

Autoencoders

Prof. Jefersson A. dos Santos

jefersson@dcc.ufmg.br



DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

U F *m* G

Roteiro

Aulas anteriores

- Redes Convolucionais
- Aplicações de CNNs
 - Classificação, detecção e segmentação semântica
 - **Aprendizado supervisionado!**

Aula de hoje

- Autoencoders
- Sparse Autoencoders
- Convolutional Autoencoders
- Denoising Autoencoders
- Stacked Autoencoders

Roteiro

Aulas anteriores

- Redes Convolucionais
- Aplicações de CNNs
 - Classificação, detecção e segmentação semântica
 - **Aprendizado supervisionado!**

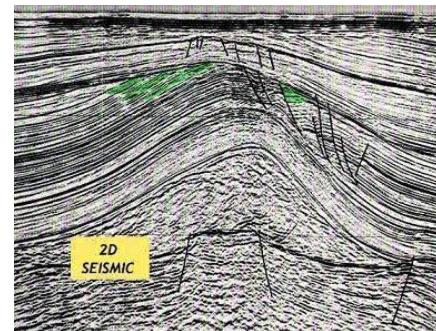
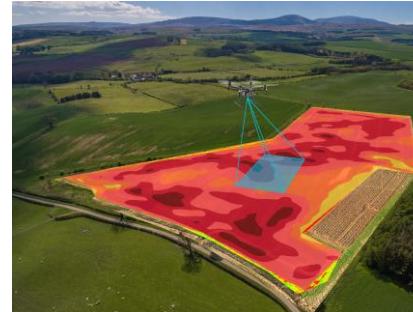
Aula de hoje

- Autoencoders
- Sparse Autoencoders
- Convolutional Autoencoders
- Denoising Autoencoders
- Stacked Autoencoders

Aprendizado não-supervisionado!

Motivação: *feature learning*

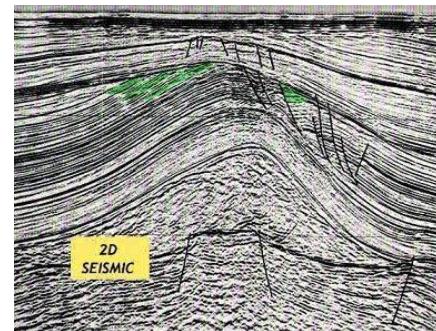
- DL exige muita amostra rotulada
- Muitas aplicações dependem de usuário especialista:
 1. Dados anotados são escassos
 2. Dados não-rotulados são abundantes



Motivação: *feature learning*

- DL exige muita amostra rotulada
- Muitas aplicações dependem de usuário especialista:
 - Dados anotados são escassos
 - Dados não-rotulados são abundantes

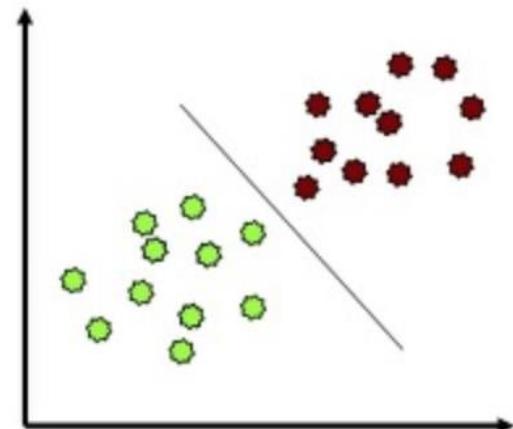
Go unsupervised!



Aprendizado não-supervisionado

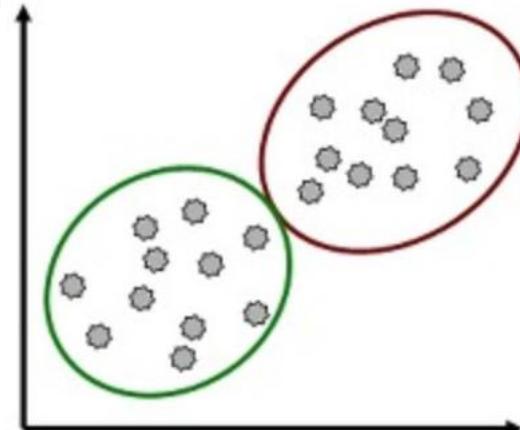
Aprendizado Supervisionado

- Para entender o contraste entre aprendizado Supervisionado e Não-supervisionado
- Na supervisionada, existe uma variável com status especial, a variável resposta Y, o label
 - Binária
 - Multi-classe
 - Contínua (regressão)
- As demais variáveis são as features.
- Modelos discriminativos: objetivo é **discriminar** entre os valores de Y
- Como inferir os valores da classe Y (0 ou 1?) num novo exemplo?
- Inferir como função das features x
- $P(Y=1 | x) = g(x)$
- Não nos interessa nada sobre x.
- Não interessa saber $P(x_1 > 3)$ ou $P(x_1 > x_2)$ ou ainda $P(x_1 > x_2 | x_3 > 0)$
- Só nos importamos com Y: como o rótulo Y responde a mudanças em x?



Aprendizado não-supervisionado

- Existem n variáveis Y_1, Y_2, \dots, Y_n (e vários exemplos)
- Interesse na distribuição conjunta das Y
- Não existe uma dicotomia Y versus features x
- Só existem as Y : são todas variáveis “resposta” mas não existe um conjunto de features ao qual elas respondem
- Interesse é descobrir estruturas não óbvias nos dados Y
- Tarefas:
 - Redução de dimensionalidade e visualização
 - Descoberta de clusters / grupos
 - Descoberta de estrutura (graphical Bayesian models)
 - Modelos generativos



Alguns dos desafios de alta dimensão

Maldição da dimensionalidade:

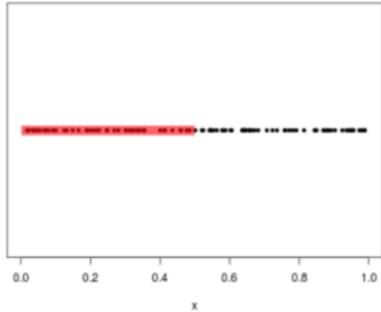
- Se refere a vários fenômenos que surgem ao analisar e organizar dados em espaços de alta dimensão
- Geralmente com centenas ou milhares de dimensões
- Não ocorrem em ambientes de baixa dimensão, como o espaço físico tridimensional da experiência cotidiana.
- Expressão foi cunhada por Richard E. Bellman ao considerar problemas na programação dinâmica.

Alguns dos desafios de alta dimensão

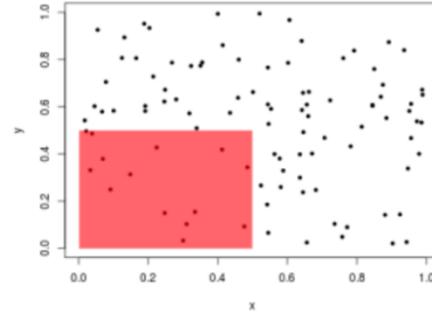
Maldição da dimensionalidade:

- Quando a dimensionalidade aumenta, o volume do espaço aumenta tão rapidamente que os dados disponíveis se tornam escassos

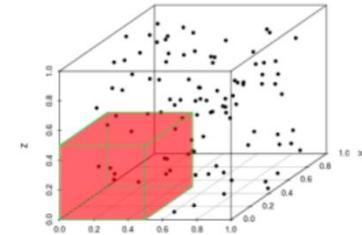
1-D: 42% of data captured.



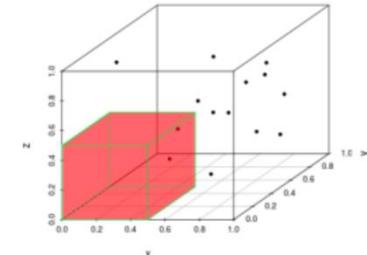
2-D: 14% of data captured.



3-D: 7% of data captured.



4-D: 3% of data captured.
 $t = 0$



Alguns dos desafios de alta dimensão

Maldição da dimensionalidade:

- Quando a dimensionalidade aumenta, o volume do espaço aumenta tão rapidamente que os dados disponíveis se tornam escassos
- Essa escassez é problemática para qualquer método que exija significância estatística.
 - Para obter um resultado estatisticamente sólido e confiável, a quantidade de dados necessários para suportar o resultado geralmente cresce exponencialmente com a dimensionalidade.
 - Geralmente dependem da detecção de áreas em que os objetos formam grupos com propriedades semelhantes;
- Todos os objetos parecem esparsos e diferentes de várias maneiras, o que impede que estratégias comuns de organização de dados sejam eficientes.

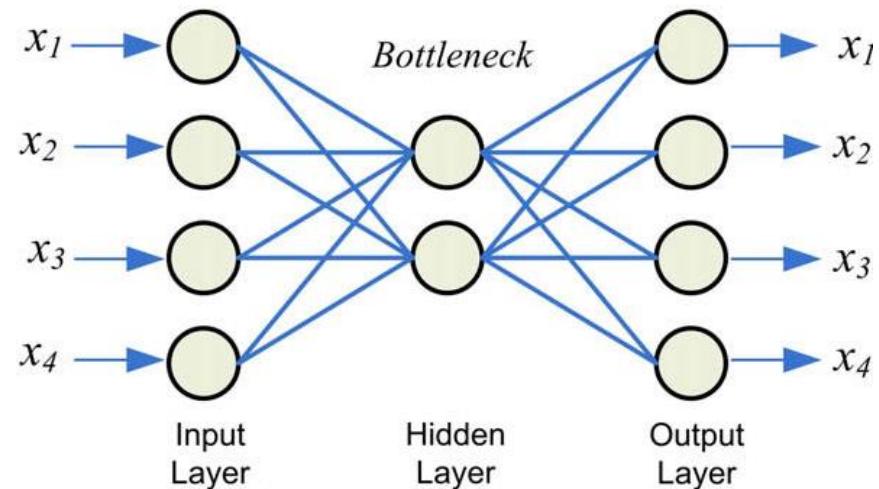
Autoencoders

Autoencoders

Uma rede neural que utiliza *backpropagation* para gerar valor de saída quase próximo ao valor de entrada.

