

Redes Convolucionais e *Deep Learning*

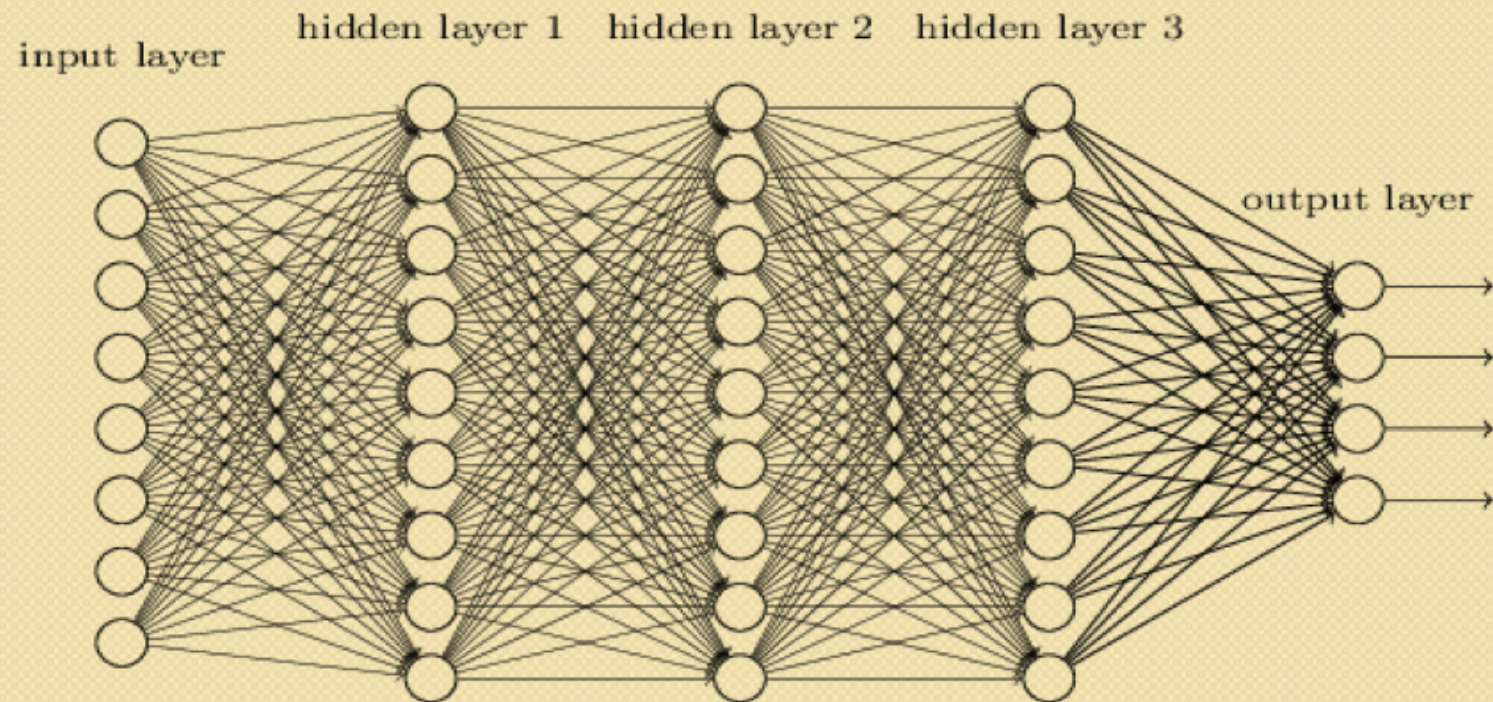
Germano C. Vasconcelos
Centro de Informática – UFPE

Hubel & Wiesel (1959)

https://www.youtube.com/watch?v=y_I4kQ5wjiw

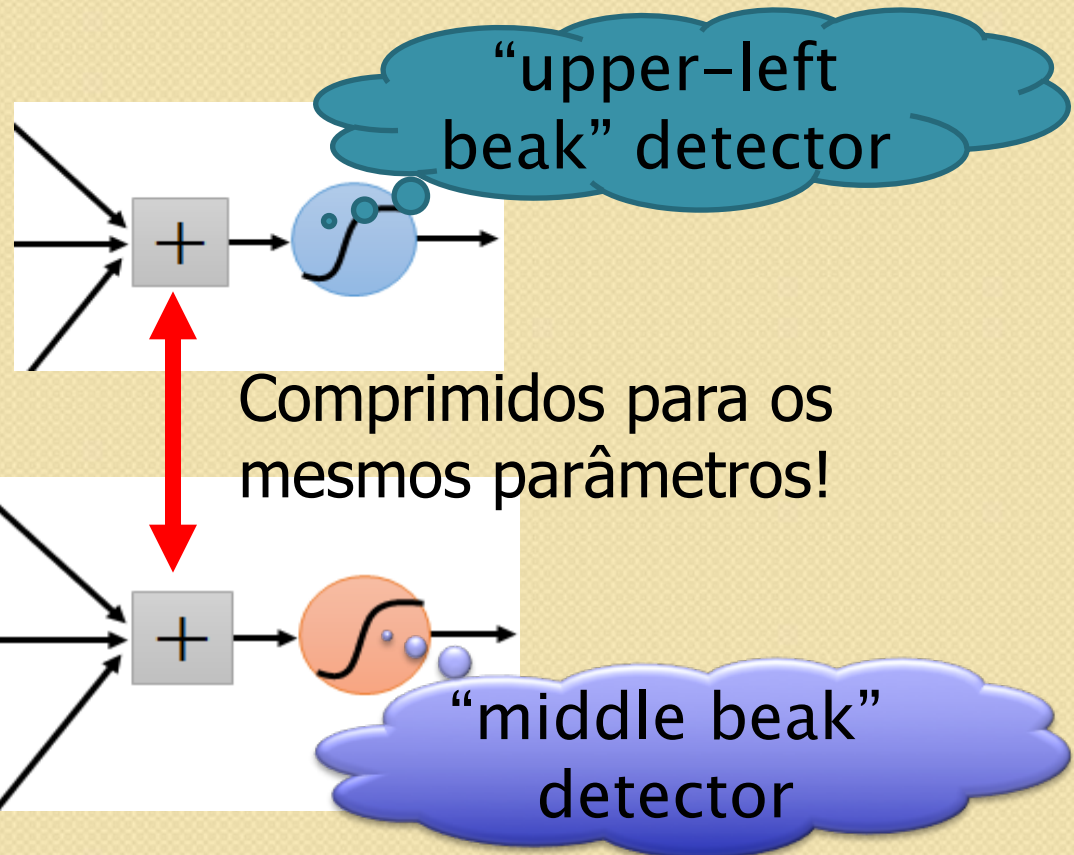
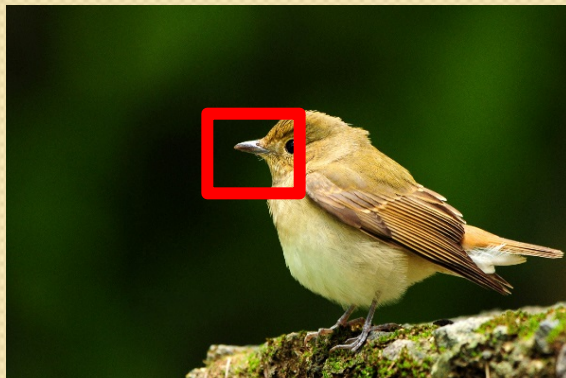
Estrutura Feedforward Tradicional

- Rede Totalmente Conectada (Fully Connected)



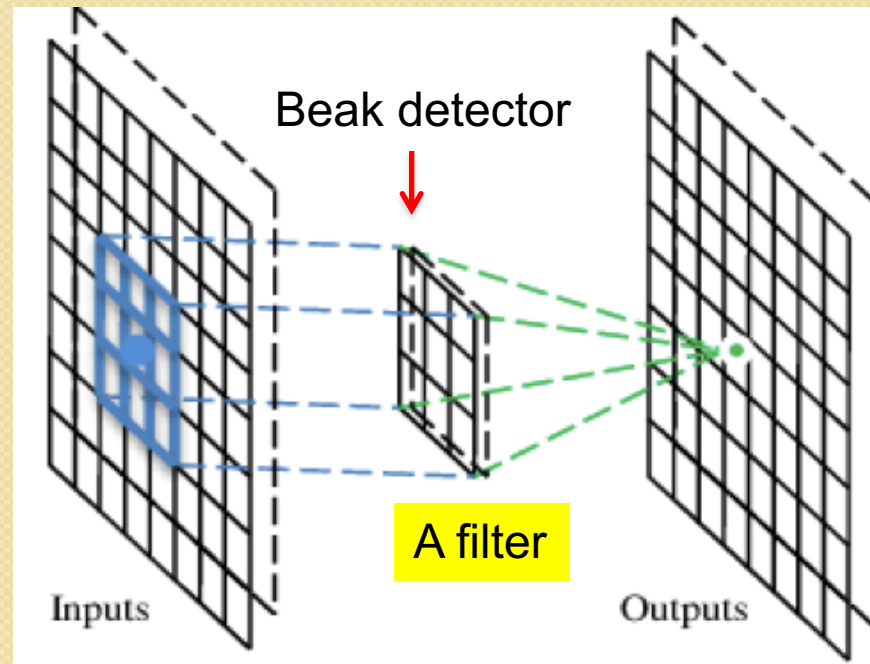
Como Lidar com Padrões em Áreas Diferentes?

Que tal detectores em áreas diferentes?



1o Passo: Camada Convolutucional

- CNN: rede neural com camadas convolucionais e outras camadas
- Camada convolutucional: filtros que fazem operação de convolução



Convolução

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Parâmetros a serem aprendidos!

1	-1	-1
-1	1	-1
-1	-1	1

Filtro 1

-1	1	-1
-1	1	-1
-1	1	-1

Filtro 2

⋮ ⋮

Cada filtro detecta
pequeno padrão(3 x 3)

Convolução

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Dot
product
→

3

-1

Convolução

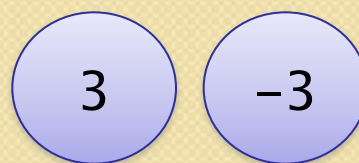
If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

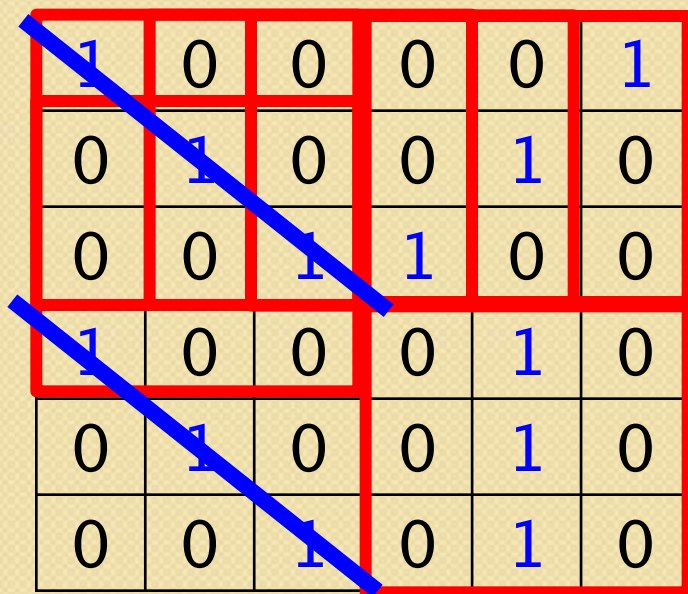
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1



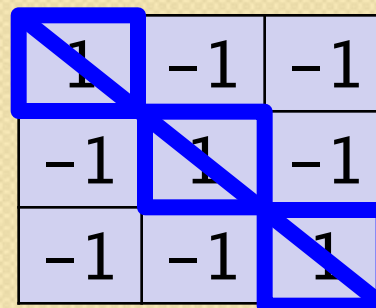
Convolução

stride=1



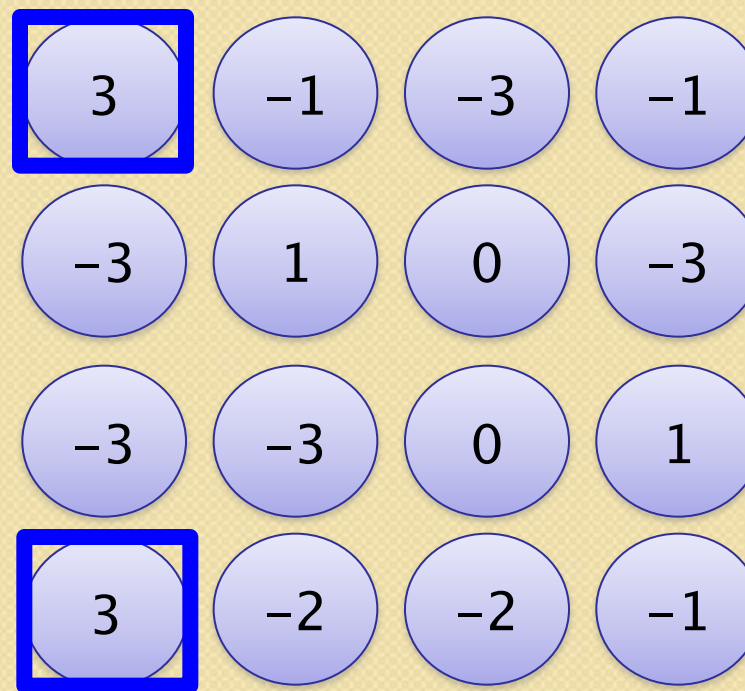
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



1	-1	-1
-1	1	-1
-1	-1	1

Filtro 1



3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Convolução

stride=1

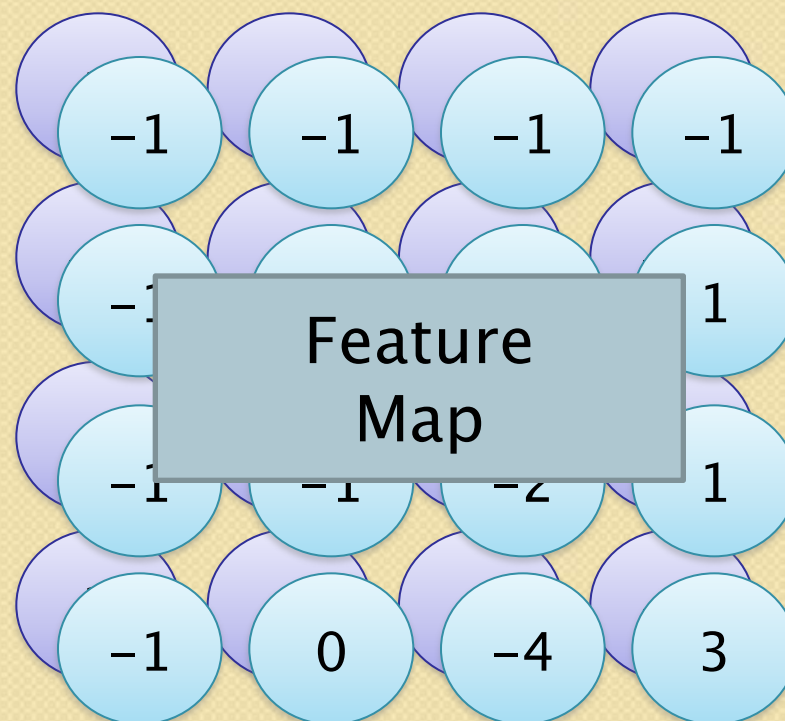
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

-1	1	-1
-1	1	-1
-1	1	-1

Filtro 2

Repita para cada filtro



Duas imagens 4 x 4
Formando matriz 2 x 4 x 4

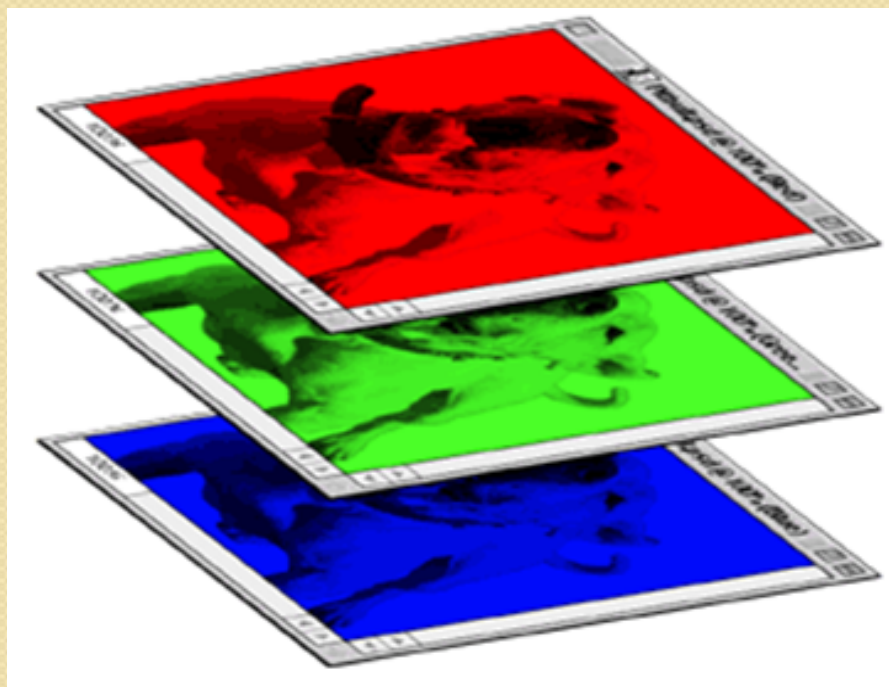
Imagem Colorida: 3 Canais RG

1	-1	-1
-1	1	-1
-1	-1	1

Filtro 1

-1	1	-1
-1	1	-1
-1	1	-1

Filtro 2



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Convolução vs Totalmente Conectada

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

imagem

1	-1	-1
-1	1	-1
-1	-1	1

-1	1	-1
-1	1	-1
-1	1	-1

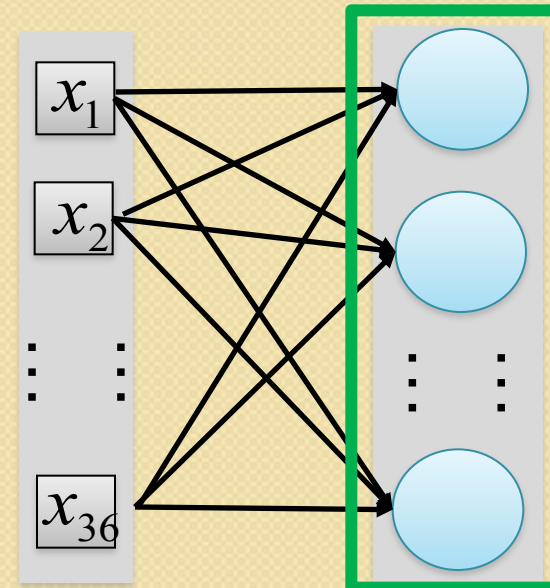


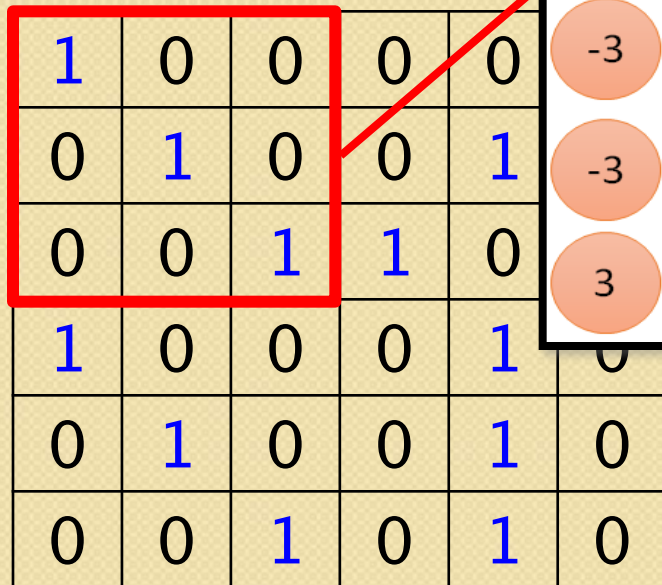
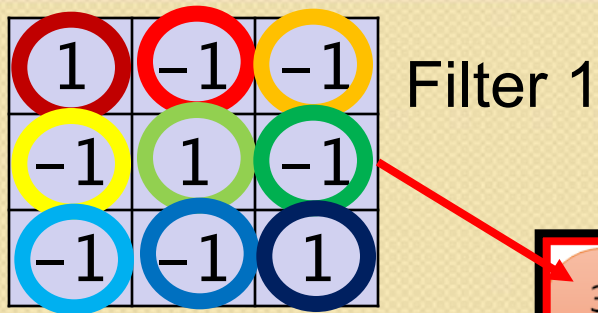
convolução

