Redes Neurais Artificiais

Prof. Dr. Hugo Valadares Siqueira

Aula 18 – Deep Learning

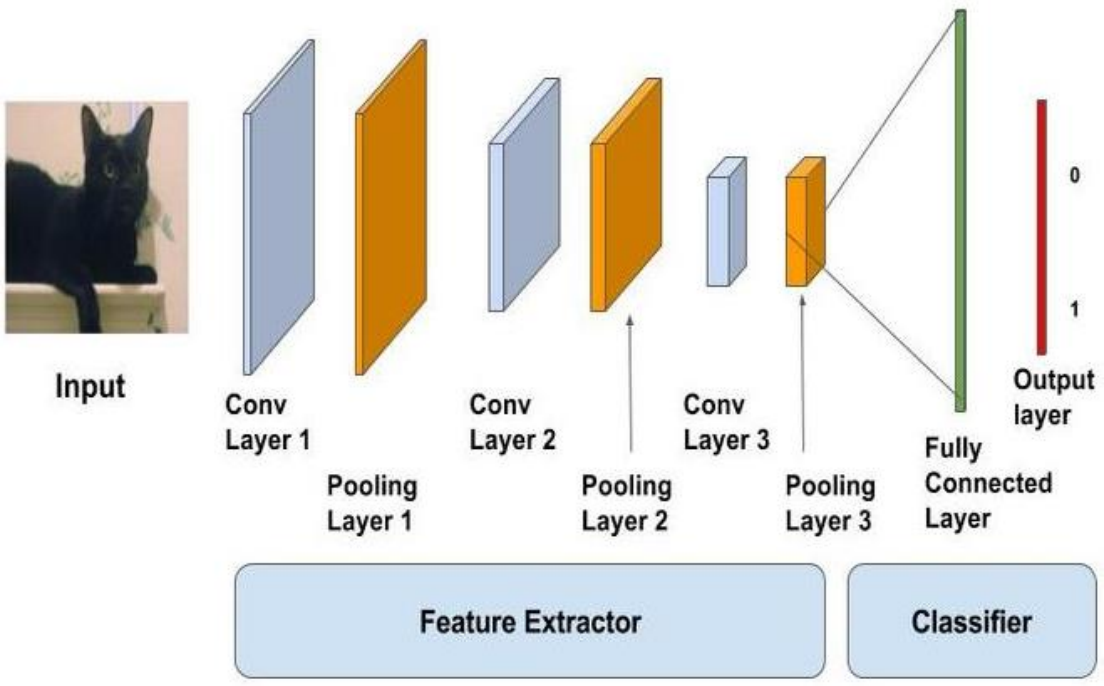
Laboratório de Inteligência Computacional e Controle Avançado

Aprendizado profundo

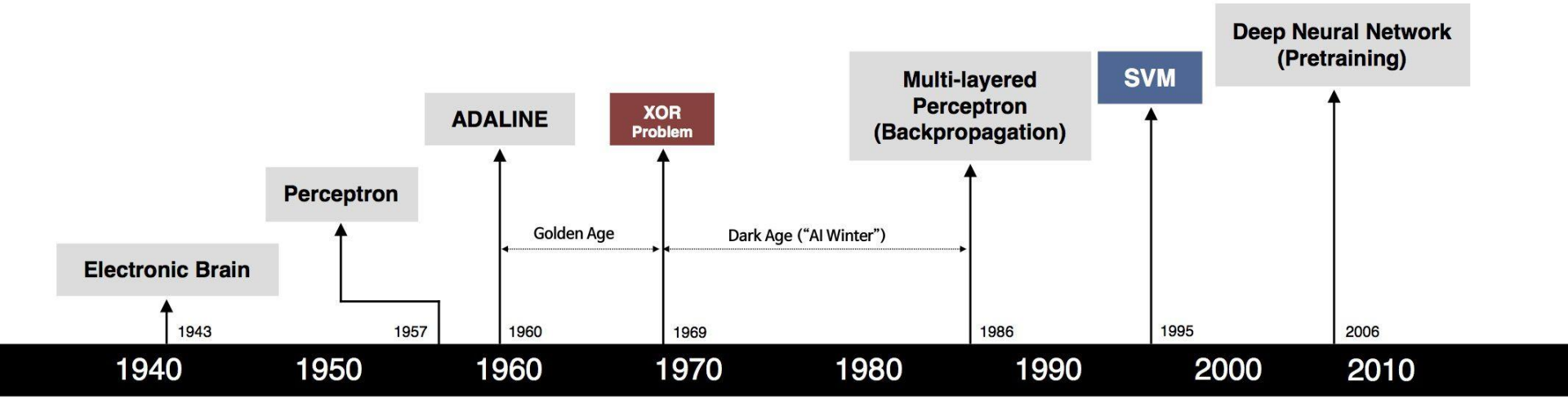
- Representação do aprendizado: aprendendo uma boa representação (características) para os dados;
- Aprendizado profundo: aprendendo várias camadas de características;
- Um modelo pode ser tão bom quanto seus recursos permitem → então devemos aprender boas características;
- Podem ser usados com muitos paradigmas de aprendizado: supervisionado, sem supervisão, semi-supervisionado, etc.
- CNN - Valores para aplicativos reais:

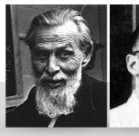
➢ Muitas camadas (> 5).

➢ Muitos parâmetros (> 10M).



Linha do tempo das RNAs




S. McCulloch – W. Pitts


F. Rosenblatt

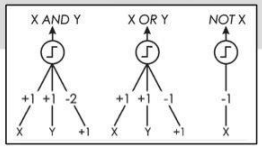

B. Widrow – M. Hoff


M. Minsky – S. Papert

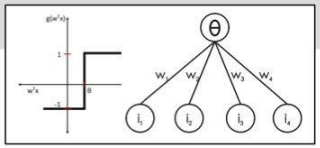

D. Rumelhart – G. Hinton – R. Wiliams


V. Vapnik – C. Cortes

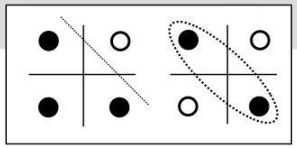

G. Hinton – S. Ruslan



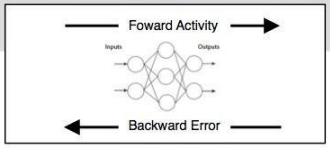
- Adjustable Weights
- Weights are not Learned



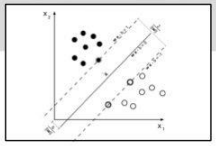
- Learnable Weights and Threshold



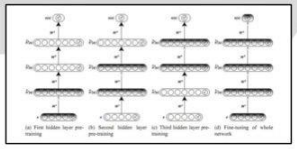
- XOR Problem



- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting



- Limitations of learning prior knowledge
- Kernel function: Human Intervention



- Hierarchical feature Learning

Deep Learning Timeline

1/1/1940 Dark Era Until 1940

1/2/1943 Neural Nets McCulloch & Pitt

1/1/1950 Computing Machinery and Intelligence Alan Turing

??? (???)

1/1/1958 Perceptron Rosenblatt

1/1/1960 ADALINE Widrow & Hof

1/1/1969 XOR problem Minsky & Papert

1/1/1974 Backpropagation Werbos (and more)

1/1/1980 Self Organizing Map Kohonen

1/1/1980 Neocogitron Fukushima

1/1/1982 Hopfield Network John Hopfield

1/1/1985 Boltzmann Machine Hinton & Sejnowski

1/1/1986 Restricted Boltzmann Machine Smolensky

1/1/1986 Multilayer Perceptron Rumelhart, Hinton & Williams

1/2/1986 RNNs Jordan

1/1/1990 LeNet Lecun

1/1/1997 Bidirectional RNN Schuster & Paliwal

1/2/1997 LSTMs Hochreiter & Schmidhuber

1/1/2006 Deep Belief Networks - pretraining Hinton

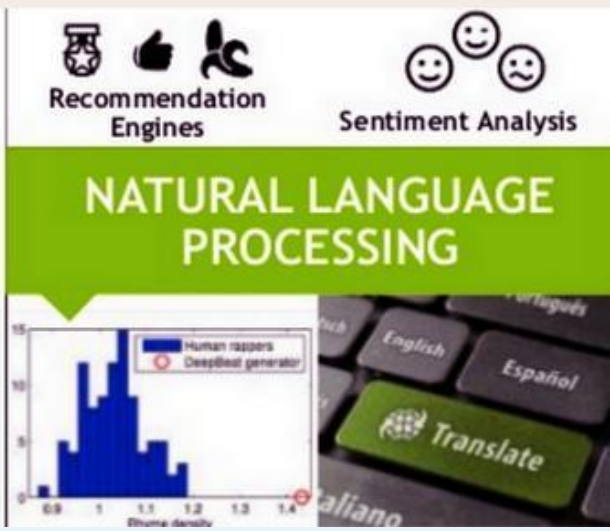
1/2/2006 Deep Boltzmann Machines Salakhutdinov & Hinton

1/1/2012 Dropout Hinton

1/1/2014 GANs Goodfellow

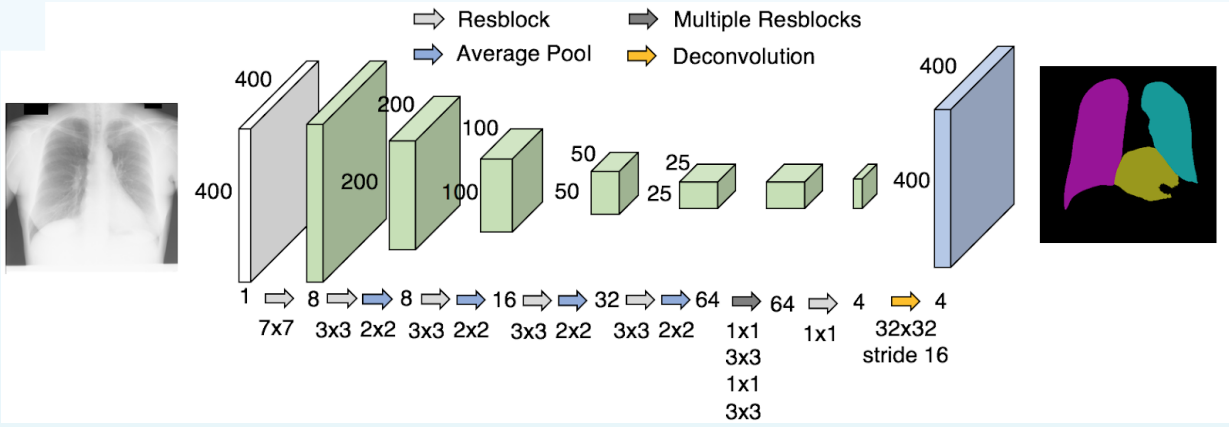
1/1/2017 Capsule Networks Sabour, Frosst, Hinton

