
Avaliação do Modelo BTHOWeN em Datasets Multiclasse

Breno Tostes Eduardo Naslausky Felipe Barrocas
Giordano Souza Maria Bianca Irace Miguel Sousa
Rafael Paladini

COPPE, Universidade Federal do Rio de Janeiro

Abstract

Este trabalho avalia o modelo BTHOWeN (Bleached Thermometer-encoded Hashed-input Optimized Weightless Neural Network) em diversos datasets multiclasse. Comparamos a performance do BTHOWeN com o modelo BloomWiSARD, analisando o impacto dos hiperparâmetros na acurácia final. Utilizamos uma abordagem gulosa para otimização, variando sequencialmente os parâmetros de tamanho de endereço, número de discriminadores, funções hash, filtros Bloom e fator de bleaching. Os resultados demonstram as vantagens do BTHOWeN em termos de acurácia e eficiência para diversos conjuntos de dados de classificação.

1 Introdução

A BTHOWeN (Bleached Thermometer-encoded Hashed-input Optimized Weightless Neural Network) é uma arquitetura de rede neural sem peso que se diferencia do modelo WiSARD (Wilkes, Stonham and Aleksander Recognition Device) [2] por:

- Incorporar **counting** Bloom filters para reduzir o tamanho do modelo e permitir bleaching [6, 5]. Diferentemente dos Bloom filters tradicionais que armazenam apenas bits (0 ou 1), os counting Bloom filters usados no BTHOWeN mantêm contadores que registram quantas vezes um padrão foi encontrado durante o treinamento, permitindo a aplicação de um limiar de bleaching (threshold) para reduzir o efeito do sobreajuste.
- Usar função hash da família H3 que não requer operações aritméticas complexas, consistindo apenas de operações XOR entre bits de entrada e máscaras aleatórias, tornando-a ideal para implementações em hardware com baixo consumo de energia [8, 7].
- Codificação com termômetro não-linear baseado na distribuição gaussiana dos dados, que concentra mais níveis de resolução próximos à média dos dados e menos nos extremos, melhorando assim a acurácia do modelo [8, 6].

2 Proposta

2.1 Metodologia

Neste trabalho, iremos replicar o BTHOWeN utilizando o BloomWisard como cerne, com os diferentes datasets multiclasse inclusos no artigo original. A proporção entre a massa de treinamento e a massa de teste foi mantida idêntica às respectivas proporções originais,

porém os hiperparâmetros foram configurados para verificar o impacto de cada um deles na acurácia final.

Os hiperparâmetros configuráveis são:

- Tamanho do endereço (ou tamanho da tupla).
- Número de discriminadores.
- Número de funções hash.
- Número de filtros de bloom.
- Fator do bleaching.

Os experimentos são feitos alterando um hiperparâmetro de cada vez, de maneira gulosa. Ao atingir um valor máximo variando apenas um hiperparâmetro, este tem seu valor mantido pelo resto do experimento, e o próximo hiperparâmetro passa a ser o variável. Ao realizar este procedimento com todos os hiperparâmetros, identificamos o melhor resultado.

Tomamos o maior valor de acurácia de todos os experimentos e sua configuração de hiperparâmetros como melhor valor obtido e o comparamos com a acurácia obtida no artigo original.

2.2 Hiperparâmetros

2.2.1 Tamanho do Endereço (Tamanho da Tupla)

O tamanho do endereço, também conhecido como tamanho da tupla, é um parâmetro da WiSARD que determina a quantidade de bits de entrada agrupados para formar endereços nos filtros de Bloom do BTHOWeN [2, 4]. Este parâmetro afeta o número total de filtros necessários (entradas totais divididas pelo tamanho da tupla) e o espaço de endereçamento de cada filtro.

Tuplas menores resultam em mais filtros e melhor generalização, enquanto tuplas maiores reduzem o número de filtros mas podem comprometer a generalização [6]. Nos experimentos, este parâmetro varia conforme o dataset, desde valores pequenos (2 para Iris) até valores maiores (28 para MNIST).

2.2.2 Codificação Termômetro Baseada em Distribuição Gaussiana

O BTHOWeN implementa uma codificação termômetro não-linear baseada na distribuição gaussiana dos valores de cada característica, diferenciando-se da codificação termômetro padrão com limiares uniformemente espaçados [6]. Nesta abordagem, se cada característica é codificada com t bits, seu intervalo de valores é dividido em $t+1$ intervalos com densidade de probabilidade igual sob uma distribuição normal ajustada aos dados de treinamento [8].

Esta técnica posiciona limiares mais próximos entre si perto da média dos dados (maior densidade) e mais espaçados nas caudas da distribuição, alocando maior resolução onde os dados são mais comuns [6]. Isso otimiza a precisão nas faixas de valores mais frequentes, evitando o desperdício de bits em outliers raros que ocorre com limiares uniformes, o que resulta em melhor performance de classificação.

2.2.3 Número de Bits por Filtro (OWeN)

O parâmetro OWeN (Optimized Weightless Neural Network) define o número de bits de entrada mapeados para cada filtro de Bloom dentro de um discriminador [6], determinando como o vetor de entrada é particionado entre os filtros. Por exemplo, em um modelo MNIST com entrada de 784 dimensões e $OWeN = 28$, cada discriminador conteria 28 filtros de Bloom, cada um processando 28 bits específicos da entrada.

A escolha do valor OWeN envolve um trade-off importante [6, 8]: valores mais baixos resultam em mais filtros com "campos receptivos" menores, potencialmente melhorando a generalização mas aumentando a quantidade de filtros necessários; valores mais altos significam menos filtros com "campos receptivos" maiores, reduzindo a complexidade do modelo mas

aumentando o risco de colisões. Nos experimentos do BTHOWeN, os valores de OWeN foram sistematicamente testados para cada dataset (por exemplo, 28, 49 e 56 bits por filtro para MNIST) [6].

2.2.4 Número de Funções Hash (FH)

O parâmetro FH determina quantas diferentes funções hash são utilizadas em cada filtro de Bloom para mapear os dados de entrada em posições de memória [6]. O BTHOWeN implementa a família de funções hash H3, que consiste apenas em operações XOR entre bits de entrada e máscaras aleatórias, tornando-a ideal para implementações em hardware [8]. Uma característica importante é que o BTHOWeN compartilha os mesmos parâmetros aleatórios de hash entre todos os filtros para economizar memória, sem degradar a acurácia [6].

