

1º tópico da agenda PSI3471 de 22 de abril de 2020:

Os Autoencoders do final da aula passada (15 de abril); caso não tenha visto o final daquele video longo para o 15 de abril.

*Auto Encoders
(em uma forma simples)*

*Uma técnica de extração similar à PCA e
bastante em uso no universo
do Deep Learning*



Detector / reconhecedor multivariado

Variáveis observadas no mundo real
 \vec{X}

Reconhecedor implementado com uma rede neural que tem como entradas as variáveis $x_1, x_2, x_3, \text{ etc}$ e como saída a variável y , dependente das variáveis x_i

decisão binária y , dados os valores X

Regressor multivariado

Variáveis observadas no mundo real
 \vec{X}

Regressor implementado com uma rede neural que tem como entradas as variáveis $x_1, x_2, x_3, \text{ etc}$ e como saída a variável y , dependente das variáveis x_i

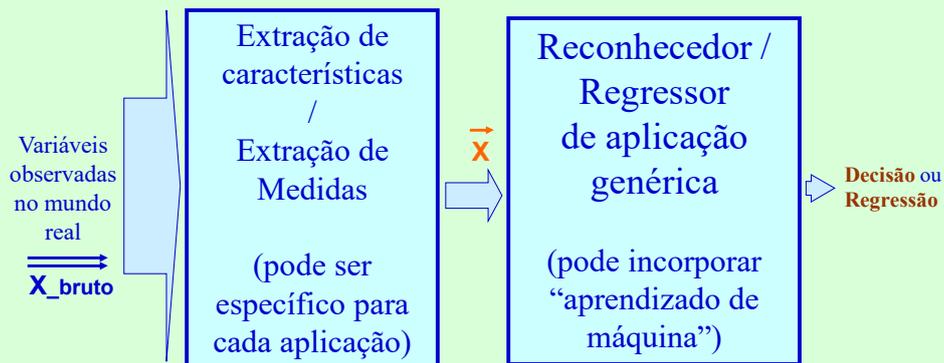
estimação de valor para o y , dados os valores X

Em vários projetos de regressão e reconhecimento (embora não em todos) há a necessidade de uma solução em 2 estágios

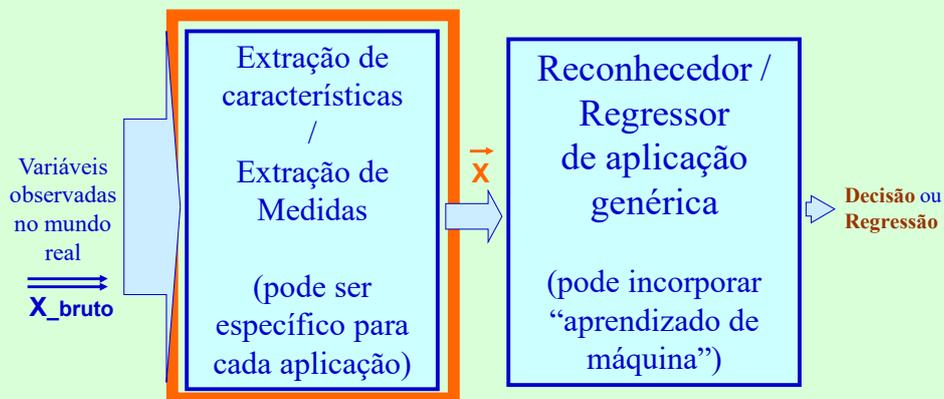
Um primeiro responsável por extração de características / extração de medidas mais relevantes dos objetos “X” brutos, medidas essas normalmente específicas para cada aplicação, e ...

Um segundo estágio atuando sobre o vetor com essas diversas medidas, e que emprega alguma técnica para a regressão / identificação de padrões, independente da aplicação, como é o caso de redes neurais artificiais

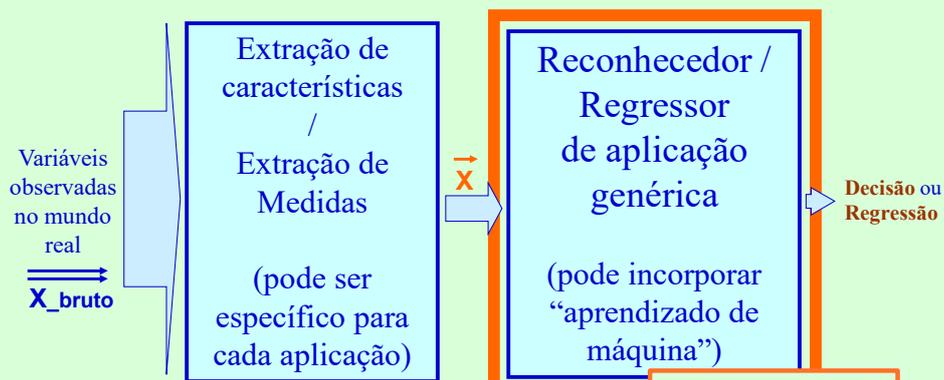
Elaborando uma Solução em dois estágios



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



O segundo estágio opera sobre o Vetor de Medidas, \vec{X} (o 1o estágio gerou tal vetor)



O Multi Layer Perceptron (MLP)

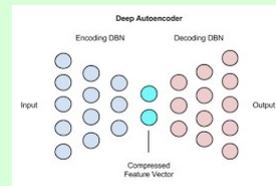
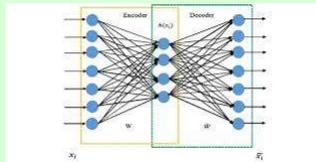
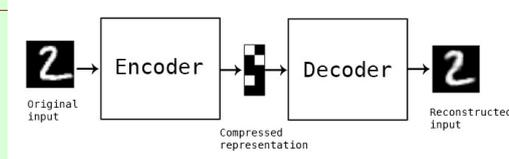
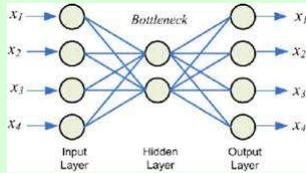
- Múltiplas entradas / Múltiplas saídas / Múltiplas camadas
- Variáveis (internas e externas) analógicas ou digitais
- Relações lineares ou não lineares entre elas