Made by Favio Vázquez



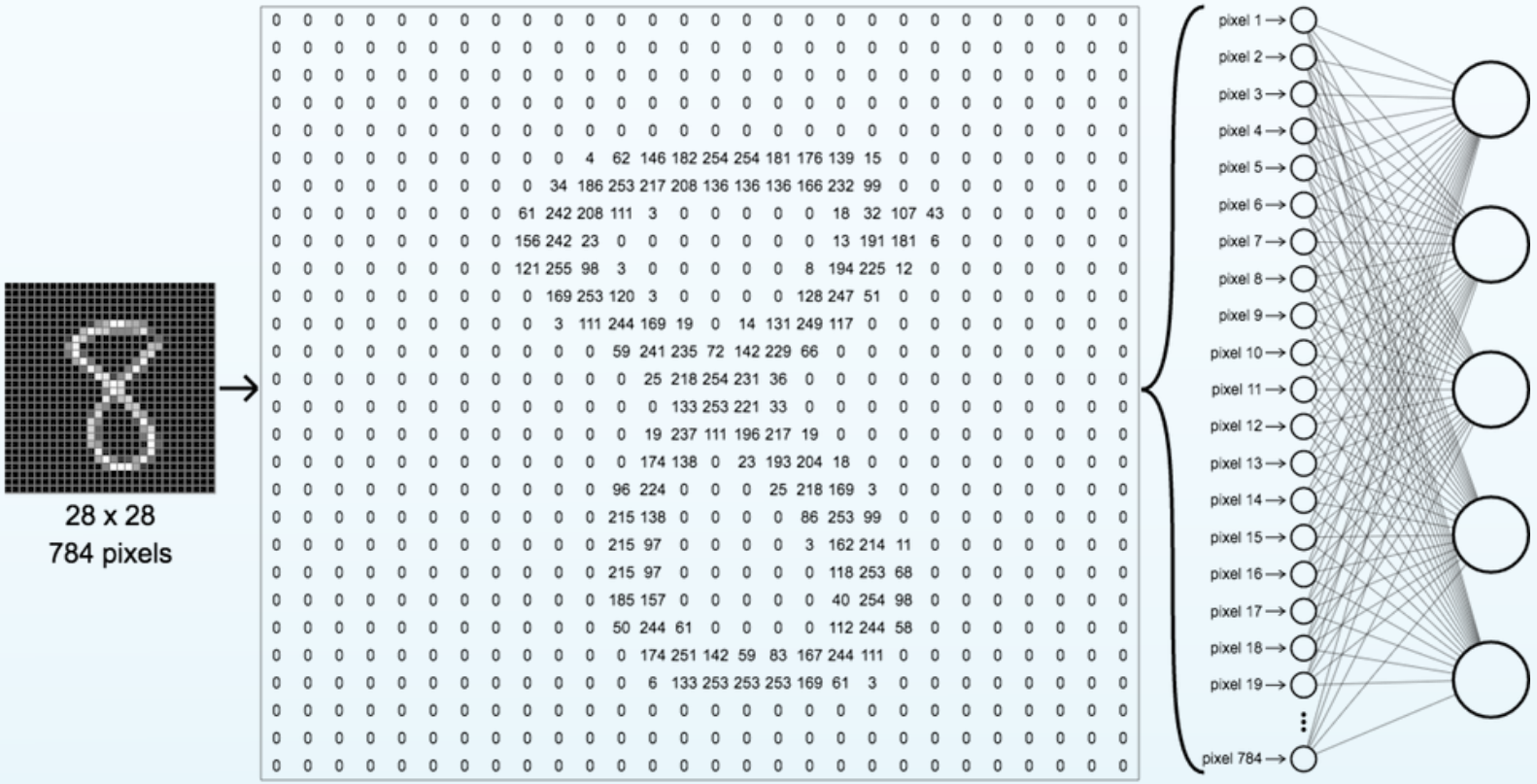
Porque uma RNA funciona?

- Aproximador universal: pode reduzir o erro a um determinado nível usando muitos neurônios e amostras em rede rasa;
- Redes profundas: adicionar uma camada pode reduzir o número de neurônios necessários na camada anterior exponencialmente;
- Hipótese múltipla: dados de alta dimensão estão em um coletor de baixa dimensão;
- Alguns modelos alcançam desempenho muito melhor do que algoritmos específicos;
- O mesmo modelo pode ser usado por muitos problemas diferentes;
- O problema passa a ser a construção de aprendizado, não a construção de modelos;
- Use rede neural única para todo o sistema.



O Problema da Extração de Características

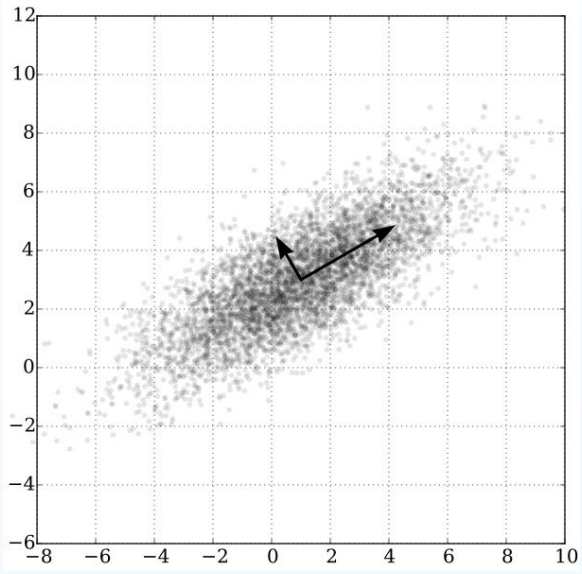
- Redes Neurais Convencionais têm dificuldades para processar dados naturais em sua forma bruta;
- Exemplos: imagens, texto...



Fonte: https://ml4a.github.io/ml4a/neural_networks/

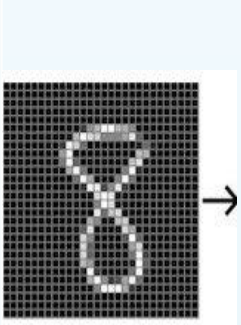
O Problema da Extração de Características

- Redes Neurais Convencionais: solução geralmente adotada é a extração de características;
- Exemplos em imagens: PCA, Atributos de textura.
- Problemas:
  - ➢ Nem sempre as características relevantes para a classificação são extraídas;
  - ➢ É dependente do contexto;
  - ➢ Muitas vezes requer um especialista para selecionar as características para determinada área.

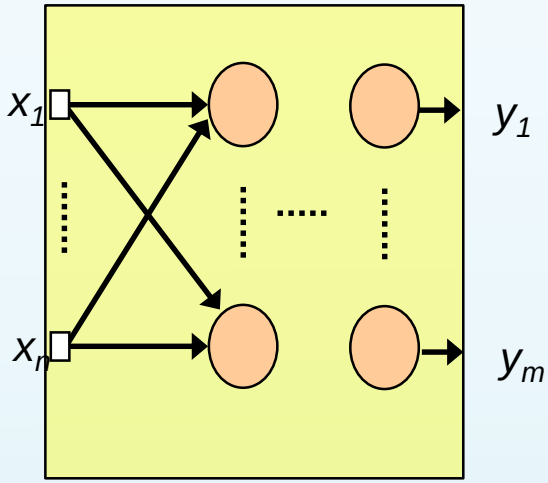


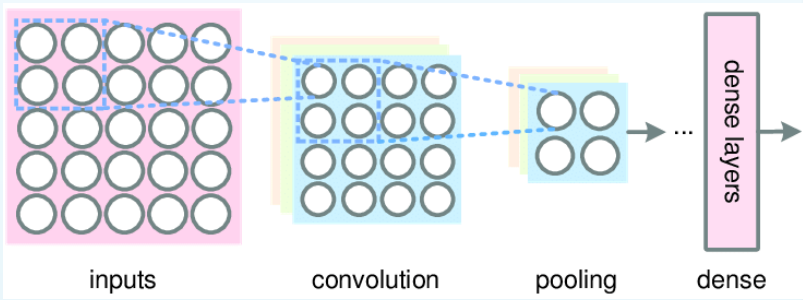
Fonte: https://en.wikipedia.org/wiki/Principal_component_analysis#/media/File:GaussianScatterPCA.svg

RNA



Extração de caracte-rísticas

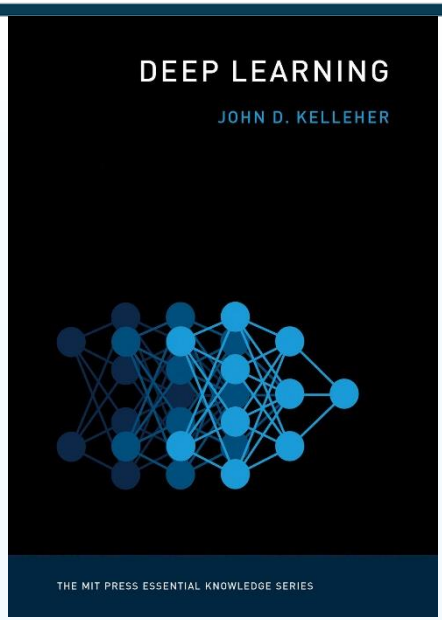




Vanishing Gradient Problem

- Redes Neurais com muitas camadas: transformações sucessivas da informação de entrada;
- Transformam a representação em um nível, começando da camada de entrada, em uma representação em um nível maior e ligeiramente mais abstrato;
- Problema: perda da informação do gradiente;
- Utilizando o Backpropagation, a informação do gradiente é propagada para trás;
- Entretanto, quanto mais se distancia da camada de saída, mais a informação do gradiente diminui;
- Este problema, aliado ao crescimento do número de parâmetros ajustáveis (pesos), são os motivos principais de redes neurais convencionais utilizarem poucas camadas;
- O problema é ainda pior quando são utilizadas funções de ativação tradicionais como sigmoidal e tangente hiperbólica;
- O uso da regra da cadeia tem o efeito de multiplicar vários números pequenos produzidos pelas funções de ativação, fazendo com que o gradiente diminua exponencialmente.

