

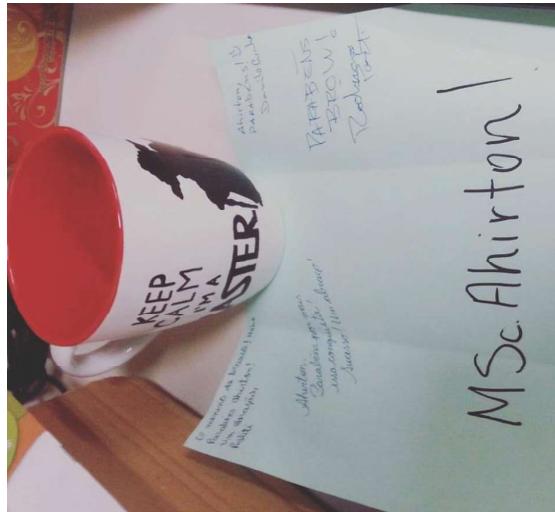
# Deep Learning

## DNNs e CNNs com Tensorflow e Keras

MSc. José Ahirton Batista Lopes Filho



# About Me



# Ferramentais

Keras



TensorFlow

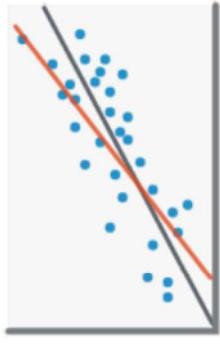


Notebook

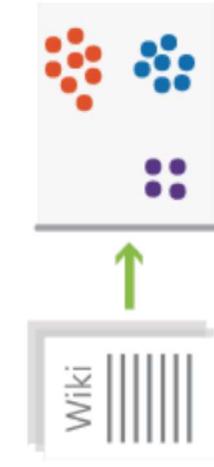
# Problemas e Técnicas Canônicas em Ciência de Dados



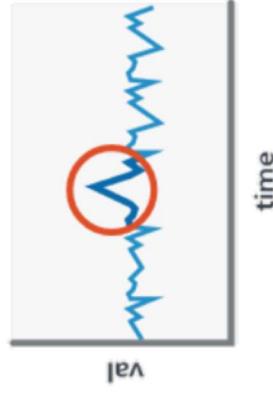
Classification  
(supervised – predictive)



Regression  
(supervised – predictive)



Clustering  
(unsupervised – descriptive)



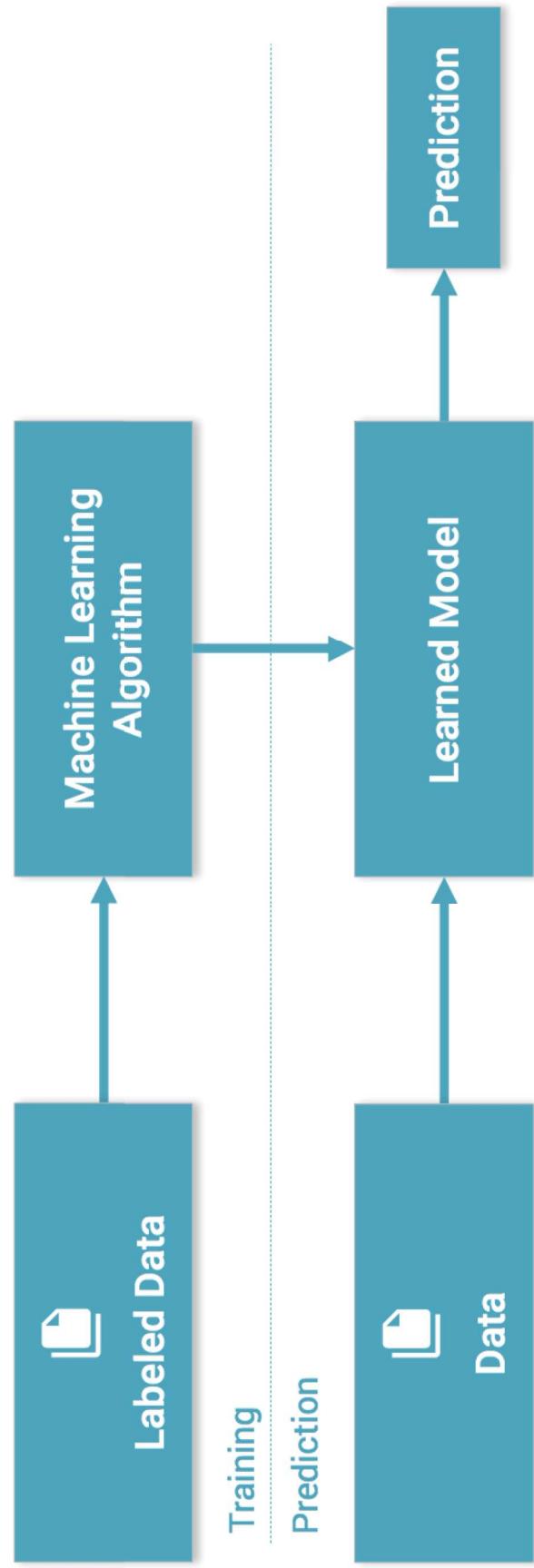
Anomaly Detection  
(unsupervised – descriptive)

# Fluxo em Inteligência Artificial



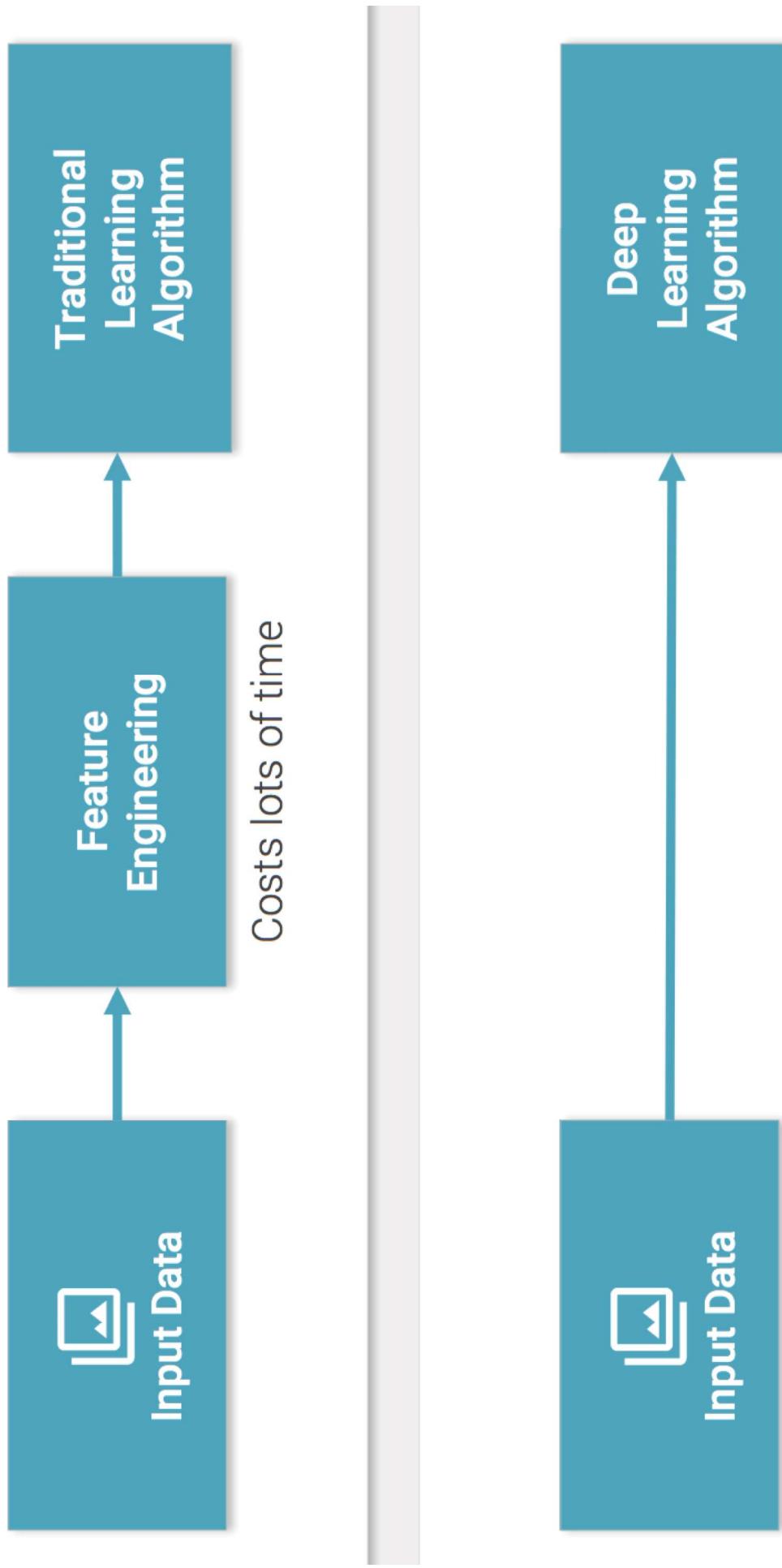
# Fluxo em Machine Learning

Machine Learning is a type of Artificial Intelligence that provides computers with the ability to **learn without being explicitly programmed**.

