

Implementação e Validação do Perceptron

Jesuino Vieira Filho
Centro Tecnológico de Joinville
Universidade Federal de Santa Catarina
Joinville, Brasil
jesuino.vieira@grad.ufsc.com

Resumo—Este relatório tem por objetivo descrever as atividades desenvolvidas durante a implementação e validação de uma rede neural artificial do tipo perceptron. O perceptron pode ser utilizado para a classificação de problemas binários linearmente separáveis. Para sua validação, foi realizado dois experimentos em diferentes base de dados (uma linearmente separável, e outra não). Tal procedimento permitiu avaliar o impacto de diferentes taxas de aprendizado para o treinamento, assim como confirmar a ineficácia deste tipo de abordagem para problemas não lineares.

Index Terms—rede neural artificial, perceptron

I. INTRODUÇÃO

O perceptron é a forma mais simples de uma rede neural artificial (RNA) usada para a classificação de padrões considerados linearmente separáveis, ou seja, padrões que ficam em lados opostos de um hiperplano [1]. Com o objetivo de aprofundar os conhecimentos sobre este tipo de RNA, este trabalho abrange a implementação e a validação de um perceptron de um único neurônio para a classificação de problemas binários. Os experimentos desenvolvidos para validar a implementação permitiram avaliar o impacto de diferentes taxas de aprendizado para o treinamento, assim como a inaptidão do perceptron para problemas não lineares.

O presente relatório encontra-se organizado da seguinte forma. Na Seção II, aspectos relacionados a implementação do perceptron são brevemente descritos, assim como os detalhes acerca dos dois experimentos realizados. Na Seção III, os resultados obtidos são relatados de forma simples e objetiva. Por fim, na Seção IV são apresentadas as interpretações dos resultados e considerações finais acerca do referido trabalho.

II. MATERIAIS E MÉTODOS

Esta seção é dividida em duas partes. Primeiro, detalhes acerca da implementação (com foco no treinamento) são brevemente apresentados. Em sequência, os experimentos realizados para validação são descritos com mais detalhes.

A. Implementação

A implementação do perceptron foi desenvolvida utilizando a linguagem de programação Python, em sua versão 3.9.6. Para tanto, um módulo denominado `perceptron` foi implementado. Neste módulo, uma classe denominada `Perceptron` permite inicializar, configurar, treinar, avaliar e utilizar a rede através de uma interface similar a da biblioteca TensorFlow (popular escolha para o desenvolvimento de aplicações de aprendizado de máquina), como evidenciado no Algoritmo 1.

Algoritmo 1: Exemplo de uso do módulo implementado.

```
m = perceptron.Perceptron()
m.compile(eta=eta, loss="mse", tol=0.0)
m.fit(X_train, y_train, epochs=50)
acc, loss = m.evaluate(X_test, y_test)
```

O usuário é capaz de configurar o modelo para a etapa de treinamento em tempo de execução, como pode ser observado na chamada dos métodos `compile` e `fit`. Nessa situação, os seguintes parâmetros podem ser controlados: a **taxa de aprendizado** (`eta`), a **função de custo** (`loss`), a **tolerância** (`tol`) e o número de **épocas de treinamento** (`epochs`).

Para treinar o perceptron, o método online de aprendizagem supervisionada foi utilizado. Isto significa que os ajustes aos pesos sinápticos são realizados exemplo a exemplo, de acordo com a Equação 1

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n) \quad (1)$$

onde \mathbf{w} é o vetor de pesos sinápticos, η é a taxa de aprendizado, d é a saída desejada, y é a saída obtida e \mathbf{x} é o vetor de entradas. A etapa de treinamento é dita concluída em duas situações: quando o número de épocas é atingido ou o resultado da função de custo é menor do que a tolerância.

A implementação da rede neural perceptron foi baseada nas definições fornecidas por Haykin [1]. Desse modo, vale ressaltar que os pesos são inicializados seguindo uma distribuição uniforme, com valores entre 0 e 1. Além disso, antes de cada época os dados são embaralhados. Por fim, dado que apenas um neurônio foi considerado, a implementação pode ser utilizada apenas para a classificação de problemas binários. Ou seja, a saída é capaz de prever apenas duas classes.

B. Experimentos

Com o objetivo de validar a implementação, repetiu-se os experimentos descritos no Capítulo 1.5 de Haykin [1]. Dessa forma, aspira-se:

- 1) avaliar o impacto da variação da taxa de aprendizado no processo de treinamento da RNA do tipo perceptron;
- 2) avaliar o desempenho da RNA do tipo perceptron para problemas linearmente separáveis;
- 3) e avaliar o desempenho da RNA do tipo perceptron para problemas não linearmente separáveis.