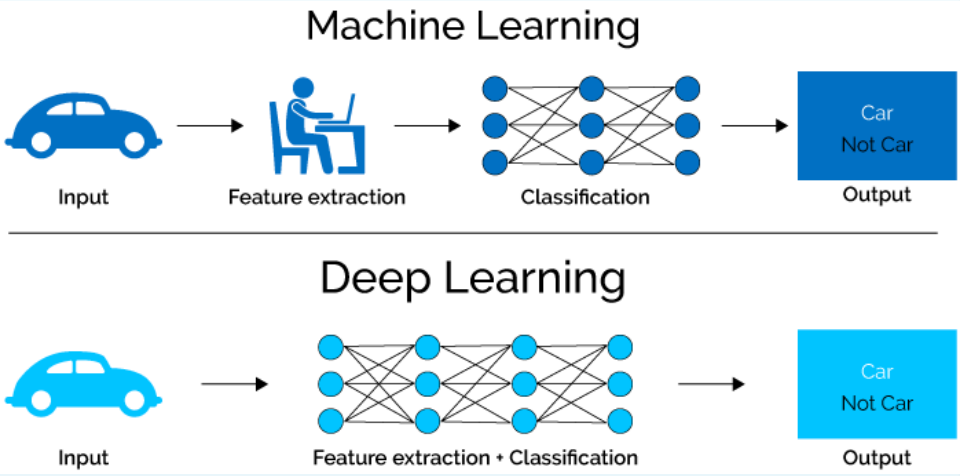
Redes Neurais Convolucionais



- *Deep Learning* (Aprendizado Profundo);
- Métodos de aprendizagem de representação:
  - ➢ Permitem que uma máquina, ao ser alimentada com dados brutos, descubra automaticamente as melhores representações para detecção ou classificação;
- Métodos de aprendizagem profunda são:
  - ➢ Métodos de aprendizagem de representação com múltiplos níveis de representação;
  - ➢ São compostos por módulos simples não-lineares, que transformam a representação em um nível ligeiramente mais abstrato;
- Com a composição de tais transformações simples, funções muito complexas podem ser estimadas;
- Para tarefas de classificação, camadas superiores de representação amplificam os aspectos das entradas que são importantes para a discriminação e suprimem variações que são irrelevantes.

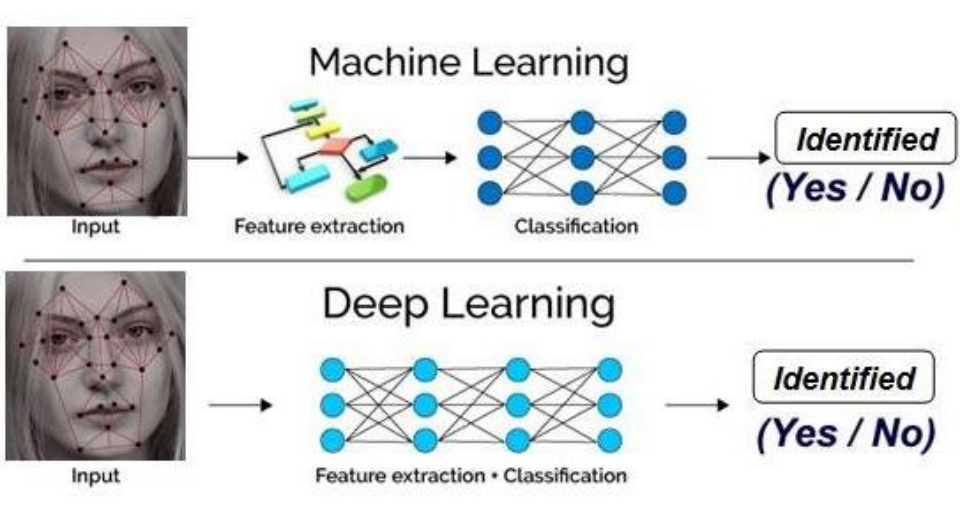
Redes Neurais Convolucionais

- Uma CNN usa uma variação de MLPs desenvolvidas de modo a demandar o mínimo pré-processamento possível;
- A rede aprende os filtros que em um algoritmo tradicional precisariam ser implementados manualmente;
- Tem-se independência de um conhecimento a priori e da ação humana no desenvolvimento de suas funcionalidades básicas, o que é uma vantagem de sua aplicação;
- Usada principalmente em reconhecimento de imagens e processamento de vídeo;
- Na saúde usa-se esta metodologia com algoritmos específicos, recorrendo a um grande número de fotografias clínicas, para o diagnóstico, com resultados muito precisos e comparáveis aos clínicos especializados;



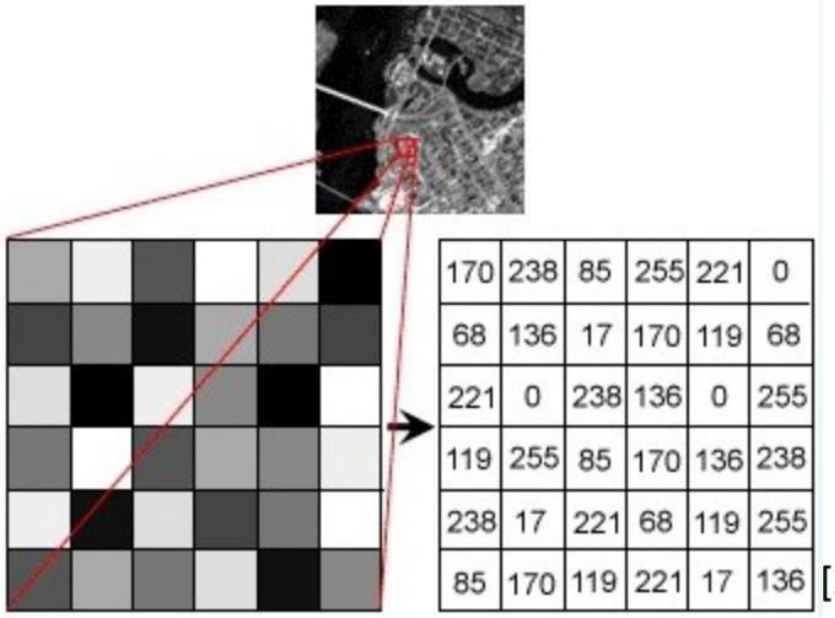
Redes Neurais Convolucionais

- Exemplo: classificação de imagens:
  - Uma imagem vem na forma de uma matriz de valores de pixel;
  - As características descobertas na primeira camada tipicamente indicam a presença ou ausência de bordas em orientações e locais específicos da imagem;
  - A segunda camada normalmente detecta pequenos padrões particulares de bordas, independentemente de pequenas variações nas posições e orientações destas;
  - A terceira camada pode combinar pequenos padrões que resultam em padrões maiores que correspondem a partes de objetos familiares;
  - Camadas subsequentes detectariam objetos por meio da combinação de padrões obtidos anteriormente.

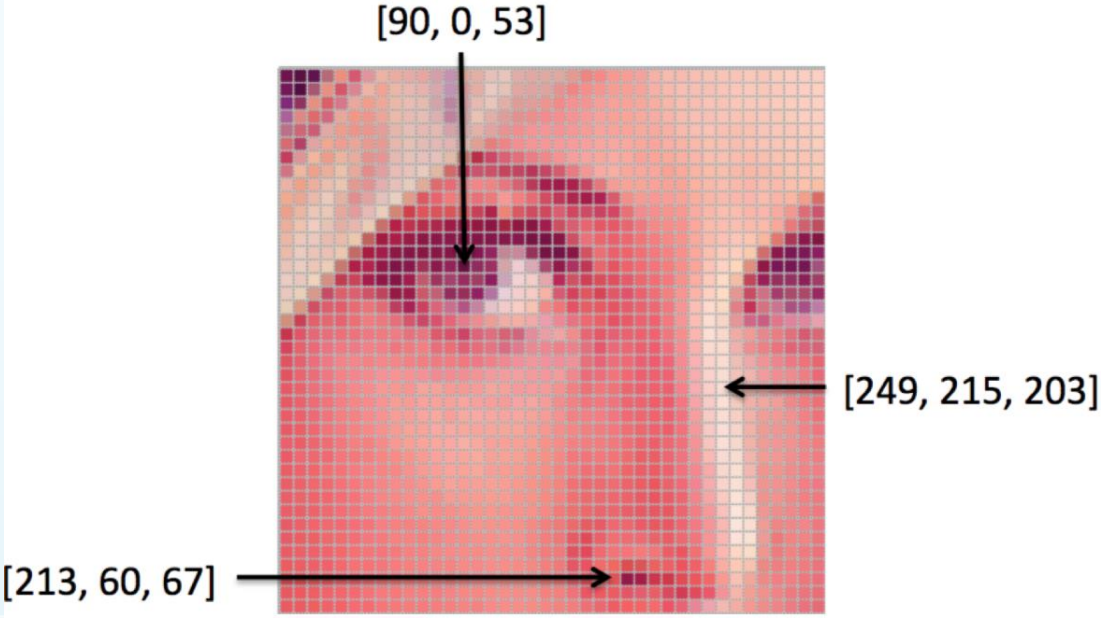


Processamento de imagens

- Uma imagem preta e branca (*grayscale*) é representada como uma matriz 2D;
- Cada elemento representa um pixel da imagem, no qual os valores variam entre 0 (preto) até 255 (branco);
- Uma imagem colorida, é normalmente representada por uma matriz 3D de forma que seja possível armazenar uma combinação das cores vermelho, verde e azul (RGB).



