

Otimização do Consumo de Energia em Redes *Ad Hoc* Aloha Empregando *Deep Learning*

Paulo F. C. Barbosa¹, Bruna A. da Silva¹, David Macêdo¹,
Cleber Zanchettin¹, Renato M. de Moraes¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP 50740-560 – Recife – PE – Brasil

{pfc, bas4, dlm, cz, renatomdm}@cin.ufpe.br

Abstract. *The algorithms commonly used for energy control in IoT networks involve optimization functions with considerable complexity and rigorous control of the test environment. It creates a gap between design, theoretical analysis and real-time processing of the network devices. In this paper, we propose a novel approach based on machine learning which considers the input and output of a power consumption control algorithm in multi-variable slotted Aloha ad hoc networks. Results show that the proposed neural network presented better performance concerning processing time and computational cost when compared to the currently used greedy search energy control algorithms.*

Resumo. *Os algoritmos normalmente empregados para controle energético em redes IoT envolvem funções de otimização com considerável complexidade e controle rigoroso do ambiente de teste. Isso gera uma lacuna entre o projeto, análise teórica e processamento em tempo real dos dispositivos da rede. O presente artigo propõe uma nova abordagem baseada em aprendizagem de máquina que considera a entrada e a saída de um algoritmo de controle de consumo de energia em redes ad hoc slotted Aloha de múltiplas variáveis. Resultados mostram que a rede neural proposta obteve melhor desempenho em relação ao tempo de processamento e custo computacional quando comparado aos algoritmos de controle energético de busca gulosa utilizados atualmente.*

1. Introdução

O gerenciamento de recursos como velocidade ideal de transmissão, tamanho do pacote de dados e distância entre os nós, por exemplo, são variáveis críticas para um controle energético eficiente, principalmente em dispositivos de baixo consumo de energia como os utilizados em redes IoTs (do inglês *Internet of Things*). Por este motivo, estudos vêm sendo realizados para definir o melhor esquema de gerenciamento desses recursos [Fadlullah et al. 2017].

Por décadas, a otimização através de funções numéricas desempenhou um papel central no tratamento dos problemas de gerenciamento de recursos em redes sem fio. Modelos e algoritmos destinados à otimização do emprego dos recursos energéticos vêm sendo desenvolvidos ao longo do tempo a fim de proporcionar um melhor desempenho em redes IoT sem fio [Sun et al. 2017].

Um número significativo destas técnicas pertencem à classe de algoritmos iterativos que trabalham com um determinado conjunto de parâmetros de rede em tempo real

(ex: realizações de canal e especificações de relação sinal-ruído) como entradas e consequentemente executam uma série de iterações (potencialmente caras) para a geração de estratégias de gerenciamento de recursos otimizadas como suas saídas [Mao et al. 2018].

Apesar desses algoritmos apresentarem excelentes resultados, muitos deles foram aplicados, simulados e observados somente através de computações numéricas e análises teóricas. A implementação em sistemas reais ainda enfrenta muitos obstáculos, principalmente em dispositivos de baixo desempenho e com pouco recurso energético. Isso se dá, em particular, pelo alto custo computacional incorrido por essas técnicas. Algoritmos do tipo WMMSE (do inglês *Weighted Sum MSE Minimization*), por exemplo, requerem operações complexas como inversão de matrizes e bissecção em cada iteração para converter o problema de maximização da taxa de soma ponderada WSR (do inglês *Weighted Sum-Rate*) em um espaço dimensional mais alto para facilitar sua execução [Shi et al. 2011], [Baligh et al. 2014].

Modelagens como a proposta em [Da Silva and De Moraes 2018] precisam de operações custosas como uma busca exaustiva para encontrar as melhores configurações que compõem as estratégias ideais para cada cenário de transmissão. Desta forma, é desenvolvido um esquema complexo de ponderação de pesos de variáveis, como a taxa de transmissão e o tamanho do pacote de dados, envolvidas na potência ótima necessária para otimizar os recursos energéticos exigidos durante as transmissões.

A natureza das soluções de gerenciamento de recursos em redes sem fio encontradas tornam sua aplicação em dispositivos que precisam de resposta em tempo real desafiadora. Isso porque as tarefas de gerenciamento são normalmente executadas em um período de milissegundos devido aos parâmetros do sistema, como condições do canal, número de usuários, entre outros, as quais mudam rapidamente.

Este trabalho propõe uma nova abordagem baseada em aprendizado de máquina para o gerenciamento dos recursos envolvidos no controle energético em redes *ad hoc slotted Aloha* para aplicações em IoT. A ideia principal é considerar um algoritmo de otimização tradicional, usado no gerenciamento de recursos em redes sem fio, como um dispositivo complexo e aprender sua relação entrada/saída usando uma rede neural profunda (DNN, do inglês *Deep Neural Network*) [LeCun et al. 2015].

A avaliação deste trabalho é motivada com base na hipótese de que, se uma DNN puder aproximar seus resultados de uma solução tradicional, ainda que com eventual erro residual, será possível aplicá-la na criação de protocolos de comunicação para sistemas reais, principalmente para economia de recursos computacionais e consequentemente energéticos. Isso se deve ao fato de que uma DNN treinada se limita a operações simples como multiplicação de vetores e funções de limiarização para obtenção dos seus resultados, diferente dos algoritmos tradicionais que envolvem operações mais complexas e custosas computacionalmente.

