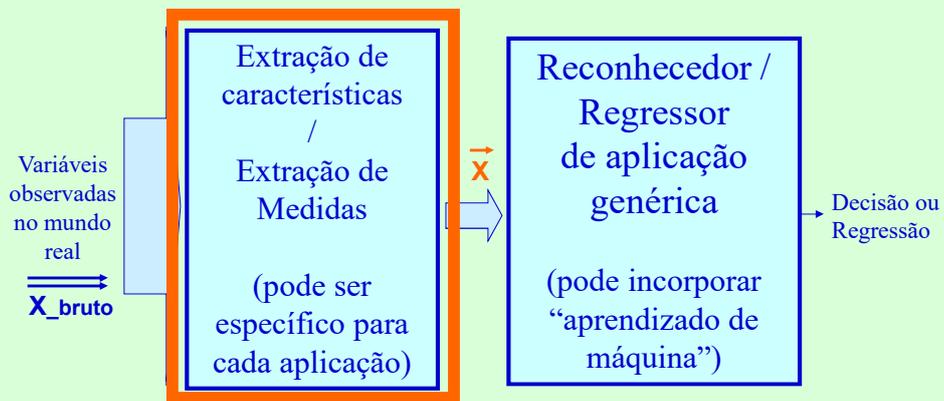


- Algumas respostas sendo dadas com ferramentas de Deep Learning

*Uma técnica **neural** para a redução de dimensionalidade do vetor de entradas X e de extração de características sendo usada no contexto de Deep Learning: Autoencoders (auto-codificadores) e Stacked Autoencoders (vários auto-codificadores encadeados) –*

... O 1o estágio gera um Vetor de Medidas, \vec{X}
 (o segundo estágio operará sobre tal vetor)

132

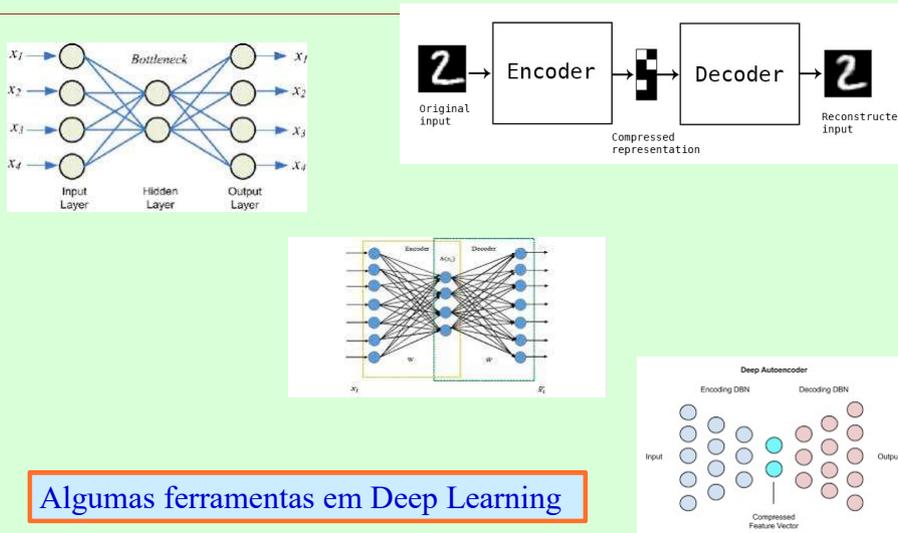


© Prof. Emilio Del Moral Hernandez

132

Autoencoders e Stacked Auto-encoders (imagens da internet)