Algumas propriedades:

- Não supervisionado
- Redução de dimensionalidade
- Aprendizado de *features* (end to end)
- *Feedforward network*

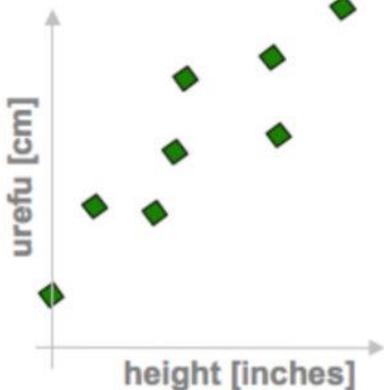


Redução de dimensionalidade

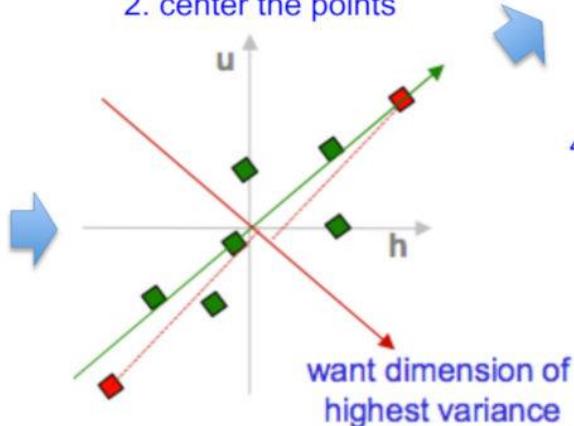
- Principal Component Analysis (PCA)
 - Aprende essencialmente uma transformação linear
 - Projeta os dados em outro espaço, onde vetores de projeções são definidos pela variação dos dados.
 - Poucos componentes concentram a maior parte da variação do conjunto de dados dos dados
 - Selecionar poucos componentes resulta em redução de dimensionalidade.

PCA in a nutshell

1. correlated hi-d data
("urefu" means "height" in Swahili)



2. center the points



3. compute covariance matrix

$$\begin{matrix} h \\ u \end{matrix} \begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \rightarrow \text{cov}(h, u) = \frac{1}{n} \sum_{i=1}^n h_i u_i$$

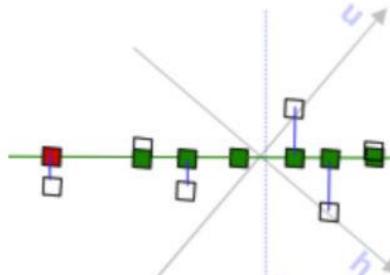
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \Lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \Lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

`eig(cov(data))`

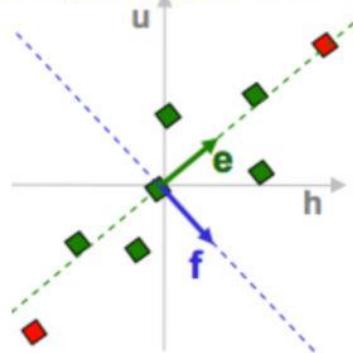
7. uncorrelated low-d data



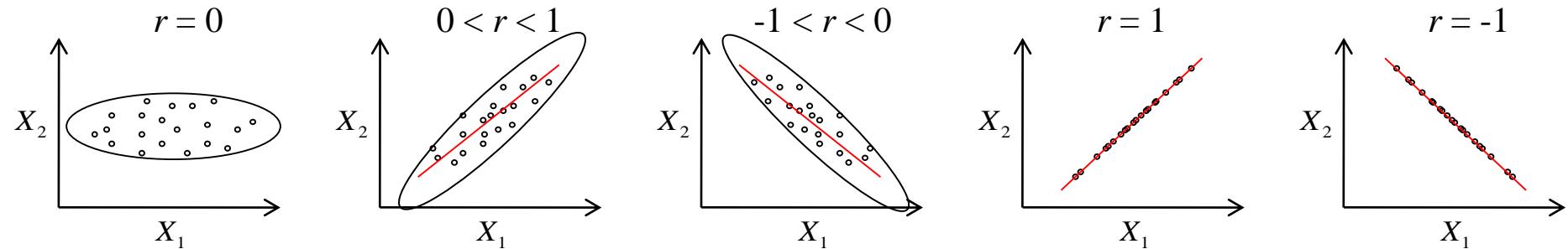
6. project data points to those eigenvectors

$$x_e = x^T e = \sum_{j=1}^a x_j e_j$$

5. pick $m < d$ eigenvectors w. highest eigenvalues



Associação entre Variáveis



$$r = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)\text{Var}(X_2)}}$$

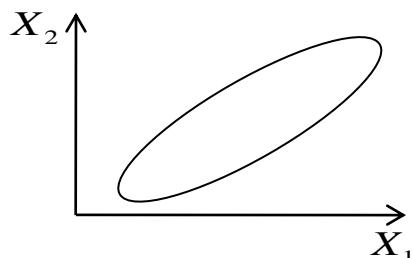
Coeficiente de Correlação (de Pearson)

1D: Quanto maior a **variância**, maior é a variabilidade e portanto maior a **informação** contida na variável. Num caso extremo, se a variância é zero, a variável não apresenta nenhuma informação a respeito do fenômeno por ela representada

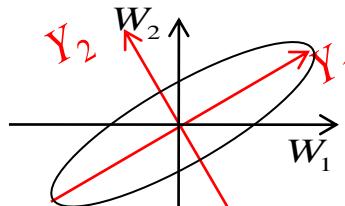
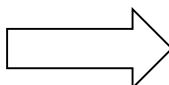
Por sua vez, a **covariância** (e a **correlação**) é interpretada como uma **redundância** em análises múltiplas (2 ou mais variáveis), já que a informação contida numa variável está parcialmente representada em outra. Num caso extremo, para que utilizar duas variáveis perfeitamente correlacionadas ($|r| = 1$), uma vez que, conhecendo-se o valor de uma, pode-se inferir com precisão o valor da outra?

Transformação por Componentes Principais (exemplo 2D)

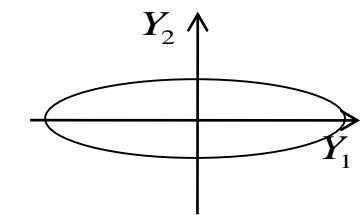
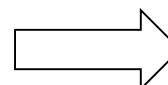
Como poderíamos eliminar a redundância entre X_1 e X_2 ?



$$Cov(X_1, X_2) > 0$$

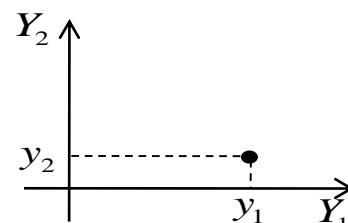
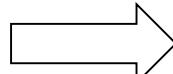
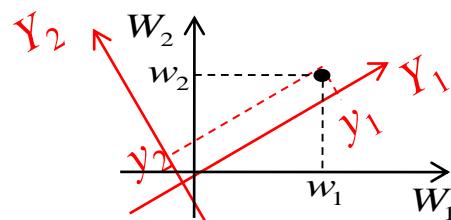


$$\begin{aligned} Cov(W_1, W_2) &= Cov(X_1, X_2) \\ W_j &= X_j - \bar{X}_j \end{aligned}$$



$$Cov(Y_1, Y_2) = 0$$

rotação com
preservação da
ortogonalidade
dos eixos



$$Y_1 = \alpha_{11}W_1 + \alpha_{12}W_2 \quad Var(Y_1) = \lambda_1$$

$$Y_2 = \alpha_{21}W_1 + \alpha_{22}W_2 \quad Var(Y_2) = \lambda_2$$

1^a CP

2^a CP

α são chamados de autovetores

λ são chamados de autovalores ($\lambda_1 > \lambda_2$)

$$Var(X_1) + Var(X_2) = \lambda_1 + \lambda_2$$

informação total é preservada

Autovalores e Autovetores

Os autovetores representam as transformações lineares aplicadas em m variáveis correlacionadas de modo a obter m variáveis não correlacionadas e podem ser representados na forma de uma matriz α ($m \times m$):

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \cdots & \alpha_{m1} \\ \alpha_{12} & \alpha_{22} & \cdots & \alpha_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{1m} & \alpha_{2m} & \cdots & \alpha_{mm} \end{bmatrix}$$

Nesta matriz, os autovetores estão organizados em colunas, ou seja, o 1º autovetor está na 1ª coluna, o 2º autovetor está na 2ª coluna e assim por diante.

Os autovalores representam a variância de cada componente (variável transformada) e podem ser representadas na forma de uma matriz diagonal λ ($m \times m$):

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix} \quad (\lambda_1 > \lambda_2 > \dots > \lambda_m)$$

Mas como são calculados os autovalores e autovetores?

Matriz de Variância-Covariância

Supondo que n amostras sejam avaliadas segundo m variáveis diferentes (atributos), podemos representar o conjunto de valores observados por um matriz \mathbf{X} , onde cada elemento x_{ij} representa o valor da i -ésima amostra para a j -ésima variável.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_m \\ x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{nm} & \cdots & x_{nm} \end{bmatrix}$$

A variância de cada variável e as covariâncias entre todos os pares de variáveis podem ser representadas através da matriz de variância-covariância $\Sigma_{\mathbf{X}}$ ($m \times m$):

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) & \cdots & Cov(X_1, X_m) \\ Cov(X_1, X_2) & Var(X_2) & \cdots & Cov(X_2, X_m) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_1, X_m) & Cov(X_2, X_m) & \cdots & Var(X_m) \end{bmatrix} \quad \begin{array}{c} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_m \end{array}$$
$$Cov(X_1, X_2) = \frac{\sum_{i=1}^n (X_{1i} - \bar{X}_1)(X_{2i} - \bar{X}_2)}{n-1}$$
$$Cov(X_k, X_l) = Cov(X_l, X_k)$$
$$Cov(X_k, X_k) = Var(X_k)$$

Cálculo dos Autovalores e Autovetores

Os autovalores (λ) e autovetores (α) são obtidos de modo a satisfazer a seguintes condições:

$$\Sigma_x \alpha = \lambda \alpha \quad \text{ou} \quad (\Sigma_x - \lambda I) \alpha = 0 \quad \text{sendo} \quad \alpha' \alpha = I \quad (\text{autovetores ortogonais})$$

Como os autovetores α são não nulos, então

$$|\Sigma_x - \lambda I| = 0 \quad \text{gera um polinômio de grau } m \text{ cujas raízes representam os } m \text{ autovalores } \lambda$$

Ordenam-se os autovalores do maior para o menor e para cada autovalor λ_k , calcula-se o autovetor α_k de modo que:

$$(\Sigma_x - \lambda_k I) \alpha_k = 0$$

Assim, os valores transformados para a componente principal k são calculados a partir da matriz X :

$$\mathbf{Y}_k = \alpha'_k \mathbf{X}'$$

Exemplo

Suponha que $\Sigma_x = \begin{bmatrix} 3,4 & -1,1 & 0,3 \\ -1,1 & 5,2 & -0,3 \\ 0,3 & -0,3 & 0,5 \end{bmatrix}$

