

Redes Neurais (Deep Learning) : introdução

Março de 2021

Ref.: Cap. 6 do livro texto

Classificação : aprendizado supervisionado e otimização

- Temos n valores (eventualmente \mathbf{x} é um vetor d -dimensional) de entrada (variáveis de controle)

$$\mathbf{x}_{1:n} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$$

- e n valores de saída (eventualmente y é um vetor d' -dimensional)

$$y_{1:n} = (y_1, \dots, y_n)$$

- Obs: no contexto de classificação, y não é uma variável assumindo valores contínuos (lembrem-se do exemplo dos explosivos)

O que são redes neurais?

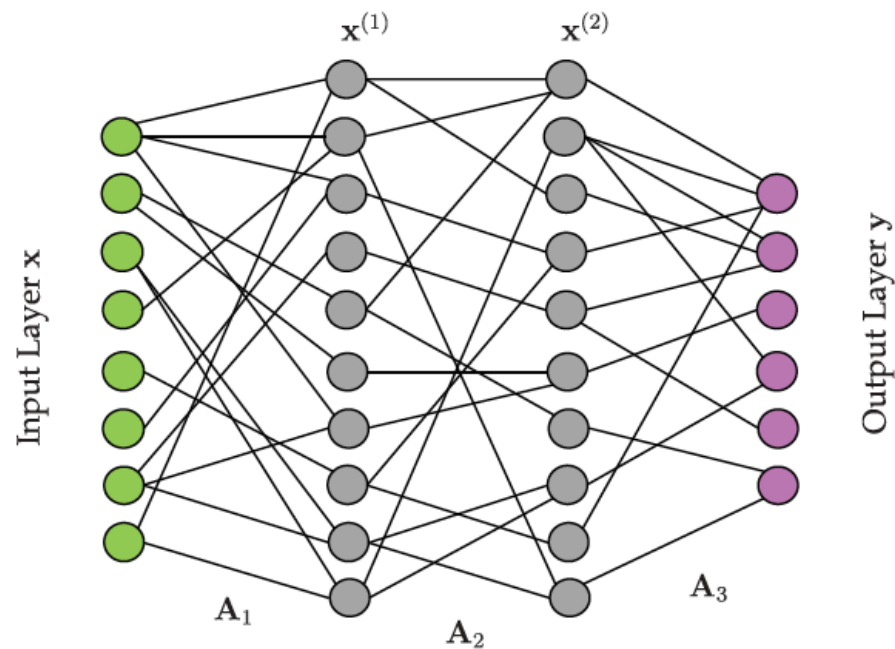


Figure 6.1 Illustration of a neural net architecture mapping an input layer x to an output layer y . The middle (hidden) layers are denoted $x^{(j)}$ where j determines their sequential ordering. The matrices A_j contain the coefficients that map each variable from one layer to the next. Although the dimensionality of the input layer $x \in \mathbb{R}^n$ is known, there is great flexibility in choosing the dimension of the inner layers as well as how to structure the output layer. The number of layers and how to map between layers is also selected by the user. This flexible architecture gives great freedom in building a good classifier.

$$y = f(x; w)$$

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Em que tipo de problemas você pode aplicar redes neurais?

- Regressão
- Classificação
- Agrupamento
- Redução de dimensão
-

Porque redes neurais se tornaram tão famosas?

- Reconhecimento de imagens
- Processamento de texto
- Reinforcement learning : carros autônomos, jogos eletrônicos,...
- E muito mais por vir..