133



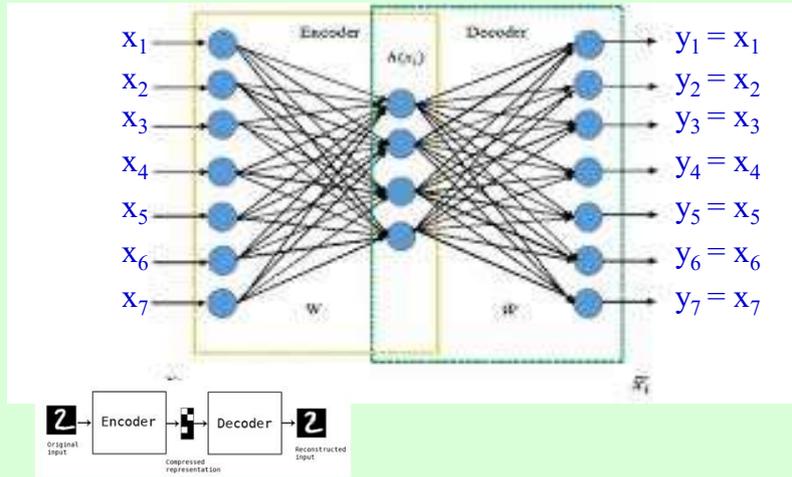
Algumas ferramentas em Deep Learning

© Prof. Emilio Del Moral Hernandez

133

Um autoencoder detalhado
(imagem da internet, adaptada)

134



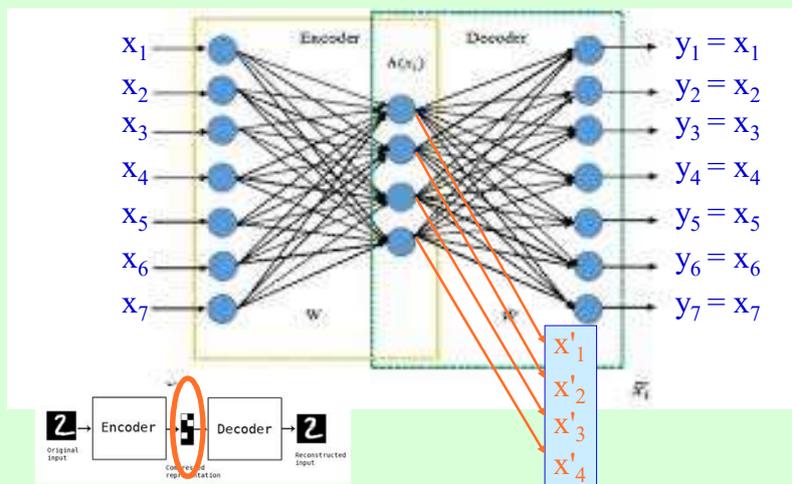
Um pouquinho de Deep Learning

© Prof. Emilio Del Moral Hernandez

134

Um autoencoder detalhado
(imagem da internet, adaptada)

135



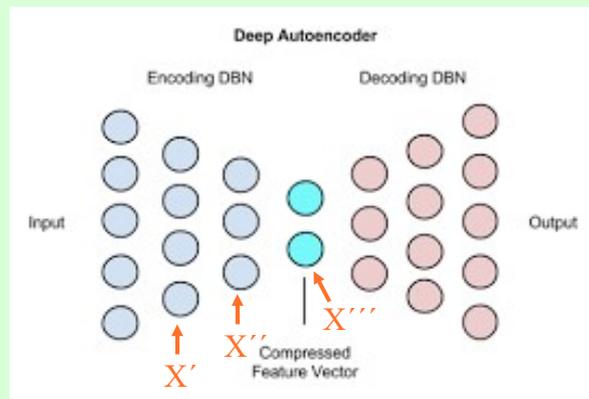
Algumas ferramentas em Deep Learning

© Prof. Emilio Del Moral Hernandez

135

Autoencoders e Stacked Auto-encoders (imagens da internet)