Então $|\Sigma_x - \lambda I| = 0$

$$\begin{aligned}\Sigma_x - \lambda I &= \begin{bmatrix} 3,4 & -1,1 & 0,3 \\ -1,1 & 5,2 & -0,3 \\ 0,3 & -0,3 & 0,5 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \\ &= \begin{bmatrix} 3,4 - \lambda & -1,1 & 0,3 \\ -1,1 & 5,2 - \lambda & -0,3 \\ 0,3 & -0,3 & 0,5 - \lambda \end{bmatrix}\end{aligned}$$

$$\begin{aligned}|\Sigma_x - \lambda I| &= (3,4 - \lambda)(5,2 - \lambda)(0,5 - \lambda) + 2(-1,1)(-0,3)(0,3) - (3,4 - \lambda)(-0,3)^2 - (5,2 - \lambda)(0,3)^2 - (0,5 - \lambda)(-1,1)^2 \\ &= (3,4 - \lambda)(5,2 - \lambda)(0,5 - \lambda) + 0,198 - 0,306 + 0,09\lambda - 0,468 + 0,09\lambda - 0,605 + 1,21\lambda \\ &= (3,4 - \lambda)(5,2 - \lambda)(0,5 - \lambda) + 1,39\lambda - 1,181 \\ &= (17,68 - 8,6\lambda + \lambda^2)(0,5 - \lambda) + 1,39\lambda - 1,181 \\ &= -\lambda^3 + 9,1\lambda^2 - 20,59\lambda + 7,659 = 0\end{aligned}$$

Encontrando as raízes do polinômio, tem-se

$$\lambda_1 = 5,7517 \quad \lambda_2 = 2,8871 \quad \lambda_3 = 0,4612$$

Exemplo

Agora, para cada autovalor, calcula-se o autovetor correspondente de modo que $\Sigma_x \alpha = \lambda \alpha$
Demonstrando para o primeiro autovalor $\lambda_1 = 5,7517$

$$\begin{bmatrix} 3,4 & -1,1 & 0,3 \\ -1,1 & 5,2 & -0,3 \\ 0,3 & -0,3 & 0,5 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{13} \end{bmatrix} = 5,7517 \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{13} \end{bmatrix} \quad \begin{aligned} -2,3517\alpha_{11} - 1,1\alpha_{12} + 0,3\alpha_{13} &= 0 \\ -1,1\alpha_{11} - 0,5517\alpha_{12} - 0,3\alpha_{13} &= 0 \\ 0,3\alpha_{11} - 0,3\alpha_{12} - 5,2517\alpha_{13} &= 0 \end{aligned}$$
$$\alpha_{11} = 5,6657\alpha_{13}$$
$$\alpha_{12} = -11,8400\alpha_{13}$$

Repetindo para os demais autovalores, chega-se a:

$$\alpha = \begin{bmatrix} 5,6657\alpha_{13} & 15,4202\alpha_{23} & -0,0858\alpha_{33} \\ -11,8400\alpha_{13} & 7,4633\alpha_{23} & 0,0434\alpha_{33} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} \end{bmatrix}$$

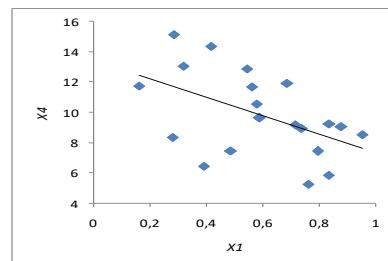
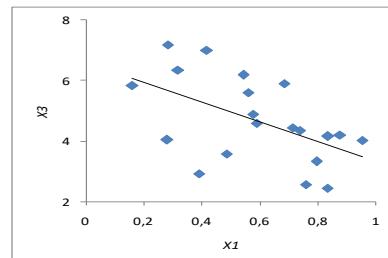
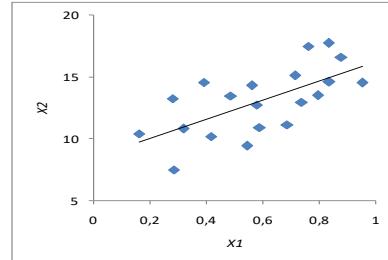
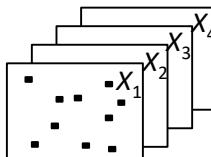
Como os autovetores são ortogonais então $\alpha' \alpha = \mathbf{I}$

Assim $\alpha_{13} = 0,0760$ $\alpha_{23} = 0,0583$ $\alpha_{33} = 0,9954$

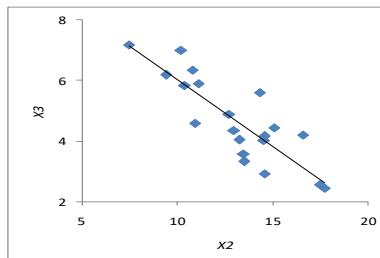
$$\alpha = \begin{bmatrix} 0,43039784 & 0,89858539 & -0,08545169 \\ -0,89943698 & 0,43491207 & 0,04318105 \\ 0,07596583 & 0,05827338 & 0,99540615 \end{bmatrix}$$

Análise de Componentes Principais

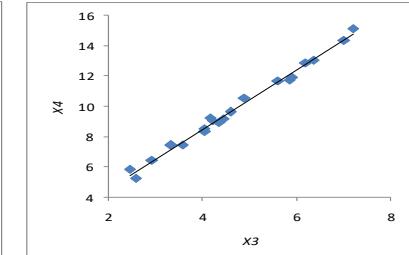
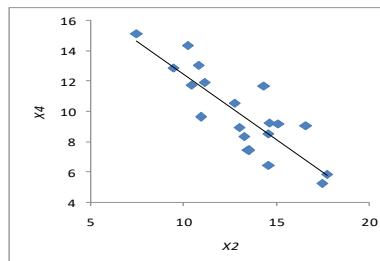
| X_1 | X_2 | X_3 | X_4 |
|--------|---------|--------|---------|
| 0,1596 | 10,4043 | 5,8457 | 11,7544 |
| 0,4160 | 10,1971 | 6,9976 | 14,3787 |
| 0,6841 | 11,1176 | 5,8942 | 11,9172 |
| 0,5421 | 9,4291 | 6,1867 | 12,8827 |
| 0,7133 | 15,0980 | 4,4443 | 9,1690 |
| 0,7587 | 17,4789 | 2,5728 | 5,2574 |
| 0,2824 | 7,4640 | 7,1848 | 15,1390 |
| 0,8336 | 14,5949 | 4,1581 | 9,2457 |
| 0,5752 | 12,7215 | 4,8841 | 10,5356 |
| 0,8308 | 17,7273 | 2,4392 | 5,8491 |
| 0,9510 | 14,5368 | 4,0345 | 8,5466 |
| 0,2796 | 13,2739 | 4,0397 | 8,3507 |
| 0,7357 | 12,9786 | 4,3470 | 8,9358 |
| 0,8759 | 16,5915 | 4,2100 | 9,0698 |
| 0,3904 | 14,5826 | 2,9220 | 6,4501 |
| 0,5584 | 14,3165 | 5,5950 | 11,6657 |
| 0,4849 | 13,4711 | 3,5743 | 7,4431 |
| 0,3169 | 10,8015 | 6,3426 | 13,0677 |
| 0,5869 | 10,9385 | 4,5867 | 9,6230 |
| 0,7940 | 13,5060 | 3,3238 | 7,4271 |



| Matriz de Variância-Covariância | | | |
|---------------------------------|---------|---------|---------|
| 0,0516 | 0,4024 | -0,1683 | -0,3171 |
| 0,4024 | 7,3698 | -3,2402 | -6,3600 |
| -0,1683 | -3,2402 | 1,9737 | 3,8806 |
| -0,3171 | -6,3600 | 3,8806 | 7,6995 |

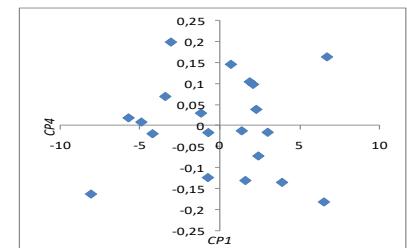
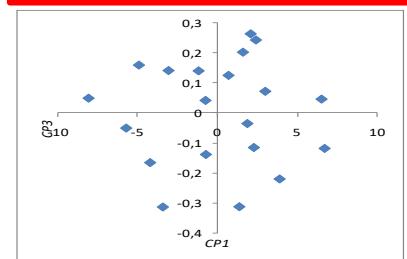
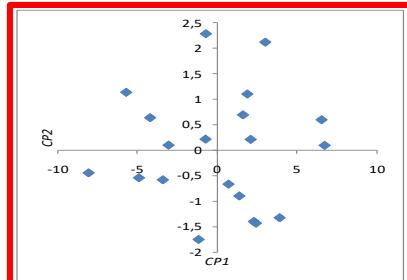


| Autovalores | | | |
|-------------|--------|---------|---------|
| 15,7615 | 1,2900 | 0,0310 | 0,0121 |
| Autovetores | | | |
| 0,0340 | 0,0583 | 0,9481 | 0,3108 |
| 0,6485 | 0,7582 | -0,0670 | -0,0087 |
| -0,3437 | 0,2787 | -0,2837 | 0,8507 |
| -0,6784 | 0,5865 | 0,1272 | -0,4238 |

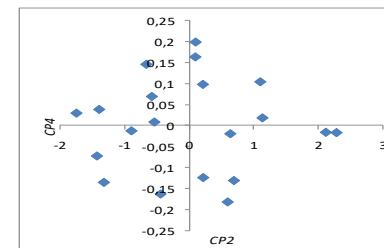
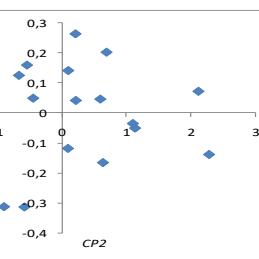


Análise de Componentes Principais

| CP_1 | CP_2 | CP_3 | CP_4 |
|---------|---------|---------|---------|
| -3,4404 | -0,5891 | -0,3154 | 0,0689 |
| -5,7422 | 1,1290 | -0,0515 | 0,0181 |
| -3,0872 | 0,0914 | 0,1410 | 0,1980 |
| -4,9424 | -0,5494 | 0,1594 | 0,0082 |
| 1,8577 | 1,0951 | -0,0362 | 0,1037 |
| 6,7000 | 0,0872 | -0,1194 | 0,1628 |
| -8,0993 | -0,4530 | 0,0486 | -0,1626 |
| 1,5819 | 0,6859 | 0,2024 | -0,1305 |
| -0,7664 | 0,2092 | 0,0411 | -0,1235 |
| 6,5079 | 0,5895 | 0,0455 | -0,1814 |
| 2,0649 | 0,2042 | 0,2638 | 0,0977 |
| 1,3543 | -0,9060 | -0,3144 | -0,0126 |
| 0,6757 | -0,6745 | 0,1249 | 0,1452 |
| 2,9795 | 2,1135 | 0,0716 | -0,0161 |
| 3,8802 | -1,3335 | -0,2217 | -0,1349 |
| -0,7436 | 2,2786 | -0,1397 | -0,0168 |
| 2,2647 | -1,4066 | -0,1165 | 0,0383 |
| -4,2393 | 0,6300 | -0,1669 | -0,0195 |
| -1,2009 | -1,7602 | 0,1400 | 0,0294 |
| 2,3949 | -1,4413 | 0,2433 | -0,0723 |

