Machine Learning : Um comparativo entre métodos de regularização em ELM's.

Lucas Kou Kinoshita
Departamento de Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
kometa1812001@gmail.com

***Abstract*—As *Extreme Learning Machines* (ELMs), propostas como uma solução eficiente para o treinamento de redes neurais de camada única de avanço (SLFNs), determinam aleatoriamente os pesos da camada oculta e calculam analiticamente os pesos de saída, resultando em um aprendizado rápido. No entanto, para garantir um mapeamento de características suficientemente rico, as ELMs podem empregar um número elevado de neurônios, o que frequentemente leva ao sobreajuste (*overfitting*). Este trabalho realiza uma análise comparativa de diferentes métodos de regularização para mitigar o *overfitting* em ELMs, focando na regularização L2 e em técnicas de poda de neurônios, como Poda Aleatória (*Random Pruning*), *Optimal Brain Damage* (OBD) adaptado para ELM's, e GAP-ELM (*Pruned* ELM via Algoritmos Genéticos) com uma função de *fitness* simplificada. Os métodos foram avaliados em cinco conjuntos de dados distintos, abrangendo problemas de classificação binária (*Statlog (Heart)*, *Breast Cancer*, Iris) e de aproximação polinomial (função quadrática e senoidal), utilizando validação cruzada de 10 *folds* para robustez estatística. Os resultados indicam que, em geral, os métodos de regularização conseguem reduzir a complexidade do modelo (número de neurônios) mantendo um desempenho de generalização comparável ou com perdas aceitáveis em relação à ELM padrão utilizada. O estudo ressalta a importância da regularização e da escolha adequada de hiperparâmetros, validada por metodologias como a validação cruzada, para otimizar o desempenho e a complexidade das ELM's.**

***Index Terms*—Extreme machine learning, ELM's, regularization, genetic algorithm, L2, GAP-ELM, *randomized pruning*, *optimal brain damage*, OBD**

I. Introdução

Propostas por Huang em [3] como solução para a lentidão resultante do processo de aprendizado por gradiente descendente e do processo de ajuste iterativo dos parâmetros utilizado em redes baseadas em *feedfowarding* anteriorees , as *extreme learning machines* (ELM's) são redes neurais de duas camadas nas quais as magnitudes dos pesos da camada oculta são determinadas de forma aleatória. Uma vez que gerar muitos neurônios aleatórios é computacionalmente barato em comparação com o ajuste iterativo de todos os parâmetros em redes tradicionais.

Como esses parâmetros não são otimizados, nem todos os neurônios gerados aleatoriamente serão "ideais" ou igualmente úteis para representar os dados de forma eficaz. Desta forma, para garantir que a camada oculta forneça um mapeamento suficientemente rico e diversificado das características de entrada, muitas vezes as ELM's valem-se de um número elevado de neurônios. O que pode comumente levar ao *overfitting*.

Devido à necessidade de combater o *overfitting*, método de regularização como o L2 e o *random prunning* foram propostos ao longo da história. Nestre trabalho, busca-se realizar uma comparação do desempenho destes e de outros métodos quando aplicados a diferentes conjuntos de dados (*datasets*).

II. Revisão bilbiográfica

A. L2

De acordo com o trabalho de Jens Flemming sobre "Generalized Tikhonov regularization" [2], o método de regularização L2, frequentemente referido como o caso padrão da regularização de Tikhonov em espaços de Hilbert, aborda a solução de equações de operador mal-postas.

Como proposto originalmente por Andrei Nikolaevich Tikhonov, a ideia central da regularização de Tikhonov é estabilizar o problema de encontrar uma solução x para uma equação da forma F(x)=y, onde pequenas perturbações nos dados y podem levar a grandes erros na solução x. Isso é feito adicionando-se um termo de penalidade (ou estabilização) à função que se deseja minimizar.

Para L2, buca-se minimizar o funcional de Tikhonov:

$$T_\alpha^{y^\delta}(x) = \frac{1}{2}\left\|Ax - y^\delta\right\|^2 + \frac{\alpha}{2}\left\|x\right\|^2 \tag{1}$$

Onde A é um operador linear limitado que mapeia do espaço da solução X para o espaço dos dados Y (ambos espaços de Hilbert), y^δ representa os dados medidos, que podem conter ruído, sendo uma aproximação do lado direito exato y, $\frac{1}{2}\left\|Ax - y^\delta\right\|^2$ é o termo de fidelidade aos dados, que mede o quão bem a solução x explica os dados observados e $\frac{\alpha}{2}\left\|x\right\|^2$ é o termo de regularização L2, responsável por penalizar soluções com norma elevada (magnitude dos componentes de x), favorecendo soluções "menores" ou "mais suaves".

