

3ª Lista de Exercícios Computacionais - Resolução

Lucas F. P. Costa
Universidade Tecnológica Federal do Paraná
Ponta Grossa, Paraná
luccos@alunos.utfpr.edu.br

I. INTRODUÇÃO

Redes de Hopfield são um tipo de rede neural com inspiração em conceitos da física e estatística, que apresentam capacidade de armazenar memória (ou estados) e são totalmente realimentadas. Nesta, consideram-se padrões de entrada para cálculo de pesos entre ligações e não se faz separação de conjuntos de treino, validação e teste [1]. Diferentemente, Máquinas de Aprendizado Extremo (ELMs, *Extreme Learning Machines*) propõem uma abordagem com arquitetura similar de uma MLP, contendo pesos não ajustáveis entre camadas, sendo que estes são calculados com um combinador linear usando, por exemplo, a pseudo-inversa de Moore-Penrose [2].

Objetiva-se com a conclusão deste trabalho solucionar duas questões referentes à disciplina de Redes Neurais [3]. A primeira consiste na avaliação do uso da rede de Hopfield para classificação de três conjuntos de padrões diferentes e na sua capacidade de restauração de memórias fundamentais frente a novos padrões ruidosos. Por sua vez, a segunda questão é sobre empregar ELMs para classificar corretamente dados das bases de dados IRIS e WINE, semelhante ao feito em [4].

A organização deste trabalho se dá por ordem das questões, contendo para cada descrição, desenvolvimento da solução e resultados obtidos com uma conclusão geral no final.

II. QUESTÃO 1

A descrição da questão é como segue.

Sobre as redes de Hopfield, a qual utiliza a regra de Hebb generalizada para treinamento, rode o toolbox MATLAB disponibilizado pelo professor para os 3 casos abaixo. O toolbox está setado para um problema com 4 entradas e um *grid* (matriz) de 5×9 elementos. Rode-o inicialmente com os dados já inseridos para entender o seu funcionamento e em seguida faça as adaptações necessárias para os demais casos.

Varie a quantidade de ruído colocando 8%, 10%, 25% e 50% e mostre uma tabela com a quantidade de acertos em cada caso. Nos casos em que a restauração da memória fundamental não é perfeita, mas é muito próxima disso, indique a porcentagem média de *pixels* distintos.

Considere 3 arquivos de dados contendo memórias fundamentais, com as seguintes especificações:

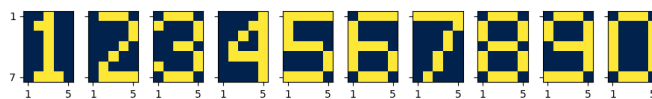
- Arquivo 1:
 - Nome: livro.dat;
 - Quantidade de memórias fundamentais: 8;
 - Dimensão da grade: 12×10 .
- Arquivo 2:
 - Nome: numeros.dat;
 - Quantidade de memórias fundamentais: 10;
 - Dimensão da grade: 7×5 .
- Arquivo 3:
 - Nome: letras.dat;
 - Quantidade de memórias fundamentais: 12;
 - Dimensão da grade: 20×20 .

A. Desenvolvimento

Neste exercício guiado, três conjuntos de padrões diferentes de entrada são providos em formatos vetoriais. Cada vetor está em uma dimensão e representa uma imagem, cujas dimensões também são informadas, pelas quais é possível representar graficamente, como é feito na Figura 1 para cada arquivo. Estes são os padrões de entrada que servirão de memórias fundamentais para a rede.



(a) Padrões do arquivo livro.dat.



(b) Padrões do arquivo numeros.dat.



(c) Padrões do arquivo letras.dat.

Figura 1: Padrões de entrada de cada arquivo disponibilizado.

A implementação, como supradescrito, foi disponibilizada pelo professor em linguagem MATLAB. Foi feita a adaptação deste para linguagem Python pelo colaborador Thiago de Camargo [6], integrante da turma, sendo a utilizada para conclusão desta questão.

Após inserir as memórias fundamentais no sistema, os padrões sofrerão um processo de ruído para os alterar, de acordo com as porcentagens requeridas. Depois, serão reinseridos para determinar o quão bem o sistema consegue recuperar as memórias fundamentais.

B. Resultados

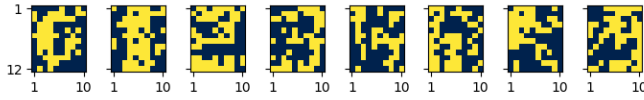
A seguir, são mostrados os resultados da metodologia proposta e requerimentos da questão para cada conjunto de padrões de entrada ou arquivo.

1) *Entrada livro.dat*: A Tabela I mostra a taxa de acerto para cada um dos oito padrões de entrada deste arquivo, para cada quantidade de ruído.

Tabela I: Taxa de acerto por padrão para valores de ruído diferentes.