Proc. Não Linear de Sinais

Kohonen, Cherkov (1998)

Autoencoders e Stacked Auto-encoders

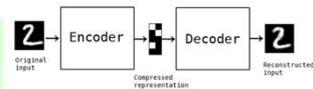
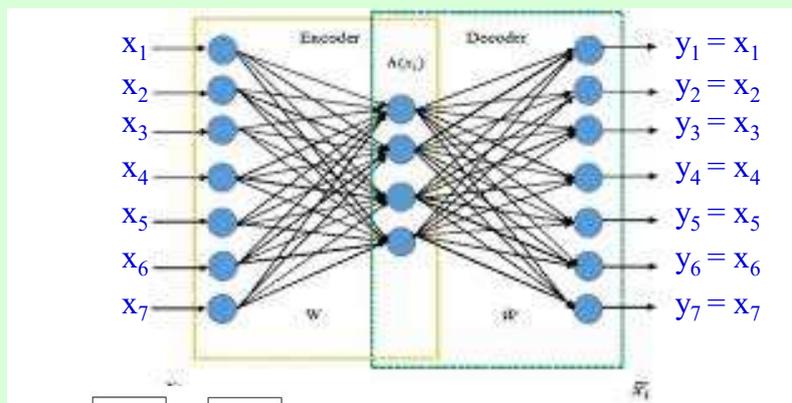
(imagens da internet)



Algumas ferramentas em Deep Learning

Um autoencoder detalhado

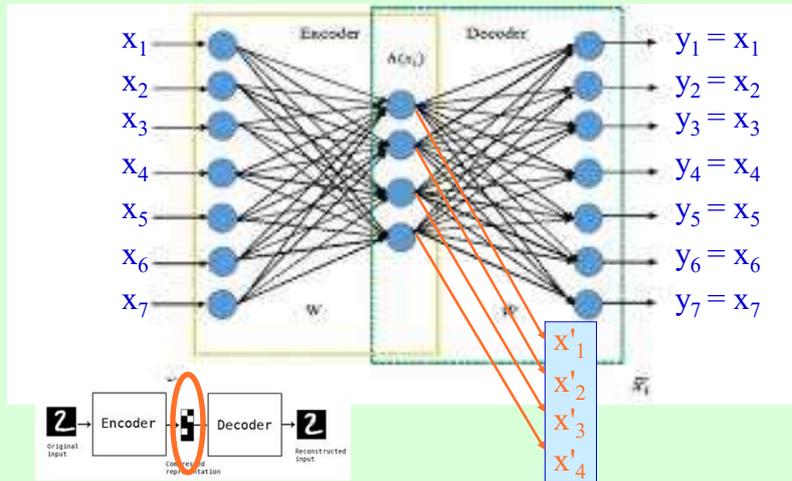
(imagem da internet, adaptada)



Um pouquinho de Deep Learning

Um autoencoder detalhado (imagem da internet, adaptada)

13

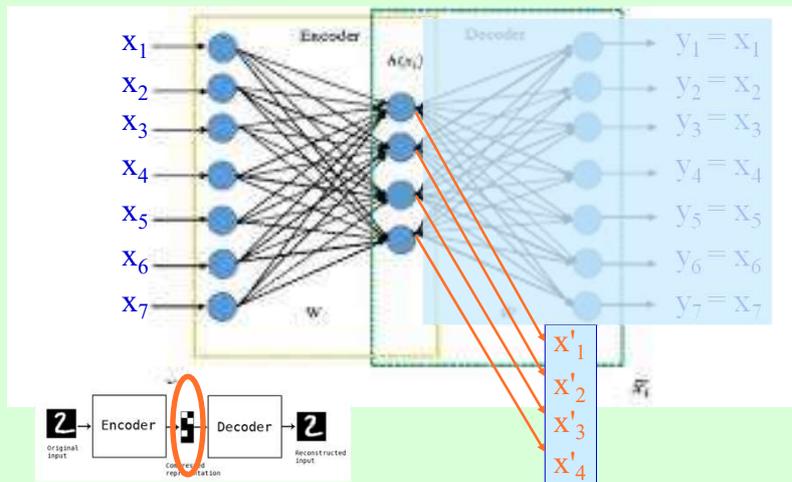


Algumas ferramentas em Deep Learning

© Prof. Emilio Del Moral Hernandez

Um autoencoder detalhado (imagem da internet, adaptada)

14

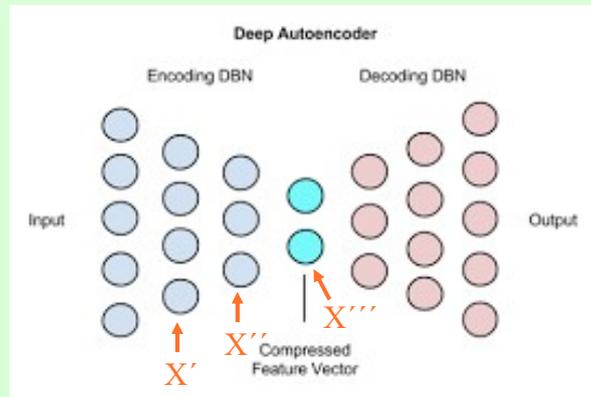


Algumas ferramentas em Deep Learning

© Prof. Emilio Del Moral Hernandez

Autoencoders e Stacked Auto-encoders (imagens da internet)

