Minicurso Data Science - SENEL 2021

Rômulo Amaral

Apresentação pessoal

Sumário

- O que é Data Science?
- Profissionais que figuram em um time de Data Science
- Conhecimento mínimo - Cientista de Dados
- Conhecimento adquirido na prática
- Tipos de aprendizado
- Machine Learning
- Deep Learning

Código

- Redes Neurais FeedForward
- Redes Neurais Convolucionais
- PCA
- KMeans
- t-SNE

O que é Data Science

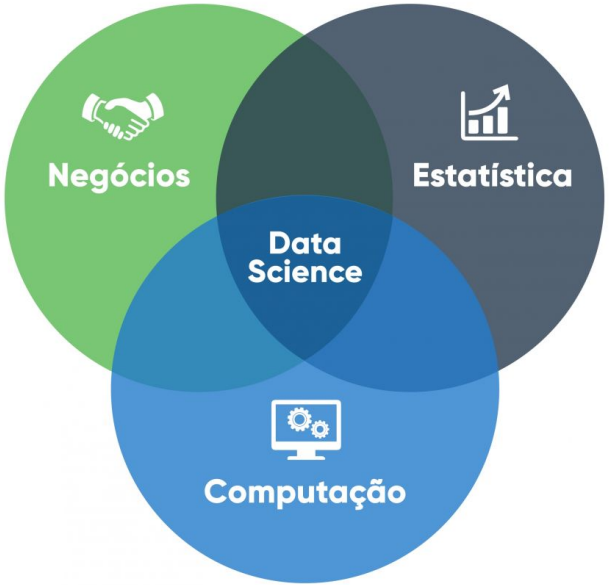


imagem retirada de https://operdata.com.br/

Profissionais que figuram em um time de Data Science

- Cientista de Dados
- Analista de Dados
- Engenheiro de Dados
- Engenheiro de Machine Learning
- Decision Scientists
- Engenheiro de Cloud
- Engenheiro de Software
- Product Owner

Conhecimento mínimo

- Cálculo diferencial e integral
- Álgebra Linear
- Estatística e Probabilidade
- Python ou R
- SQL

Conhecimento mínimo - Cálculo

Conceitos:

1 variável

- Limite -> Para definir derivada e integral
- Derivada (Conceito e regra da cadeia) -> Back propagation (deep learning)
- Integral -> soma contínua

Múltiplas variáveis

- Gradiente -> Otimização, gradiente descendente, ADAM, etc

Conhecimento mínimo - Álgebra Linear

- Vetor
- Matriz
- Tensor
- Produto interno e produto vetorial
- Distância cosseno
- Transformação Linear
- Autovetor e autovalor
- Decomposição espectral

Conhecimento mínimo - Estatística e Probabilidade

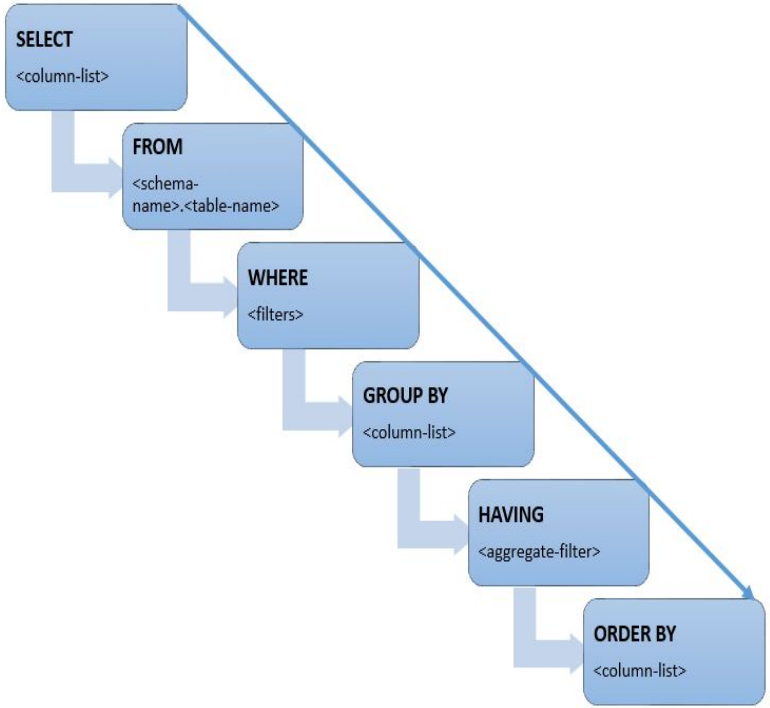
- Variável aleatória
- Dependência e independência
- Teorema de Bayes
- Distribuições
- Histograma, boxplot, etc
- Momentos
- Divergências
- Teste A/B
- Testes de Hipótese
- MCMC

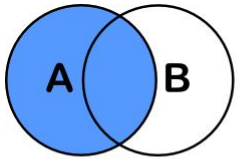
Conhecimento mínimo Python - Bibliotecas

Conhecimento mínimo Python - Linguagem

- Curso básico de Python
- POO
- Estilo “Pythonico”
- PEP8

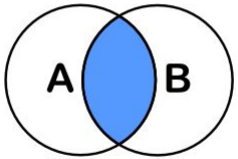
Conhecimento mínimo de SQL



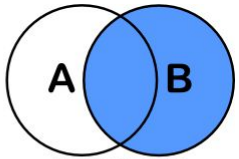


SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key

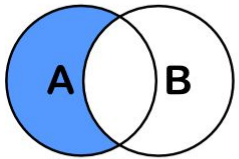
CHEATSHEET
SQL JOINS



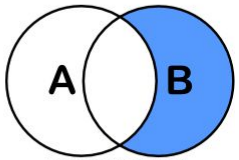
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key



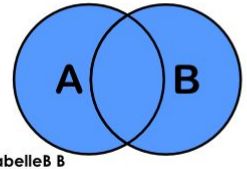
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key



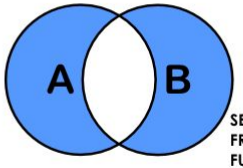
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL



SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL



SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key



SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL

Conhecimento adquirido na prática

Minicurso

Tipos de aprendizado

- Aprendizado Supervisionado
- Aprendizado Não-Supervisionado
- Aprendizado Semi-Supervisionado
- Aprendizado por Reforço