Padrões	Acerto por ruído (%)			
	8%	10%	25%	50%
Padrão 1	100,00	100,00	100,00	51,67
Padrão 2	100,00	100,00	100,00	32,50
Padrão 3	100,00	97,50	97,50	46,67
Padrão 4	100,00	100,00	100,00	30,00
Padrão 5	100,00	100,00	100,00	62,50
Padrão 6	100,00	100,00	100,00	90,00
Padrão 7	100,00	100,00	77,50	50,00
Padrão 8	95,00	100,00	100,00	30,00
Média	99,37	99,69	96,87	49,17

Observa-se que conforme aumenta a quantidade de ruído, maior é a dificuldade de recuperação das memórias fundamentais. Não obstante, em nenhum dos casos foi encontrada resposta perfeita. Para ruído igual a 25%, tem-se em média 3,125% de *pixels* distintos, ilustrado pela Figura 2 que contém (a) a entrada ruidosa e (b) a saída do sistema, sendo que a entrada pode ser verificada na Figura 1(a).



(a) Entrada com ruído de 25%.



(b) Recuperação da memória fundamental.

Figura 2: Padrões com ruído e sequente resposta da rede na tentativa de recuperação.

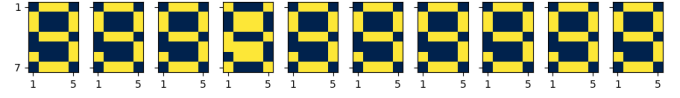
2) *Entrada numeros.dat*: Neste caso, são ao todo 10 padrões de entrada, cujos acertos são expostos na Tabela II.

Tabela II: Taxa de acerto por padrão para valores de ruído diferentes para entrada *numeros.dat*.

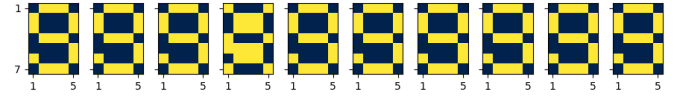
Padrões	Acerto por ruído (%)			
	8%	10%	25%	50%
Padrão 1	54,29	54,29	45,71	54,29
Padrão 2	71,43	71,43	71,43	71,43
Padrão 3	94,29	94,29	94,29	5,71
Padrão 4	54,29	54,29	45,71	45,71
Padrão 5	80,00	80,00	80,00	80,00
Padrão 6	85,71	85,71	85,71	85,71
Padrão 7	57,14	57,14	57,14	88,57
Padrão 8	97,14	97,14	97,14	97,14
Padrão 9	91,43	91,43	91,43	8,57
Padrão 10	82,86	82,86	82,86	17,14
Média	76,86	76,86	75,14	55,42

Um comportamento inusitado se deu neste caso, com res-

postas visualmente similares para todas quantidades de ruído testadas e baixo desempenho geral. A Figura 3 mostra o resultado de recuperação para todas as quatro quantidades de ruído aplicadas. No melhor caso, para ruído de 8%, a porcentagem de *pixels* diferente foi de 23,14%.



(a) Resposta para ruído de 8%.



(b) Resposta para ruído de 10%.



(c) Resposta para ruído de 25%.



(d) Resposta para ruído de 50%.

Figura 3: Respostas da rede para todas quantidades de ruído aplicadas.

Percebe-se que frequentemente a recuperação resulta em um padrão similar ao do dígito 8, por vezes com *pixels* invertidos. Quando é verificada a capacidade de armazenamento para alcançar aproximadamente 100% de acerto, $C^{100\%}$, observa-se que:

$$C^{100\%} = \frac{N}{4 \ln N} = \frac{(7 \cdot 5)}{4 \ln (7 \cdot 5)} = 2,46.$$

Ou seja, a quantidade de padrões máxima para essa dimensão é de 2, muito abaixo do total apresentado, o que explica os resultados obtidos, levando a estados espúrios.

3) *Entrada letras.dat*: Na Tabela III estão os acertos por quantidade de ruído para os 12 padrões de entrada, sendo que no melhor caso, a porcentagem de *pixels* diferentes foi de 17,35%, para ruído de 8%.

Nota-se um comportamento similar ao analisado anteriormente para as respostas da rede, o que pode ser visto na Figura 4. No entanto, neste caso, quando calculada a capacidade de armazenamento, verifica-se que $C^{100\%}$ é igual a 16,69, valor que permite a quantidade de padrões usados (12). Mesmo assim é perceptível o fato de que, de forma geral, não foi encontrado um estado de equilíbrio.

Tabela III: Taxa de acerto por padrão para valores de ruído diferentes para entrada `letras.dat`.