15



Algumas ferramentas em Deep Learning

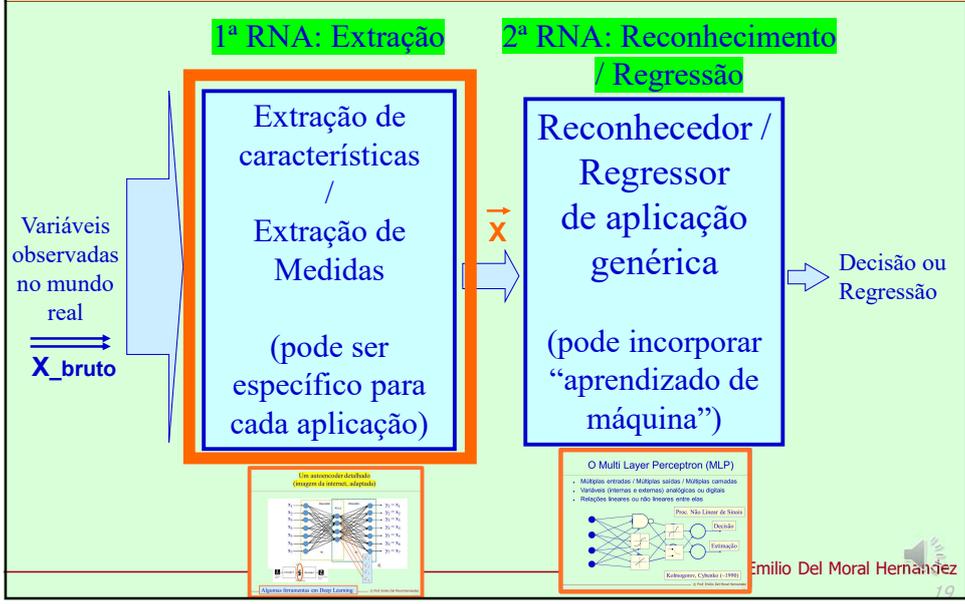
© Prof. Emilio Del Moral Hernández

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.

© Prof. Emilio Del Moral – EPU SP

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

19



20

20

2º tópico da agenda PSI3471 de 22 de abril de 2020:

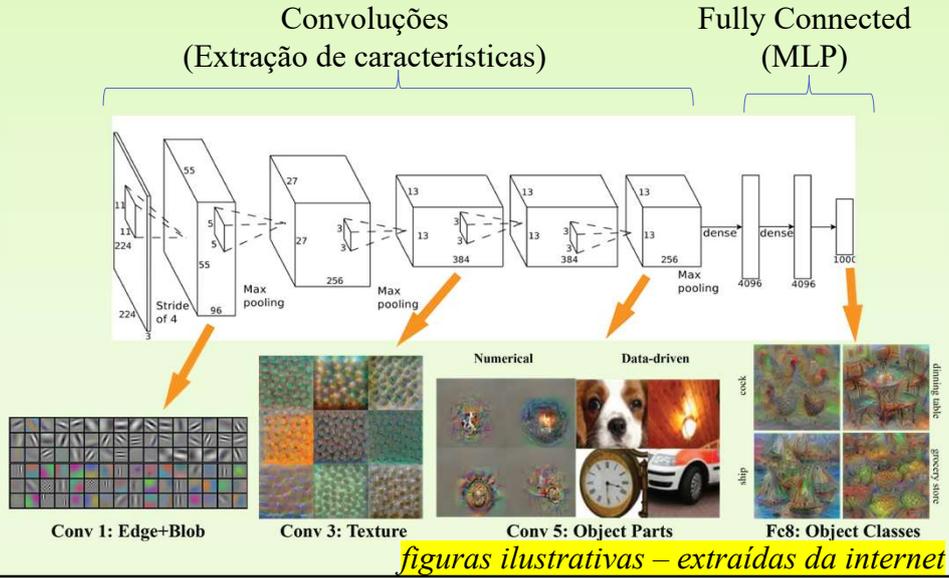
As redes neurais convolucionais:

*Atuando sobre sinais de tempo
Atuando sobre imagens*

***Falemos também sobre as
Redes Neurais Convolucionais –
ou ConvNets –
ou Convolutional Neural Networks***

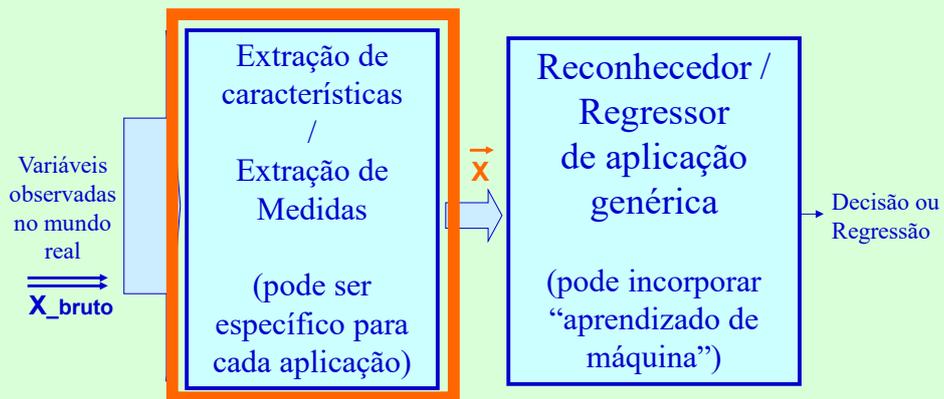
***Trata-se de outra técnica atual de grande
emprego no contexto de Deep Learning e
que também traz algumas soluções para o
sobreadaptnizado em aplicações de alta
dimensão de entrada***

