

# Aprendizado de Características em Profundidade

Alexandre Xavier Falcão e Giovani Chiachia

Instituto de Computação - UNICAMP

[afalcao@ic.unicamp.br](mailto:afalcao@ic.unicamp.br)

- Uma das metas principais em inteligência artificial é fazer com que a máquina aprenda a partir de exemplos (imagens, sons, dados) de um dado problema, de modo similar ao dos seres humanos.

- Uma das metas principais em inteligência artificial é fazer com que a máquina aprenda a partir de exemplos (imagens, sons, dados) de um dado problema, de modo similar ao dos seres humanos.
- O aprendizado em profundidade busca este objetivo, com técnicas para aprender níveis de **representação e abstração** dos exemplos que sejam próximos dos seus significados.

- Uma das metas principais em inteligência artificial é fazer com que a máquina aprenda a partir de exemplos (imagens, sons, dados) de um dado problema, de modo similar ao dos seres humanos.
- O aprendizado em profundidade busca este objetivo, com técnicas para aprender níveis de **representação e abstração** dos exemplos que sejam próximos dos seus significados.
- Esta aula abordará uma dessas técnicas (*convolutional neural networks*) para o aprendizado de características de imagem.

Considere uma base de imagens com vários exemplos de cada um de  $c$  conceitos diferentes.

Considere uma base de imagens com vários exemplos de cada um de  $c$  conceitos diferentes.

- O objetivo é associar um **vetor de características** para cada imagem de modo que imagens de um mesmo conceito sejam representadas por vetores **similares** (i.e., próximos no espaço de características).

Considere uma base de imagens com vários exemplos de cada um de  $c$  conceitos diferentes.

- O objetivo é associar um **vetor de características** para cada imagem de modo que imagens de um mesmo conceito sejam representadas por vetores **similares** (i.e., próximos no espaço de características).
- O problema consiste em aprender os **parâmetros** e **hiperparâmetros** da função de extração dos vetores de características.

# Convolutional Neural Networks (CNN)

- São uma variante das redes Multilayer Perceptron (MLP).



# Convolutional Neural Networks (CNN)

- São uma variante das redes Multilayer Perceptron (MLP).
- Foram originalmente propostas com inspiração no cortex visual.

# Convolutional Neural Networks (CNN)

- São uma variante das redes Multilayer Perceptron (MLP).
- Foram originalmente propostas com inspiração no cortex visual.
- Os neurônios são sensíveis apenas a uma pequena região do espaço de entrada (*receptive field*).

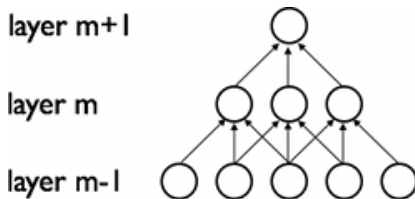


Figura de <http://deeplearning.net/tutorial/lenet.html>

# Convolutional Neural Networks (CNN)

- Inicialmente, dois tipos básicos de neurônios:

# Convolutional Neural Networks (CNN)

- Inicialmente, dois tipos básicos de neurônios:
  - **Simples**: filtragem por um kernel (filtro) seguida de função de ativação.

# Convolutional Neural Networks (CNN)

- Inicialmente, dois tipos básicos de neurônios:
  - **Simples:** filtragem por um kernel (filtro) seguida de função de ativação.
  - **Complexos:** agregação de estímulos de uma dada região de entrada, criando certa invariância com relação a posição exata em que eles ocorreram.

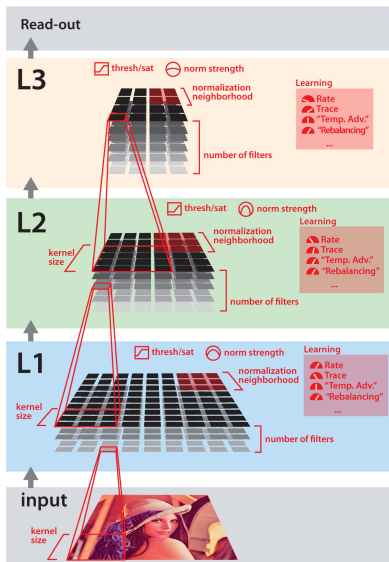
# Convolutional Neural Networks (CNN)

- Inicialmente, dois tipos básicos de neurônios:
  - **Simples**: filtragem por um kernel (filtro) seguida de função de ativação.
  - **Complexos**: agregação de estímulos de uma dada região de entrada, criando certa invariância com relação a posição exata em que eles ocorreram.
- Mais recentemente, novas operações, tal como a **normalização divisiva**: inibição local de estímulos similares.