Para facilitar a compreensão, nomeia-se os experimentos realizados como **Experimento 1** (destinado aos objetivos 1 e

2) e **Experimento 2** (destinado ao objetivo 3). A base de dados utilizada em ambos experimentos (conhecida como *double-moon*) é representada por dois atributos, os quais formam um par de “luas” frente a frente de maneira assimétrica quando representados no plano bidimensional, como ilustrado na Figura 1. Cada lua pertence a uma classe, e a distância vertical d que separa as duas luas é ajustável.

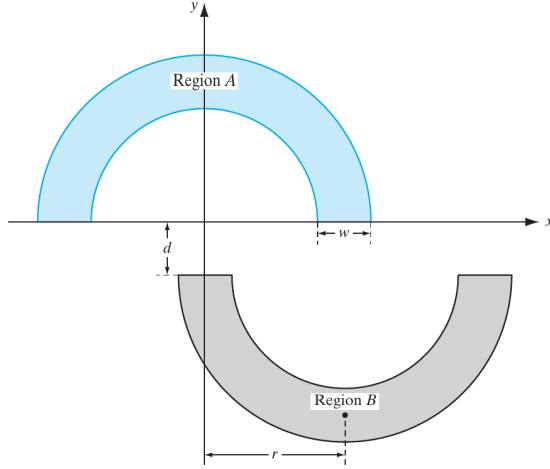


Figura 1: O problema de classificação *double-moon* [1].

Desta forma, duas situações foram analisadas: $d = 1$ (Experimento 1) e $d = -4$ (Experimento 2). Note que, com uma distância negativa, tem-se um problema não linearmente separável. De fato, um dos objetivos dos experimentos é confirmar a ineficácia do perceptron para este tipo de problema. Considerando a situação em que $d = 1$, o impacto de diferentes taxas de aprendizado foi analisado. Neste caso, cinco valores foram utilizados, variando linearmente de $1e-5$ à $1e-0$. A Tabela I facilita a visualização dos experimentos realizados.

Tabela I: Sumário dos experimentos realizados e parâmetros utilizados. “L. Sep.” significa linearmente separável e “mse” significa erro quadrático médio.

Experimento	L. Sep.	η	tol	loss	epochs
1	Sim	$1e-5, \dots, 1e-0$	0	mse	50
2	Não	$1e-5$	0	mse	50

Assim como em [1], optou-se por utilizar uma base de dados com 3000 pontos, em que 1/3 é utilizado para treino e 2/3 para teste. Tal separação foi realizada de maneira estratificada para manter o balanceamento entre as diferentes classes. Além disso, ambas as bases de dados foram geradas manualmente.

III. RESULTADOS

De modo a facilitar a apresentação dos resultados, a seção atual é dividida em duas partes. Primeiro, os resultados relacionados ao Experimento 1 são apresentados, e em seguida, os do Experimento 2.

A. Experimento 1

Na Figura 2 são apresentados as curvas de aprendizado para os cinco valores de taxa de aprendizado utilizados, os quais variam linearmente de $1e-5$ à $1e-0$. Nota-se que todos os modelos convergiram para um erro quadrático médio (função de custo utilizada) igual a zero. Ou seja, independente da taxa de aprendizado foi possível resolver o problema e definir um hiperplano que separa perfeitamente as duas classes.

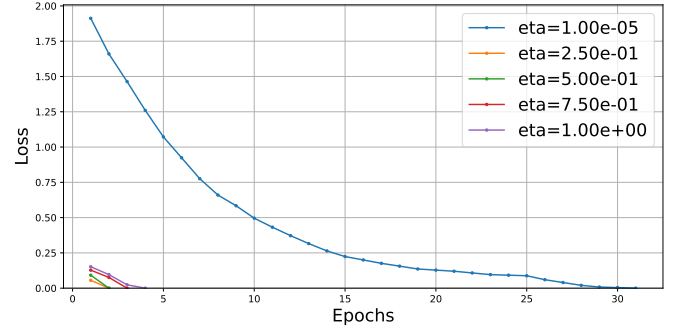


Figura 2: Curva de aprendizado para diferentes valores de taxa de aprendizado. Valores referentes ao Experimento 1.

Para o modelo com menor valor de taxa de aprendizado ($1e-5$), observa-se que a convergência é atingida apenas na 31ª época. Isto é resultado de uma menor taxa de atualização dos pesos sinápticos – função taxa de aprendizado na Equação 1.