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136



Camadas

- Existem diversos tipos de camadas em uma CNN;
- As camadas mais básicas são: camadas de convolução, camadas de agrupamento (*pooling*) e camadas totalmente conectadas;

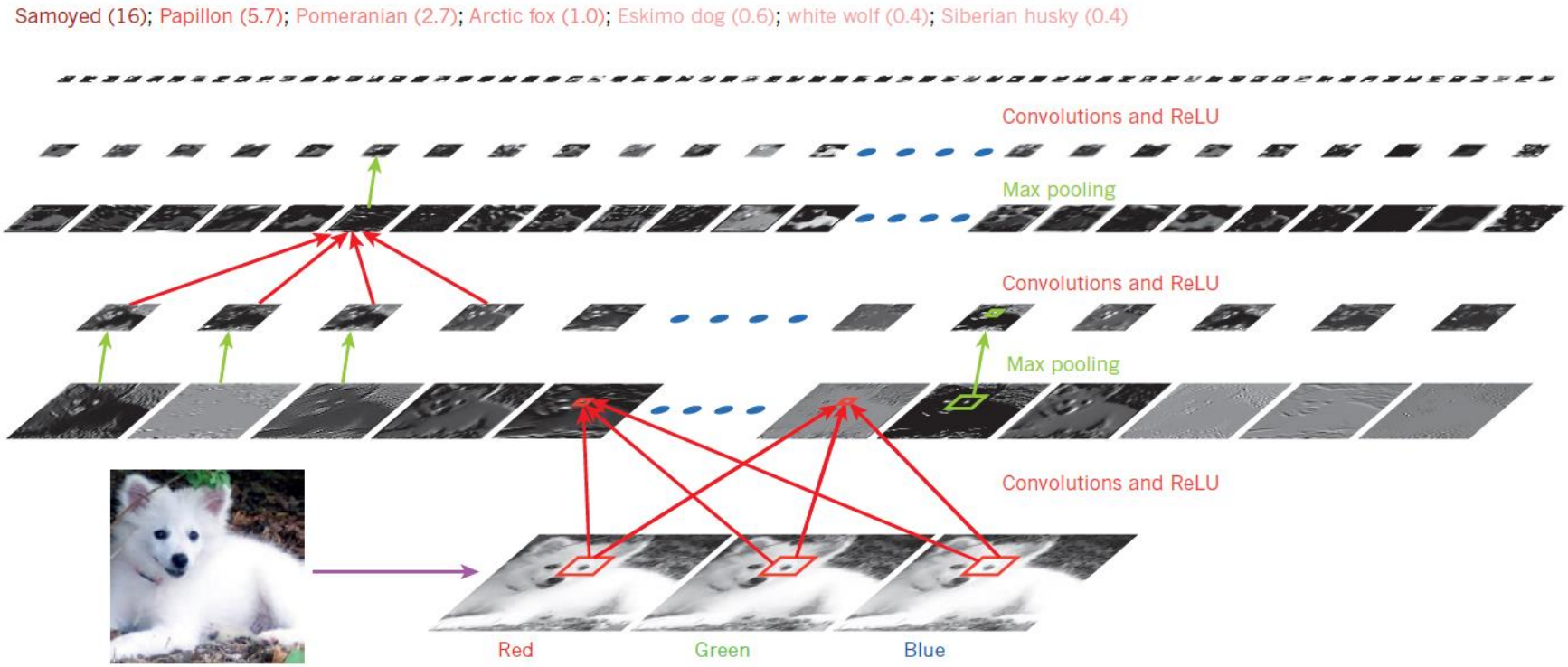
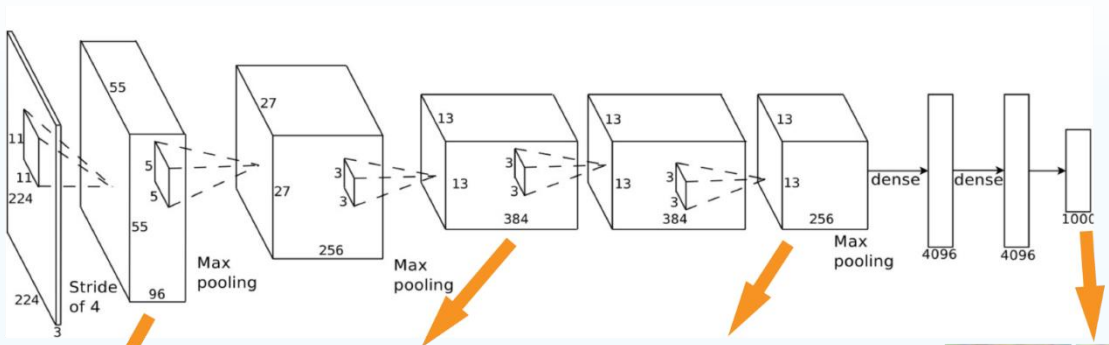
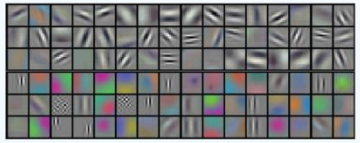


Figure 2 | Inside a convolutional network. The outputs (not the filters) of each layer (horizontally) of a typical convolutional network architecture applied to the image of a Samoyed dog (bottom left; and RGB (red, green, blue) inputs, bottom right). Each rectangular image is a feature map corresponding to the output for one of the learned features, detected at each of the image positions. Information flows bottom up, with lower-level features acting as oriented edge detectors, and a score is computed for each image class in output. ReLU, rectified linear unit.

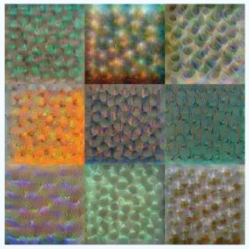
Fonte: LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning", Nature, 521(7553), 436-444.

Camadas

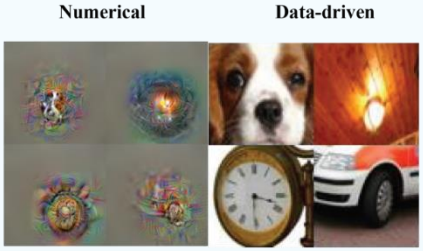




Conv 1: Edge+Blob

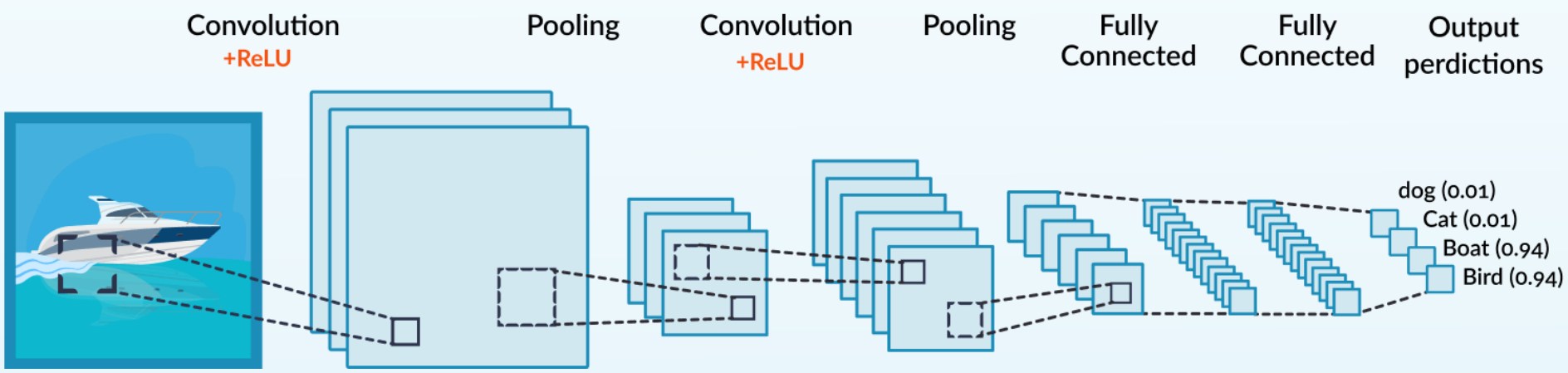


Conv 3: Texture



Conv 5: Object Parts

Fc8: Object Classes



Fonte: https://hackernoon.com/learning-ai-if-you-suck-at-math-p5-deep-learning-and-convolutional-neural-nets-in-plain-english-cda79679bbe3

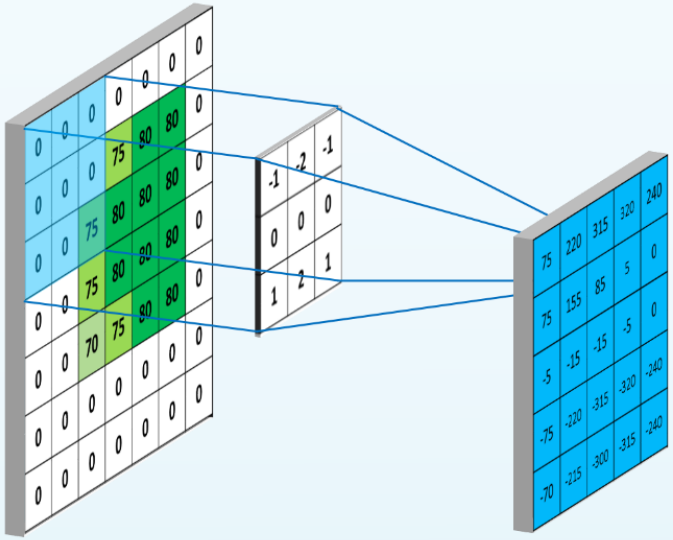
Convolução

- Matematicamente, uma convolução é uma operação linear representada por $(*)$ que a partir de duas funções, gera uma terceira (*feature map*);
- No contexto de imagens, podemos entender esse processo como um filtro/*kernel* que transforma uma imagem de entrada;
- Um *kernel* aqui é entendido como uma matriz utilizada para uma operação de multiplicação de matrizes;
- Esta operação é aplicada diversas vezes em diferentes regiões da imagem;
- A cada aplicação, a região é alterada por um parâmetro conhecido como *stride;*
- Normalmente o *stride* possui o valor 1, o que significa que a transformação será aplicada em todos os *pixels* da imagem;
- A matriz resultante possui dimensionalidade menor que a imagem de origem.

