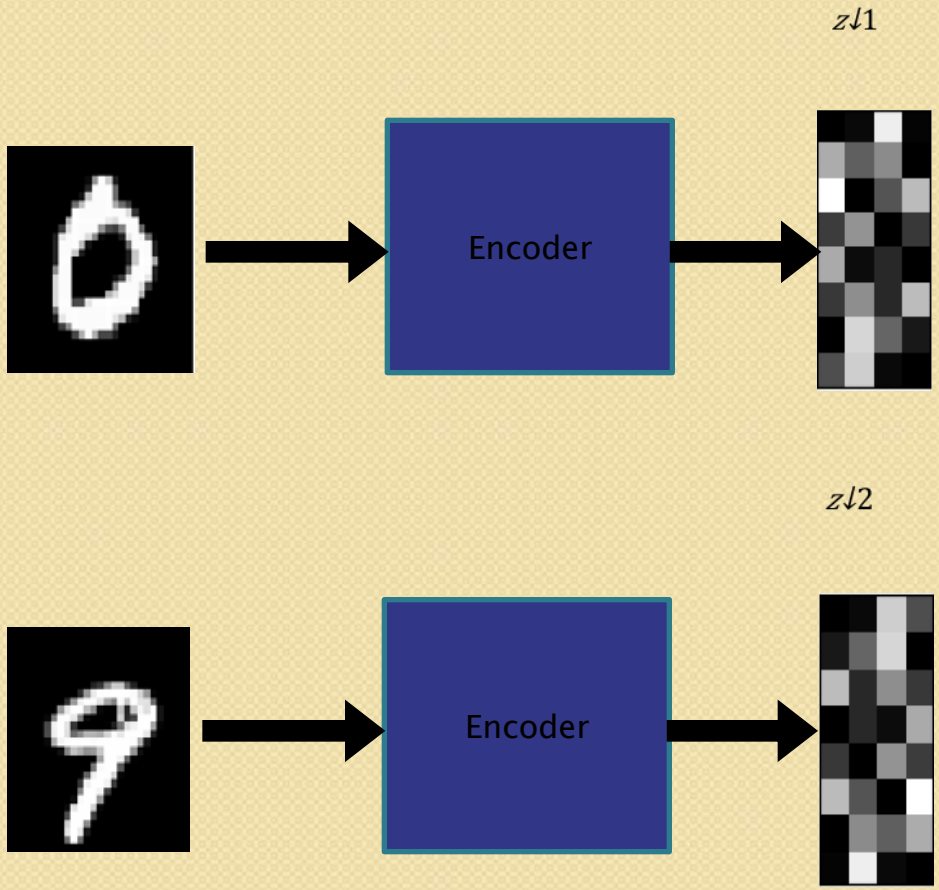
Subrecompleto AE

- Camada escondida é **Sobrecompleta** se maior que a de entrada
  - ❑ Sem compressão na camada intermediária.
  - ❑ Each hidden unit could copy a different input component.

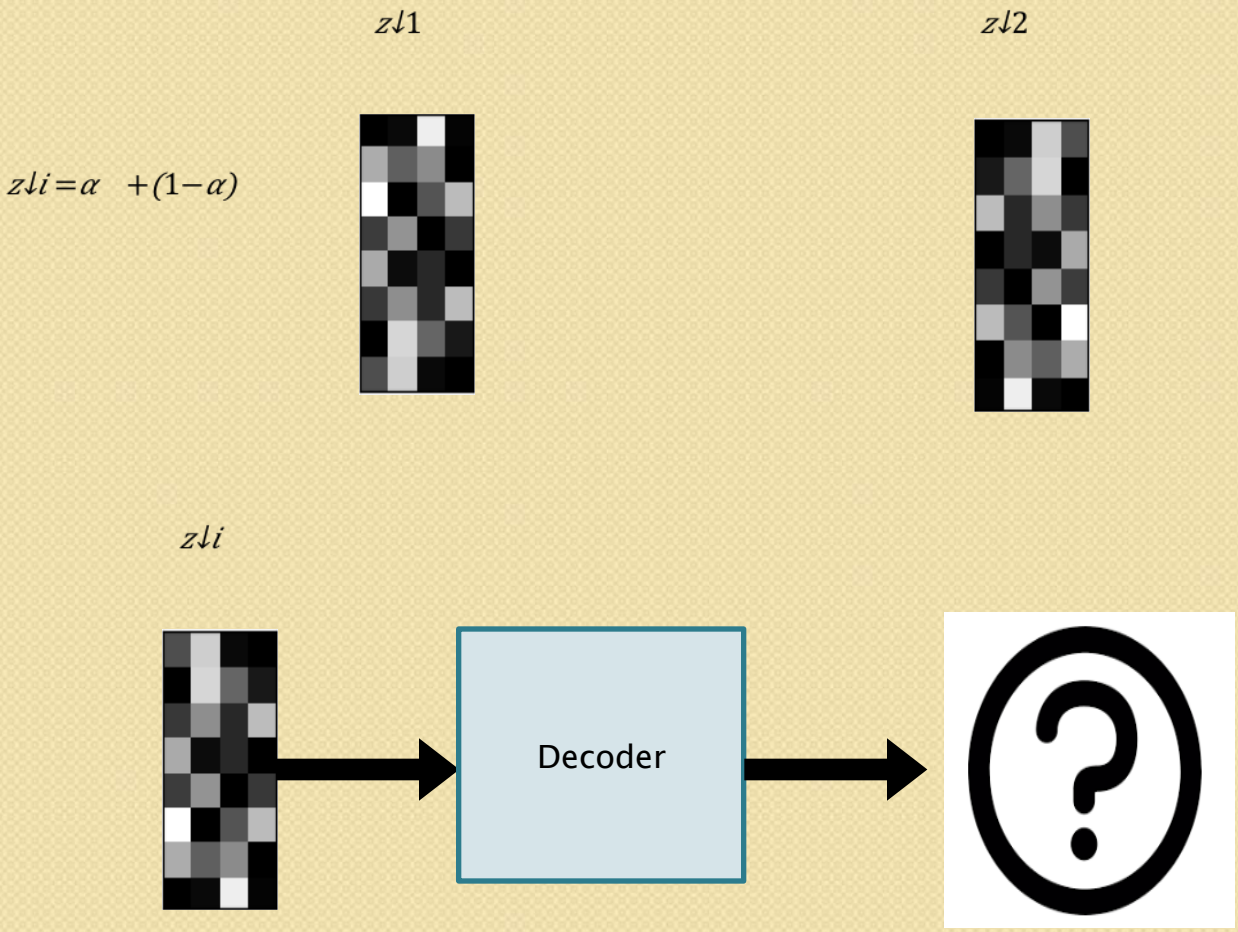


- Adicionar dimensões é útil para treinar um classificador linear.
- Um código em dimensão maior ajuda a modelar distribuições mais complexas
- Não há garantias que unidades intermediarias consigam extrair boas características

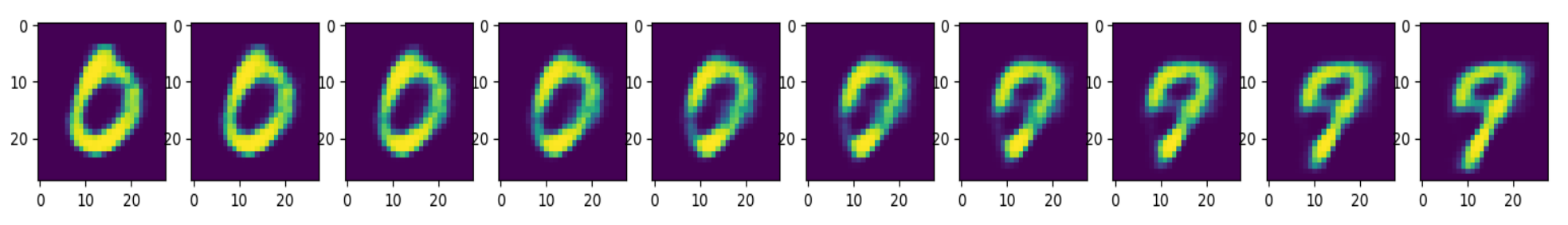
Simples interpolação no espaço latente - keras