B.

Randomized pruning O artigo [5] propõe que redes neurais grandes, mesmo com pesos atribuídos aleatoriamente, podem conter sub-redes menores que alcançam um desempenho notável sem que seus pesos sejam alterados. Portanto, se a distribuição dos pesos for adequadamente escalonada, a rede

conterá uma sub-rede que funciona bem sem nunca modificar os valores dos pesos. De fato, os autores demonstram que essas "sub-redes não treinadas" são capazes de atingir resultados satisfatórios.

As ELM's por sua vez, por natureza possuem pesos fixos atribuídos aleatoriamente na camada oculta. Ou seja, ao ser inicializada com um número elevado de neurônios, uma ELM pode ser considerada uma rede superparametrizadas que "esconde" sub-redes eficazes.

Ao reduzir aleatoriamente o número de neurônios, a poda aleatória diminui a complexidade do modelo. Se a ELM original for excessivamente grande, essa redução pode ser capaz de mitigar o sobreajuste *overfitting*. Assim, a poda aleatória, ao remover neurônios possivelmente ruidosos, redundantes ou que simplesmente não contribuem positivamente para a tarefa, pode resultar em um modelo mais enxuto que, por manter uma porção dos neurônios aleatórios originais (alguns dos quais podem constituir uma sub-rede eficaz), pode exibir uma capacidade de generalização aprimorada em relação ao modelo não podado.

C.

Optimal brain damage O Optimal Brain Damage (OBD) é um método para reduzir o tamanho de uma rede neural através da remoção de pesos considerados menos importantes. A ideia fundamental do OBD é utilizar informações da segunda derivada da função de erro (objetivo) para realizar um compromisso (trade-off) entre a complexidade da rede e o erro obtido no conjunto de treinamento. O método parte do princípio de que é possível pegar uma rede já treinada e funcional, remover uma porção significativa de seus pesos (até metade ou mais) e obter uma rede que funcione tão bem quanto.

O OBD define a saliência de um parâmetro (peso) como a mudança na função objetivo (erro) que seria causada pela remoção daquele parâmetro. Com este objetivo, Le Cun, Denker e Solla [4] propõem um modelo local da função de erro, aproximando-a por uma série de Taylor para prever analiticamente o efeito da perturbação (remoção) de um peso. Onde a mudança δE na função objetivo E devido a uma perturbação δu_i no parâmetro u_i é dada por:

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta u_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(\|\delta U\|^3) \tag{2}$$

onde $g_i = \frac{\partial E}{\partial u_i}$ são os componentes do gradiente e $h_{ij} = \frac{\partial^2 E}{\partial u_i \partial u_j}$ são os elementos da matriz Hessiana.

D. Genetic algorithms for pruned ELM

Proposto por Alencar, Rocha Neto e Gomes em [1], o GAP-ELM utiliza Algoritmos Genéticos (AGs) para realizar a poda de neurônios da camada oculta. A abordagem seleciona um subconjunto dos neurônios ocultos gerados aleatoriamente pela ELM, com o objetivo de otimizar uma função de adaptabilidade (*fitness function*) multiobjetivo que estabelece um compromisso entre a acurácia do classificador e o número de neurônios podados (ou a taxa de neurônios mantidos).

No GAP-ELM, cada indivíduo da população do algoritmo genético é representado por um vetor binário, onde cada gene indica se um neurônio candidato da camada oculta será incluído (valor 1) ou descartado (valor 0) do modelo final. A função de adaptabilidade, conforme detalhado pelos autores de [1], é uma combinação ponderada da taxa de erro e da taxa de neurônios, expressa como:

$$f(g) = \alpha e_{rate}(g) + (1 - \alpha) n_{rate}(g) \tag{3}$$

Onde $e_{rate}(g)$ é a taxa de erro de classificação e $n_{rate}(g)$ é a proporção de neurônios ativos em relação ao total inicial. Para calcular a taxa de erro de forma eficiente durante a avaliação da *fitness*, o método emprega a estatística PRESS (*Prediction Sum of Squares*), que permite uma estimativa do erro de validação cruzada *leave-one-out* (LOO) sem o alto custo computacional dos procedimentos padrão.