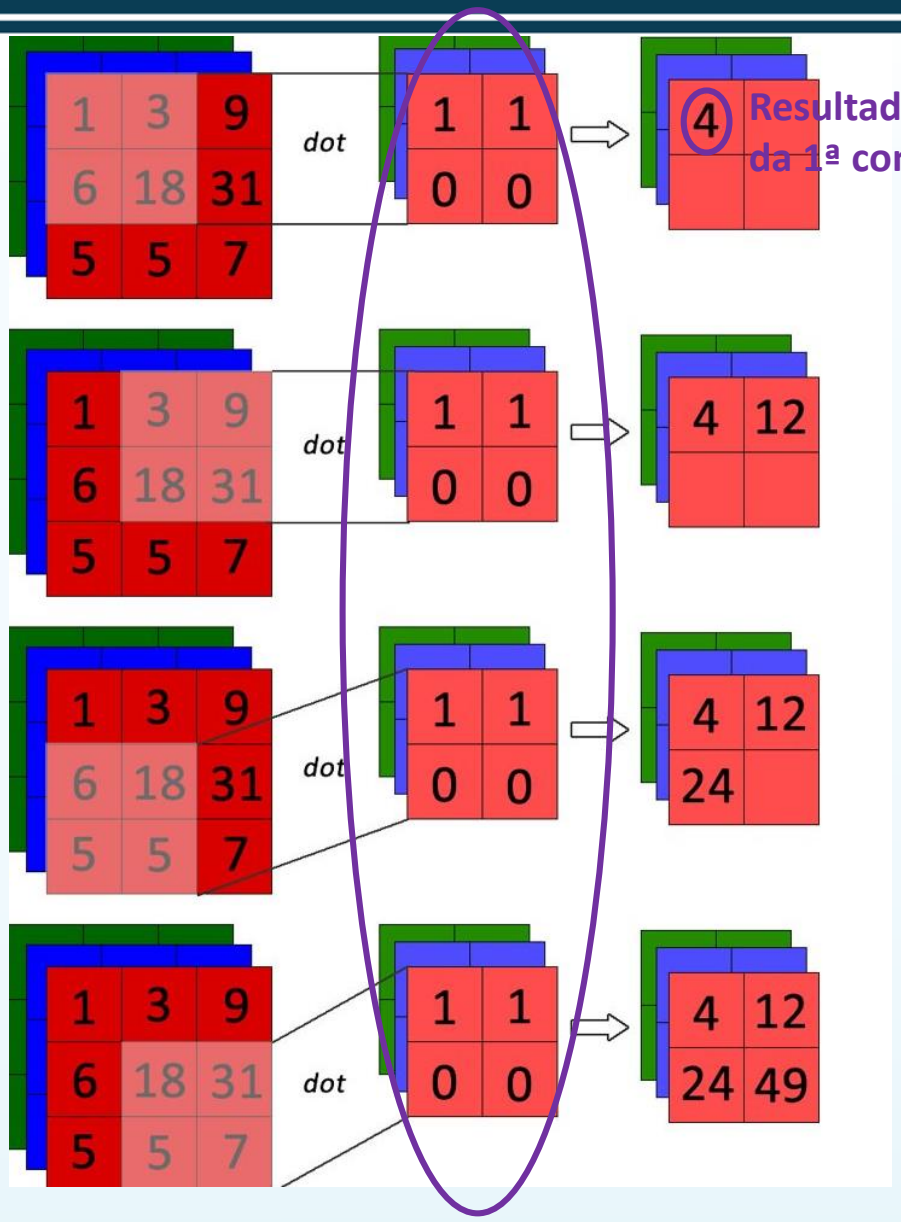
Camadas convolucionais

- Utilizam pequenos campos receptivos;
- Diminui assim a complexidade;
- Convolução: $y = x * w$:

$$y[i] = \sum_{k=0}^{k=m-1} x^p[i+m-k]w[k]$$



http://www.computacaointeligente.com.br/wp-content/uploads/2017/07/conv.jpg



Convolução

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

Imagem original

1	0	1
0	1	0
1	0	1

Filtro (*kernel*)

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

1a. convolução

3a.

1	1	1x1	0x0	0x1
0	1	1x0	1x1	0x0
0	0	1x1	1x0	1x1
0	0	1	1	0
0	1	1	0	0

4	3	4

6a.

1	1	1	0	0
0	1	1x1	1x0	0x1
0	0	1x0	1x1	1x0
0	0	1x1	1x0	0x1
0	1	1	0	0

4	3	4
2	4	3

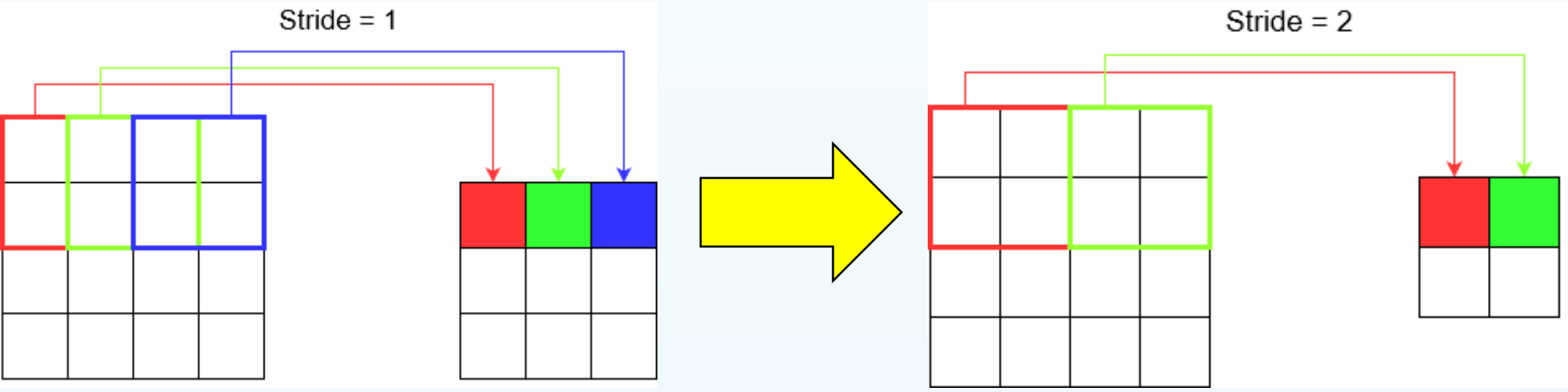
9a. - final

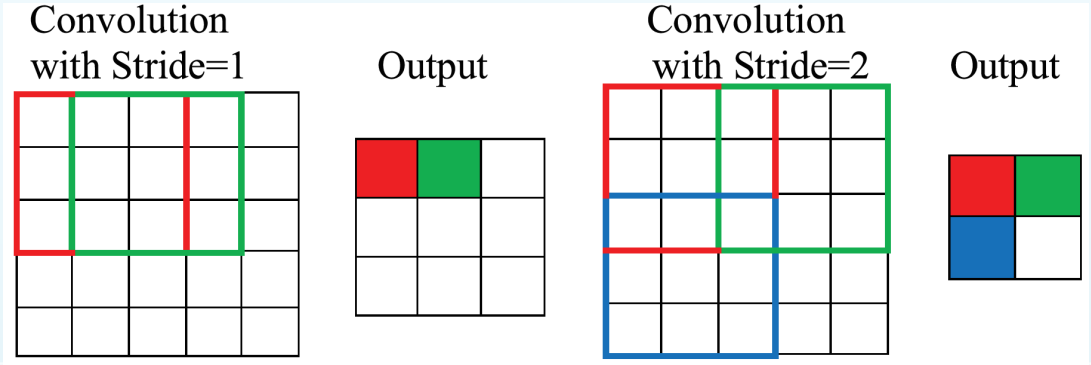
1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

Stride

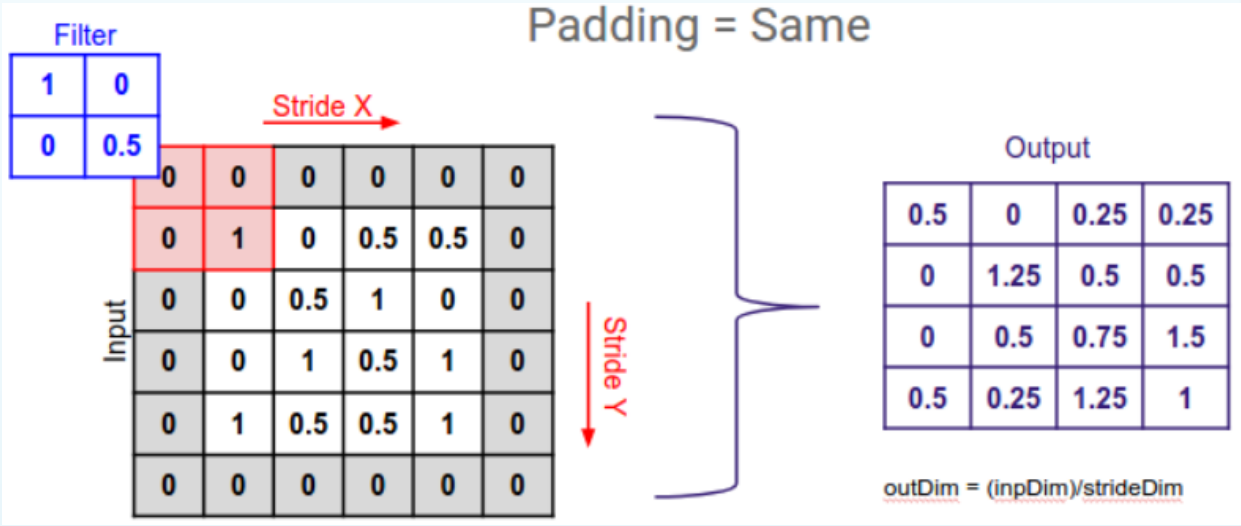
- Número de deslocamentos que o filtro fará no momento da convolução;
- Se possui o valor 1, o que significa que a transformação será aplicada em todos os *pixels* da imagem;





Padding

- Muitas convoluções podem impactar na assertividade da CNN se o tamanho da imagem for muito reduzido;
- Para contornar esse cenário, normalmente é utilizado o conceito de *Padding* – como acrescentar zeros ao redor da imagem;
- *Padding* é um processo em que alguns pixels são adicionados ao redor da imagem antes da operação de convolução, de forma a manter a dimensionalidade na imagem resultante durante a operação;
- Esse processo é utilizado porque essas imagens resultantes podem conter elementos que facilitam a identificação da classe alvo para a rede;
- Utilizado nas camadas iniciais.



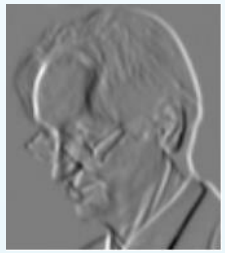
Kernels

- Filtros de convolução;
- Alguns filtros de convolução mais utilizadas com imagens são como ao lado:

$*$

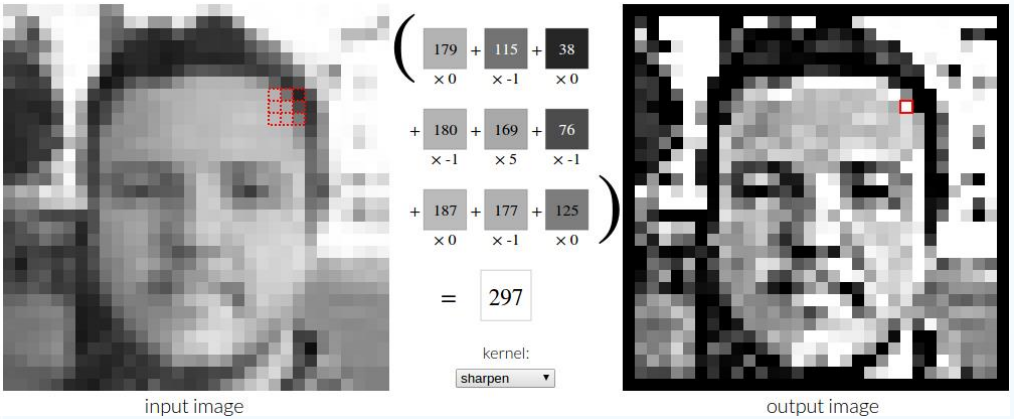
1	0	-1
2	0	-2
1	0	-1

$=$



Operation	Kernel ω	Image result g(x,y)
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	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Kernels



$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Below, for each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right. Hover over a pixel on either image to see how its value is computed.