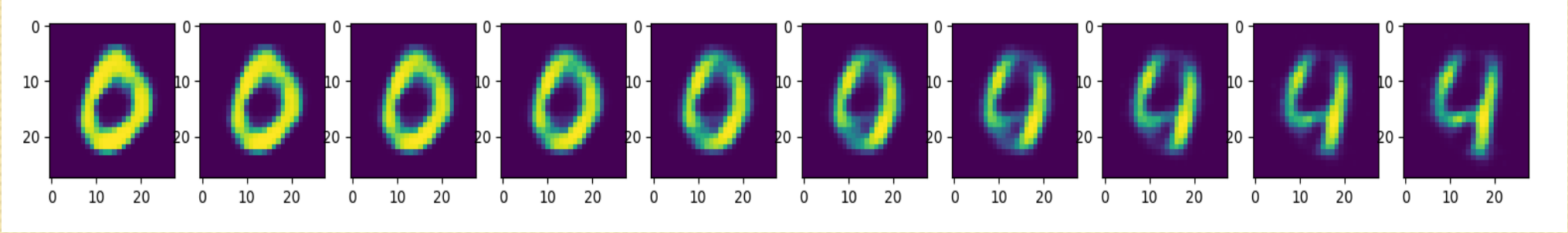


Simples interpolação no espaço latente – keras



Simples interpolação no espaço latente – keras





Convolutional AE – Keras example results

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

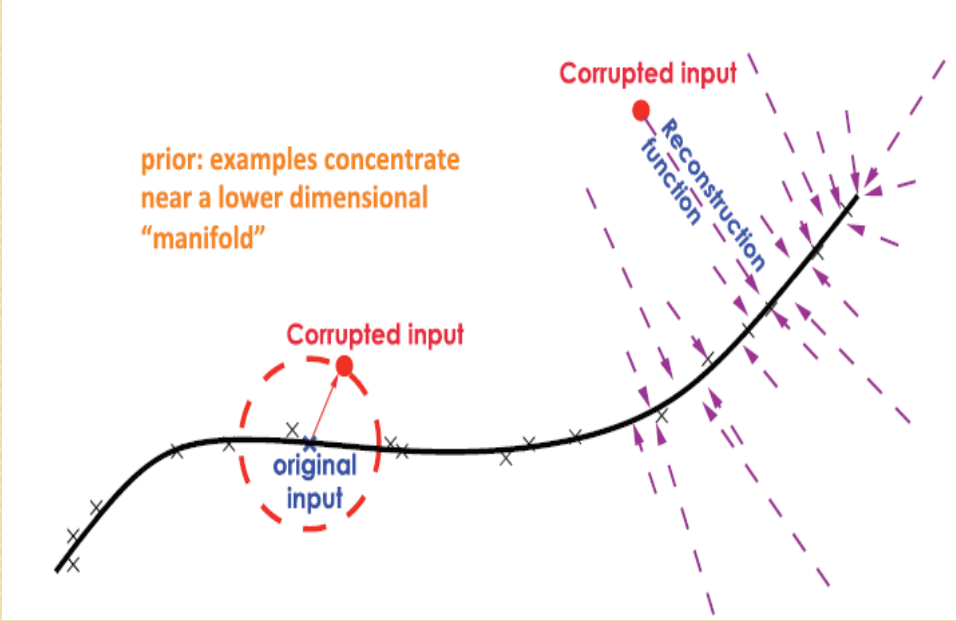
Autoencoder com filtragem de ruído

(*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*)

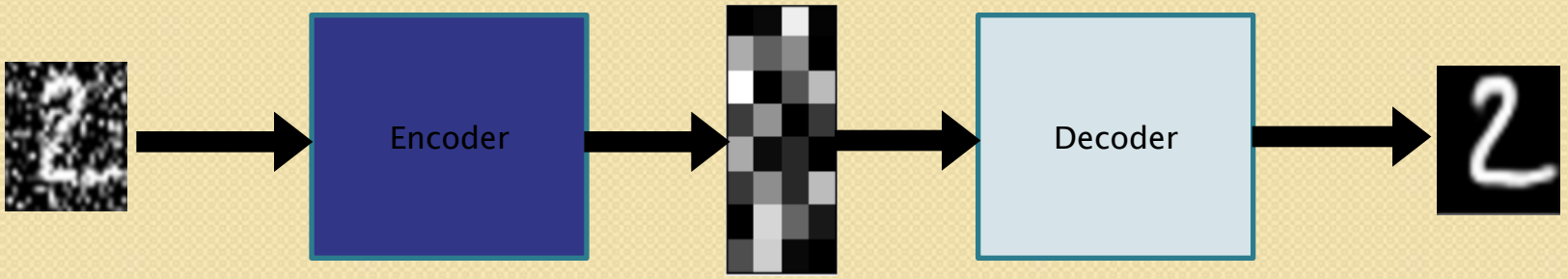


Representation Learning - A Review and New Perspectives (2013)

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



Entrada ruidosa

Representação no espaço latente

Entrada de-ruidosa

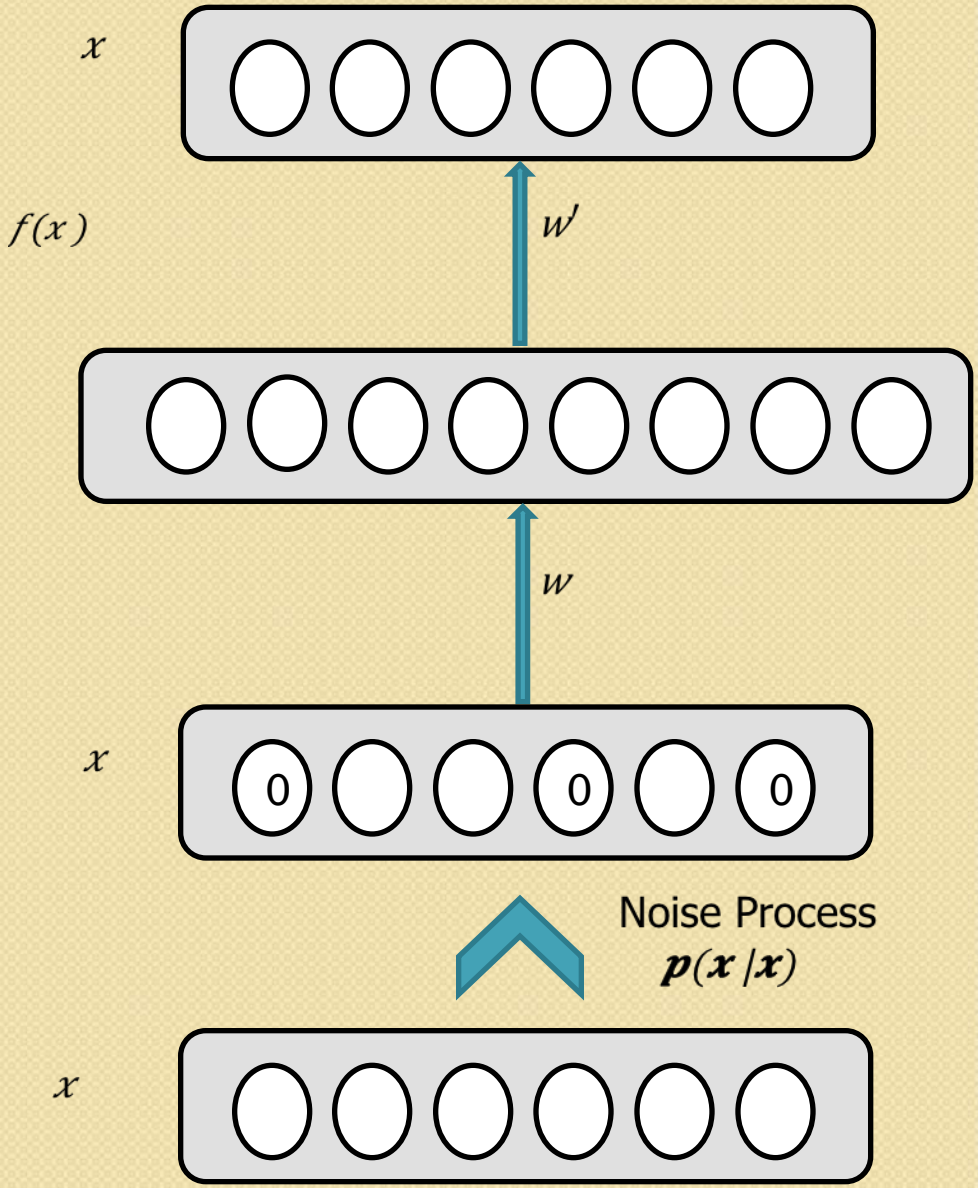
Denoising Autoencoders

- Reconstruction x computed from the corrupted input x.
- Loss function compares 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 prune to changes' model.