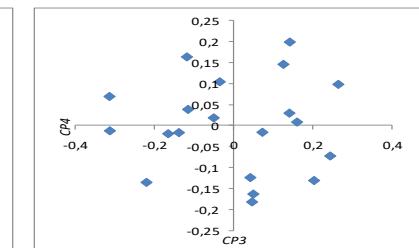


| Matriz de Variância-Covariância | | | | |
|---------------------------------|--------|--------|--------|---|
| 15,7615 | 0 | 0 | 0 | 0 |
| 0 | 1,2900 | 0 | 0 | 0 |
| 0 | 0 | 0,0310 | 0 | 0 |
| 0 | 0 | 0 | 0,0121 | |



A primeira componente guarda 92,2% da variação total

As 2 primeiras CP, acumulam 99,7% da variação total



PCA example: Eigen Faces

input: dataset of N face images



face: $K \times K$ bitmap of pixels

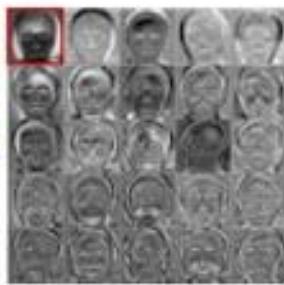


"unfold" each bitmap to
 K^2 -dimensional vector

arrange in a matrix
each face = column

...
 $K^2 \times N$

can visualize
eigenvectors:
 m "aspects"
of prototypical
facial features



"fold" into a $K \times K$ bitmap



PCA

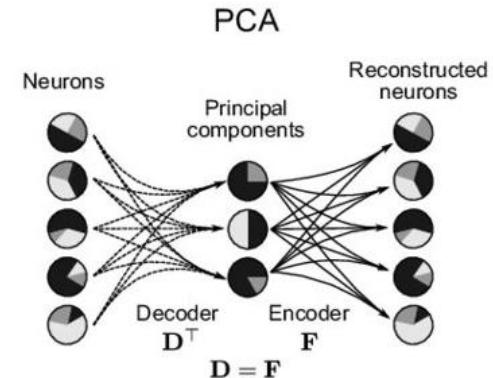
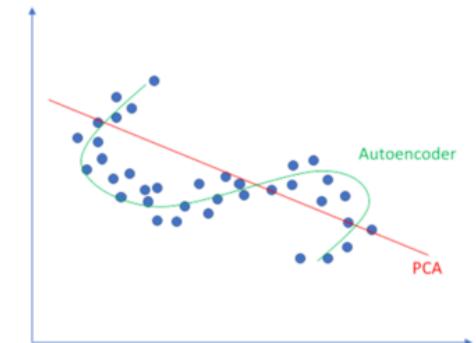
$K^2 \times m$

set of m eigenvectors
each is K^2 -dimensional

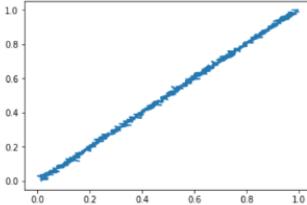
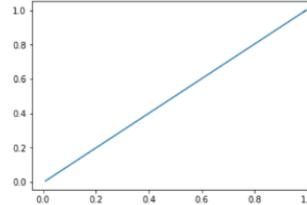
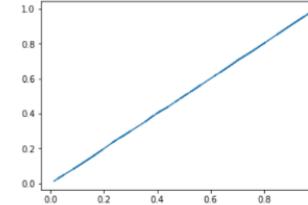
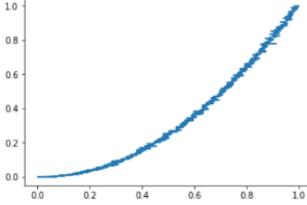
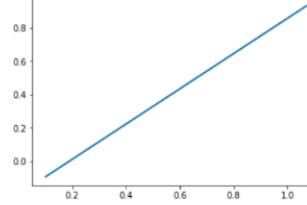
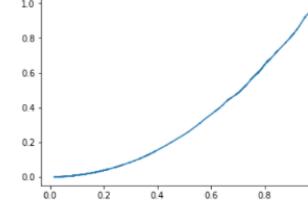
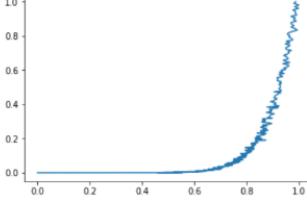
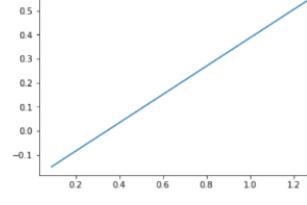
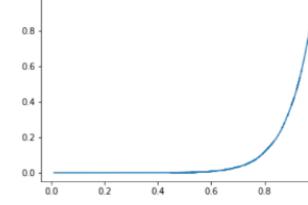
AE vs PCA

- Vantagens do AE:
 - Pode aprender transformações não lineares
 - uma função de ativação não linear e várias camadas
 - Não precisa aprender camadas densas
 - Pode usar camadas convolucionais
 - É mais eficiente
 - Pode fazer uso de camadas pré-treinadas de outro modelo

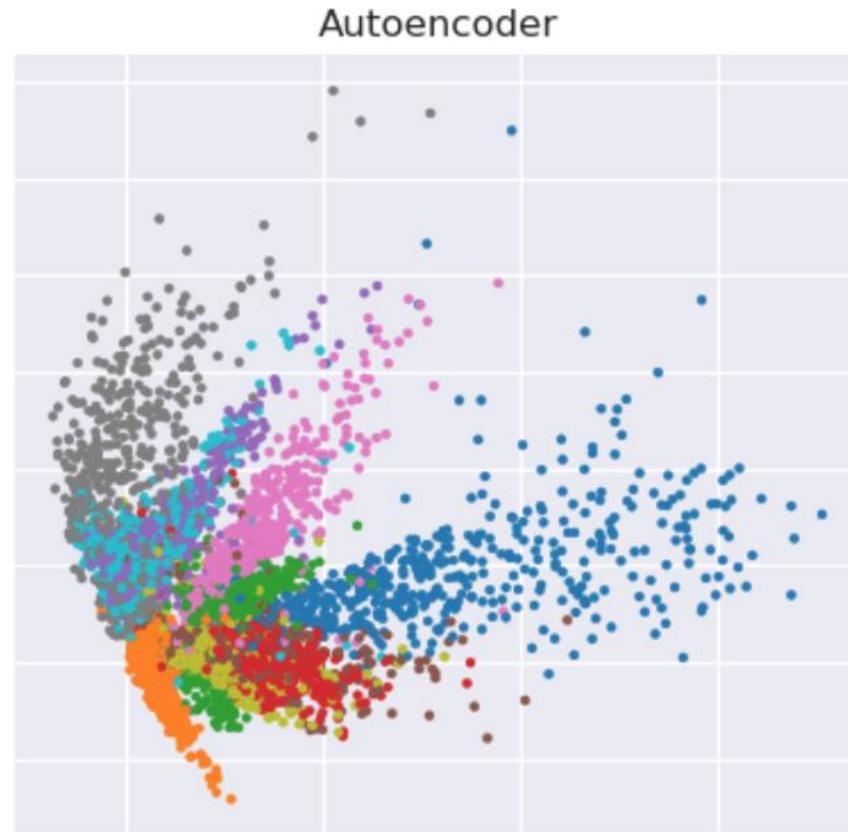
Linear vs nonlinear dimensionality reduction



AE vs PCA

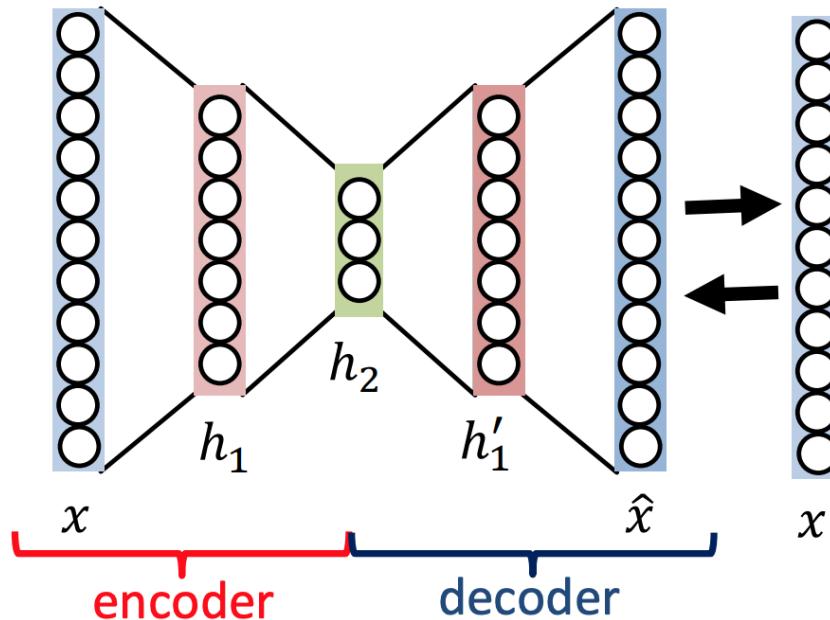
| Function | Feature Space | PCA Reconstruction | Auto Encoder Reconstruction |
|------------|--|---|--|
| $y=mx+c$ |  |  |  |
| $y=mx^2+c$ |  |  |  |
| $y=mx^8+c$ |  |  |  |

AE vs PCA



Autoencoders

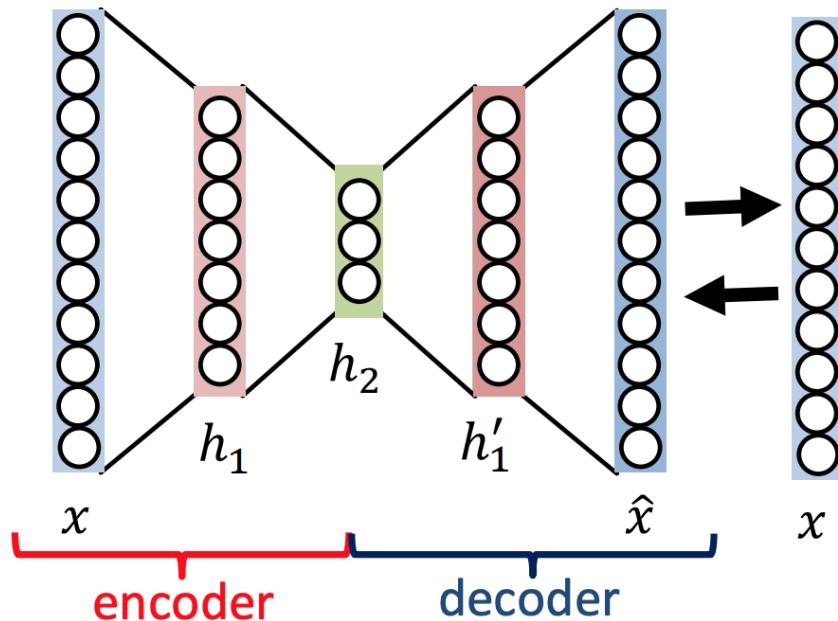
- Visa reproduzir a entrada na saída.
- Compreende um codificador que compacta os dados seguido por um decodificador que recupera a dimensão original.