Classificação em Redes Neurais Convolucionais



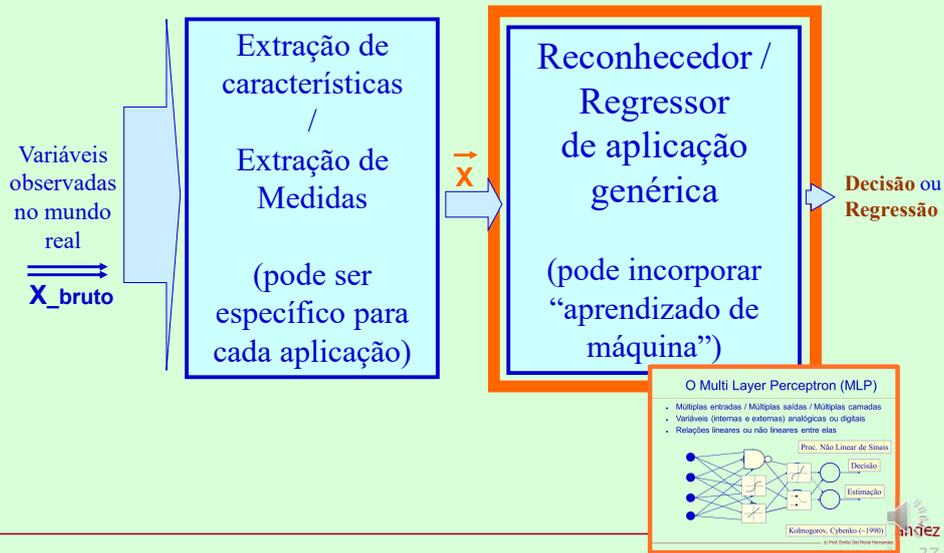
... O 1o estágio gera um Vetor de Medidas, \vec{X}
(o segundo estágio operará sobre tal vetor)

25



O segundo estágio opera sobre o Vetor de Medidas, \vec{X} (o 1o estágio gerou tal vetor)

27



um parênteses para aquecer motores ...

29

... as CNNs / ConvNets / CvNNs para imagens se relacionam de alguma forma com sinais de tempo (voz, elétricos e biológicos), cenário que explora já de muito as convoluções em sistemas SLIT ??? (sistemas lineares invariantes no tempo)

Soma de Convolução

- Lembrando:

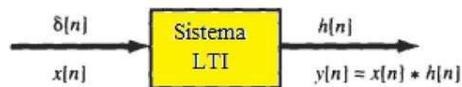
$$T\{\delta[n]\} = h[n]$$

- $h[n]$ – resposta ao impulso
- E que o sistema é LTI, logo:

$$h[n - k] = T\{\delta[n - k]\}$$

- Substituindo em

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] T\{\delta[n - k]\}$$



- Obtemos a **Soma de Convolução**

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k]$$

ou

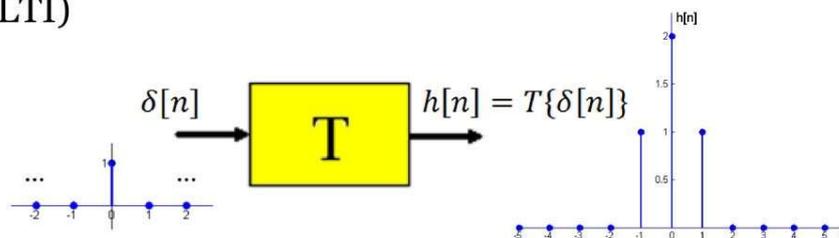
$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k]$$

- * representa convolução

$$y[n] = x[n] * h[n]$$

Resposta ao Impulso

- Seja o Sistema Linear Invariante no Tempo (LTI)

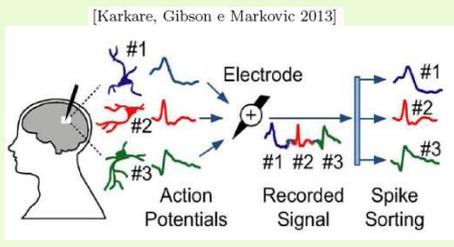


- $h[n]$ – resposta ao impulso

Relação entre Convolução e o conceito de Template matching e Correlação cruzada, em sinais de tempo ...

Neural Signal Processing in Analog CMOS IC Neural Spikes Detection

Júlio Cesar Saldaña Pumarica

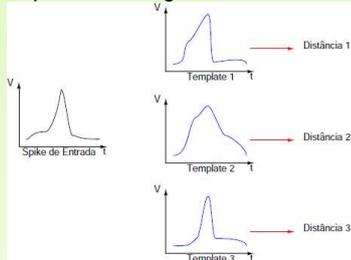


Neural Signal Processing in Analog CMOS IC

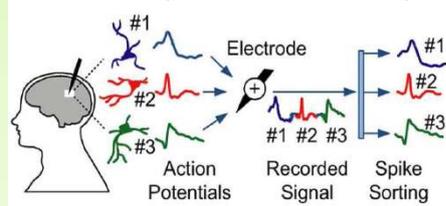
Neural Spikes Detection

Júlio Cesar Saldaña Pumarica

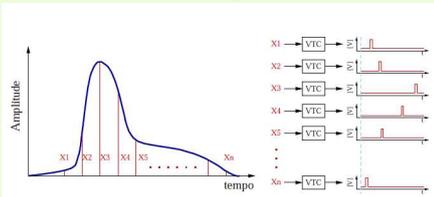
Template Matching based on minimum distance



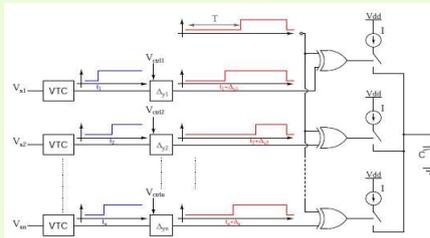
[Karkare, Gibson e Markovic 2013]



Time based encoding



Manhattan Distance



Sistemas Implantáveis de Registro Neural e BCIs

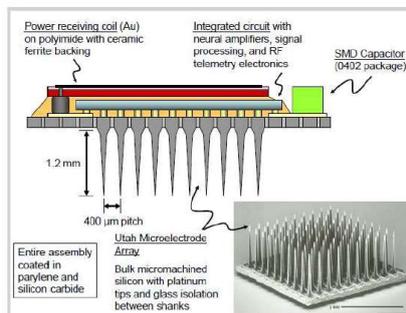


Figure 30.2.1: Complete Integrated Neural Interface (INI) assembly. [Harrison et al. 2006], [Harrison et al. 2007]

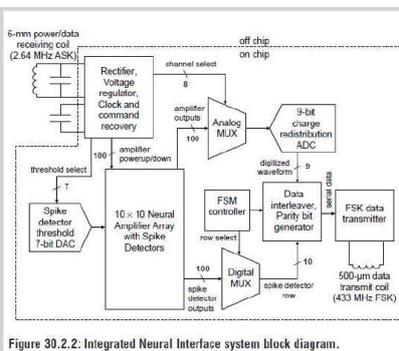


Figure 30.2.2: Integrated Neural Interface system block diagram.