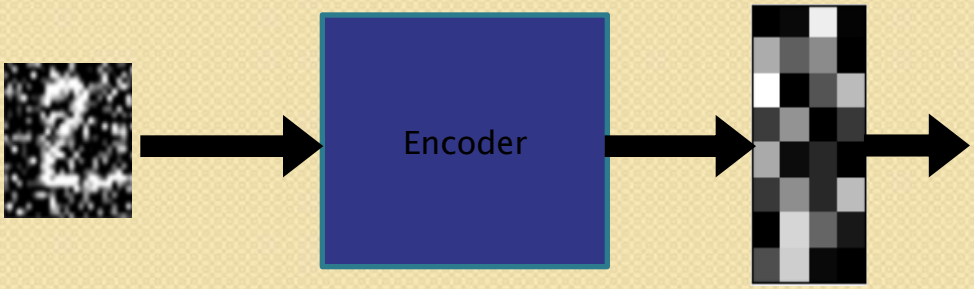
➢ We are forcing the hidden layer to learn a generalized structure of the data.



Denoising Autoencoders

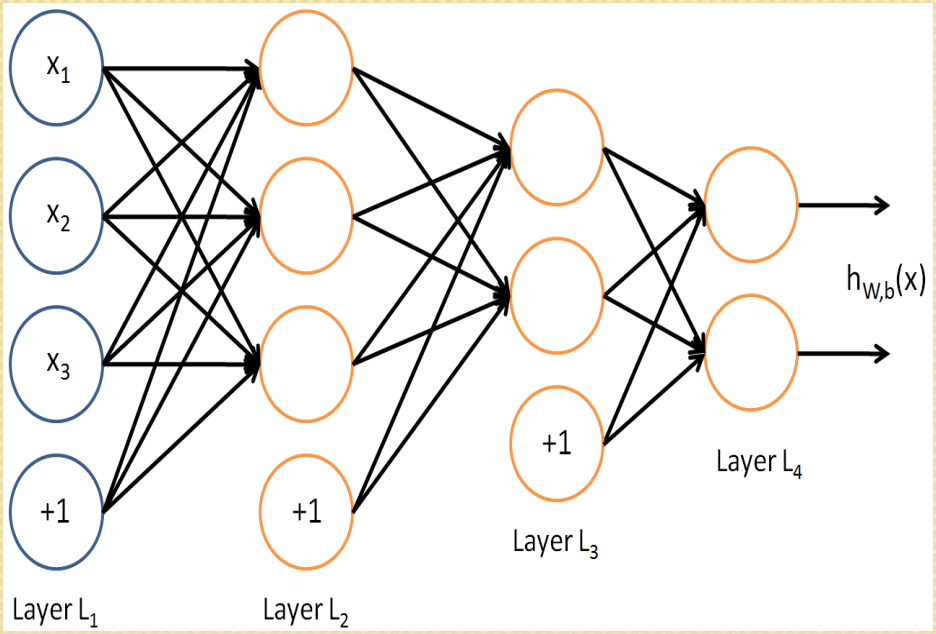
Caso de uso:

- Extrair representações robustas para um classificador



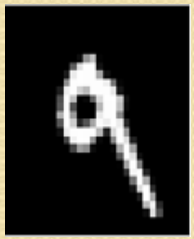
Entrada ruidosa

Representação no espaço latente



Processo - Denoising Autoencoders

Taken some input x



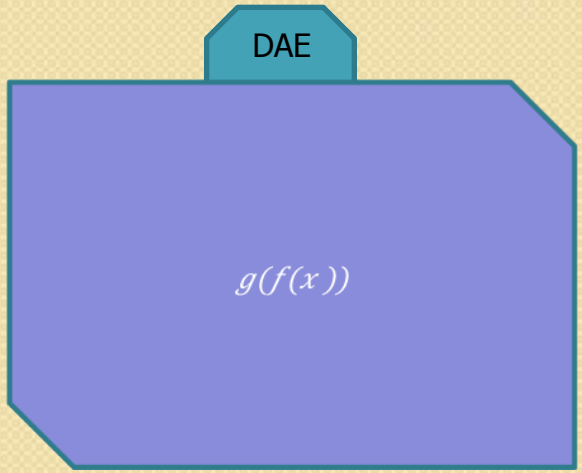
Apply Noise

x

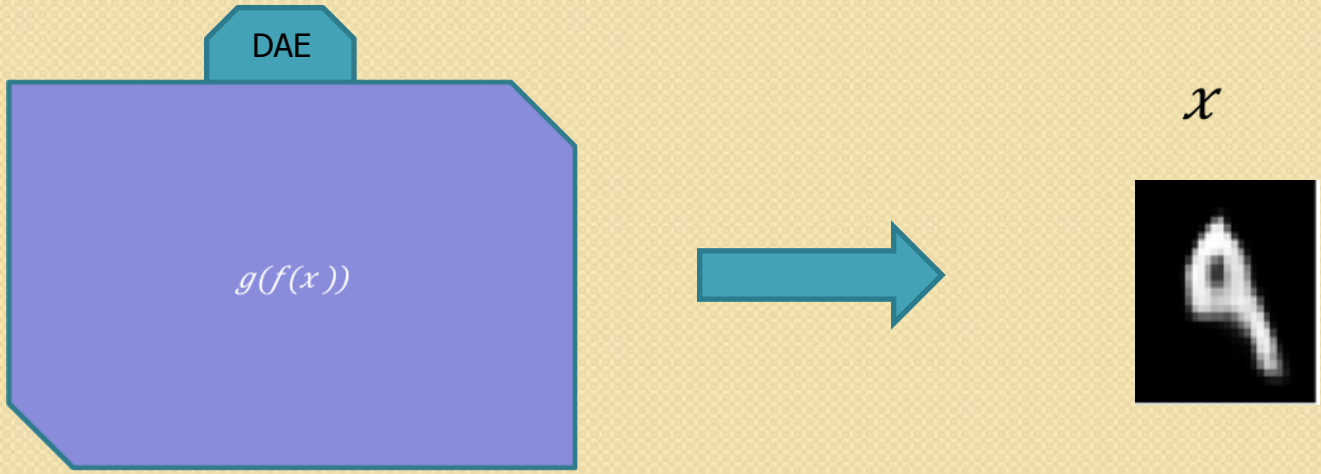
Processo Denoising Autoencoders

x

Encode And Decode



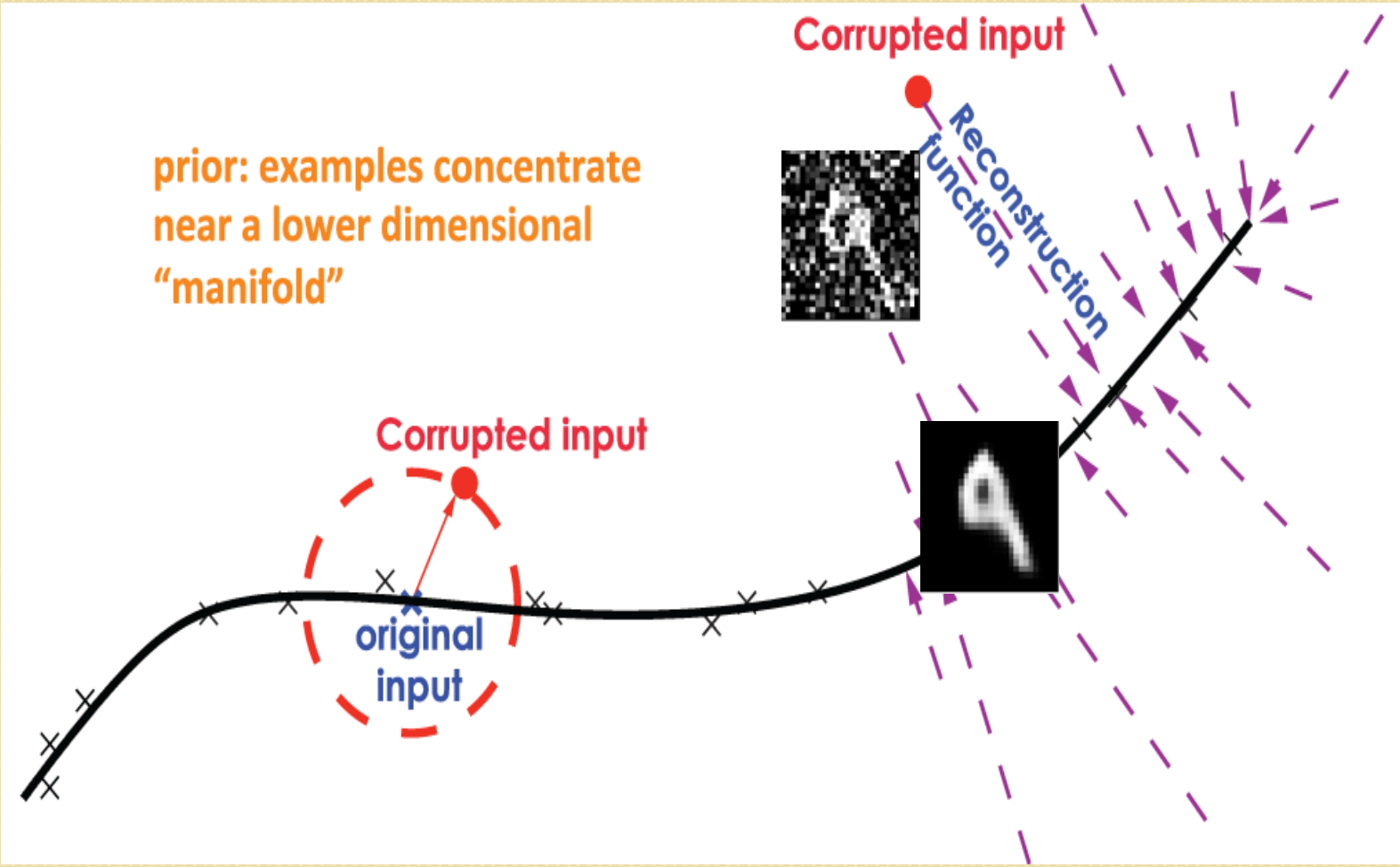
Processo - Denoising Autoencoders



Processo - Denoising Autoencoders

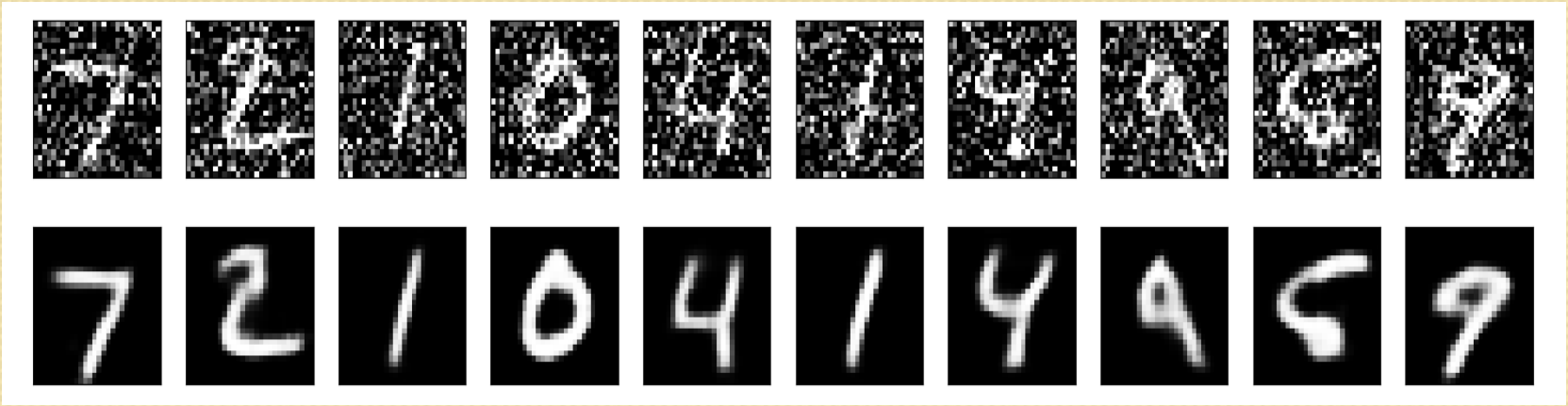
x Compare x

Processo - Denoising Autoencoders



Denoising convolutional AE – keras

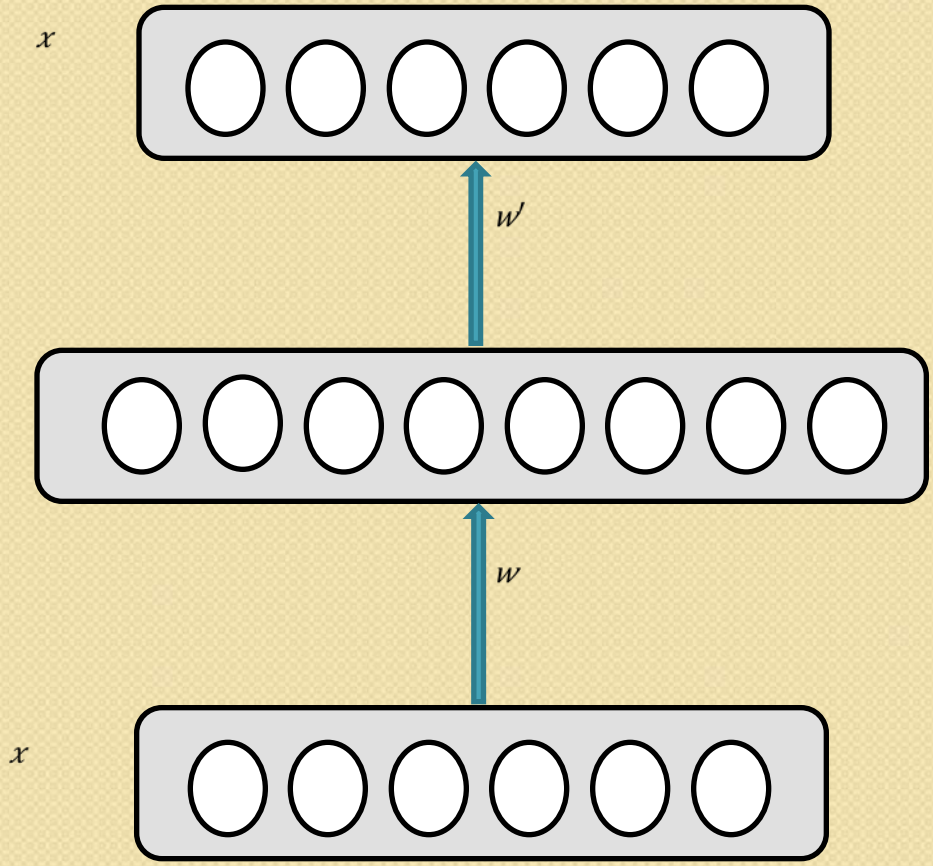
- 50 epochs.
- Noise factor 0.5
- 92% accuracy on validation set.