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Fully-
connected

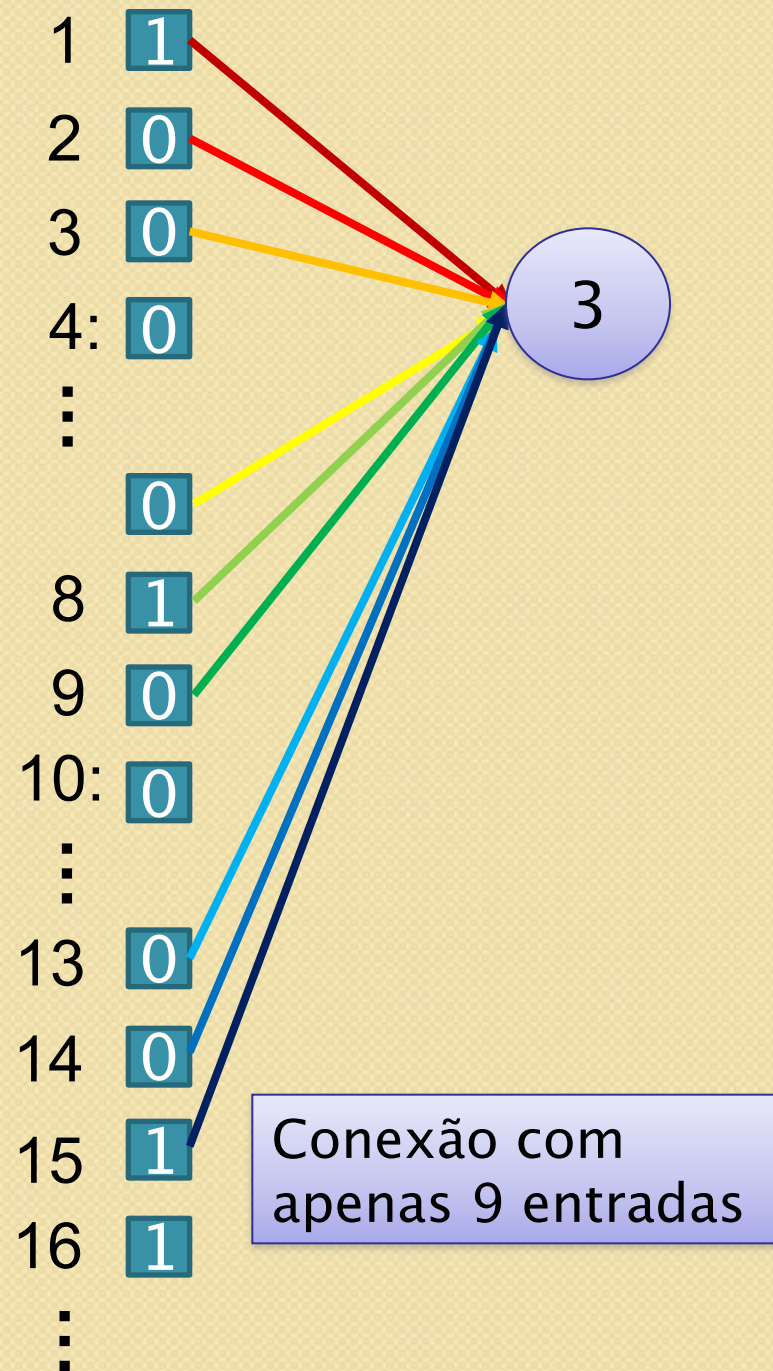
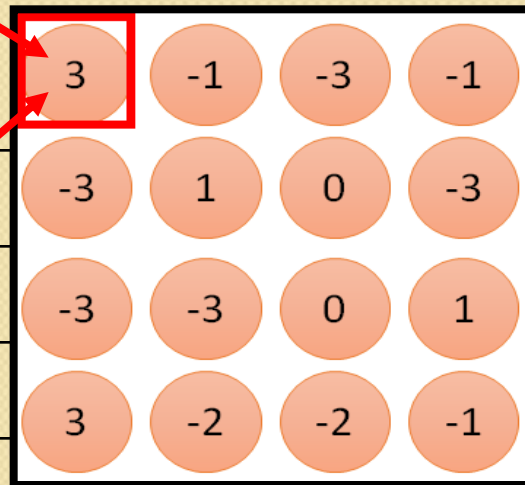
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

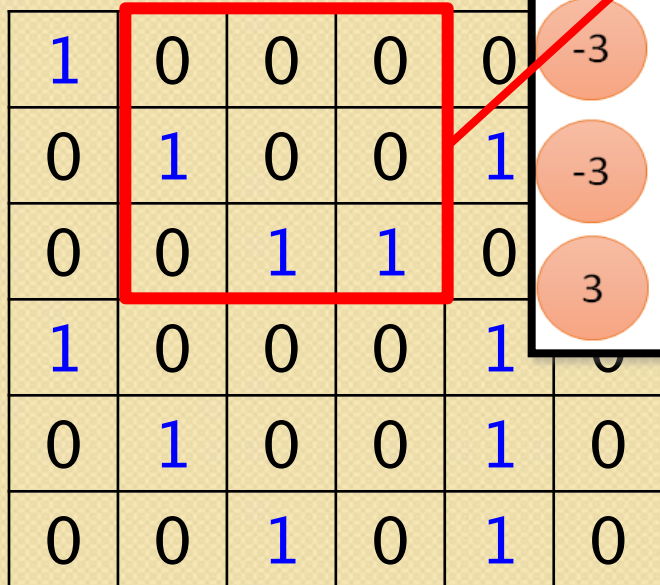
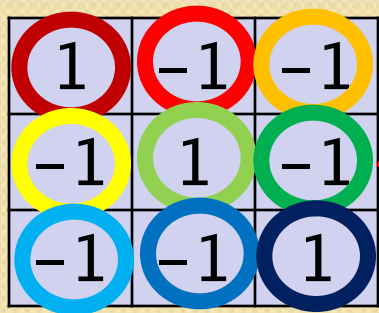




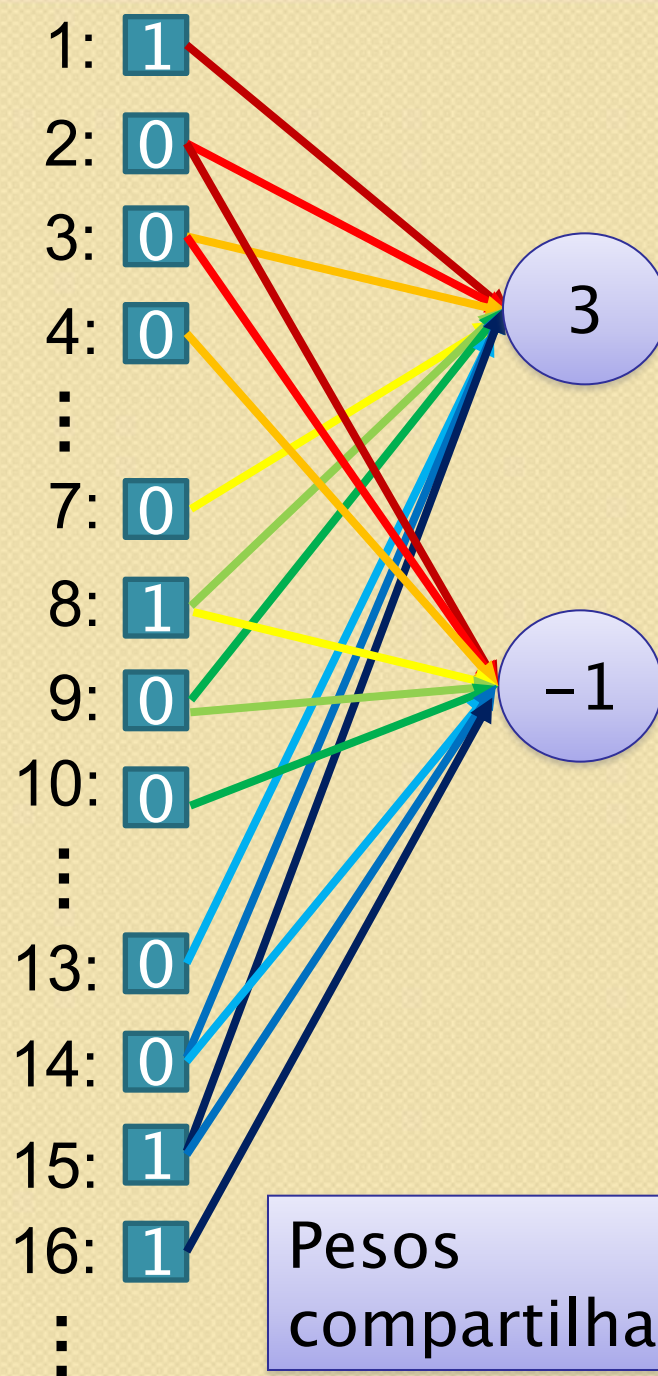
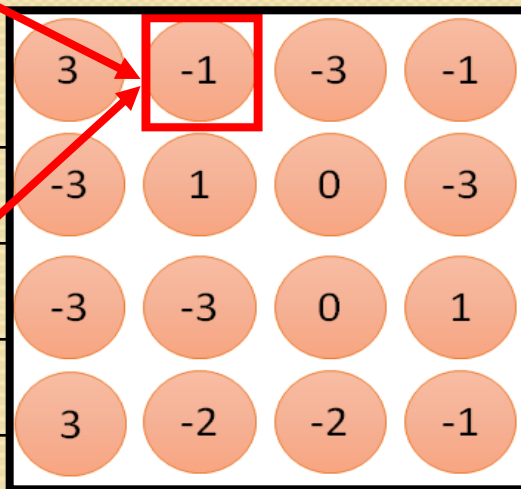
6 x 6 image

Menos parâmetros





6 x 6 image



Pesos
compartilhados

Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filtro 1

-1	1	-1
-1	1	-1
-1	1	-1

Filtro 2

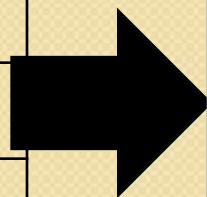
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Max Pooling

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

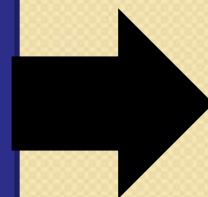
6 x 6 image



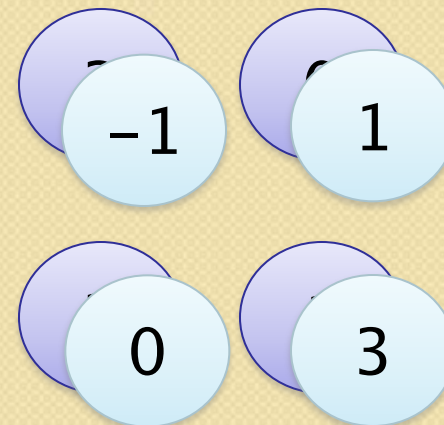
Conv



Max
Pooling



Nova imagem
menor



2 x 2 image

Cada filtro
é um canal

Convolução...

Main CNN idea for text:

Compute vectors for n-grams and group them afterwards

Example: "this takes too long" compute vectors for:

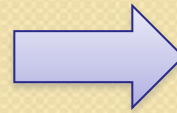
This takes, takes too, too long, this takes too, takes too long, this takes too long

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input matrix

1	0	1
0	1	0
1	0	1

Convolutional
3x3 filter



1 _{x0}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

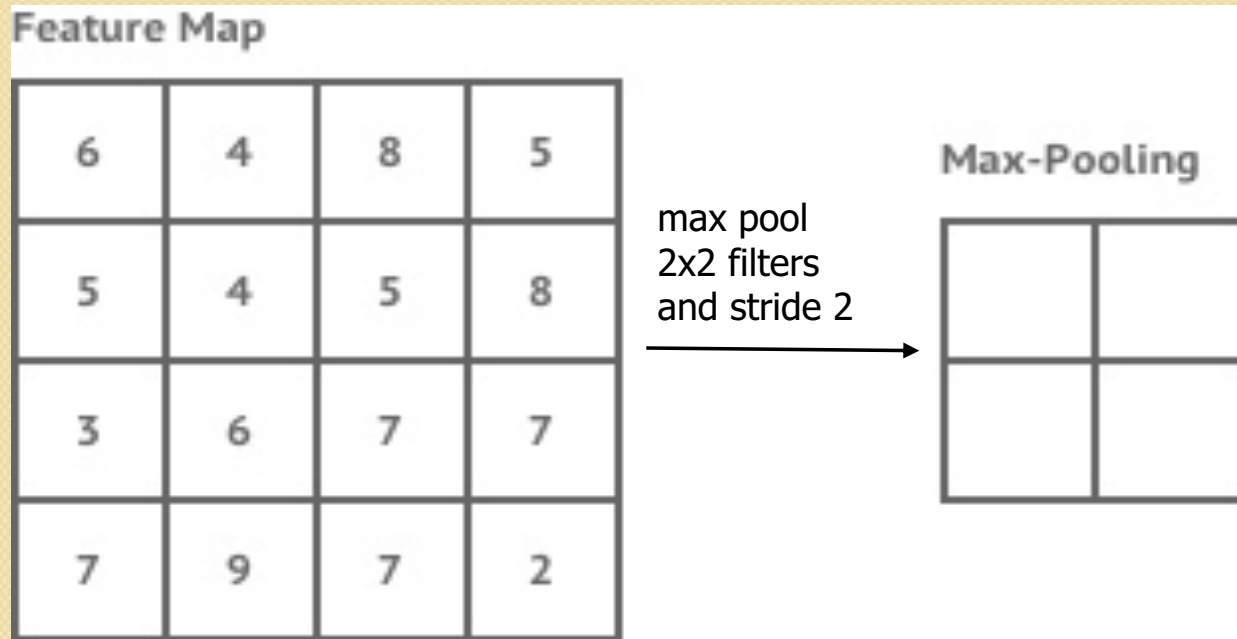
Convolved
Feature

http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

Max Pooling

Main CNN idea for text:

Compute vectors for n-grams and **group them afterwards**



Por Que Fazer *Pooling*?

- Subsampling não muda o objeto
pássaro



Subsampling



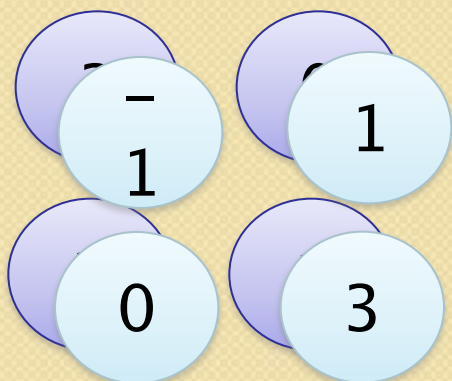
pássaro

Subamostrar pixels para tornar imagem menor



Menos parâmetros para caracterizar a imagem

CNN Completa



Uma nova imagem

Menor que a original

Número de canais=Número de Filtros



Convolution

Max Pooling

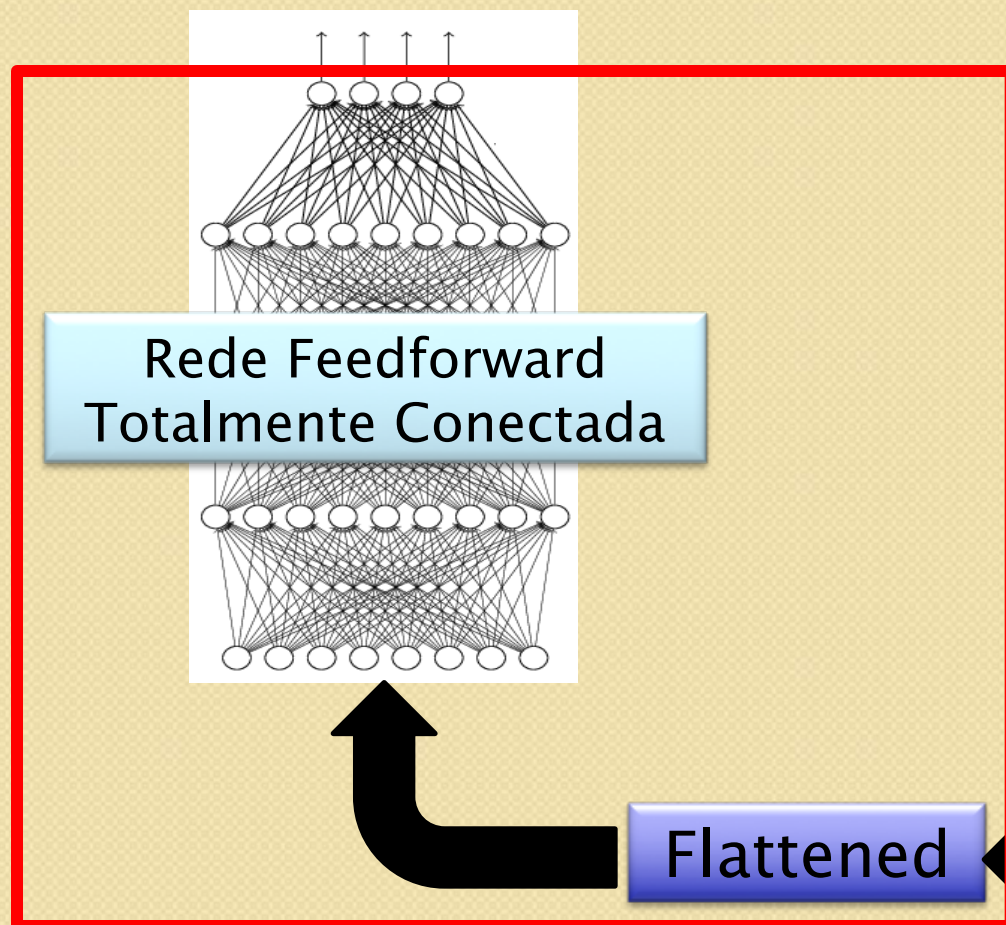
Convolution

Max Pooling

Pode se
repetir
várias
vezes

CNN Completa

cat dog



Convolution

Max Pooling

Nova imagem

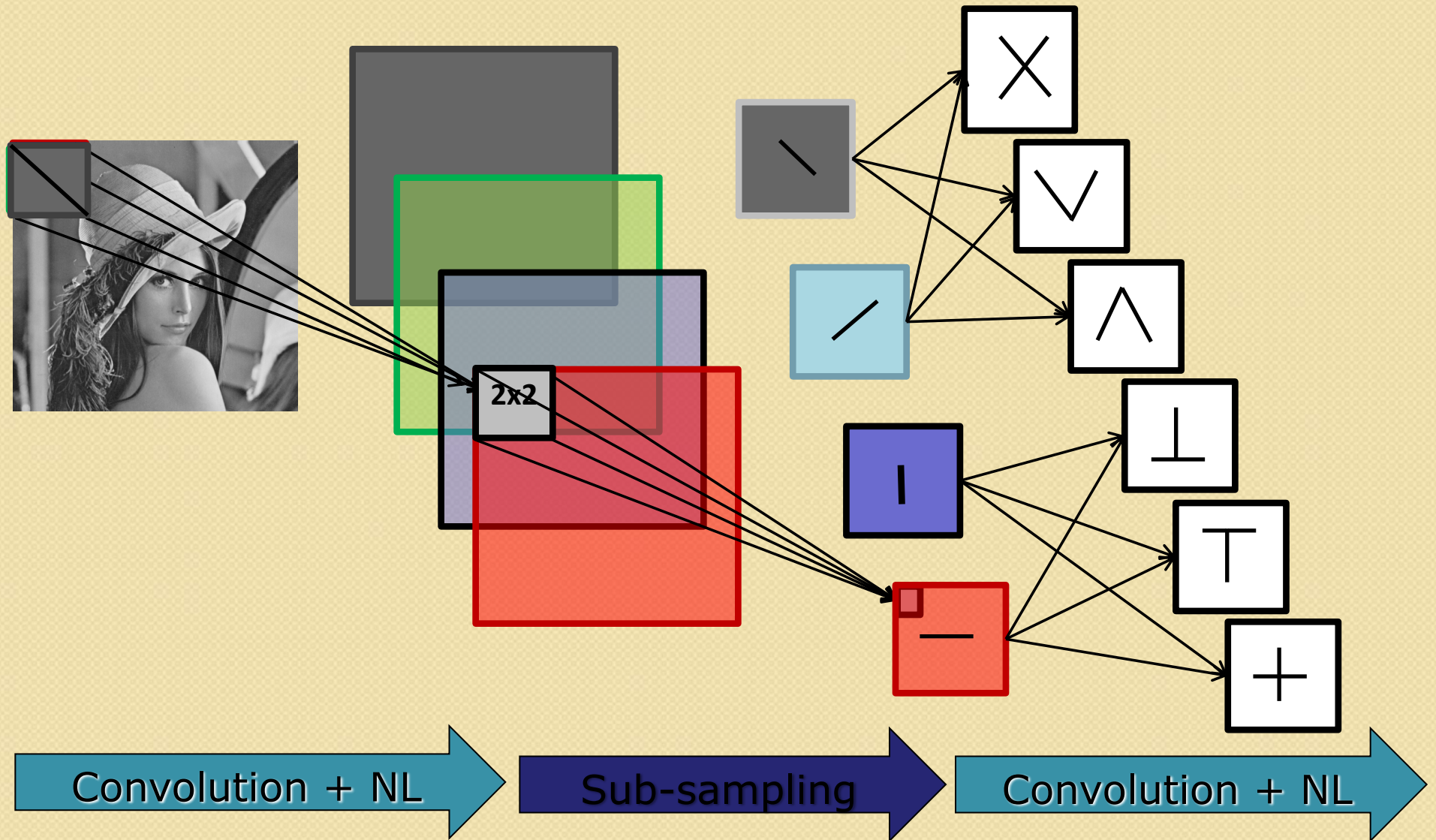
Convolution

Max Pooling

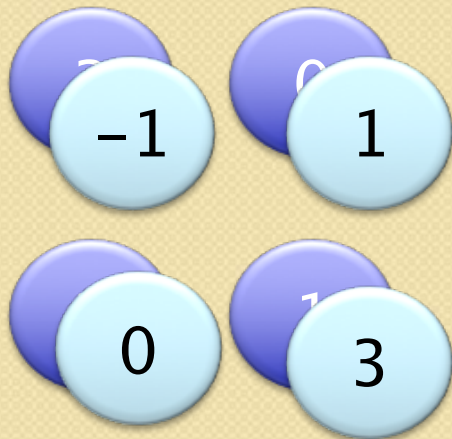
Nova imagem

Flattened

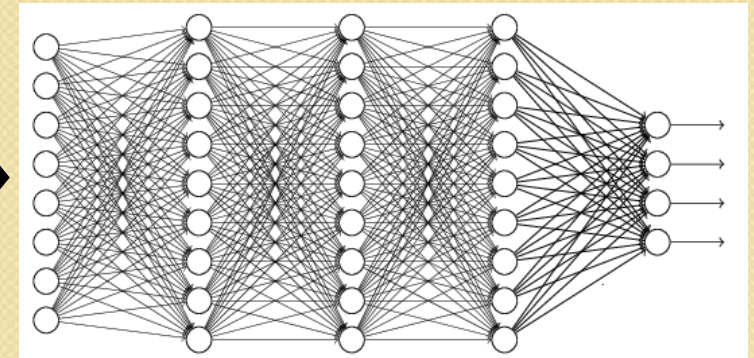
Rede Neural CNN



Achatamento (Flattening)