Deep Learning . Para além de gatos e cachorros

$$\begin{aligned} \mathbf{u}(\mathbf{s}) &= -K(\mathbf{s})\nabla p(\mathbf{s}), & \mathbf{s} \in \mathcal{S}, \\ \nabla \cdot \mathbf{u}(\mathbf{s}) &= f(\mathbf{s}), & \mathbf{s} \in \mathcal{S}, \\ \mathbf{u}(\mathbf{s}) \cdot \hat{\mathbf{n}}(\mathbf{s}) &= 0, & \mathbf{s} \in \partial\mathcal{S}, \\ \int_{\mathcal{S}} p(\mathbf{s})d\mathbf{s} &= 0, \end{aligned}$$

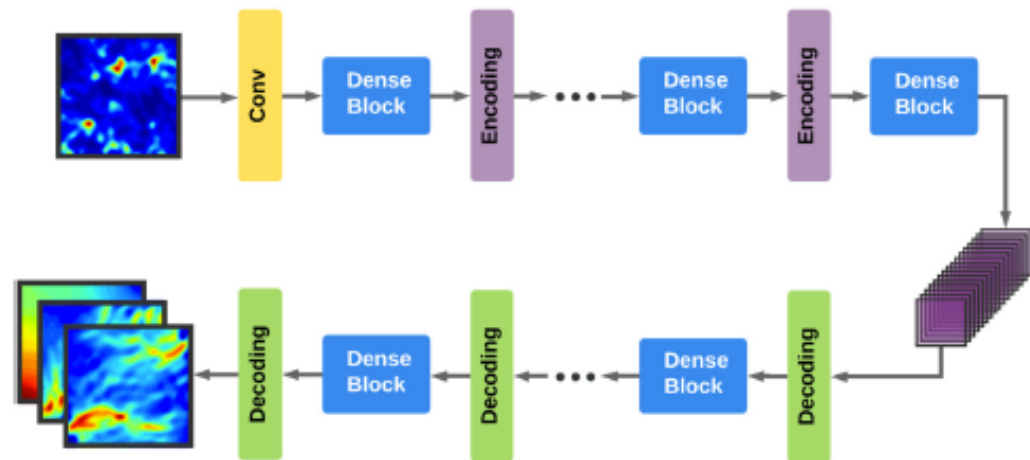
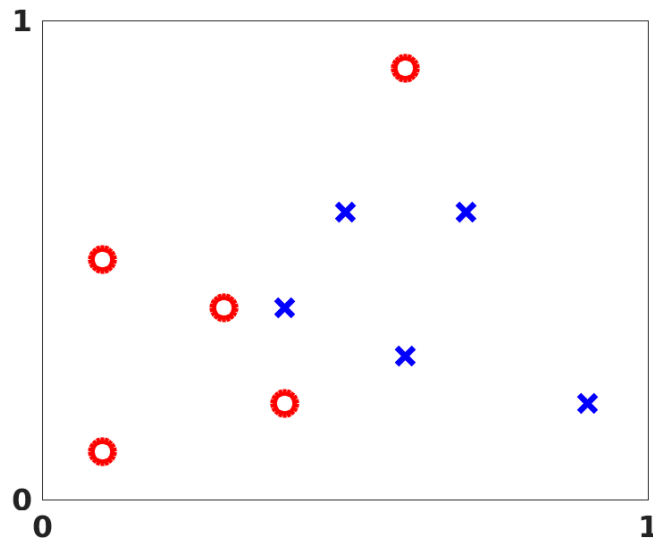


Fig. 3. Network architecture: DenseED.

Bayesian deep convolutional encoder–decoder networks for
surrogate modeling and uncertainty quantification
Y Zhu, N Zabaras
Journal of Computational Physics 366, 415-447