136



Algumas ferramentas em Deep Learning

© Prof. Emilio Del Moral Hernandez

136

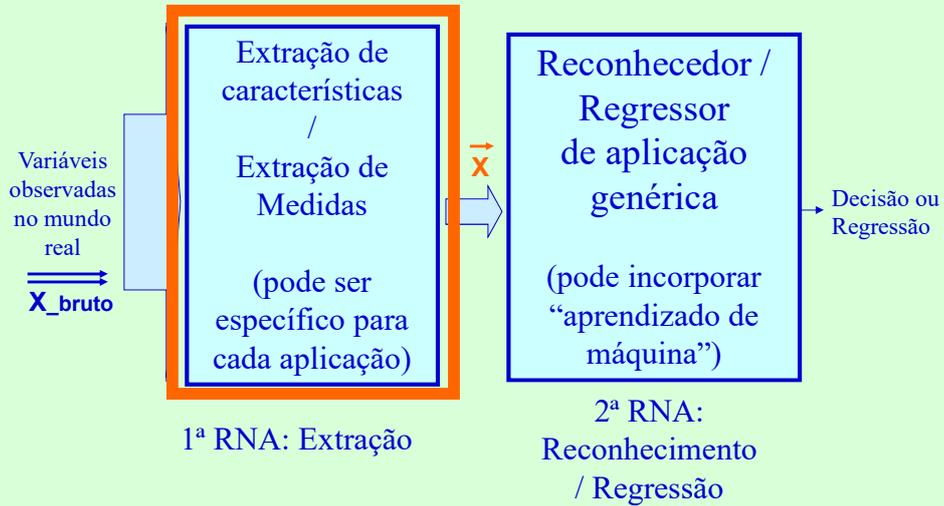
Note que nesta técnica, o primeiro estágio da “solução em dois estágios” também é uma rede neural (não só o segundo estágio é uma RNA), mas esse primeiro estágio é uma RNA específica, desenhada apenas para a codificação compacta de variáveis; ela não realiza a regressão ou o reconhecimento, que são feitos pela segunda rede neural.

137

© Prof. Emilio Del Moral – EPUSP

... O 1o estágio gera um Vetor de Medidas, \vec{X}
 (o segundo estágio operará sobre tal vetor)

138



© Prof. Emilio Del Moral Hernandez

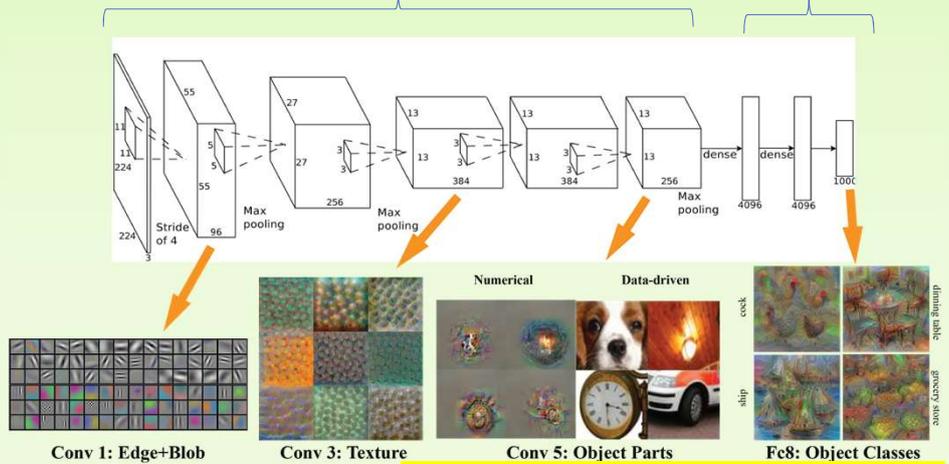
138

Mark Cappello Ferreira de Sousa

Classificação em Redes Neurais Convolucionais

Convoluções
 (Extração de características)

Fully Connected
 (MLP)



figuras ilustrativas – extraídas da internet

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x_{[i,j,0]}$	$w_{[i,j,0]}$	$w_{[i,j,0]}$	$o_{[i,j,0]}$
0 0 0 0 0 0 0	-1 0 1	0 1 -1	2 3 3
0 0 0 1 0 2 0	0 0 1	0 -1 0	3 7 3
0 1 0 2 0 1 0	1 -1 1	0 -1 1	8 10 -3
0 1 0 2 2 0 0	$w_{[i,j,1]}$	$w_{[i,j,1]}$	$o_{[i,j,1]}$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	-8 -3
0 2 1 2 2 0 0	1 -1 1	1 -1 0	-3 1 0
0 0 0 0 0 0 0	0 1 0	1 -1 0	-3 -8 -5
$x_{[i,j,1]}$	$w_{[i,j,2]}$	$w_{[i,j,2]}$	
0 0 0 0 0 0 0	-1 -1 -1	-1 1 -1	
0 2 1 2 1 1 0	1 -1 0	0 -1 -1	
0 2 1 2 0 -1 0	0 -1 0	1 0 0	
0 0 2 1 0 1 0	Bias b0 (1x1x1)	Bias b1 (1x1x1)	
0 1 2 2 2 2 0	$b_{[i,j,0]}$	$b_{[i,j,0]}$	
0 0 1 2 0 1 0	1	0	
0 0 0 0 0 0 0			
$x_{[i,j,2]}$			
0 0 0 0 0 0 0			
0 2 1 1 2 0 0			
0 1 0 0 1 0 0			
0 0 1 0 0 0 0			
0 1 0 2 1 0 0			
0 2 2 1 1 1 0			
0 0 0 0 0 0 0			