Agora, vejamos a aplicação desse conceito do “Template matching / Correlação cruzada / Convolução” na área mais conhecida de uso de Redes Neurais Convolucionais

...

Reconhecimento de Imagens

The screenshot shows a web browser displaying a Medium article. The browser's address bar shows the URL: `medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c68850a207`. The Medium logo is visible at the top of the page. The author's profile is shown as Firdouss Doukkali, with a bio: "Machine Learning Engineer and Chief Unicorn Scientist. Global Shaper at World Economic Forum. English, French, German, Arabic, and Japanese speaker. @phidaouss". The article title is "Convolutional Neural Networks (CNN, or ConvNets)". The text of the article begins with: "Convolutional Neural networks allow computers to see, in other words, Convnets are used to recognize images by transforming the original image through layers to a class scores. CNN was inspired by the visual cortex. Every time we see something, a series of layers of neurons gets activated, and each layer will detect a set of features such as lines, edges. The high level of layers will detect more complex features in order to recognize what we saw." Below this, it says: "This article will present my brief notes about the elements that constitute Convolutional Neural Networks." and "ConvNet has two parts: feature learning (Conv, Relu, and Pool) and classification (FC and softmax)." At the bottom of the article preview, there is a notification: "Never miss a story from Firdouss Doukkali, when you sign up for Medium. Learn more" and a "GET UPDATES" button. The Windows taskbar is visible on the right side of the browser window, showing the time as 11:15 on 23/05/2018.

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b5da207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, j, 0]$	$w_0[i, j, 0]$	$w_1[i, j, 0]$	$o[i, j, 0]$
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	$w_0[i, j, 1]$	$w_1[i, j, 1]$	$o[i, j, 1]$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	2 -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
$x[i, j, 1]$	$w_0[i, j, 2]$	$w_1[i, j, 2]$	
0 0 0 0 0 0 0	-1 -1 1	-1 1 -1	
0 1 2 2 2 0 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			
$x[i, j, 2]$	Bias b_0 (1x1x1)	Bias b_1 (1x1x1)	
0 1 2 2 2 0 0	1	0	
0 0 1 3 0 1 0			
0 0 0 0 0 0 0			
0 0 2 1 0 1 0			
0 1 2 1 1 1 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 $K=3$ 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-d7c688b5da207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, j, 0]$	$w_0[i, j, 0]$	$w_1[i, j, 0]$	$o[i, j, 0]$
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	$w_0[i, j, 1]$	$w_1[i, j, 1]$	$o[i, j, 1]$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	2 -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
$x[i, j, 1]$	$w_0[i, j, 2]$	$w_1[i, j, 2]$	
0 0 0 0 0 0 0	-1 -1 1	-1 1 -1	
0 1 2 2 2 0 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			
$x[i, j, 2]$	Bias b_0 (1x1x1)	Bias b_1 (1x1x1)	
0 1 2 2 2 0 0	1	0	
0 0 1 3 0 1 0			
0 0 0 0 0 0 0			
0 0 2 1 0 1 0			
0 1 2 1 1 1 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 $K=3$ 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-d7c68850a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, j, 0]$	$w0[i, j, 0]$	$w1[i, j, 0]$	$o[i, j, 0]$
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	$w0[i, j, 1]$	$w1[i, j, 1]$	$o[i, j, 1]$
0 2 0 0 2 0 0	-1 0 1	1 0 0	-8 -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
$w[i, j, 1]$	$w0[i, j, 2]$	$w1[i, j, 2]$	
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 0 0 0 0 0	0 -1 0	1 0 0	
Bias b0 (1x1x1)	Bias b1 (1x1x1)		
0 1 2 2 2 2 0	0 1 2 2 2 0		
0 0 1 2 0 1 0	0 0 1 2 0 1 0		
0 0 0 0 0 0 0	0 0 0 0 0 0 0		
$w[i, j, 2]$			
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 0 0			
0 0 0 0 0 0 0			

A demo of a Conv layer with $K=3$ 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-d7c68850a207

Medium

Sign in Get started

Input Volume (pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[i, j, 0]$	$w0[i, j, 0]$	$w1[i, j, 0]$	$o[i, j, 0]$
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	$w0[i, j, 1]$	$w1[i, j, 1]$	$o[i, j, 1]$
0 2 0 0 2 0 0	-1 0 1	1 0 0	-8 -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
$w[i, j, 1]$	$w0[i, j, 2]$	$w1[i, j, 2]$	
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 0 0 0 0 0	0 -1 0	1 0 0	
Bias b0 (1x1x1)	Bias b1 (1x1x1)		
0 1 2 2 2 2 0	0 1 2 2 2 0		
0 0 1 2 0 1 0	0 0 1 2 0 1 0		
0 0 0 0 0 0 0	0 0 0 0 0 0 0		
$w[i, j, 2]$			
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 0 0			
0 0 0 0 0 0 0			

A demo of a Conv layer with $K=3$ 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

The diagram illustrates a convolutional layer with the following components:

- Input Volume (pad 1) (7x7x3):** A 7x7 grid of values for each of the 3 channels. The top-left 3x3 region is highlighted in blue.
- Filter W0 (3x3x3):** A 3x3 grid of weights for each of the 3 channels. The top-left 3x3 region is highlighted in red.
- Filter W1 (3x3x3):** A 3x3 grid of weights for each of the 3 channels. The top-left 3x3 region is highlighted in red.
- Output Volume (3x3x2):** A 3x3 grid of values for each of the 2 channels. The top-left 3x3 region is highlighted in green.
- Bias b0 (1x1x3):** A 1x1 grid of bias values for each of the 3 channels.
- Bias b1 (1x1x1):** A 1x1 grid of bias values for the 1 channel.

