

# Deep Learning

Prof. Jefferson T. Oliva  
[jeffersonoliva@utfpr.edu.br](mailto:jeffersonoliva@utfpr.edu.br)

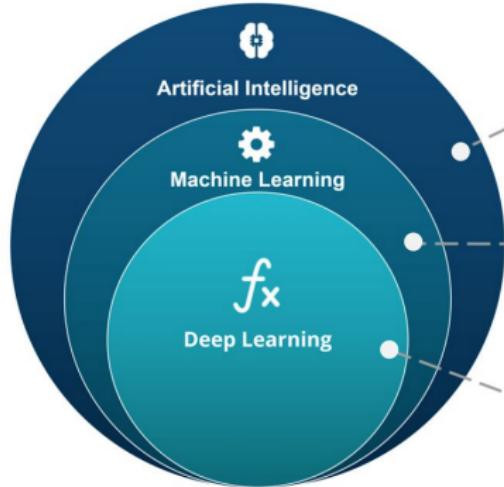
Processamento de Imagens  
Engenharia de Computação  
Departamento Acadêmico de Informática (Dainf)  
Universidade Tecnológica Federal do Paraná (UTFPR)  
Campus Pato Branco



This work is licensed under a Creative Commons "Attribution-ShareAlike 4.0 International" license.



- Redes Neurais Convolucionais
- Autoencoder Profundo

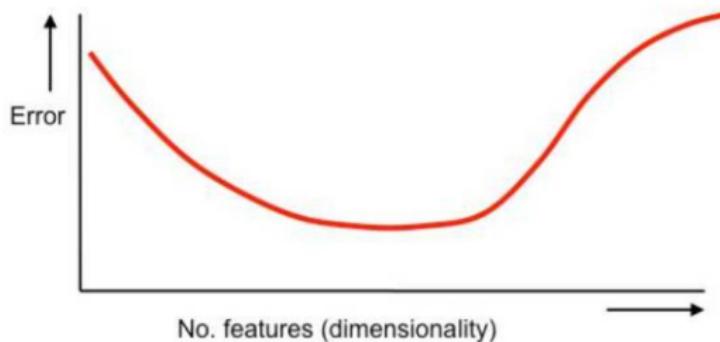


**ARTIFICIAL INTELLIGENCE**  
A technique which enables machines to mimic human behaviour

**MACHINE LEARNING**  
Subset of AI technique which use statistical methods to enable machines to improve with experience

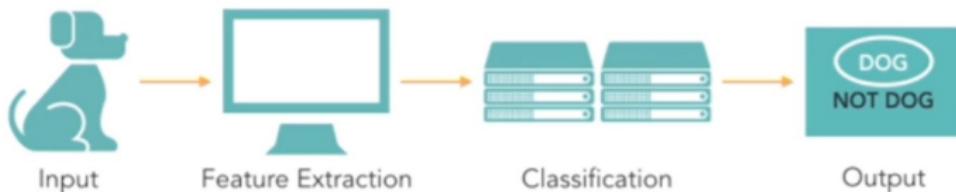
**DEEP LEARNING**  
Subset of ML which make the computation of multi-layer neural network feasible

- Redes Neurais Artificiais *feed-forward*
  - Maldição da dimensionalidade



- Uma das soluções encontradas para maldição de dimensionalidade é pré-processamento de dados:
  - Redução da dimensionalidade (às vezes por humanos)
  - Desafiante e altamente dependente da tarefa
- Se pensarmos no cérebro humano, não há indícios de que ele resolva esse problema dessa forma
- Aprender características automaticamente em múltiplos níveis de abstração permite ao sistema mapear funções complexas sem depender de características intermediárias inteligíveis aos humanos

## TRADITIONAL MACHINE LEARNING



## DEEP LEARNING

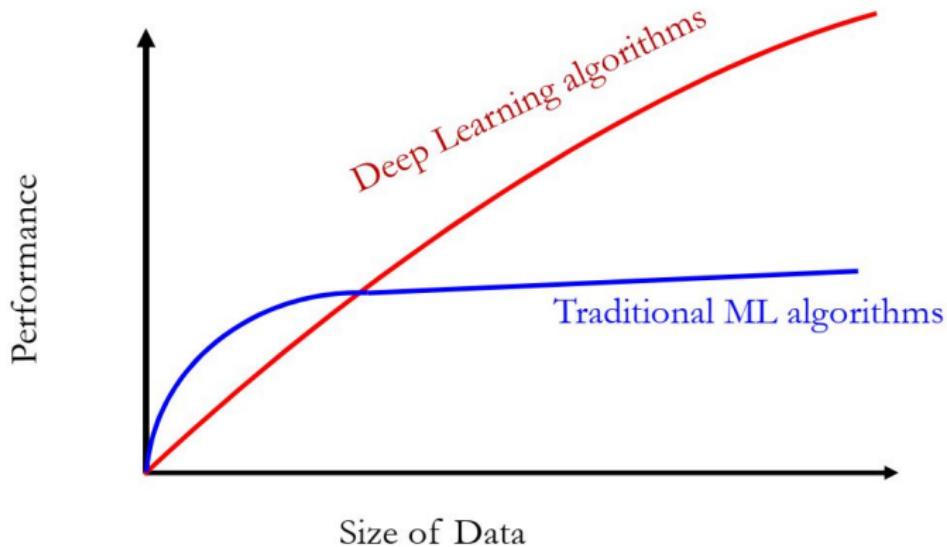


- Aprendizagem Profunda aprende em vários níveis de abstração
  - Representações mais abstratas extraem informações para classificadores ou preditores
  - Características intermediárias aprendidas podem ser compartilhadas entre diferentes tarefas
- Por décadas, diversos pesquisadores tentaram, sem sucesso, treinar redes neurais de múltiplas camadas profundas
  - Inicializadas com pesos aleatórios levavam a mínimos locais
- Hinton et al (2006) melhoraram o desempenho de uma rede neural profunda com etapa de pré-treinamento por aprendizagem não supervisionada, uma camada após outra a partir da primeira