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

GET UPDATES

toggle movement

© Prof. Emilio Del Moral Hernandez

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x_{[i,j,0]}$	$w_{[i,j,0]}$	$w_{[i,j,0]}$	$o_{[i,j,0]}$
0 0 0 0 0 0 0	-1 0 1	0 1 -1	2 3 3
0 0 0 1 0 2 0	0 0 1	0 -1 0	3 7 3
0 1 0 2 0 1 0	1 -1 1	0 -1 1	8 10 -3
0 1 0 2 2 0 0	$w_{[i,j,1]}$	$w_{[i,j,1]}$	$o_{[i,j,1]}$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	-8 -3
0 2 1 2 2 0 0	1 -1 1	1 -1 0	-3 1 0
0 0 0 0 0 0 0	0 -1 0	1 -1 0	-3 -8 -5
$x_{[i,j,1]}$	$w_{[i,j,2]}$	$w_{[i,j,2]}$	
0 0 0 0 0 0 0	-1 -1 -1	-1 1 -1	
0 2 1 2 1 1 0	1 -1 0	0 -1 -1	
0 2 1 2 0 -1 0	0 -1 0	1 0 0	
0 0 2 1 0 1 0	Bias b0 (1x1x1)	Bias b1 (1x1x1)	
0 1 2 2 2 2 0	$b_{[i,j,0]}$	$b_{[i,j,0]}$	
0 0 1 2 0 1 0	1	0	
0 0 0 0 0 0 0			
$x_{[i,j,2]}$			
0 0 0 0 0 0 0			
0 2 1 1 2 0 0			
0 1 0 0 1 0 0			
0 0 1 0 0 0 0			
0 1 0 2 1 0 0			
0 2 2 1 1 1 0			
0 0 0 0 0 0 0			

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

GET UPDATES

toggle movement

© Prof. Emilio Del Moral Hernandez

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, i, 0]$	$w0[i, i, 0]$	$w1[i, i, 0]$	$o[i, i, 0]$
0 0 0 0 0 0 0	-1 0 1	0 1 -1	2 3 3
0 0 0 1 0 2 0	0 0 1	0 1 0	3 7 3
0 1 0 2 0 1 0	1 -1 1	0 1 1	8 10 -3
0 1 0 2 2 0 0	-1 0 1	-1 0 0	o[i, i, 1]
0 2 0 0 2 0 0	1 -1 1	1 -1 0	-8 -8 3
0 2 1 2 2 0 0	0 1 0	1 -1 0	-3 1 0
0 0 0 0 0 0 0	0 1 0	1 -1 0	-3 -8 -5
$x[i, i, 1]$	$w0[i, i, 1]$	$w1[i, i, 1]$	
0 0 0 0 0 0 0	-1 0 1	-1 1 2	
0 0 0 1 0 2 0	0 0 1	1 1 0	
0 1 0 2 0 1 0	1 -1 1	0 -1 -1	
0 1 0 2 2 0 0	-1 0 1	0 -1 -1	
0 2 1 2 2 0 0	0 1 0	1 0 0	
0 0 0 0 0 0 0	0 1 0	1 0 0	
$x[i, i, 2]$	$w0[i, i, 2]$	$w1[i, i, 2]$	
0 0 0 0 0 0 0	-1 0 1	-1 1 2	
0 0 0 1 0 2 0	0 0 1	1 1 0	
0 1 0 2 0 1 0	1 -1 1	0 -1 -1	
0 1 0 2 2 0 0	-1 0 1	0 -1 -1	
0 2 1 2 2 0 0	0 1 0	1 0 0	
0 0 0 0 0 0 0	0 1 0	1 0 0	