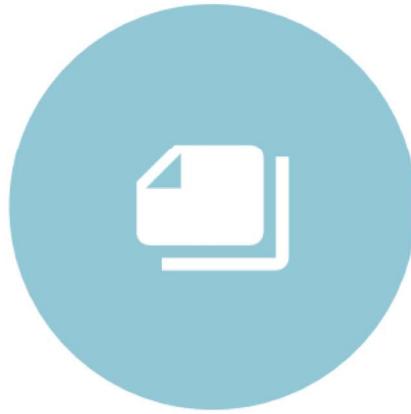


Provides **various techniques** that can learn from and make predictions on data

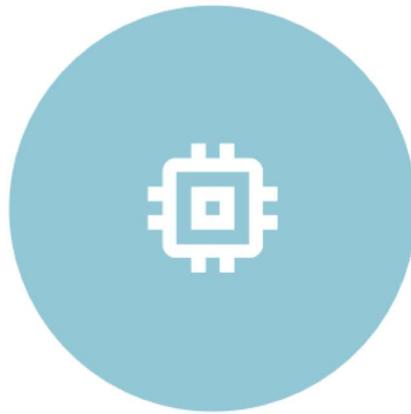
# ML vs. Deep Learning



# Porque Deep Learning?



Big Data  
(Digitalization)

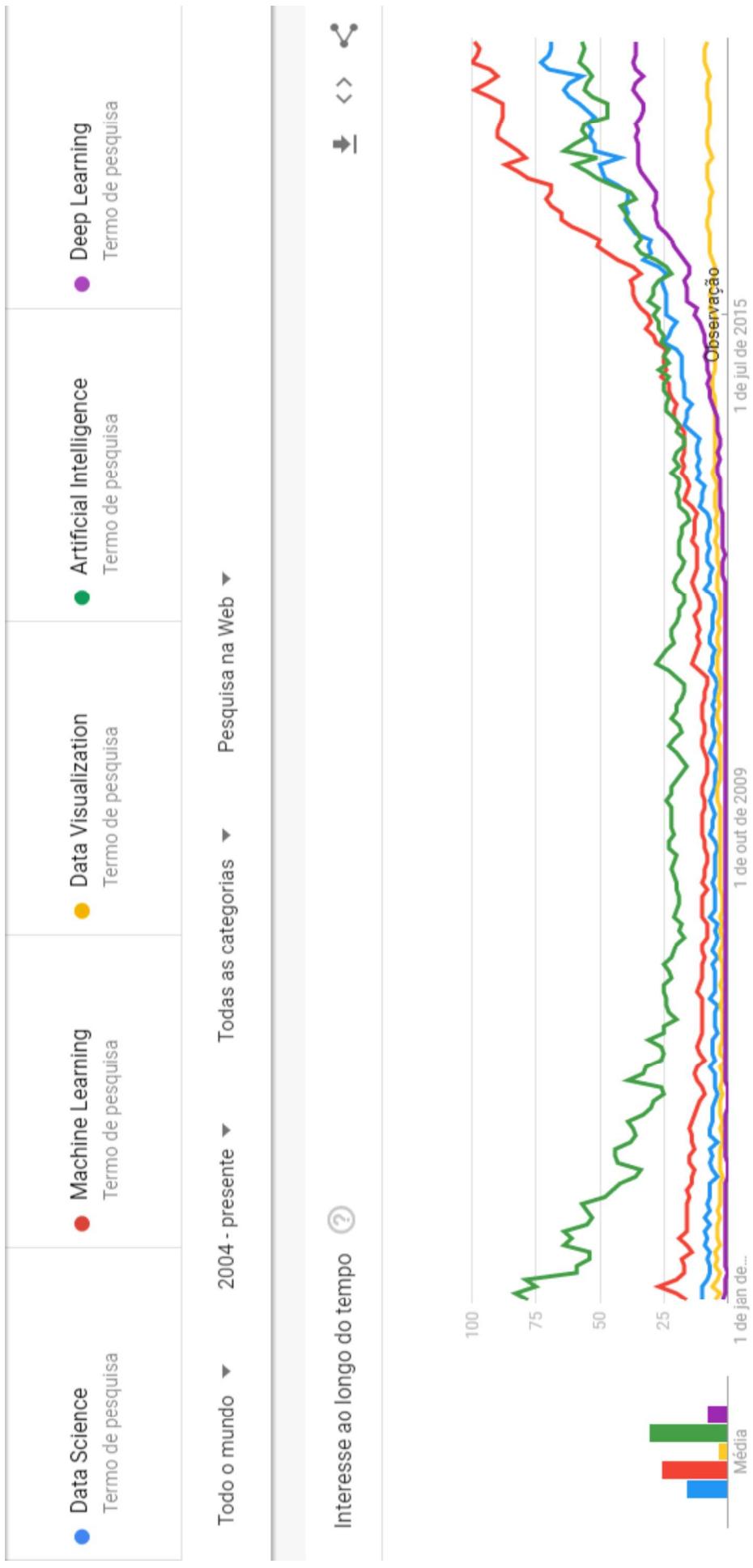


Computation  
(Moore's Law, GPUs)



Algorithmic  
Progress

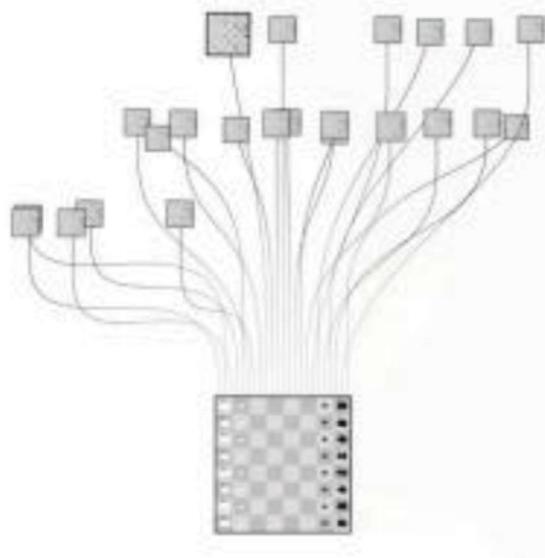
# Deep Learning



# Deep Learning

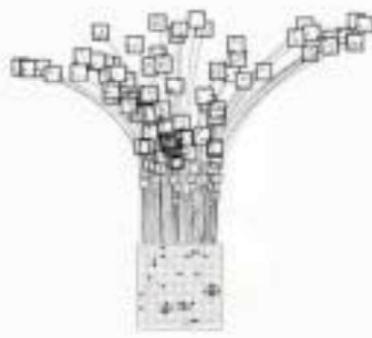


# Deep Learning



**Chess:**  $10^{47}$

Deep Blue, Feb 10, 1996



**Go:**  $10^{170}$

AlphaGo, March, 2016

# Pesquisadores em Deep Learning



Geoffrey Hinton: University of Toronto & Google



Yann LeCun: New York University & Facebook



Andrew Ng: Stanford & Baidu



Yoshua Bengio: University of Montreal



Jürgen Schmidhuber: Swiss AI Lab & NNAISENSE

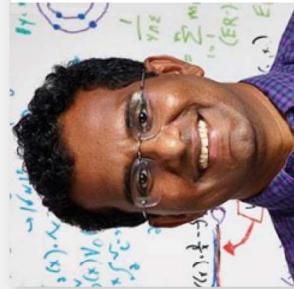
# Porque Deep Learning?



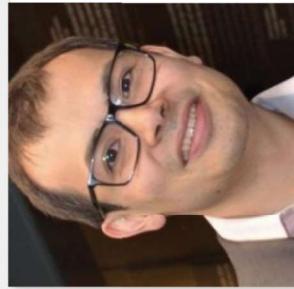
I have worked all my life in Machine Learning, and I've never seen one **algorithm knock over benchmarks like Deep Learning**  
– Andrew Ng (Stanford & Baidu)



Deep Learning is an algorithm which has **no theoretical limitations of what it can learn**; the more data you give and the more computational time you provide, the better it is – Geoffrey Hinton (Google)



