

Aprendizado Profundo

Sarajane Marques Peres

Abril de 2020

Material baseado em:

Deep Learning. Ian Goodfellow, Yoshua Bengio e Aaron Courville.
MIT Press, 2016.

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

Neural Networks: a comprehensive foundation. Simon Haykin
Prentice Hall, 1999.

Aprendizado profundo - motivação

Desafio na IA

Fornecer soluções para tarefas que são facilmente resolvidas por humanos mas que são difíceis de descrever formalmente – problemas cuja solução é intuitiva e que parece automática: reconhecimento de palavras faladas, faces, imagens

- **Aprendizado:** aprender a partir de experiências, obter conhecimento a partir da experiência, evitando que humanos precisem especificar todo o conhecimento que o computador precisa ter para resolver um problema.
- **Hierarquia de conceitos:** entender o mundo em termos de uma hierarquia de conceitos na qual cada conceito é definido em termos de suas relações com conceitos mais simples. Se nós desenharmos um grafo mostrando conceitos construídos sobre outros conceitos teremos um gráfico profundo, com muitas camadas.

Aprendizado profundo - motivação

- O caso do **jogo de xadrez** representa um problema que exige pouco conhecimento sobre o mundo, e que possui um conjunto de regras objetivo e finito.
- O caso de **reconhecimento de imagens**, ou de identificação de objetos em uma imagem requer muito conhecimento sobre o mundo, e é um conhecimento subjetivo e intuitivo.



A questão chave é:

Como incorporar conhecimento e informação em um programa de computador!

Aprendizado profundo - motivação

Contraponto

Raciocínio automático sobre conhecimento codificado por meio de linguagens formais:
knowledge based approaches.

Ponto fraco

A dificuldade com conhecimento codificado formalmente sugere que sistemas de IA precisam ter a habilidade de adquirir seu próprio conhecimento, por meio da extração de padrões escondidos em dados – isso é aprendizado de máquina.

Tendência

Unir o que há de melhor nas abordagens baseadas em conhecimento com o que há de melhor nas abordagens de aprendizado de máquina (profundas ou não).

Aprendizado profundo - motivação

- Algoritmos de aprendizado de máquina dependem fortemente da representação dos dados - a forma como os dados são apresentados ao algoritmo.
- Cada pedaço de informação incluída na representação dos dados é conhecida como *feature* (característica descritiva).
- O algoritmo de aprendizado de máquina (tradicional) não tem o controle sobre quais características são fornecidas a ele, embora, a depender do algoritmo usado, determinadas características influenciarão mais ou menos a resposta, ou modelo, gerada.

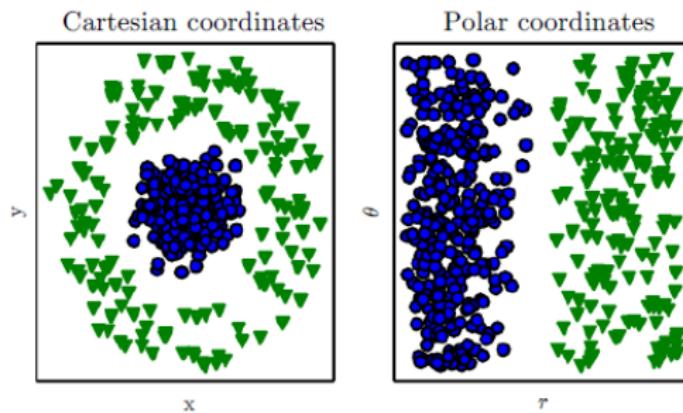
Pense: fazer contas com algarismos arábicos e fazer contas com algarismos romanos.

$$\begin{array}{r} 2009 \quad | \quad 19 \\ 109 \quad \underline{-} \quad 105 \\ 95 \\ \textcircled{14} \end{array}$$

$$\begin{aligned} & \text{MCCXXIII} + \text{MCXIV} \\ & = \text{MCXXIII} + \text{MCXIII} \\ & \begin{array}{r} \text{M} \quad \text{CC} \quad \text{XX} \quad \text{III} \\ + \quad \text{M} \quad \text{C} \quad \text{X} \quad \text{III} \\ \hline \text{MM} \quad \text{CCC} \quad \text{XXX} \quad \text{IIIIII} \end{array} \\ & = \text{MMCCCXXXVII} \end{aligned}$$

O problema da representação

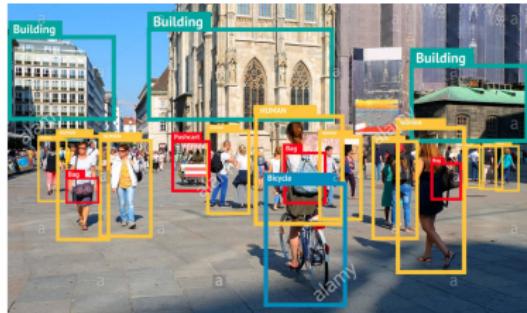
Representação com coordenadas cartesianas e representação com coordenadas polares.