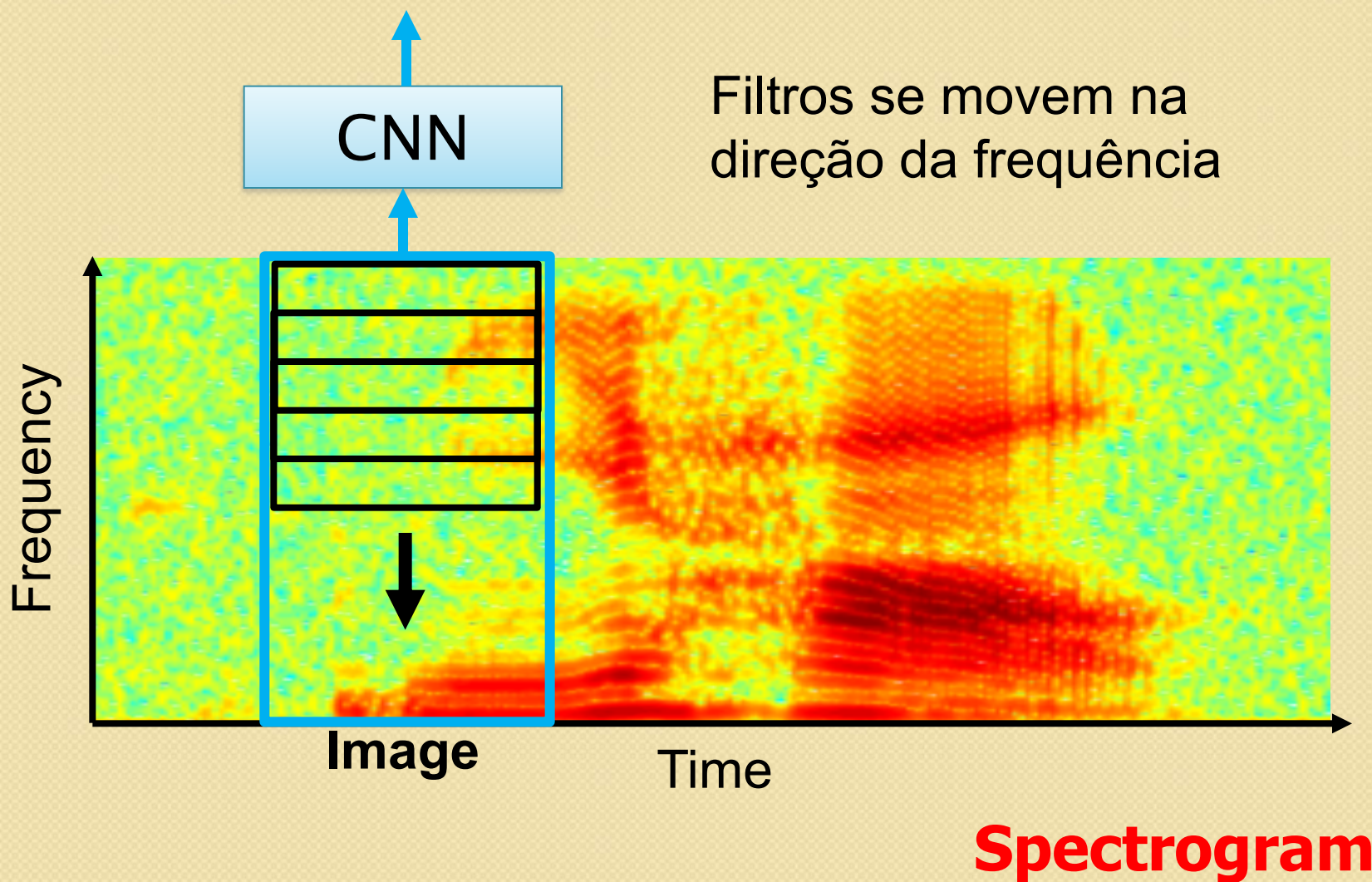


Achatada

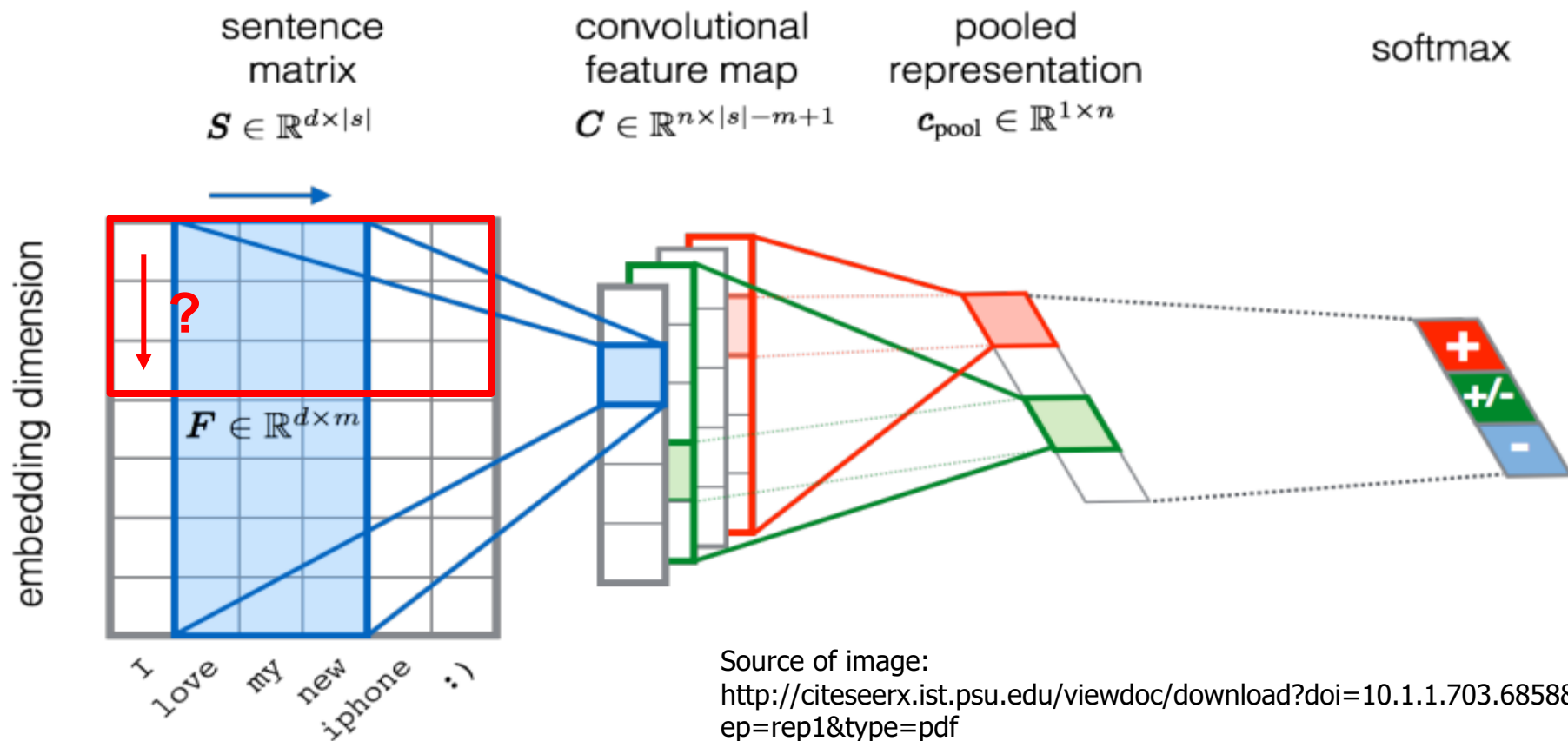


Rede Feedfoward
Totalmente Conectada

CNN em Reconhecimento da Fala



CNN em Classificação de Textos



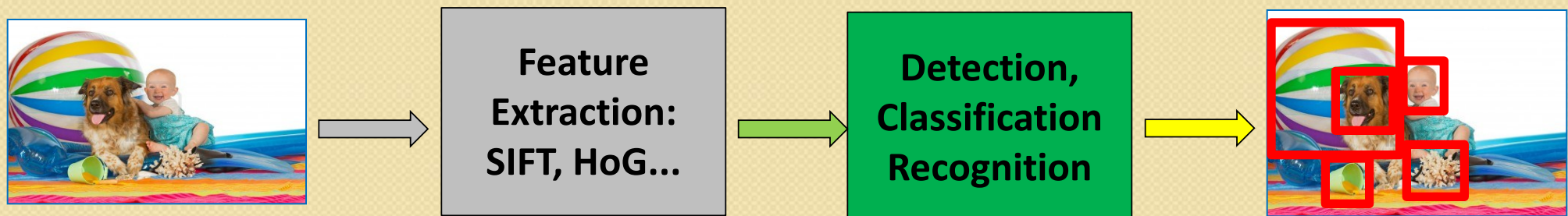
Solução Típica em Visão Computacional



Para reconhecer coisas?

Solução Típica em Visão Computacional

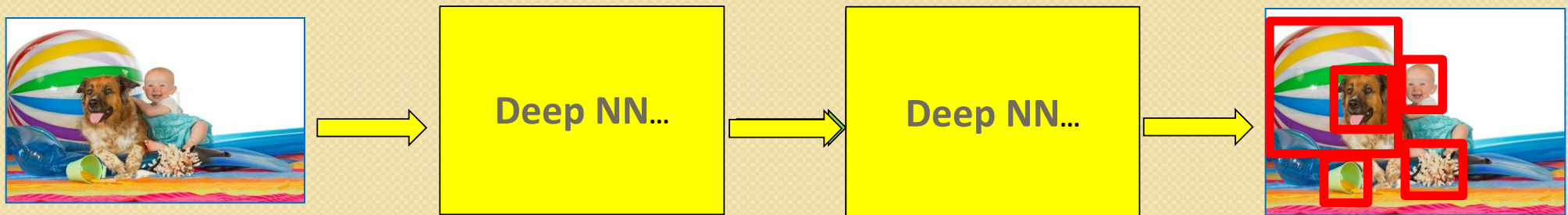
1. Definir e seleccionar características:
2. SURF, HoG, SIFT, RIFT.
3. Acrescentar classificadores



Definição de características é dependente do domínio e consome tempo

Solução em Deep Learning: Pipeline Baseado em Visão

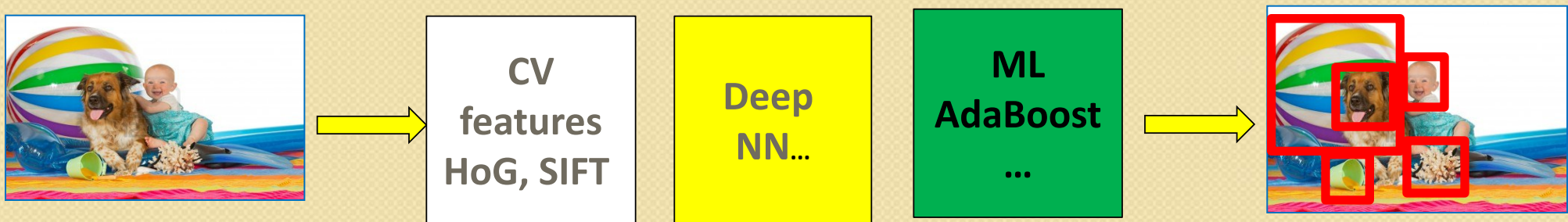
- Extrair características automaticamente baseado nos dados
- Combina a extração com classificação
- Papel do especialista: definir topologia e treinamento



Promessa do Deep Learning?
Treinar boas características automaticamente
Mesmo método para diferentes domínios

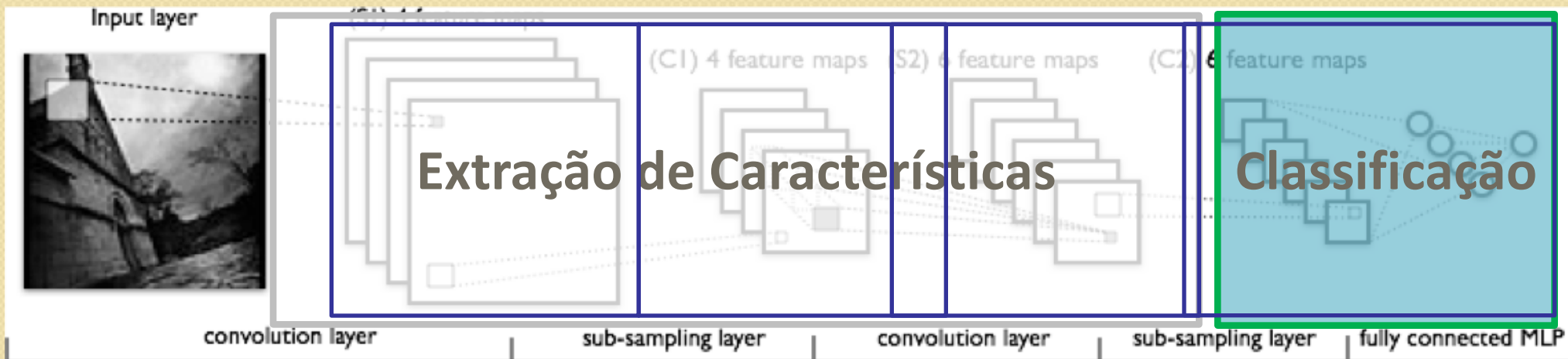
Visão Computacional + Deep Learning + Machine Learning

- Combinar características pré-definidas com características aprendidas
- Melhores métodos para classificação de múltiplas classes
- Especialistas em CV+DL+ML necessários para criar melhores soluções

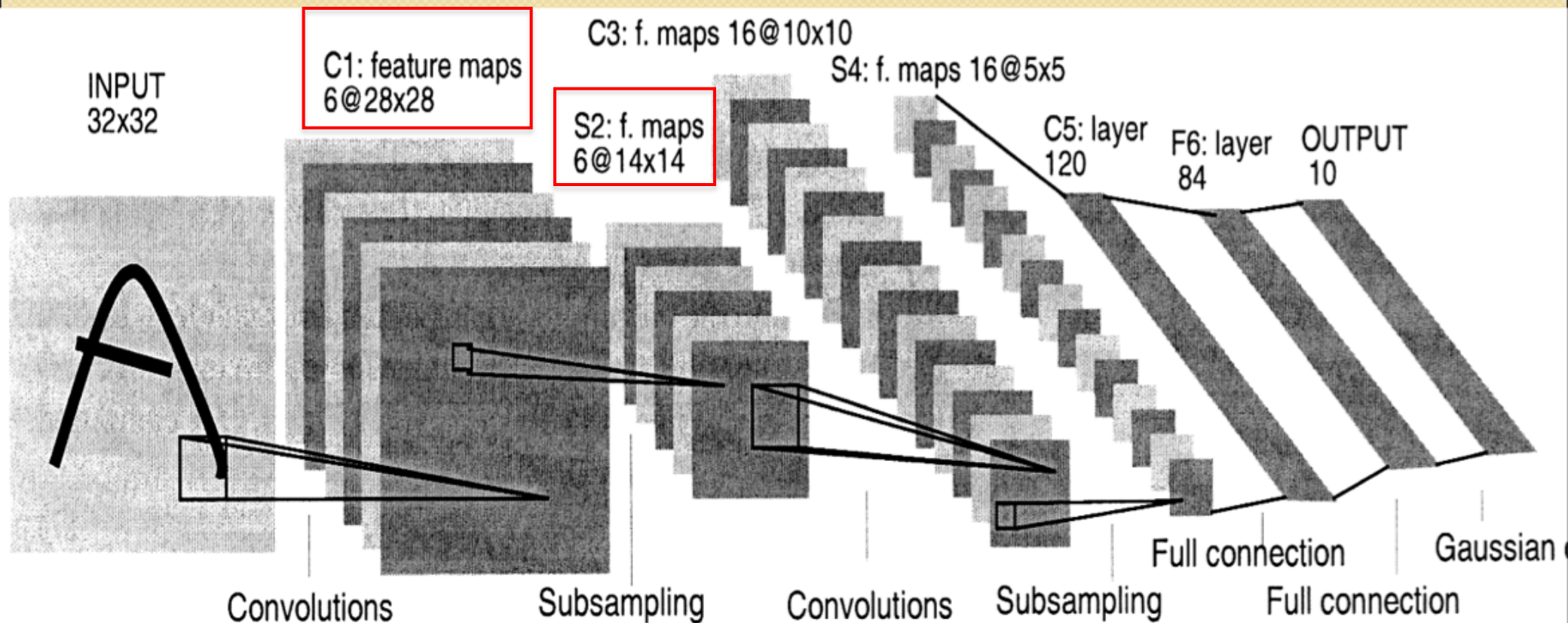


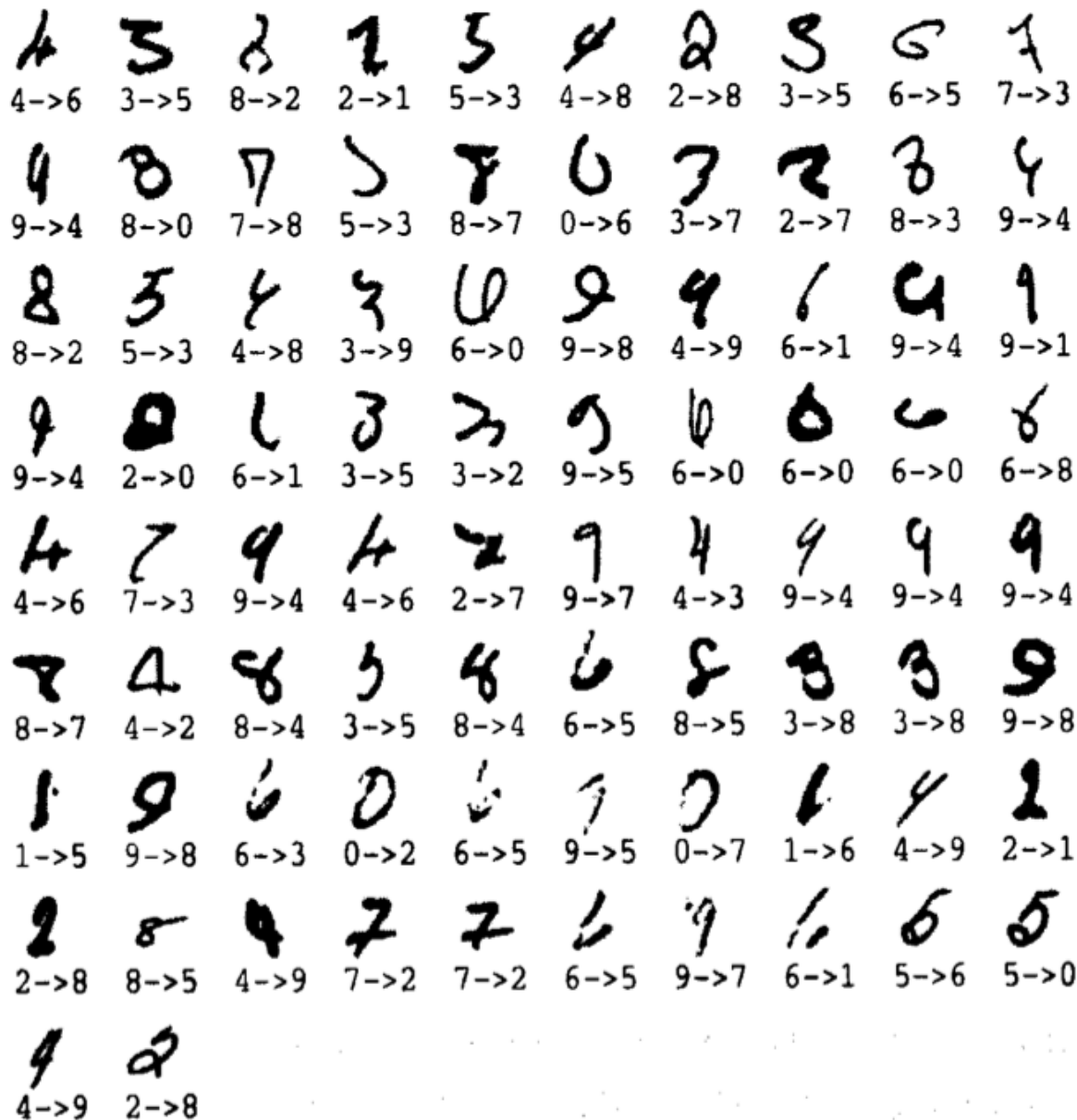
CNN Multi-camadas

- Convolutional + Non-Linear Layer
- Sub-sampling Layer
- Convolutional + Non-Linear Layer
- Fully connected layers
- Supervised



Como Tudo Começou? LeNet5



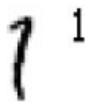
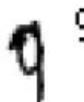
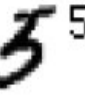
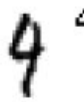
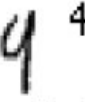
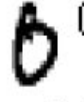
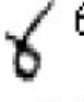
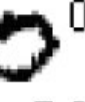
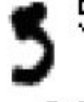
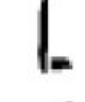
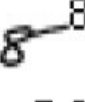
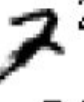
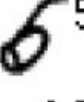


Os 82 erros da LeNet5

Muitos casos facilmente reconhecidos por humanos

Taxa de erro humana estimada em 20 a 30 erros

Erros na Rede de Ciresan *et.al.*

 2 17	 1 71	 8 98	 9 59	 9 79	 5 35	 8 23
 9 49	 5 35	 4 97	 9 49	 4 94	 2 02	 5 35
 6 16	 4 94	 0 60	 6 06	 6 86	 1 79	 1 71
 9 49	 0 50	 5 35	 8 98	 9 79	 7 17	 1 61
 7 27	 8 58	 2 78	 6 16	 5 65	 4 94	 0 60

Ígito em cima: resposta certa. Dígitos de baixo: 2 melhores apostas da rede.