Autoencoders

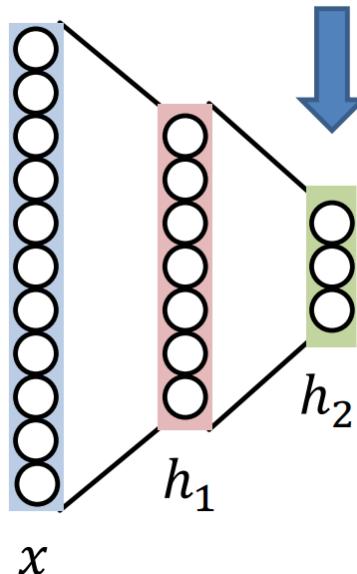
- A função de perda está relacionada à similaridade entrada \leftrightarrow saída, mais uma regularização(por exemplo, L2):

$$L(\mathbf{W}) = \frac{1}{N} \sum_i \|x_i - \hat{x}_i\|^2 + \lambda R(\mathbf{W})$$



Autoencoders

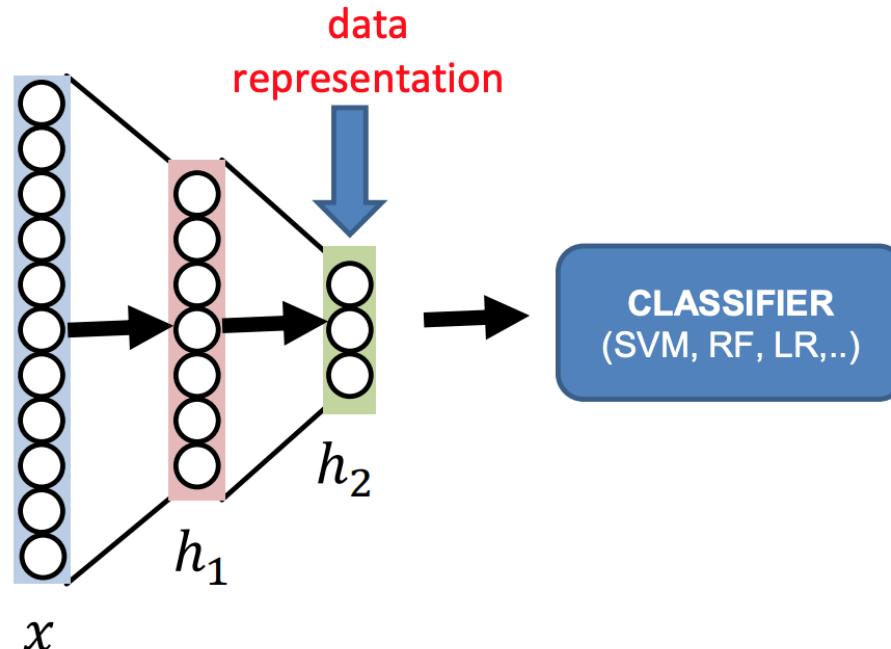
- Na verdade, não estamos interessados no resultado, mas no “gargalo” que captura as características mais salientes dos dados de treinamento.



Learning
features!

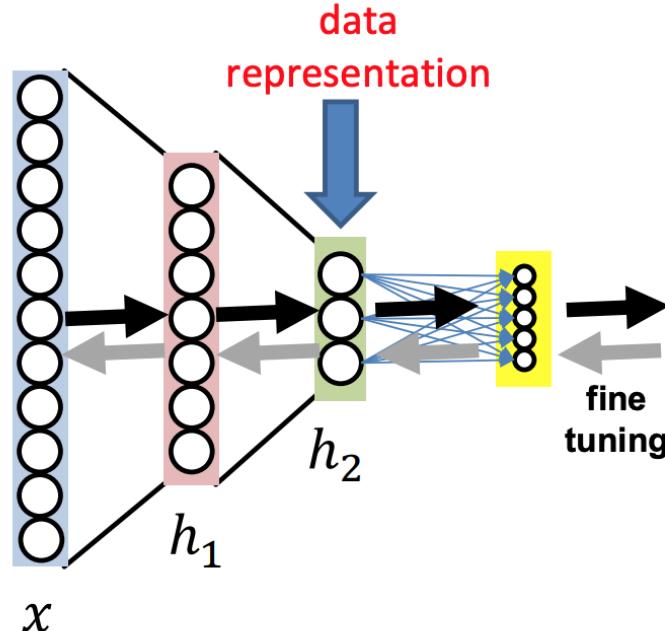
Autoencoders

- Na verdade, não estamos interessados no resultado, mas no “gargalo” que captura as características mais salientes dos dados de treinamento.



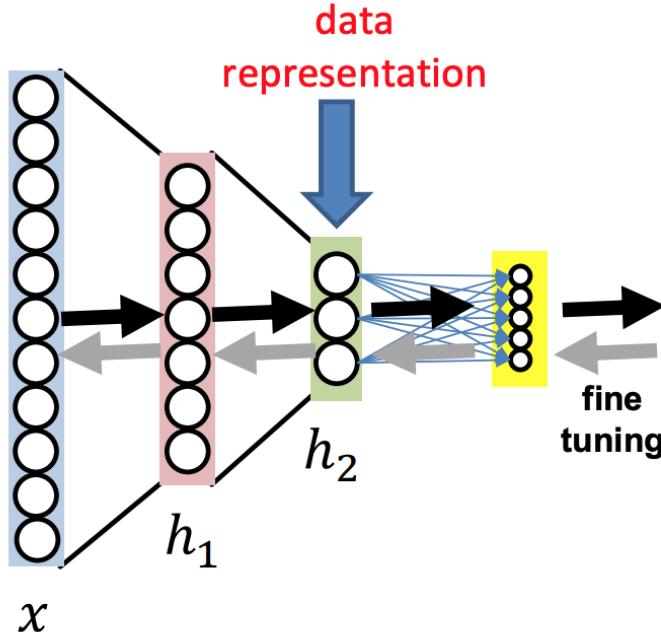
Autoencoders

- Na verdade, não estamos interessados no resultado, mas no “gargalo” que captura as características mais salientes dos dados de treinamento.



Autoencoders

- Na verdade, não estamos interessados no resultado, mas no “gargalo” que captura as características mais salientes dos dados de treinamento.

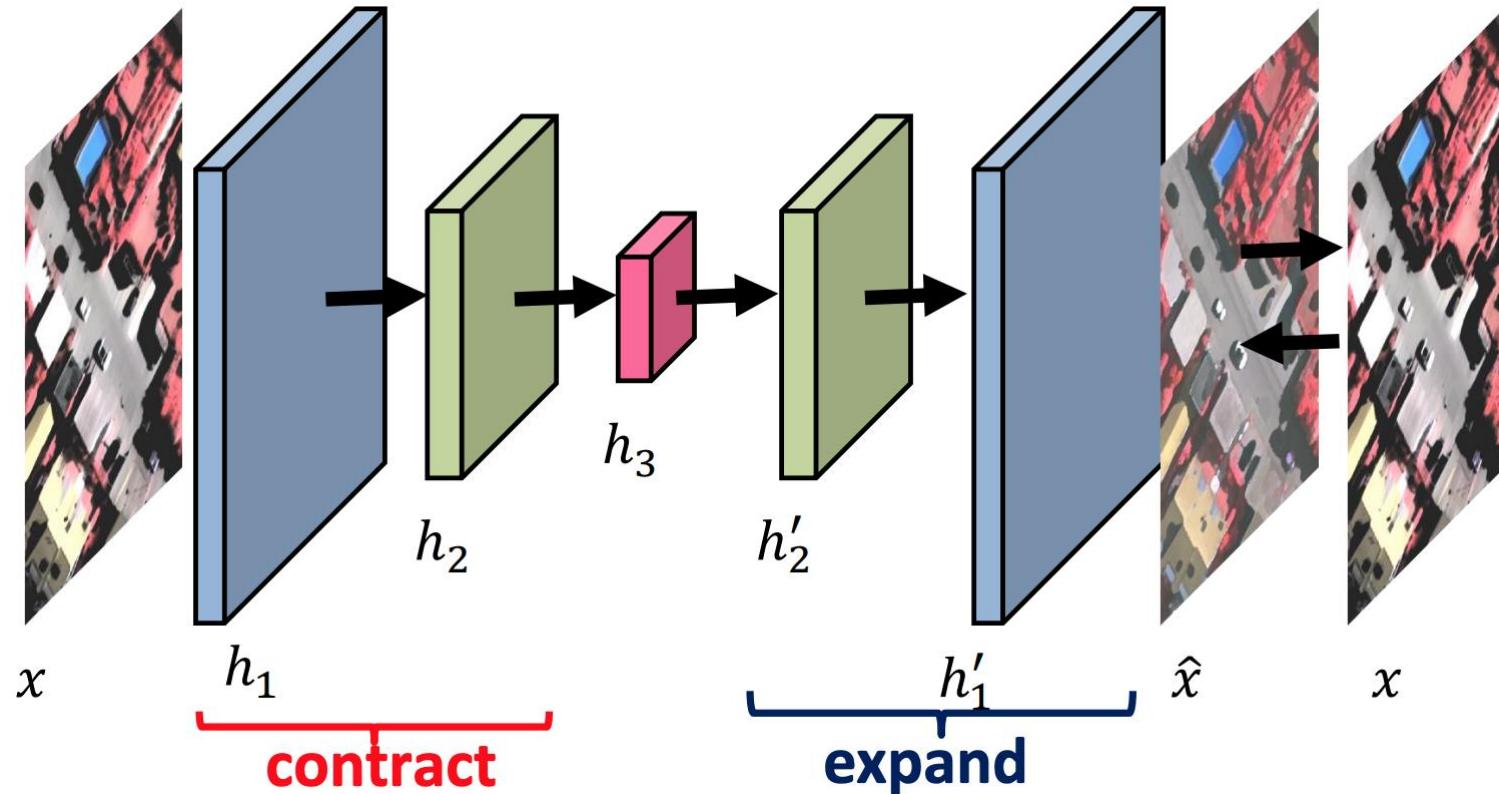


Fine tuning:
falaremos mais
disso no fim da aula!