O número de funções hash tem uma relação complexa com a taxa de falsos positivos: valores maiores reduzem falsos positivos (todas as k posições precisam colidir para causar um falso positivo), mas aumentam o custo computacional e o número de posições a serem definidas por padrão [6]. Valores típicos variam de 1 a 6 dependendo do dataset, com o valor ótimo sendo determinado empiricamente para cada caso.

2.2.5 Número de Filtros de Bloom (FE - Filter Entries)

O parâmetro FE define o número de entradas em cada filtro de Bloom, ou seja, o tamanho do vetor utilizado para armazenar os padrões aprendidos [6]. Este tamanho é sempre uma potência de 2 (128, 256, 512, 1024, 2048, etc.).

Aumentar o tamanho do filtro reduz a probabilidade de colisões e a taxa de falsos positivos, mas aumenta a memória necessária [6, 8]. Este parâmetro afeta diretamente o equilíbrio entre precisão e eficiência de memória do modelo e, em implementações de hardware, traduz-se na quantidade de memória alocada para cada filtro, influenciando diretamente o consumo de energia [8].

2.2.6 Limiar de Bleaching (b)

O limiar de bleaching é um hiperparâmetro exclusivo da arquitetura BTHOWeN [6]. Enquanto filtros de Bloom tradicionais armazenam apenas bits (0 ou 1), no BTHOWeN cada posição mantém um contador durante o treinamento, registrando a frequência de cada padrão [6, 3]. O parâmetro b define o valor mínimo que um contador deve ter para que um padrão seja considerado válido durante a inferência, funcionando como mecanismo de regularização que filtra padrões pouco frequentes, evitando sobreajuste [1].

Após o treinamento, um valor ótimo de b é determinado usando um conjunto de validação, geralmente via busca binária [6, 8]. Os contadores são então binarizados: posições com contadores $\geq b$ se tornam 1, e as demais 0. Na inferência, o modelo considera apenas padrões vistos pelo menos b vezes durante o treinamento. Este processo permite distinguir padrões relevantes de ocorrências aleatórias, melhorando a generalização e beneficiando datasets desbalanceados, onde classes majoritárias poderiam saturar a memória sem este mecanismo [6]. Valores típicos de b variam de 1 a 20, sendo específicos para cada dataset e configuração.

2.3 Escolha dos Hiperparâmetros

A seleção de hiperparâmetros para o BTHOWeN segue uma abordagem iterativa que explora sistematicamente o espaço de configurações. Partindo de baselines para cada dataset, o processo altera um hiperparâmetro por vez, testando múltiplas variações: cinco para `bits_per_input`, seis para `filter_inputs`, cinco para `filter_entries` e quatro para `filter_hashes`. Seguimos uma estratégia gulosa, fixando o melhor valor de cada parâmetro antes de passar ao próximo, o que oferece eficiência computacional e facilita a análise do impacto individual de cada hiperparâmetro.

Para mitigar o risco de convergência para ótimos locais, repetimos o processo com múltiplos pontos de partida e validamos as configurações resultantes com validação cruzada 5-fold. A

configuração final é selecionada com base no equilíbrio entre acurácia e robustez, seguida pela determinação do valor ideal de bleaching (testando valores de 1 a 20). Esta metodologia, embora computacionalmente intensiva, garante uma exploração abrangente do espaço de hiperparâmetros e produz configurações que equilibram eficazmente acurácia, eficiência de memória e estabilidade.

2.4 Datasets

2.4.1 MNIST

O dataset **MNIST** (Modified National Institute of Standards and Technology) é uma coleção de dígitos manuscritos. Ele inclui **60k imagens de treinamento** e **10k imagens de teste**. Todas as imagens estão em escala de cinza e possuem tamanho de **28×28 pixels**.

2.4.2 Ecoli

O dataset **Ecoli** é usado para prever onde proteínas celulares se localizam com base em suas sequências de aminoácidos. Ele contém **336 proteínas**, cada uma descrita por **sete atributos numéricos** derivados da sequência. As proteínas são classificadas em **oito possíveis locais celulares**.

2.4.3 Iris

O dataset **Iris** contém **150 observações** de flores de íris, cada uma descrita por:

- Comprimento da sépala.
- Largura da sépala.
- Comprimento da pétala.
- Largura da pétala.

A classificação é feita em uma de **três espécies**: **Iris Setosa**, **Versicolor** ou **Virginica**.

2.4.4 Glass

O dataset **Glass** contém **214 instâncias** de fragmentos de vidro, cada uma descrita por **10 atributos**. Com ele, conseguimos prever o tipo de vidro com base em sua composição química e índice de refração.

2.4.5 Letter

O dataset **Letter** contém letras manuscritas. As imagens dos caracteres foram baseadas em **20 fontes diferentes**, e cada letra dentro dessas fontes foi distorcida aleatoriamente para produzir **20k entradas**, onde **16k** foram usadas para treinamento e **4k** para teste.

2.4.6 Wine

O dataset **Wine** reúne **178 amostras** de vinho de três cultivares de uvas da região de Piemonte, Itália. Cada amostra é descrita por **13 atributos químicos contínuos** – como teor de álcool, magnésio e intensidade de cor. O objetivo é identificar a cultivar correta entre as **três classes** disponíveis.

2.4.7 Segment

O dataset **Image Segmentation** (Segment) contém **2.310 segmentos** de imagens externas, distribuídos igualmente entre **sete classes** de região: tijolo, céu, folhagem, cimento, janela, caminho e grama. Cada segmento é representado por **19 atributos numéricos** que descrevem características de cor e textura. A tarefa é prever a classe da região na qual o segmento se enquadra.

2.4.8 Shuttle

O dataset **Statlog Shuttle** possui cerca de **58.000 registros** de telemetria do sistema de controle do ônibus espacial. Cada registro é descrito por **nove atributos numéricos** derivados de sensores a bordo e está rotulado em um de **sete possíveis estados operacionais**. O conjunto é fortemente desbalanceado, com predominância da classe 1.

2.4.9 Vehicle

O dataset **Vehicle Silhouettes** contém **846 silhuetas** de veículos, cada uma descrita por **18 medidas geométricas** extraídas da imagem, como área, compactidade e momentos. Os veículos devem ser classificados em uma de **quatro categorias**: ônibus, Opel, Saab ou van.

2.4.10 Vowel

O dataset **Vowel Recognition** inclui **990 amostras** de fala gravadas por 15 locutores. Cada amostra é caracterizada por **10 atributos acústicos contínuos** (coeficientes derivados do espectro) e deve ser classificada em uma de **11 vogais** do inglês, como /a/, /e/ ou /i/.