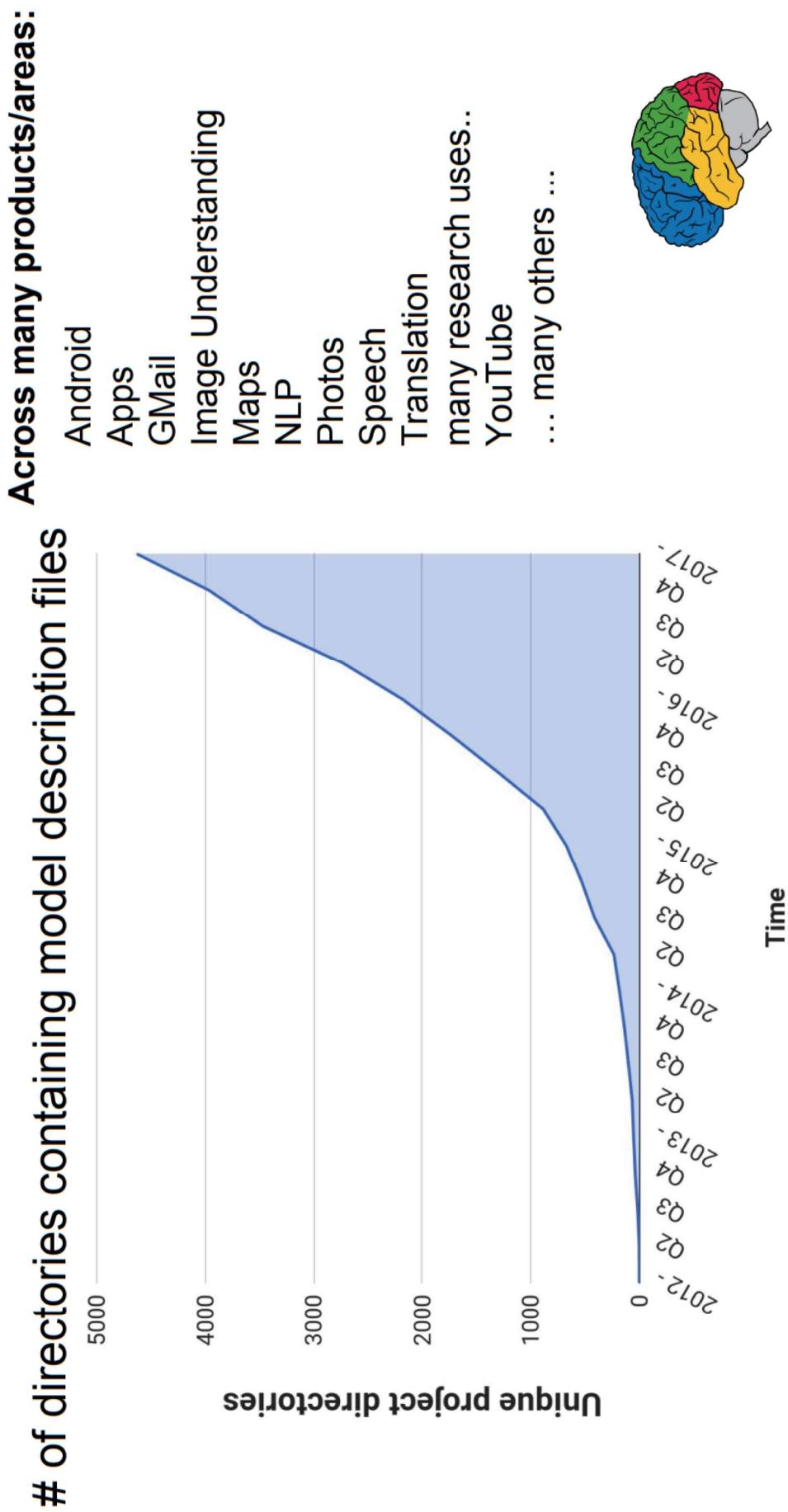
Human-level artificial intelligence **has the potential to help humanity thrive more than any invention that has come before it** – Dileep George (Co-Founder Vicarious)



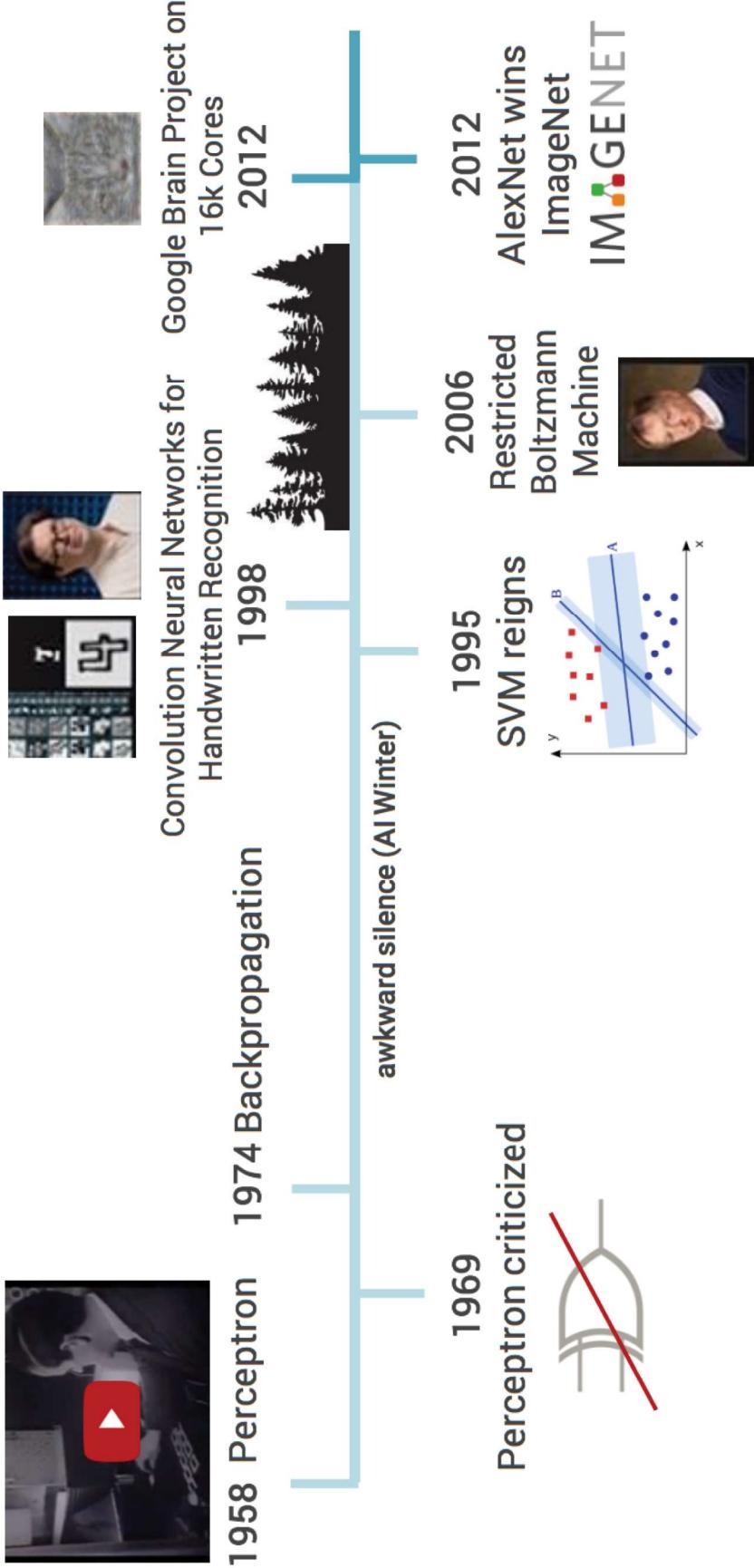
For a very long time it will be a **complementary tool** that human scientists and human experts can use to help them with the things that humans are not naturally good – Demis Hassabis (Co-Founder DeepMind)

# Porque Deep learning?

## Growing Use of Deep Learning at Google



# Rumo a Deep Learning



# Rumo a Deep Learning



ImageNet: The “computer vision World Cup”