Machine Learning

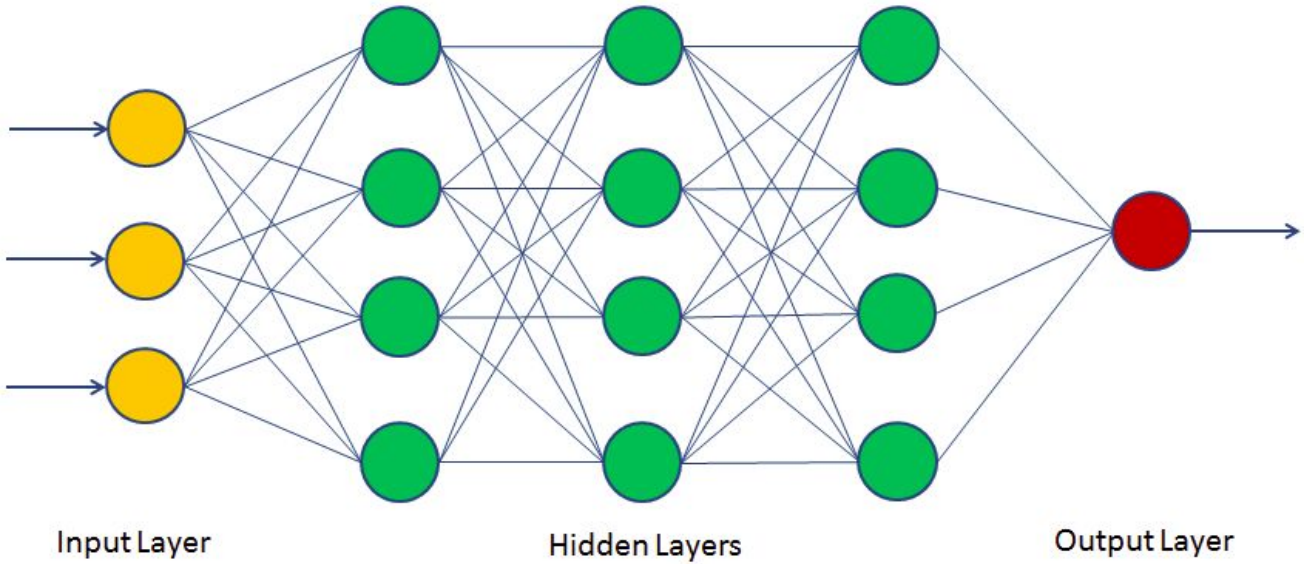
- Regressão linear
- Regressão logística
- SVM
- Random Forest
- Kmeans
- T-sne
- PCA
- XGBoost
- LGBM
- Catboost

Deep Learning

- Redes Neurais
- Redes Neurais Convolucionais
- Redes neurais Recorrentes
- GANs
- Transformers
- Vision Transformers

Redes Neurais FeedForward

Imagem de uma rede neural



Conteúdo de uma rede neural

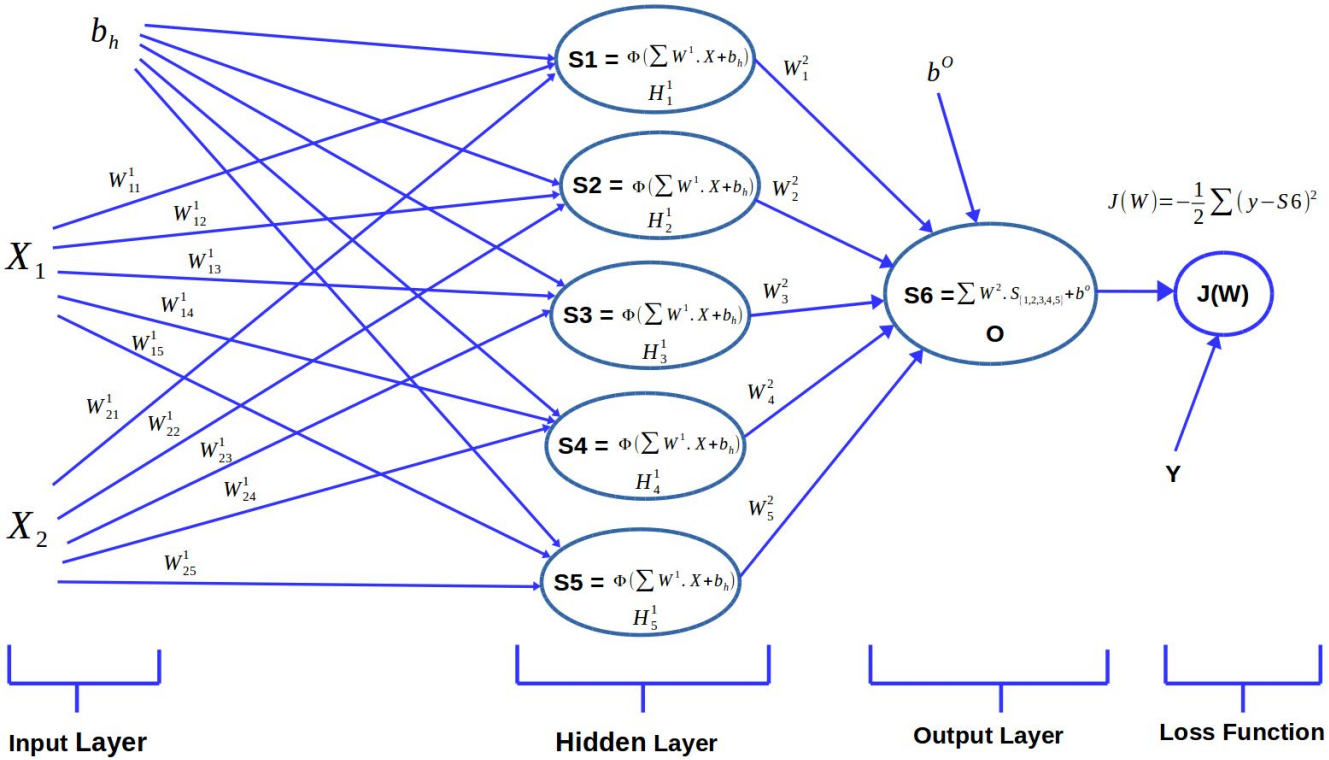
- camada de input
- camada de output
- 1 ou mais camadas intermediárias (2 já vira deep learning)
- camada composta de neurônios
- Os neurônios são conectados com as camadas adjacentes
- Função de ativação
- Função de perdas