É interessante notar, porém, que para os demais quatro valores da taxa de aprendizado, quanto maior este valor ficou, mais o modelo demorou para convergir. De fato, é importante lembrar que o número de épocas que o modelo leva para convergir é significativamente afetado pelo peso inicial das conexões sinápticas, os quais foram inicializados de maneira uniforme, com valores entre 0 e 1.

Não obstante, o modelo com taxa de aprendizado $1e-5$ reportou uma acurácia de 99.8% no conjunto de teste, sendo o único modelo que não obteve 100% de aproveitamento neste experimento. Nota-se que, eventualmente, alguns objetos do conjunto de teste não foram representados no espaço de atributos de treinamento, levando a falha de classificação para estes casos extremos – também conhecidos como *edge cases*.

B. Experimento 2

Na Figura 3 é apresentado a curva de aprendizado do modelo obtido no Experimento 2. Nota-se que, aproximadamente na 10ª época da etapa de treinamento, o modelo converge para um ponto ótimo e nenhuma melhoria é observada.

De fato, a base de dados utilizada no Experimento 2 não é linearmente separável e o perceptron não é capaz de obter um desempenho perfeito. Tal imperfeição pode ser observada na Figura 4, a qual ilustra os hiperplanos de decisões obtidos em cada época de treinamento, destacando o último em preto. O modelo obtido reportou uma acurácia de 86%.

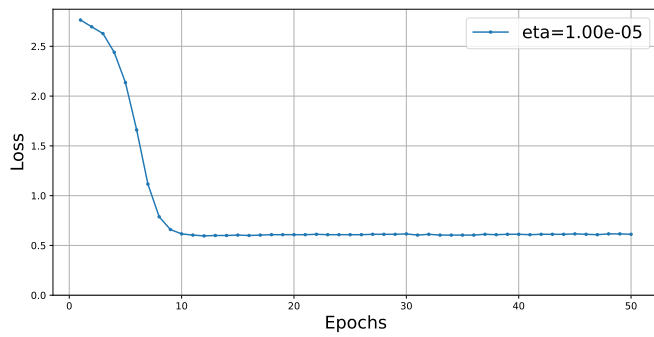


Figura 3: Curva de aprendizado para do modelo utilizado no Experimento 2.

IV. DISCUSSÕES

A implementação da rede neural do tipo perceptron realizada permitiu um maior entendimento a respeito do funcionamento da rede como um todo. Os experimentos realizados para a validação também foram de grande valia, uma vez que as observações apresentadas em Haykin [1] foram confirmadas.

Finalmente, os três objetivos estabelecidos a partir dos experimentos foram alcançados com sucesso:

- 1) Uma menor taxa de aprendizado implica uma convergência mais lenta, dado que a atualização dos pesos é realizada em passos menores. A convergência também é influenciada pelos pesos sinápticos iniciais.
- 2) Independente da taxa de aprendizado, o perceptron irá convergir para problemas linearmente separáveis.
- 3) Caso o problema não seja linearmente separável, o perceptron convergirá para um ponto ótimo, mas não será capaz de obter um resultado perfeito.

Além das observações extraídas dos objetivos evidenciados, também é importante notar que a precisão do modelo nos dados de teste pode ser menor do que a precisão do treinamento. Isso geralmente indica que há diferenças entre o tipo de dados utilizados na etapa de treinamento e os dados de teste.

REFERÊNCIAS

- [1] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2009.

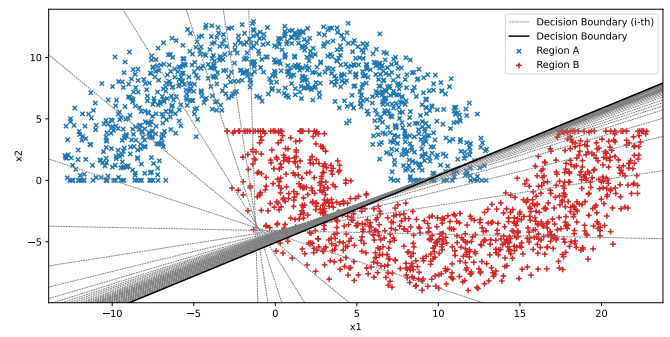


Figura 4: Hiperplano de decisão definido pelo modelo obtido no Experimento 2. O hiperplano final é dado pela reta em cor preta, enquanto as retas em cor cinza são de épocas passadas. O conjunto de dados para teste foi utilizado na imagem.