Bias b0 (1x1x3)
b0[i, i, 0]

Bias b1 (1x1x1)
b1[i, i, 0]

toggle movement

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

GET UPDATES

© Prof. Emilio Del Moral Hernandez

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, i, 0]$	$w0[i, i, 0]$	$w1[i, i, 0]$	$o[i, i, 0]$
0 0 0 0 0 0 0	-1 0 1	0 1 -1	2 3 3
0 0 0 1 0 2 0	0 0 1	0 1 0	3 7 3
0 1 0 2 0 1 0	1 -1 1	0 1 1	8 10 -3
0 1 0 2 2 0 0	-1 0 1	-1 0 0	o[i, i, 1]
0 2 0 0 2 0 0	1 -1 1	1 -1 0	-8 -8 -3
0 2 1 2 2 0 0	0 1 0	1 -1 0	-3 1 0
0 0 0 0 0 0 0	0 1 0	1 -1 0	-3 -8 -5
$x[i, i, 1]$	$w0[i, i, 1]$	$w1[i, i, 1]$	
0 0 0 0 0 0 0	-1 0 1	-1 1 2	
0 0 0 1 0 2 0	0 0 1	1 1 0	
0 1 0 2 0 1 0	1 -1 1	0 -1 -1	
0 1 0 2 2 0 0	-1 0 1	0 -1 -1	
0 2 1 2 2 0 0	0 1 0	1 0 0	
0 0 0 0 0 0 0	0 1 0	1 0 0	
$x[i, i, 2]$	$w0[i, i, 2]$	$w1[i, i, 2]$	
0 0 0 0 0 0 0	-1 0 1	-1 1 2	
0 0 0 1 0 2 0	0 0 1	1 1 0	
0 1 0 2 0 1 0	1 -1 1	0 -1 -1	
0 1 0 2 2 0 0	-1 0 1	0 -1 -1	
0 2 1 2 2 0 0	0 1 0	1 0 0	
0 0 0 0 0 0 0	0 1 0	1 0 0	

Bias b0 (1x1x3)
b0[i, i, 0]

Bias b1 (1x1x1)
b1[i, i, 0]

toggle movement

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

GET UPDATES

© Prof. Emilio Del Moral Hernandez

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, i, 0]$ 0 0 0 0 0 0 0 0 0 0 1 0 2 0 0 1 0 2 0 1 0 0 1 0 2 2 0 0 0 2 0 0 2 0 0 0 2 1 2 2 0 0 0 0 0 0 0 0 0	$w0[i, i, 0]$ -1 0 1 0 0 1 1 -1 1 $w0[i, i, 1]$ 1 0 1 -1 0 1 1 -1 1 $w0[i, i, 2]$ 1 1 1 1 -1 0 0 -1 0	$w1[i, i, 0]$ 0 1 1 0 1 1 0 1 1 $w1[i, i, 1]$ -1 0 0 1 -1 0 1 -1 0 $w1[i, i, 2]$ -1 1 2 -1 1 2 1 0 0	$o[i, i, 0]$ 2 3 3 3 7 3 8 10 -3 $o[i, i, 1]$ -8 -8 -3 -3 1 0 -3 -8 -8

Toggle movement

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

GET UPDATES

© Prof. Emilio Del Moral Hernandez

145

146

... até que a imagem toda seja varrida por um mesmo filtro.

© Prof. Emilio Del Moral Hernandez

146

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Medium

Input Volume (pad 1) (7x7x3)

$x[i, j, 0]$	0	0	0	0	0	0
$x[i, j, 1]$	0	0	0	0	0	0
$x[i, j, 2]$	0	0	0	0	0	0
$x[i, j, 0]$	0	0	0	0	0	0
$x[i, j, 1]$	0	0	0	0	0	0
$x[i, j, 2]$	0	0	0	0	0	0
$x[i, j, 0]$	0	0	0	0	0	0
$x[i, j, 1]$	0	0	0	0	0	0
$x[i, j, 2]$	0	0	0	0	0	0

Filter W0 (3x3x3)

$w0[i, j, 0]$	1	0	2
$w0[i, j, 1]$	0	1	1
$w0[i, j, 2]$	1	1	0

Filter W1 (3x3x3)