Arrows indicate the flow of information from the input volume through the filters and bias to the output volume. A "toggle movement" button is located at the bottom right of the diagram.

A demo of a Conv layer with $K=3$ filters, each with a spatial extent $F=3$, $m_{in}=1$, and a stride $S=2$, and m_{out} .

Never miss a story from Firdaouss Doukkali, when you sign up for Medium. Learn more [GET UPDATES](#)

... até que a imagem toda seja varrida por um mesmo “filtro / template”.

44

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c6885da207

CONV layer: The objective of a CONV layer is to extract features of the input volume.

Medium

Sign in Get started

164

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	$w0[i, i, 1]$	$w1[i, i, 1]$	$o[i, i, 1]$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	-8 -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 -3
$x[i, i, 1]$	$w0[i, i, 2]$	$w1[i, i, 2]$	
0 0 0 0 0 0 0	-1 1 1	-1 1 1	
0 2 1 2 1 1 0	1 1 0	0 2 1	
0 2 1 2 0 1 0	0 -1 0	2 0 0	
0 0 2 1 0 1 0			
0 0 1 2 2 2 0	Bias $b0(1x1x1)$	Bias $b1(1x1x1)$	
0 0 1 2 0 1 0	1	0	
0 0 0 0 0 0 0			
$x[i, i, 2]$			
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 0 1 0 2 1 0			
0 2 2 1 1 0 0			
0 0 0 0 0 0 0			

toggle movement

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

GET UPDATES

© Prof. Emilio Del Moral Hernandez

45

Convolutional Neural N

medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c6885da207

CONV layer: The objective of a CONV layer is to extract features of the input volume.

Medium

Sign in Get started

164

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	$w0[i, i, 1]$	$w1[i, i, 1]$	$o[i, i, 1]$
0 2 0 0 2 0 0	-1 0 1	-1 0 0	-8 -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 -3
$x[i, i, 1]$	$w0[i, i, 2]$	$w1[i, i, 2]$	
0 0 0 0 0 0 0	-1 1 1	-1 1 1	
0 2 1 2 1 1 0	1 1 0	0 2 1	
0 2 1 2 0 1 0	0 -1 0	2 0 0	
0 0 2 1 0 1 0			
0 0 1 2 2 2 0	Bias $b0(1x1x1)$	Bias $b1(1x1x1)$	
0 0 1 2 0 1 0	1	0	
0 0 0 0 0 0 0			
$x[i, i, 2]$			
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 0 1 0 2 1 0			
0 2 2 1 1 0 0			
0 0 0 0 0 0 0			

toggle movement

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

GET UPDATES

© Prof. Emilio Del Moral Hernandez

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

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

Teoria Redes Convolucionais

ConvNets - Convolução

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

faces	cars
	
	

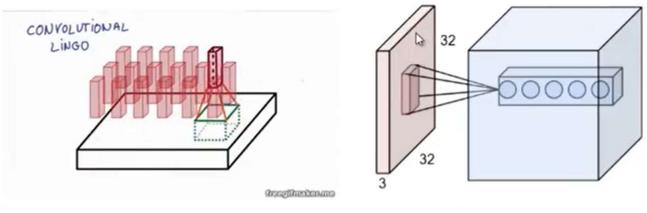
2:42 / 18:25

© Prof. Emilio Del Moral Hernandez

ConvNets – Convolução



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

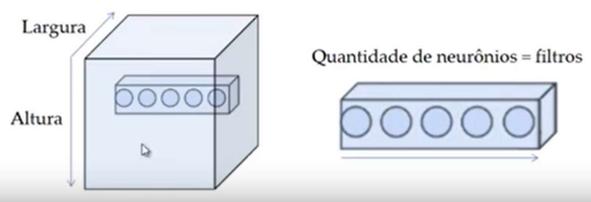


3:05 / 18:25

ConvNets - Convolução



- Os neurônios representam os filtros
- Neurônios que estão no mesmo plano composto pelos eixos da altura e largura, compartilham os mesmos pesos



6:01 / 18:25

Teoria Redes Convolucionais

ConvNets – Convolução

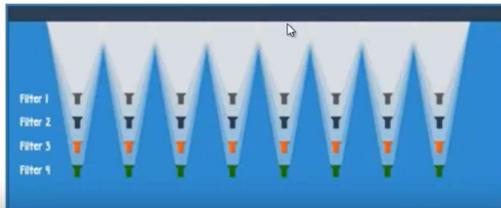


Diagram illustrating 1D convolution. It shows a horizontal row of 8 blue trapezoidal shapes representing input nodes. Below these nodes are four filters, labeled 'Filtro 1', 'Filtro 2', 'Filtro 3', and 'Filtro 4'. Each filter is represented by a smaller trapezoid with a colored top (green, orange, blue, and red respectively) and a black outline. The filters are positioned such that they overlap with the input nodes. A mouse cursor is visible over the diagram.

6:59 / 18:25

© Prof. Emilio Del Moral Hernandez

52

Teoria Redes Convolucionais

ConvNets – Convolução

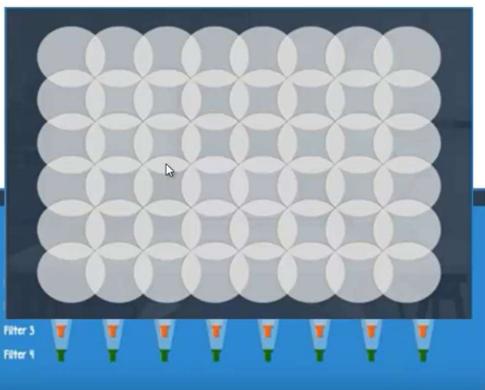


Diagram illustrating 2D convolution. It shows a 5x5 grid of light gray circular nodes. Below the grid are two filters, labeled 'Filtro 3' and 'Filtro 4'. Each filter is represented by a smaller circle with a colored top (orange and red respectively) and a black outline. The filters are positioned such that they overlap with the nodes in the grid. A mouse cursor is visible over the diagram.