Padrões	Acerto por ruído (%)			
	8%	10%	25%	50%
Padrão 1	71,00	71,00	71,00	29,00
Padrão 2	83,25	83,25	83,25	83,25
Padrão 3	95,00	95,00	95,00	5,25
Padrão 4	87,50	87,50	87,50	87,50
Padrão 5	91,50	91,50	91,50	8,50
Padrão 6	79,00	79,00	79,00	79,00
Padrão 7	87,00	87,00	87,00	13,00
Padrão 8	75,00	75,00	75,00	75,00
Padrão 9	88,75	88,75	88,75	11,25
Padrão 10	84,25	84,00	84,25	16,00
Padrão 11	73,00	73,00	73,00	73,00
Padrão 12	76,50	76,50	76,50	23,50
Média	82,65	82,62	82,65	42,02

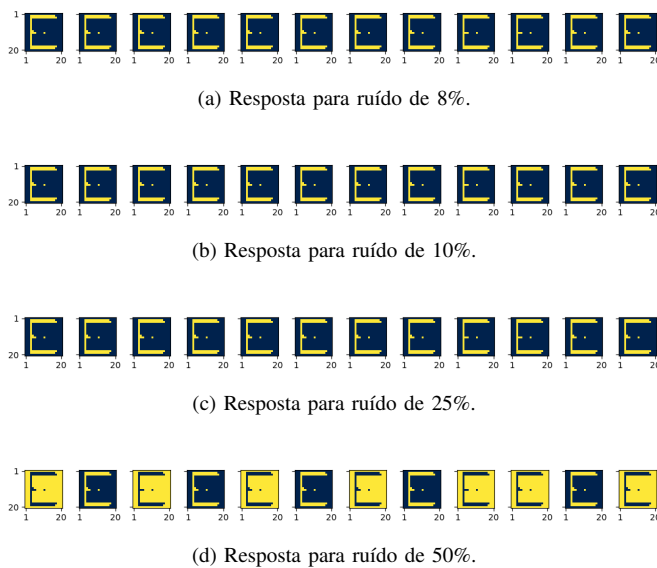


Figura 4: Respostas da rede para todas quantidades de ruído aplicadas.

III. QUESTÃO 2

A descrição da questão é feita a seguir.

Encontre no site UCI Machine Learning (<https://archive.ics.uci.edu/ml/index.php>) as bases IRIS e WINE. Implemente a rede ELM treinada com a pseudo-inversa de Moore Penrose. Faça um gráfico boxplot dos resultados de classificação em termos do MSE e apresente, para cada uma das 30 rodadas, a taxa percentual de acerto. Em cada caso, compare os resultados para números distintos de neurônios na camada intermediária.

A. Desenvolvimento

As bases de dados IRIS e WINE foram usadas e descritas anteriormente em outra atividade, conforme está em [4]. No entanto, vale lembrar que ambas contêm dados de múltiplas classes, não sendo esta uma tarefa de classificação binária.

Semelhantemente, os dados devem ser normalizados, para evitar efeitos indesejados quanto a discrepância existente entre cada informação.

Para analisar o desempenho da ELM sobre os dados, são feitos testes variando a quantidade de neurônios da camada intermediária com 30 repetições de treinamento, obtendo ao final medidas de acertos e erros que permitem concluir sobre a efetividade da solução.

B. Resultados

Os resultados da aplicação da ELM são mostrados a seguir, por base de dados usada.

1) *Base de dados IRIS*: A Figura 5 mostra o *boxplot* de MSE para as quantidades de neurônios testadas, que foram: 4, 8, 16, 32 e 64. Em virtude do grande número de rodadas, na Tabela IV são mostradas somente as taxas de acerto médias e desvio padrão.

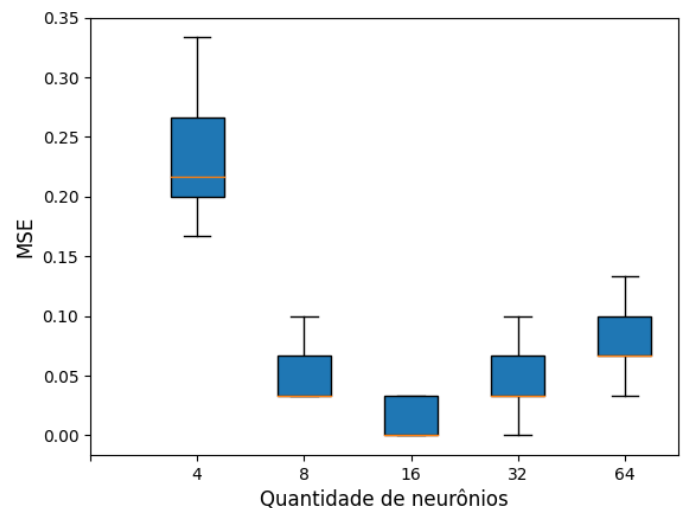


Figura 5: Gráfico *boxplot* de MSE por quantidade de neurônios testados.