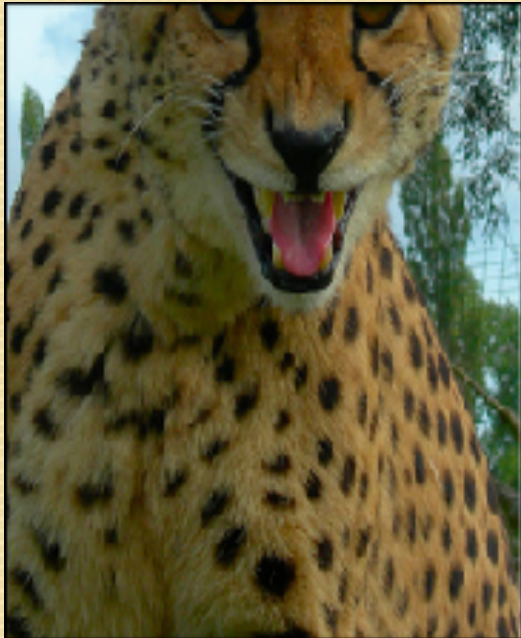
Resposta certa quase sempre nas 2 melhores apostas.

Tirando a média, performance aumentou para 25 erros.

Competição ILSVRC-2012 no Dataset ImageNet

- 1.2 milhões de imagens em alta resolução.
- **Tarefa de classificação:**
 - Encontre a classe correta nas suas 5 melhores apostas. 1000 classes.
- **Tarefa de localização:**
 - Para cada aposta, desenhe uma caixa entorno do objeto. Sua caixa tem de ter pelo menos 50% de sobreposição com a caixa correta.
- Alguns dos melhores métodos de visão computacional (VC) foram usados: Oxford, INRIA, XRCE, Tokio
 - Sistemas de VC com tarefas complicadas de múltiplos estágios
 - Estágios iniciais envolvem refinamento a mão de parâmetros

Exemplos de Apostas



cheetah

cheetah

leopard

snow leopard

Egyptian cat



bullet train is like a plane, with in-train magazine and a jet
hat that you can plug your headphones into and listen to

bullet train

bullet train

passenger car

subway train

electric locomotive



hand glass

scissors

hand glass

frying pan

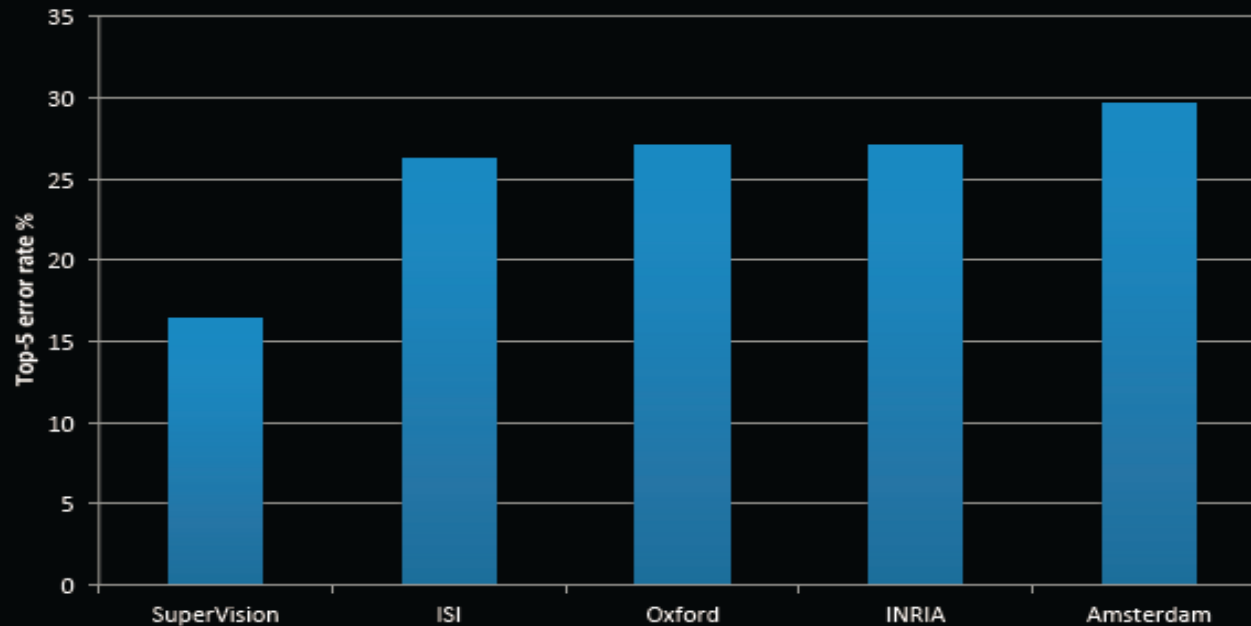
stethoscope

Taxas de Erros: ILSVRC-2012

	classification	classification & localization
• University of Tokyo	• 26.1%	53.6%
• Oxford University Computer Vision Group	• 26.9%	50.0%
• INRIA (French national research institute in CS) + XRCE (Xerox Research Center Europe)	• 27.0%	
• University of Amsterdam	• 29.5%	
• University of Toronto (Alex Krizhevsky)	• 16.4%	34.1%

Classificações na Imagenet 2012

- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) – 26.2% error



Rede Neural CNN para ImageNet

- Alex Krizhevsky (NIPS 2012): rede muito profunda baseada no trabalho de Yann Le Cun
 - 7 camadas intermediárias sem contar camadas de max pooling
 - Primeiras camadas convolucionais
 - Últimas 2 totalmente conectadas

Alguns Exemplos de Classificação: ILSVRC

			
lens cap	abacus	slug	hen
reflex camera Polaroid camera pencil sharpener switch combination lock	abacus typewriter keyboard space bar computer keyboard accordion	slug zucchini ground beetle common newt water snake	hen cock cocker spaniel partridge English setter
			
tiger	chambered nautilus	tape player	planetarium
tiger tiger cat tabby boxer Saint Bernard	lampshade throne goblet table lamp hamper	cellular telephone slot reflex camera dial telephone iPod	planetarium dome mosque radio telescope steel arch bridge

Outros



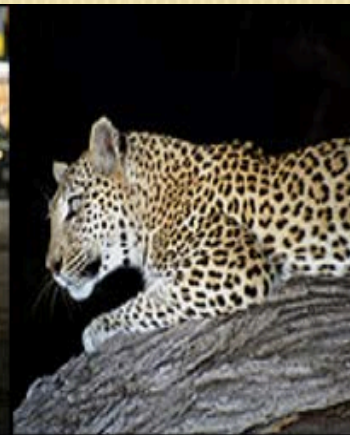
mite



container ship



motor scooter



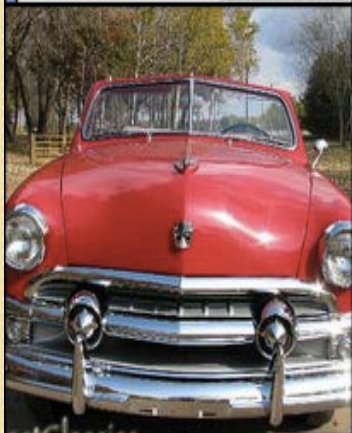
leopard

	mite
	black widow
	cockroach
	tick
	starfish

	container ship
	lifeboat
	amphibian
	fireboat
	drilling platform

	motor scooter
	go-kart
	moped
	bumper car
	golfcart

	leopard
	jaguar
	cheetah
	snow leopard
	Egyptian cat



grille



mushroom



cherry



Madagascar cat

	convertible
	grille
	pickup
	beach wagon
	fire engine

	agaric
	mushroom
	jelly fungus
	gill fungus
	dead-man's-fingers

	dalmatian
	grape
	elderberry
	ffordshire bullterrier
	currant

	squirrel monkey
	spider monkey
	titi
	indri
	howler monkey

Topologia da CNN de Krizhevsky

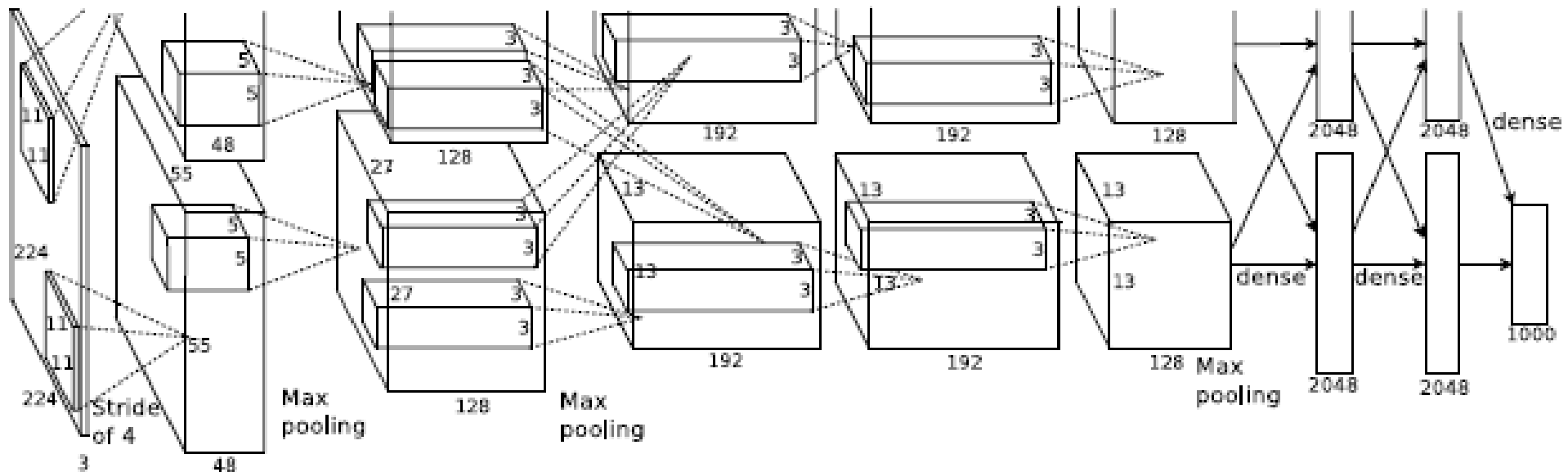
- 5 convolutional layers
- 3 fully connected layers + soft-max
- 650K neurons , 60 Mln weights

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



Explosão

The image is a screenshot of the MIT Technology Review website, specifically the '10 Breakthrough Technologies 2013' article. The page has a dark header with navigation links: HOME, MENU, CONNECT, THE LATEST, POPULAR, MOST SHARED, and a user icon. Below the header, the article title '10 BREAKTHROUGH TECHNOLOGIES 2013' is prominently displayed, along with a sub-header 'MIT Technology Review'. To the right of the title are three tabs: 'Introduction', 'The 10 Technologies', and 'Past Years'. The main content area features a grid of ten technology cards. The first card, 'Deep Learning', is highlighted with a red rounded rectangle. It describes how massive computational power enables machines to recognize objects and translate speech in real time. Other cards include 'Temporary Social Media', 'Prenatal DNA Sequencing', 'Additive Manufacturing', 'Baxter: The Blue-Collar Robot', 'Memory Implants', 'Smart Watches', 'Ultra-Efficient Solar Power', 'Big Data from Cheap Phones', and 'Supergrids'. Each card contains a brief description and a right-pointing arrow at the bottom right corner.

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart. →

Temporary Social Media

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous. →

Prenatal DNA Sequencing

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child? →

Additive Manufacturing

Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts. →

Baxter: The Blue-Collar Robot

Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people. →

Memory Implants

Smart Watches

Ultra-Efficient Solar Power

Big Data from Cheap Phones

Supergrids

MIT Technology Review, April 23rd, 2013

ILSVRC 2012: Top Rankers

<http://www.image-net.org/challenges/LSVRC/2012/results.html>

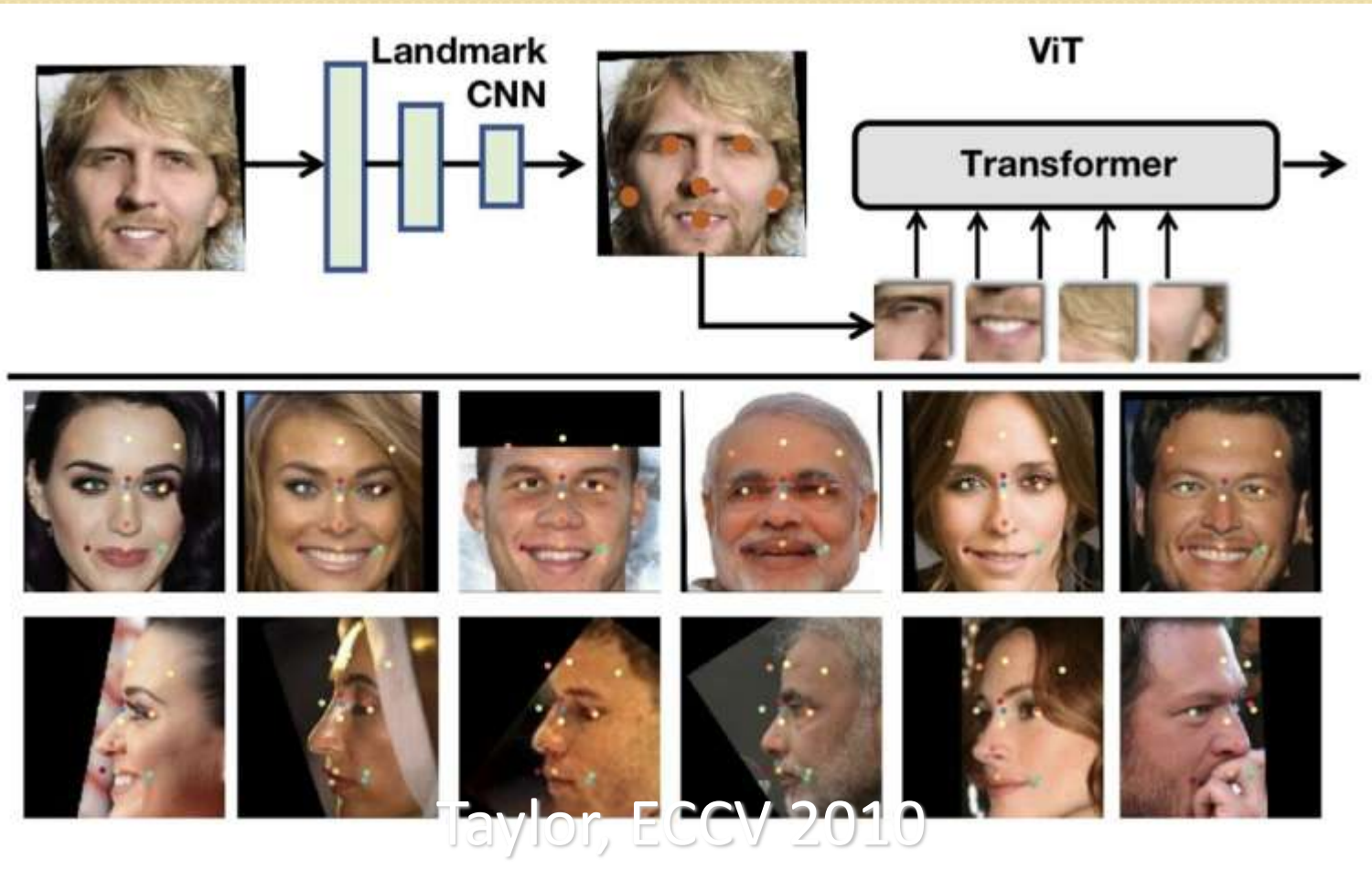
N	Error-5	Algorithm	Team	Authors
1	0.153	Deep Conv. Neural Network	Univ. of Toronto	Krizhevsky et al
2	0.262	Features + Fisher Vectors + Linear classifier	ISI	Gunji et al
3	0.270	Features + FV + SVM	OXFORD_VGG	Simonyan et al
4	0.271	SIFT + FV + PQ + SVM	XRCE/INRIA	Perronin et al
5	0.300	Color desc. + SVM	Univ. of Amsterdam	van de Sande et al

Imagenet 2013: Top Rankers

<http://www.image-net.org/challenges/LSVRC/2013/results.php>

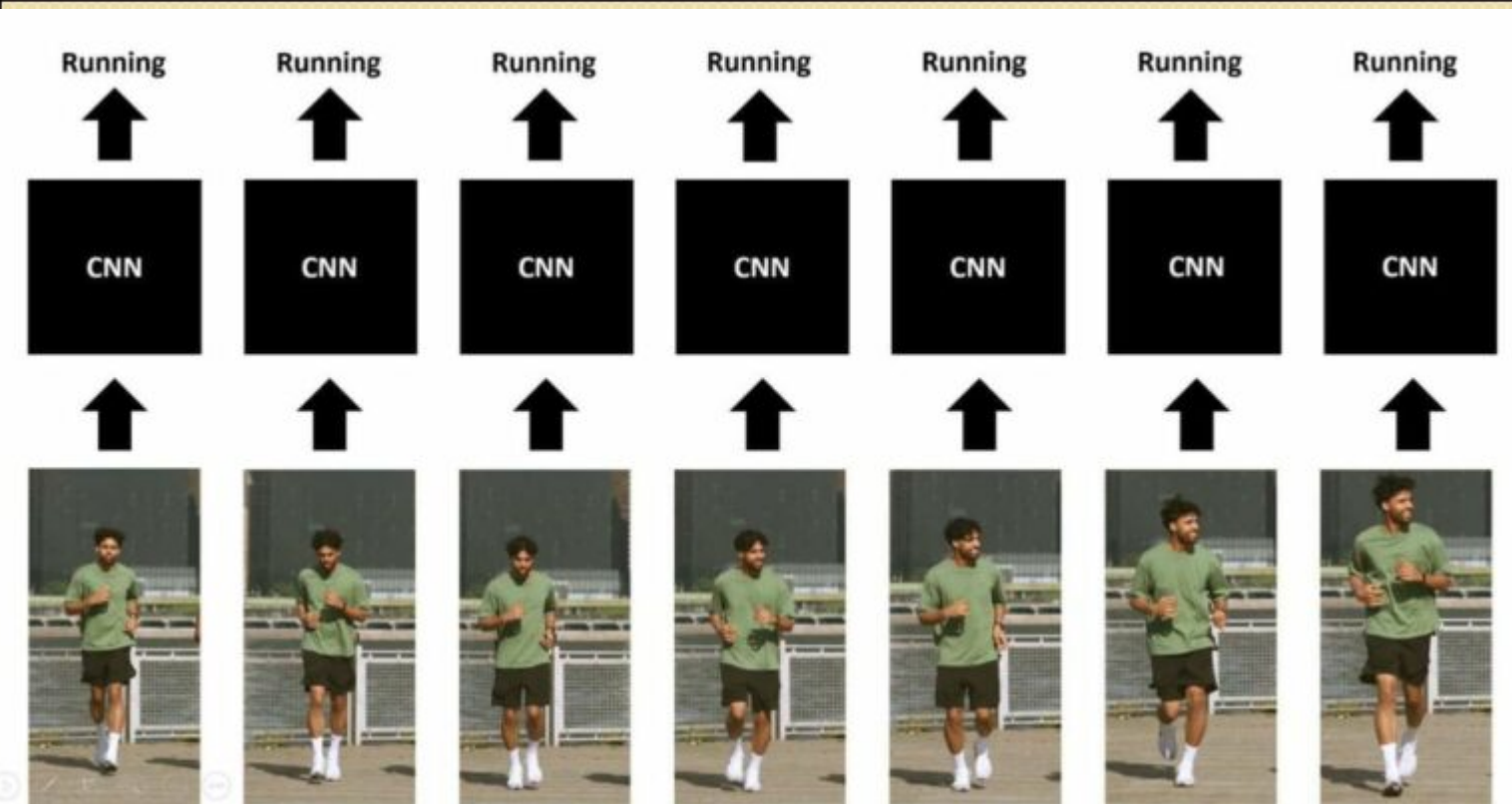
N	Error-5	Algorithm	Team	Authors
1	0.117	Deep Convolutional Neural Network	Clarifi	Zeiler
2	0.129	Deep Convolutional Neural Networks	Nat.Univ Singapore	Min LIN
3	0.135	Deep Convolutional Neural Networks	NYU	Zeiler Fergus
4	0.135	Deep Convolutional Neural Networks		Andrew Howard
5	0.137	Deep Convolutional Neural Networks	Overfeat NYU	Pierre Sermanet et al

CNN: Faces

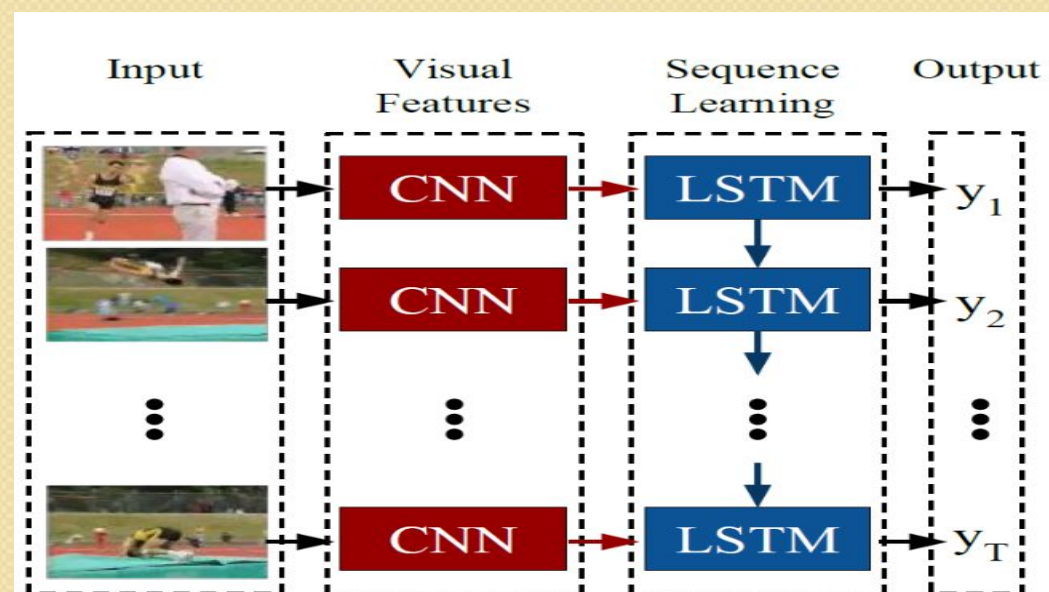


CNN: Carros Autônomos





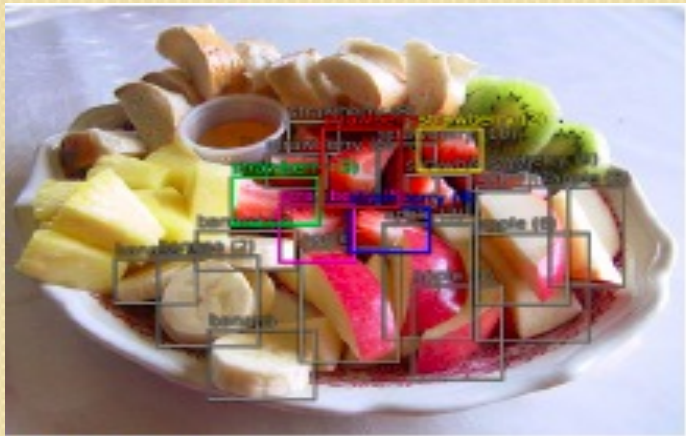
CNN: Video Monitoring



CNN: Detecção de Ações



CNN: Detecção



Groundtruth:

strawberry
strawberry (2)
strawberry (3)
strawberry (4)
strawberry (5)
strawberry (6)
strawberry (7)
strawberry (8)
strawberry (9)
strawberry (10)
apple
apple (2)
apple (3)