Convolutional Autoencoders

Convolutional Autoencoders

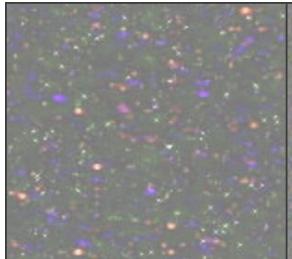
- Treinamento: tenta reconstruir a imagem de entrada



Denoising Autoencoders

Problema: Ruído em imagens

- Fotografias digitais → ruído ocorre principalmente devido às características da câmera e tempo de exposição a luz na aquisição.
- Três tipos mais comuns de ruído:



Ruído com Padrão Fixo
Exposição Longa
Baixo ISO



Ruído Aleatório
Exposição Curta
Alto ISO



Ruído em Bandas
Câmera Suscetível
Sombras Clareadas



Problema: Ruído em imagens

- Fotografias digitais → ruído ocorre principalmente devido às características da câmera e tempo de exposição a luz na aquisição.
- Três tipos mais comuns de ruído:

| | | |
|--|---|--|
| | | |
| Ruído com Padrão Fixo Exposição Longa Baixo ISO | Ruído Aleatório Exposição Curta Alto ISO | Ruído em Bandas Câmera Suscetível Sombras Clareadas |



Imagens médicas



Imagens de radar (SAR)

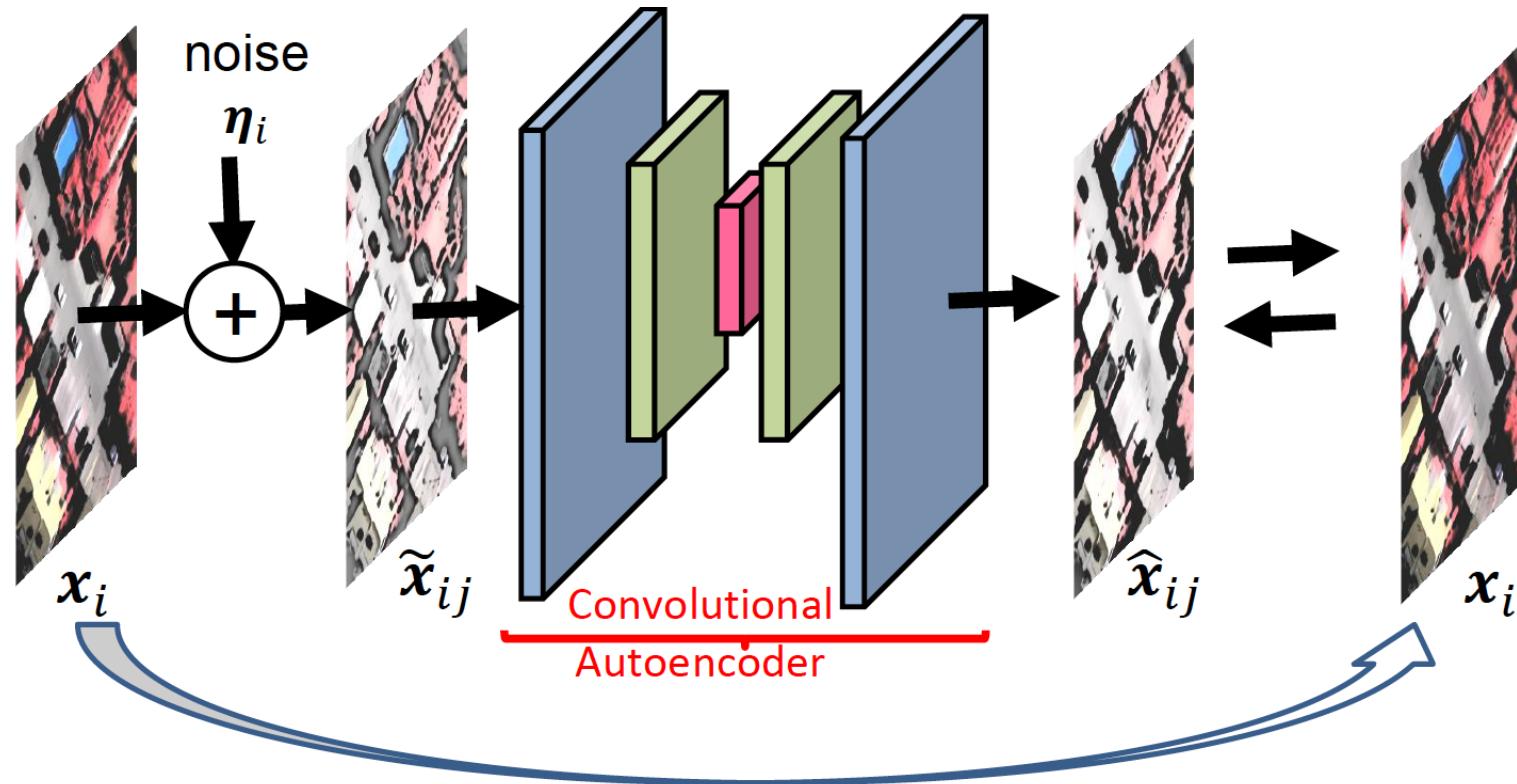






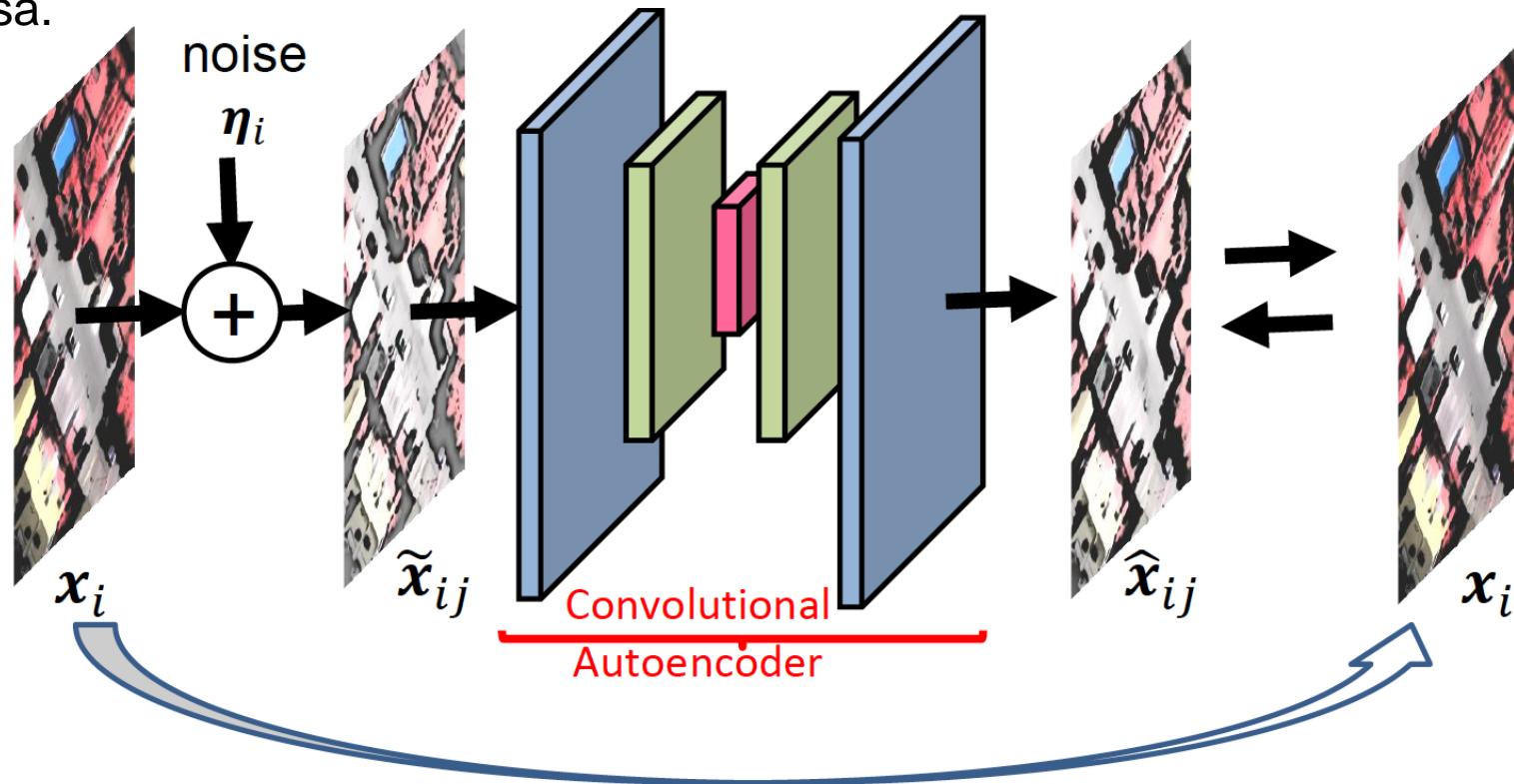
Denoising Autoencoders

1) Adiciona ruído a uma imagem “limpa” para criar versões ruidosas da imagem.



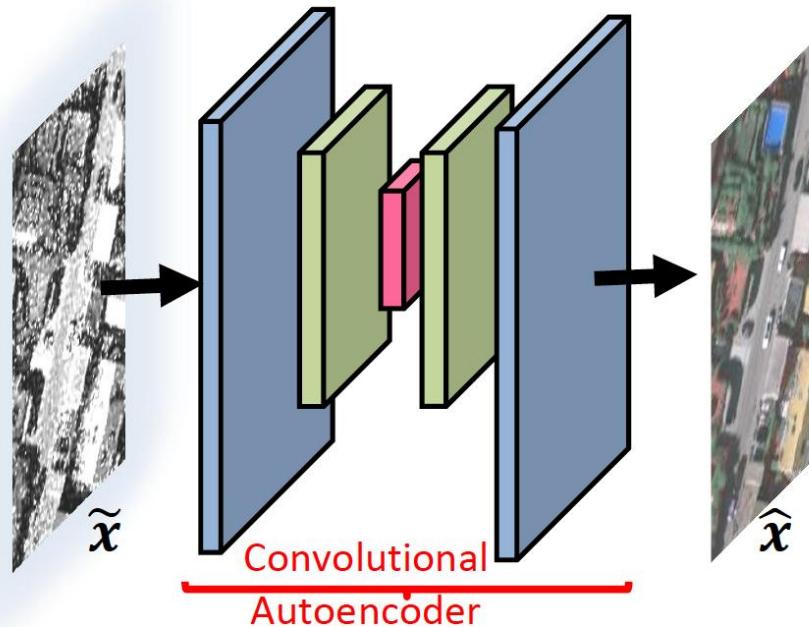
Denoising Autoencoders

2) Treina o autoencoder pra reconstruir a imagem sem ruído a partir da imagem ruidosa.



