

Avaliação de Classificadores utilizando Técnicas de Estimativa de Densidades

Victor São Paulo Ruela
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
Email: victorspruela@ufmg.br

Resumo—A modelagem de dados não-lineares com redes neurais artificiais depende da qualidade projeção aplicada sobre as entradas, geralmente feita através de funções de kernel. A otimização de seus parâmetros é uma etapa importante e pode ser feita via técnicas de estimativa de densidade. Além de fornecer uma forma automática de seleção dos parâmetros ótimos, estas técnicas possuem a tendência de gerar projeções ortogonais das entradas no espaço de projetado. A partir desta observação, este trabalho tem como objetivo avaliar o desempenho de classificadores lineares sobre esta projeção, considerando problemas de benchmark presentes na literatura.

I. INTRODUÇÃO

O desempenho de redes neurais artificiais sobre dados não-lineares é altamente dependente da projeção aplicada sobre as entradas, geralmente feita através de kernels. Estas funções possuem diversos parâmetros a serem ajustados, os quais irão afetar diretamente a qualidade do modelo obtido. O ajuste de seus parâmetros é geralmente realizado via técnicas de busca exaustiva, como validação cruzada [1]. Embora amplamente utilizadas, tais técnicas não utilizam informações presentes nos dados. Isso motivou o desenvolvimento de, por exemplo, técnicas baseadas em estimativa de densidade para analisar a estrutura dos dados e reduzir a necessidade de interação do usuário [2].

Baseado no KDE (*Kernel Density Estimation*) [3], esta abordagem explora o comportamento da projeção das funções de similaridade calculadas sobre kernel escolhido. É determinada uma função sobre os parâmetros do kernel, a qual pode ser utilizada para determinar os parâmetros que melhor se ajustam aos dados. É possível, por exemplo, usar este conceito para a determinação da largura ótima de kernels radiais (RBF) para o algoritmo SVM [2]. Um resultado importante desta técnica está no fato dela poder gerar projeções ortogonais no espaço de verossimilhanças. Isso sugere a possibilidade de utilizar modelos lineares sobre esta projeção, como o Perceptron e aprendizado Hebbiano.

Este trabalho tem como objetivo avaliar o desempenho dos modelos Perceptron e Hebbiano [11] sobre as projeções no espaço de verossimilhanças. Serão considerados tanto os kernels gaussiano e perceptron de múltiplas camadas (MLP), disponibilizados pelo professor. Além destes modelos lineares, serão considerados também o ELM [4] com regularização, SVM [5], [2] e RBF com otimização de largura. Este último modelo não possui publicação associada e é proposto como

forma de extensão ao enunciado original do trabalho. Um experimento será desenhado para compará-los estatisticamente sobre diferentes bases de dados de benchmark disponíveis na literatura.

II. REVISÃO DA LITERATURA

A. Métodos baseados em Kernel

III. METODOLOGIA

A. Bases de Dados

Neste trabalho serão consideradas três bases de dados referentes a problemas de classificação binários e regressão multivariada disponíveis no repositório da UCI [6], totalizando seis problemas. Antes do treinamento, os dados de entrada serão normalizados para o intervalo $[0, 1]$ e filtrados para remoção de valores inválidos. Um sumário das bases de dados consideradas pode ser vista na Tabela I.

TABELA I
PRINCIPAIS CARACTERÍSTICAS DAS BASES DE DADOS UTILIZADAS

| | Instâncias | Atributos | Proporção |
|--------------------|------------|-----------|-----------|
| Breast Cancer | 569 | 32 | 0.66 |
| Liver Disorder | 245 | 6 | 0.58 |
| Statlog (Heart) | 270 | 13 | 0.66 |
| Boston Housing | 506 | 13 | N/A |
| Wine Quality (Red) | 1599 | 11 | N/A |
| Diabetes | 442 | 10 | N/A |

B. Desenho do experimento

A partir das recomendações para desenho de experimento para comparação de algoritmos proposta em [7], a seguinte metodologia será adotada:

- Para cada base de dados:
 - 1) Particionar os dados D em k partições para validação cruzada, mantendo a mesma proporção entre os rótulos
 - a) Criar o conjunto de treino $T = D - k$
 - b) Para cada modelo:
 - i) Executar busca exaustiva com validação cruzada de z sobre T para os coeficientes de regularização λ
 - ii) Escolher λ que obtém o melhor ajuste médio
 - iii) Avaliar a métrica do modelo sobre k

- 2) Estimar o intervalo de confiança de 95% do valor médio da métrica sobre k usando *bootstrapping*

O número de partições consideradas será de $k = 10$. Para o item 1(b)i, serão considerados um número fixo de valores igualmente espaçados dentro de um intervalo pré-definido. Além disso, serão considerados $z = 5$ partições, como forma de controlar um pouco o tamanho do experimento. Serão consideradas as métricas AUC para classificação e erro quadrático médio (MSE) para regressão, respectivamente. A implementação deste experimento, bem como dos modelos utilizados, será feita em Python e utilizando principalmente os pacotes *numpy* [8] e *scikit-learn* [9].

Os algoritmos ELM e RBF foram implementados seguindo as notas de aula do professor, sendo que o RBF irá considerar o k -means para o cálculo dos centros e raios. O Perceptron utilizado está disponível na biblioteca *scikit-learn* diretamente, suportando o uso de regularização. Já o Adaline foi implementado usando o algoritmo MLP disponível no *scikit-learn*, considerando um único neurônio na camada escondida e aprendizado via o algoritmo do gradiente estocástico. Isso foi necessário uma vez que esta implementação suporta o uso de regularização. O experimento será realizado em um Notebook Intel Core i7 Quad Core com 8Gb de memória RAM, sendo que o uso de paralelização será utilizado sempre que possível visto a enorme quantidade de vezes que os modelos serão treinados.

C. Aprendizado Hebbiano

Uma variação presente na literatura para controlar a generalização do ELM consiste no uso do aprendizado Hebbiano após a camada escondida [10]. Conforme sugerido pelo autor, podemos substituir o cálculo da pseudo-inversa da matriz de projeção aleatória por um Perceptron Hebbiano com pesos normalizados. Será considerada o algoritmo de aprendizado Hebbiano considerando somente um neurônio, similar ao Perceptron simples. Essa abordagem é melhor descrita em [11], da qual podemos retirar a seguinte regra para problemas de classificação binários:

$$w = \frac{\sum_{i=1}^N y_i \mathbf{h}_i}{\left\| \sum_{i=1}^N y_i \mathbf{h}_i \right\|} \quad (1)$$

onde \mathbf{h}_i é a i -ésima linha da matriz de projeção aleatória do algoritmo ELM original. É importante ressaltar que esta regra assume que os dados foram normalizados para possuir média zero e desvio padrão unitário. Para validar a implementação, o algoritmo foi executado sobre duas bases de dados simples e 100 neurônios na camada escondida, cujos resultados podem ser vistos na Figura ??.

Através destes resultados, podemos concluir que o uso do aprendizado Hebbiano é capaz de atingir a regularização desejada. Entretanto, nota-se que a superfície de separação torna-se predominantemente linear, de forma que seja esperado que esta abordagem possua desempenho limitado para problemas que não sejam linearmente separáveis.

D. RBF com Otimização de Escala

IV. RESULTADOS

A. Problemas de Classificação

Para os problemas de classificação foram considerados os algoritmos Perceptron, RBF, ELM e ELM com aprendizado Hebbiano. Foi estabelecido um número fixo de 20 neurônios na camada escondida e não foi aplicada regularização ao ELM Hebbiano. 50 valores para o coeficiente de regularização foram escolhidos no intervalo $[0, 1]$. Os gráficos boxplot para cada conjunto de dados considerado é exibido nas Figuras ??, ?? e ??. Os intervalos de confiança calculados são exibidos na Tabela II.