Objetivos

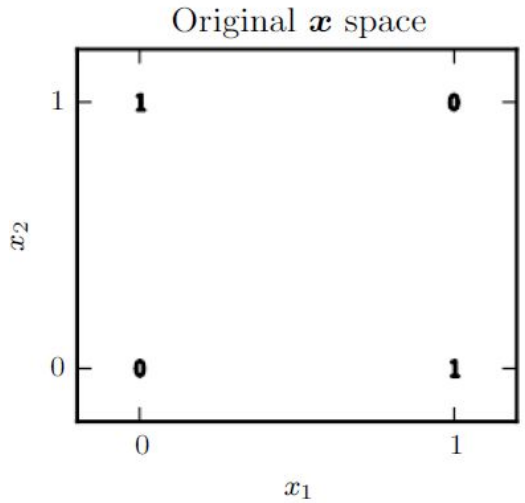
Fazer o mapa do input para o output

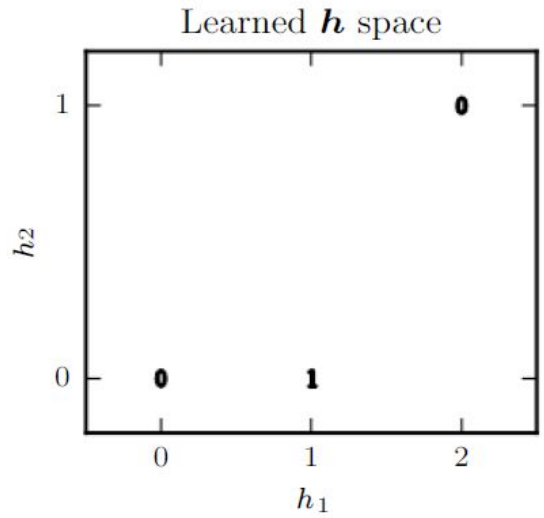
Ou seja, aproximar uma função

Cálculo

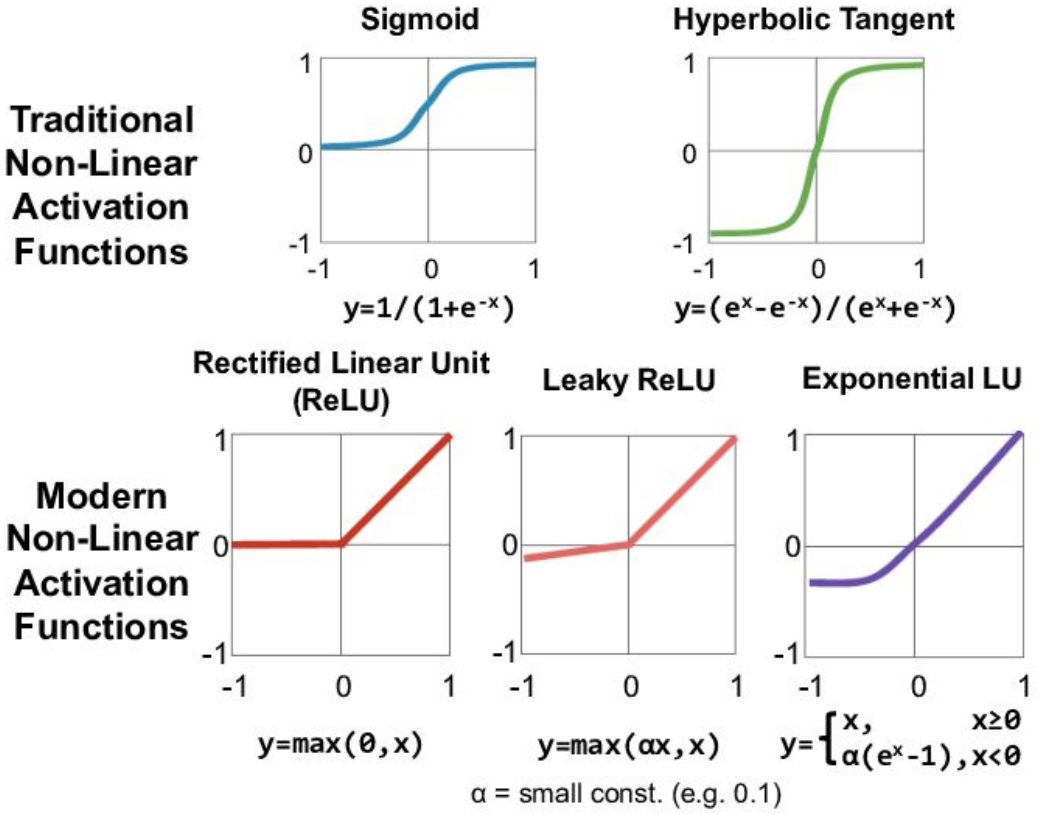

b_h
X_1
X_2
W^1_{11}
W^1_{12}
W^1_{13}
W^1_{14}
W^1_{15}
W^1_{21}
W^1_{22}
W^1_{23}
W^1_{24}
W^1_{25}
S1 = $\Phi(\sum W^1 . X + b_h)$ H^1_1
S2 = $\Phi(\sum W^1 . X + b_h)$ H^1_2
S3 = $\Phi(\sum W^1 . X + b_h)$ H^1_3
S4 = $\Phi(\sum W^1 . X + b_h)$ H^1_4
S5 = $\Phi(\sum W^1 . X + b_h)$ H^1_5
W^2_1
W^2_2
W^2_3
W^2_4
W^2_5
b^O
S6 = $\sum W^2 . S_{|1,2,3,4,5|} + b^o$
O
$J(W) = -\frac{1}{2}\sum (y - S6)^2$
J(W)
Y
Input Layer
Hidden Layer
Output Layer
Loss Function

Por que da função de ativação?