* Goodfellow, Bengio e Courville (2016) e David Warde-Farley. Pg. 4.

Como você descreveria um carro? Usando as rodas? Como descrever as rodas? Que forma elas assumem em uma imagem?

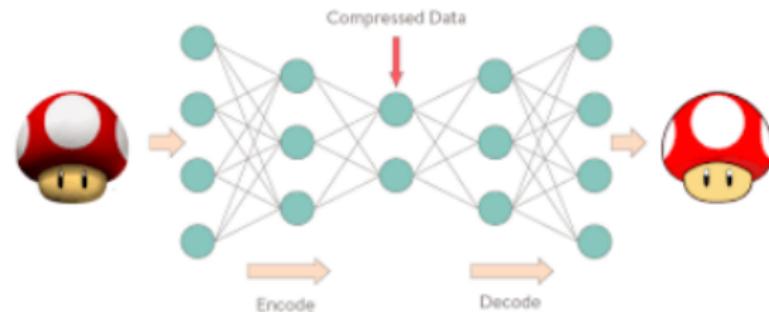
O problema da representação



Aprendizado profundo - motivação

Uma solução para esse problema é usar o aprendizado de máquina tanto para **descobrir o mapeamento da representação (entrada) para a saída**, quanto a **representação em si: representation learning**.

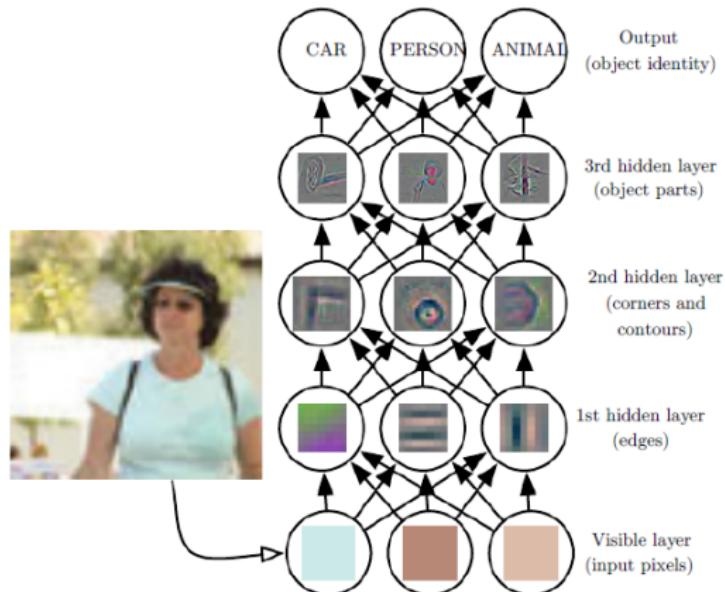
O exemplo clássico é o *autoencoder*: uma combinação de uma função de codificação - que converte os dados em uma representação, com uma função de decodificação - que converte a representação de volta ao formato original.



<https://towardsdatascience.com/deep-autoencoders-using-tensorflow-c68f075fd1a3>

Aprendizado profundo - motivação

Aprendizado profundo atua fortemente na questão da representação, construindo conceitos complexos a partir de conceitos simples. O conceito referente à imagem de uma pessoa pode ser representado usando conceitos simples como "cantos e cortornos", que por sua vez podem ser definidos em termos de arestas.



* Goodfellow, Bengio e Courville (2016) Pg. 6.

Aprendizado profundo - motivação

Aprendizado profundo e o Multilayer Perceptron (MLP)

O MLP constrói uma função matemática que mapeia um conjunto de valores de entrada em um conjunto de valores de saída. A função é formada pela composição de muitas funções mais simples: nós podemos encarar o resultado de cada função simples como uma nova representação para a sua entrada.

Aprendizado profundo - o que é profundo?