# Convolutional Neural Networks (CNN)

- Inicialmente, dois tipos básicos de neurônios:
  - **Simplex**: filtragem por um kernel (filtro) seguida de função de ativação.
  - **Complexo**: agregação de estímulos de uma dada região de entrada, criando certa invariância com relação a posição exata em que eles ocorreram.
- Mais recentemente, novas operações, tal como a **normalização divisiva**: inibição local de estímulos similares.
- Imagens naturais  $\Rightarrow$  alta correlação local  $\Rightarrow$  neurônios replicados de maneira a cobrir todo o espaço de entrada  $\Rightarrow$  convolução.

# Exemplo de CNN



- Parâmetros: coeficientes dos filtros.
- Hiperparâmetros: todo o resto, *i.e.*,

+ # camadas,  
+ *receptive field* dos neurônios,  
+ sequência das operações,  
etc.



# Convolução com um banco de filtros (neurônios simples)

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Aprendidos** de forma **discriminativa** ou **generativa**

# Convolução com um banco de filtros (neurônios simples)

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Aprendidos** de forma **discriminativa** ou **generativa**
  - **Globalmente**, e.g., *backpropagation*.

# Convolução com um banco de filtros (neurônios simples)

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Aprendidos** de forma **discriminativa** ou **generativa**
  - **Globalmente**, e.g., *backpropagation*.
  - **Camada por camada**
    - Clusterização: a partir de amostragem aleatória (na camada anterior) cobrindo os  $c$  conceitos.
    - Sparse coding, Restricted Boltzmann Machines, etc.

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Gerados aleatoriamente**

- Tal estratégia pode proporcionar bom desempenho, desde que bem empregada.

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Gerados aleatoriamente**

- Tal estratégia pode proporcionar bom desempenho, desde que bem empregada.
- Alguns truques:
  - Filtro com média dos coeficientes igual à zero e com norma unitária (*i.e.*, dentro da *unit sphere*).
  - Boa função de ativação (*e.g.*, *rectified linear*, etc.)

Os coeficientes de um banco com  $N$  filtros  $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_N$  podem ser

- **Gerados aleatoriamente**

- Tal estratégia pode proporcionar bom desempenho, desde que bem empregada.
- Alguns truques:
  - Filtro com média dos coeficientes igual à zero e com norma unitária (*i.e.*, dentro da *unit sphere*).
  - Boa função de ativação (*e.g.*, *rectified linear*, etc.)
- É importante que os filtros (kernels) sejam **linearmente independentes**.

# Convolução com um banco de filtros

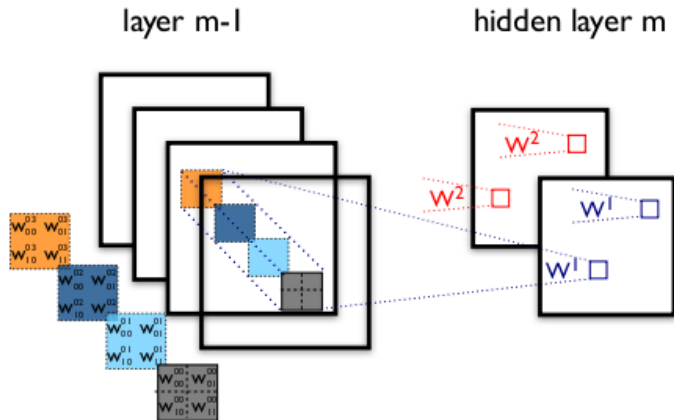
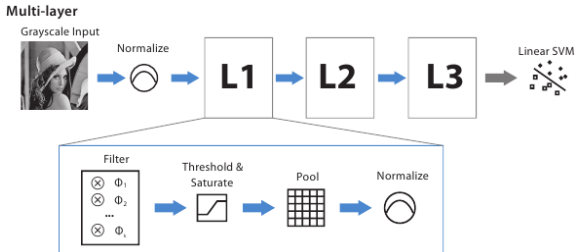


Figura de <http://deeplearning.net/tutorial/lenet.html>

A figura abaixo ilustra as operações na arquitetura adotada.