Contractive autoencoders

- Evitar características indesejadas

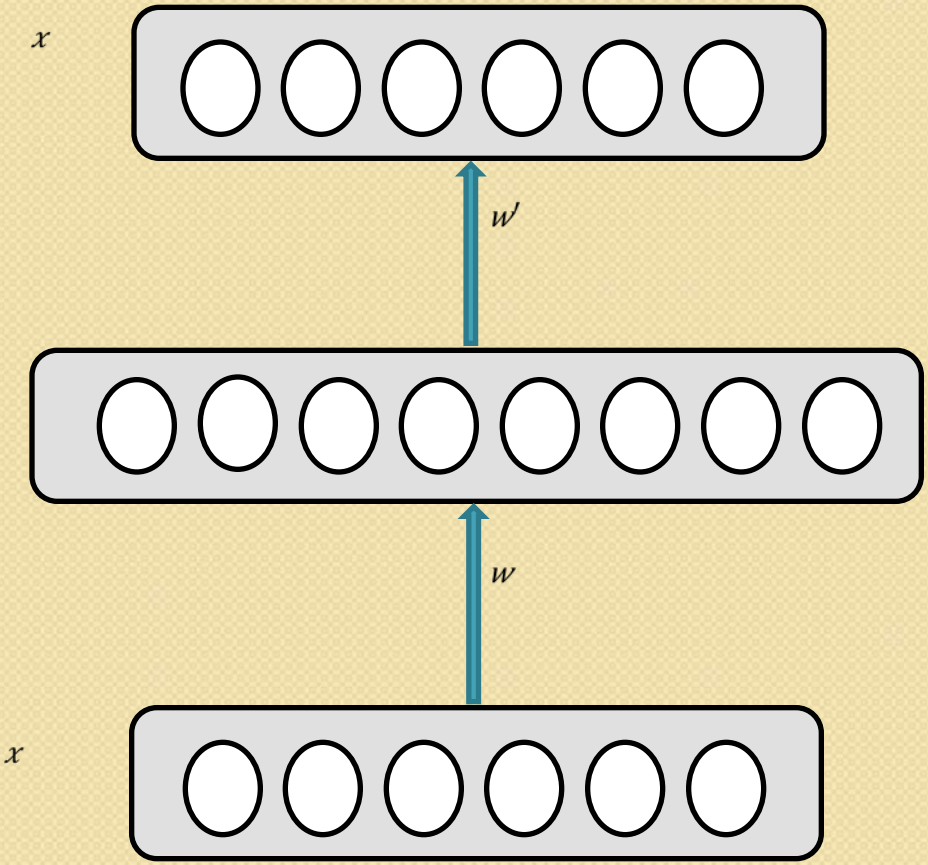
- Adicionar um termo de regularização $\Omega(x)$ à função de perda para limitar as representações na camada escondida



Contractive autoencoders

- – Idea: extrair características que representam apenas variações no conjunto de treinamento. Ficar invarriante a quaiquer outras variações

– Pontos próximos entre si no espaço de entrada devem manter essa propriedade no espaço latente



Autoencoder contrativa

(*contractive autoencoder*)

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

$$-\sum_{k=1}^{d}[x_k \log z_k + (1-x_k)\log(1-z_k)] + \left\| \frac{\partial h(x)}{\partial x} \right\|^2$$

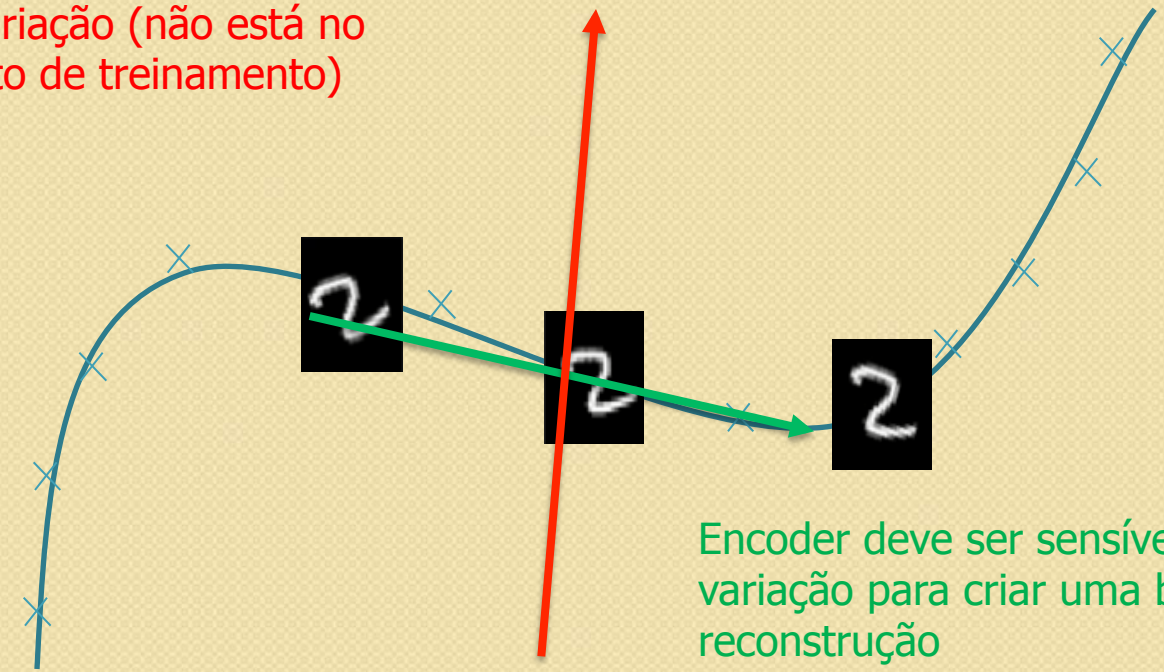
penalização

"mantenha boas representações" + "descarte todas as representações" = "mantenha apenas boas representações"

Contractive Auto-Encoders: Explicit Invariance during Feature Extraction (2011)

Contractive autoencoders

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



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

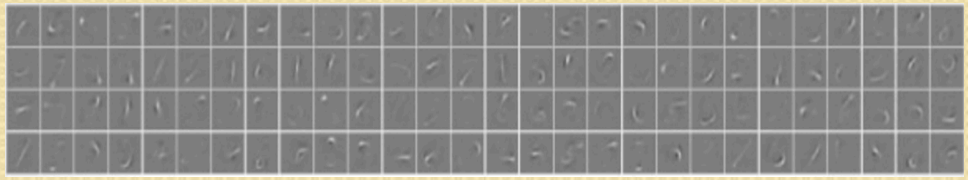
Autoencoder esparso
(sparse autoenconder)

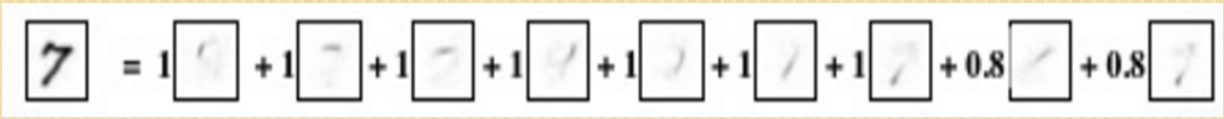
- Objetivo: apenas pequena quantidade de unidades da camada oculta seja ativada para cada padrão de entrada.

- 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

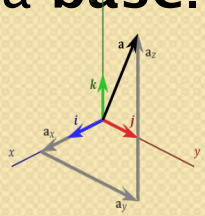
Autoencoder esparso
(sparse autoenconder)

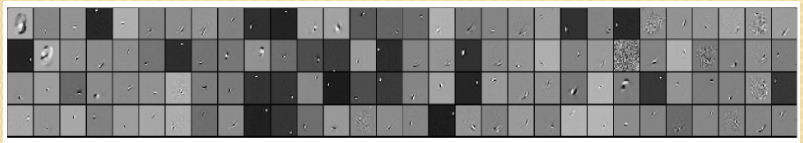
- Motivação biológica: visual córtex (V1)



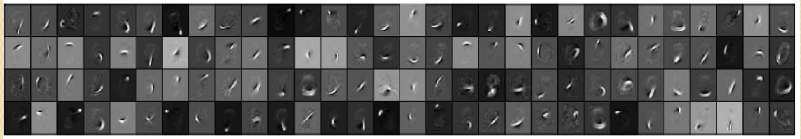


Exemplo: imagens de 28x28 pixels podem ser representadas por uma qtd. pequena de **códigos** a partir de uma **base**.