# Grandes usuários de Deep Learning



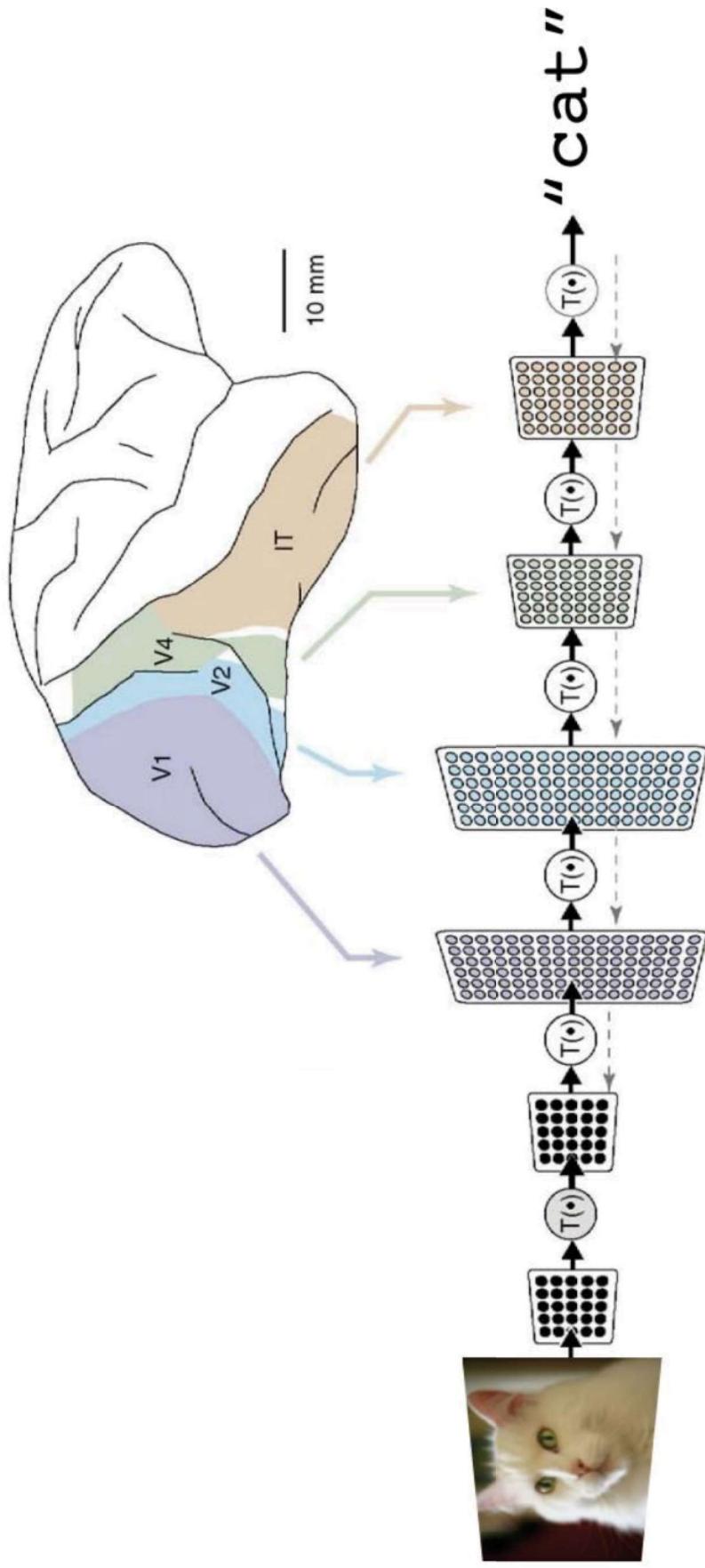
YAHOO! GOOGLE



NVIDIA®

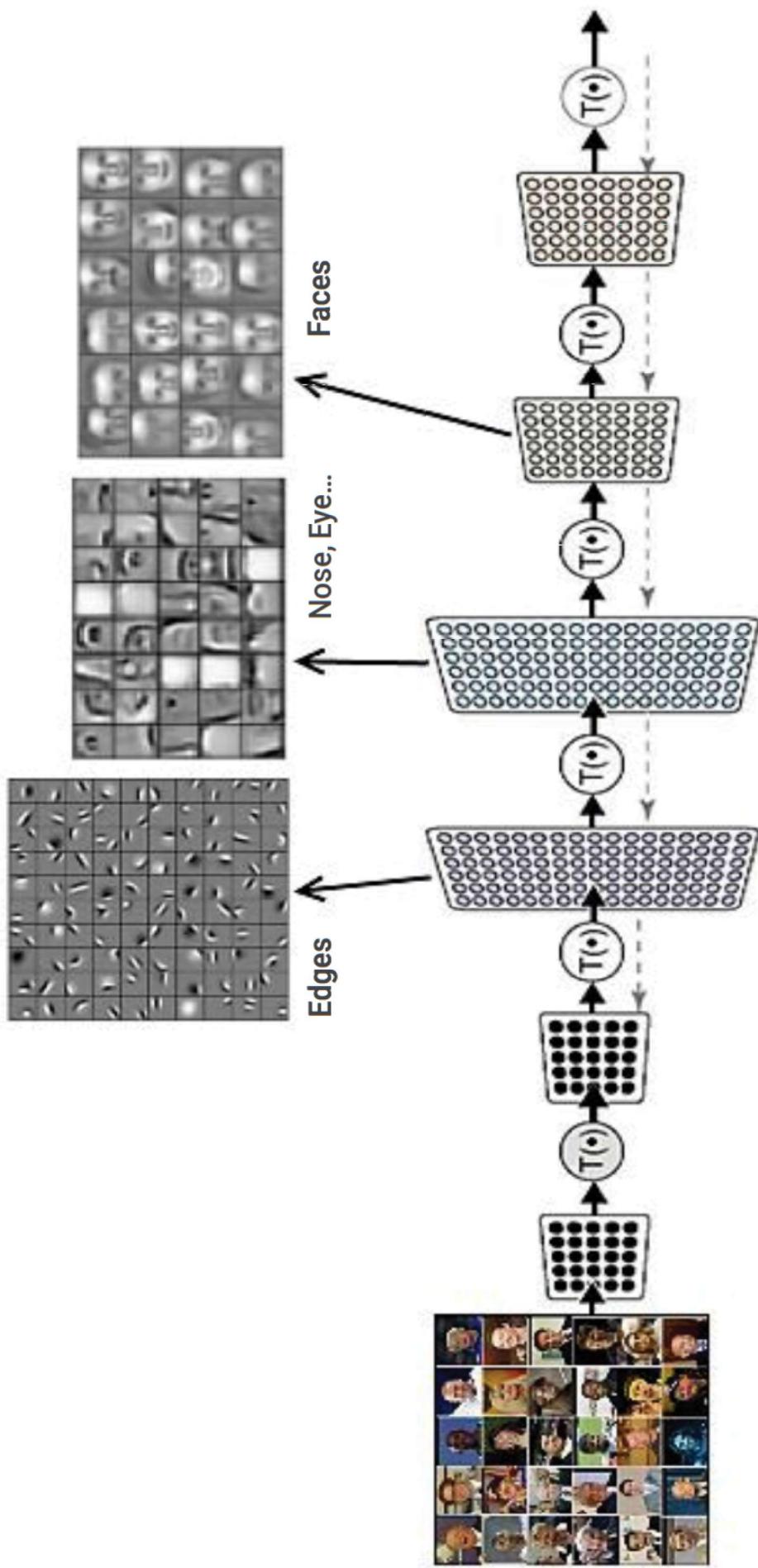
Baidu 百度

# Deep Learning

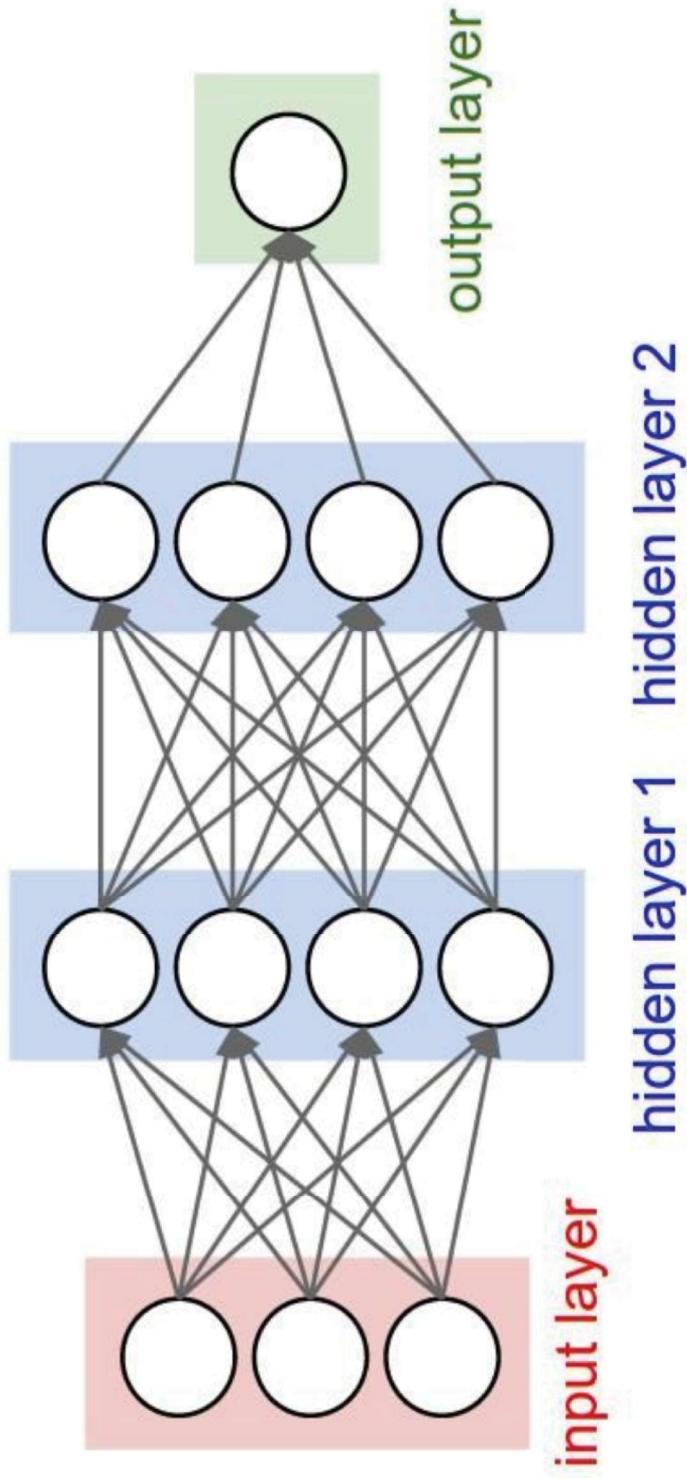


- Uma rede neural profunda consiste em uma hierarquia de camadas, em que cada camada transforma os dados de entrada em representações mais abstratas (por exemplo, borda -> nariz -> face).
- A camada de saída combina esses recursos para fazer previsões.