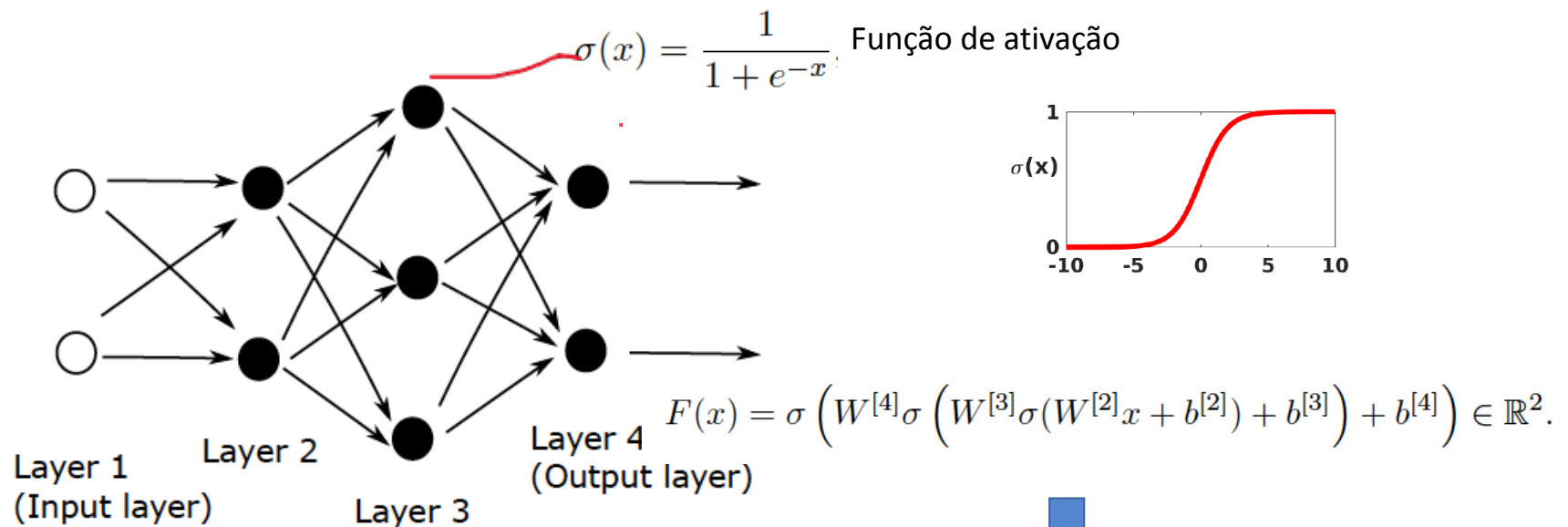
Um modelo para classificação



Catherine F. Highamy, Desmond J. Highamz,
Deep Learning: An Introduction for Applied Mathematicians, SIAM
REVIEW 2019 SIAM. Published by SIAM under the terms
Vol. 61, No. 4, pp. 860–891

“To be concrete, consider the set of points shown in Figure . This shows labeled data some points are in category A, indicated by circles, and the rest are in category B, indicated by crosses. For example, the data may show oil drilling sites on a map, where category A denotes a successful outcome. Can we use this data to categorize a newly proposed drilling site? Our job is to construct a mapping that takes any point in \mathbb{R}^2 and returns either a circle or a cross. Of course, there are many reasonable ways to construct such a mapping.”


Diversidade de possibilidades...



modelo determinado pelas matrizes W^i e vetores b^i (certamente também pela arquitetura da rede)

Racionalizando e treinando a rede (não se trata de um classificador clássico neste exemplo)

- O treinamento consiste em encontrar os 23 parâmetros que definem o modelo. Isto pode ser feito, no contexto de aprendizado supervisionado, através do estabelecimento de um problema de otimização (minimização da distância entre dados e valores obtidos com o modelo, a rede neural).

$$\text{Cost} \left(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]} \right) = \frac{1}{10} \sum_{i=1}^{10} \frac{1}{2} \| y(x^{\{i\}}) - F(x^{\{i\}}) \|_2^2.$$


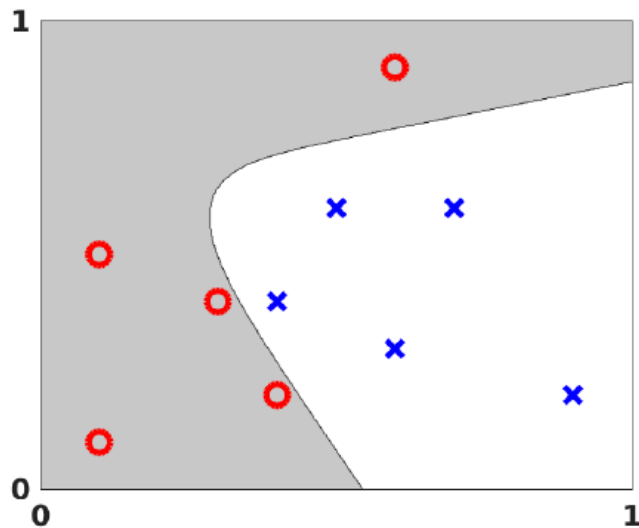
dados

Racional do classificador (um pouco de abstração)