III. Metodologia

Nesta seção, serão abordados os processos em metodologias adotadas durante a implementação e comparação dos diferentes métodos de regularização. Com o intuito de compreender melhor cada uma das etapas aplicadas durante a realização do trabalho, serão descritos os tratamentos realizados em cada conjunto de dados testado, a implementação de cada um dos algoritmos experimentados e os parâmetros utilizados.

A. Conjuntos de dados

Os conjuntos foram escolhidos de forma arbitrária, sendo os três primeiros (*Statlog(Heart), Breast Cancer, Iris*) problemas de classificação binária e o dois últimos (função quadrática e senoidal) problemas de aproximação polinomial.

1) Statlog (Heart): Obtida da database referenciada em VI-B. As classes, que inicialmente são separadas por valores 1 e 2 foram mapeadas para -1 e 1 para serem identificadas com sucesso pelos métodos implementados.

2) Breast Cancer: Obtida do pacote *"mlbench"*. Neste dataset, as classes "maligna" e "benigna" foram mapeadas para valores numéricos 1 e -1, a coluna "ID" foi removida e as linhas que possuíam valores NA foram omitidas e todos os valores foram assertidos para o formato numérico.

3) Iris: Assim como o conjunto de dados *Breast Cancer*, também obtida do pacote *"mlbench"*. O *dataset* Iris possuía 3 classes distintas originalmente, sendo elas "setosa", "versicolor" e "virginica", como este trabalho busca avaliar apenas problemas de classificação binários, as três classes foram convertidas em "setosa" e "não-setosa".

Por fim, as *features* selecionados para o treinamento dos modelos foram *Sepal.Length, Sepal.Width, Petal.Length* e *Petal.Width*

4) Função quadrática: O primeiro problema de aproximação polinomial elaborado é dado por:

$$y = x^2 - 2x + 1 + \beta \tag{4}$$

Onde y é o polinômio a ser aproximado e β é a função que determina o ruído das amostras (com desvio padrão de 0.5).

5) Função senoidal: O segundo problema de aproximação polinomial elaborado trata de uma função senoidal dada por:

$$y = sin(x) + X_i \tag{5}$$

$$X_i \sim \mathcal{N}\left(0, \left(\frac{\sigma_{\text{noise}}}{2}\right)^2\right) \tag{6}$$

Onde X_i é simplesmente a função que determina o ruído da $i$ésima amostra.

B. Métodos de regularização

Todos os método mencionados anteriormente em II foram implementados mas é necessário fazer algumas ressalvas.

Primeiramente, como o método OBD foi inicialmente proposto para redes neurais treinadas a partir de *back propagation*, aplicá-lo a uma ELM requer uma adaptação já que os pesos dos neurônios da camada oculta são fixos e aleatoriamente determinados. Assumiu-se que a importância de um neurônio oculto pode ser inferida a partir da saliência do(s) seu(s) peso(s) de saída correspondente(s) em w. Esta saliência foi implementada à partir do cálculo do erro médio quadrático de $H\omega$ em relação à saída Y da rede e da hessiana de H (onde H é a matriz da camada oculta da rede). Os neurônios foram então ordenados de acordo com suas respectivas saliências para que se obtenha uma nova matriz de pesos Z com neurônios de baixa saliência removidos (podados), a partir da qual a rede foi retreinada.

Com relação ao método GAP-ELM, o método de cálculo da função de adaptabilidade (*fitness function*) proposta no trabalho de Alencar [1] foi simplificado, no lugar de usar o erro do LOO calculado via PRESS, utilizou-se apenas a taxa de erro do treino.

O método de poda aleatória implementado realiza a escolha dos neurônios a serem excluídos simplesmente por meio do méotodo *sample()* pertencente à linguagem R.