Learned $\boldsymbol{h}$ space
1
0
h_2
0
1
2
h_1
0
0
1

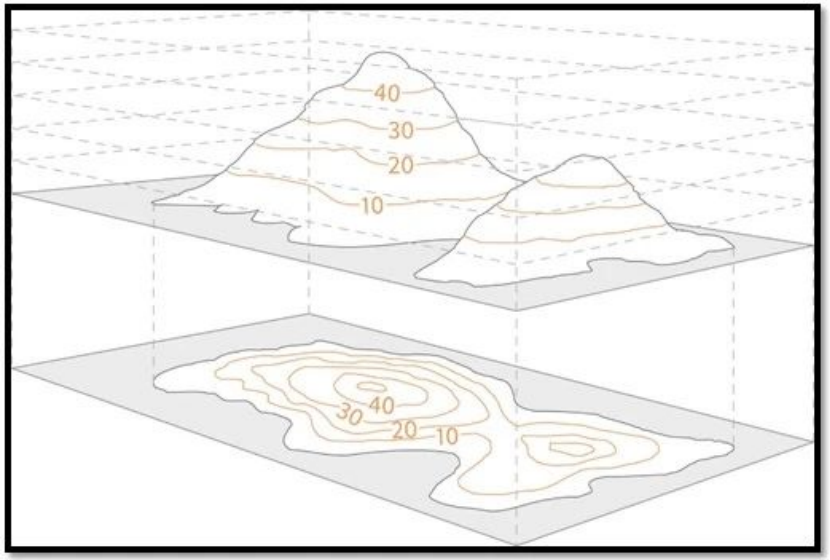
Função de ativação



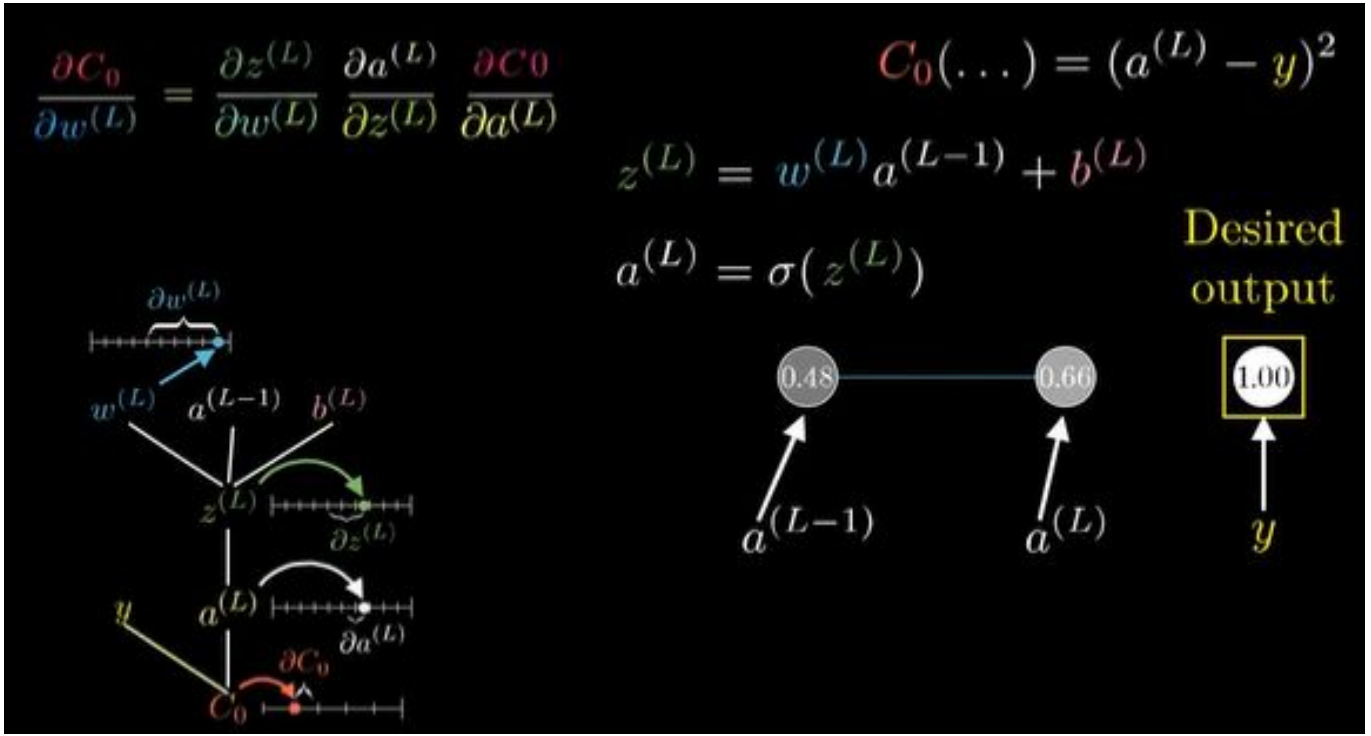
Função de perdas

- **mean_squared_error**
- mean_absolute_error
- mean_absolute_percentate_error
- mean_squared_logarithmic_error
- squared_hinge
- hinge
- categorical_hinge
- logcosh
- **categorical_crossentropy**
- **sparse_categorical_crossentropy**
- **binary_crossentropy**
- kullback_leibler_divergence
- poisson
- cosine_proximity

Backpropagation + gradiente descendente

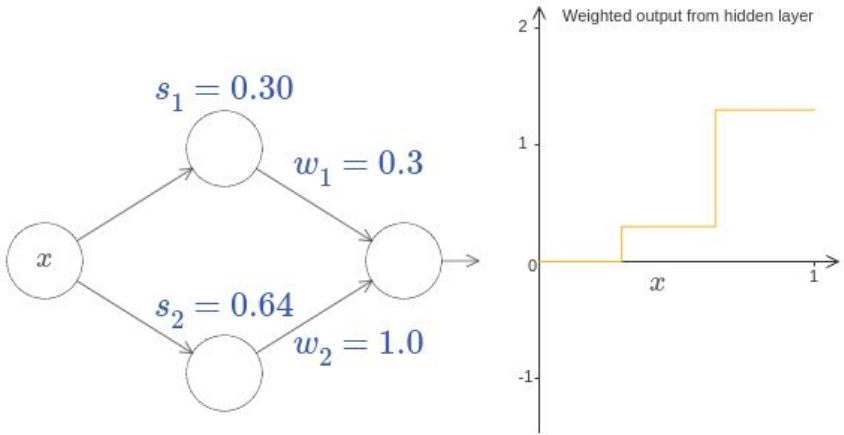


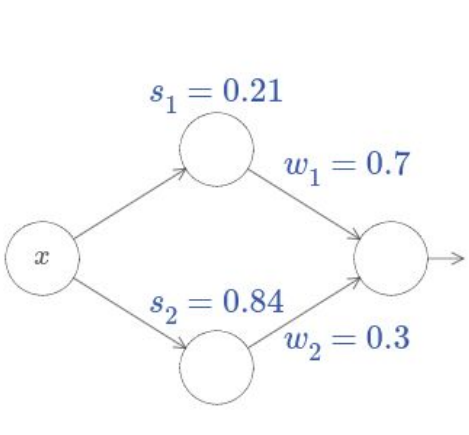
Backpropagation

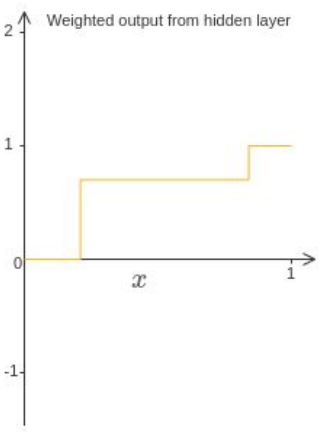


Aproximador universal

link: http://neuralnetworksanddeeplearning.com/chap4.html







Redes Neurais Convolucionais

Operação de Convolução

Exemplo: Sensor e foguete:

Seja x(t) a posição do foguete no tempo t. Podemos tirar uma média das localizações do foguete para uma maior garantia de sua localidade, supondo que as leituras do sensor sejam ruidosas. Faz sentido que leituras recentes tenham um peso maior que leituras antigas.

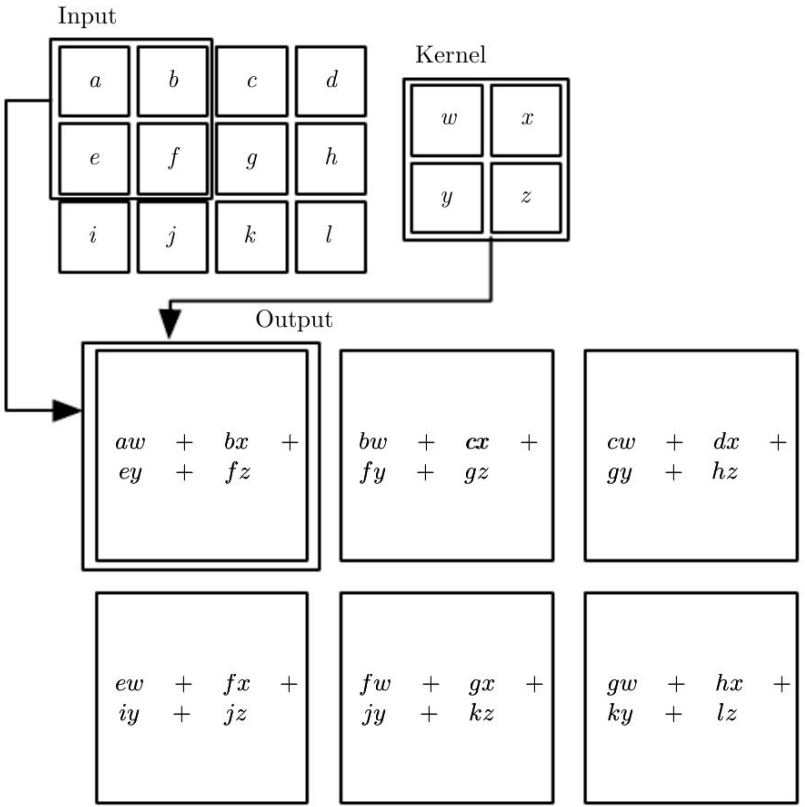
w(t): função de peso

a: idade da medida

Dessa maneira, a estimativa seria:

$$s(t) = \int x(a)w(t-a)da$$

Operação de Convolução



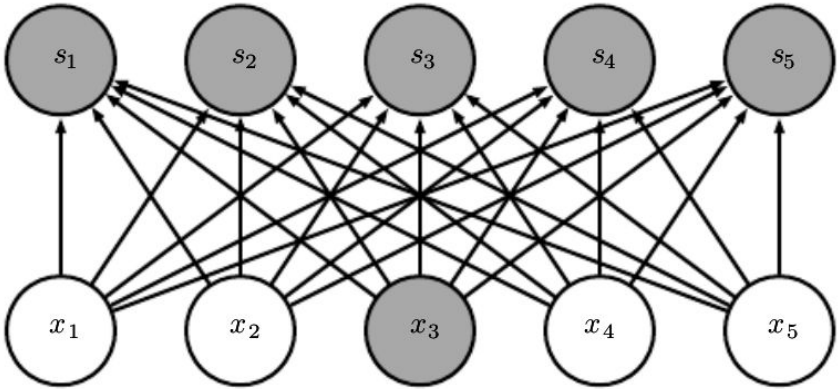
Motivação

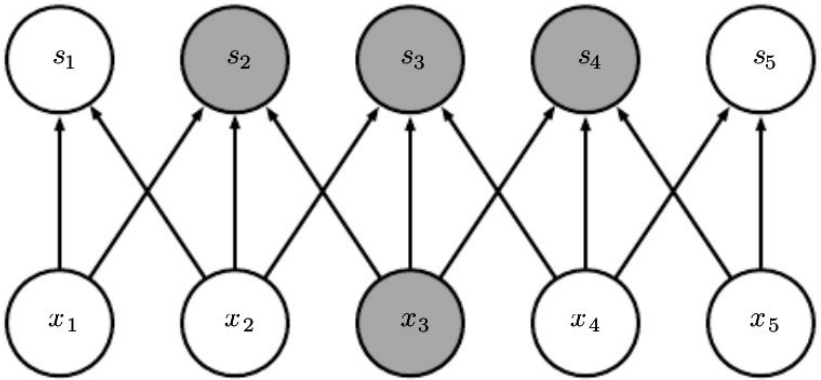
A motivação por trás do uso da operação de convolução (com ou sem o Kernel shiftado) são: Interações esparsas, Compartilhamento de parâmetros e representações equivariantes.

1 - Conexões esparsas: A utilização de um kernel menor que o input faz com que possamos detectar padrões interessantes em pequenas regiões, mesmo que o input seja muito grande. Sendo assim, menos parâmetros precisam ser salvos e ainda há uma diminuição no número de operações necessárias.

2 - Compartilhamento de parâmetros: Em redes feedforward, um dado peso wi do vetor de pesos de um neurônio é utilizado apenas 1 vez, em um determinado elemento e então nunca mais usado. Na convolução, cada elemento do kernel é aplicado em basicamente todos os elementos do input (menos em alguns casos nas fronteiras).

Motivação - Conexões Esparsas e Compartilhamento de parâmetros





Exemplificação da redução de operações - Edge detection

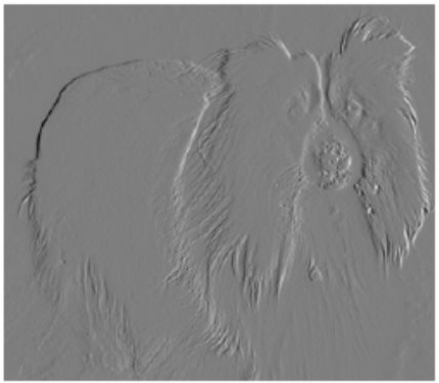
Imagem original: 280 x 320

Imagem das bordas: 280 x 319

Imagem feita realizando apenas a subtração do pixel atual pelo pixel vizinho da esquerda. Mesma coisa que pensar em um kernel com 2 elementos.

Requer apenas 319 * 280 * 3 operações (2 multiplicações e 1 adição)

No esquema de multiplicação de matriz, o número de operações seriam: 280*320*319*280. Convolução cerca de 30 mil vezes menos operações.



Motivação

3 - A convolução é equivariante a translação.

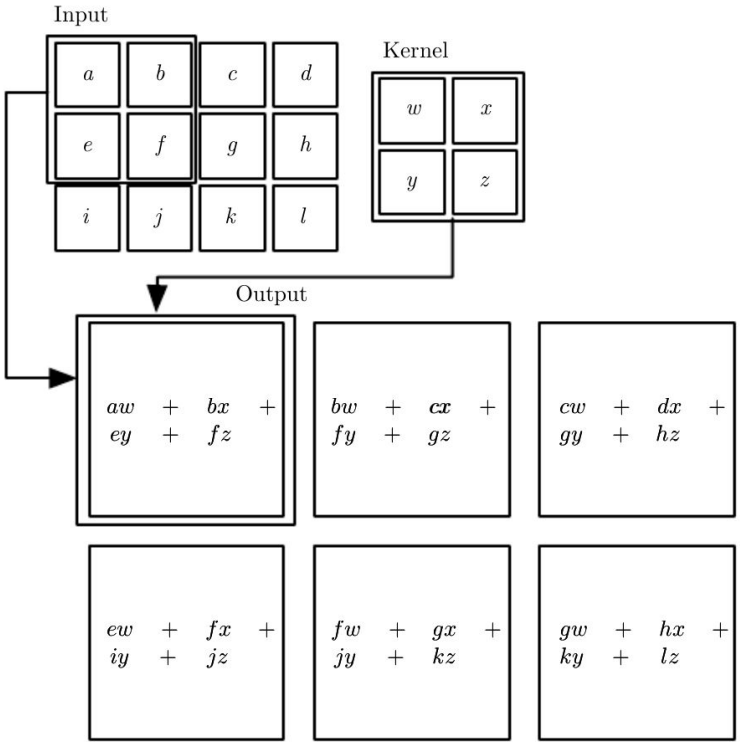
2 funções são equivariantes entre si se:

$$f(g(x)) = g(f(x))$$

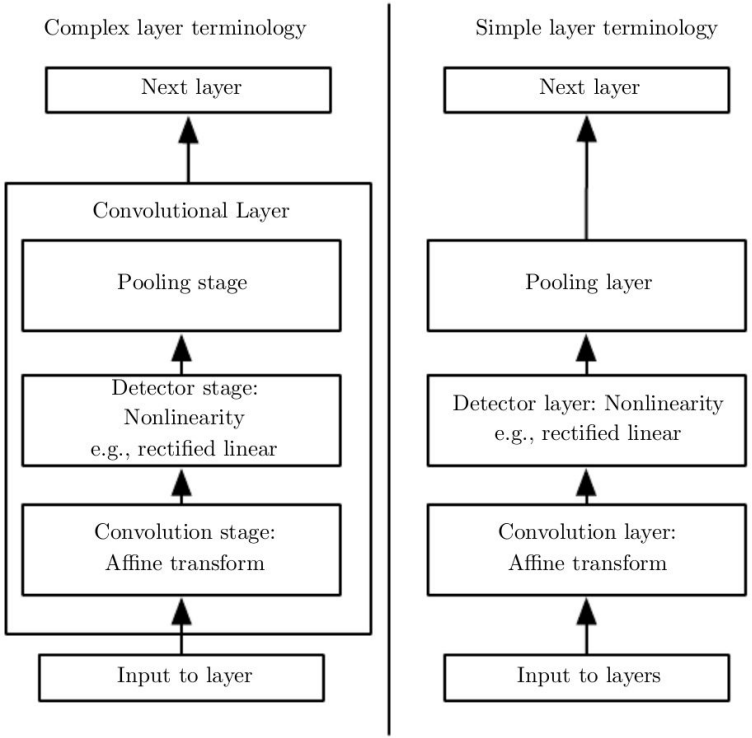
No caso da convolução, se g é uma função que translada a entrada, então a convolução é equivariante a g.

Note que a convolução é equivariante com relação a translação, mas não em mudança de escala e rotação.

Intuição da equivariância com relação a translação



3 estágios básicos de uma CNN



Pooling

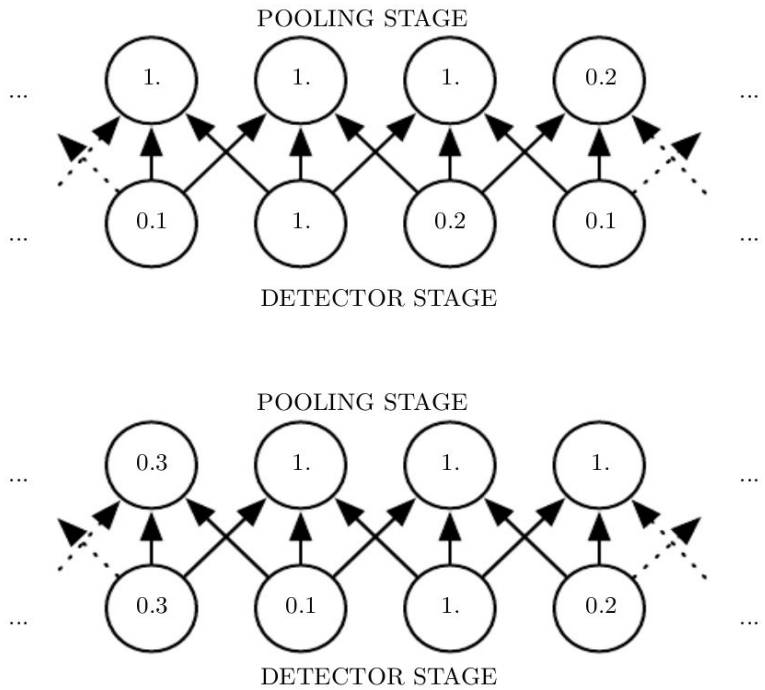
Uma função de “pooling” coloca no lugar do output de uma rede em uma certa localização um “sumário estatístico” dos outputs próximos.

Por exemplo, o “max pooling” retorna o valor máximo do output dentro de uma certa vizinhança retangular. Outras funções de pooling como “média dos elementos do retângulo”, norma, etc também são utilizadas.

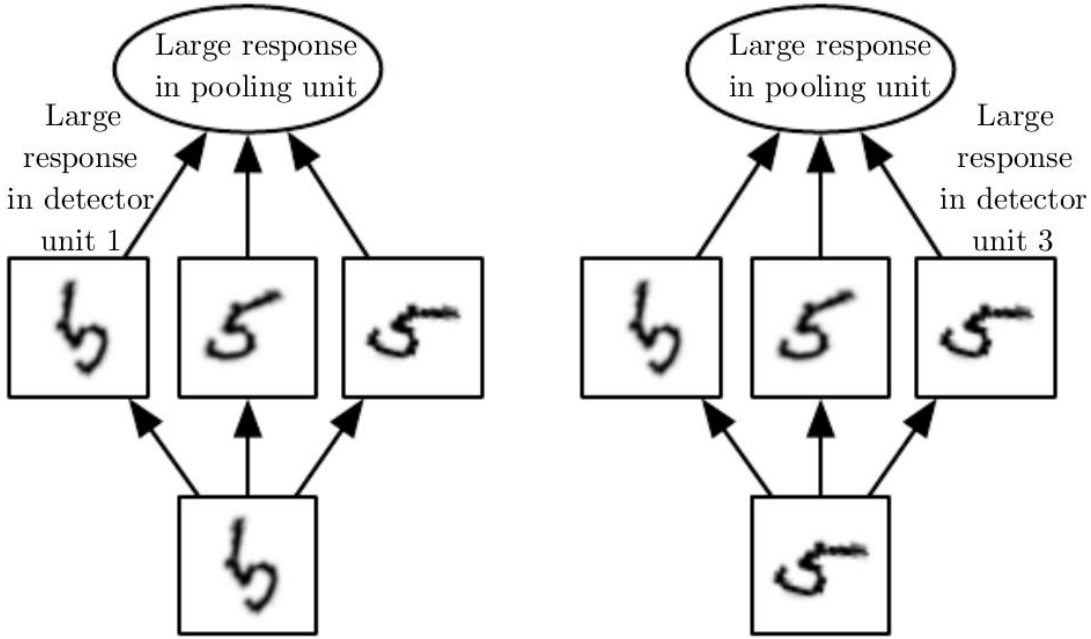
Em todo caso, o pooling ajuda com que a representação se torne invariante a pequenas translações do input. Isso torna importante se estamos mais preocupados se uma certa feature está presente do que se ela está presente em um local específico.

Exemplo max pooling

Figura de baixo é uma translação da primeira camada. Apesar de todos os elementos da camada de baixo terem mudado, apenas metade dos elementos da camada de cima mudou, devido ao max pooling.



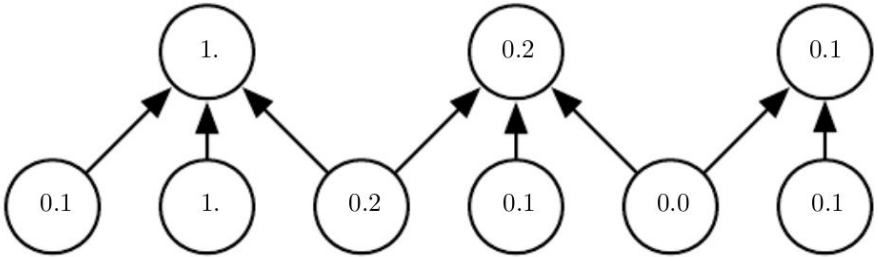
Aprendendo outras invariâncias



Pooling com downsampling

O pooling com downsampling pode ser feito para diminuir ainda mais o gasto computacional.