$w1[i, j, 0]$	0	1	-1
$w1[i, j, 1]$	0	-1	0
$w1[i, j, 2]$	-1	1	1

Output Volume (3x3x2)

$o[i, j, 0]$	2	3	3
$o[i, j, 1]$	3	7	3
$o[i, j, 0]$	8	10	3
$o[i, j, 1]$	-8	-8	-3
$o[i, j, 0]$	-3	1	0
$o[i, j, 1]$	-3	-8	-8

logfile viewer

A demo of a Conv layer with $K=2$ filters, each with a spatial extent $F=3$, moving at a stride $S=2$, and input

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more

© Prof. Emilio Del Moral Hernandez

Num MLP tradicional, quantos parâmetros “w’s” temos só na primeira camada (com digamos 10 neurônios) se a imagem de entrada tiver 1M pixels?

Multiple Back-Propagation Version 2.2.5

Topology RMS Output vs Desired (training data) Output vs Desired (testing data)

20-10-5-1

Learning

Main Network

Learning Rate 0.7

Momentum 0.7

Space Network

Learning Rate 0.7

Momentum 0.7

Configuration

Network

Generate C code

Load

Save

Input Sensitivity

Weights

Randomize

View

Load

Save

Root Mean Square Error

Main Network

Training 1.0000000000

Testing 1.0000000000

Space Network

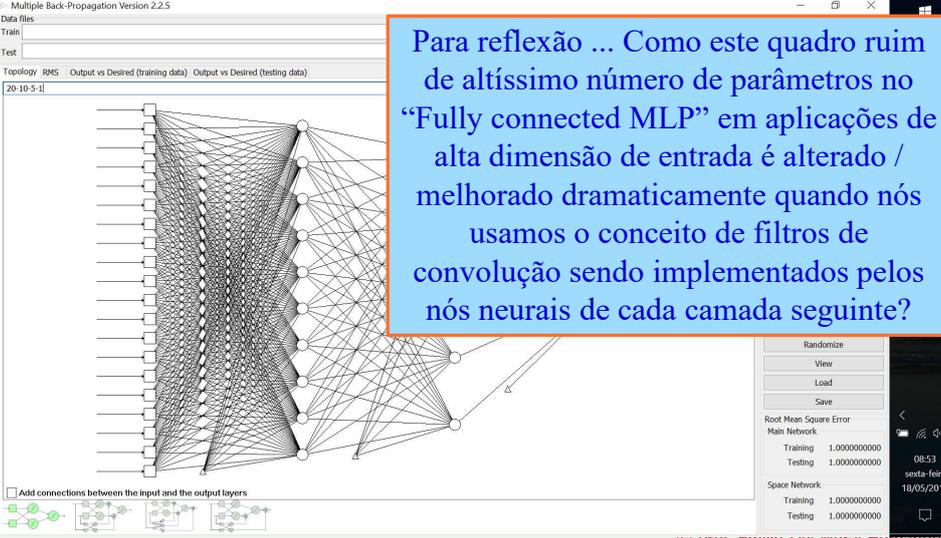
Training 1.0000000000

Testing 1.0000000000

© Prof. Emilio Del Moral Hernandez

Num MLP tradicional, quantos parâmetros “w’s” temos só na primeira camada (com digamos 10 neurônios) se a imagem de entrada tiver 1M pixels?

149



Multiple Back-Propagation Version 2.2.5

Data files

Train

Test

Topology RMS Output vs Desired (training data) Output vs Desired (testing data)

20-10-5-1

Para reflexão ... Como este quadro ruim de altíssimo número de parâmetros no “Fully connected MLP” em aplicações de alta dimensão de entrada é alterado / melhorado dramaticamente quando nós usamos o conceito de filtros de convolução sendo implementados pelos nós neurais de cada camada seguinte?