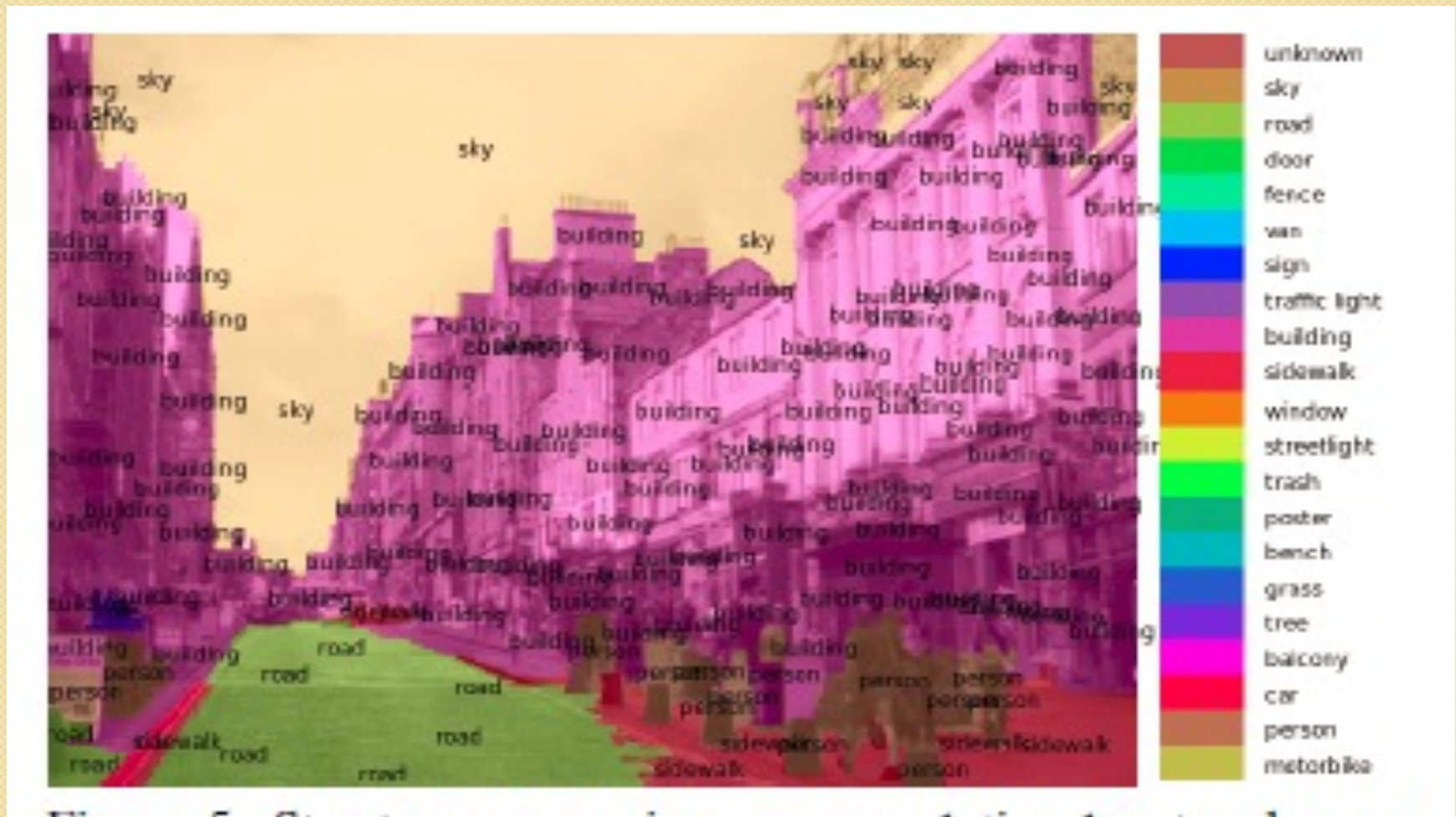


Groundtruth:

tv or monitor
tv or monitor (2)
tv or monitor (3)
person
remote control
remote control (2)

Sermanet, CVPR 2014

CNN: Descrição de Cenários



Farabet, PAMI 2013

CNN: Rotulamento Semântico Indoor em RGBD

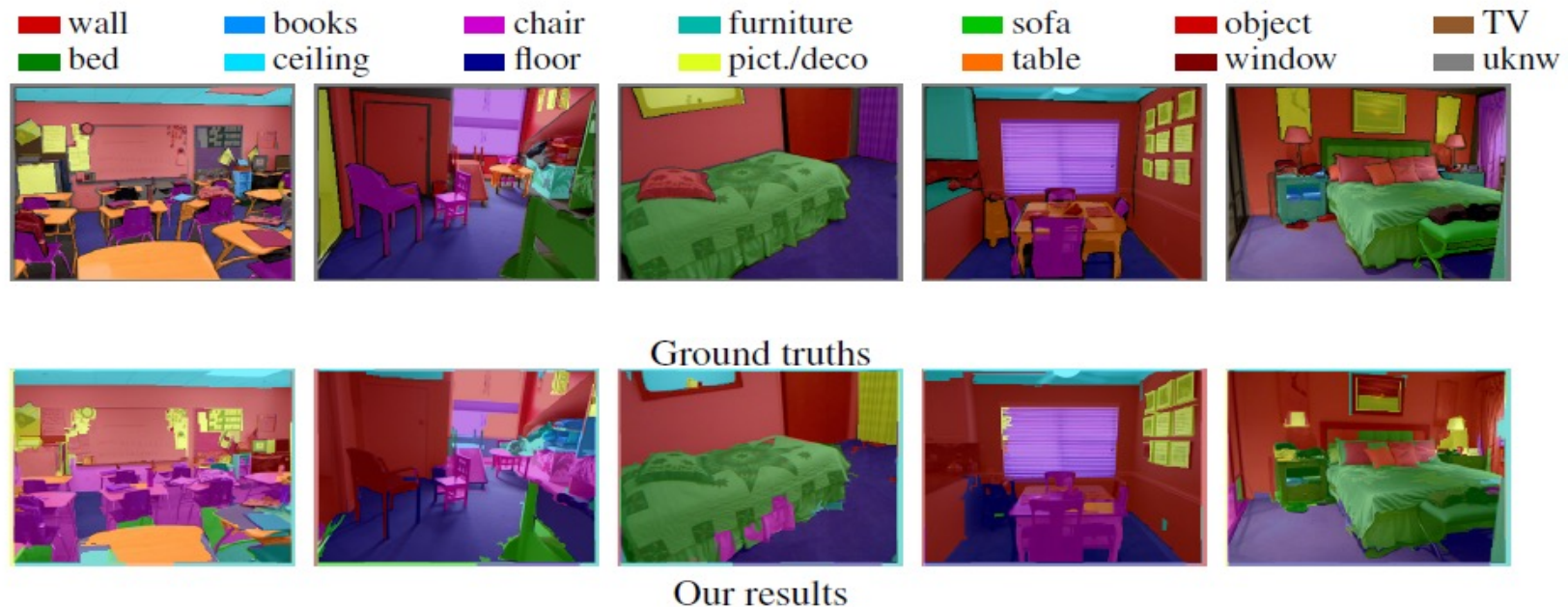


Figure 2: Some scene labelings using our Multiscale Convolutional Network trained on RGBD images.

38,1 °C

36,1 °C

37,9 °C

36,5 °C



ANOMALIA - DIAGNÓSTICO

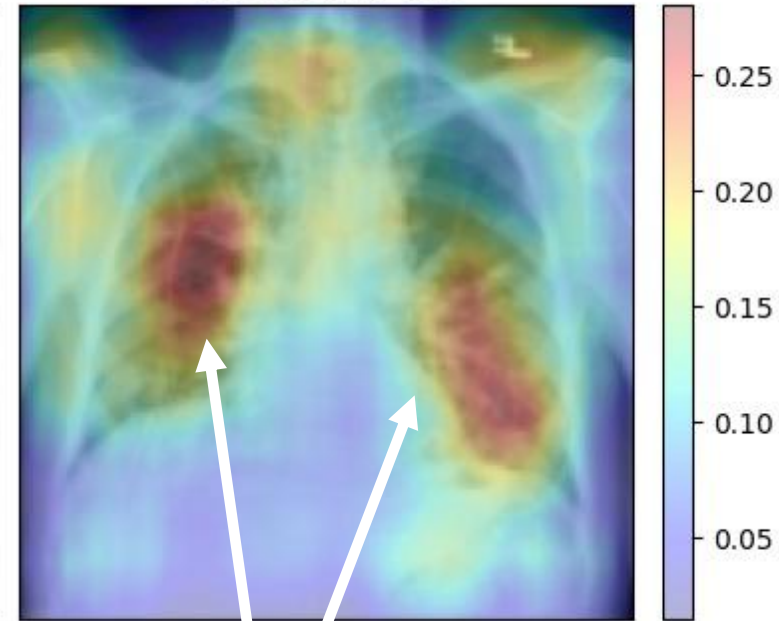
Input Image



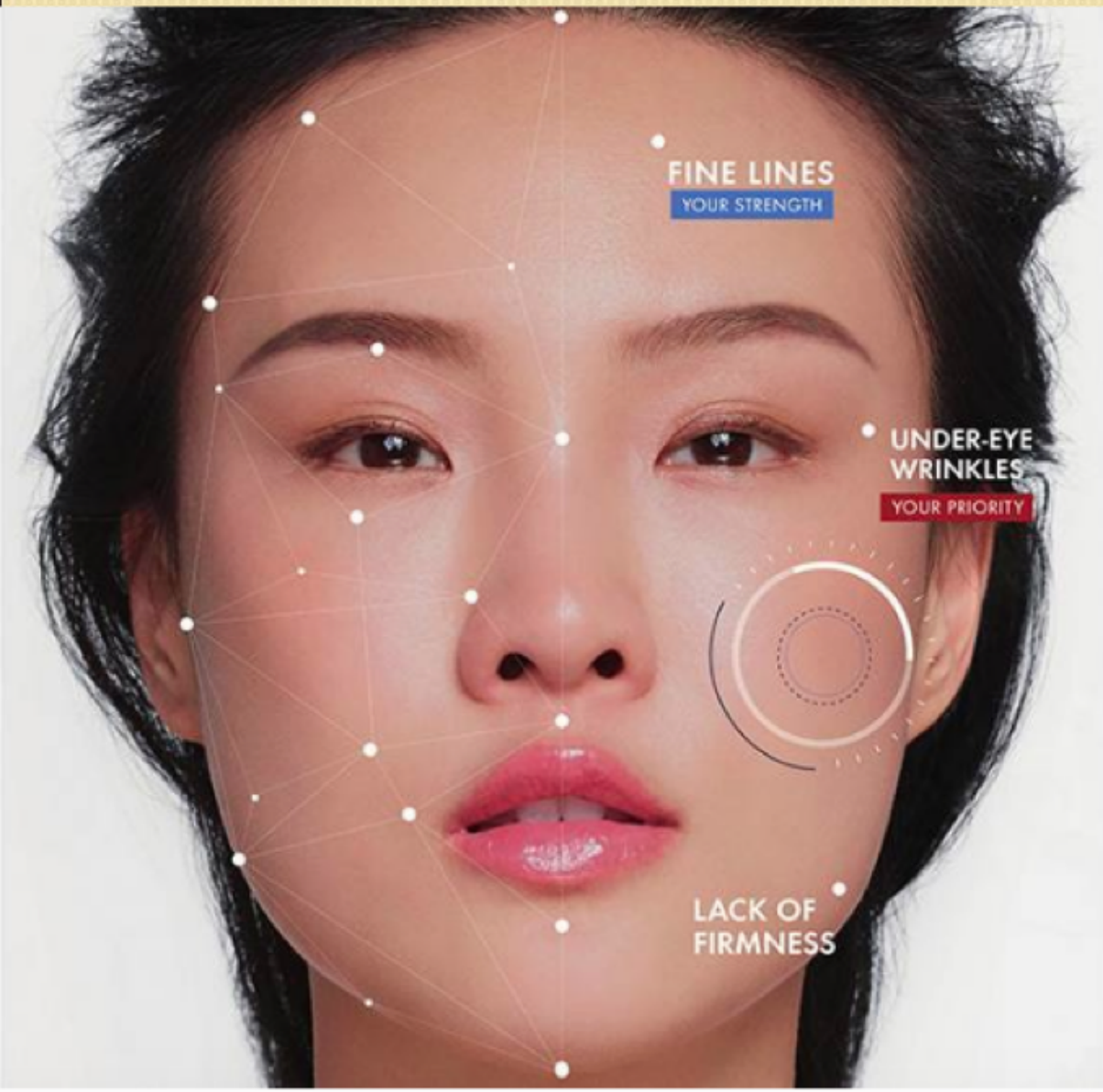
Reconstructed Image



Anomaly Map



PNEUMONIA



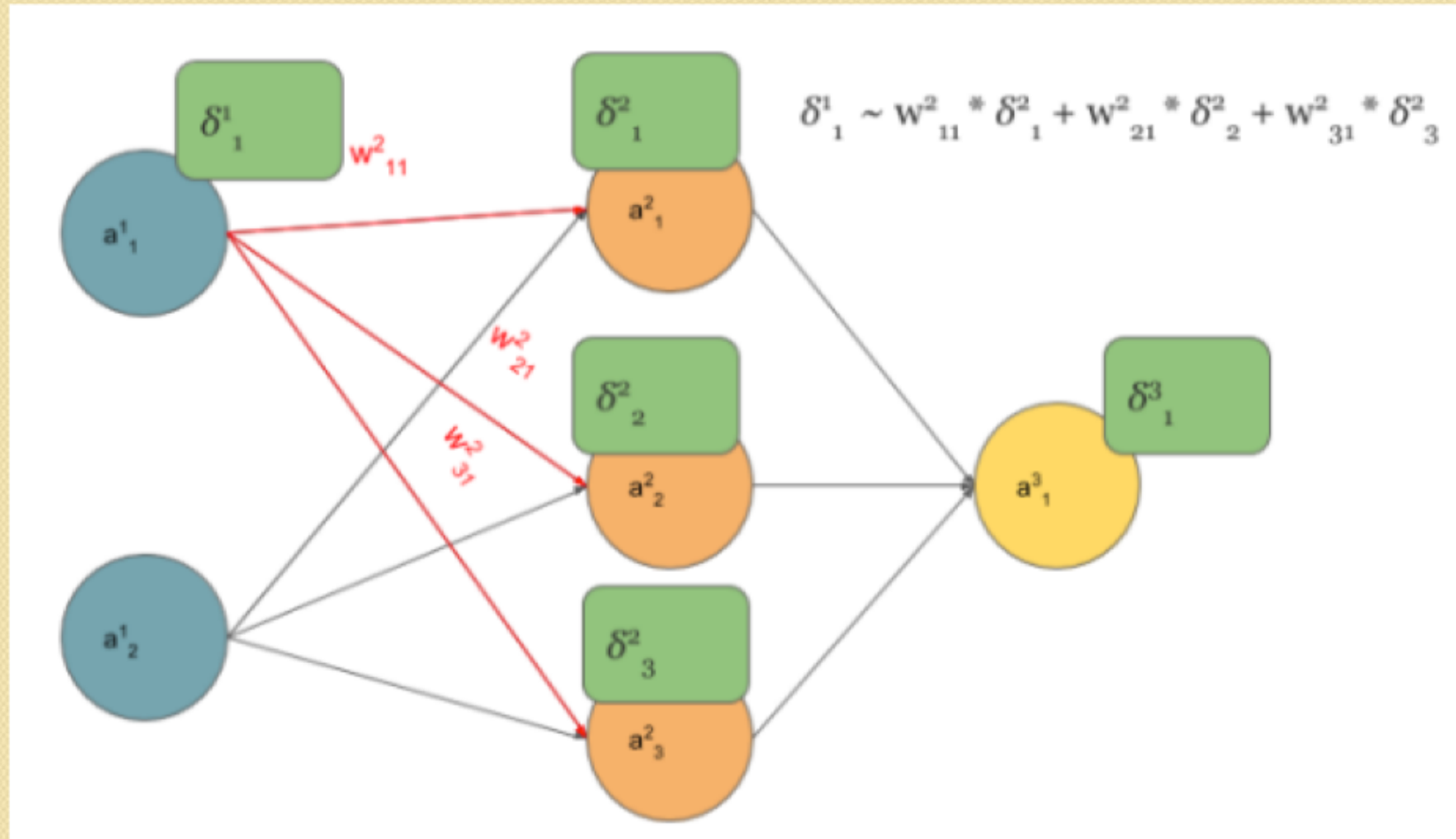
Empresa lider em
Cosméticos e IA

<https://www.youtube.com/watch?v=kfRGxkIIF5M>

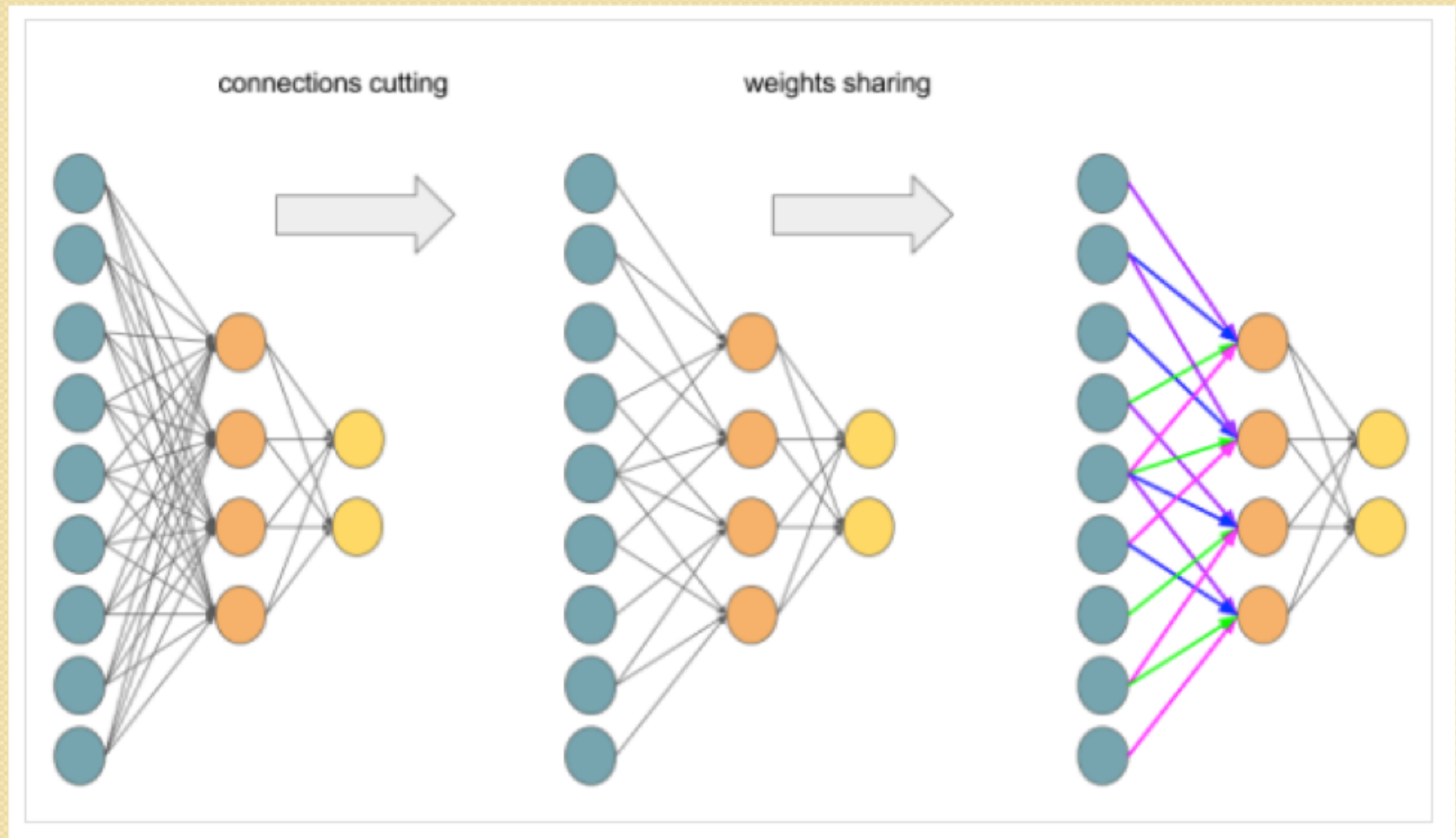


Como se Treina uma CNN?