Um tipo de modelo que tenta descobrir características genéricas dos dados:

- Múltiplas camadas constroem melhor espaço de características:
  - Primeira camada aprende as características de primeira ordem (por exemplo, bordas em imagem)
  - Segunda camada aprende características de maior ordem (por exemplo, combinação de bordas e outras características)
  - Camadas são treinadas por método não-supervisionado e as características alimentam uma camada supervisionada
    - A rede inteira é então ajustada de modo supervisionado

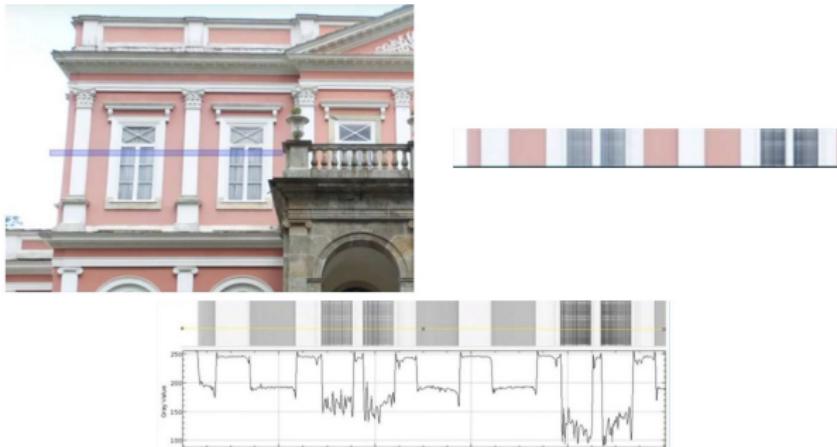
- Desempenho x tamanho de dados



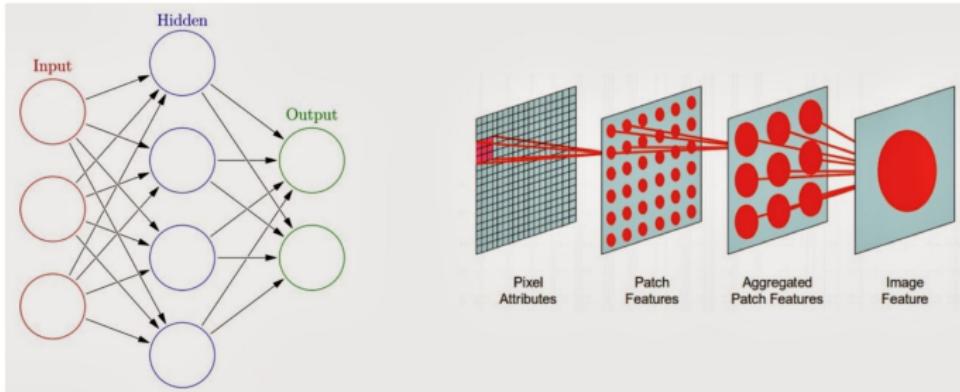
- Exemplos de modelos
  - Supervisionados
    - Redes neurais convolucionais
    - Redes neurais recorrentes
  - Não-supervisionados
    - *Deep belief nets*
    - *Stacked denoising autoencoders*
    - *Sparse autoencoders*

## Redes Neurais Convolucionais

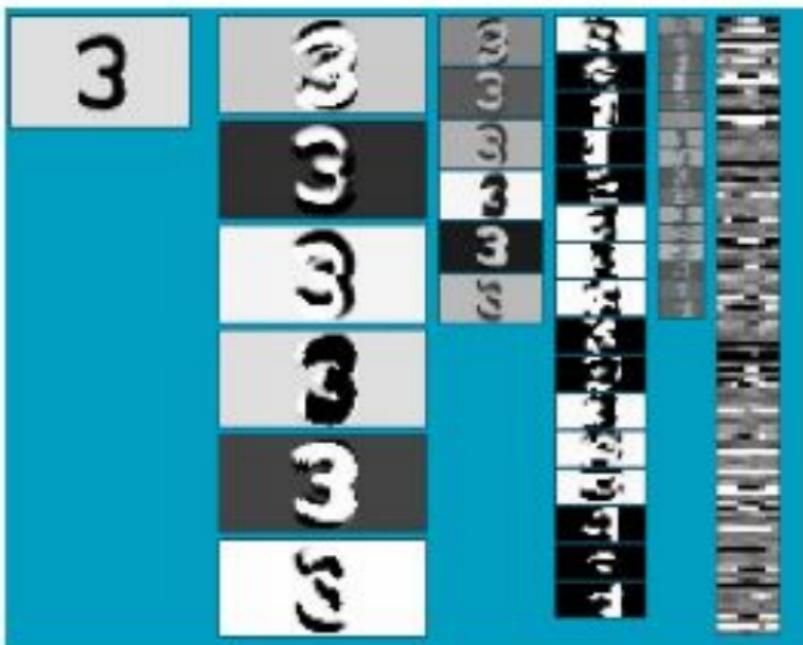
- Operações básicas empregadas no modelo:
  - Convolução
  - Inserção de não-linearidade
  - Sub-amostragem



- As camadas são treinadas de modo não-supervisionado de modo a descobrir características gerais do espaço de entrada
  - Aprende características direto da imagem
- As características finais alimentam uma camada supervisionada
  - A rede inteira é então ajustada de modo supervisionado
  - Em outras palavras, se ajusta às classes