A partir destes resultados, podemos chegar às seguintes conclusões:

- Para um intervalo de confiança de 95%, podemos afirmar que todos os algoritmos possuem desempenho médio superior ao modelo RBF nas bases de dados avaliadas.
- Para a base de dados *Breast Cancer*, os modelos ELM e Perceptron obtiveram desempenhos bastante similares. Além disso, podemos afirmar que eles obtiveram desempenho médio superior ao ELM Hebbiano para o intervalo de confiança de 95%.
- Para a base de dados *Statlog (Heart)* não podemos rejeitar a hipótese de que os modelos ELM, ELM Hebbiano e Perceptron possuam desempenhos iguais para o intervalo de confiança de 95%.
- Para a base de dados *Liver Disorder*, podemos afirmar que para um intervalo de confiança de 95%, o ELM possui desempenho superior aos demais modelos.
- Uma hipótese para o desempenho muito abaixo do esperado do RBF pode estar no número de neurônios escolhidos para a camada escondida. Este hiperparâmetro não foi ajustado para que a comparação com o ELM seja justa e também para limitar o tamanho do experimento a ser executado.
- A diferença de desempenho entre o ELM e sua versão com aprendizado Hebbiano está no fato deste última poder estar resultando em uma regularização excessiva. Isso sugere que uma implementação diferente do modelo Hebbiano é indicado para controlar melhor a sua generalização.
- É interessante notar que para problemas lineares não houve uma perda muito grande na AUC média dos modelos ELM e ELM Hebbiano. Entretanto, nota-se uma maior variabilidade para este último modelo, o que sugere uma maior sensibilidade à base de dados em estudo.

B. Problemas de Regressão

Para os problemas de classificação foram considerados os algoritmos Adaline, RBF e ELM. Não foi possível realizar uma implementação do ELM Hebbiano que funcionasse com problemas de regressão, logo este modelo teve que ser descartado. Foi estabelecido um número fixo de 20 neurônios na camada escondida e a regularização foi considerada somente para todos os algoritmos. 50 valores de coeficiente de

TABELA II
INTERVALOS DE CONFIANÇA DE 95% CALCULADOS PARA A AUC MÉDIA

| | ELM | ELM Hebbiano | Perceptron | RBF |
|-----------------|----------------------------|---------------------|----------------------------|---------------------|
| Breast Cancer | 0.948 (0.935,0.963) | 0.887 (0.853,0.920) | 0.950 (0.930,0.971) | 0.512 (0.505,0.519) |
| Liver Disorder | 0.677 (0.621,0.733) | 0.582 (0.536,0.628) | 0.532 (0.509,0.553) | 0.515 (0.498,0.537) |
| Statlog (Heart) | 0.812 (0.769,0.851) | 0.762 (0.698,0.819) | 0.772 (0.700,0.846) | 0.547 (0.522,0.569) |

regularização foram escolhidos no intervalo $[0, 1]$. Nenhum outro ajuste de hiper-parâmetro foi feito para o Adaline, sendo usados os valores padrão da biblioteca *scikit-learn*. Os gráficos boxplot para cada conjunto de dados considerado é exibido nas Figuras ??, ?? e ?. Os intervalos de confiança calculados são exibidos na Tabela II.

A partir destes resultados, podemos chegar às seguintes conclusões:

- Para um intervalo de confiança de 95%, podemos afirmar que o ELM teve desempenho médio superior aos demais algoritmos para as bases de dados *Diabeters* e *Wine Quality (Red)*. Embora não podemos afirmar estatisticamente que o ELM seja superior, é importante notar que ele apresentou uma variabilidade muito menor.
- O Problema *Boston Housing* é conhecido pela sua característica linear, o que explica os bons resultados obtidos para o Adaline.
- Ao contrário do que ocorreu para os problemas de classificação, o RBF obteve bons resultados quando utilizado para regressão.
- O Adaline possuiu um desempenho bem abaixo do esperado, apresentando muita variação e uma MSE médio superior se o problema apresentar comportamento predominantemente não-linear. Isso está de acordo com a característica deste modelo.