3 Resultados

3.1 Resultados por dataset

Nas tabelas a seguir, apresentamos os resultados obtidos com diferentes configurações do BTHoWeN para cada dataset. As configurações são organizadas da seguinte forma:

- **Bloom WiSARD**: Resultados obtidos pela implementação original do Bloom WiSARD, apresentados como referência para comparação.
- **Base**: Configuração inicial do BTHoWeN utilizada como ponto de partida para os experimentos, com parâmetros utilizados nos experimentos originais publicados pelo autor.
- **Var. X**: Variações da configuração base, onde modificamos sistematicamente um hiperparâmetro por vez, seguindo a abordagem gulosa descrita na metodologia. Os números indicam a sequência das variações testadas.

Para cada configuração, apresentamos os valores dos hiperparâmetros (b, OWeN, FE, FH), a acurácia obtida, o percentual de empates ocorridos durante a classificação e o valor de bleaching que produziu os melhores resultados. A coluna “Bleaching” indica o valor ótimo encontrado durante o treinamento, que permite ao modelo obter a melhor acurácia possível.

3.1.1 Tabelas

Tabela 1: Parâmetros e métricas do dataset Iris

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	960	2.0	3
Base	3	2	128	1	980	12.0	2
Var. 1	4	2	128	1	920	16.0	1
Var. 2	3	2	256	1	980	14.0	2
Var. 3	3	2	128	2	980	8.0	2
Var. 4	4	2	128	2	860	4.0	9
Var. 5	3	2	256	2	980	0.0	2
Var. 6	4	2	256	2	900	12.0	3

Tabela 2: Parâmetros e métricas do dataset Ecoli

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	799	N/A	N/A
Base	10	128	2	10	786	8.9	7
Var. 1	4	128	2	11	821	10.7	1
Var. 2	3	256	2	10	813	19.6	1
Var. 3	3	128	3	10	786	15.2	7
Var. 4	4	128	3	11	839	17.9	1
Var. 5	3	256	3	10	848	10.7	1
Var. 6	4	256	4	10	830	13.4	1

Tabela 3: Parâmetros e métricas do dataset Glass

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	726	N/A	N/A
Base	3	128	3	9	577	39.4	1
Var. 1	4	128	3	10	563	38.0	1
Var. 2	3	256	3	9	493	33.8	1
Var. 3	3	128	4	9	549	19.7	4
Var. 4	4	128	4	10	592	40.8	1
Var. 5	3	256	4	9	676	29.6	1
Var. 6	4	256	4	10	676	28.2	1

Tabela 4: Parâmetros e métricas do dataset Letter

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	848	N/A	N/A
Base	3	128	3	9	734	18.6	6
Var. 1	4	128	3	10	736	17.4	5
Var. 2	3	256	3	9	789	15.2	3
Var. 3	3	128	4	9	707	19.2	5
Var. 4	4	128	4	10	719	18.7	6
Var. 5	3	256	4	9	775	15.6	4
Var. 6	4	256	5	12	811	11.3	4
Var. 7	11	256	5	18	840	7.6	3
Var. 8	15	256	5	35	884	3.9	3

Tabela 5: Parâmetros e métricas do dataset Wine

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	926	N/A	N/A
Base	9	13	128	3	983	N/A	1
Var. 1.1	10	13	256	4	1.000	1.7	1
Var. 1.1	10	13	256	4	949	3.39	1
Var. 2	11	9	256	4	983	1.69	1
Var. 3	11	13	256	2	966	3.38	1
Var. 4	9	9	256	4	966	1.69	1
Var. 5	10	17	256	4	966	11.8	1
Var. 6	11	15	256	2	966	N/A	N/A
Var. 7	7	9	256	4	932	5.08	1

Tabela 6: Parâmetros e métricas do dataset Segment

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Base	9	27	1.14	2	925	N/A	1
Var. 1	10	16	256	4	938	9.35	1
Var. 2	8	20	512	3	924	9.24	1
Var. 3.1	10	18	1.14	3	942	8.57	2
Var. 3.2	10	18	1.14	3	944	6.49	2
Var. 4	10	20	512	2	944	8.96	1
Var. 5	16	16	256	3	941	10.8	8
Var. 6	10	14	512	4	939	8.10	2
Var. 7	9	20	1.14	2	937	23.7	1
Var. 8	15	15	256	4	936	9.8	1
Var. 9	9	32	2048	4	936	34.9	1

Tabela 7: Parâmetros e métricas do dataset Shuttle

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	868	N/A	N/A
Base	9	27	1024	2	0	0	0
Var. 1.1	11	29	1024	2	999	0.11	1
Var. 1.1	11	29	1024	2	998	0.17	1
Var. 2	11	25	1024	3	999	0.10	1
Var. 3	8	27	1024	1	998	0.21	4
Var. 4	9	23	512	3	998	0.21	8
Var. 5	8	23	2048	1	998	0.70	1
Var. 6	7	27	1024	2	989	2.55	5
Var. 8	11	27	1024	2	976	4.99	276

Tabela 8: Parâmetros e métricas do dataset Vehicle

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	926	N/A	N/A
Base	16	16	256	3	N/A	N/A	N/A
Var. 1	14	14	512	4	755	32.1	1
Var. 11	15	12	256	2	755	20.2	1
Var. 19	18	16	512	2	748	30.3	1
Var. 9	18	18	512	3	737	25.5	1
Var. 18	16	14	512	2	734	20.2	1
Var. 10	15	16	512	3	726	32.8	1
Var. 15	14	12	512	3	726	27.5	1

Tabela 9: Parâmetros e métricas do dataset Vowel

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
Bloom WiSARD	N/A	N/A	N/A	N/A	876	N/A	N/A
Base	15	15	256	4	0	0	0
Var. 16	15	13	512	5	924	24.4	1
Var. 8	16	11	256	3	918	21.8	1
Var. 9	16	11	256	5	918	0	0
Var. 19	17	11	512	5	918	23.0	1
Var. 18	14	11	128	3	912	28.8	1
Var. 17	16	13	512	5	909	0	0
Var. 5	16	17	256	5	906	0	0

Tabela 10: Parâmetros e métricas do dataset MNIST

Configuração	b	OWeN	FE	FH	Acurácia	Empates (%)	Bleaching
ULEEN	6	49	81.1	4	952	N/A	N/A
Base	2	28	1.14	2	929	1.61	8
Var. 1	16	16	256	3	915	N/A	N/A
Var. 2	15	15	256	4	913	N/A	N/A
Var. 3	9	27	1.14	2	933	N/A	N/A
Var. 4	4	16	512	2	918	2.1	16
Var. 5	8	20	512	3	921	N/A	N/A
Var. 6	4	24	256	2	916	N/A	N/A
Var. 7	8	32	2048	4	943	0.36	6