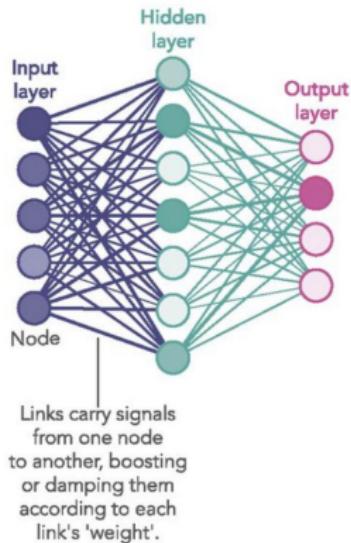


# Redes Neurais Convolucionais

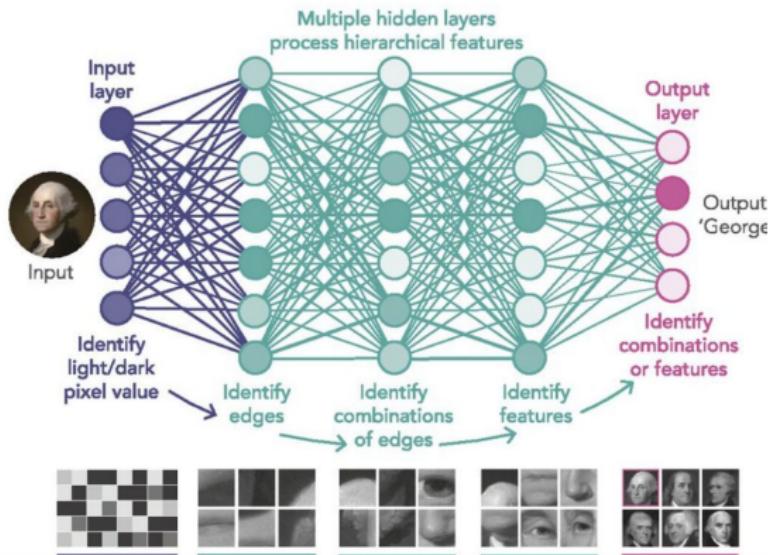


# Redes Neurais Convolucionais

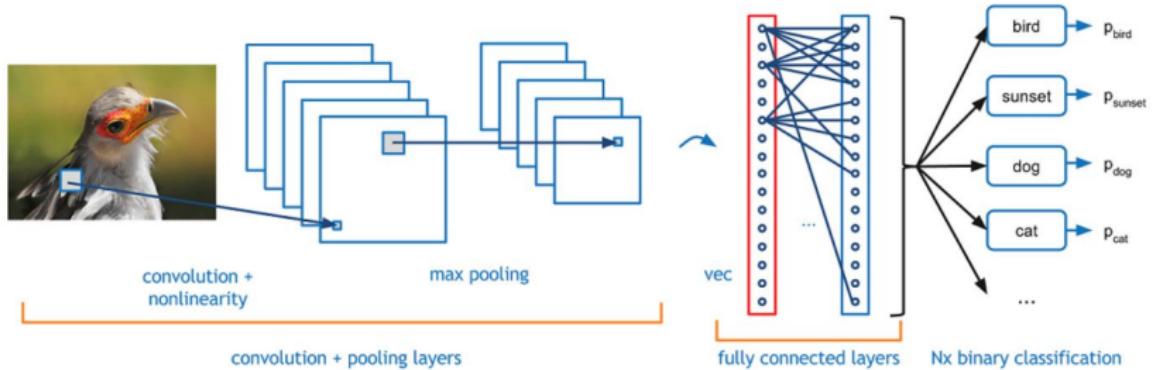
1980S-ERA NEURAL NETWORK



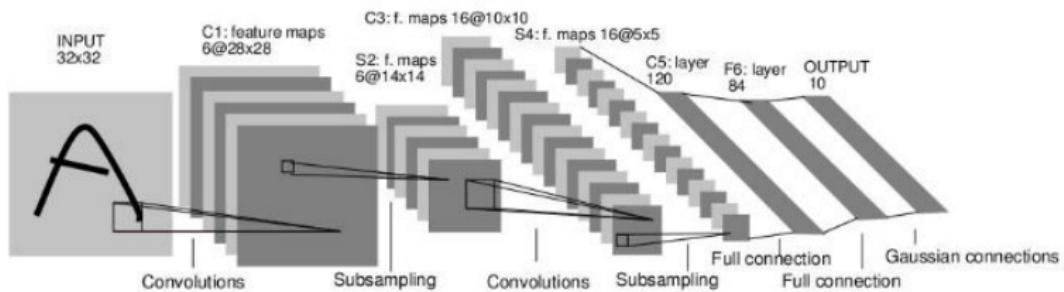
DEEP LEARNING NEURAL NETWORK



# Redes Neurais Convolucionais

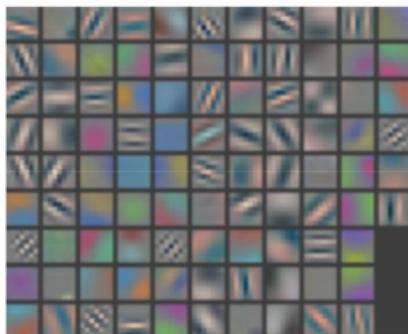


- Cada camada combina (*merge, suaviza*) trechos (*patches*) das camadas anteriores



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, [Gradient-based learning applied to document recognition](#), Proceedings of the IEEE 86(11): 2278–2324, 1998.