Randomize

View

Load

Save

Root Mean Square Error

Network	Training	Testing
Main Network	1.0000000000	1.0000000000
Space Network	1.0000000000	1.0000000000

08:53

seita-feira

18/05/2018

© Prof. Emilio Del Moral Hernandez

149

Vídeo bem interessante gerado por alunos da PSI5886-Prof Emilio <https://www.youtube.com/watch?v=2dz4qLq-nMU&feature=youtu.be>

150

Teoria Redes Convolucionais

Redes Convolucionais – Parte 2

Trabalho final da disciplina
PSI5886 – Princípios de Neurocomputação

Grupo:
Bruno Giordano
Fábio Teixeira
Wanderson Ferreira
Bruno Franceschini Canale

Escola Politécnica da Universidade de São Paulo

0:14 / 18:25

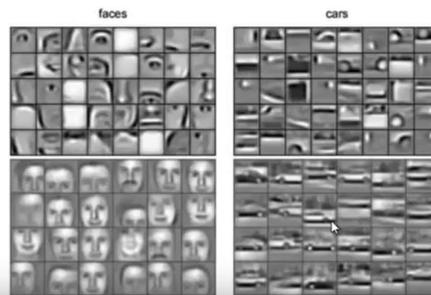
© Prof. Emilio Del Moral Hernandez

150

ConvNets - Convolução



- As redes convolucionais aplicam filtros ao longo da imagem, procurando representações características para então classificá-las

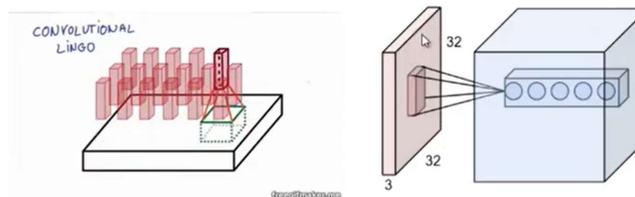


2:42 / 18:25

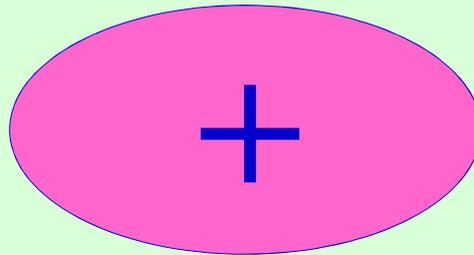
ConvNets – Convolução



- As camadas são volumes que representam convoluções – imagens são filtradas



3:05 / 18:25



- um pouco sobre a PSI5886 do 3º período

PSI 5886 – Prof Emilio – no 3º período da pós-graduação

Janus

uspdigital.usp.br/janus/DisiplinaAux?tipo=D

USP Universidade de São Paulo
Brasil

Login

- Entrar

Apresentação

- Apresentação

Acesso

- Esqueci a senha
- Primeiro acesso

Acesso público

- Período de matrícula
- Disciplinas oferecidas
- Disciplinas in English
- Catálogo de disciplinas
- Orientadores

Portal Alumni

Catálogo de disciplinas

Discipline Catalog

Disciplina PSI5886
Princípios de Neurocomputação

Área de Concentração: 3142

Criação: 11/09/2017

Ativação: 11/09/2017

Nr. de Créditos: 8

Carga Horária:

Teórica (por semana)	Prática (por semana)	Estudos (por semana)	Duração	Total
3	0	7	12 semanas	120 horas

Docente Responsável:
Emilio Del Moral Hernandez

Objetivos:
Oferecer ao aluno a teoria básica de redes neurais artificiais e o contato com as suas aplicações mais importantes.

Justificativa:
Esta disciplina vem sendo oferecida regularmente na nossa Escola uma vez ao ano e tem tido boa procura e grande interesse por parte dos alunos. Ao final desta disciplina o aluno é capaz de entender com clareza o funcionamento e a aplicabilidade de redes neurais artificiais aos problemas de engenharia, modelagem e processamento de informação, e tem uma sólida base para evoluir no estudo de novos modelos de redes neurais que estão sendo propostos e debatidos nos meios acadêmicos, conferências técnicas e revistas técnicas da área.

Conteúdo:

- 1 - Conceitos em neurofisiologia (Breve abordagem) e conexão com as redes de neurônios artificiais.
- 2 - Diversidade de modelos neurais. Modelos simples, lineares e sigmóides.
- 3 - Arquiteturas neurais mais importantes: MLP (Perceptron de Múltiplas Camadas), Redes Associativas de Hopfield e Mapas Auto-organizados de Kohonen.
- 4 - Teia de aprendizagem: princípio de Hebb, aprendizagem competitiva e aprendizagem supervisionada.