3.2 Gráficos

Nas figuras a seguir, apresentamos gráficos de dispersão que ilustram a relação entre taxa de erro (1 - acurácia) e taxa de empates para diferentes configurações do BTHOWeN em cada dataset. Os pontos são categorizados em:

- **Melhor ponto:** Configuração que obteve o melhor desempenho (azul)
- **Melhor que as bases:** Configurações que superaram a linha base (verde)
- **Demais variações:** Outras configurações testadas (vermelho)

As linhas de referência indicam:

- **Referência Bloom WiSARD:** Desempenho da implementação de referência do Bloom WiSARD (linha pontilhada)
- **BTHOWeN Base Estudo:** Desempenho da configuração base do BTHOWeN utilizada como ponto de partida (linha tracejada)

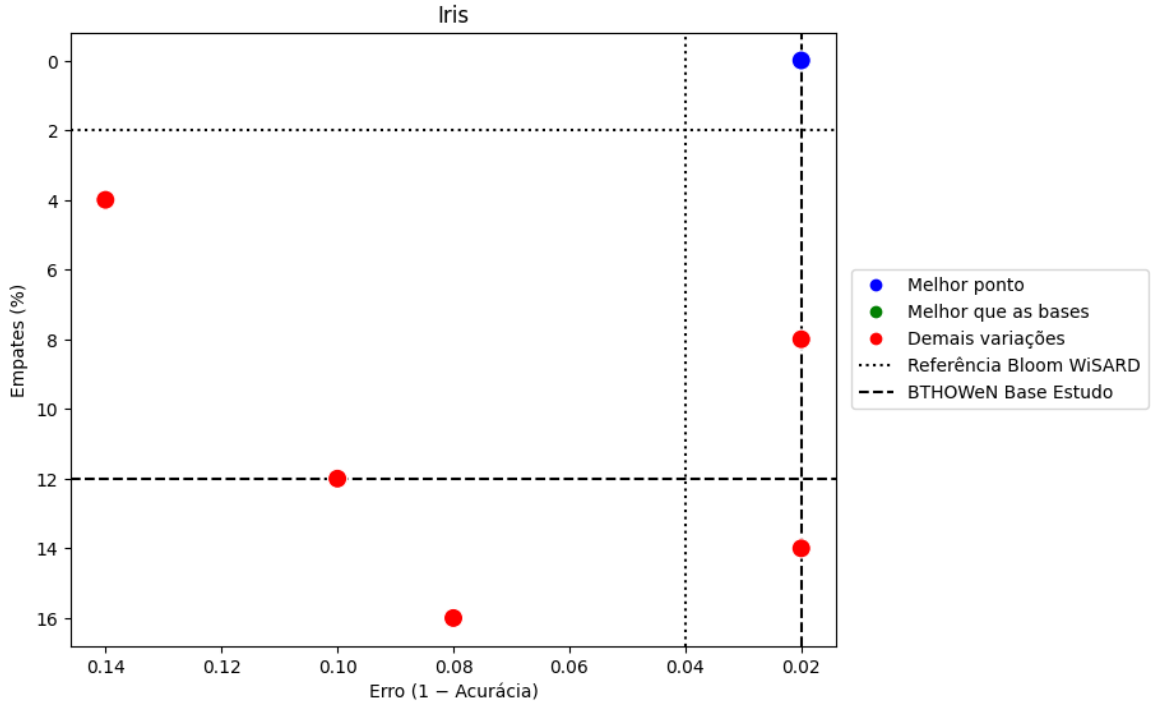


Figura 1: Relação entre erro e empates para o dataset Iris

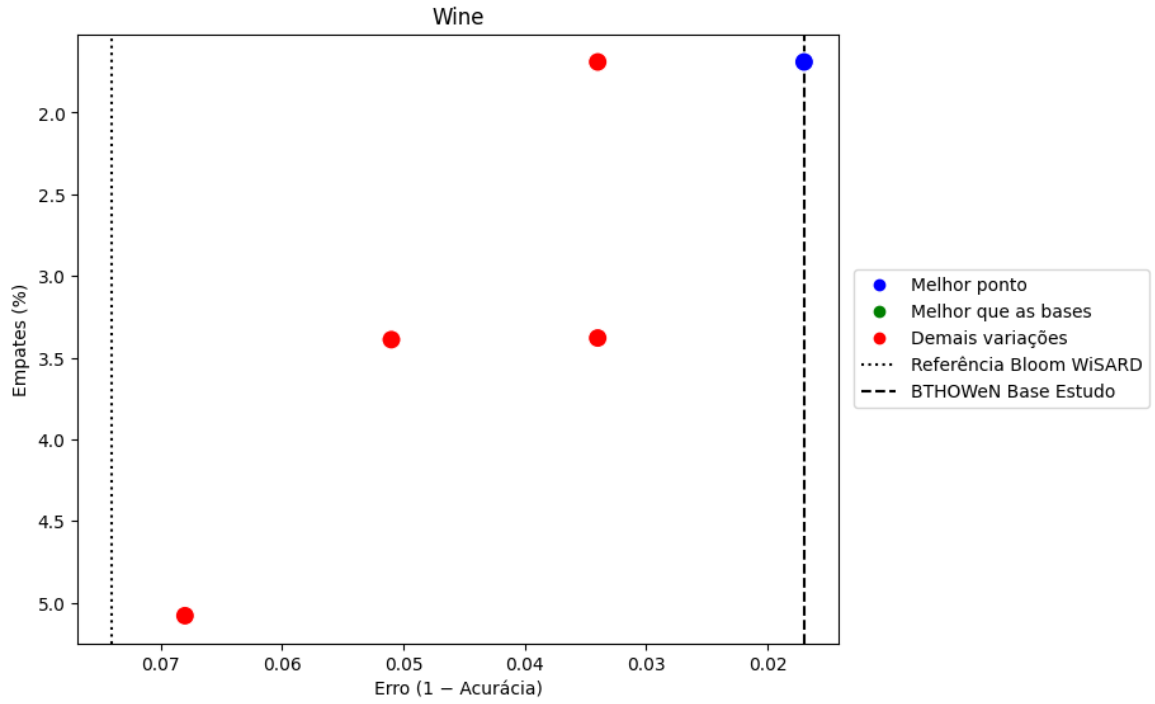


Figura 2: Relação entre erro e empates para o dataset Wine

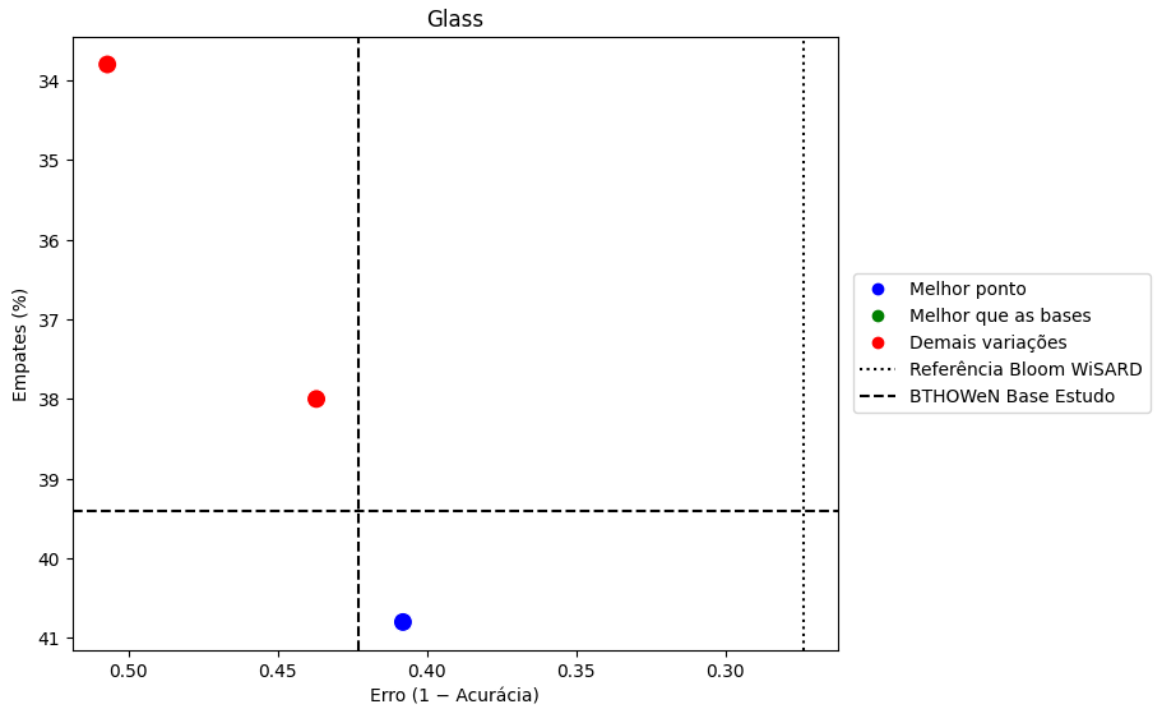


Figura 3: Relação entre erro e empates para o dataset Glass