- Definição dos filtros
- Camadas próximas da entrada tende a ter menos filtros
- Os tamanhos dos filtros mais utilizados são encontrados na literatura e variam:
  - Para imagens  $28 \times 28$ , o tamanho é  $5 \times 5$  na primeira camada
  - Imagens maiores ( $256 \times 256$ ) tendem a utilizar filtros maiores:  $12 \times 12, 15 \times 15$

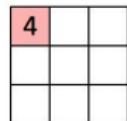


# Redes Neurais Convolucionais

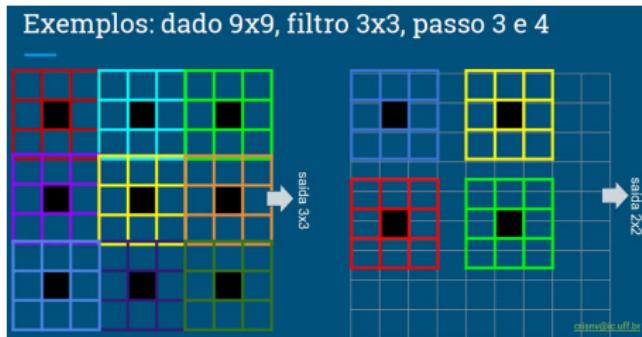
## Convolução

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

Image

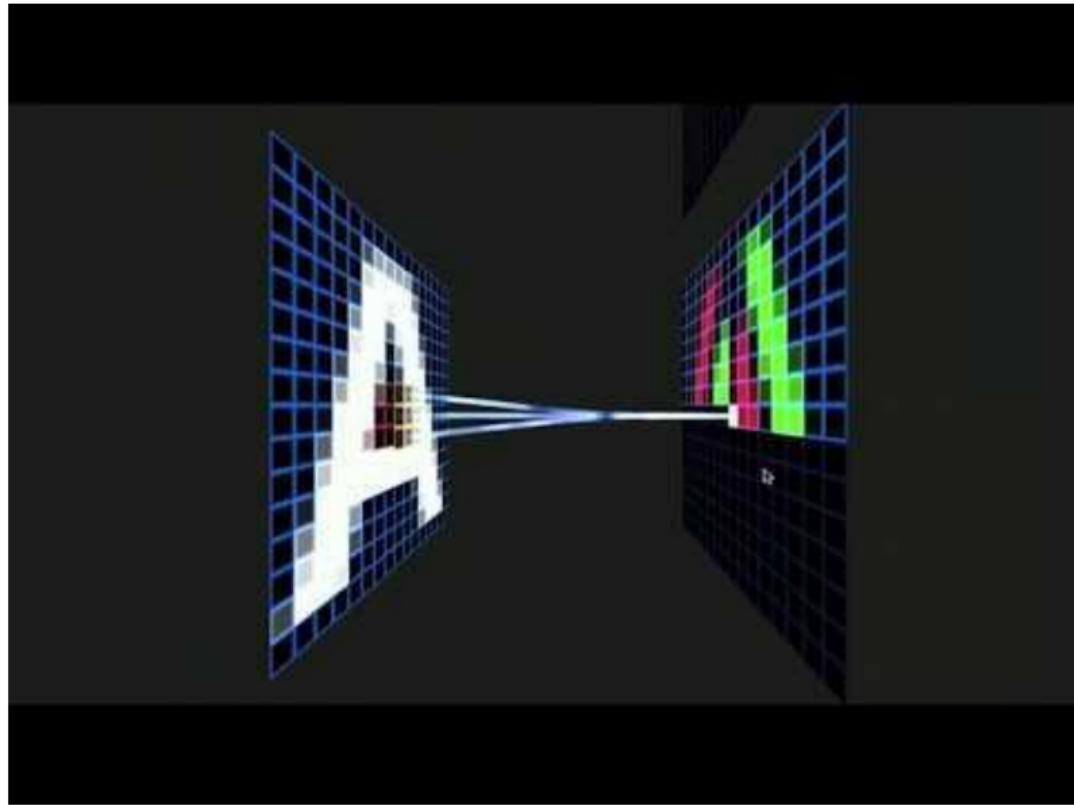


Convolved Feature



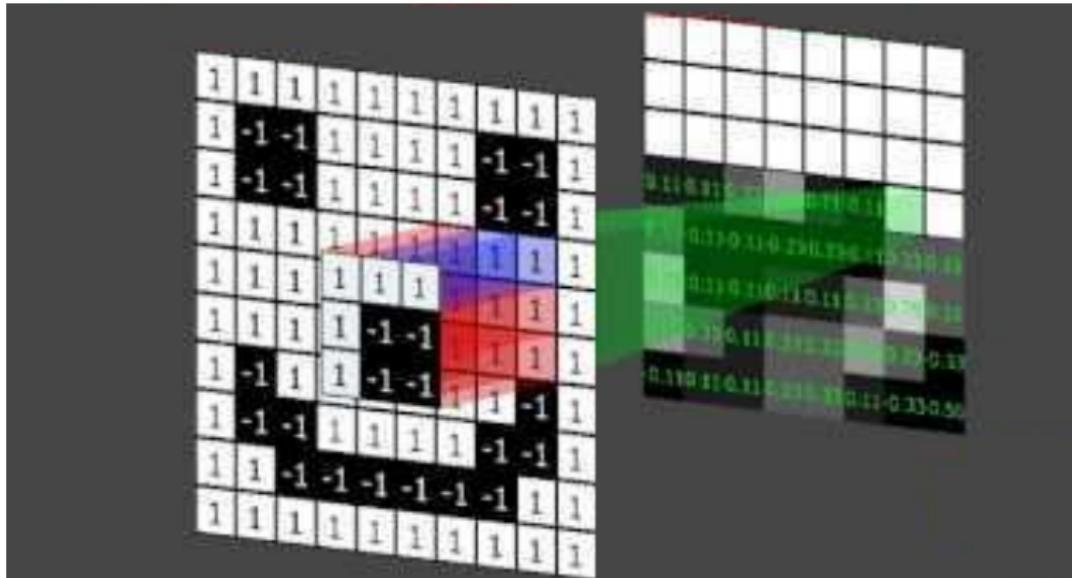