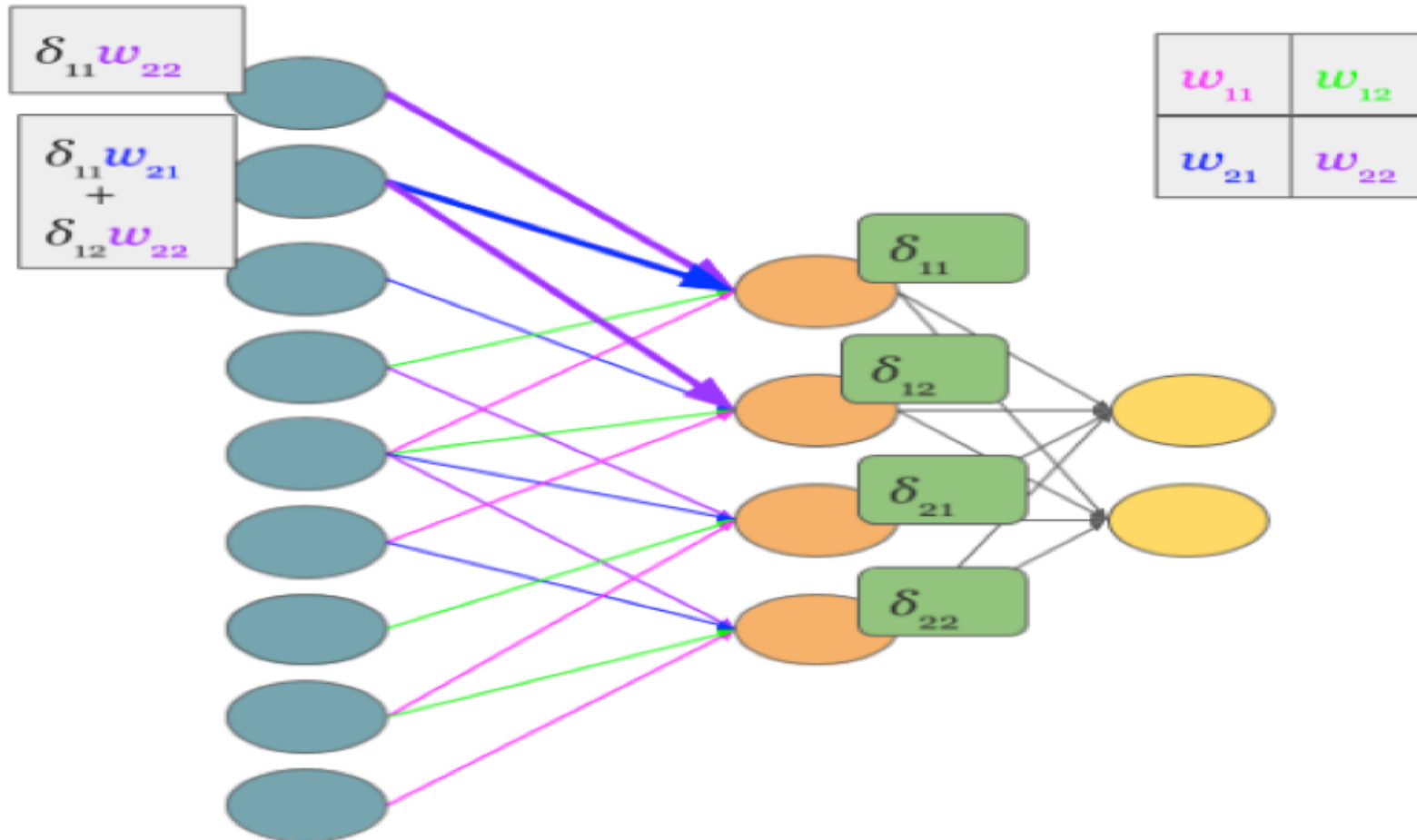
Lembrando Gradiente Descendente (Backpropagation)



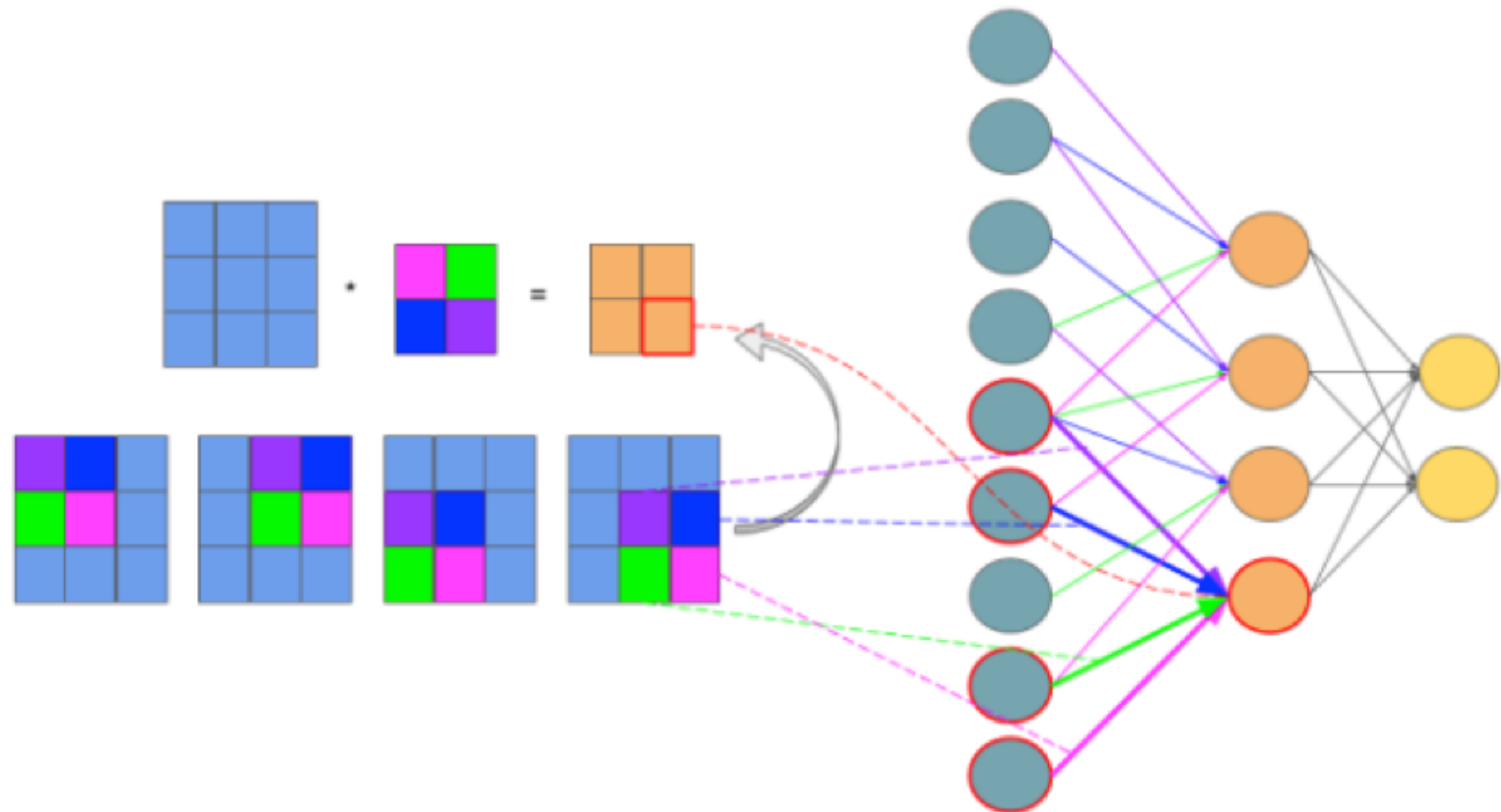
Transformando Rede Totalmente Conectada em CNN



Transformando Rede Totalmente Conectada em CNN



Operação de Convolução



Definição do Erro para Ajuste de Cada Conexão: Forward

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

W_{11}	W_{12}
W_{21}	W_{22}

Definição do Erro para Ajuste de Cada Conexão: Forward

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

W_{11}	W_{12}
W_{21}	W_{22}

h_{11}	h_{12}
h_{21}	h_{22}

Input Size : 3x3, Filter Size : 2x2, Output Size : 2x2

$$h_{11} = W_{11}X_{11} + W_{12}X_{12} + W_{21}X_{21} + W_{22}X_{22}$$

$$h_{12} = W_{11}X_{12} + W_{12}X_{13} + W_{21}X_{22} + W_{22}X_{23}$$

$$h_{21} = W_{11}X_{21} + W_{12}X_{22} + W_{21}X_{31} + W_{22}X_{32}$$

$$h_{22} = W_{11}X_{22} + W_{12}X_{23} + W_{21}X_{32} + W_{22}X_{33}$$

Relembrando: Regra Delta Generalizada ou Error-Back Propagation


$$\Delta_p W_{ji} \propto \delta_j \cdot O_{pi}$$


$$\Delta_p W_{ji} = \eta \cdot \delta_j \cdot O_{pi}$$

Definição do Erro para Ajuste de Cada Conexão:Backward

δ₁₁

$$\partial W_{11} = X_{11} \partial h_{11} + X_{12} \partial h_{12} + X_{21} \partial h_{21} + X_{22} \partial h_{22}$$

δ₁₂

$$\partial W_{12} = X_{12} \partial h_{11} + X_{13} \partial h_{12} + X_{22} \partial h_{21} + X_{23} \partial h_{22}$$

δ₂₁

$$\partial W_{21} = X_{21} \partial h_{11} + X_{22} \partial h_{12} + X_{31} \partial h_{21} + X_{32} \partial h_{22}$$

δ₂₂

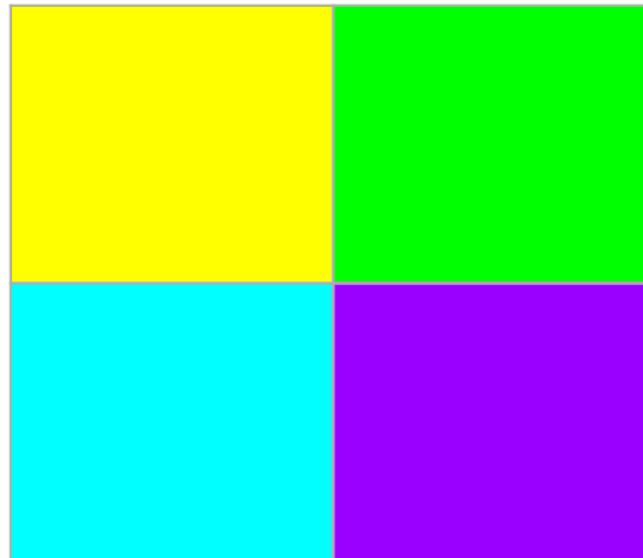
$$\partial W_{22} = X_{22} \partial h_{11} + X_{23} \partial h_{12} + X_{32} \partial h_{21} + X_{33} \partial h_{22}$$

Final derivatives after performing back propagation

$$\partial h_{ij} \text{ represents } \frac{\partial L}{\partial h_{ij}}$$

$$\partial w_{ij} \text{ represents } \frac{\partial L}{\partial w_{ij}}$$

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}



∂h_{11}	∂h_{12}
∂h_{21}	∂h_{22}

Definição do Erro para Unidades da Camada Anterior

w_{22}	w_{21}
w_{12}	w_{11}

rot_180(w)

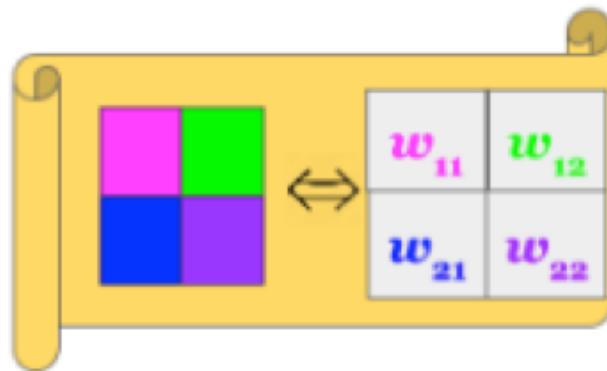
*

δ_{11}	δ_{12}
δ_{21}	δ_{22}

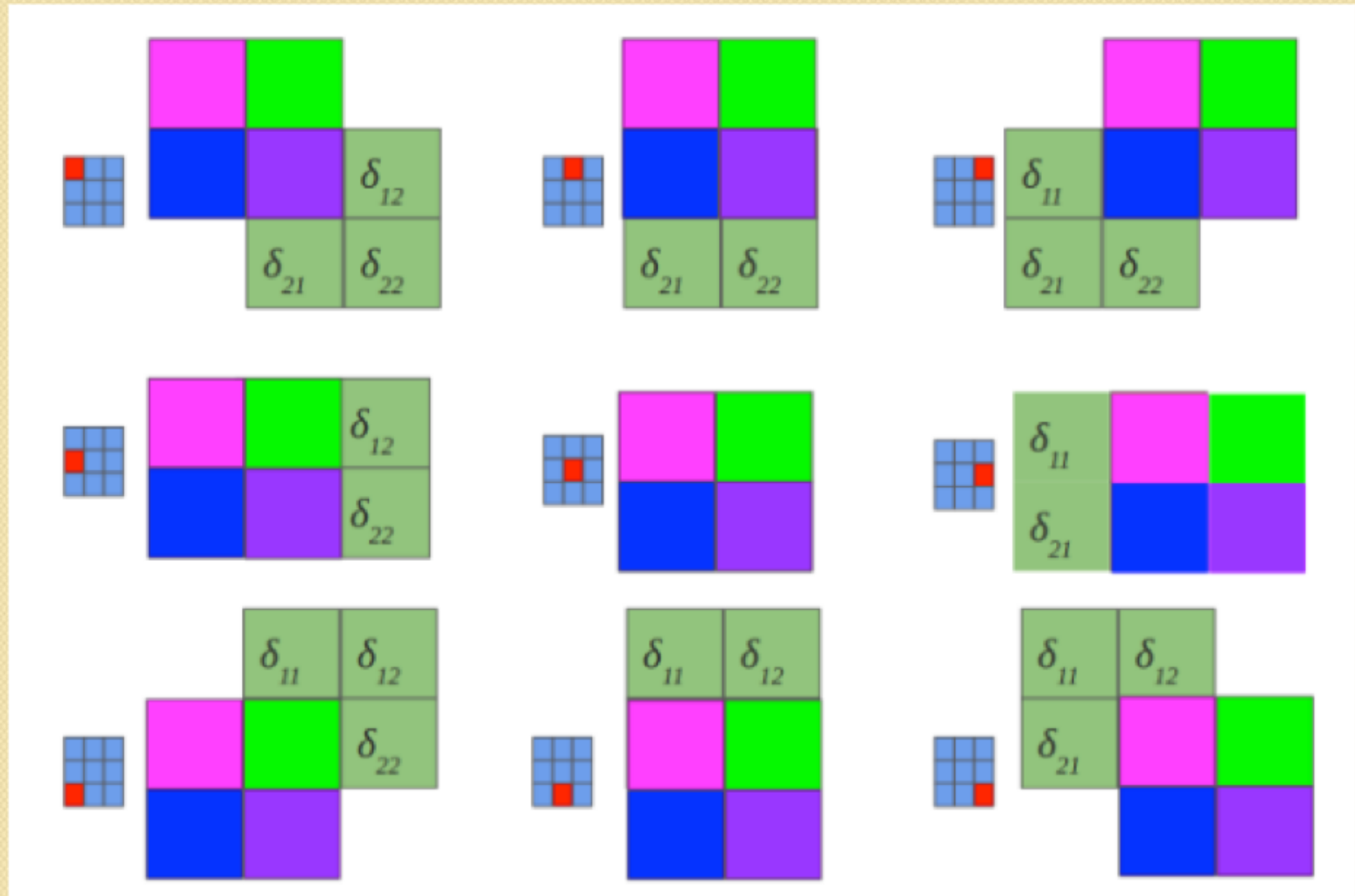
grads from orange layer

=

$\delta_{11} w_{22}$	$\delta_{11} w_{21} + \delta_{12} w_{22}$	$\delta_{12} w_{21}$
$\delta_{11} w_{12} + \delta_{21} w_{22}$	$\delta_{11} w_{11} + \delta_{12} w_{12} + \delta_{21} w_{21} + \delta_{22} w_{22}$	$\delta_{12} w_{11} + \delta_{22} w_{21}$
$\delta_{21} w_{12}$	$\delta_{21} w_{11} + \delta_{22} w_{12}$	$\delta_{22} w_{11}$



Definição do Erro para Unidades da Camada Anterior



Definição do Erro para Unidades da Camada

In the standard MLP, we can define an error of neuron j as:

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

where z_j^l is just:

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l$$

and for clarity, $a_j^l = \sigma(z_j^l)$, where σ is an activation function such as sigmoid, hyperbolic tangent or relu ([https://en.wikipedia.org/wiki/Rectifier_\(neural_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))).

But here, we do not have MLP but CNN and matrix multiplications are replaced by convolutions as we discussed before. So instead of z_j we do have a $z_{x,y}$:

$$z_{x,y}^{l+1} = w^{l+1} * \sigma(z_{x,y}^l) + b_{x,y}^{l+1} = \sum_a \sum_b w_{a,b}^{l+1} \sigma(z_{x-a,y-b}^l) + b_{x,y}^{l+1}$$

Above equation is just a convolution operation during feedforward phase illustrated in the above picture titled 'Feedforward in CNN is identical with convolution operation'

Now we can get to the point and answer the question Hello, when computing the gradients CNN, the weights need to be rotated, Why ? (<https://plus.google.com/111541909707081118542/posts/P8bZBNpg84Z>)

We start from statement:

$$\delta_{x,y}^l = \frac{\partial C}{\partial z_{x,y}^l} = \sum_{x'} \sum_{y'} \frac{\partial C}{\partial z_{x',y'}^{l+1}} \frac{\partial z_{x',y'}^{l+1}}{\partial z_{x,y}^l}$$

We know that $z_{x,y}^l$ is in relation to $z_{x',y'}^{l+1}$ which is indirectly showed in the above picture titled 'Backpropagation also results with convolution'. So sums are the result of chain rule. Let's move on:

Definição do Erro para Unidades da Camada Anterior

$$\frac{\partial C}{\partial z_{x,y}^l} = \sum_{x'} \sum_{y'} \frac{\partial C}{\partial z_{x',y'}^{l+1}} \frac{\partial z_{x',y'}^{l+1}}{\partial z_{x,y}^l} = \sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} \frac{\partial (\sum_a \sum_b w_{a,b}^{l+1} \sigma(z_{x'-a,y'-b}^l) + b_{x',y'}^{l+1})}{\partial z_{x,y}^l}$$

First term is replaced by definition of error, while second has become large because we put it here expression on $z_{x',y'}^{l+1}$. However, we do not have to fear of this big monster – all components of sums equal 0, except these ones that are indexed: $x = x' - a$ and $y = y' - b$. So:

$$\sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} \frac{\partial (\sum_a \sum_b w_{a,b}^{l+1} \sigma(z_{x'-a,y'-b}^l) + b_{x',y'}^{l+1})}{\partial z_{x,y}^l} = \sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{a,b}^{l+1} \sigma'(z_{x,y}^l)$$

If $x = x' - a$ and $y = y' - b$ then it is obvious that $a = x' - x$ and $b = y' - y$ so we can reformulate above equation to:

$$\sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{a,b}^{l+1} \sigma'(z_{x,y}^l) = \sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{x'-x,y'-y}^{l+1} \sigma'(z_{x,y}^l)$$

OK, our last equation is just ...

$$\sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{x'-x,y'-y}^{l+1} \sigma'(z_{x,y}^l) = \delta^{l+1} * w_{-x,-y}^{l+1} \sigma'(z_{x,y}^l)$$

Where is the rotation of weights? Actually $ROT180(w_{x,y}^{l+1}) = w_{-x,-y}^{l+1}$.

So the answer on question Hello, when computing the gradients CNN, the weights need to be rotated, Why ? (<https://plus.google.com/111541909707081118542/posts/P8bZBNpg84Z>) is simple: the rotation of the weights just results from derivation of delta error in Convolution Neural Network.