V. CONCLUSÕES

Neste trabalho foi feita uma avaliação do desempenho dos modelos ELM, RBF, Adaline, Perceptron e ELM Hebbiano sobre bases de dados de benchmark presentes na literatura. A partir de um experimento desenhado para tal finalidade, os resultados foram comparados estatisticamente utilizando a técnica de *bootstrapping* para a média das métricas AUC e erro quadrático médio, considerando um intervalo de confiança de 95%. Destaca-se o excelente desempenho do ELM em todos os conjuntos de dados, tanto de regressão e classificação. É importante destacar também o desempenho muito abaixo do esperado para o RBF nos problemas de classificação, sugerindo a necessidade de um melhor ajuste do número de neurônios na camada escondida, por exemplo. Notou-se também que os modelos lineares avaliados obtiveram bons resultados dependendo da base de dados em questão, conseguindo atingir um desempenho estatisticamente equivalente aos demais.

Uma surpresa foi o ELM Hebbiano, onde embora tenha sido usada uma abordagem bem simples de aprendizado Hebbiano, não apresentou um resultado muito inferior em relação ao ELM original para classificação. Destaca-se o fato de que

não foi possível utilizar a estratégia de aprendizado Hebbiano com sucesso em problemas de regressão. Isso sugere que uma abordagem diferente deve ser utilizada nesta situação. Durante a execução do experimento, pôde ser observado também que o modelo RBF possui um tempo de treinamento muito maior que os demais. Isso fez com que ele fosse o gargalo de todo o experimento neste quesito, o que limitou um pouco o potencial de executar um número maior de partições para a validação cruzada, bem como avaliar um intervalo maior para o fator de regularização.

Como trabalhos futuros, é sugerido um ajuste fino de demais hiper-parâmetros do RBF como forma de tentar melhorar o seu desempenho. Além disso, também é interessante avaliar outras formas de aprendizado Hebbiano que poderiam ser utilizados com o ELM, bem como para melhorar seu poder de generalização e também ser aplicável a problemas de regressão com sucesso. Outra tarefa importante seria a realização de uma etapa de pré-processamento mais completa sobre bases de dados considerados, o qual não foi feita por restrições de tempo mas teria potencial para melhorar os resultados obtidos de todos os modelos.

REFERÊNCIAS

- [1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [2] Murilo VF Menezes, Luiz CB Torres, and Antonio P Braga. Width optimization of rbf kernels for binary classification of support vector machines: A density estimation-based approach. *Pattern Recognition Letters*, 128:1–7, 2019.
- [3] Shitong Wang, Zhaohong Deng, Fu-lai Chung, and Wenjun Hu. From gaussian kernel density estimation to kernel methods. *International Journal of Machine Learning and Cybernetics*, 4(2):119–137, 2013.
- [4] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 985–990. IEEE, 2004.
- [5] M Menezes, L Torres, and A Braga. Otimização da largura de kernels rbf para máquinas de vetores de suporte: Uma abordagem baseada em estimativa de densidades. In *XIII Congresso Brasileiro de Inteligência Computacional*, 2017.
- [6] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [7] Steven L Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery*, 1(3):317–328, 1997.
- [8] Charles R. Harris, K. Jarrod Millman, St’efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern’andez del R’io, Mark Wiebe, Pearu Peterson, Pierre G’erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

TABELA III
INTERVALOS DE CONFIANÇA DE 95% CALCULADOS PARA O MSE MÉDIO

| | Adaline | ELM | RBF |
|---------------------------|---------------------|----------------------------|---------------------|
| Boston Housing | 0.069 (0.039,0.096) | 0.012 (0.008,0.016) | 0.044 (0.025,0.061) |
| Diabetes | 0.078 (0.066,0.091) | 0.029 (0.025,0.033) | 0.054 (0.048,0.061) |
| Wine Quality (Red) | 0.037 (0.031,0.043) | 0.017 (0.016,0.019) | 0.024 (0.022,0.026) |

- [10] Euler Guimarães Horta. Aplicação de máquinas de aprendizado extremo ao problema de aprendizado ativo. 2015.
- [11] Manuel Fernandez-Delgado, Jorge Ribeiro, Eva Cernadas, and Senén Barro Ameneiro. Direct parallel perceptrons (dpps): fast analytical calculation of the parallel perceptrons weights with margin control for classification tasks. *IEEE transactions on neural networks*, 22(11):1837–1848, 2011.