# Deep Learning



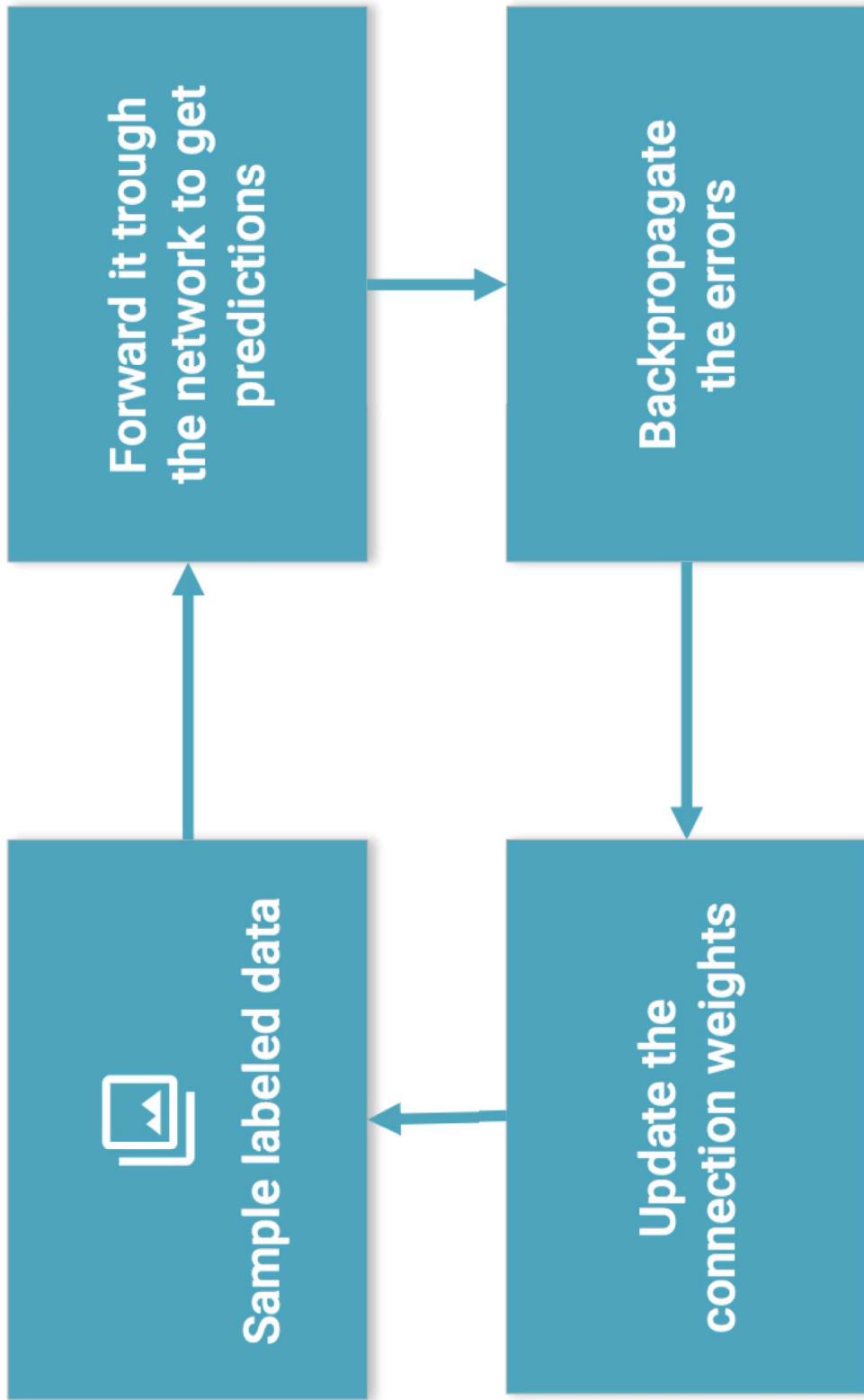
# Deep Learning



- Cada camada é representada como uma série de neurônios e, progressivamente, extrai recursos de maior e maior nível da entrada até que a camada final faça essencialmente uma decisão sobre o que a entrada mostra;

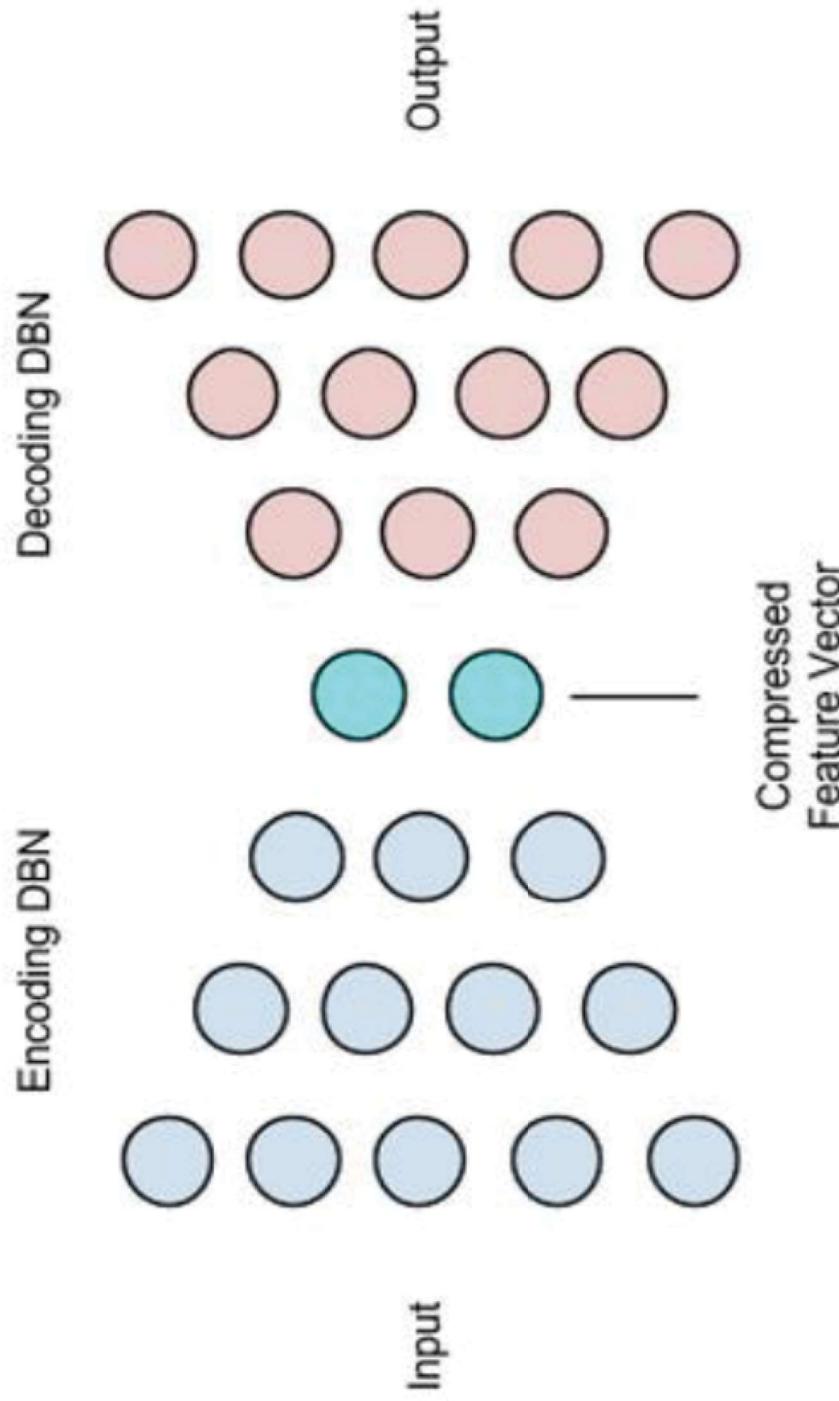
- Quanto mais camadas a rede tiver, maiores serão as características que ela aprenderá.

# Deep Learning – Fluxo de Treinamento



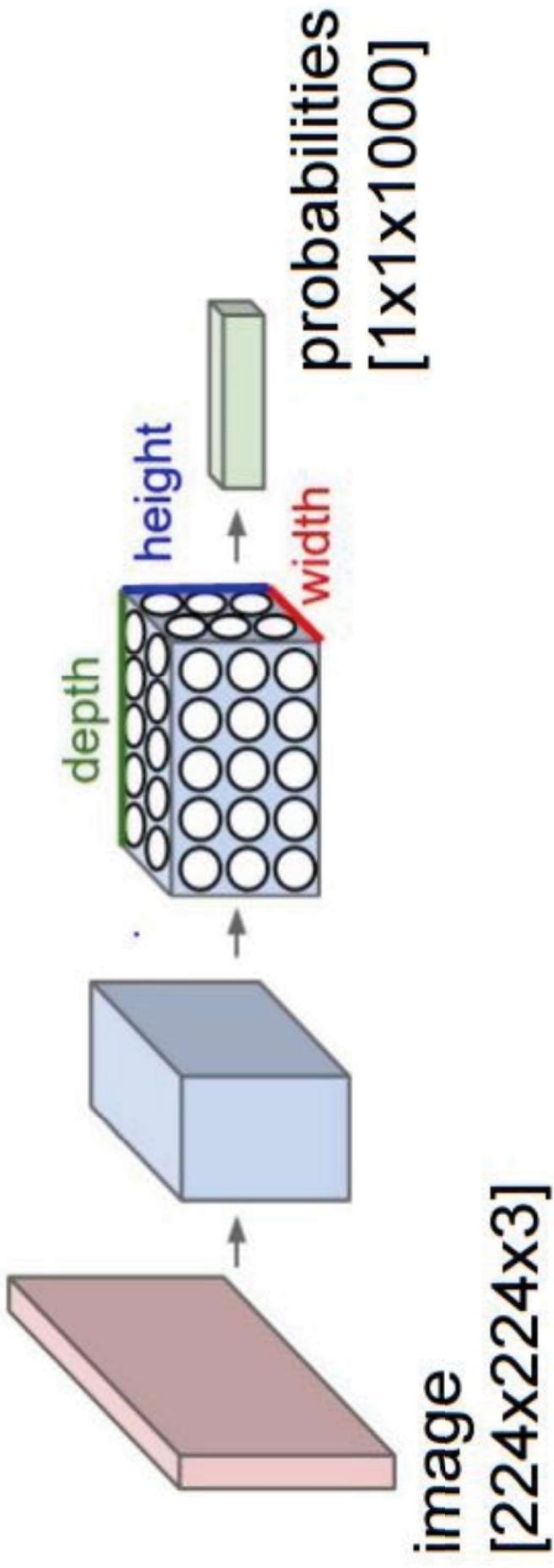
- Ou seja, a rede acaba gerando um sinal de erro que mede a diferença entre as previsões da rede e os valores desejados e, em seguida, usa-se este sinal de erro para alterar os pesos (ou parâmetros) para que as previsões fiquem cada vez mais precisas.