OK, we are really close to the end. One more ingredient of backpropagation algorithm is update of weights $\frac{\partial C}{\partial w_{a,b}^{l+1}}$:

Definição do Erro para Unidades da Camada Anterior

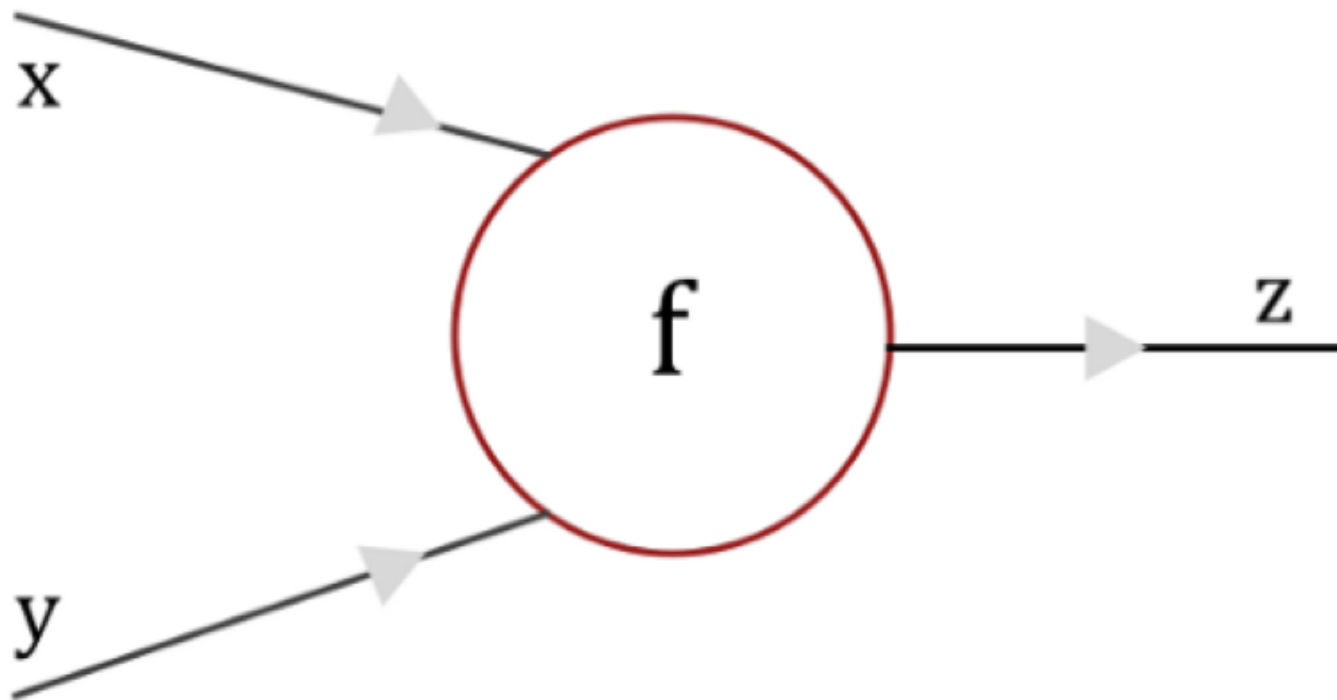
$$\frac{\partial C}{\partial w_{a,b}^l} = \sum_x \sum_y \frac{\partial C}{\partial z_{x,y}^l} \frac{\partial z_{x,y}^l}{\partial w_{a,b}^l} = \sum_x \sum_y \delta_{x,y}^l \frac{\partial (\sum_{a'} \sum_{b'} w_{a',b'}^l \sigma(z_{x-a',y-b'}^l) + b_{x,y}^l)}{\partial w_{a,b}^l} =$$

$$\sum_x \sum_y \delta_{x,y}^l \sigma(z_{x-a,y-b}^{l-1}) = \delta_{a,b}^l * \sigma(z_{-a,-b}^{l-1}) = \delta_{a,b}^l * \sigma(ROT180(z_{a,b}^{l-1}))$$

So paraphrasing the backpropagation algorithm (http://neuralnetworksanddeeplearning.com/chap2.html#the_backpropagation_algorithm) for CNN:

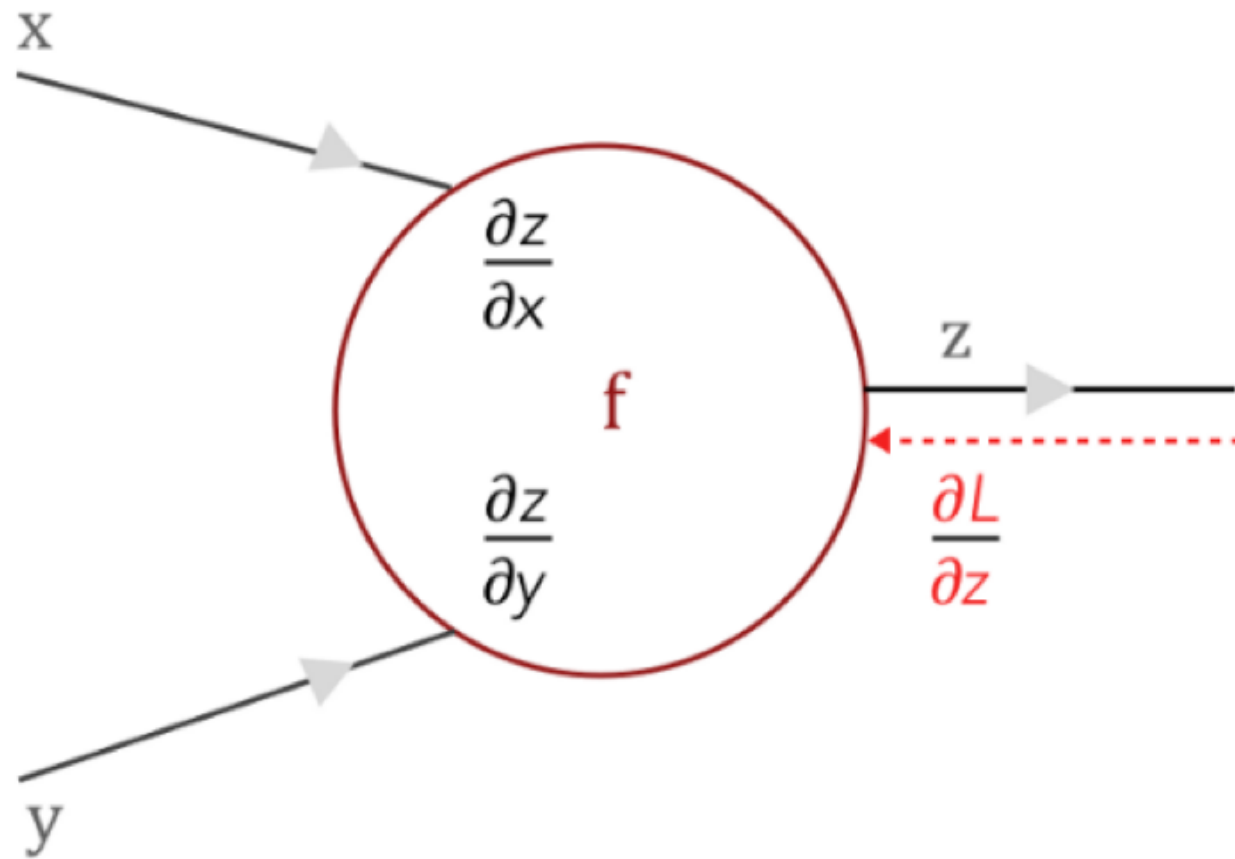
1. Input x: set the corresponding activation a^1 for the input layer.
2. Feedforward: for each $l = 2, 3, \dots, L$ compute $z_{x,y}^l = w_{x,y}^l * \sigma(z_{x,y}^{l-1}) + b_{x,y}^l$ and $a_{x,y}^l = \sigma(z_{x,y}^l)$
3. Output error δ^L : Compute the vector $\delta^L = \nabla_a C \odot \sigma'(z^L)$
4. Backpropagate the error: For each $l = L-1, L-2, \dots, 2$ compute $\delta_{x,y}^l = \delta_{x,y}^{l+1} * ROT180(w_{x,y}^{l+1}) \sigma'(z_{x,y}^l)$
5. Output: The gradient of the cost function is given by $\frac{\partial C}{\partial w_{a,b}^l} = \delta_{a,b}^l * \sigma(ROT180(z_{a,b}^{l-1}))$

The end 😊



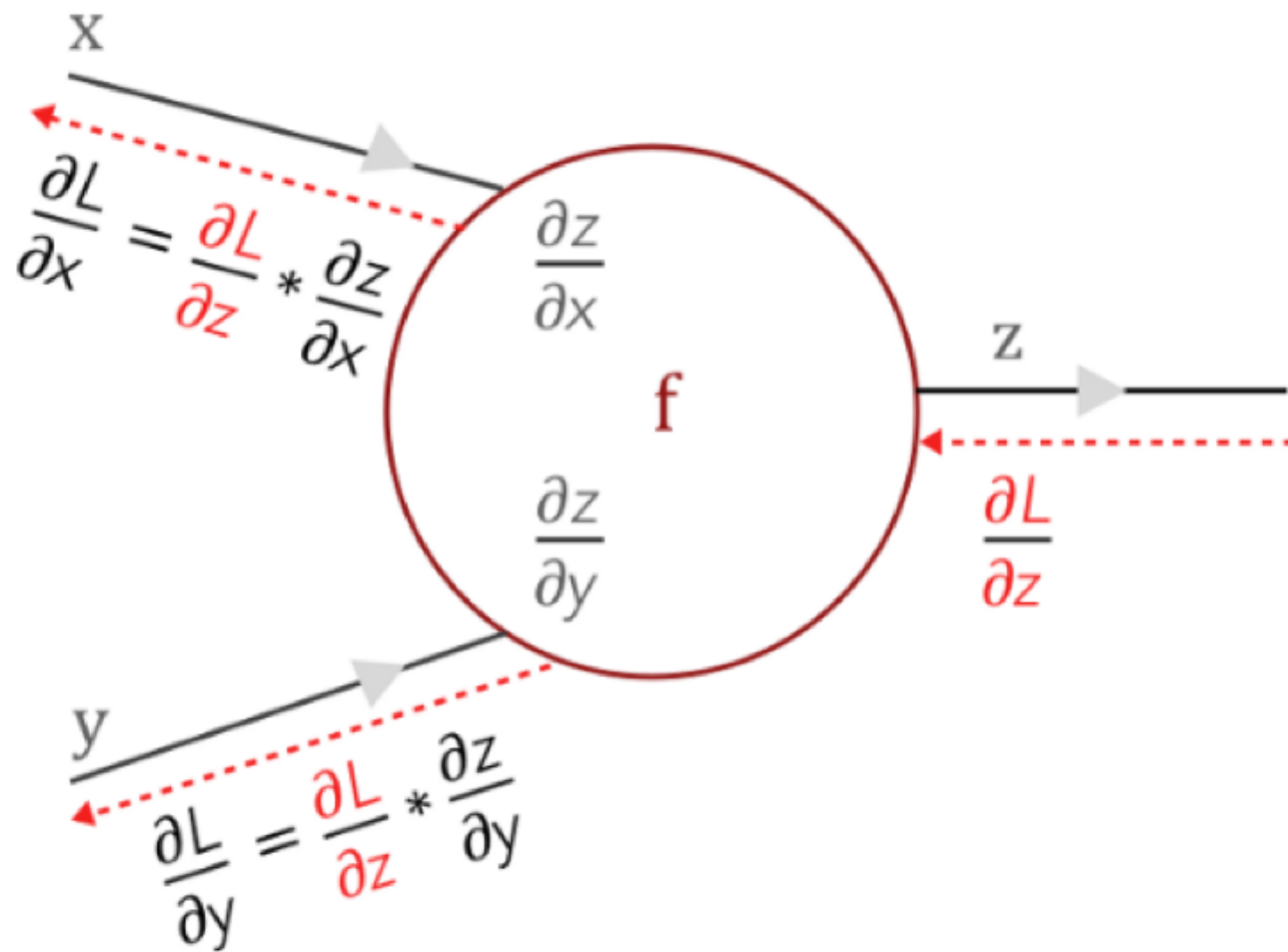
A simple function f which takes x and y as inputs and outputs z

We can easily compute the *local gradients* — differentiating z with respect to x and y as $\partial z / \partial x$ and $\partial z / \partial y$



$\frac{\partial z}{\partial x}$ & $\frac{\partial z}{\partial y}$ are local gradients

$\frac{\partial L}{\partial z}$ is the loss from the previous layer which
has to be backpropagated to other layers



$\frac{\partial z}{\partial x}$ & $\frac{\partial z}{\partial y}$ are local gradients

$\frac{\partial L}{\partial z}$ is the loss from the previous layer which has to be backpropagated to other layers

Now, let's assume the function f is a *convolution* between **Input X** and a **Filter F**. Input X is a 3x3 matrix and Filter F is a 2x2 matrix, as shown below:

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

Input **X**

F_{11}	F_{12}
F_{21}	F_{22}

Filter **F**

$$\begin{array}{|c|c|} \hline O_{11} & O_{12} \\ \hline O_{21} & O_{22} \\ \hline \end{array} = \text{Convolution} \left(\begin{array}{|c|c|c|} \hline X_{11} & X_{12} & X_{13} \\ \hline X_{21} & X_{22} & X_{23} \\ \hline X_{31} & X_{32} & X_{33} \\ \hline \end{array}, \begin{array}{|c|c|} \hline F_{11} & F_{12} \\ \hline F_{21} & F_{22} \\ \hline \end{array} \right)$$

Output **O**

Input **X**

Filter **F**

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

Input **X**

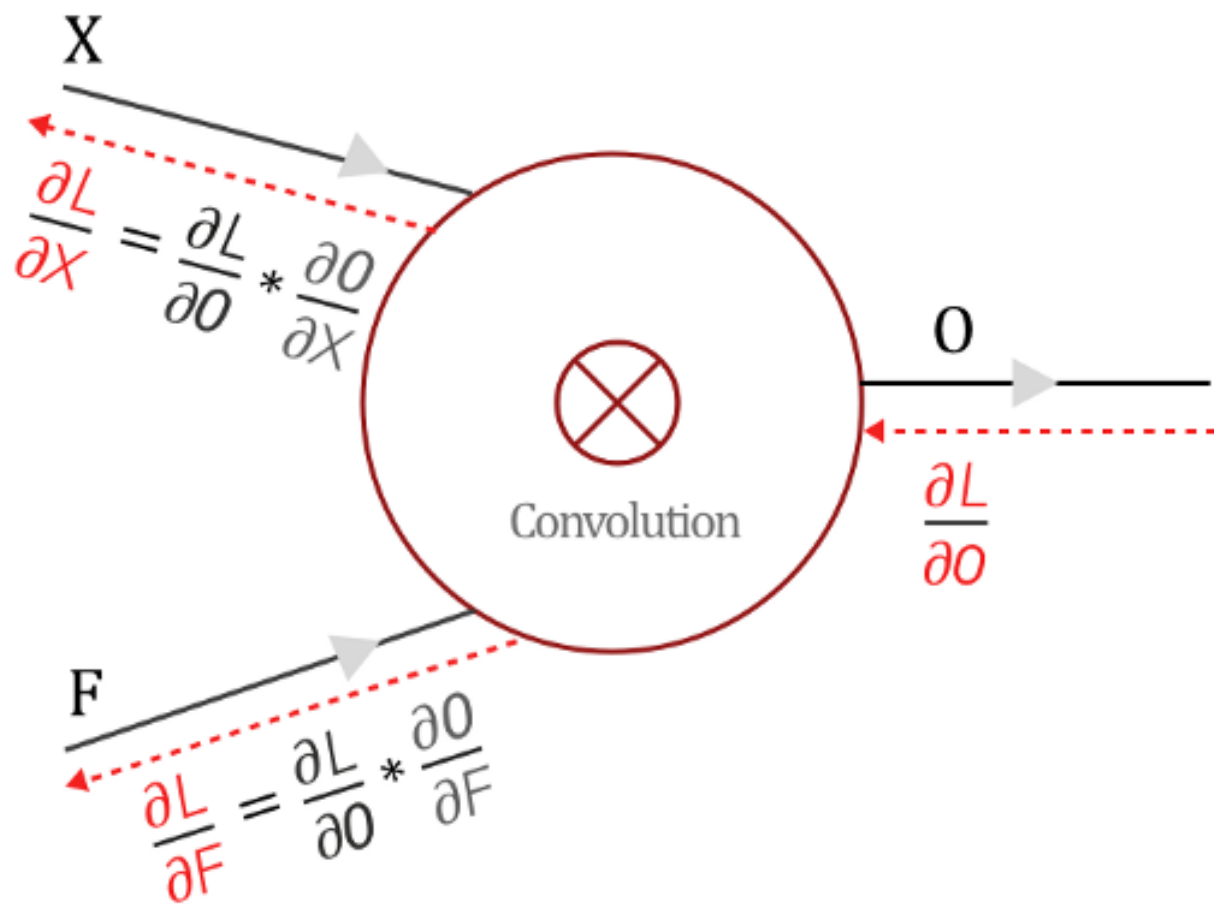


F_{11}	F_{12}
F_{21}	F_{22}

Filter **F**

$X_{11}F_{11}$	$X_{12}F_{12}$	X_{13}
$X_{21}F_{21}$	$X_{22}F_{22}$	X_{23}
X_{31}	X_{32}	X_{33}

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$



$\frac{\partial O}{\partial X}$ & $\frac{\partial O}{\partial F}$ are local gradients

$\frac{\partial L}{\partial z}$ is the loss from the previous layer which has to be backpropagated to other layers

So let's find the gradients for X and F — $\partial L / \partial X$ and $\partial L / \partial F$

Finding $\partial L / \partial F$

This has two steps as we have done earlier.

- Find the local gradient $\partial O / \partial F$
- Find $\partial L / \partial F$ using chain rule

Step 1: Finding the local gradient — $\partial O / \partial F$:

This means we have to differentiate Output Matrix O with Filter F. From our convolution operation, we know the values. So let us start differentiating the first element of O- O^{11} with respect to the elements of F — F^{11} , F^{12} , F^{21} and F^{22}

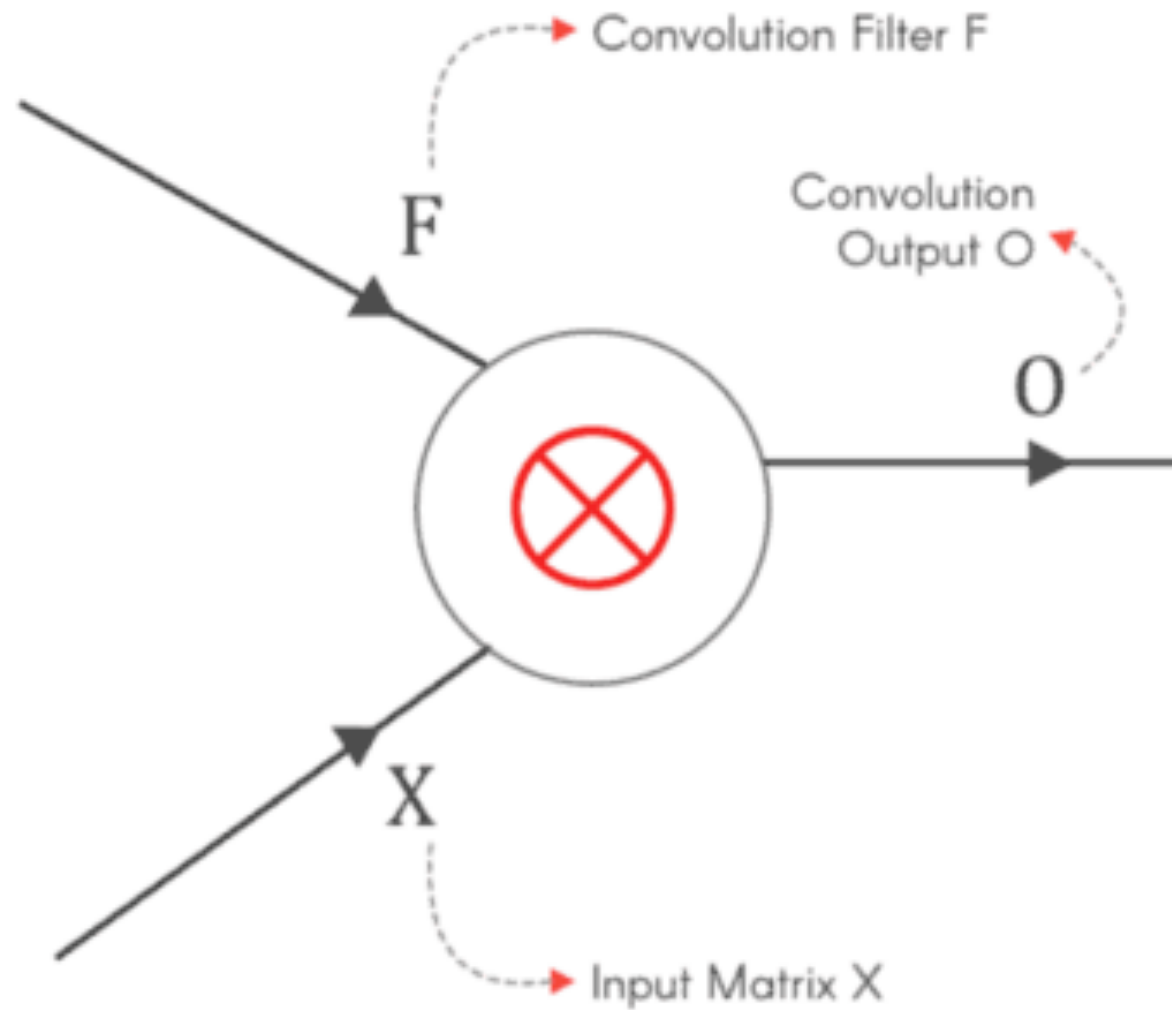
Local Gradients $\longrightarrow$ (A)

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Finding derivatives with respect to F_{11} , F_{12} , F_{21} and F_{22}

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11} \quad \frac{\partial O_{11}}{\partial F_{12}} = X_{12} \quad \frac{\partial O_{11}}{\partial F_{21}} = X_{21} \quad \frac{\partial O_{11}}{\partial F_{22}} = X_{22}$$