- número de instruções sequenciais que devem ser executadas ao passar pela arquitetura: o comprimento do caminho mais longo no fluxo de trabalho para produzir uma saída dado uma entrada. Isso depende da granularidade do passo computacional assumido pela linguagem de representação.
- profundidade do grafo que descreve como os conceitos estão relacionados com outros conceitos. Porém, quanta computação é necessária para processar um conceito pode variar.

então ?

Não há um consenso sobre qual profundidade é necessária para que um modelo seja qualificado como profundo.

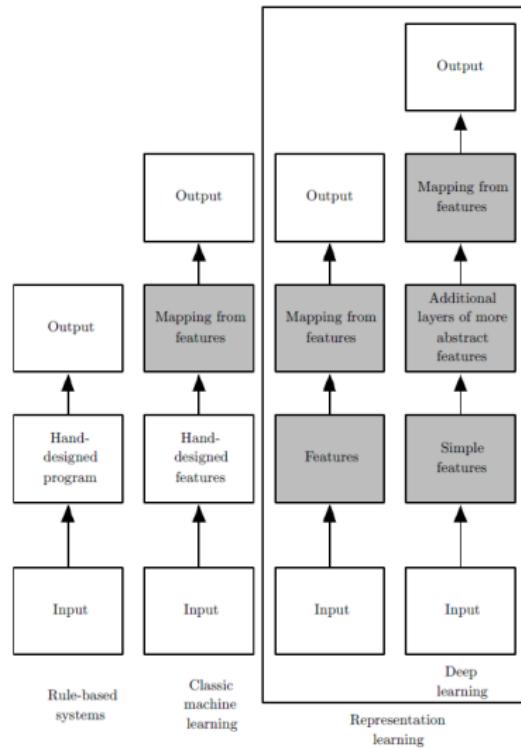
Aprendizado profundo - uma definição

Os autores do livro resumem que:

Aprendizado profundo é

... um tipo de aprendizado de máquina que possui muito poder e muita flexibilidade porque aprende a representar o mundo como uma hierarquia de conceitos aninhados, sendo que cada conceito é definido em termos de conceitos mais simples, e como representações mais abstratas computadas em termos de outras menos abstratas.

Aprendizado profundo - uma definição



* Goodfellow, Bengio e Courville (2016) Pg. 10.

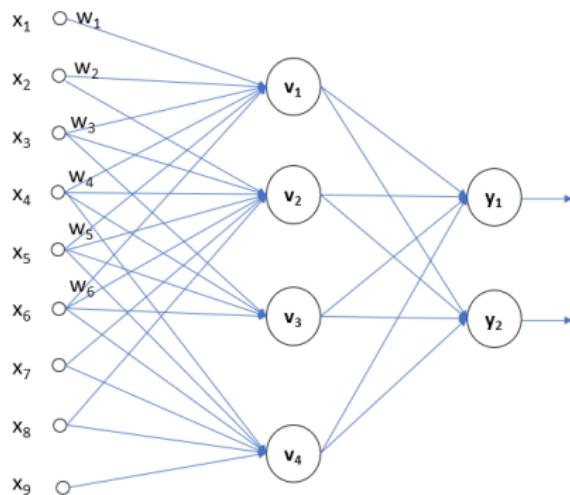
Redes neurais convolucionais - princípio

Como embutir conhecimento a priori em um projeto de rede neural?

- restringindo a arquitetura da rede usando um padrão de conexões locais conhecidos como campos receptivos;
- restringindo a escolha dos pesos sinápticos usando pesos compartilhados.

A computação da entrada dos neurônios escondidos resultará em uma operação de convolução!

Redes neurais convolucionais - princípio



Exemplo com seis conexões locais por neurônio escondido. A entrada nos neurônios escondidos (v) é: $v_j = \sum_{i=6}^6 w_i v_{i+j-1}$ com $j = 1, 2, 3, 4$, em que $\{w_i\}_{i=1}^6$ formam o conjunto de pesos compartilhados por todos os neurônios escondidos, e x_k é o sinal escolhido do conjunto de entradas, sendo $k = i + j - 1$.

Redes neurais convolucionais - a convolução

Considere

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a).$$

- $x(t)$ é uma função de localização de espaçonaves no tempo (t)
- $w(t)$ é uma função de ponderação no tempo (t), de forma que o tempo recente recebe ponderação maior que o tempo antigo
- $*$ é uma operação de convolução
- $s(t)$ é o resultado da convolução da função de ponderação $w(t)$ sobre a função de localização $x(t)$ – uma estimativa suavizada da posição da espaçonave no tempo (t)
- a é a idade da medida

Terminologia

- o primeiro argumento (a função $x(t)$) para a convolução é chamado de *entrada*;
- o segundo argumento (a função $w(t)$) para a convolução é chamado de *kernel*;
- a saída pode ser dita o *mapa de características*.