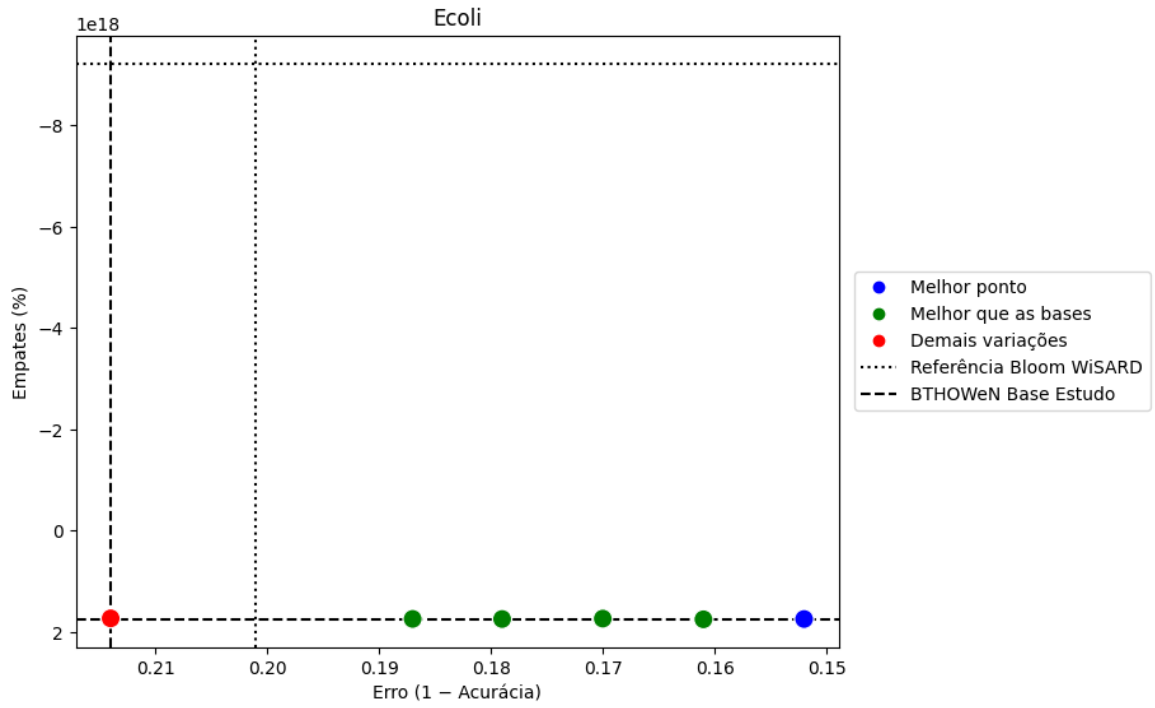


Figura 4: Relação entre erro e empates para o dataset Ecoli

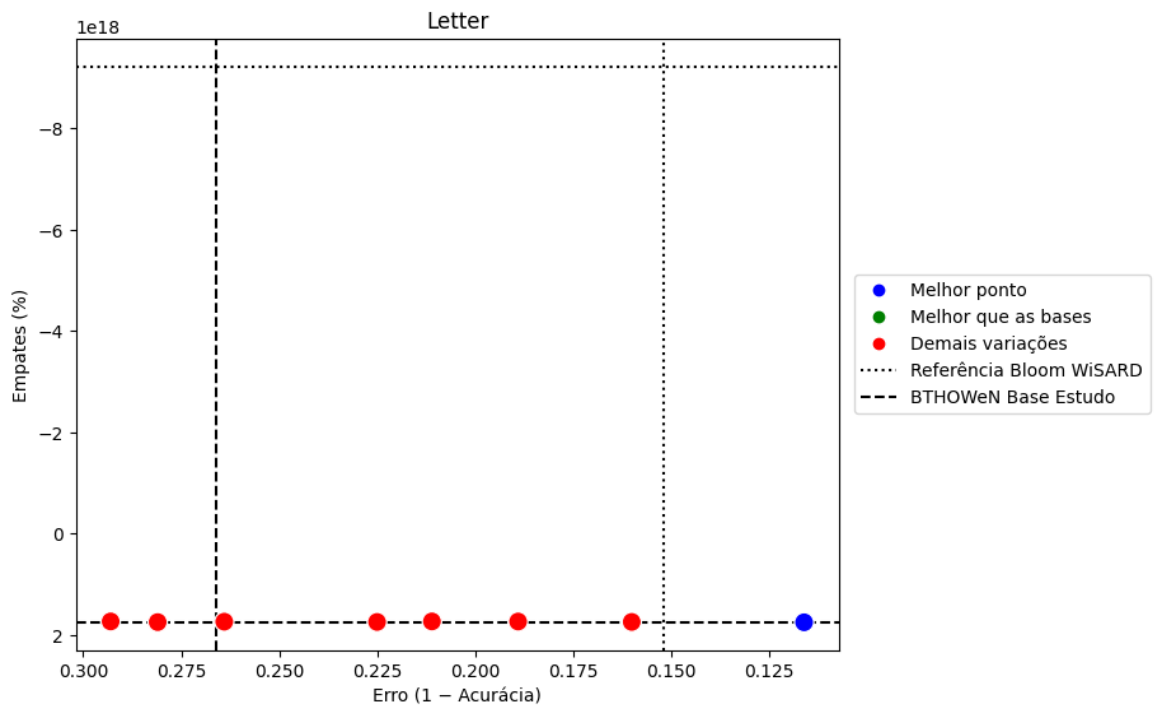


Figura 5: Relação entre erro e empates para o dataset Letter

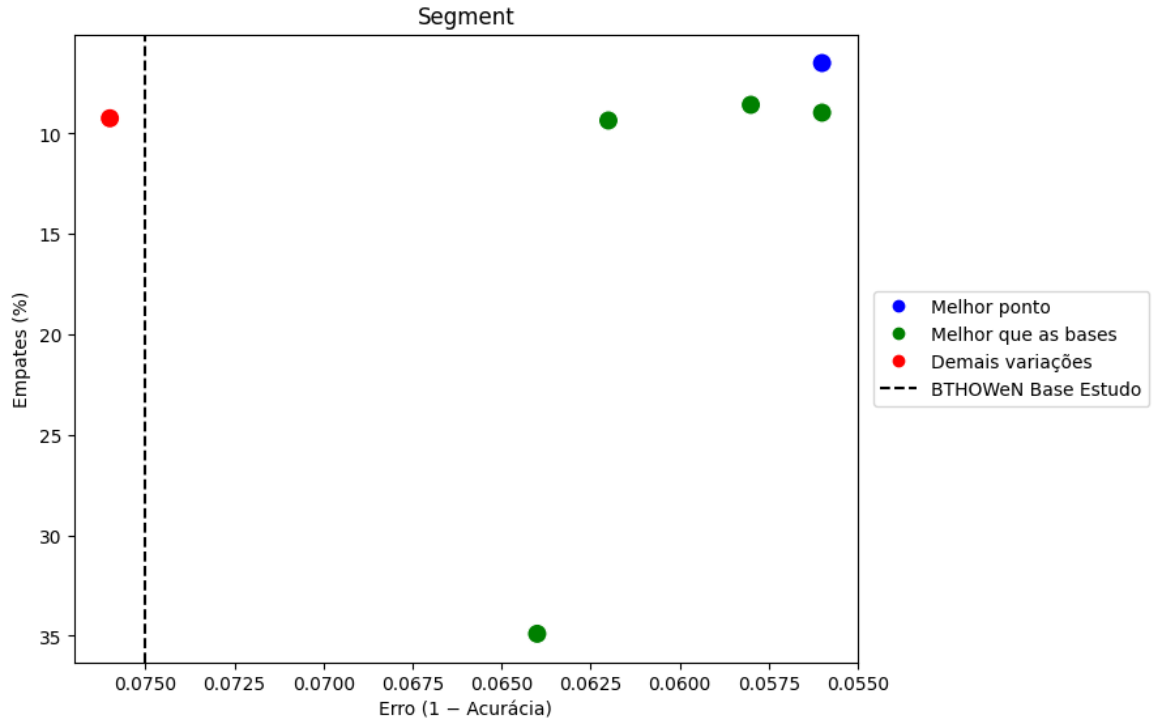


Figura 6: Relação entre erro e empates para o dataset Segment

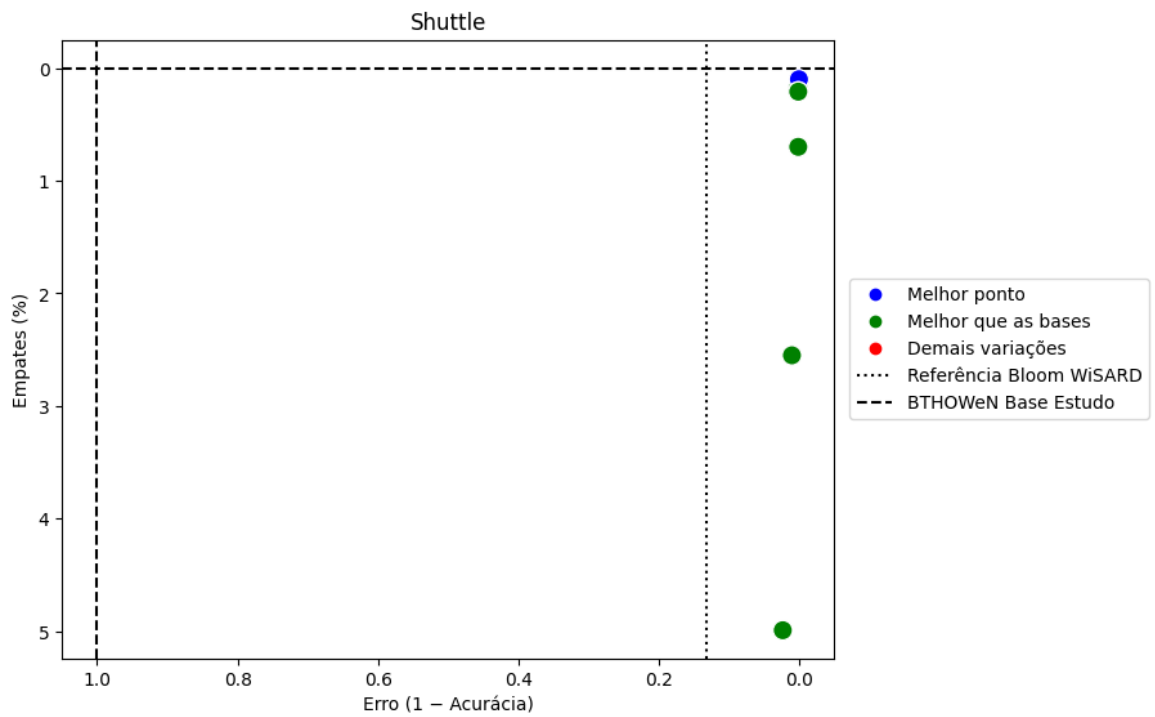


Figura 7: Relação entre erro e empates para o dataset Shuttle

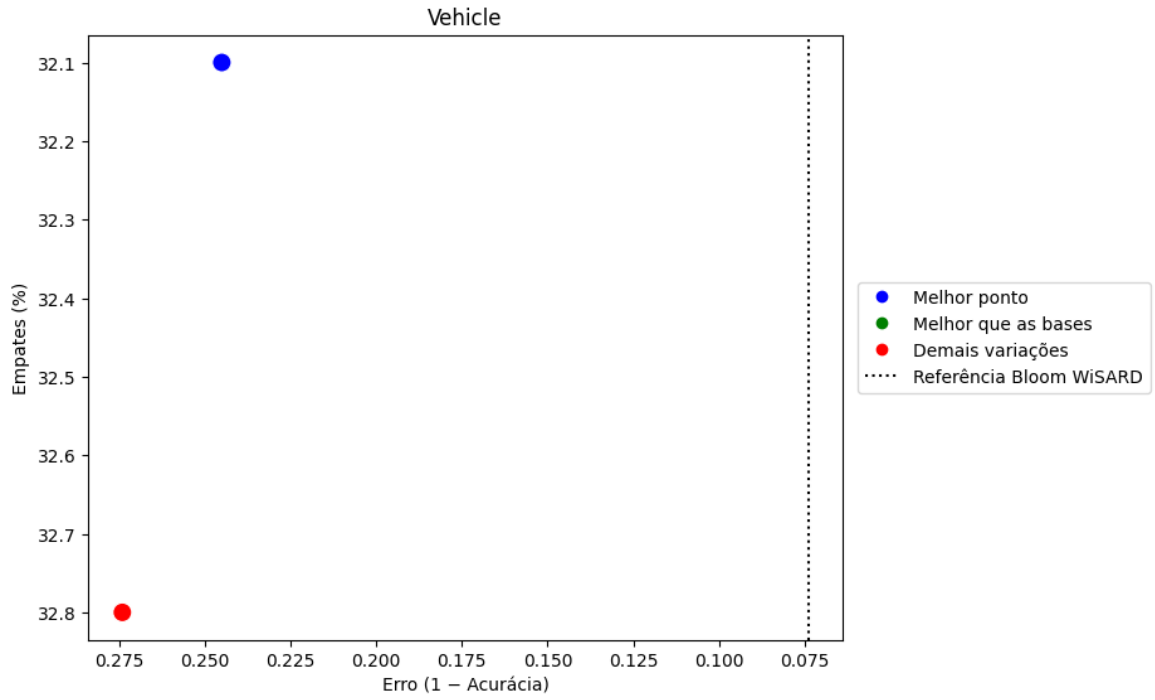


Figura 8: Relação entre erro e empates para o dataset Vehicle

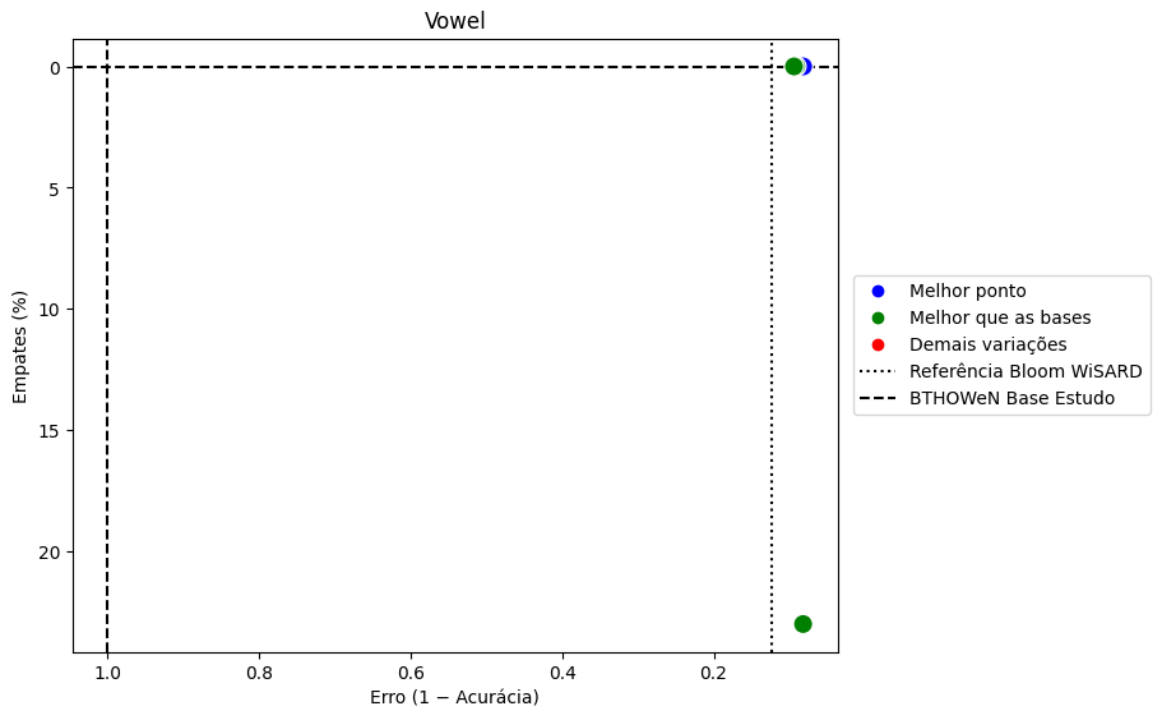


Figura 9: Relação entre erro e empates para o dataset Vowel

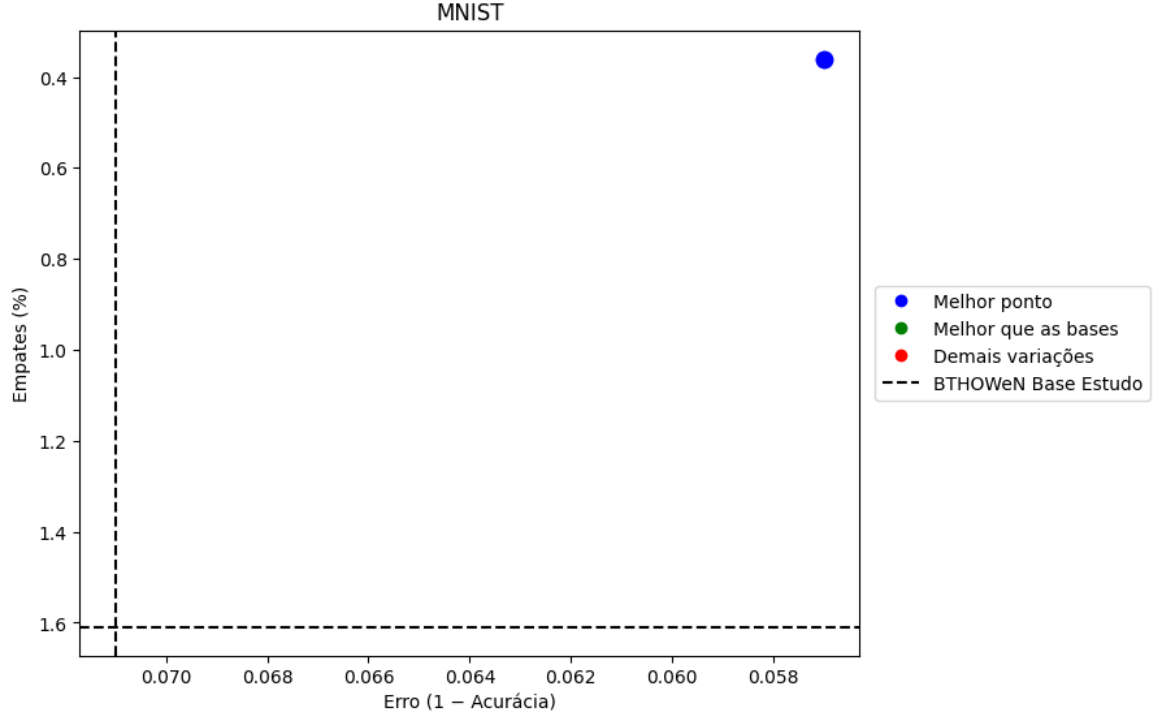


Figura 10: Relação entre erro e empates para o dataset MNIST

3.3 Resultados agregados

Apresentamos na Tabela 1 a comparação de acurácia e percentual de empates entre a implementação de referência (Bloom WiSARD), nossa configuração base e a melhor configuração alcançada para cada dataset.