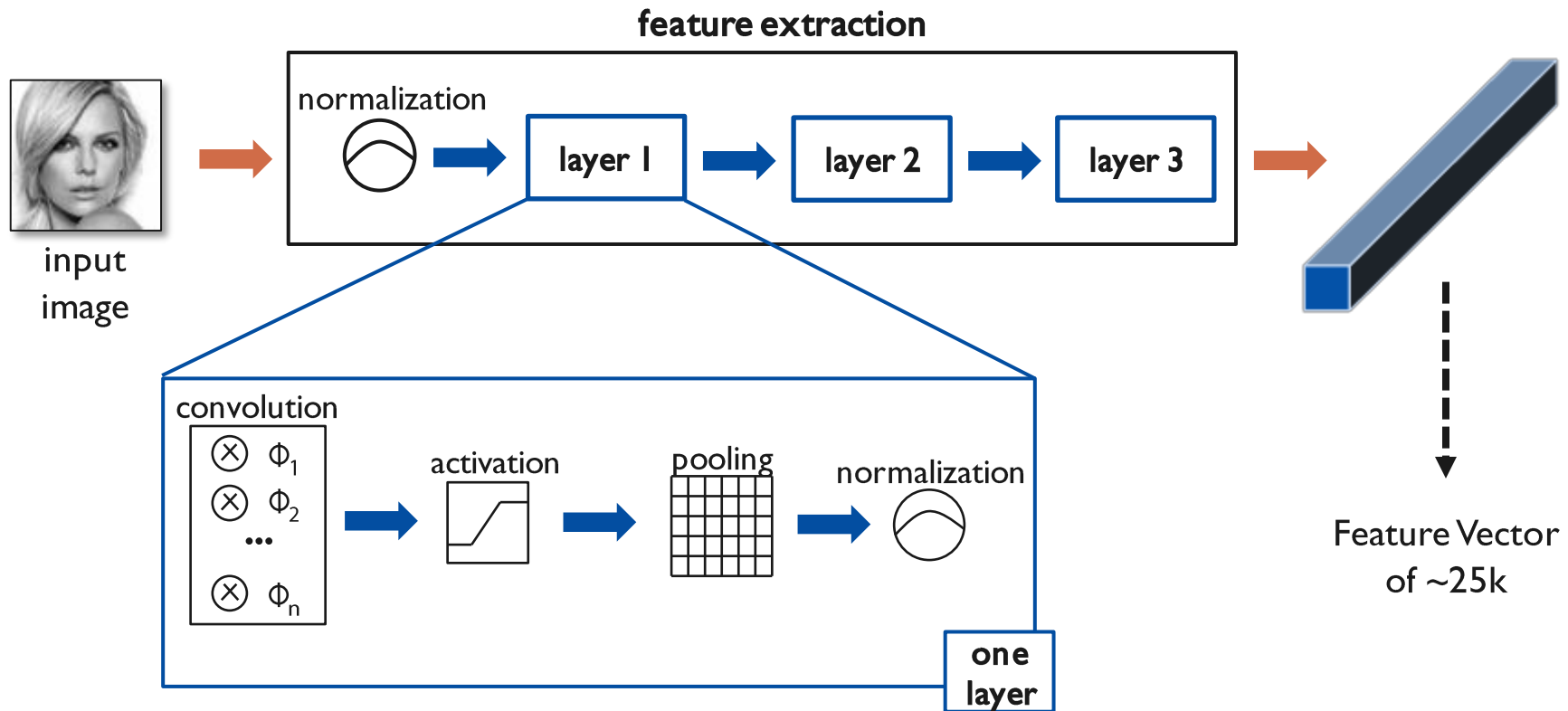
\* Figura do artigo de N. Pinto e D. Cox em IEEE Intl. Conf. on Automatic Face and Gesture Recognition, 2011.

\* As operações a seguir são repetidas com parâmetros distintos para cada camada, os quais precisam ser aprendidos.