Redes neurais convolucionais

Em aprendizado de máquina:

- a *entrada* é, geralmente, é um array multidimensional de dados;
- o *kernel* é, geralmente, um array multidimensional de parâmetros que são adaptados por um algoritmo de treinamento;
- esses arrays multidimensionais são chamados de **tensores**.

A operação de convolução para o caso de array bi-dimensional, como uma imagem, é:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

ou, de forma comutativa

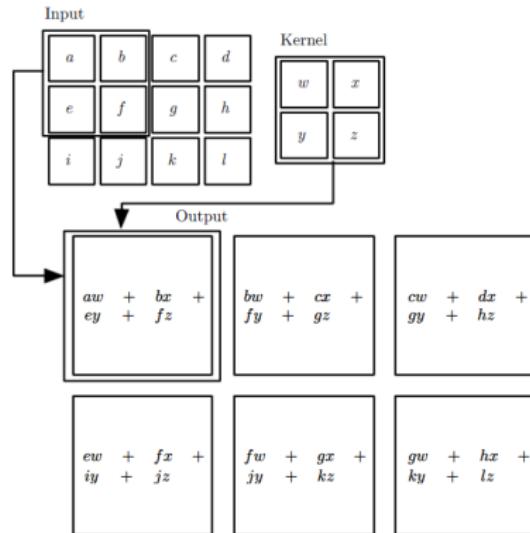
$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n)$$

ou, na forma de correlação cruzada

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Redes neurais convolucionais

Exemplo de convolução com arrays bidimensionais



* Goodfellow, Bengio e Courville (2016) e David Warde-Farley. Pg. 334.

Redes neurais convolucionais

Redes neurais convolucionais - Convolution Neural Networks - CNN

É um caso especial do Multilayer Perceptron (MLP). É um MLP projetado, por exemplo, para reconhecer formas bi-dimensionais, de maneira invariante à translação (e outras distorções a depender das operações combinadas). Ela é uma rede de aprendizado supervisionado, cuja estrutura inclui algumas restrições.

- Extração de características
- Mapeamento de características
 - invariância de translação/deslocamento (turno)
 - redução no número de parâmetros livres
- Sub-amostragem

São redes neurais simples que usam convolução (um tipo especializado de operação linear) no lugar de multiplicação de matrizes em pelo menos uma de suas camadas.

Essa rede neural é útil para processamento de dados com *grid-like topology*: séries temporais (caso 1D) e imagens (caso 2D).

Redes neurais convolucionais

Extração de características: cada neurônio recebe as entradas ponderadas de seu campo receptivo local, realizando a extração de características locais. Uma vez que a característica foi extraída, a sua localização exata (em relação à entrada) se torna menos importante enquanto sua posição relativa em relação a outras características é (aproximadamente) preservada.

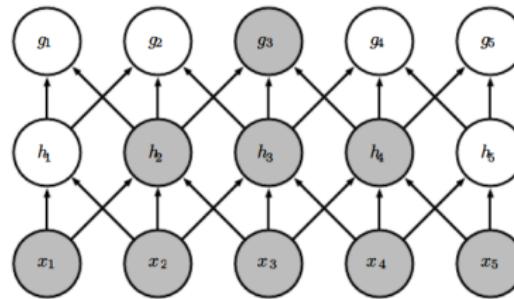
Mapeamento de características: cada camada de processamento da rede neural é composta de múltiplos mapas de características, e cada um desses mapas, na camada, compartilha os mesmos pesos sinápticos. Isso gera dois efeitos benéficos:

- o mapa de características é produzido por convolução (permitindo a invariância) seguida de uma função sigmoide (que realiza um achatamento)¹
- os pesos compartilhados reduzem o número de parâmetros livres da rede

¹ Segundo Haykin em sua explicação sobre redes neurais convolucionais. Para Goodfellow, a convolução deve ser seguida de uma função *rectified linear* (uma unidade ReLU - *rectified linear unit*).