7:34 / 18:25

© Prof. Emilio Del Moral Hernandez

53

Teoria Redes Convolucionais

ConvNets - Convolução

Hiperparâmetros da camada de convolução

- **F** = Tamanho do filtro – F x F
- **S** = *Stride* – Deslocamento de *pixels* do filtro na convolução
- **K** = Quantidade de filtros
- **W** = Tamanho da entrada – W x W
- **P** = *Zero-Padding* – Adiciona zeros na periferia das imagens

Área da Saída!

$$\frac{W - F + 2P}{S} + 1$$

9:26 / 18:25

© Prof. Emilio Del Moral Hernandez

Teoria Redes Convolucionais

ConvNets – Demais Camadas

Existem outras camadas típicas nas redes convolucionais:

- ReLUs – *Rectified Linear Units*
- *Pooling*
- *Fully Connected – Layer*

11:39 / 18:25

© Prof. Emilio Del Moral Hernandez

ConvNets - ReLUs



Camada que aplica operação linear matricial para descartar os valores negativos da convolução entre os filtros e a imagem

Rectified Linear Units (ReLUs)



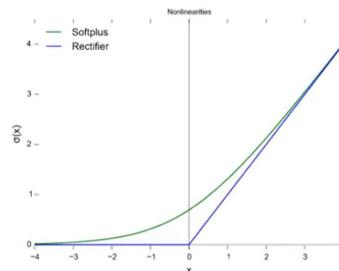
ConvNets - ReLUs



Função de Ativação

$$f(x) = \max(0, x)$$

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$



ConvNets – Pooling Layer



Camada responsável em identificar os filtros ativados, descartando os valores insignificantes – Diminui as dimensões ao longo da rede.



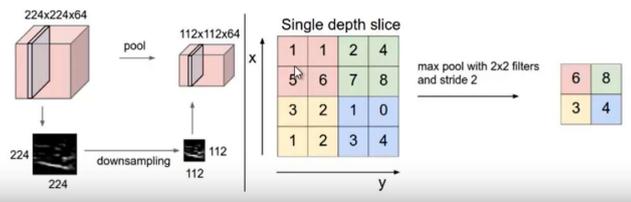
13:34 / 18:25

© Prof. Emilio Del Moral Hernandez

ConvNets – Pooling Layer



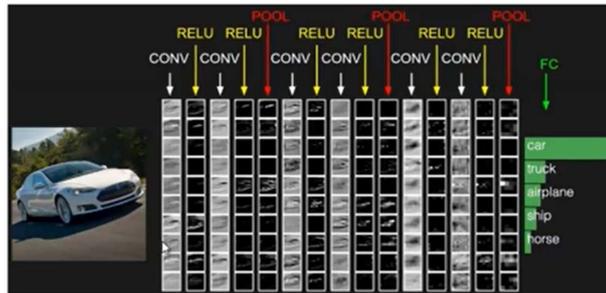
- Camadas com filtros 2 x 2 (F = 2) deslocando dois pixels ao longo da imagem (S = 2)
- De cada janela, extrai-se o maior número
- Realizam *downsampling* nas imagens



14:55 / 18:25

© Prof. Emilio Del Moral Hernandez

Topologia Típica

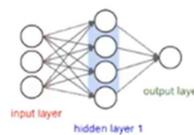
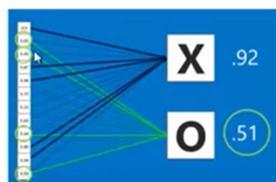


17:20 / 18:25

ConvNets – FC Layer



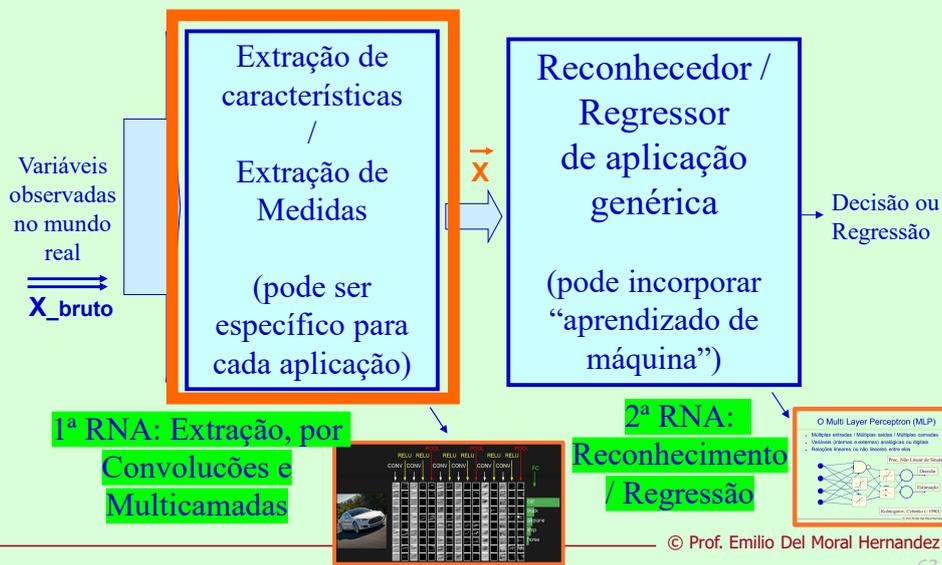
- Com a sequência de combinações de camadas de convolução com *pooling*, a dimensão é reduzida até atingir o formato de um vetor, o qual alimenta uma *fully-connected layer*, permitindo assim a classificação:



16:48 / 18:25

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

63



Sugiro fortemente aproveitar este vídeo de 18mins, sobre as ConvNets / CNNs

64

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

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

Para refletir e elaborar em escrito ...

66

Vários dos elementos construtivos / de arquitetura e de treinamento e operação / inferência em uma ConvNN como a apresentada, estão diretamente relacionados com tudo o aprendido nas 12 aulas anteriores ... Elenque (e elabore) ao menos um elemento ou dois para cada aula!