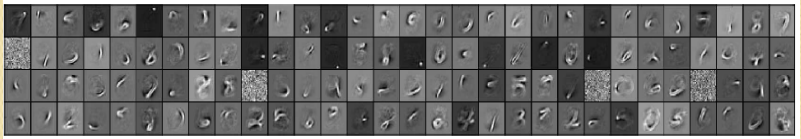




(a) $k = 70$



(b) $k = 40$

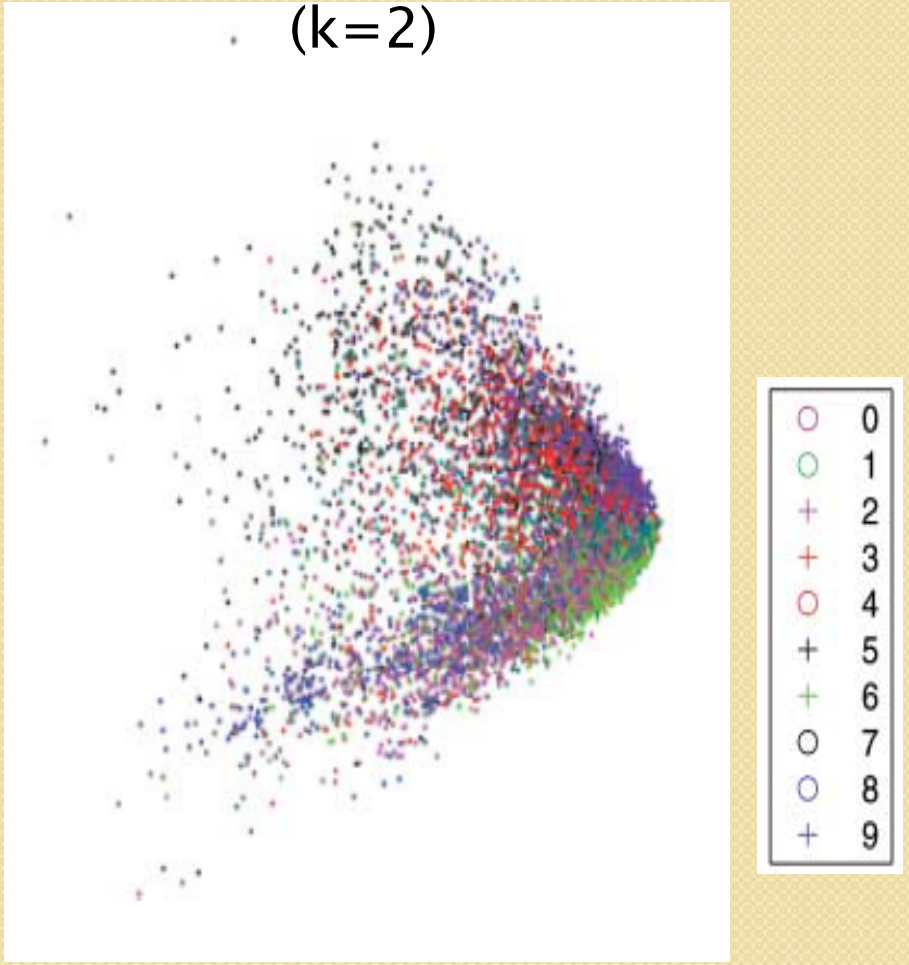


(c) $k = 25$

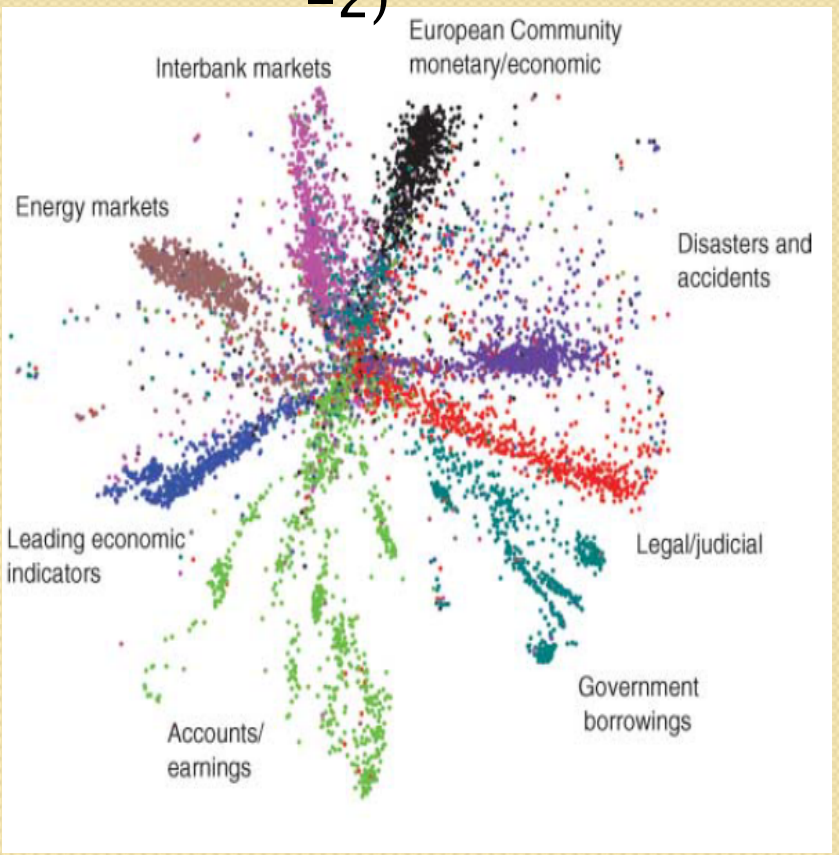
(d) $k = 10$

Aplicações (I) – redução de dimensionalidade

PCA (k=2)



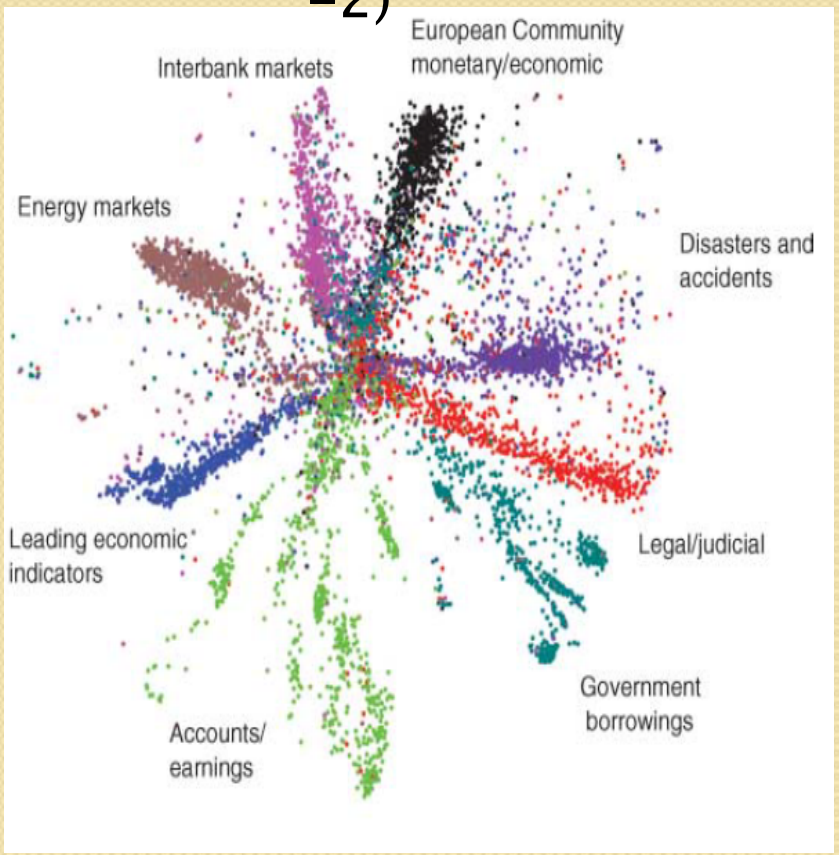
Autocodificadora (2000-500-250-125-2)