Denoising Autoencoders

3) Uso: aplica o autoencoder treinado para remover o ruído de uma imagem de entrada.



Sparse Autoencoders

Sparse Coding

- A teoria da aproximação esparsa (também conhecida como representação esparsa) lida com soluções esparsas para sistemas de equações lineares.
- *Sparse Coding:*
 - É uma classe de métodos não supervisionados para aprender representar dados com eficiência.
 - O objetivo é encontrar um **conjunto de vetores de base ϕ_i** , de modo que possamos representar um vetor de entrada x como uma combinação linear desses vetores de base:

$$\mathbf{x} = \sum_{i=1}^k a_i \phi_i$$

Sparse Coding

- A teoria da aproximação esparsa (também conhecida como representação esparsa) lida com soluções esparsas para sistemas de equações lineares.
- *Sparse Coding:*
 - É uma classe de métodos não supervisionados para aprender representar dados com eficiência.
 - O objetivo é encontrar um **conjunto de vetores de base ϕ_i** , de modo que possamos representar um vetor de entrada x como uma combinação linear desses vetores de base:

$$\mathbf{x} = \sum_{i=1}^k a_i \phi_i$$

Parece um
Perceptron?!



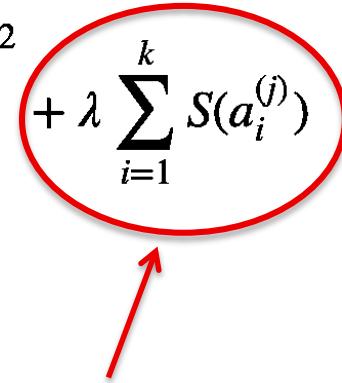
Sparse Coding

- Definimos a função de custo em um conjunto de m vetores de entrada como:

$$\underset{a_i^{(j)}, \phi_i}{\text{minimize}} \sum_{j=1}^m \left\| \mathbf{x}^{(j)} - \sum_{i=1}^k a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^k S(a_i^{(j)})$$

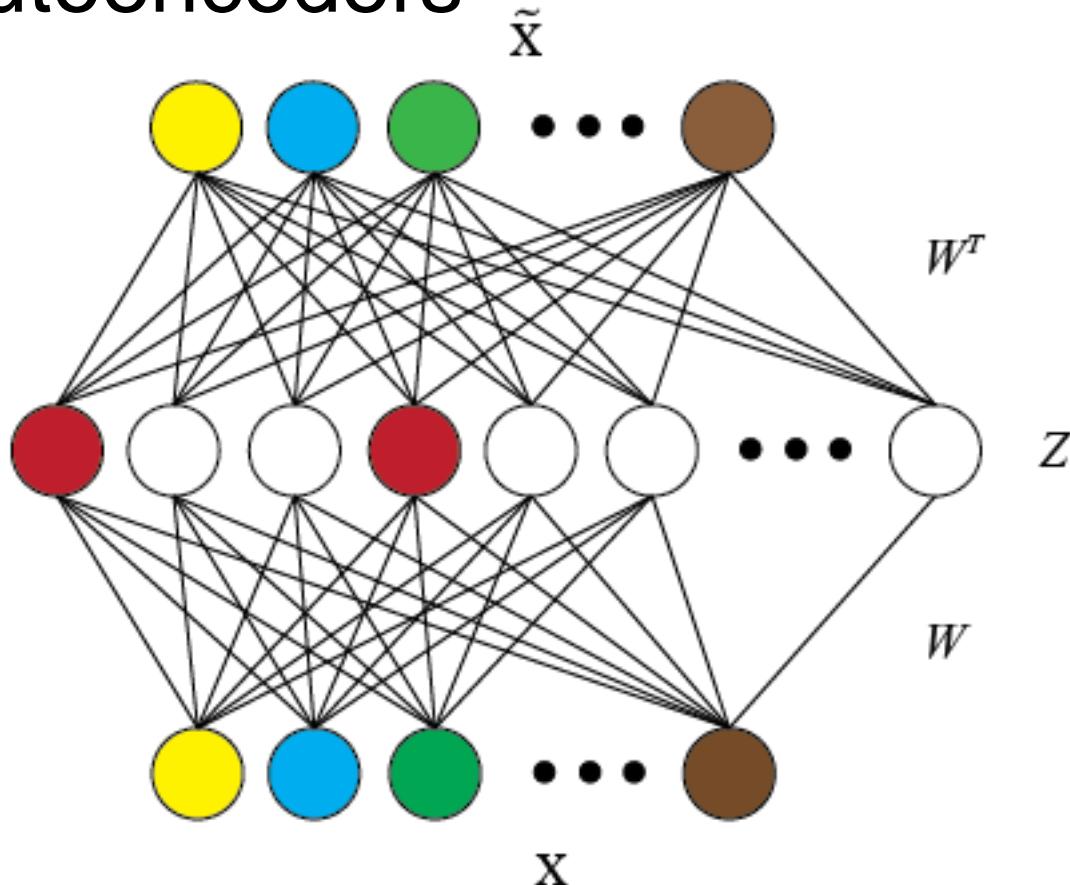
Sparse Coding

- Definimos a função de custo em um conjunto de m vetores de entrada como:

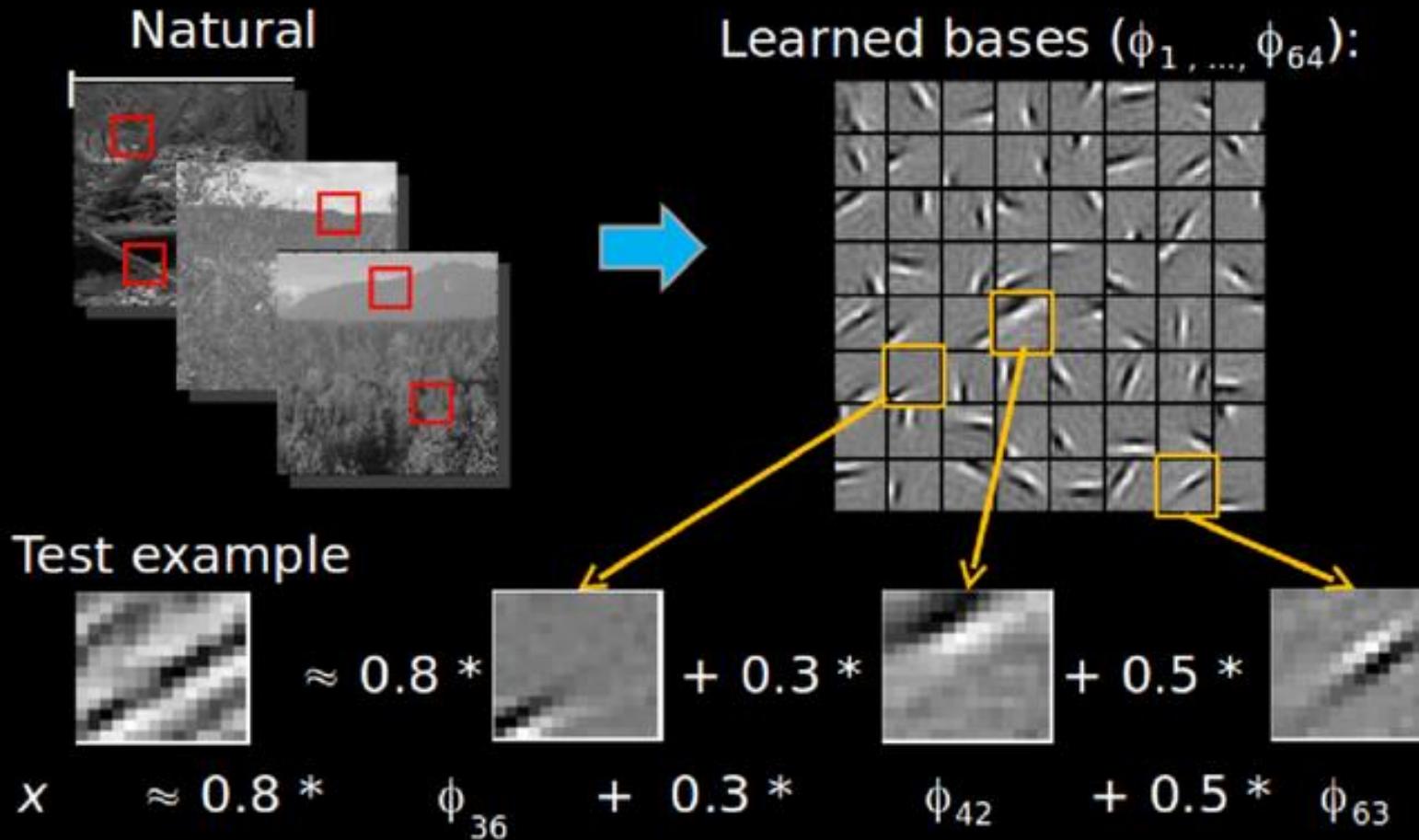
$$\underset{a_i^{(j)}, \phi_i}{\text{minimize}} \sum_{j=1}^m \left\| \mathbf{x}^{(j)} - \sum_{i=1}^k a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^k S(a_i^{(j)})$$


Penaliza a_i por ser muito maior que zero!