# Deep Learning – Deep Autoencoders



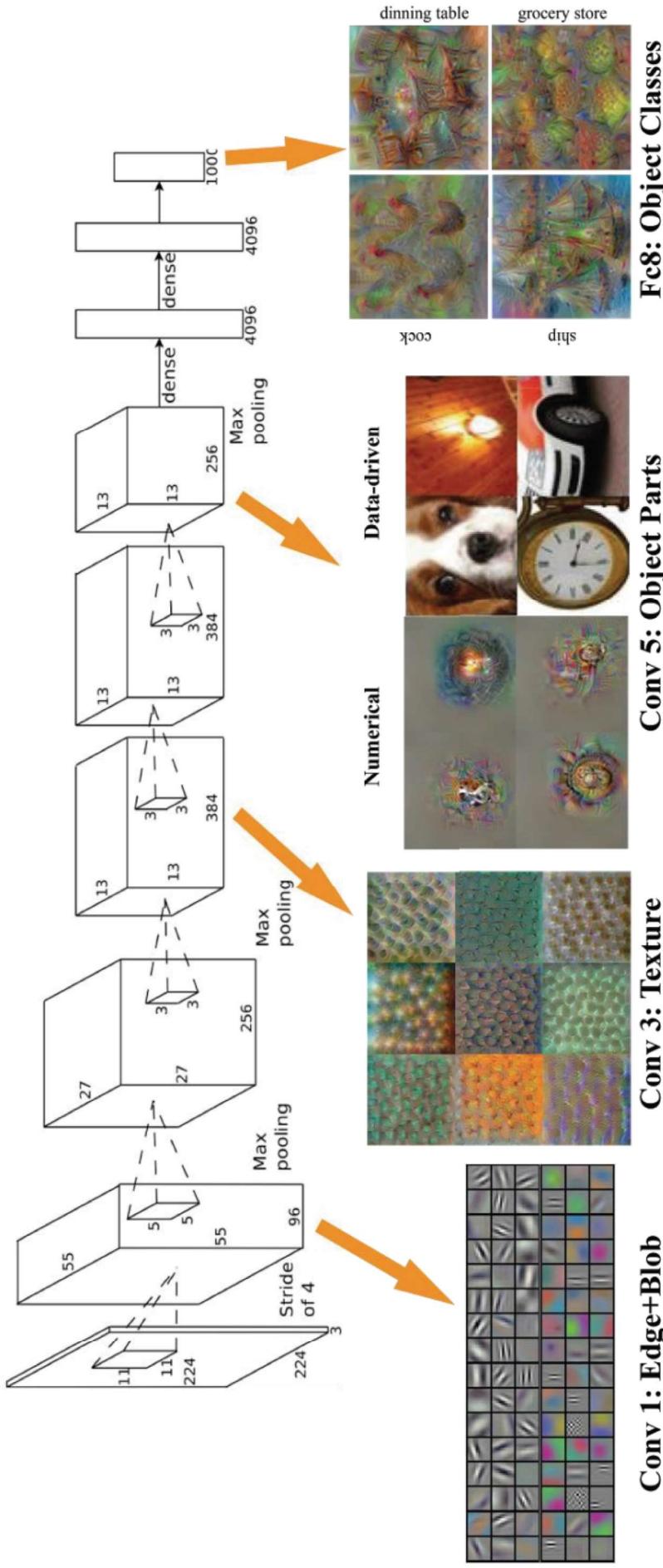
- Composto por duas redes simétricas de crenças profundas. A rede de codificação aprende a comprimir a entrada para um vetor condensado (redução de dimensionalidade). A rede de decodificação pode ser usada para reconstruir os dados.
- Ex: Modelagem de Tópicos: Documento em uma coleção é convertido em um Bag-of Words e transformado em um vetor de recurso compactado usando um autoencoder. A distância de todos os outros documentos vetoriais pode ser medida e os vetores de documentos próximos se enquadram no mesmo tópico.

# Deep Learning - Convolutional Neural Nets (CNN)



- Redes neurais convolucionais aprendem uma representação complexa de dados visuais usando grandes quantidades de dados. Eles são inspirados pelo sistema visual humano e aprendem múltiplas camadas de transformações, que são aplicadas uma sobre a outra para extrair uma representação progressivamente mais sofisticada da entrada.
- Cada camada de uma CNN recebe um volume 3D de números e gera um volume 3D de números. Por exemplo, A imagem é um cubo  $224 * 224 * 3$  (RGB) e será transformada em  $1 * 1000$  vetores de probabilidades.

# Deep Learning - Convolutional Neural Nets (CNN)



- A camada de convolução é um detector de recursos que aprende automaticamente a filtrar as informações não necessárias de uma entrada usando o kernel de convolução.
- As camadas de agrupamento calculam o valor máximo ou médio de um recurso específico em uma região dos dados de entrada (redução de tamanho das imagens de entrada).
- Também ajuda a detectar objetos em alguns locais incomuns e reduz o tamanho da memória.

# Arquitetura em RNs – Camadas e Número de Neurônios

- A Camada de Entrada

- Toda RN tem exatamente uma delas, sem exceções.
- Com relação ao número de neurônios que compõem essa camada, esse parâmetro é completa e exclusivamente determinado a partir do momento que você conheça a forma de seus dados de treinamento.
- Especificamente, o número de neurônios que compõem essa camada é igual ao número de atributos (colunas) em seus dados. Algumas configurações de RN adicionam um nó adicional para o *bias*.

# Arquitetura em RNNs – Camadas e Número de Neurônios

- **A Camada de Saída**

- Bem como a camada de entrada, toda RN tem exatamente uma camada de saída. Determinar seu tamanho (número de neurônios) é simples; é completamente determinado pela configuração do modelo escolhido.
- Ou seja, se sua RN está sendo executado no Modo de Máquina ou no Modo de Regressão. Modo de máquina: retorna um rótulo de classe (por exemplo, "Cliente VIP" / "Cliente Comum"). O modo de regressão retorna um valor (por exemplo, débitos contraídos).
- Se a RN for um regressor, a camada de saída terá um único nó.
- Se a RN for um classificador, ele também terá um único nó, a menos que o *softmax* seja usado, caso em que a camada de saída terá um nó por rótulo de classe em seu modelo.

# Arquitetura em RNNs – Camadas e Número de Neurônios

## • As camadas ocultas

- **Quantas camadas ocultas?** Bem, se seus dados são linearmente separáveis (o que você geralmente sabe quando você começa a codificar uma RN), então você não precisa de nenhuma camada oculta. Naturalmente, você não precisa de uma RN.
- Além disso, como você provavelmente sabe, há uma montanha de comentários sobre a questão da configuração de camadas ocultas em RNNs
- Uma questão dentro deste assunto em que há um consenso é a diferença de desempenho da adição de camadas ocultas adicionais: as situações em que o desempenho melhora com uma segunda (ou terceira, etc.) camada oculta são muito poucas. Uma camada oculta é suficiente para a grande maioria dos problemas.
- Então, e quanto ao tamanho da (s) camada (s) oculta (s) - quantos neurônios? Existem algumas regras empíricas, destas, a mais comum é que "o tamanho ótimo da camada oculta é geralmente entre o tamanho da entrada e o tamanho das camadas de saída".
- Em suma, para a maioria dos problemas, é possível obter um desempenho decente (mesmo sem uma segunda etapa de otimização) definindo a configuração da camada oculta usando apenas duas regras: (i) o número de camadas ocultas é igual a uma; e (ii) **o número de neurônios nessa camada é a média dos neurônios nas camadas de entrada e saída.**

# Arquitetura em RNNs – Camadas e Número de Neurônios

- O processo de construção de uma Rede Neural Convolucional envolve sempre quatro etapas principais.
  - Etapa 1: convolução
  - Etapa 2: Pooling
  - Etapa 3: Flattening
  - Etapa 4: Full Connection
- **Camada Densa (Fully Connected):**
  - Uma camada densa é apenas uma camada regular de neurônios em uma rede neural. Cada neurônio recebe entrada de todos os neurônios na camada anterior, portanto, densamente conectados.
  - A camada tem uma matriz ponderada  $W$ , um vetor de polarização  $b$  e as ativações da camada anterior  $a$ .
- **Max Pooling / Pooling**
  - O Max Pooling ou Pooling é um processo de discretização baseado em amostra.
  - O objetivo é reduzir a amostragem de uma representação de entrada (imagem, matriz de saída de camada oculta, etc.), reduzindo sua dimensionalidade e permitindo suposições sobre os recursos contidos nas sub-regiões categorizadas.

# Arquitetura em RNNs – Camadas e Número de Neurônios

- **Flattening (Nivelamento):**
  - O passo de achatamento é necessário para que você possa fazer uso de camadas totalmente conectadas após algumas camadas convolucionais.
  - Camadas totalmente conectadas não têm uma limitação local como camadas convolucionais (que só observam alguma parte local de uma imagem usando filtros convolucionais).
  - Isso significa que você pode combinar todos os recursos locais encontrados nas camadas convolucionais anteriores.
  - Cada canal de mapa de recursos na saída de uma camada CNN é uma matriz 2D "achatada" (*flat*) criada pela adição dos resultados de múltiplos kernels 2D (um para cada canal no canal de entrada).