# ConvNets

## HT-L3-1st<sup>1</sup>



# Convolução com um banco de filtros

- Considerando uma imagem em convolução com  $m$  bandas, podemos definir o banco de filtros

$$\hat{K} = (\mathcal{B}, \vec{K}_i), i = 1, 2, \dots, n,$$

onde  $n$  é o número de filtros do banco e

$\vec{K}_i = (K_{1,i}, K_{2,i}, \dots, K_{m,i})$  é um filtro do banco.

# Convolução com um banco de filtros

- Considerando uma imagem em convolução com  $m$  bandas, podemos definir o banco de filtros

$$\hat{K} = (\mathcal{B}, \vec{K}_i), i = 1, 2, \dots, n,$$

onde  $n$  é o número de filtros do banco e

$\vec{K}_i = (K_{1,i}, K_{2,i}, \dots, K_{m,i})$  é um filtro do banco.

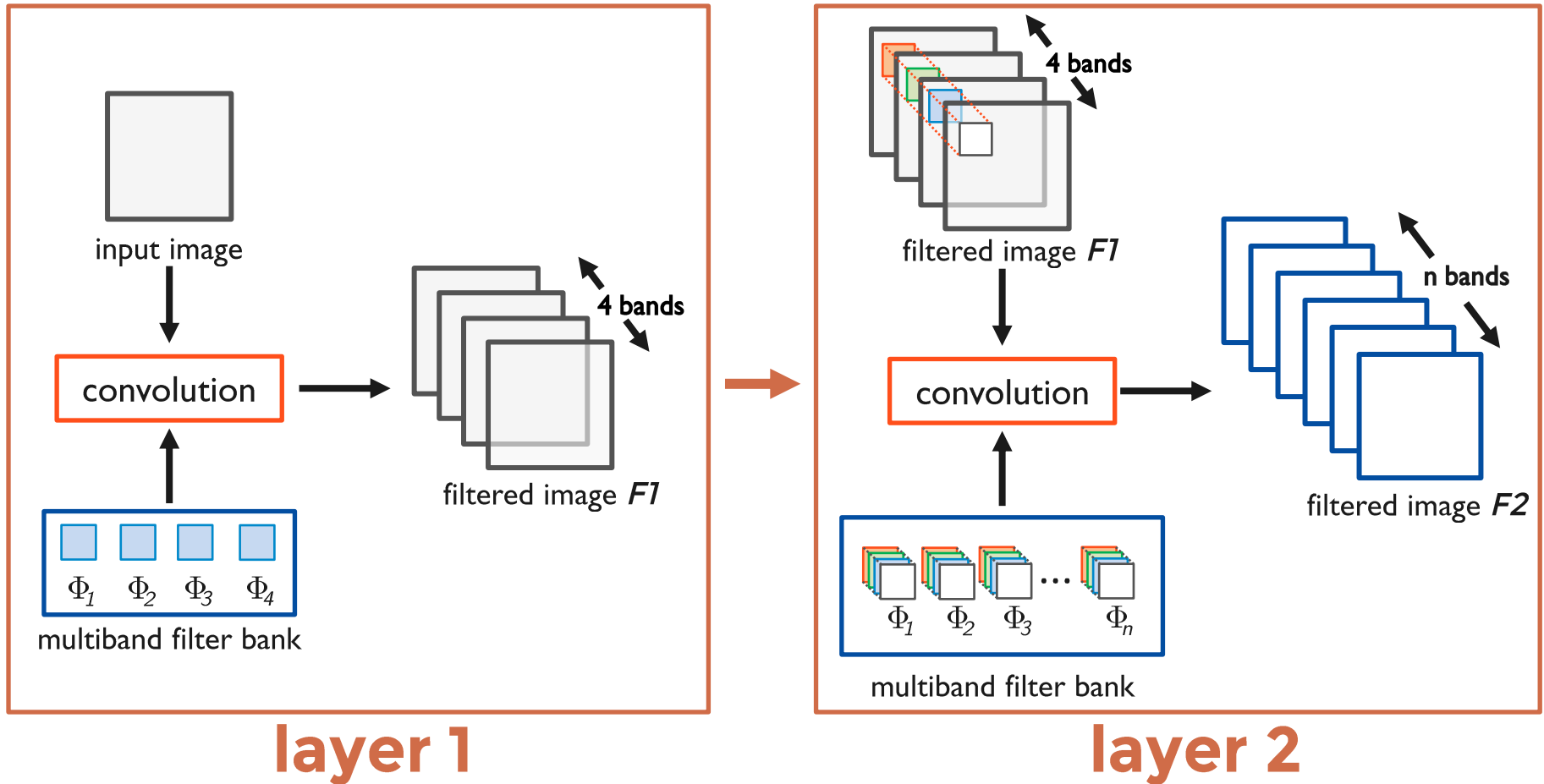
- Para  $\hat{J} = \hat{I}' * \hat{K}$ ,