1	1	1
1	1	1
1	1	1

Unweighted 3x3 smoothing kernel

0	1	0
1	4	1
0	1	0

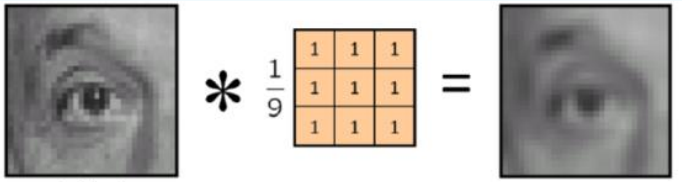
Weighted 3x3 smoothing kernel with Gaussian blur

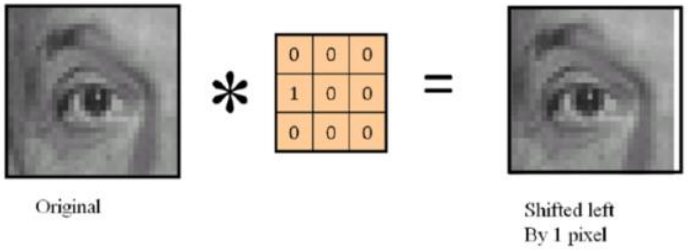
0	-1	0
-1	5	-1
0	-1	0

Kernel to make image sharper

-1	-1	-1
-1	9	-1
-1	-1	-1

Intensified sharper image



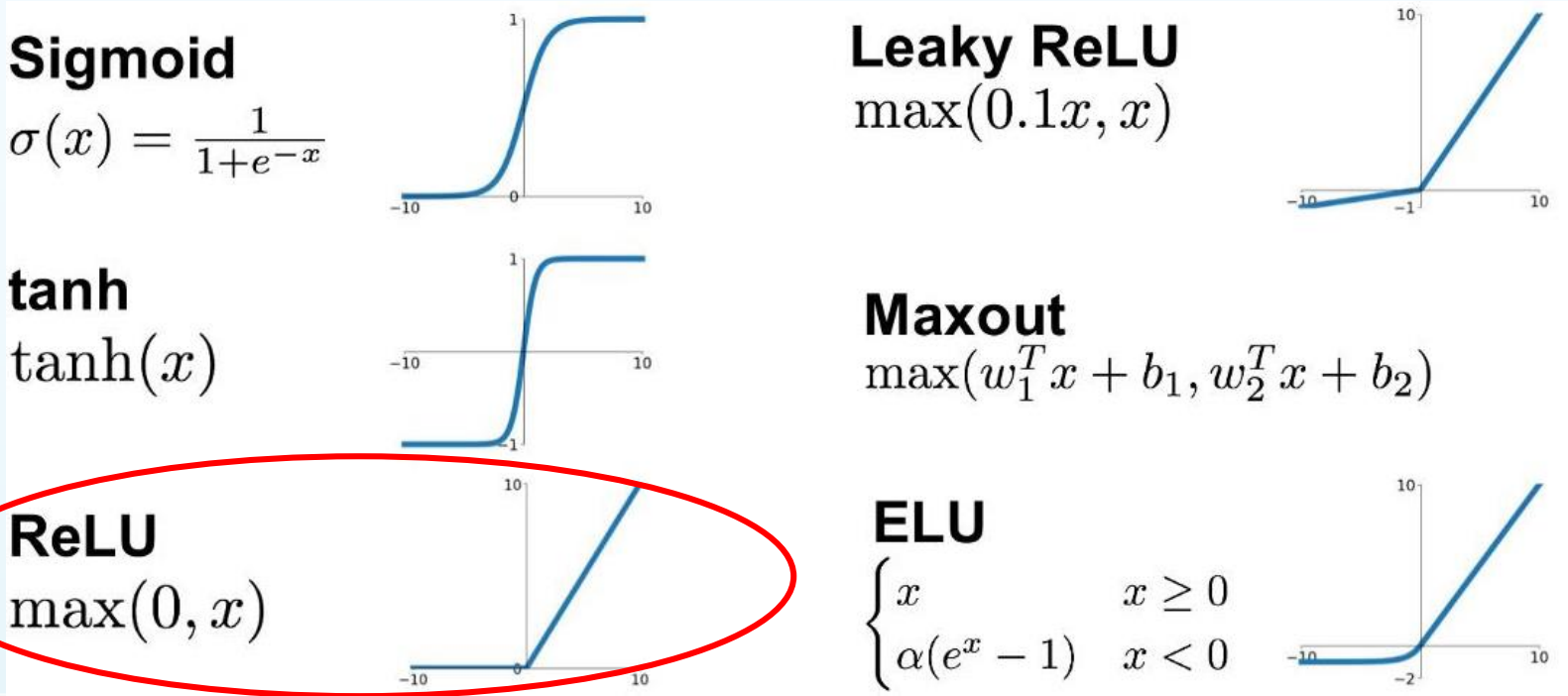


Gaussian Blur

Sharpened image

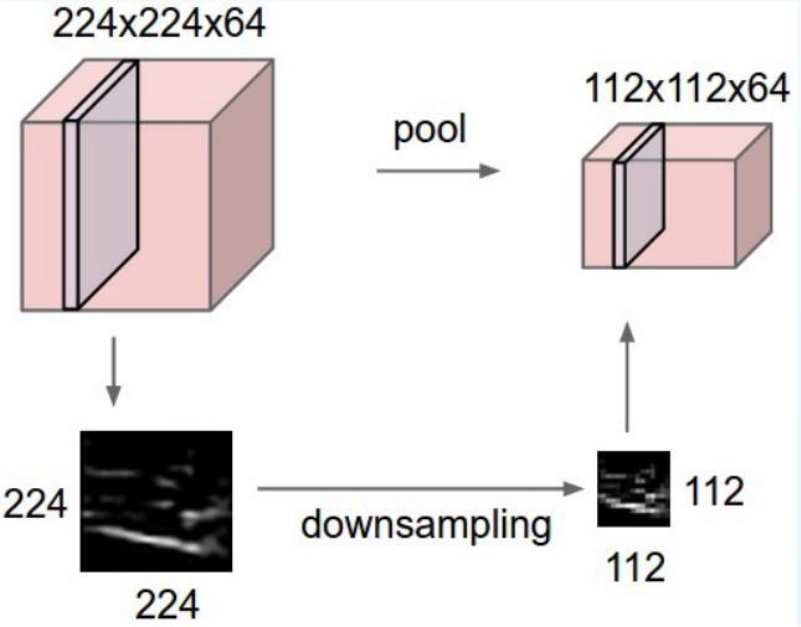
Camadas ReLu

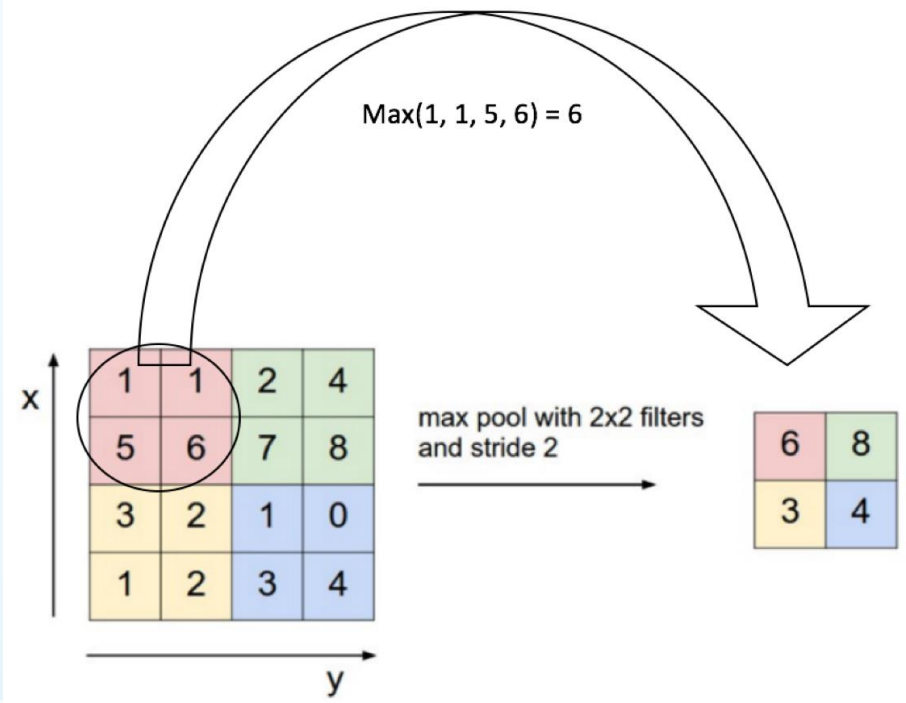
- Para adicionar não linearidade a rede, utiliza-se as funções de ativação;
- A mais utilizada no contexto de imagens é a função ReLU;
- ReLU é a abreviação para *Rectified Linear Units* (unidade linear retificada);
- Trata-se de uma função de ativação linear retificada, que só possui resposta não-nula no 1º quadrante, sendo dada por: $f(u)=max(0,u)$:



Pooling

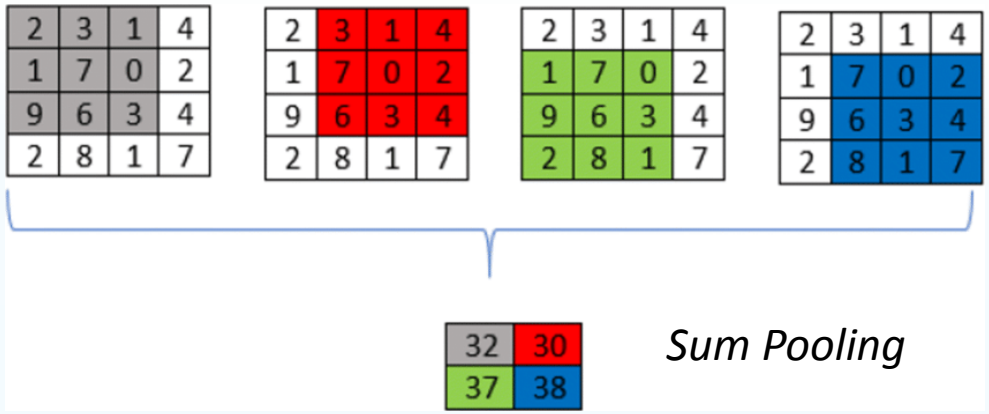
- *Pooling* é um processo de redução de amostragem (*downsamping*);
- Realiza, assim, redução da dimensionalidade/*features maps*;
- De forma grosseira, pode-se entender essa transformação como uma redução do tamanho da imagem;
- A principal motivação dessa operação no modelo é de diminuir sua variância a pequenas alterações e também de reduzir a quantidade de parâmetros treinados pela rede.