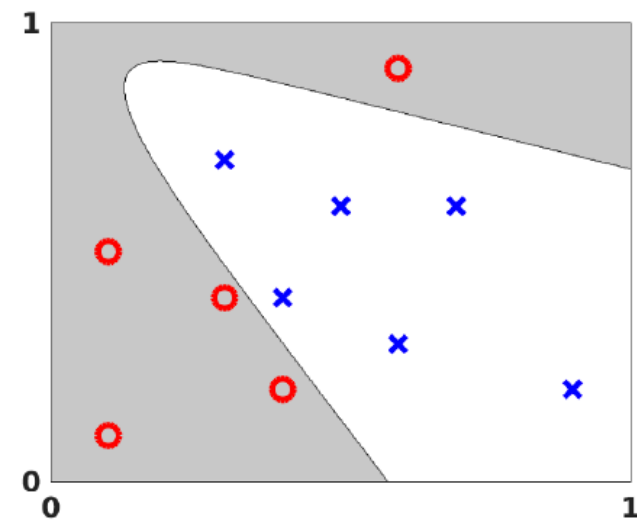
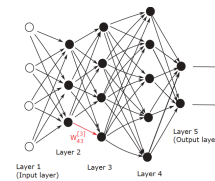
rameters. We will require $F(x)$ to be close to $[1, 0]^T$ for data points in category A and close to $[0, 1]^T$ for data points in category B. Then, given a new point $x \in \mathbb{R}^2$, it would be reasonable to classify it according to the largest component of $F(x)$; that is, category A if $F_1(x) > F_2(x)$ and category B if $F_1(x) < F_2(x)$, with some rule to break ties. This requirement on F may be specified through a *cost function*. Denoting the

$$y(x^{\{i\}}) = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if } x^{\{i\}} \text{ is in category A,} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } x^{\{i\}} \text{ is in category B.} \end{cases}$$

Usando um algoritmo de otimização de sua plataforma (neste caso a função lsqnonlin do Matlab) e extrapolando com o modelo treinado



Com mais dados



e se uma outra arquitetura fosse utilizada?
Balanço entre maior capacidade de representação dos dados e dificuldade em treinar