Tabela 11: Comparação de acurácia (Ac , $\times 10^{-3}$) e percentual de empates (Emp , %) para cada dataset. Ref = implementação de referência (Bloom WiSARD), Base = configuração inicial, Melhor = melhor configuração obtida. Valores em **negrito** indicam o melhor resultado por métrica na linha.

Dataset	Ac Ref	Ac Base	Ac Melhor	Emp Ref	Emp Base	Emp Melhor
Iris	960	980	980	2	12	0
Ecoli	799	786	848	—	8.9	10.7
Glass	726	577	676	—	39.4	28.2
Letter	848	734	884	—	18.6	3.9
Wine	926	983	1000	—	—	1.7
Segment	—	925	944	—	—	6.49
Shuttle	868	0	999	—	0	0.1
Vehicle	926	—	755	—	—	20.2
Vowel	876	0	924	—	0	24.4

A seguir, na Tabela 12, apresentamos os parâmetros ótimos identificados para cada dataset, junto com as métricas de desempenho obtidas:

Tabela 12: Parâmetros ótimos para cada dataset. Em **vermelho** destacamos valores menores que os observados na referência Bloom WiSARD; em **verde** os valores maiores.

Dataset	b	OWeN	FE	FH	Melhor Bleaching	Execução	Empates (%)
Iris	3	2	256	2	2	1	0
Ecoli	3	256	3	10	1	1	10.7
Glass	4	256	4	10	1	1	28.2
Letter	15	256	5	35	3	1	3.9
Wine	10	13	256	4	1	—	1.7