Finalmente, vale notar que o termo de penalização de L2 mencionado em II foi implementado como λ, seguindo o problema de minimização do erro J, análogo ao funcional de Tiknov (equação 1):

$$J = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \sum_{j=1}^{p}\lambda\omega_j^2 \tag{7}$$

C. Experimentos

Cada um dos experimentos realizados foi feito por meio do método de *10 fold Cross-Validation* para garantir significância estatística nos resultados obtidos e também como forma de validar os hiperparâmetros escolhidos para cada um dos métodos.

É importante notar que, além dos resultados obtidos neste trabalho, a variação dos hiperparâmetros de cada método (por exemplo: quantidade inicial de neurônios, população, taxa de mutação, número máximo de iterações e taxa de elitização) pode resultar em melhor ou pior desempenho do que os que foram aqui observados.

Finalmente, os parâmetros utilizados para a realização de cada um dos experimentos são listados abaixo:

- $p_{inicial} = 150$ para todos os métodos
- Termo de *bias* $+1$ para todos os métodos
- *Pruning rate* de 0.2 para a random prunned (RP) ELM
- $\lambda = 0.1$ para o método L2
- *Pruning rate* de 0.1 para a OBD com máximo de 10 iterações e mínimo de 5 neurônios
- 20 indivíduos (população), máximo de 30 iterações, valor alvo de adaptabilidade de 0.99, taxa de mutação de 0.1, taxa de *crossover* de 0.8 e taxa de elitismo de 0.1 para o método baseado em algoritmos genéticos.

IV. Resultados e discussão

Nesta seção serão abordados os resultados obtidos pelos experimentos.

TABLE I
STATLOG (HEART)

Modelo	**Acc Treino**	**Acc Teste**	p **final**
Standard	0.7625 ± 0.0295	0.6592 ± 0.0757	150 ± 0.0
RP	0.7354 ± 0.0424	0.6407 ± 0.0856	120.0 ± 0.0
L2	0.7465 ± 0.0312	0.6630 ± 0.0564	150 ± 0.0
OBD	0.5634 ± 0.0165	0.5592 ± 0.1228	50.0 ± 0.0
GAP	0.6243 ± 0.0307	0.6333 ± 0.0947	72.4 ± 4.1150

TABLE II
BREAST CANCER

Modelo	**Acc Treino**	**Acc Teste**	p **final**
Standard	0.9871 ± 0.0022	0.9502 ± 0.0250	150 ± 0.0
RP	0.9832 ± 0.0038	0.9458 ± 0.0293	120.0 ± 0.0
L2	0.9855 ± 0.0026	0.9459 ± 0.0257	150 ± 0.0
OBD	0.0		
GAP	0.9556 ± 0.0040	0.9414 ± 0.0207	66.8 ± 4.3410

TABLE III
IRIS

Modelo	**Acc Treino**	**Acc Teste**	p **final**
Standard	1.0 ± 0.0	0.9933 ± 0.0211	150 ± 0.0
RP	1.0 ± 0.0	0.9933 ± 0.0211	120.0 ± 0.0
L2	1.0 ± 0.0	1.0 ± 0.0	150 ± 0.0
OBD	1.0 ± 0.0	1.0 ± 0.0	50 ± 0.0
GAP	1.0 ± 0.0	0.9933 ± 0.0211	103.6 ± 4.7889

TABLE IV
POLINÔMIO QUADRÁTICO

Modelo	**MSE Treino**	**MSE Teste**	p **final**
Standard	12.4767 ± 0.1730	12.3998 ± 1.4684	150 ± 0.0
RP	12.4767 ± 0.1730	12.3998 ± 1.4684	120.0 ± 0.0
L2	12.4771 ± 0.1730	12.4003 ± 1.4680	150 ± 0.0
OBD	12.4767 ± 0.1732	12.3995 ± 1.4681	50 ± 0.0
GAP	12.4767 ± 0.1730	12.3998 ± 1.4684	103.6 ± 4.7889

As expectativas iniciais quanto aos métodos de regularização pressupõem uma diminuição do *overfitting* através da minimização da função de erro por meio da penalização da magnitude dos pesos ou da poda de neurônios (seja ela otimizada por algoritmos genéticos, pelo OBD ou aleatória). Desta forma, com relação aos resultados dos