# Redes Neurais Convolucionais

## Convolução



# Redes Neurais Convolucionais

## Convolução

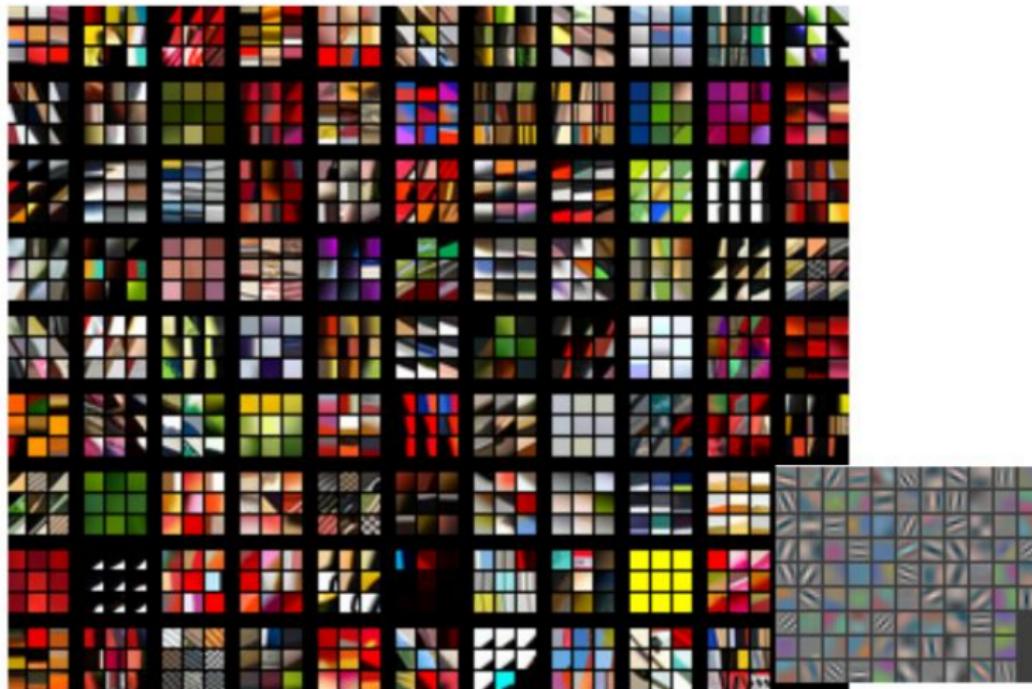


- Exemplos de filtros: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>

# Redes Neurais Convolucionais

## Convolução

- Filtro de convolução - camada 1



# Redes Neurais Convolucionais

## Convolução

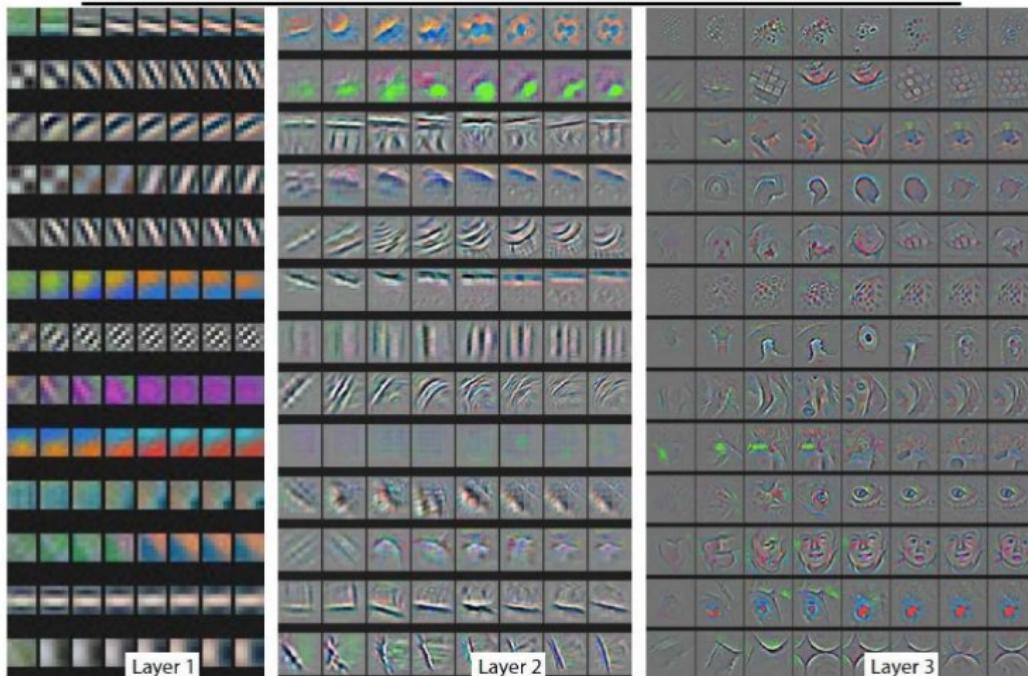
- Filtro de convolução - camada 2



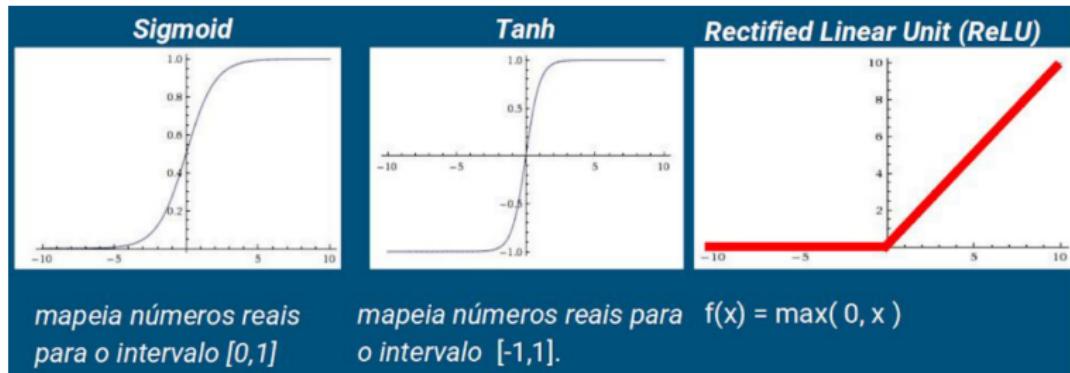
# Redes Neurais Convolucionais

## Convolução

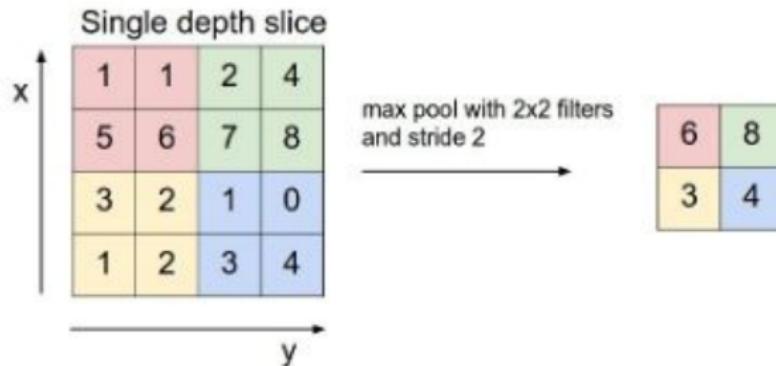
- Filtro de convolução - mais camadas



- Função de ativação