# Arquitetura em RNNs – Camadas e Número de Neurônios

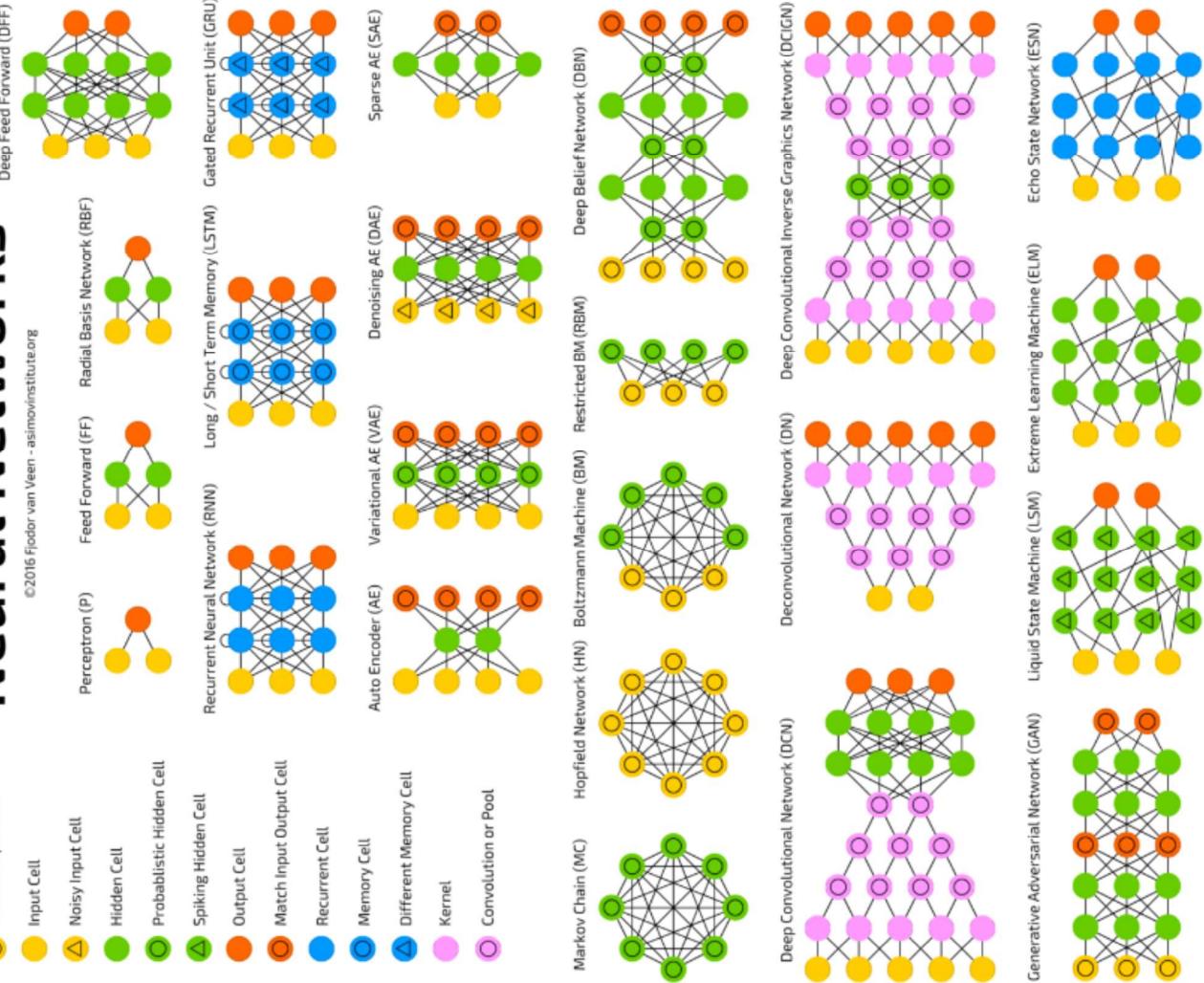
- Regularizadores:
  - Dropout
    - Dropout é uma técnica de regularização que visa reduzir a complexidade do modelo com o objetivo de evitar *overfitting*.
    - Usando dropout, você desativa aleatoriamente certas unidades (neurônios) em uma camada com uma certa probabilidade  $p$  de uma distribuição de Bernoulli (normalmente 50%, mas este é outro hiperparâmetro que pode ser modificado).
    - Se você definir a metade das ativações de em uma camada a zero, a rede neural não será capaz de confiar em ativações específicas em um determinado passo de *feed-forward* durante o treinamento. Como consequência, a rede neural aprenderá **representações diferentes e redundantes**; a rede **não pode confiar na neurônios particulares e a combinação** (ou interação) destes para estar presente. Outro efeito colateral agradável é que o treinamento será mais rápido.
  - Notas técnicas adicionais: O dropout é aplicado apenas durante o treinamento, e você precisa redimensionar as ativações neuronais remanescentes. Por exemplo, se você definir 50% das ativações em uma determinada camada para zero, você precisará aumentar as outras por um fator de 2. Finalmente, se o treinamento tiver terminado, você usaria a rede completa para testes (ou em outras palavras, você define a probabilidade de

# Arquitetura em RNNs – Camadas e Número de Neurônios

- Otimizadores:
  - RMSProp
    - O otimizador RMSprop é semelhante ao algoritmo de descida de gradiente com momentum.
    - O otimizador do RMSprop restringe as oscilações na direção vertical. Portanto, podemos aumentar nossa taxa de aprendizado e nosso algoritmo pode tomar medidas maiores na direção horizontal convergindo mais rapidamente. A diferença entre RMSprop e gradiente descendente é como os gradientes são calculados.
  - Adam
    - Adam é um algoritmo de otimização que pode ser usado em vez do procedimento clássico de descida de gradiente estocástico para atualizar os pesos da rede de forma iterativa com base nos dados de treinamento.
    - A descida de gradiente estocástica mantém uma única taxa de aprendizado (denominada alfa) para todas as atualizações de peso e a taxa de aprendizado não muda durante o treinamento. Uma taxa de aprendizado é mantida para cada peso de rede (parâmetro) e adaptada separadamente à medida que o aprendizado se desdobra.
    - O método calcula as taxas individuais de aprendizagem adaptativa para diferentes parâmetros a partir das estimativas do primeiro e segundo momentos dos gradientes.

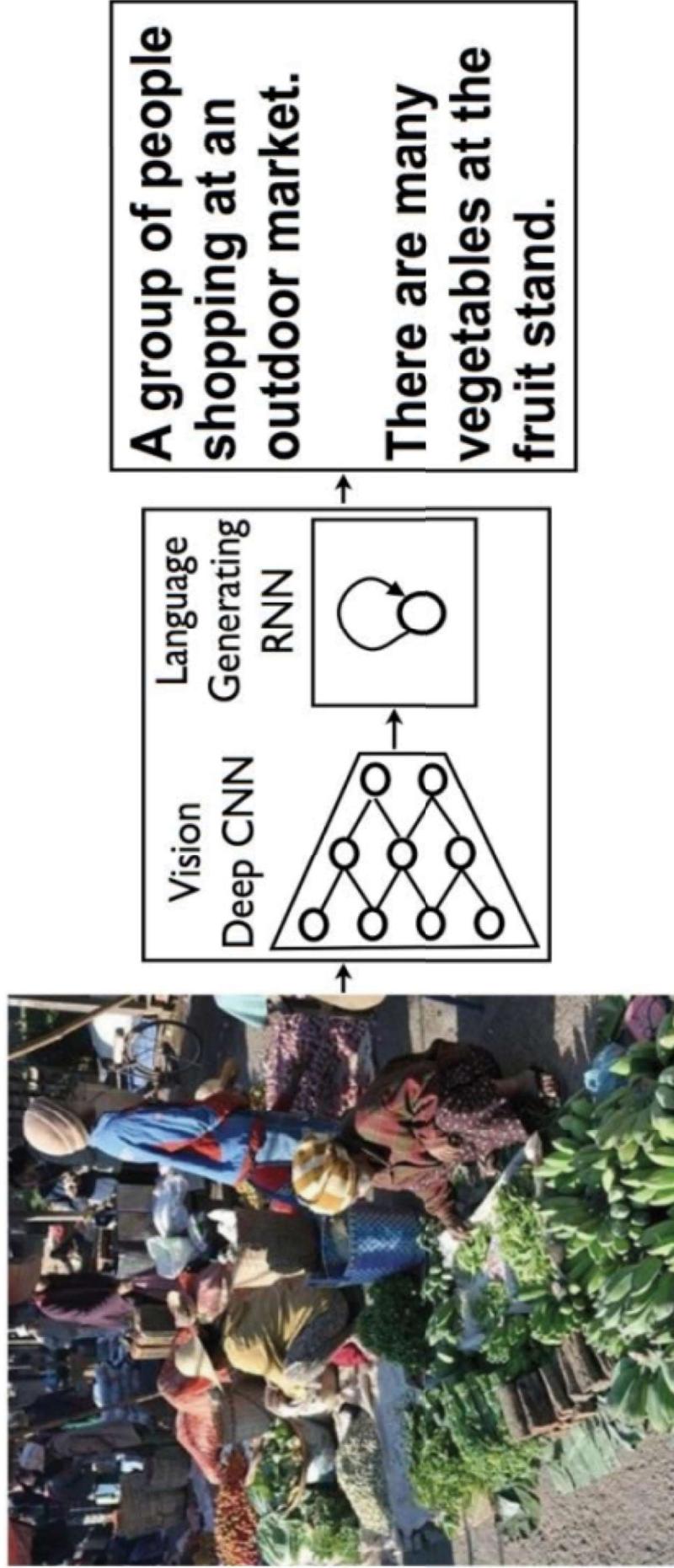
# Neural Networks

A mostly complete chart of



© 2016 Fjodor van Veen - [asimovinstitute.org](http://asimovinstitute.org)