TABLE V
POLINÔMIO SENOIDAL

Modelo	**MSE Treino**	**MSE Teste**	p **final**
Standard	0.5161 $\pm$ 0.0083	0.5328 $\pm$ 0.0576	150 $\pm$ 0.0
RP	0.5161 $\pm$ 0.0083	0.5363 $\pm$ 0.0609	120.0 $\pm$ 0.0
L2	0.5214 $\pm$ 0.0094	0.5166 $\pm$ 0.0722	150 $\pm$ 0.0
OBD	0.5218 $\pm$ 0.0090	0.5164 $\pm$ 0.0723	50.0 $\pm$ 0.0
GAP	0.5168 $\pm$ 0.0083	0.5328 $\pm$ 0.0576	103.6 $\pm$ 4.7889

experimentos realizados, espera-se que haja uma diminuição da acurácia em conjuntos de treinamento sem efeitos negativos significantes nos conjuntos de teste. As tabelas I, II, III, IV e V retratam os dados resultados dos experimentos realizados.

De fato, os valores de acurácia para o método L2 nas tabelas I e II mostram um leve decaimento nos conjuntos de treino e manutenção da acurácia nos conjuntos de teste, o mesmo ocorre com os valores de MSE para este método na tabela V (leve aumento do MSE nos conjuntos de treino e diminuição deste nos conjuntos de teste).

Quanto aos métodos de poda, é notável que em todas as tabelas, com exceção da III, mesmo com a diminuição do número de neurônios p, os modelos foram capazes de encontrar boas soluções, ou seja, resultados de acurácia e MSE que apresentam perdas aceitáveis ou manutenção da qualidade quando comparados à rede não regularizada (nomeada como "Standard"). Como todas as redes foram treinadas com um número inicial de 150 neurônios, a proposta de [5], comentada na seção de revisões bibliográficas II, mostra-se notável, visto que a diminuição significativa do número de neurônios das redes resultarou em sub-redes eficazes provenientes do modelo inicial, fator evidenciado pelos valores de acurácia de teste e p final do ODB e da GAP-ELM retratados na tabela II por exemplo.

Em relação às tabelas III e IV, nota-se uma variação quase nula nos valores de acurácia e MSE apresentados ao decorrer dos experimentos realizados. Embora isto nos garanta que o processo de regularização funcionou bem, visto que, mesmo com a variação do número de neurônios, os modelos mantiveram um desempenho praticamente constante, é indicado que, futuramente, sejam utilizados conjuntos de dados mais "problemáticos" para a realização dos experimentos, por exemplo, pela identificação de outra espécie de planta no *dataset* Iris que não seja a "setosa" ou da regressão de polinômios mais complexos. Desta forma os experimentos produziriam resultados que permitem uma melhor visualização da variação das métricas dos resultados provenientes da utilização dos diferentes métodos de regularização.

V. CONCLUSÃO

A análise comparativa foi conduzida em cinco *datasets* distintos, abrangendo problemas de classificação e aproximação polinomial, avaliando os métodos com base em suas métricas principais, e.g., acurácia de teste, MSE, e número final de neurônios.

Finalmente, podemos concluir que as expectativas prospostas para a realização deste trabalho foram alcançadas com sucesso, uma vez que os modelos desejados foram implementados em suas totalidades e os resultados obtidos condizem com aqueles esperados em sua grande maioria. Nota-se que a implementação e parametrização dos métodos de regularização não é uma tarefa trivial e que métodos como o *10 fold CV* devem ser utilizados para garantir relevância estatística e favorecer melhores *insigths* acerca dos parâmetros utilizados em cada experimento.

O aluno ressalta ainda que foi capaz de adquirir novos conhecimentos atrelados aos métodos explorados, principalmente OBD e GAP-ELM mas também em relação a métodos não citados neste trabalho como *skeletonization*, bem como foi capaz de aprofundar conhecimentos previamente existentes acerca do processo de regularização de redes neurais.

VI. APÊNDICE

A. Repositório

https://github.com/LucasKouKinoshita/TP_intermerdi-rio_RNA

B. Statlog (Heart) - source

http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/heart.dat

REFERENCES

[1] Alisson S.C. Alencar, Ajalmar R. Rocha Neto, and João Paulo P. Gomes. A new pruning method for extreme learning machines via genetic algorithms. *Applied Soft Computing*, 44:101–107, 2016.

[2] Jens Flemming. Generalized tikhonov regularization. 2011.

[3] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501, 2006. Neural Networks.

[4] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

[5] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.