  - Feita por elemento: estabelecem as fronteiras para os valores de saída de um neurônio



- *Pooling* espacial
  - Comprime e suaviza os dados
  - Normalmente toma a média ou o valor máximo entre *patches* disjuntos

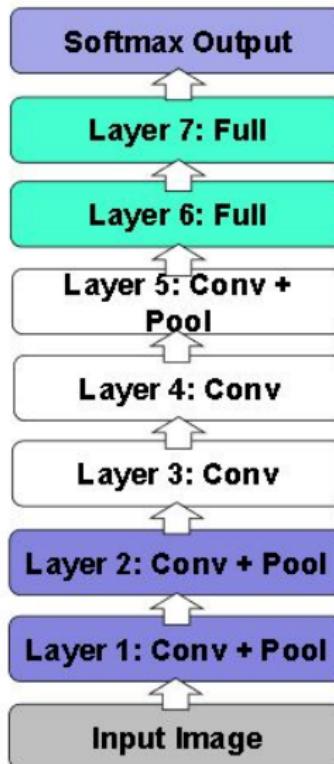


- Há diversos outros parâmetros a serem estudados que são definidos em uma rede de aprendizado profundo
  - Leitura recomendada: <https://www.deeplearningbook.org>

- Possui 8 camadas no total
- Treinamento na base ImageNet  
(<https://www.image-net.org/challenges/LSVRC/>)
  - Dataset com mais de 15 milhões de imagens rotuladas e 22.000 categorias
- Link do artigo: [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)