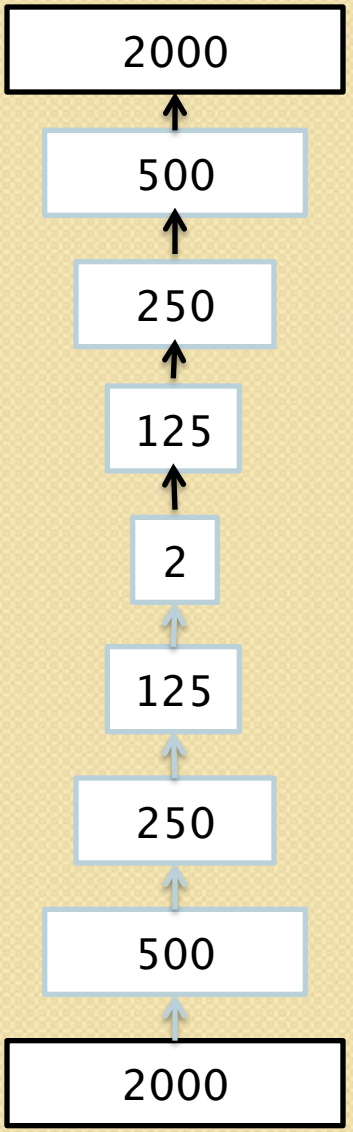


Reducing the Dimensionality of Data with Neural Networks (Science, 2006)

Aplicações (I) – redução de dimensionalidade

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

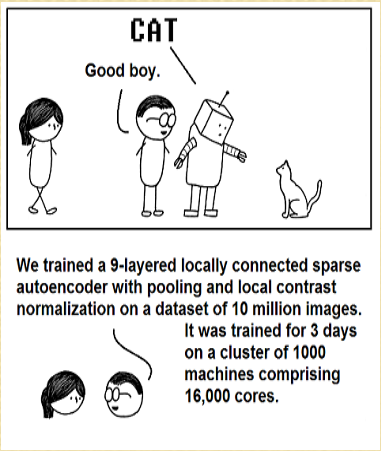


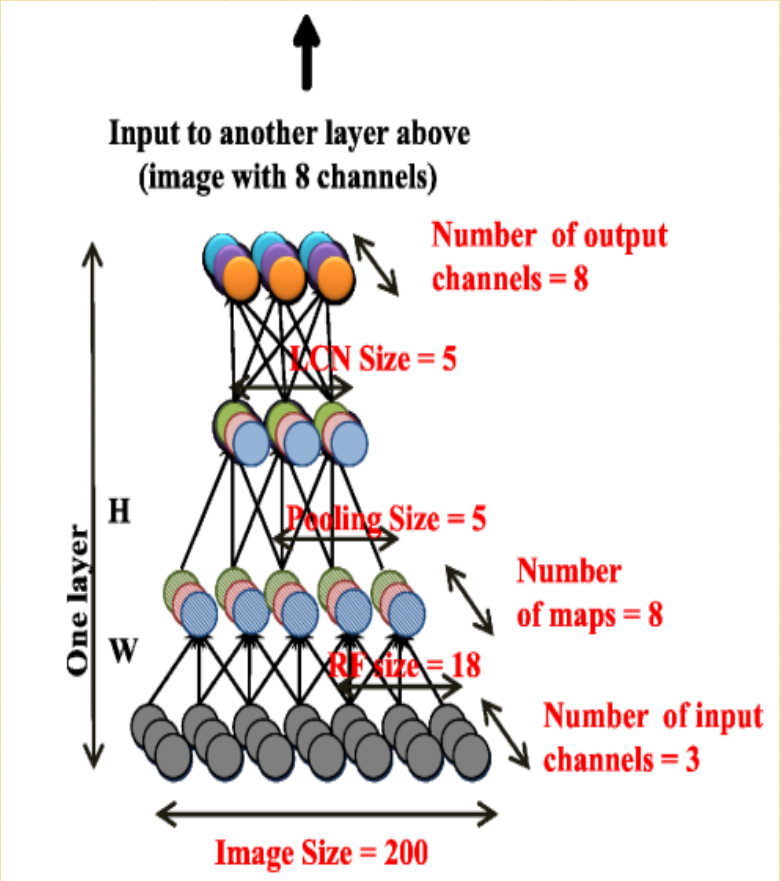


Reducing the Dimensionality of Data with Neural Networks (Science, 2006)

Aplicações (II) – aprendizado de conceitos

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





Le et al, Building high-level features using large-scale unsupervised learning, 2012

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)

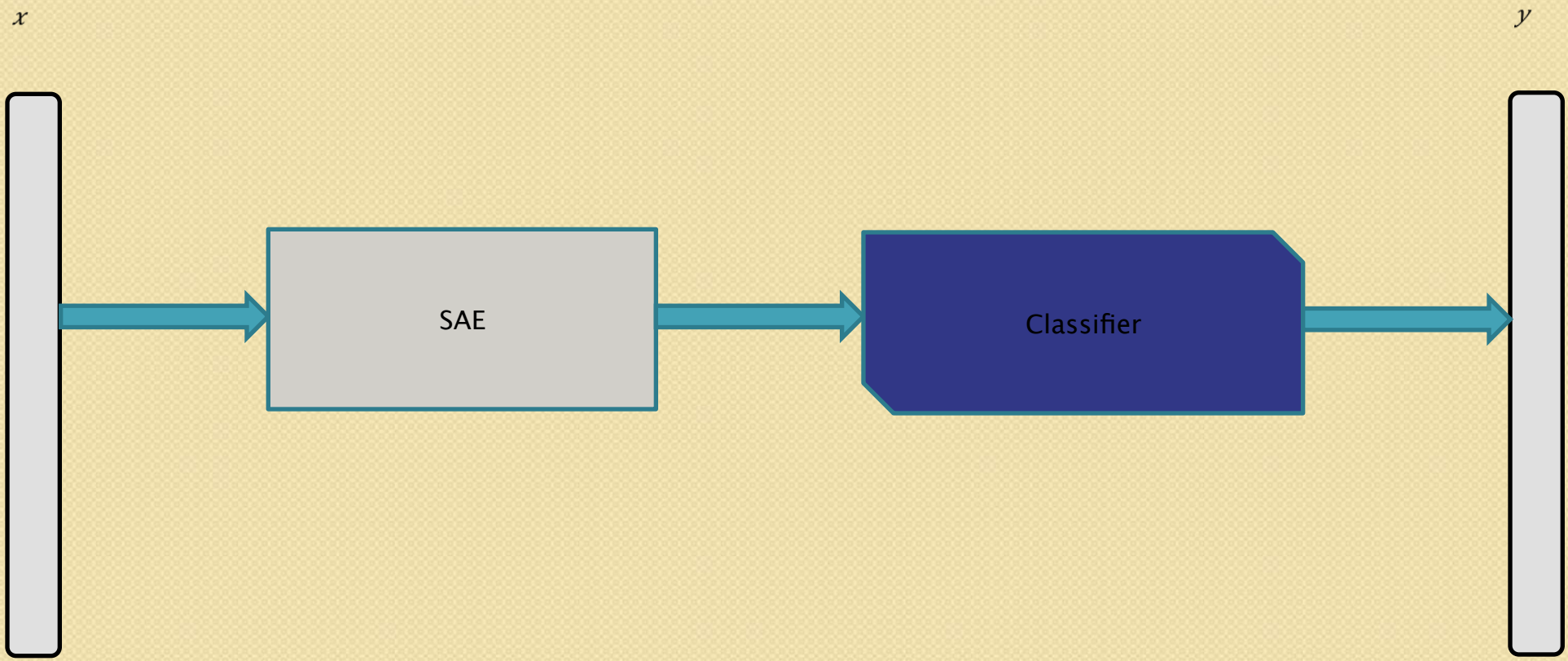
Stacked AE

- 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)