A estrutura matemática das redes neurais

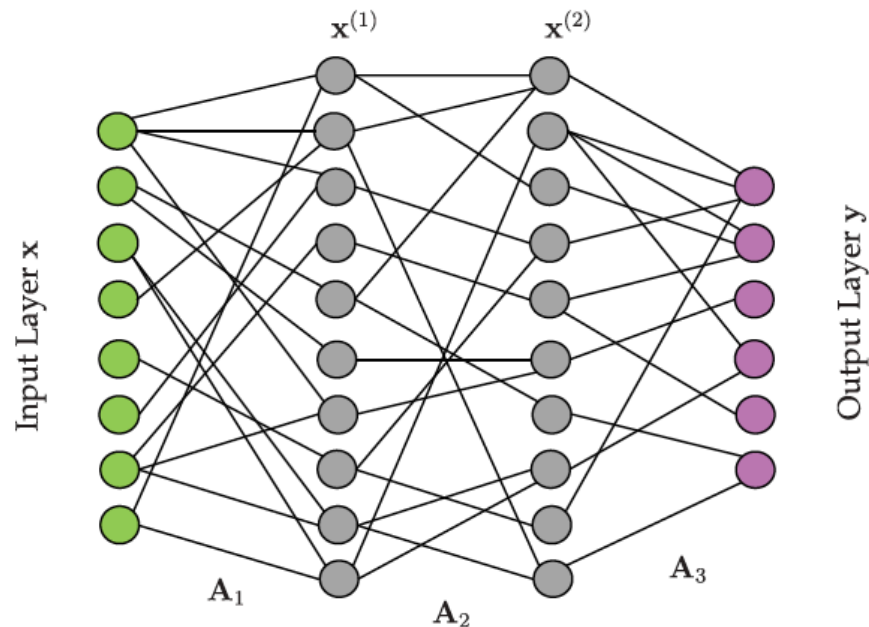
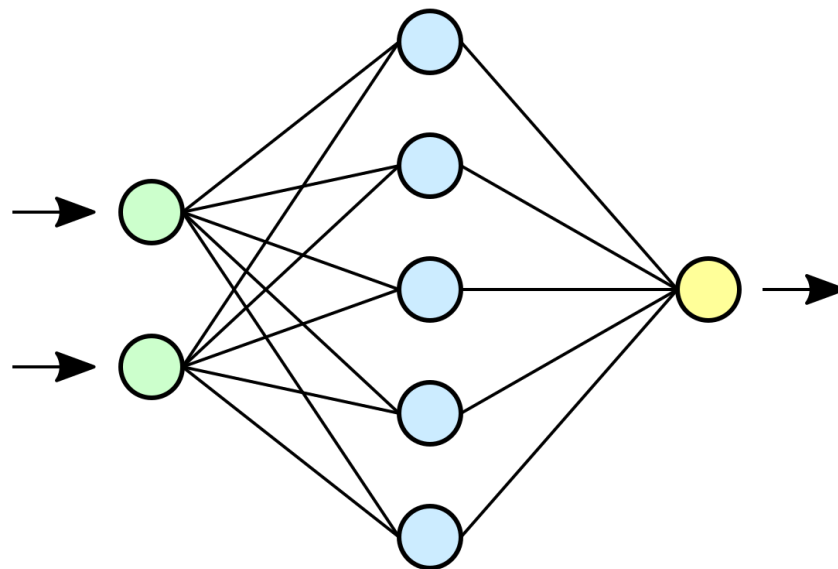


Figure 6.1 Illustration of a neural net architecture mapping an input layer x to an output layer y . The middle (hidden) layers are denoted $x^{(j)}$ where j determines their sequential ordering. The matrices A_j contain the coefficients that map each variable from one layer to the next. Although the dimensionality of the input layer $x \in \mathbb{R}^n$ is known, there is great flexibility in choosing the dimension of the inner layers as well as how to structure the output layer. The number of layers and how to map between layers is also selected by the user. This flexible architecture gives great freedom in building a good classifier.

$$y = f(x; w)$$

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Começemos pelo caso de uma única camada (interna – hidden layer)



Quantos parâmetros são necessários para definir esta rede?

Camada de entrada : $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{z} = W^{(0)}\mathbf{x} + \mathbf{b}^{(0)}$$

$$[W^{(0)}]_{qn} ; \mathbf{b}^{(0)} \in \mathbb{R}^q$$

$$\mathbf{x}^{(1)} = h(\mathbf{z})$$

$$\underbrace{h}_{\text{função de ativação}} : \mathbb{R}^q \rightarrow \mathbb{R}^q$$

$$\mathbf{y} = W^{(1)}\mathbf{z} + \mathbf{b}^{(1)}$$

$$[W^{(1)}]_{mq} ; \mathbf{b}^{(1)} \in \mathbb{R}^m$$

$$\mathbf{y} = W^{(1)}h(W^{(0)}\mathbf{x} + \mathbf{b}^{(0)}) + \mathbf{b}^{(1)}$$

Algumas funções de ativação muito utilizadas

$$f(x) = x \quad - \text{ linear}$$

$$f(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \quad - \text{ binary step}$$

$$f(x) = \frac{1}{1 + \exp(-x)} \quad - \text{ logistic (soft step)}$$

$$f(x) = \tanh(x) \quad - \text{ TanH}$$

$$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \quad - \text{ rectified linear unit (ReLU).}$$