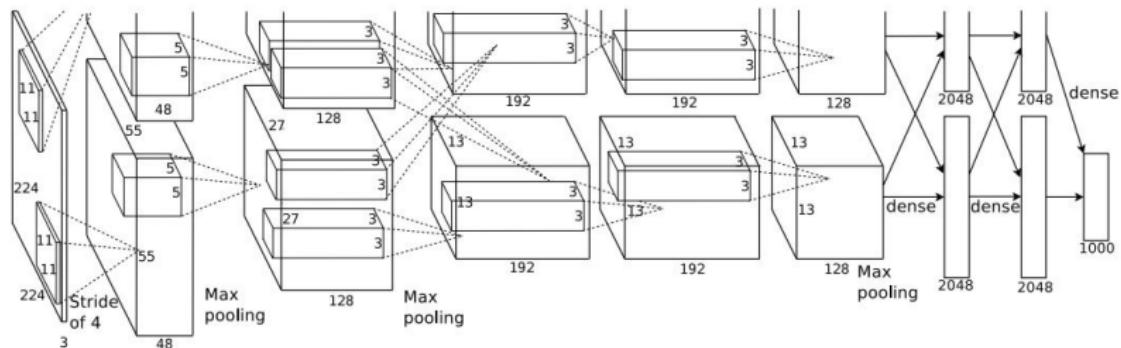
# Redes Neurais Convolucionais

## AlexNet



# Redes Neurais Convolucionais

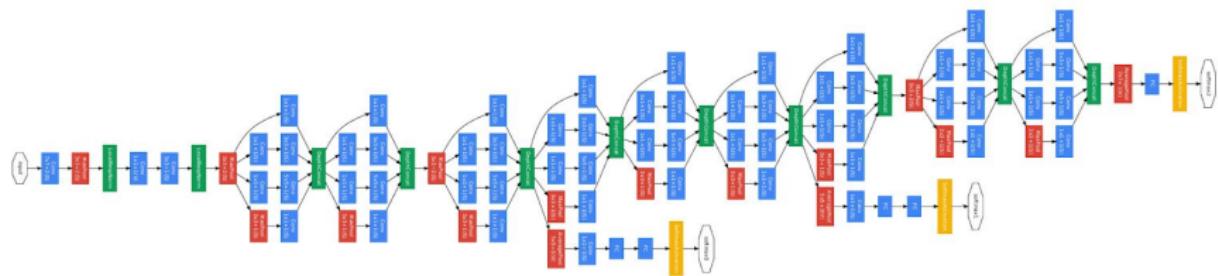
## AlexNet



# Redes Neurais Convolucionais

## Googlenet

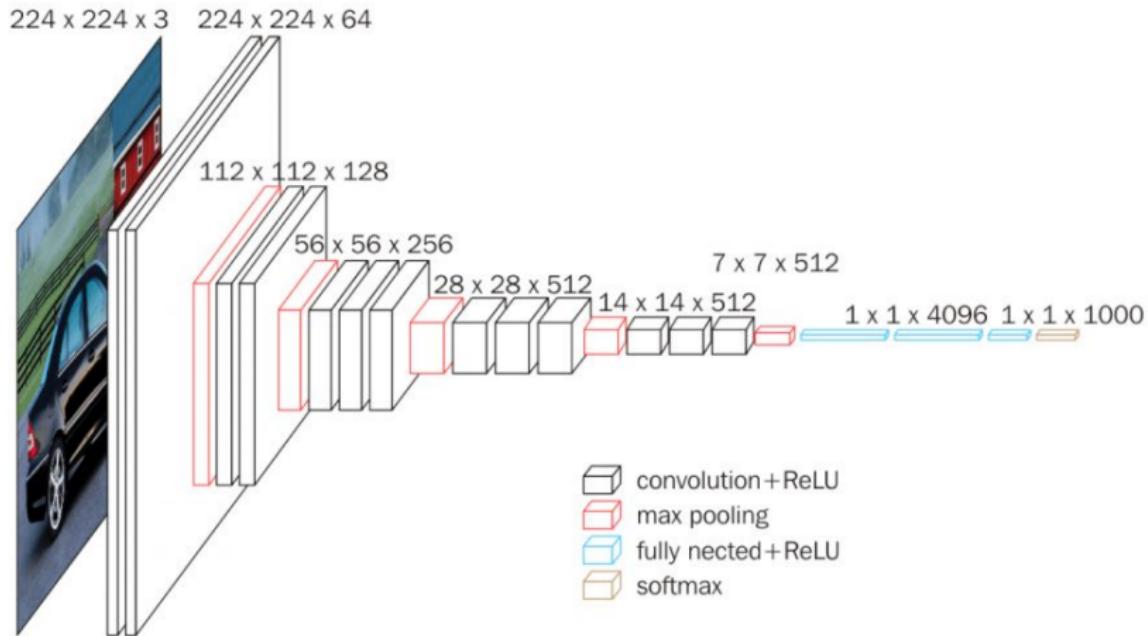
- <https://sites.google.com/site/aidysft/objectdetection/recent-list-items>



# Redes Neurais Convolucionais

## VGG16

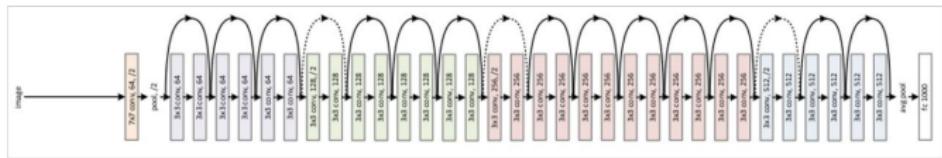
- <https://neurohive.io/en/popular-networks/vgg16/>



# Redes Neurais Convolucionais

## Resnet50

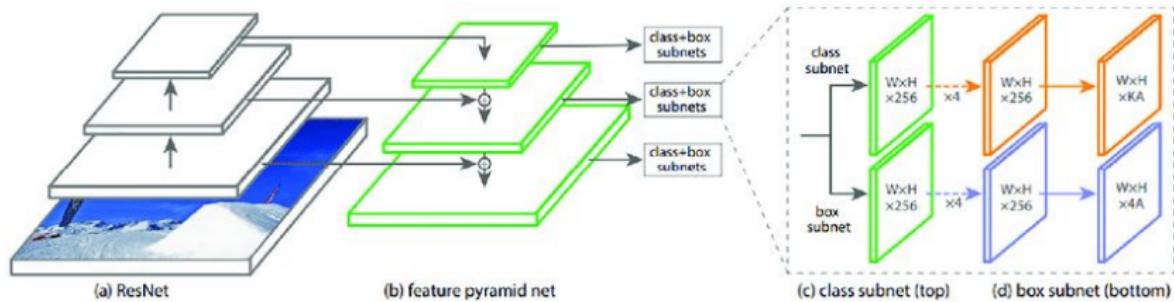
- <https://www.kaggle.com/keras/resnet50>



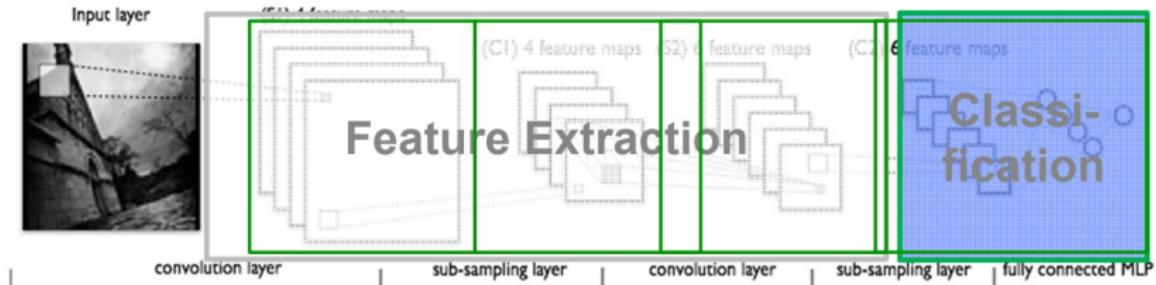
# Redes Neurais Convolucionais

## RetinaNet

- <https://www.kaggle.com/keras/resnet50>



- Arquitetura da rede neural convolucional
  - Camada convolucional + Não-linear (ReLU)
  - Camada de subamostragem
  - Camada convolucional + Não-linear (ReLU)
  - Linearização da camada + camadas totalmente conectadas de treinamento supervisionado



