



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
CAMPUS FORTALEZA

APRENDIZADO DE MÁQUINA

RAIANE ROCHA REIS

RELATÓRIO: MÉTODOS DE CLASSIFICAÇÃO

FORTALEZA

2023

SUMÁRIO

1	INTRODUÇÃO	2
2	METODOLOGIA	3
2.1	MLP (Perceptron Multicamadas)	3
2.2	Naive Bayes	3
2.3	Support Vector Machine (SVM)	4
2.3.1	<i>kernel RBF</i>	5
2.3.2	<i>kernel Polinomial</i>	5
2.3.3	<i>kernel Linear</i>	5
3	RESULTADOS E DISCUSSÃO	6
3.1	Seção 1	6
3.2	Seção 2	6
3.3	Seção 3	6
4	CONCLUSÃO	7
	REFERÊNCIAS	8

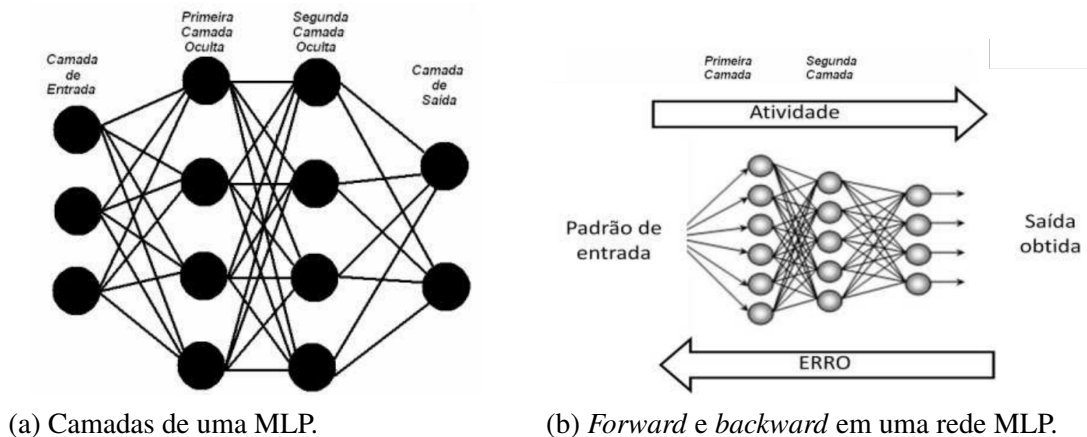
1 INTRODUÇÃO

2 METODOLOGIA

2.1 MLP (Perceptron Multicamadas)

Como é mostrado na Figura 1a, uma Rede Neural Multicamadas (MLP – *MultiLayer Perceptron*) é formada por um conjunto de neurônios, também conhecidos como *Perceptrons*. Uma MLP consiste em uma camada de entrada, juntamente com uma ou mais camadas ocultas. No processo de treinamento, é empregada uma técnica chamada retropropagação (*backpropagation*), que ocorre em duas fases distintas: a propagação para frente (*forward*) e a retropropagação propriamente dita (*backward*), assim como ilustra a Figura 1b. Durante a propagação para frente, os dados são passados pela rede, camada por camada, permitindo que as saídas da rede sejam calculadas. Em seguida, durante a fase de retropropagação, os erros entre as saídas previstas e os valores reais são calculados e propagados de volta através da rede, ajustando os pesos das conexões para minimizar esses erros. Esse processo iterativo é fundamental para o treinamento eficaz de uma MLP, permitindo que ela aprenda e se adapte (ORRù *et al.*, 2020).

Figura 1 – Estrutura e atividade de uma rede MLP, imagens de (ORRù *et al.*, 2020).



2.2 Naive Bayes

Naive Bayes é um método de classificação e tem como base o teorema de Bayes. Este método gera uma tabela de probabilidades utilizadas para classificar os dados, as *features* dos dados são analisadas de forma independente, por isso o nome *Naive*, que significa ingênuo. A Equação 2.1, abaixo, descreve o funcionamento desse método de classificação (MAHESH, 2018).