Redes neurais convolucionais

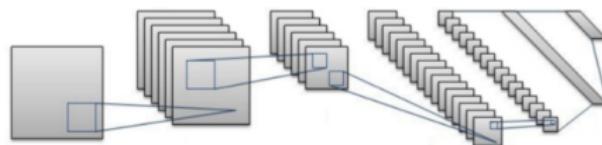
Campos receptivos de neurônios em camadas mais profundas são maiores do que os campos receptivos de unidades de camadas mais rasas. Isso mostra que mesmo que as conexões da CNN sejam esparsas, unidades em camadas mais profundas estão indiretamente recebendo informação derivada de toda a, ou uma grande parte da, entrada.



* Goodfellow, Bengio e Courville (2016) e David Warde-Farley. Pg. 337.

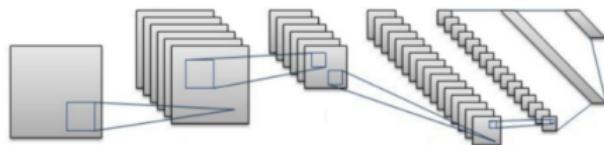
Redes neurais convolucionais

Sub-amostragem: (ou pooling) Cada camada convolucional é seguida por uma camada computacional que executa uma média local e uma sub-amostragem, por meio da qual a resolução do mapa de características é reduzida. Esta operação tem o efeito de reduzir a sensibilidade da saída do mapa de características em termos de deslocamentos ou outros distorções.



Um função pooling substitui uma saída da camada anterior, em determinadas localizações, por um resumo estatístico das saídas vizinhas. Por exemplo, operação *max-pooling* retorna a saída máxima dentro de uma vizinhança retangular. Ainda é possível usar médias, normas ou distâncias no lugar da operação *max*.

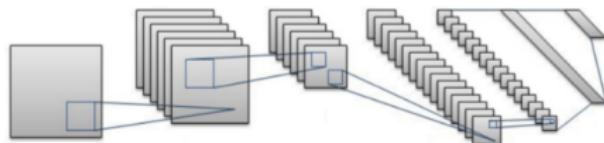
Redes neurais convolucionais



Considere (ex. Haykin):

- arquitetura: uma camada de entrada, quatro camadas escondidas, uma camada de saída
- entrada: imagens de 28×28 neurônios sensoriais, referentes as 26 caracteres
- o layout computacional altera entre convoluções e subamostragem (como descrito na sequência)

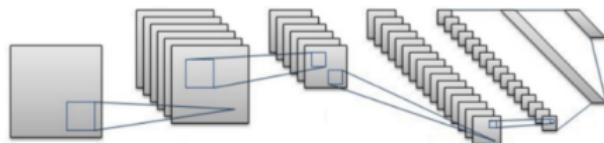
Redes neurais convolucionais



- A primeira camada executa convolução. Ex.: seis mapas de características, cada um com 24×24 neurônios. Cada neurônio está associado a campos receptivos de tamanho 5×5 .

Redes neurais convolucionais

Redes neurais convolucionais



- A segunda camada executa a sub-amostragem (ou pooling) e média local. Ex.: Ela também consiste de seis mapas de características, cada um com 12×12 neurônios. Cada neurônio tem um campo receptivo de 2×2 .

Redes neurais convolucionais

- A terceira camada escondida executa a segunda convolução. Ex.: Ela consiste de 16 mapas de características, cada um com 8×8 neurônios. Cada neurônio tem conexões provenientes de vários mapas de características da camada anterior.
- A quarta executa a segunda sub-amostragem e média local. Ex.: Ela consiste de 12 mapas de características de 4×4 neurônios.
- A camada de saída é o estágio final de convolução. Ex.: Um MLP de duas camadas em que cada neurônio na saída é associado a um dos 26 caracteres possíveis. Cada neurônio da “entrada” é associado ao campo receptivo de tamanho 4×4 , com pesos sinápticos provenientes de vários mapas de características da camada anterior.

É comum encontrar a linearização e composição desses mapas bidimensionais em um array unidimensional para alimentar a MLP final, ou o que também pode ser encontrado como uma rede feedforward densa para classificação.

Redes neurais convolucionais - treinamento

- rede neural de treinamento supervisionado;
- defina os hiperparâmetros da rede;
- inicialize os pesos (parâmetros de kernel e pesos da camada densa) aleatoriamente;
- aplique backpropagation

Alternativamente, os filtros (kernels) podem ser projetados à mão ou centróides de uma execução do k-means sobre os dados de treinamento podem ser usados.

Aprendizado Profundo

● Sarajane M. Peres - sarajane@usp.br

Disciplina de Inteligência Artificial

Bacharelado em Sistemas de Informação - BSI

Escola de Artes, Ciências e Humanidades - EACH

Universidade de São Paulo - USP