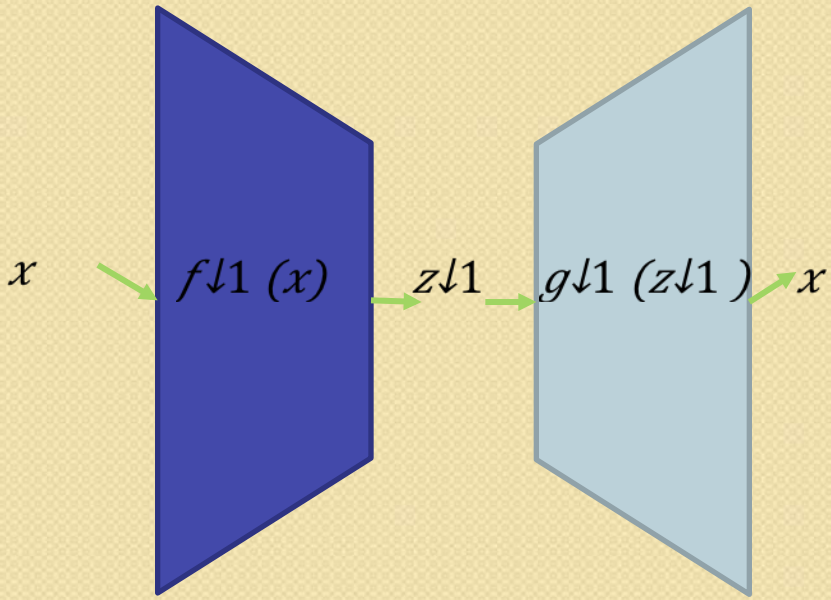
- Consiste de 2 etapas:
  - 1. Treinar cada camada AE uma após a outra
  - 2. Acoplar qualquer classificador no final (MLP/ SVM / RBF / SGB)

Stacked AE



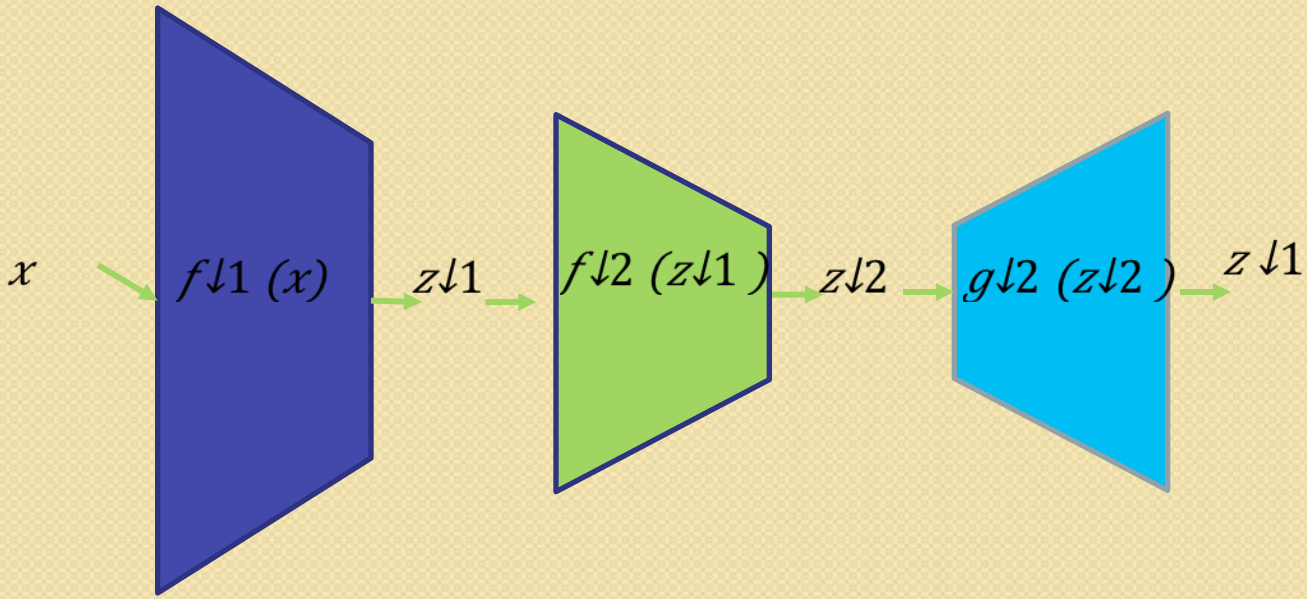
Stacked AE – treinamento

First Layer Training
(AE 1)



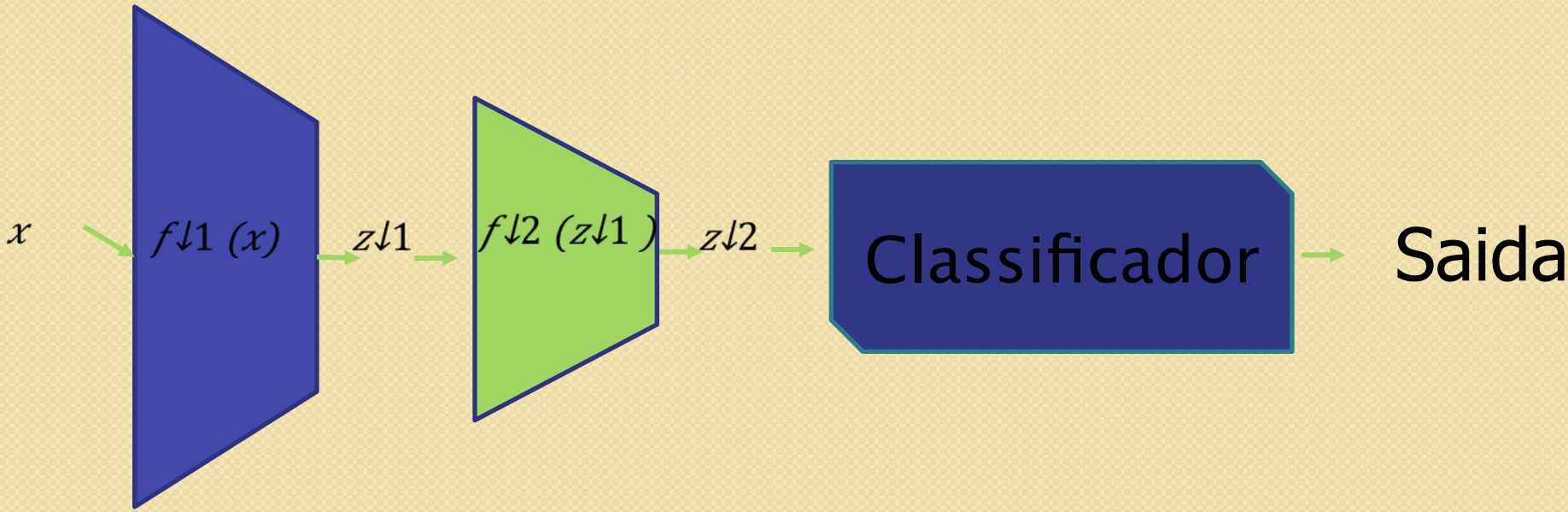
Stacked AE – treinamento

Second Layer Training (AE 2)



Stacked AE – treinamento

Adicione um classificador



Qual autoencoder?

- Ambos denoising AE e contractive AE funcionam bem!

References

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://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders
6. http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf
7. https://codeburst.io/deep-learning-types-and-autoencoders-a40ee6754663

Exemplo AE

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

Contractive autoencoders

- Definitions and reminders:
- – Frobenius norm (L2): $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$
- – Jacobian Matrix: $J_f(x) = \partial f(x)/\partial x =$

$$\begin{bmatrix} \partial f(x)_1/\partial x_1 & \cdots & \partial f(x)_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial f(x)_m/\partial x_1 & \cdots & \partial f(x)_m/\partial x_n \end{bmatrix}$$

Contractive autoencoders

- Our new loss function would be:
- $$L^*(x) = L(x) + \lambda \Omega(x)$$
- where $\Omega(x) = \|J_f(x)\|_F^2$ or simply: $\sum_{i,j} \left(\partial f(x)_j / \partial x_i \right)^2$

and where λ controls the balance of our reconstruction objective and the hidden layer "flatness".

Contractive autoencoders

- Our new loss function would be:
- $$L\uparrow^{*}(x) = L(x) + \lambda\Omega(x)$$
- $L(x)$ – would be an encoder that keeps good information ($\lambda \to 0$)
- $\Omega(x)$ – would be an encoder that throws away all information ($\lambda \to \infty$)

Combination would be an encoder that keeps **only** good information.