Tabela IV: Taxa de acerto médio e desvio padrão por quantidade de neurônios.

Qtde. de neurônios	Acerto médio (%)	Desvio padrão
4	77,00	4,25
8	94,78	2,39
16	98,44	1,66
32	96,00	2,64
64	91,89	2,39

Nota-se pela análise do gráfico e da tabela que o pior modelo foi aquele com apenas quatro neurônios, mostrando maior variação e valor de MSE e pior taxa de acerto, enquanto que o melhor foi usando 16 neurônios, sendo que a melhor rodada obteve 100% de acerto e MSE igual a 0. Também, pela variação dos resultados, percebe-se que as quantidades de neurônios testadas foram adequadas, encontrando um ponto ótimo justamente com a mediana das opções.

2) *Base de dados WINE*: O gráfico *boxplot* de MSE para quantidade de neurônios é ilustrado na Figura 6 e a Tabela ?? mostra os acertos médios e desvio padrão.

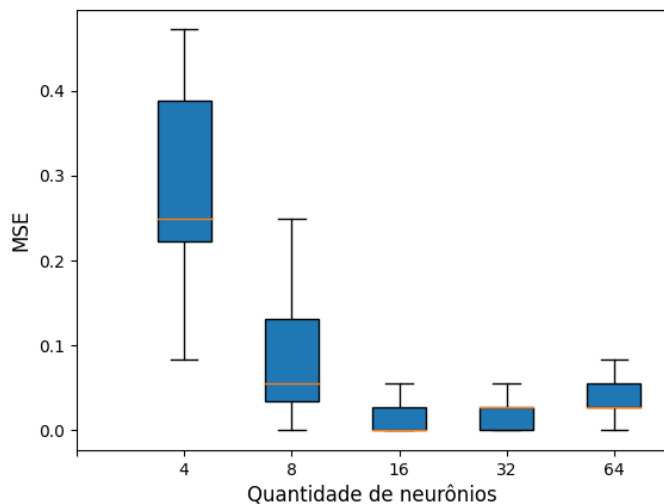


Figura 6: *Boxplot* de MSE por quantidade de neurônios na base WINE.

Tabela V: Taxa de acerto médio e desvio padrão por quantidade de neurônios.

Qtde. de neurônios	Acerto médio (%)	Desvio padrão
4	75,28	11,51
8	91,48	6,45
16	98,98	1,68
32	97,78	1,81
64	96,39	2,79

Um desempenho péssimo em relação as demais se dá quando o número de neurônios é igual a quatro, diferente do que se pode ver para os casos de 16 e 32: ambos tiveram resultado ótimo e muito similares. Fazendo o desempate pelo valor médio de acerto, se sobressai a configuração de 16 neurônios, o mesmo que para a base de dados IRIS. A mesma conclusão é aplicável sobre a escolha das quantidades de neurônios.

IV. CONCLUSÃO

Duas questões envolvendo o uso de redes de Hopfield e ELMs foram abordadas, explicadas e resolvidas. Tais tarefas permitem observar como se comportam os sistemas para problemas diferentes e sua real aplicabilidade, bem como limites que devem ser respeitados.

REFERÊNCIAS

- [1] SIQUEIRA, H. V. **Redes Neurais Artificiais**: Aula 14 - Redes de Hopfield. Ponta Grossa: Universidade Tecnológica Federal do Paraná, 2021. 19 slides, color.
- [2] HUANG, G.-B. ZHU, Q.-Y. SIEW, C.-K. Extreme Learning Machine: Theory and Applications, **Neurocomputing**, v. 70, n. 1-3, p. 489-501, 2006.
- [3] SIQUEIRA, H. V. **Lista 3**. Disponível em: <http://paginapessoal.utfpr.edu.br/hugosiqueira/disciplinas-mestrado/redes-neurais-artificiais/Lista%202020-%20Redes%20Neurais.pdf>. Acesso em: 12 mar. 2021.

- [4] COSTA, L. F. P. **2ª Lista de Exercícios Computacionais - Resolução**. Disponível em: <https://mega.nz/file/H7hSlQKB#QzcFiRDN5k4edpRgJl4obFbGtSSeUSm1GyvF88c51M>. Acesso em: 5 mai. 2021.
- [5] DUA, D. GRAFF, C. **UCI Machine Learning Repository**. Disponível em: <http://archive.ics.uci.edu/ml>. Irvine, CA: University of California, School of Information and Computer Science. Acesso em: 12 abr. 2021.
- [6] CAMARGO, T. F. O. de. **Toolbox Lista 3 (Python)**. Disponível em: http://paginapessoal.utfpr.edu.br/hugosiqueira/disciplinas-mestrado/redes-neurais-artificiais/toolbox-lista-3/Hopfield_python.py. Acesso em: 29 mai. 2021.