- Teorema de Bayes:

$$P(c|x) = P(c)P(x|c)/P(x) \quad (2.1)$$

- $P(c|x)$: **Probabilidade Posterior**, probabilidade do dado (x) pertencer a classe (c);
- $P(c)$: **Probabilidade Anterior**, probabilidade da classe (c) ocorrer antes do dado (x);
- $P(x|c)$: **Verossimilhança**, caso a classe (c) seja verdadeira, a probabilidade (x) ocorrer em (c);
- $P(x)$: **Probabilidade marginal x**, probabilidade de (x) ser observado independente de (c).

2.3 Support Vector Machine (SVM)

É uma técnica supervisionada utilizada para classificação e regressão. Esse método cria um vetor que divide as classes com margens, para minimizar os erros (MAHESH, 2018). A Figura 2, mostra como a divisão de classes acontece.

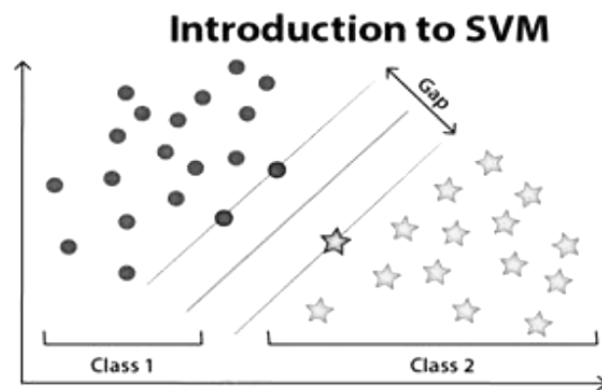


Figura 2 – Divisão de classes por vetores, Figura de (MAHESH, 2018).

Kernel é uma função responsável por calcular a similaridade entre pares de conjuntos, as entradas, e o resultado são números escalares que representam o grau semelhança entre as amostras (NORONHA; FERNANDES, 2016). Uma função é chamada de *kernel* quando satisfaz as condições do teorema de mercer. De acordo com esse teorema se a matriz k é positivamente definida, ou seja k é a matriz do produto interno dos dados de entrada (FALCÃO *et al.*, 2017).

2.3.1 *kernel RBF*

Os *kernels* Função de base radial (RBF - *radial basis function*) Aplica uma transformação nos pontos utilizando uma função gaussiana, definida pela a Equação 2.2 (ALMEIDA *et al.*, 2020).

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\alpha^2}\right) \quad (2.2)$$

2.3.2 *kernel Polinomial*

Kernel Polinomial, podem introduzir curvaturas no hiperplanos, ou seja, são capazes de separar os dados de maneira não linear. Como mostra a Equação 2.3 Além das amostras, este *kernel* depende de uma variável **d** (grau do polinômio) (NORONHA; FERNANDES, 2016).

$$k(x_i, x_j) = [(x_i \cdot x_j) + 1]^d \quad (2.3)$$

2.3.3 *kernel Linear*

O *kernel* linear tem como base a equação da reta (Equação 2.4), portanto divide as classes de maneira linear, o que trás uma limitação se comparado aos *kernels* já apresentados neste trabalho (ALMEIDA *et al.*, 2020).

$$k(x_i, x_j) = x_i \cdot x_j \quad (2.4)$$

3 RESULTADOS E DISCUSSÃO

3.1 Seção 1

3.2 Seção 2

3.3 Seção 3

4 CONCLUSÃO

REFERÊNCIAS

- ALMEIDA, A.; CARVALHO, F.; MENINO, F. **Introdução ao machine learning**. 2020. Disponível em: <<https://dataat.github.io/introducao-ao-machine-learning/regress%C3%A3o.html#kernels>>.
- FALCÃO, V. B. *et al.* Aplicação de metodologias de reconhecimento de padrões para detecção de dano em vigas de concreto. Universidade Federal de Alagoas, 2017.
- MAHESH, B. Machine learning algorithms - a review. **International Journal of Science and Research**, 2018.
- NORONHA, D. H.; FERNANDES, M. A. Implementação em fpga de máquina de vetores de suporte (svm) para classificação e regressão. **XIII Encontro Nacional de Inteligência Artificial e Computacional-ENIAC**, 2016.
- ORRù, P. F.; ZOCCHEDDU, A.; SASSU, L.; MATTIA, C.; COZZA, R.; ARENA, S. Machine learning approach using mlp and svm algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. **Sustainability**, v. 12, 2020.