© Prof. Emilio Del Moral Hernandez

66

67

--- Aulas do Prof Emilio em PS13471 da #1 (em 02/março) à #13 (e 22/abril) ---

Plano aproximado de temáticas aula a aula / semana a semana – versão de 01 de março
(atividades com entreas simples aproximadamente semanais ocorrerão durante as aulas)

#1 (02/março – 2ªf) Conceitos em regressão linear e não linear. Regressão univariada e Regressão multivariada; Contrastes com técnica mais simples já conhecida: a regressão linear univariada (reta média). Diferenças principais entre 1) a Regressão e 2) o Reconhecimento / Detecção / Identificação de Padrões Multivariados. Conceitos em Reconhecimento / Detecção / Identificação de Padrões. Entendendo primeiro as ferramentas de modelagem mais simples: as ferramentas lineares multivariadas ... elementos matemáticos em regressão linear multivariada e em discriminadores lineares multivariados. Dados empíricos para calibração de parâmetros em regressores e em discriminadores; o aprendizado supervisionado.

#2 (04/março – 4ªf) Neurônios biológicos como base para os neurônios artificiais; equações e propriedades matemáticas do neurônio artificial isolado; relação do neurônio isolado com os regressores e com os discriminadores lineares multivariados.

#3 (09/março – 2ªf) Foco nas Redes de Neurônios Artificiais (RNAs) concatenando múltiplos neurônios; arquiteturas neurais diversas; arquiteturas MLP – Multi Layer Perceptron; Relação de RNAs com a implementação de regressores e reconhecedores **NÃO LINEARES MULTIVARIADOS**. O ensaio computacional de redes neurais – ambientes simples como o MBP – Multiple Back Propagation – e ambientes mais complexos alternativos, de sua escolha.

#4 (11/março – 4ªf) Grafos da computação em redes neurais do tipo MLP; notação matemática para pesos sinápticos, variáveis de entrada, saída e variáveis intermediárias no MBP; exemplares empíricos / observações empíricas para o treinamento supervisionado em MLPs; aprendizado com base na propagação reversa do erro (EBP – Error Back Propagation).

© Prof. Emilio Del Moral Hernandez

67

68
<p>#5 (16/março – 2ªf) Foco no aprendizado da Rede Neural MLP – O Gradiente descendente e a otimização de pesos sinápticos com base no conjunto de treino e EBP; dedução das fórmulas do EBP, em sala de aula e em conjunto com os alunos: trabalho focado num peso sináptico específico da rede, escolhido pelo professor para máxima complexidade da dedução.</p> <p>#6 (18/março – 4ªf) Discussão da extensão simples das deduções do EBP para demais pesos sinápticos; redundâncias nos cálculos dos diversos pesos da rede neural e otimização do esforço computacional. Regra “Delta” de aprendizado de Widrow, para neurônio isolado; Aprendizado por EBP recursivo, camada a camada.</p> <p>#7 (23/março – 2ªf) Foco em Reconhedores – Circuitos lógicos genéricos e suas implementações através de computação neural; relação com discriminadores. Construindo reconhedores de padrões com fronteiras de separação genéricas: separadores lineares; bancos de separadores lineares como aproximadores efetivos de fronteiras genéricas.</p> <p>#8 (25/março – 4ªf) Técnicas de medida de qualidade em reconhedores de padrões: matriz de confusão, conceitos de falsos positivo e negativo, especificidade e sensibilidade. Curvas ROC.</p> <p>#9 (30/março – 2ªf) Foco agora em Regressores - Casos simples de aproximação de funções univariadas. Teorema de Cybenko: o MLP como aproximador universal de funções multivariadas; implicações práticas do teorema para a implementação de regressores e reconhedores de padrões não lineares multivariados genéricos.</p> <p>#10 (01/abril – 4ªf) Medidas de qualidade diversificadas para regressores multivariados (distintas do erro quadrático médio); Flutuação do desempenho do modelo com as particulares amostras de treino e de teste e técnicas de reamostragem; técnica de validação cruzada, k-fold cross validation e leave one out. Sobreajuste / sobreaprendizado / perda de generalização em regressão polinomial e em redes neurais; limitação do número de nós neurais para evitar o sobreajuste e otimizar a generalização da rede neural; partição do volume de observações em conjuntos de treino, validação e teste.</p> <p style="text-align: center;"><i>06 e 08 de abril: Semana Santa – não há aula</i></p>
© Prof. Emilio Del Moral Hernandez 68

69
<i>06 e 08 de abril: Semana Santa – não há aula</i>
<p>#11 (13/abril – 2ªf) Soluções em reconhecimento e em regressão empregando dois estágios: primeiro estágio para extração de características / extração de medidas e para a redução de dimensionalidade + segundo estágio MLP como reconhecedor / regressor genérico; técnicas diversas de extração de características em projetos concretos. Extensões de análise harmônica (Fourier) para a extração de características; conceitos básicos em análise tempo-frequência. PCA - Análise de Componentes Principais como extrator de características.</p> <p>#12 (15/abril – 4ªf) Aprofundando a conexão dos temas que aprendemos com o Deep Learning; conexão com redes neurais convolucionais; camadas convolucionais, Autoencoders e Autoencoders em cascata como extratores de características e redutores de dimensionalidade. Apanhado das temáticas aprendidas e troca de ideias sobre a “A4 de consulta” + sessão de dúvidas p/ P1;</p> <p style="text-align: center;"><i>22 de abril (4ªf) e 27 de abril (2ªf): P1 em PSI3471 – ou seja, dentro da semana de provas P1 da Elétrica (em 20 de abril, pela ponte de Tiradentes, não há aula na 2ª feira)</i></p> <p>#13 (22/abril – 4ªf) 1) Parte das temáticas e atividades desta aula após a P1 serão definidas mais adiante no semestre, de acordo com a evolução das aulas anteriores; 2) Conexão do aprendido até o momento com as próximas temáticas no curso.</p>
© Prof. Emilio Del Moral Hernandez 69