Similarly, we can find the local gradients for O_{12} , O_{21} and O_{22}



<https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c>

Operação de Convolução

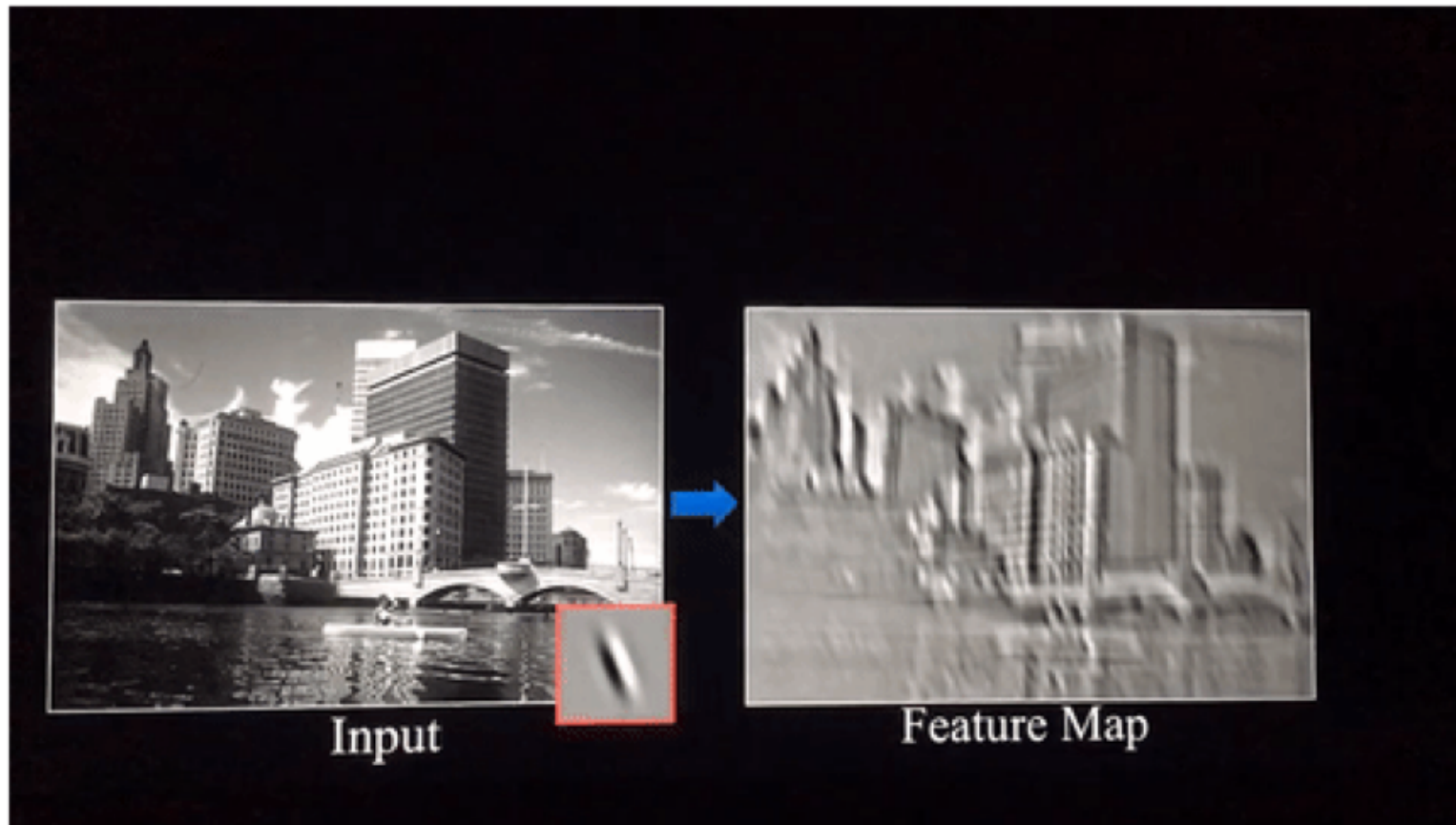
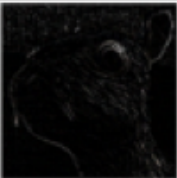
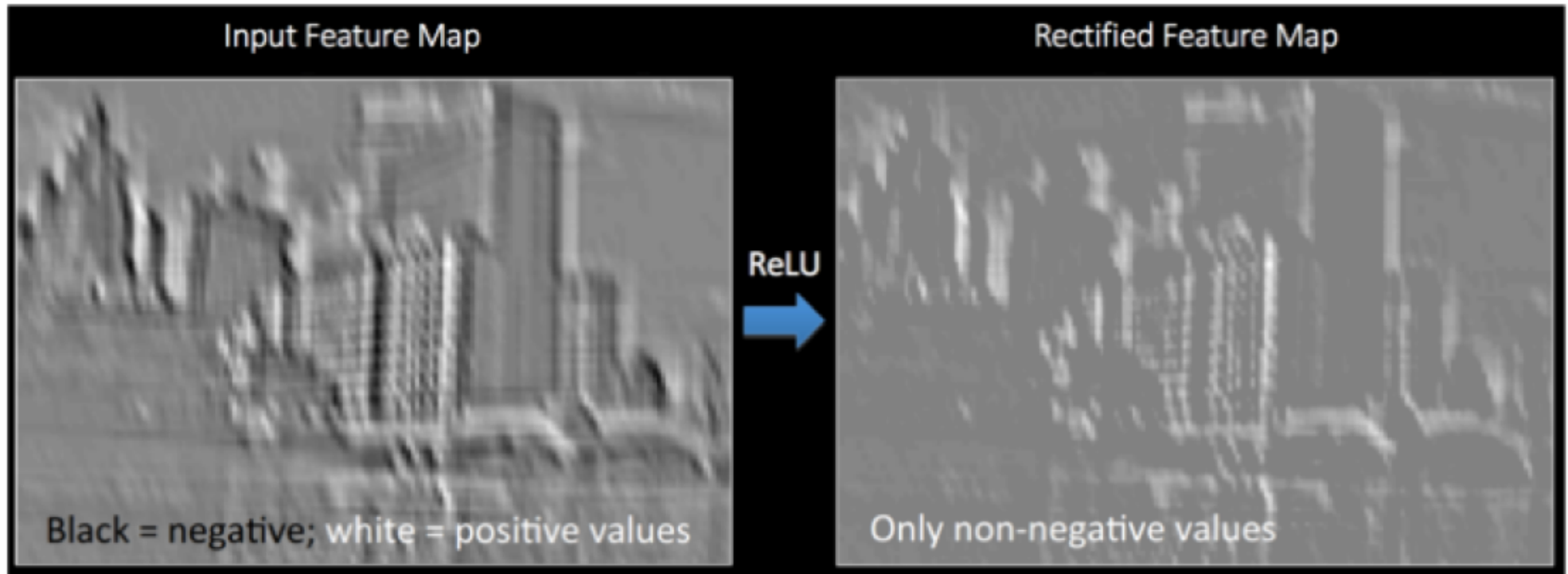


Figure 6: The Convolution Operation. Source [9]

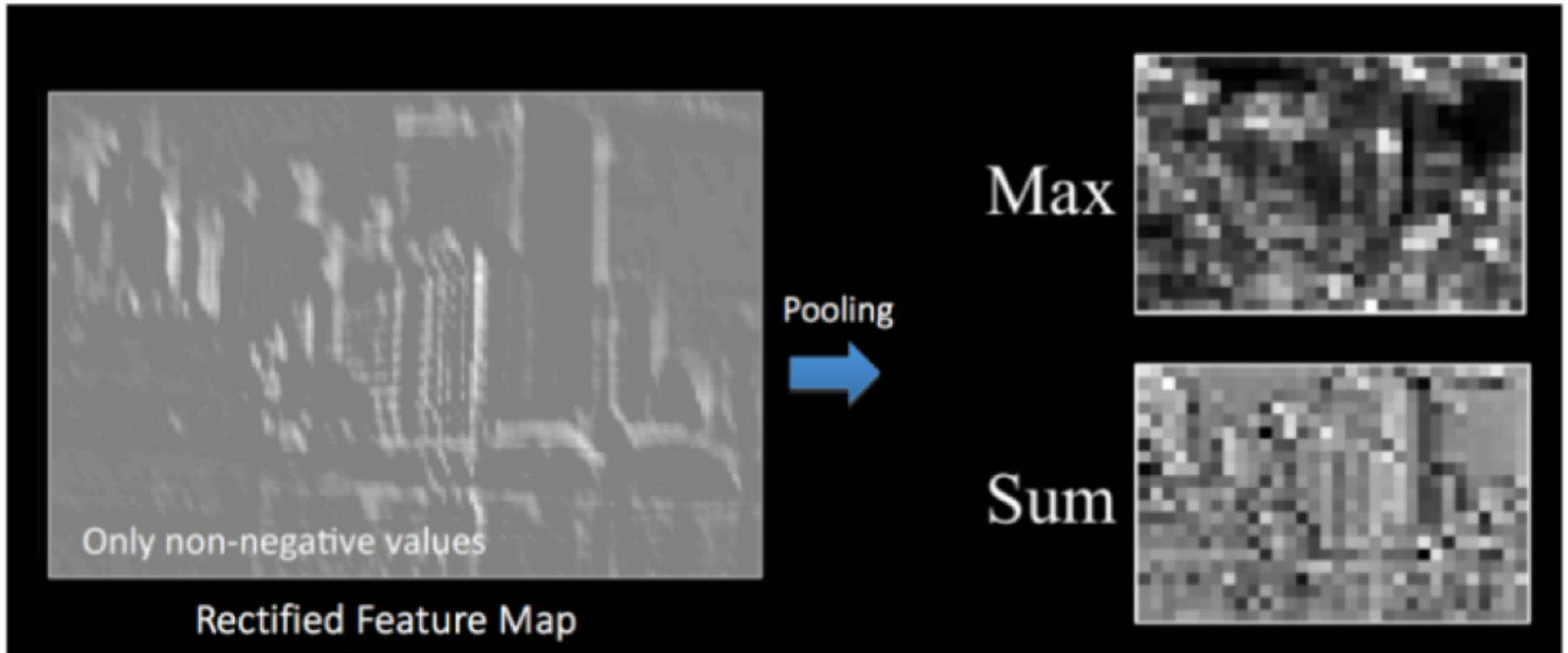
Efeito de Diferentes Filtros de Convolução

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

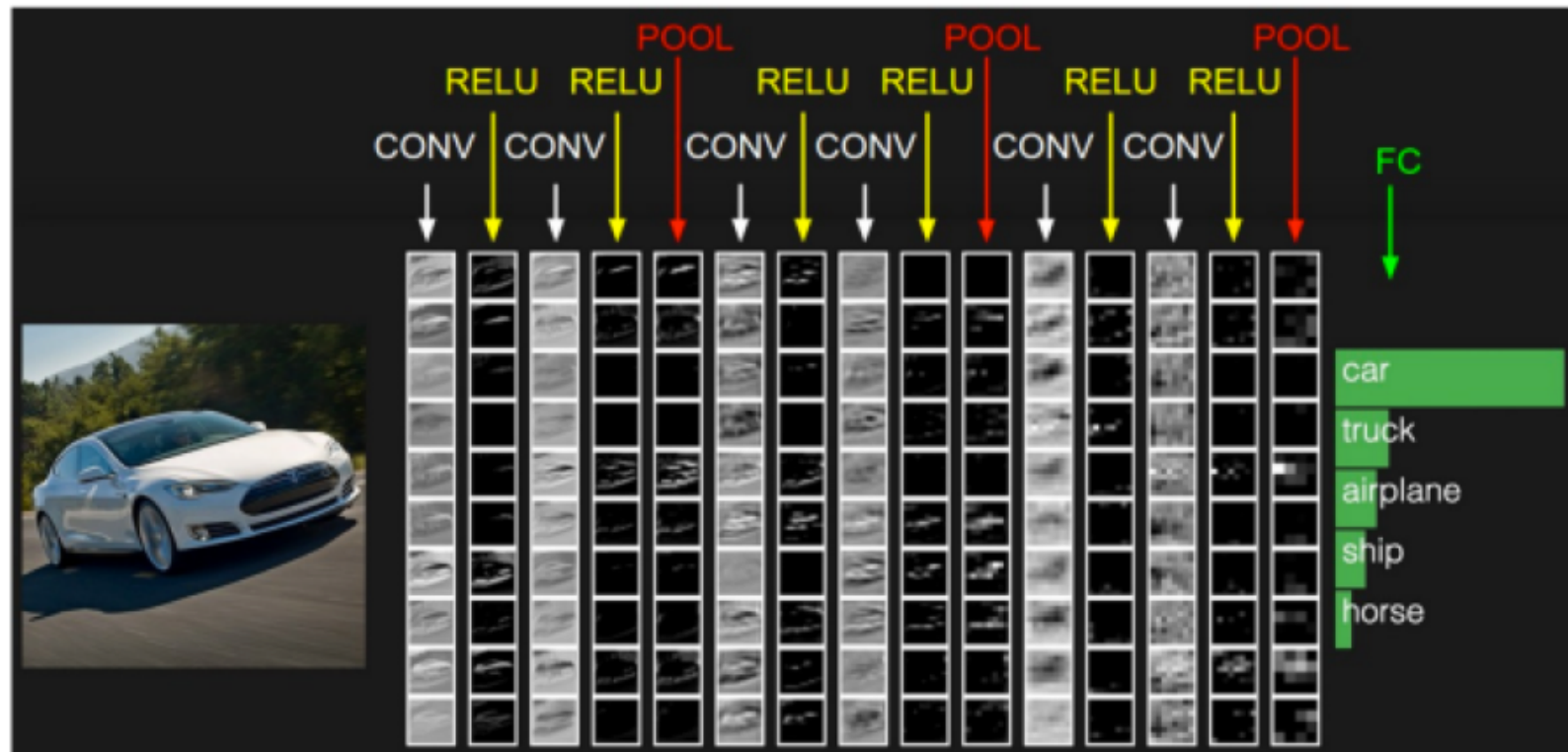
Efeito da Função RELU



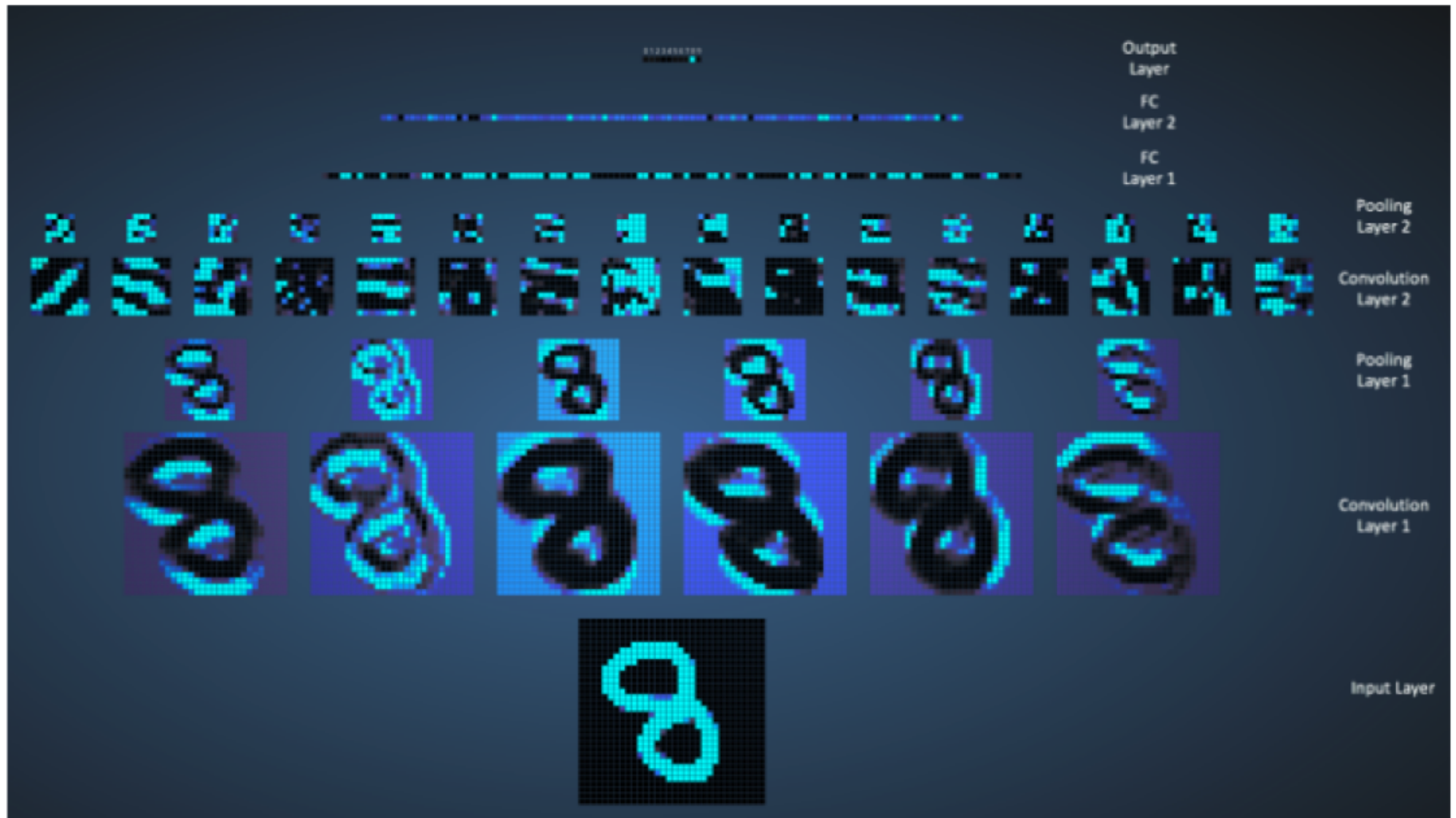
Efeito do Max Pooling



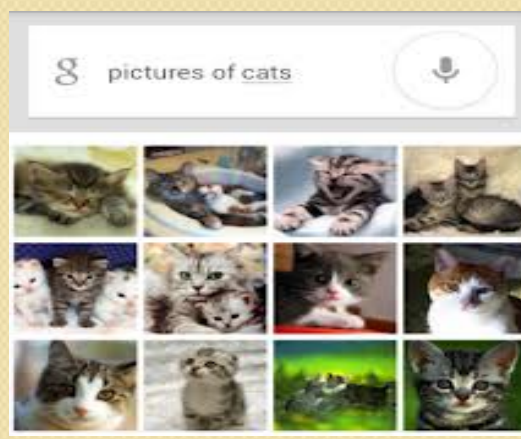
CNN para Reconhecimento de Objetos



Transformações em uma Rede CNN



Da Pesquisa para Tecnologia

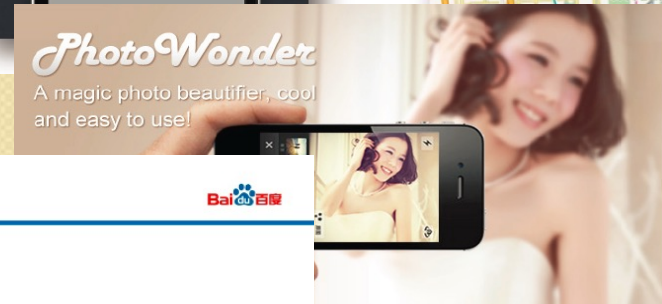


Deep Learning – breakthrough reconhecimento visual e da fala

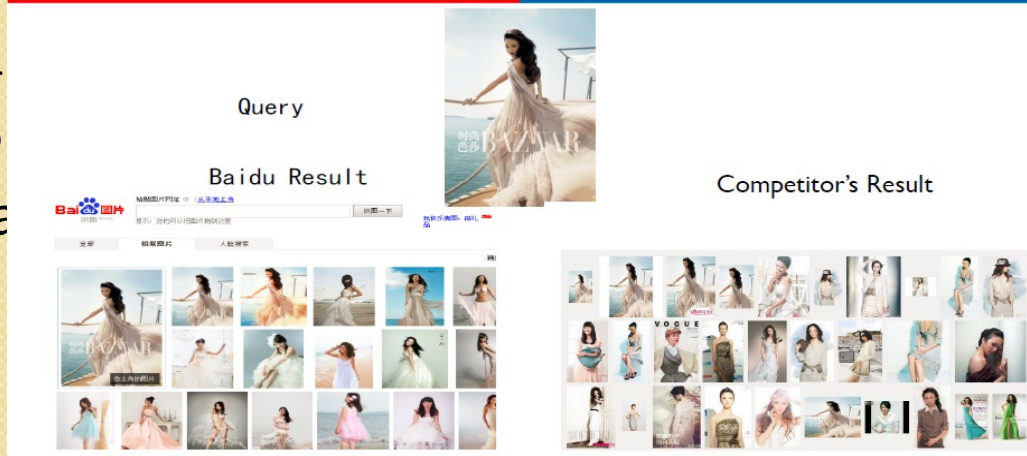
Buzz

- July 2012 – Started DL lab
- Nov 2012 – Big improvement in OCR:
 - Speech – reduce Error rate
 - OCR – reduce Error rate by
- 2013 launch of Baidu Visual Search
 - Voice search
 - Photo Wo
 - Visual search

Baidu map voice search



Baidu Visual Search



Buzz

Scientists See Promise in Deep-Learning Programs



Han Zhong/The New York Times

A voice recognition program translated a speech Chinese.

By JOHN MARKOFF

Published: November 23, 2012

The New York Times

[Microsoft On Deep Learning for Speech](#) goto 3:00-5:10

Buzz



Google Scoops Up Neural Networks Startup DNNresearch To Boost Its Voice And Image Search Tech



RIP EMPSON ✓

Tuesday, March 12th, 2013

5 Comments



Why Google invest in Deep Learning

Buzz

Courant's LeCun to Lead Facebook's New Artificial Intelligence Group

December 9, 2013

143

Facebook has named New York University Professor Yann LeCun the director of a new laboratory devoted to research in artificial intelligence and deep learning.

"As one of the most respected thinkers in this field, Yann has done groundbreaking research in deep learning and computer vision," said Mike Schroepfer, Facebook's chief technology officer. "We're thrilled to welcome him to Facebook."

Facebook is building the team across three locations: Facebook's headquarters in Menlo Park, Calif., New York City, and London.

Machine learning is a branch of artificial intelligence that involves computers "learning" to extract knowledge from massive data sets and rendering informed analyses and judgments often predicting outcomes.

LeCun, a professor at NYU's Courant Institute of Mathematical Sciences, is a pioneer in the growing field. In the 1980s, LeCun proposed one of the early versions of the back-propagation algorithm, the most popular method for training artificial neural networks. In the late 1980s and early 1990s at AT&T Bell Laboratories, he developed the convolutional network model, a pattern-recognition model whose architecture mimics, in part, the visual cortex of animals.

ENTERPRISE

research

uncategorized

Facebook Taps 'Deep Learning' Giant for New AI Lab

BY CADE METZ 12.09.13 3:14 PM

Follow @cademetz

Share 239

Tweet 214

+1 43

Share 48

Pin it



Yann LeCun. Photo: Josh Valcarlos/WIRED

Facebook is building a research lab dedicated to the new breed of artificial intelligence, after hiring one of the preeminent researchers in the field: New York University professor Yann LeCun.

NYU "Deep Learning" Professor LeCun Will Head Facebook's New Artificial Intelligence Lab, Dec 10, 2013

O Resultado? 27/03/19

Turing Award Won by 3 Pioneers in Artificial Intelligence



From left, Yann LeCun, Geoffrey Hinton and Yoshua Bengio. The researchers worked on key developments for neural networks, which are reshaping how computer systems are built.

From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

Mini-projeto: Repetir o Experimento do MNIST com CNNs

<http://neuralnetworksanddeeplearning.com/chap6.html>