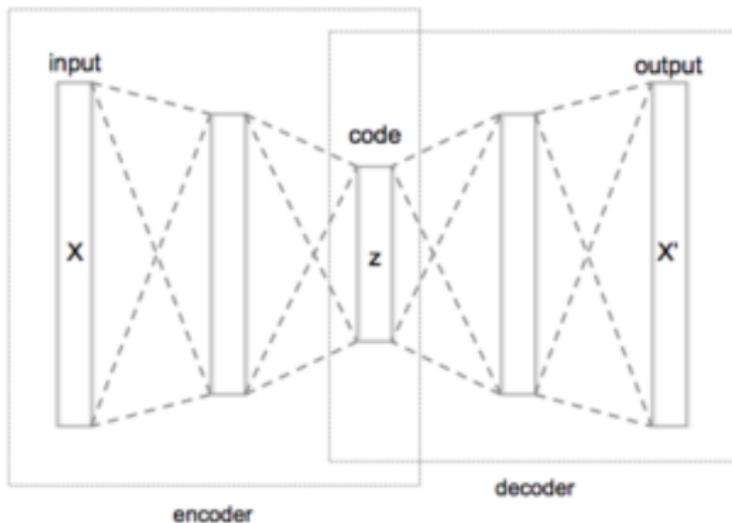
## Autoencoder Profundo

- Um tipo de modelo que tenta descobrir características genéricas dos dados:
  - Aprende a identificar funções através do aprendizado de sub-características
  - Compressão: pode usar novas características como um novo conjunto de treinamento
- Ao colocar uma camada escondida menor que a entrada, a rede é forçada a criar uma representação compacta do espaço de entrada

- O número de neurônios na camada oculta deve ser menor que a dimensão de entrada para evitar que a rede aprenda a solução trivial, ou seja, simplesmente copiar a entrada
- Com menos neurônios para codificar a entrada, a rede é forçada a aprender uma representação compacta do espaço de entrada
- Cada camada abstrai um pouco mais a informação da camada anterior, criando representações de alto nível
  - Isso facilita o trabalho de camadas superiores, pois elas passam a trabalhar sobre conceitos de mais alto nível

- Cada camada abstrai um pouco mais a informação da camada anterior, criando representações de alto nível
  - Isso facilita o trabalho de camadas superiores, pois elas passam a trabalhar sobre conceitos de mais alto nível
    - ➊ Treinar a primeira camada utilizando dados sem rótulos
    - ➋ Fixar os parâmetros da primeira camada e usar a sua saída, como entrada não-supervisionada, para treinar a segunda camada
    - ➌ Repetir 1 e 2 de acordo com o número de camadas desejada
    - ➍ Usar a saída da camada final como entrada para uma camada (modelo) supervisionada e treinar de modo supervisionado (mantendo os pesos das camadas anteriores fixos)
    - ➎ Liberar todos os pesos e realizar ajuste fino da rede como um todo utilizando uma abordagem supervisionada

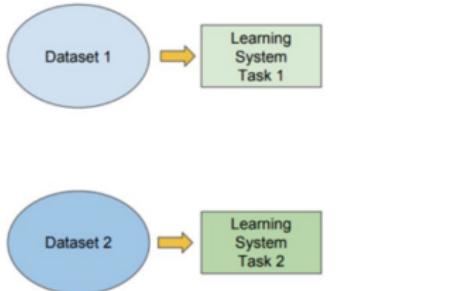
# Autoencoder Profundo



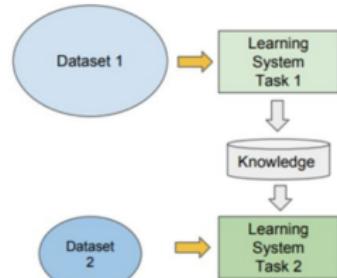
- Sub-características criadas em arquiteturas profundas podem ser compartilhadas entre múltiplas tarefas: transferência de aprendizado

## Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Traditional Learning vs Transfer Learning

# Autoencoder Profundo

## • Ferramentas

Software	Creator	Platform	Written in	Interface
<a href="#">Apache MXNet</a>	Apache Software Foundation	Linux, macOS, Windows, AWS, Android, iOS, JavaScript	<a href="#">Small C++ core library</a>	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl
<a href="#">Apache SINGA</a>	<a href="#">Apache Incubator</a>	Linux, macOS, Windows	<a href="#">C++</a>	Python, C++, Java
<a href="#">Caffe</a>	Berkeley Vision and Learning Center	Linux, macOS, Windows	<a href="#">C++</a>	Python, MATLAB, C++
<a href="#">DeepLearning4J</a>	<a href="#">Skymind engineering team; DeepLearning4J community; originally Adam Gibson</a>	Linux, macOS, Windows, Android (Cross-platform)	C++, Java	Java, Scala, Clojure, Python(Keras), Kotlin
<a href="#">Intel Data Analytics Acceleration Library</a>	Intel	Linux, macOS, Windows on Intel CPU	C++, Python, Java	C++, Python, Java[10]
<a href="#">Keras</a>	François Chollet	Linux, macOS, Windows	<a href="#">Python</a>	Python, R
<a href="#">PyTorch</a>	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	Linux, macOS	Python, C, CUDA	<a href="#">Python</a>
<a href="#">TensorFlow</a>	<a href="#">Google Brain team</a>	Linux, macOS, Windows, Android	C++, Python, CUDA	Python (Keras), C/C++, Java, Go, R, Julia
<a href="#">Theano</a>	<a href="#">Université de Montréal</a>	Cross-platform	<a href="#">Python</a>	Python (Keras)
<a href="#">Torch</a>	Ronan Collobert, Koray Kavukcuoglu, Clement Farabet	Linux, macOS, Windows, Android, iOS	C, Lua	Lua, LuaJIT, C, utility library for C++/OpenCL

# Referências I



Data Science Academy.

Deep Learning Book, 2022.

Disponível em:

<<https://www.deeplearningbook.com.br/>>.

Acesso em: 02 de Dezembro de 2024.



Goodfellow, I.

Deep learning. MIT Press, 2016.

Disponível em <<https://www.deeplearningbook.org/>>.

Acesso em: 02 de Dezembro de 2024.



LeCun Y.; Bottou L.; Bengio Y.; Haffner P.

Gradient-based Learning applied to Document Recognition.

Proceedings of the IEEE 86(11): 2278-2324, 1998.



Wiggers, K. L.

Notas de aula – Processamento de Imagens: deep learning.

UTFPR. Pato Branco, PR, 2024.