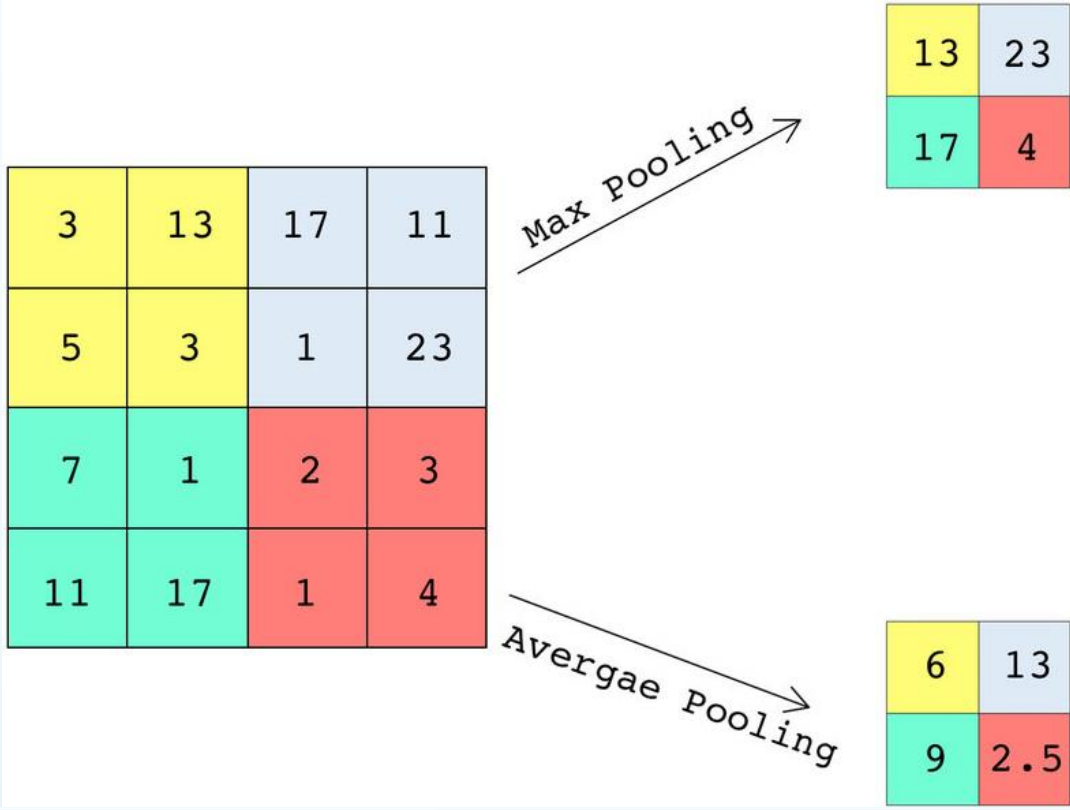




Pooling

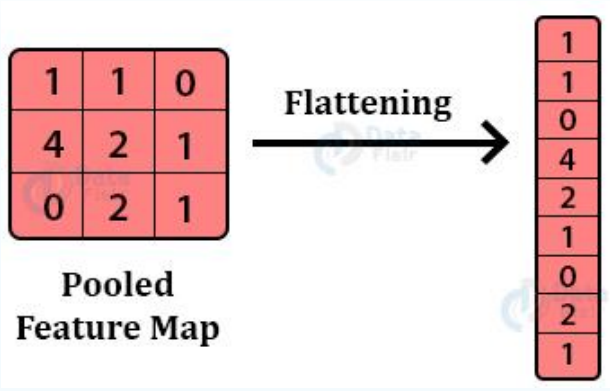
- Existem 3 operações básicas de *Pooling:*
  - ➢ *Max Pooling* (mais utilizada)
  - ➢ *Sum Pooling*;
  - ➢ *Avarage Pooling*;
- Todas elas seguem o mesmo princípio e só se diferem na forma como calculam o valor final;
- A operação de Max Pooling retira o maior elemento de determinada região da matriz considerando o tamanho do pool aplicado;
- Particiona a imagem de entrada em um conjunto de retângulos não-sobrepostos e, para cada sub-região, retorna o máximo.

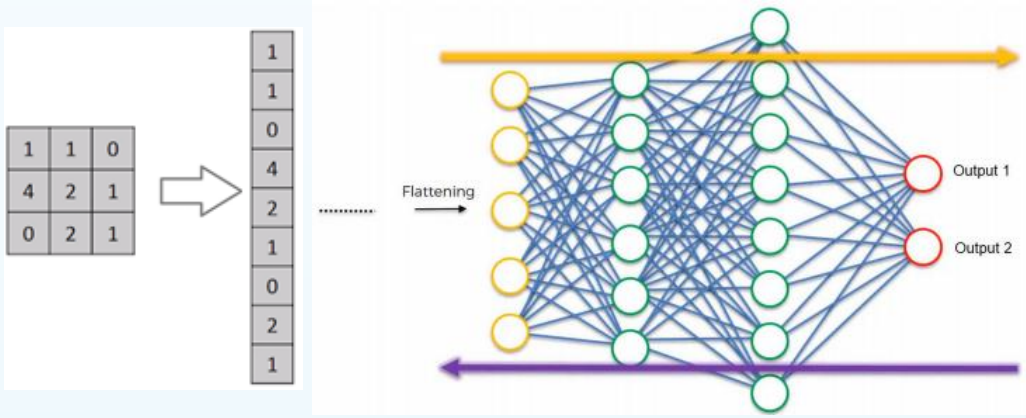


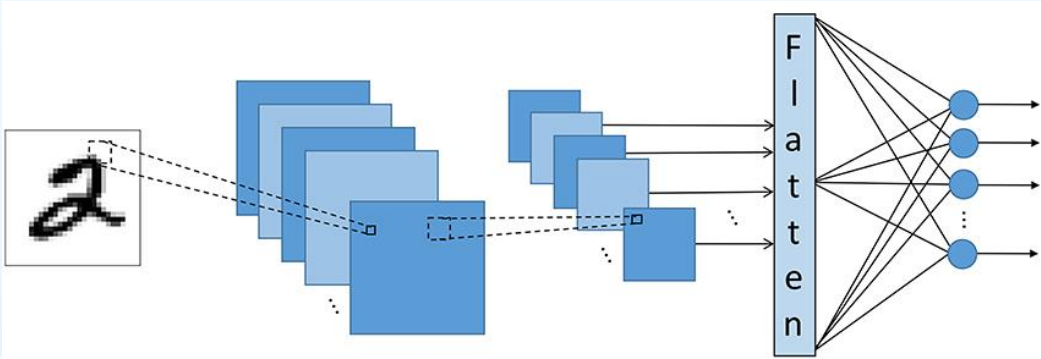


Camadas Totalmente Conectadas

- Inicialmente, antes de chegar até as camadas totalmente conectadas é necessário realizar a conversão das matrizes de características resultantes da camada de Pooling em um vetor coluna (*flattening*);
- Este será usado como entrada para a RNA clássica, semelhante a MLP;
- São as camadas finais da rede;
- O erro é calculado e propagado de volta para permitir o ajuste dos pesos e dos filtros;
- Os filtros procuram por determinadas características, as quais podem não estar corretas;
- Então os filtros são ajustados para procurarem por novas características.

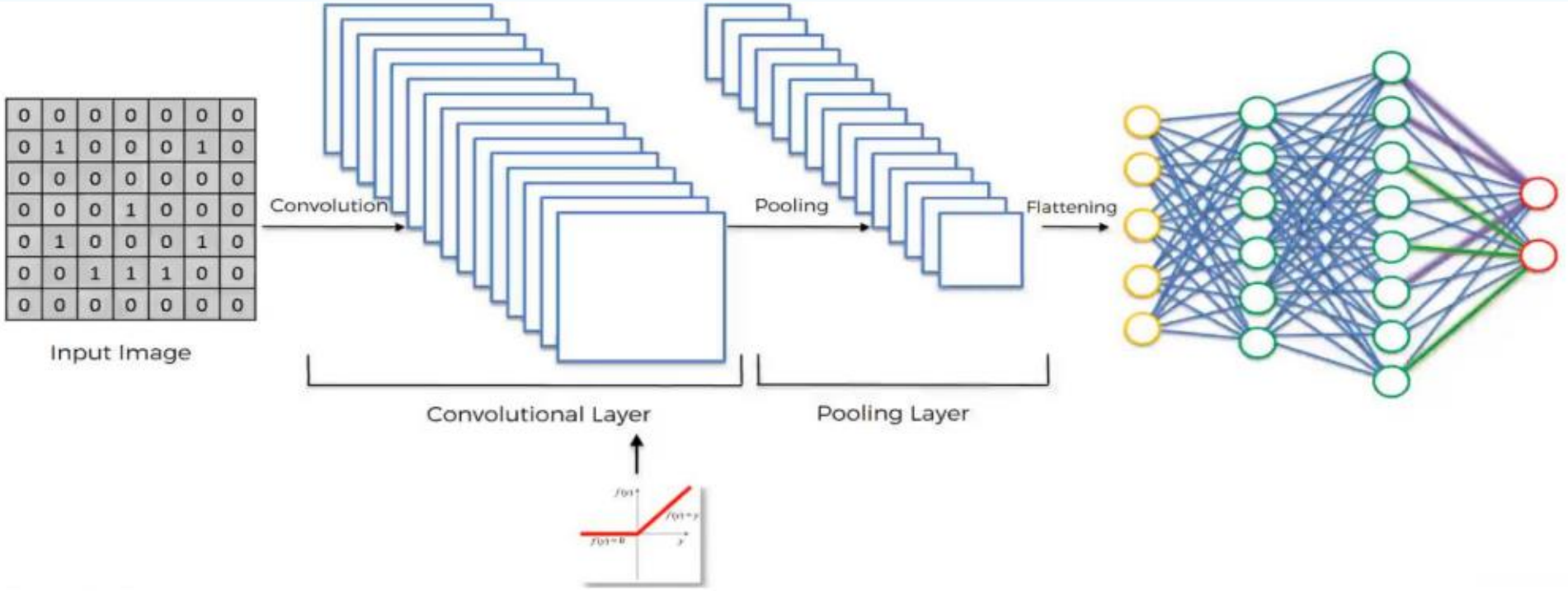






CNN

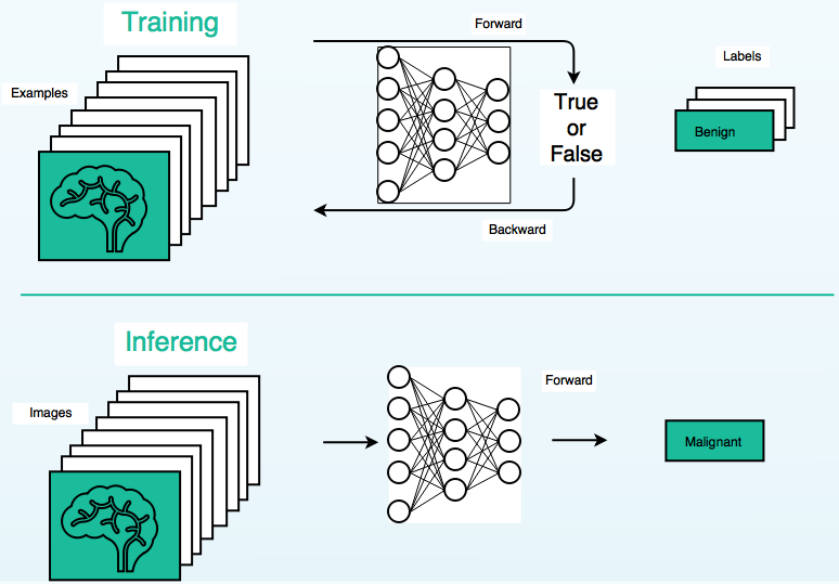
- Uma CNN genérica é mostrada desde sua imagem de entrada como a classe de saída, que no exemplo possui somente 2 classes.