Sparse Autoencoders



Sparse coding illustration



| | ϕ_{36} | ϕ_{42} | ϕ_{63} |
|-----------|-------------|-------------|-------------|
| a | 1 | .1 | .1 |
| aardvark | 0 | .2 | 0 |
| aardwolf | 0 | 0 | .1 |
| able | 0 | 0 | 0 |
| account | 1 | .1 | 0 |
| acid | 0 | .1 | 0 |
| across | 0 | 0 | .1 |
| ... | 0 | 0 | 0.1 × .2 |
| baby | 1 | 1.2 | .1 |
| back | 0 | 0 | .2 |
| ... | 0 | 0 | 0 |
| cradle | 1 | 1 | .3 |
| ... | 0 | 0 | 0 |
| ... | 0 | .1 | 0 |
| ... | 0 | 0 | 0 |
| zylophone | 1 | .1 | 1.2 |

Document \approx 0.7×Topic36+0.4×Topic42+0.1×Topic63

Sparse Autoencoders

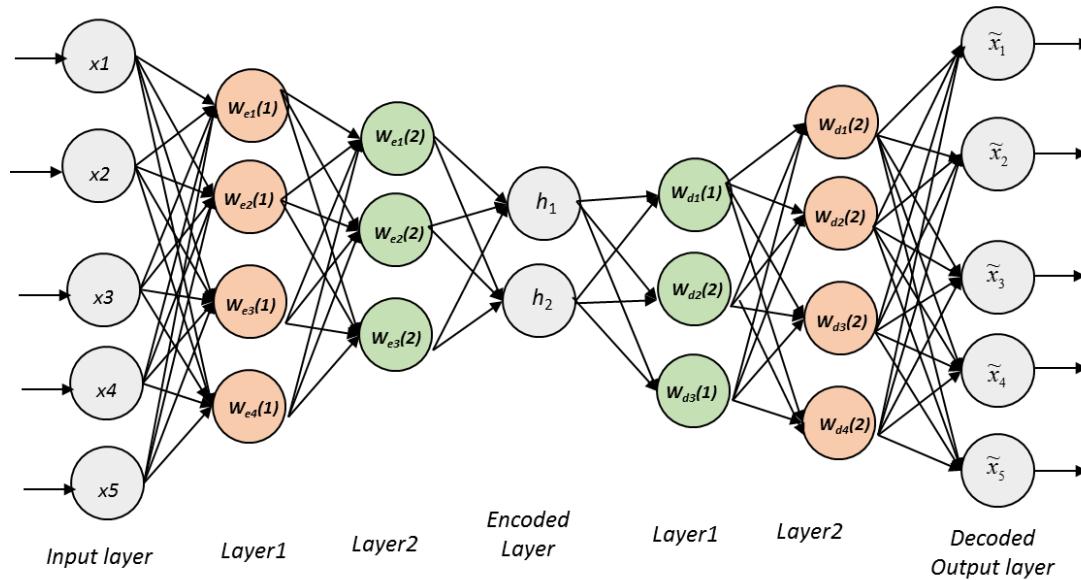
- “Em alguns domínios, como a visão computacional, essa abordagem não é, por si só, competitiva em comparação com abordagens de extração *hand-crafted*.
- As características aprendidas acabam sendo úteis para uma série de problemas (incluindo áudio, texto etc.)
- Há versões mais sofisticadas do *sparse autoencoder* que se saem surpreendentemente bem e, em muitos casos, são competitivas ou superiores até mesmo à outras abordagens do estado da arte.”

(Andrew NG, 2011)

Stacked Autoencoders

Stacked Autoencoders

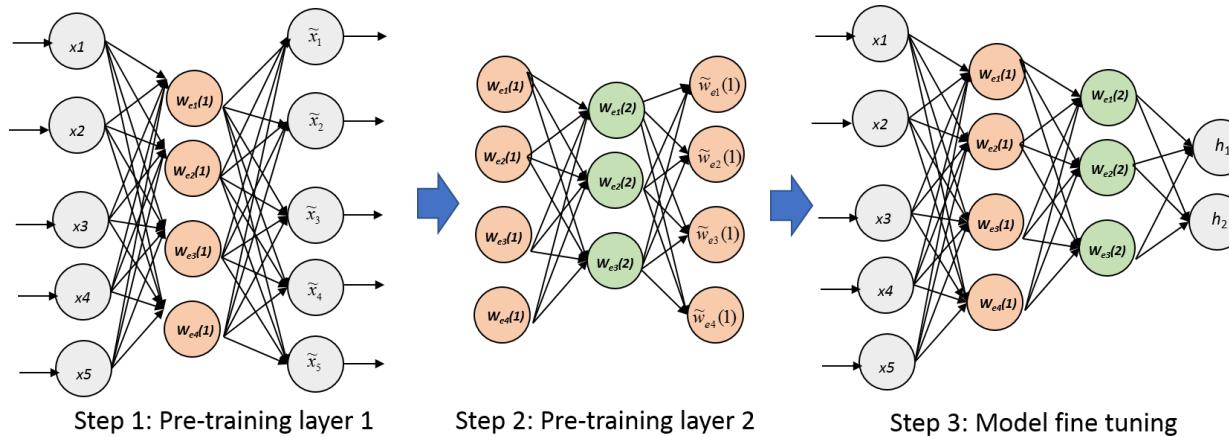
É uma rede neural composta por várias camadas de sparse autoencoders, nos quais a saída de cada camada oculta é conectada à entrada da camada oculta sucessiva.



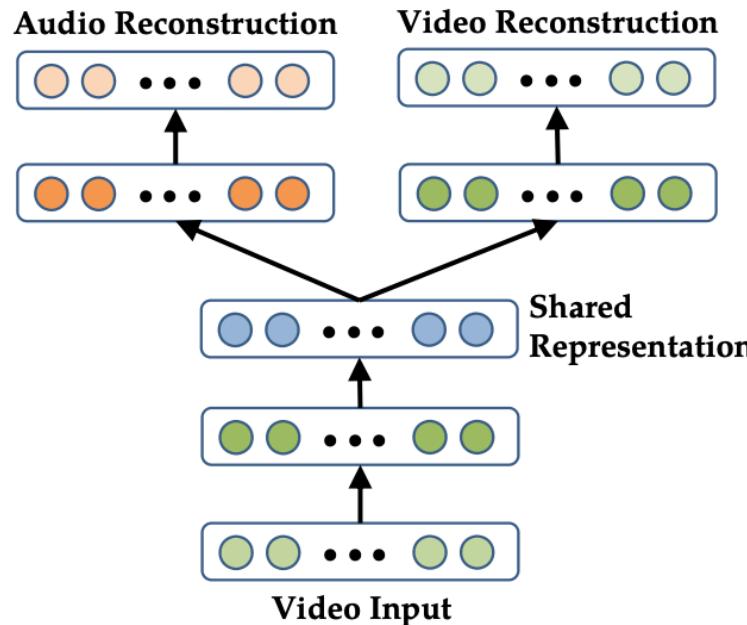
Geralmente usado para aprender *deep features* a partir de poucos dados de treinamento!

Stacked Autoencoders

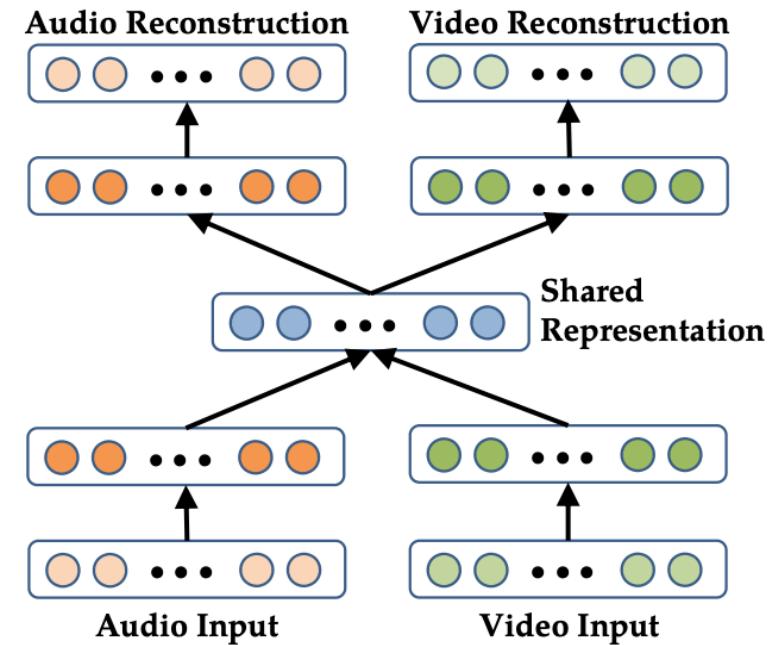
- Treine o autoencoder usando dados de entrada e adquira os dados aprendidos.
- Os dados aprendidos da camada anterior são usados como entrada para a próxima camada e continuam até o treinamento ser concluído.
- Depois que todas as camadas ocultas forem treinadas, use o algoritmo de backpropagation para minimizar a função de custo e os pesos serão atualizados com o conjunto de treinamento para obter o ajuste fino.



Multimodal Stacked Autoencoders



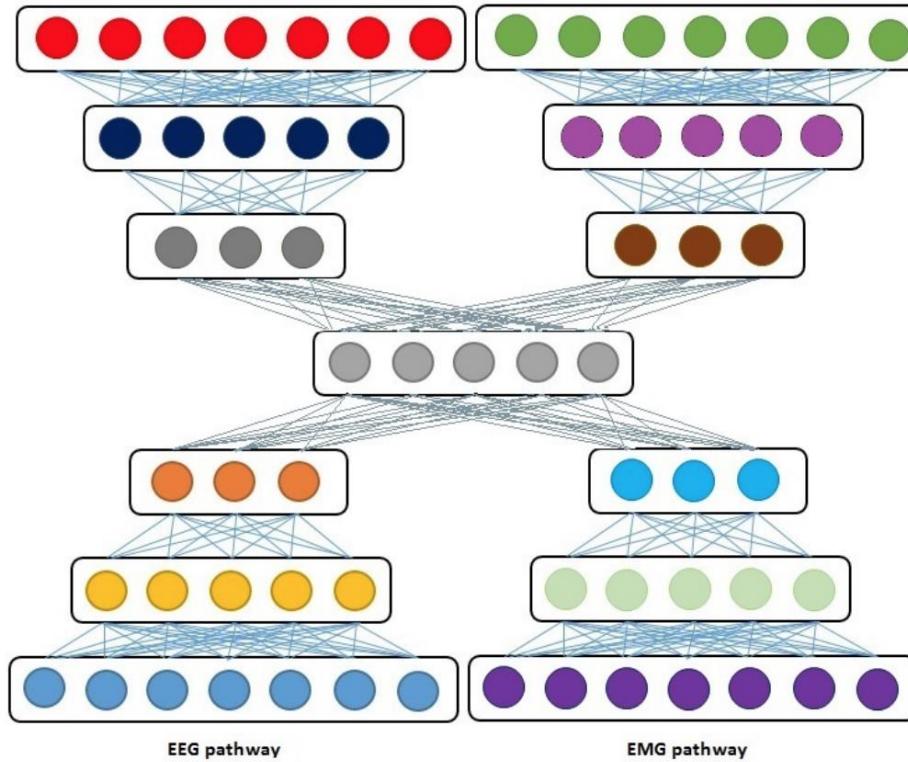
(a) Video-Only Deep Autoencoder



(b) Bimodal Deep Autoencoder

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 689-696).

Multimodal Stacked Autoencoders



Said, Ahmed Ben, et al. "Multimodal deep learning approach for joint EEG-EMG data compression and classification." 2017 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2017.