Segment	10	18	1024	3	2	—	6.49
Shuttle	11	25	1.14	3	1	—	0.1
Vehicle	15	12	256	2	1	—	20.2
Vowel	15	13	512	5	1	—	24.4
MNIST	6	49	8192	4	—	—	—

4 Conclusão

Referências

- [1] Danilo Carvalho, Hugo Carneiro, Felipe M. G. França, and Priscila M. V. Lima. B-bleaching: Agile overtraining avoidance in the wisard weightless neural classifier. *ESANN 2013*, pages 267–272.
- [2] Aluizio S. Lima Filho, Gabriel P. Guarisa, Leopoldo A. D. Lusquino Filho, Luiz F. R. Oliveira, Felipe M. G. França, and Priscila M. V. Lima. wisardpkg - a library for wisard-based models. *arXiv preprint arXiv:2005.00887*, 2020.
- [3] Leopoldo A. D. Lusquino Filho, Luiz F. R. Oliveira, Aluizio Lima Filho, Gabriel P. Guarisa, Lucca M. Felix, Priscila M. V. Lima, and Felipe M. G. França. Extending the weightless wisard classifier for regression. *Neurocomputing*, 419:125–138, 2020.
- [4] Leandro Santiago. Redes sem peso wisard - soluções para memória eficiente, 2023. Apresentação sobre modelos WiSARD e otimizações de memória.
- [5] Leandro Santiago, Felipe M. G. França, Priscila M. V. Lima, and Maurício Breternitz Jr. Memory-efficient weightless neural network using bloom filter. In *ESANN 2019: European Symposium on Artificial Neural Networks*, 2019.
- [6] Leandro Santiago, Leticia Verona, Fabio Rangel, Fabrício Firmino, Daniel S. Menasché, Wouter Caarls, Mauricio Breternitz Jr, Sandip Kundu, Priscila M. V. Lima, and Felipe M. G. França. Weightless neural networks as memory segmented bloom filters. *Neurocomputing*, 416:354–366, 2020.
- [7] Zachary Susskind. *Weightless Neural Networks for Fast, Low-Energy Inference*. PhD thesis, The University of Texas at Austin, 2024.
- [8] Zachary Susskind, Aman Arora, Igor D. S. Miranda, Luis A. Q. Villon, Rafael F. Katopodis, Leandro S. Araújo, Diego L. C. Dutra, Priscila M. V. Lima, Felipe M. G. França, Mauricio Breternitz Jr, and Lizy K. John. Weightless neural networks for efficient edge inference. In *PACT ’22: International Conference on Parallel Architectures and Compilation Techniques*, New York, NY, USA, 2022. ACM.