$$J_i(p) = \sum_{\forall q \in \mathcal{B}} \vec{I}'(q) \cdot \vec{K}_i(p - q)$$

gera a banda  $i$  da imagem  $\hat{J} = (D_I, \vec{J})$ ,  $\vec{J} = (J_1, J_2, \dots, J_N)$  e os valores de  $J_i(p)$  são ainda submetidos à função de ativação.  $\vec{I}'(q)$  são os valores normalizados da imagem, como será explicado ao final do processo.

# ConvNets

## Convolution



# Relação entre convolução e perceptron

A convolução com cada kernel  $\hat{K}_i = (\mathcal{B}, \vec{K}_i)$  tem uma relação com a passagem da informação por um neurônio centrado em cada pixel.

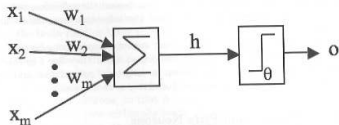
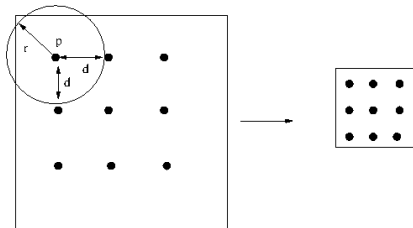


FIGURE 1.6: A picture of McCulloch and Pitt's mathematical model of a neuron. The inputs  $x_i$  are multiplied by the weights  $w_i$ , and the neurons sum their values. If this sum is greater than the threshold  $\theta$  then the neuron fires, otherwise it does not.

\* Figura do livro Machine Learning: An algorithmic perspective, por Stephen Marsland, 2009.

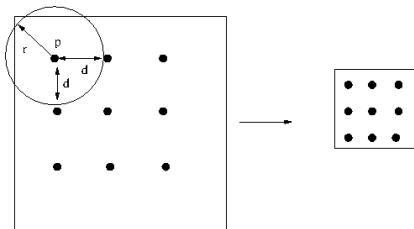
# Pooling (neurônios complexos)

Operação de pooling em uma adjacência  $\mathcal{C}$  (e.g., circular de raio  $r$ ), centrada no pixel  $p$  e feita a cada  $d$  pixels (**stride**).



# Pooling (neurônios complexos)

Operação de pooling em uma adjacência  $\mathcal{C}$  (e.g., circular de raio  $r$ ), centrada no pixel  $p$  e feita a cada  $d$  pixels (**stride**).

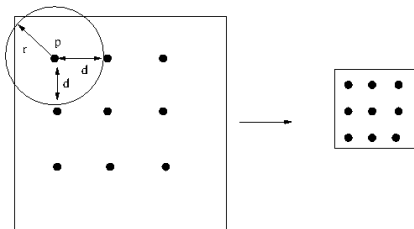


Efeitos:

- Cria certa invariância à translação (na forma apresentada).
- Quando  $d > 1$ , reduz significativamente a resolução espacial da imagem.

# Pooling (neurônios complexos)

Operação de pooling em uma adjacência  $\mathcal{C}$  (e.g., circular de raio  $r$ ), centrada no pixel  $p$  e feita a cada  $d$  pixels (**stride**).



Efeitos:

- Cria certa invariância à translação (na forma apresentada).
- Quando  $d > 1$ , reduz significativamente a resolução espacial da imagem.