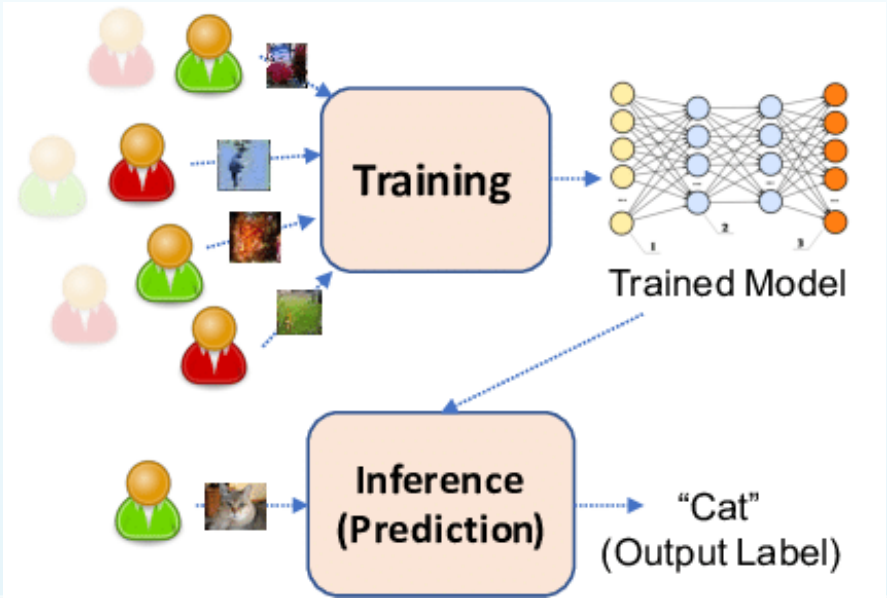


Arquitetura simplificada de uma Rede Neural Convolucional
Fonte: SuperDataScience (2017)

Treinamento

- Inicialmente, foram utilizadas técnicas não-supervisionadas para as primeiras camadas;
- Atualmente, utiliza-se apenas Backpropagation;
- Quando o conjunto de treinamento tem muitos dados, diminui-se o problema dos ótimos locais;
- Em geral, em vez de apresentar exemplos um a um, vários exemplos são apresentados para a atualização dos pesos;
- *Batch size*: número de exemplos apresentados de cada vez.





Treinamento

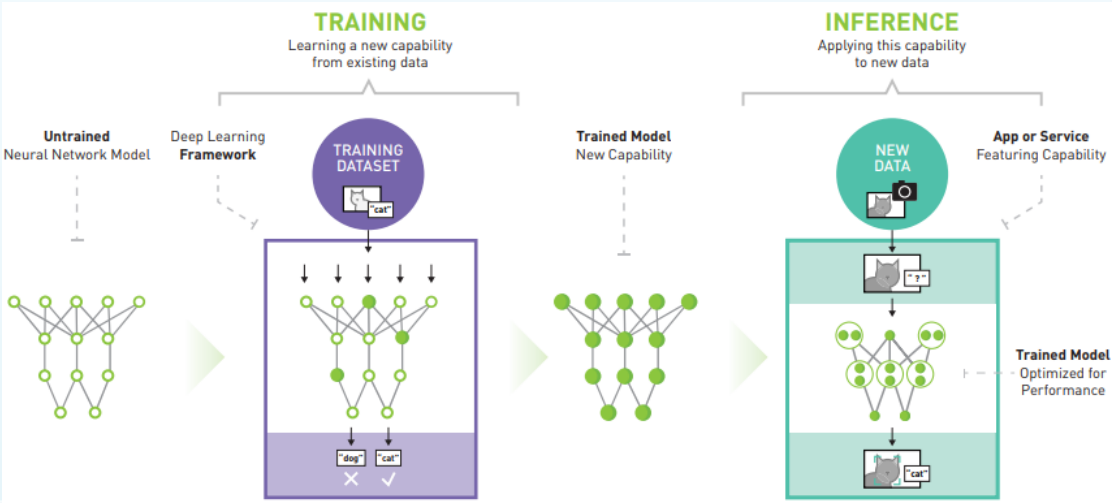
- Técnicas para minimizar o *overfitting* geralmente são adotadas;
- Aumento da base de dados (*data augmentation*):

➢No caso de imagens, é comum aumentar artificialmente a base considerando-se imagens alteradas, por exemplo, por ampliação, redução, rotação e deslocamento;

- Regularização de pesos:

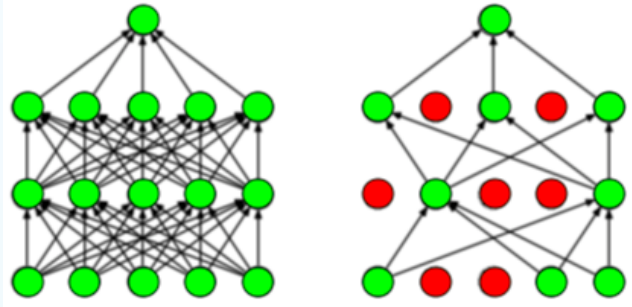
➢Visa reduzir problemas de mal-condicionamento:

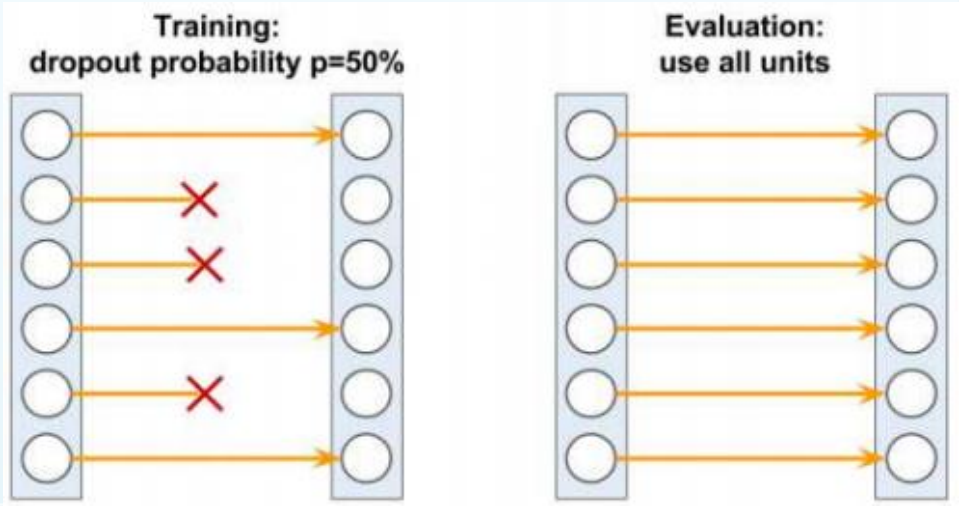
- *Dropout* (remoção de neurônios da rede): neurônios selecionados aleatoriamente são ignorados durante o treinamento;
- Existem vários parâmetros para definir o treinamento e a operação das mesmas.



Dropout

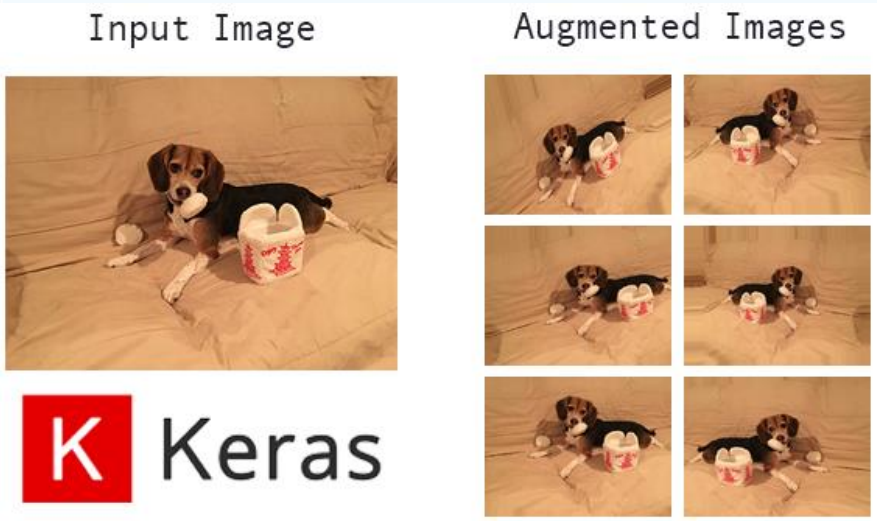
- *Dropout* é, em sua maioria das vezes, aplicado em neurônios da camada escondida ou camadas de mais alto nível;
- Durante o treinamento uma fração dos neurônios é aleatoriamente descartada baseada em uma probabilidade configurada pelo usuário;
- O efeito desse abandono aleatório força a rede a aprender uma representação redundante dos dados;
- A rede não pode depender de uma ativação de qualquer conjunto de neurônios escondidos, pois podem ser desativadas a qualquer momento durante o treinamento;
- São forçadas a aprender padrões mais gerais e robustos a partir dos dados;
- Contribui para impedir o overfitting.





Aumento da base de dados (data augmentation)

- Técnica que gera novos exemplares de dados de treinamento a fim de aumentar a generalidade do modelo;
- problemas do mundo real, normalmente não se tem um dataset adequado para treinar a CNN;
- A rede tende ao overfitting por não ter capacidade de generalizar adequadamente;



Aumento da base de dados (data augmentation)