Também é utilizado para fazer com que inputs de tamanhos diferentes fiquem com tamanhos iguais no layer de classificação, variando então o tamanho do “offset” escolhido.



Stride

Exemplo: Suponha Kernel como um tensor 4D (elementos **K**i,j,k,l)

Input dado **V**i,j,k (imagem com 3 canais, por exemplo)

O offset entre output e input é de k linhas e l colunas

O output é Z com o mesmo formato que V.

Dessa forma Z fica:

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n}$$

Stride

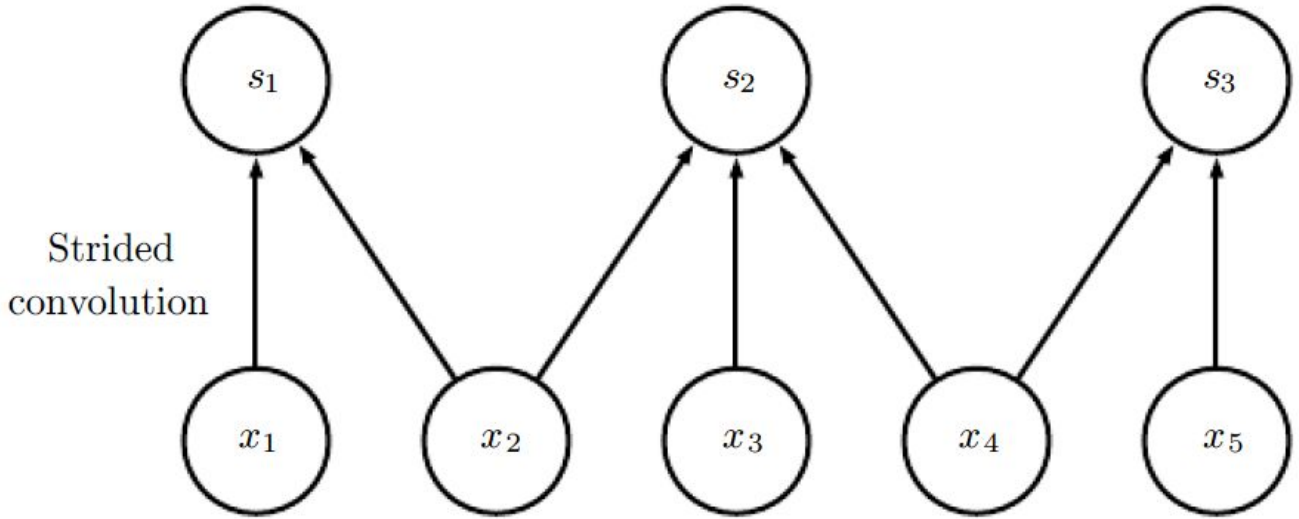
O Stride é utilizado para se reduzir o custo computacional. Dessa forma algumas posições serão puladas.

Dessa forma o novo output fica:

$$Z_{i,j,k} = c(\mathbf{K}, \mathbf{V}, s)_{i,j,k} = \sum_{l,m,n} \left[V_{l,(j-1)\times s+m,(k-1)\times s+n} K_{i,l,m,n} \right]$$

Dessa maneira o “s” é referido como stride e é utilizado como uma forma de downsampling pra convolução, a fim de economizar poder computacional.

Stride



Zero-padding

Zero-padding é utilizado para fazer com que o input fique mais largo, para que então se possa ser controlado a largura do output. Sem isso, a dimensão fica reduzida de 1 a menos que a dimensão do kernel a cada camada que é feita a convolução.

3 casos extremos de zero-padding:

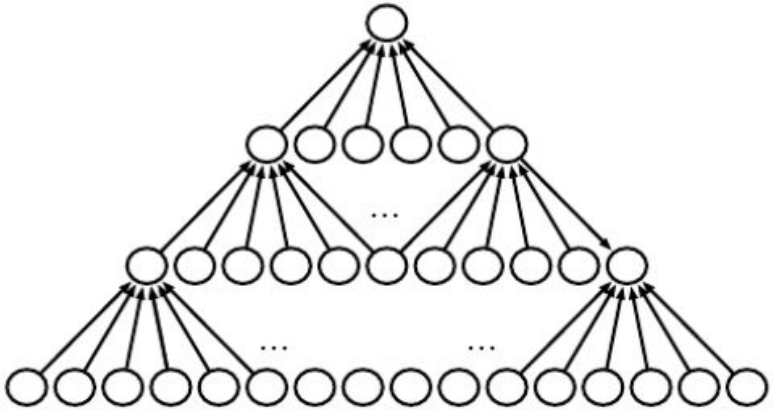
1 - Valid Convolution: Número de convoluções limitadas por causa da redução do número de pixels

2 - Same Convolution: Utilização do número máximo de convoluções (borda fica comprometida)

3 - Full convolution:

Zero-padding

No padding

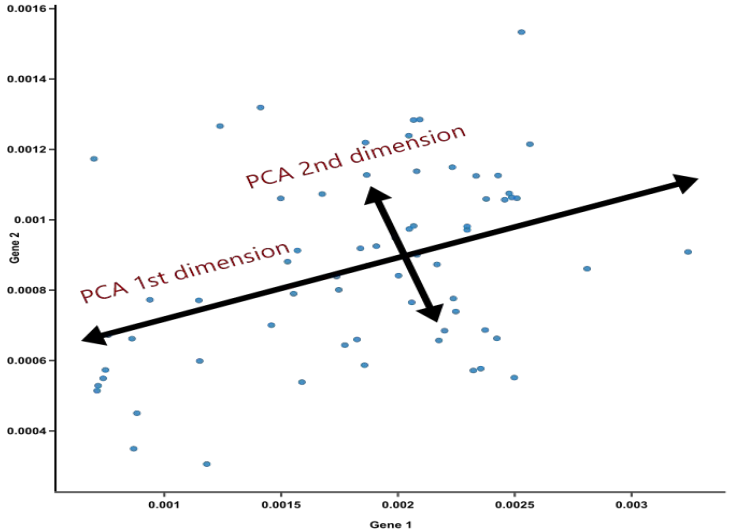


Zero-padding

PCA

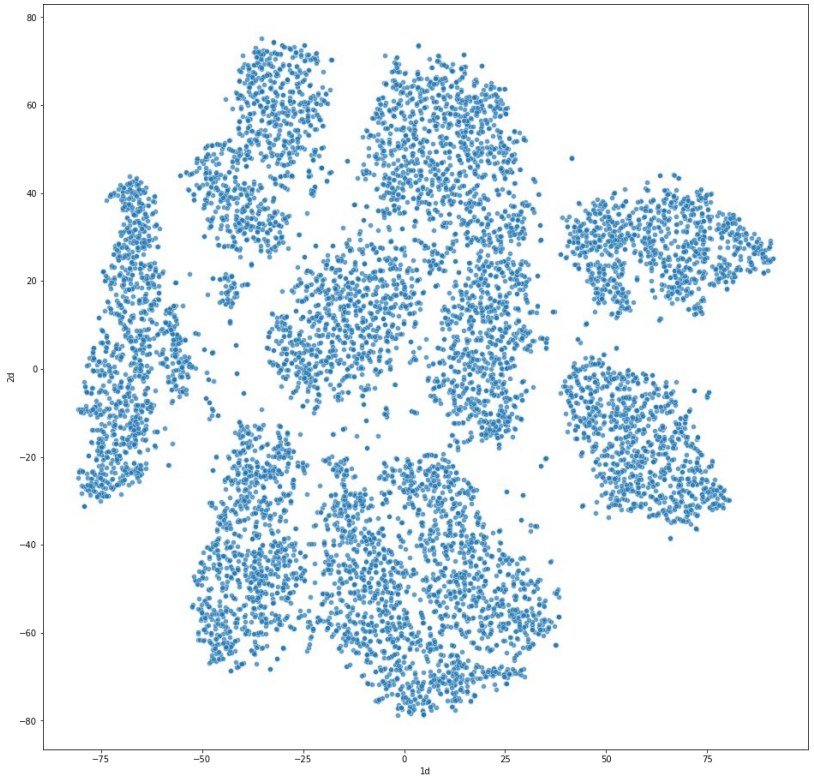
PCA

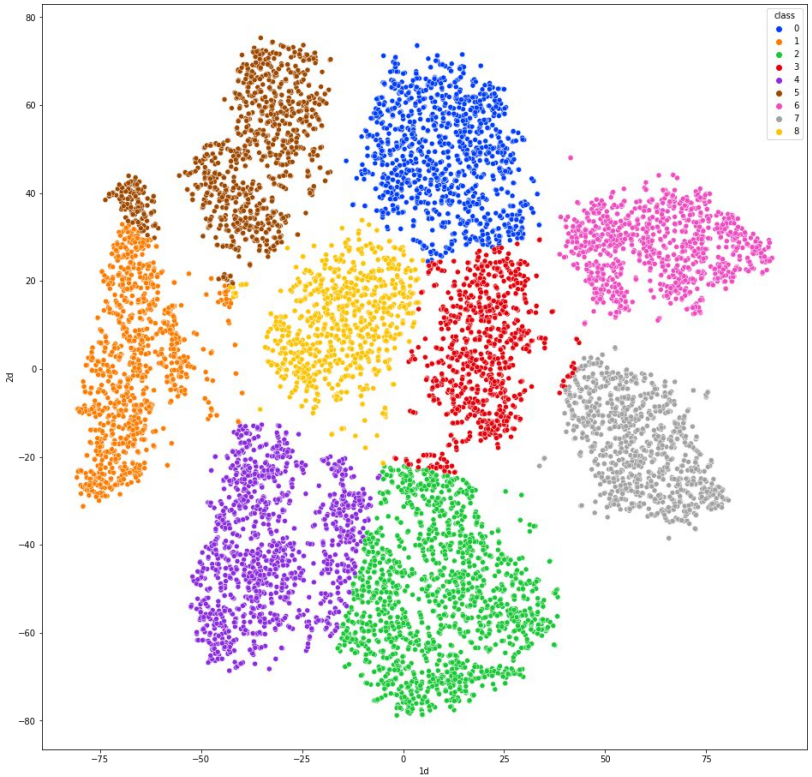
- Novas componentes escritas como combinações lineares das componentes antigas
- Componentes em ordem com relação a maior explicabilidade da variância
- Componentes são ortogonais



Kmeans

Kmeans


80
60
40
20
0
−20
−40
−60
−80
2d
−75
−50
−25
0
25
50
75
1d

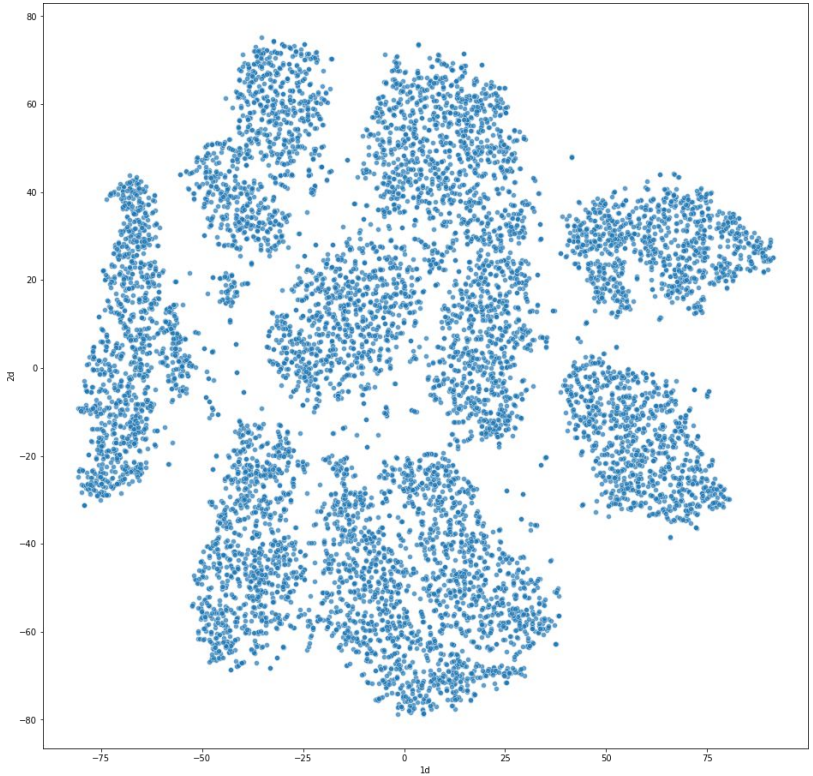

class
0
1
2
3
4
5
6
7
8
80
60
40
20
0
−20
−40
−60
−80
2d
−75
−50
−25
0
25
50
75
1d

Kmeans

- Processo interativo onde se é calculado a distância entre cada ponto e o centróide
- Ponto pertence a classe com o centróide mais próximo
- Número de classes determinado por algumas heurísticas como “elbow curve”, método da silhueta, etc.

t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-SNE



t-SNE

Ideia:

Pontos similares devem ficar próximos após o mapa

- Cada par de pontos possui uma distribuição de probabilidade associada, de tal maneira que pontos similares possuem alta probabilidade
- Faz a mesma coisa para o mapa de 2-3 dimensões
- Após isso, minimiza-se a divergência KL para os pares.

Referências

- GOODFELLOW, I. J., BENGIO, Y., COURVILLE, A. Deep Learning Cambridge, MA, USA, MIT Press, 2016.
- van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). "Visualizing Data Using t-SNE" (PDF). *Journal of Machine Learning Research*. **9**: 2579–2605.
- I. T. Jolliffe. Principal Component Analysis and Factor Analysis, pages 115–128. Springer New York, New York, NY, 1986. https://doi.org/10.1007/978-1-4757-1904-87doi : 10.1007/978 − 1 − 4757 − 1904 − 87.
- MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201.