© Prof. Emilio Del Moral Hernandez

177

Janus x +
uspdigital.usp.br/janus/DisiplinaAux?tipo=D

funcionamento e a aplicabilidade de redes neurais artificiais aos problemas de engenharia, modelagem e processamento de informação, e tem uma sólida base para evoluir no estudo de novos modelos de redes neurais que estão sendo propostos e debatidos nos meios acadêmicos, conferências técnicas e revistas técnicas da área.

Conteúdo:

1 - Conceitos em neurofisiologia (breve abordagem) e conexão com as redes de neurônios artificiais. 2 - Diversidade de modelos neurais. Modelos pulsados, binários e sigmoidais. 3 - Arquiteturas neurais mais importantes: MLP (Perceptron de Múltiplas Camadas), Redes Associativas de Hopfield e Mapas Auto-organizados de Kohonen. 4 - Leis de aprendizado: princípio de Hebb, aprendizado competitivo e aprendizado supervisionado. Desenvolvimento do aprendizado por Error Back Propagation. 5 - Aplicações de redes neurais em engenharia, em modelagem e em sistemas de processamento de informação. 6 - Modelos neurais pulsados e outros modelos incorporando aspectos de neurodinâmica sendo desenvolvidos mais recentemente. 7 - Elementos de "Deep Learning"

Forma de Avaliação:

A avaliação é feita com base em duas componentes de atividade: uma primeira referente ao "trajeto de curso / atividades de estudo e práticas", com base em resumo

Observação:

A avaliação é feita com base em duas componentes de atividade: uma primeira referente ao "trajeto de curso / atividades de estudo e práticas", com base em resumo de leituras solicitadas, pequenos projetos e exercícios realizados ao longo do semestre, semanalmente ou quinzenalmente, normalmente individuais e ocasionalmente em grupo; a segunda componente se refere a um trabalho de fim de curso, que pode envolver simulações computacionais e/ou aprofundamentos teóricos, realizado em grupos de 2 ou 3 alunos, com apresentação aos

© Prof. Emilio Del Moral Hernandez
178

Conteúdo planejado em PSI5886:

1 - Conceitos em neurofisiologia (breve abordagem) e conexão com as redes de neurônios artificiais.

2 - Diversidade de modelos neurais. Modelos pulsados, binários e sigmoidais.

3 - Arquiteturas neurais mais importantes:
MLP (Perceptron de Múltiplas Camadas), Redes Associativas de Hopfield e Mapas Auto-organizados de Kohonen.

4 - Leis de aprendizado: princípio de Hebb, aprendizado competitivo e aprendizado supervisionado. Desenvolvimento do aprendizado por Error Back Propagation.

5 - Aplicações de redes neurais em engenharia, em modelagem e em sistemas de processamento de informação.

6 - Modelos neurais pulsados e outros modelos incorporando aspectos de neurodinâmica sendo desenvolvidos mais recentemente.

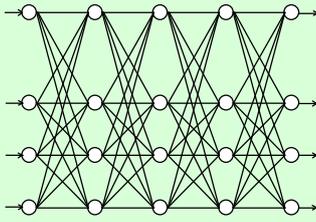
7 - Elementos de "Deep Learning"

© Prof. Emilio Del Moral Hernandez
179

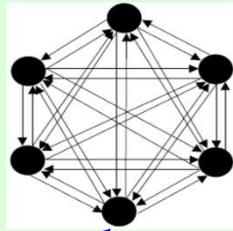
Três arquiteturas neurais importantes (abordadas em pósgrad – PSI 5886)

184

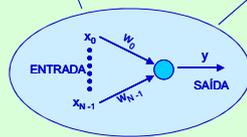
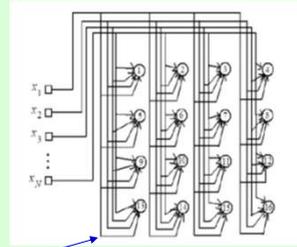
1) MLP
- Multi Layer
Perceptron



2) Memória
Associativa
de Hopfield



3) Mapas Auto-
Organizáveis
de Kohonen



© Prof. Emilio Del Moral Hernandez

184