# Deep Learning – CNN + RNN – Gerador de Legenda



# Deep Learning para Ciência de Dados – Requerimentos



Grande *dataset* com boa qualidade;



Objetivos mensuráveis e descritíveis (para definição do custo);

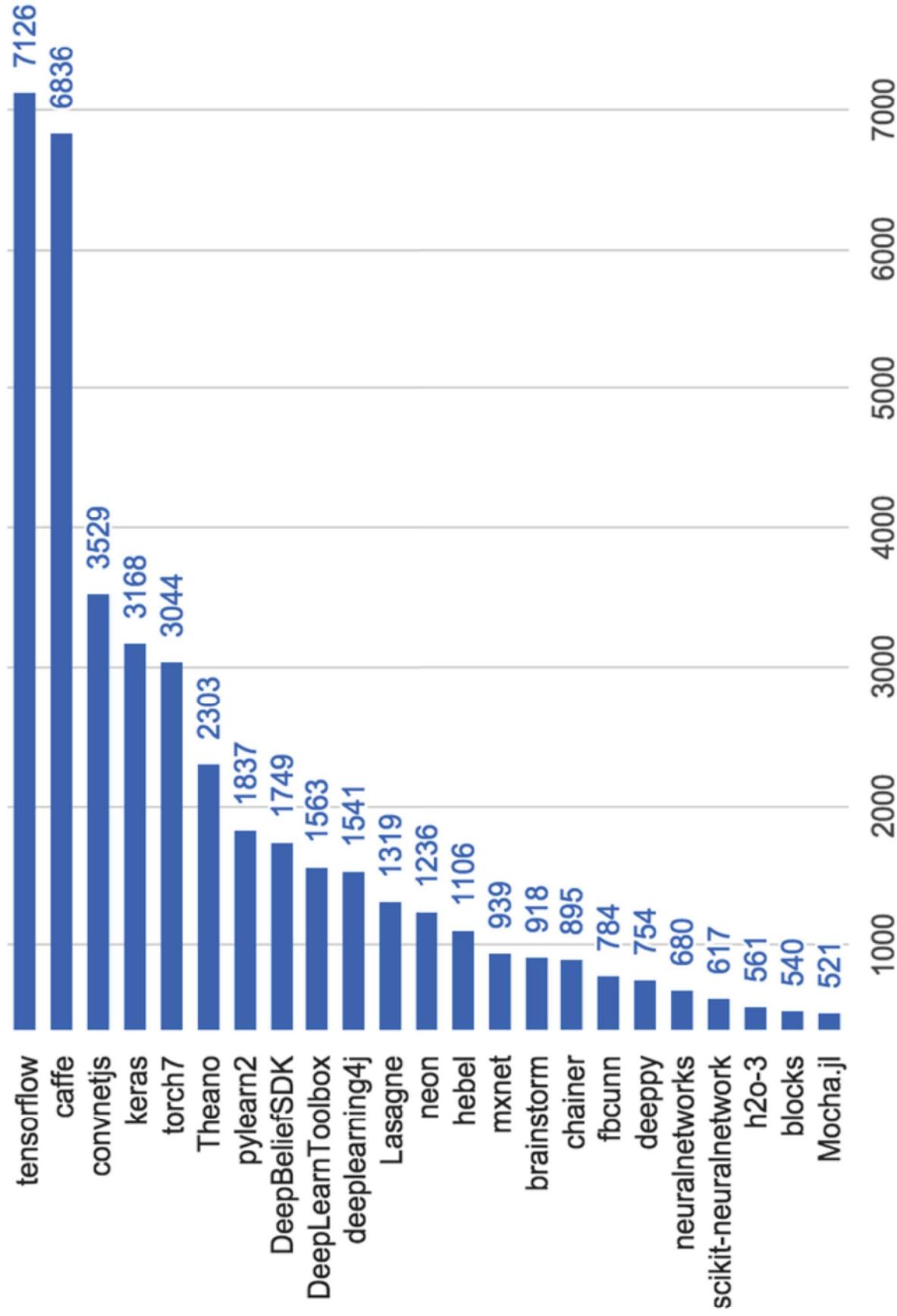


**Poder computacional** (Ex: instâncias de GPUs);



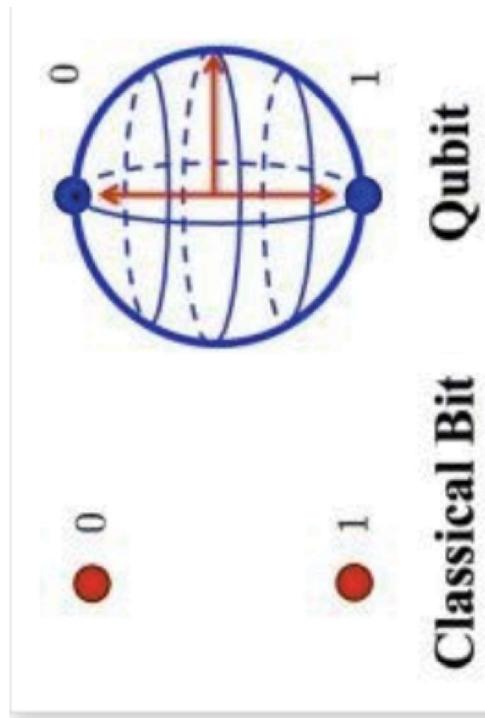
Excede em tarefas onde a unidade básica (pixel, palavra) tem muito pouco significado por si mesma, mas a combinação de tais unidades tem um significado útil.

# Ferramentas em Deep Learning



# Deep Learning – Computação Quântica

- Laboratório de Inteligência Artificial Quântica - iniciativa conjunta da NASA e do Google para estudar como a computação quântica pode promover o aprendizado de máquina.
- Computadores quânticos lidam com o que são chamados bits quânticos ou qubits que podem facilmente ter um valor de um ou zero ou qualquer coisa entre eles.
- A computação quântica representa uma mudança de paradigma, uma mudança radical na forma como fazemos computação e em uma escala que tem um poder inimaginável - **Eric Ladizinsky** (Co-fundador D-Wave)



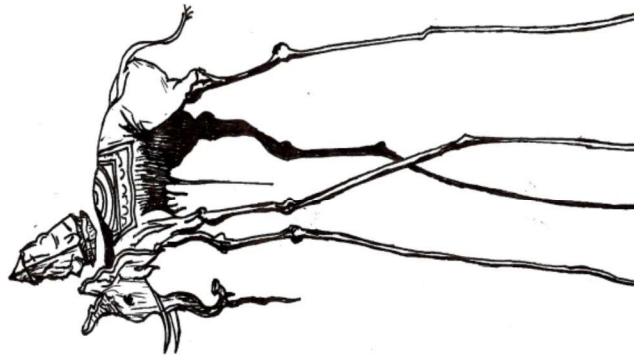
# Conclusões

- Avanços significativos em reforço profundo e aprendizado não supervisionado;
- Arquiteturas maiores e mais complexas baseadas em vários módulos / técnicas intercambiáveis;
- Modelos mais profundos que podem aprender com muito menos casos de treinamento;
- Problemas mais difíceis, como o entendimento de vídeo e o processamento de linguagem natural, serão abordados com sucesso por algoritmos de aprendizado profundos.

# Conclusões

- Estamos em busca de máquinas que aprendem a representar o mundo a partir da experiência;
- Deep Learning não é mágica!
  - Apenas estatísticas em uma caixa preta, super eficaz em padrões de aprendizado;
- Nós não descobrimos criatividade e empatia humana (ainda)!
- Extensa transição da pesquisa para produtos de consumo nos últimos anos;
- Fará com que as ferramentas que você usa todos os dias funcionem melhor, mais rápido e de maneira mais inteligente.

**“É preciso provocar sistematicamente confusão. Isso promove a criatividade. Tudo aquilo que é contraditório gera vida”**



**Salvador Dalí**

# Contatos



ahirtonlopes@gmail.com  
@ahirtonLopes

# AI Brasil



facebook



WhatsApp



meetup



LinkedIn



Dmitriy Ryaboy  
@squarecog

Follow

Replying to [@squarecog @josh\\_wills](#)

(DL is cool as hell for the right problem. But the number of people who claim knowledge having written 10 lines of Keras...)

RETWEET	LIKES
1	8



9:45 PM - 17 Feb 2017

# Obrigado!

## Deep Learning



What society thinks I do  
What other computer  
scientists think I do



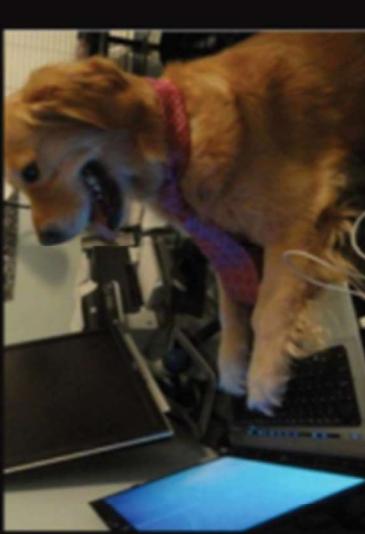
What my friends think I do



What mathematicians think I do



What I actually do



What I think I do

from keras import \*