Determinar o *receptive field* desses neurônios é um problema em aberto.



pooling  
region

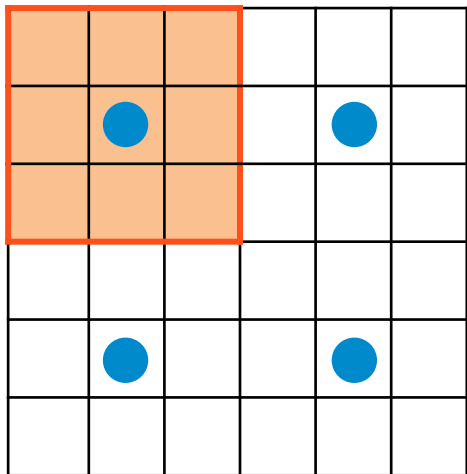
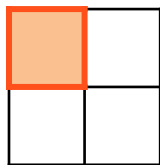


image band  $i$



pooled  
feature



pooled  
image band  $i$

$$L_{\beta} = 3, \\ s = 3 \text{ (no overlapping)}$$

Normalmente  $d < r$  e a nova imagem  $\hat{I} = (D_I, \vec{I})$  gerada com o *pooling*, tem  $\vec{I} = (I_1, I_2, \dots, I_N)$  e essa operação é normalmente definida por

$$I_i(p) = \sqrt[\alpha]{\sum_{\forall q \in \mathcal{C}(p)} J_i(q)^\alpha},$$

onde  $i = 1, 2, \dots, N$  e  $\alpha$  controla a sensibilidade da operação, *i.e.*, quanto maior, mais importância será dada ao maior coeficiente de entrada.

Seja  $\hat{I} = (D_I, \vec{I})$  uma imagem antes da normalização, a operação

$$I'_j(p) = \frac{I_j(p)}{\sqrt{\sum_{j=1}^m \sum_{\forall q \in \mathcal{A}(p)} I_j(q) * I_j(q)}},$$

gera a imagem  $\hat{I}' = (D_I, \vec{I}')$ , onde  $j = 1, 2, \dots, m$  são as bandas da imagem e  $\mathcal{A}$  é normalmente quadrada, mas poderia ser circular de raio  $r$  (e.g.,  $r = \sqrt{2}, \sqrt{5}, \dots$ ).

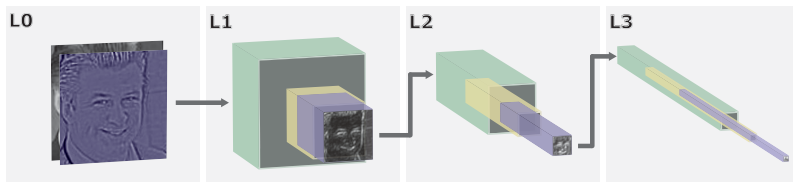
Seja  $\hat{I} = (D_I, \vec{I})$  uma imagem antes da normalização, a operação

$$I'_j(p) = \frac{I_j(p)}{\sqrt{\sum_{j=1}^m \sum_{\forall q \in \mathcal{A}(p)} I_j(q) * I_j(q)}},$$

gera a imagem  $\hat{I}' = (D_I, \vec{I}')$ , onde  $j = 1, 2, \dots, m$  são as bandas da imagem e  $\mathcal{A}$  é normalmente quadrada, mas poderia ser circular de raio  $r$  (e.g.,  $r = \sqrt{2}, \sqrt{5}, \dots$ ).

Quando aplicada diretamente sobre a imagem, essa operação ressalta as regiões de borda (textura).

# Ilustração de modelo



\* O vetor de atributos concatena os atributos de cada superpixel em L3 da esquerda para direita e de cima para baixo.

# Algumas fontes sobre o assunto

- [deeplearning.net](http://deeplearning.net)
- [ufldl.stanford.edu/wiki/index.php/UFLDL\\_Tutorial](http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial)

# Referências

Cox, David, and Nicolas Pinto. "Beyond simple features: A large-scale feature search approach to unconstrained face recognition." Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on.

Menotti, D., Chiachia, G., Pinto, A., Robson Schwartz, W., Pedrini, H., Xavier Falcao, A., & Rocha, A. (2015). Deep Representations for Iris, Face, and Fingerprint Spoofing Detection. Information Forensics and Security, IEEE Transactions on, 10(4), 864-879.

Martins, S.B., Chiachia, G., and Falcão, A.X. "A fast and robust negative mining approach for user enrollment in face recognition systems". Master Dissertation, University of Campinas, Brazil, 2015

<http://ic.unicamp.br/~chiachia/talks/a-few-things-about-deep-learning/>