Portanto, essa aproximação, dependendo da precisão requerida, pode reduzir substancialmente o tempo de processamento para alocação de recursos dos dispositivos IoTs. Além disso, o treinamento dessa rede é bastante conveniente tendo em vista que as amostras para seu treinamento são obtidas utilizando apenas algoritmos de otimização tradicionais em dados simulados conforme proposto em [Mao et al. 2018].

O restante deste artigo está dividido da seguinte forma. A Seção 2 discute os

trabalhos relacionados. A Seção 3 descreve o método e a técnica propostos. A Seção 4 descreve como os experimentos foram realizados. Os resultados são apresentados e discutidos na Seção 5 e as considerações finais são descritas na Seção 6.

2. Trabalhos Relacionados

As redes IoT, no geral, são constituídas de muitos nós sensores com poder computacional e recursos energéticos limitados. Uma vez que o tempo de vida do nó depende da capacidade de suas baterias, faz-se necessário, dentre tantos fatores, a otimização dos recursos de *hardware*, redução do custo de processamento e operação dos circuitos de transmissão e recepção.

Variáveis como taxa de transferência e taxa de perda de pacotes podem ser aprimoradas ao projetar um protocolo inteligente capaz de lidar com grandes volumes de dados durante as transmissões. Isso significa que redes IoT precisam de técnicas que garantam estabilidade e robustez na comunicação entre os nós [Kapoor et al. 2018].

Para resolução desses problemas, trabalhos como [Mohammadi et al. 2018, Mamdouh et al. 2018] aplicam técnicas de classificação e regressão com Aprendizagem Profunda (*Deep Learning*) em diversas áreas de redes de comunicação, demonstrando um desempenho promissor em várias tarefas como identificação de fluxo, detecção de intrusão, roteamento, *compressed sensing*, predição de tráfego e detecção de sinais, por exemplo. [Mao et al. 2018] propõe o uso de *Deep Learning* para aproximação de algoritmos exatos, agregando baixo volume de processamento de dados à resoluções matemáticas eficientes.

De início, pode parecer incerto o uso de uma rede neural para aproximar o comportamento de um algoritmo do tipo iterativo, porém [Sun et al. 2017] mostra que o processo utilizado para o treinamento da rede neural manipula gradientes que, dependendo das entradas, poderão convergir para uma solução de maximização ou minimização ótima.

Além disso, [Sun et al. 2017] verificou que na aproximação de algoritmos exatos multiobjetivos como WMMSE que representa o mapeamento contínuo das variáveis otimizadas gerando um conjunto compacto de soluções, pode ser aproximado por uma rede neural *feedforward* com apenas uma camada oculta.

Alguns resultados obtidos por [Huan et al. 2016] revelaram que se é possível se incluir a inicialização de algoritmo determinístico como um recurso de entrada para outro sistema, é possível um aprendizado do comportamento desse algoritmo por uma rede neural bem treinada.

Ao contrário dos algoritmos com estruturas relativamente simples, que podem ser facilmente desdobradas, os de gerenciamento de recursos sem fio geralmente envolvem iterações computacionalmente complexas, com parâmetros multiobjetivos, operações como inversão de matrizes, SVD (do inglês *Singular Value Decomposition*) e/ou bissecção. Portanto, suas iterações não são passíveis de aproximação por uma única camada da rede neural [Samuel and Sharma 2017].

No geral, a aproximação de algoritmos tradicionais de gerenciamento de recursos em redes sem fio com DNN é um tópico emergente e pouco explorado [Nguyen et al. 2018]. Seu sucesso depende de questionamentos como: tipo do algoritmo de otimização tradicional utilizado, capacidade de aproximação por uma rede neural, quantidade de camadas/neurônios necessários para essa aproximação, avaliação de quão robusto

ou sensível é o método na prática, custo computacional da DNN para um sistema embarcado IoT, tamanho de armazenamento, entre outros.

No caso específico de redes *ad hoc slotted* Aloha lineares (i.e. redes cujos nós estão dispostos de maneira linear e equidistantes uns dos outros), não foram encontrados, até este momento, de conhecimento dos autores, trabalhos na literatura que empreguem o uso de *Deep Learning* para mapear pontos de operação dos parâmetros de interesse que objetivem melhorar o consumo de energia nestas redes.

3. Uma Proposta *Deep Learning* para Redução de Consumo de Energia

Este trabalho objetiva reduzir o consumo de energia em redes *ad hoc slotted* Aloha através da otimização da taxa de transmissão (R) de dados, do número de bits (N_b) por quadro (ou pacote), do número de saltos (i) da fonte até o destino e da potência ótima de transmissão (P_0). Para isso, faz-se uso de métodos de aprendizado como o *Deep Learning* a fim de diminuir o gasto energético dos modelos tradicionais. Assim, foi construído um modelo de aproximação baseado em aprendizado de máquina agregando a qualidade das soluções obtidas por funções matemáticas, a velocidade da convergência e a otimização multiobjetivo das heurísticas bioinspiradas.

O trabalho proposto compreende uma sequência de quatro etapas, conforme ilustrado na Figura 1. Primeiramente, deve ser definido um modelo de consumo energético que sirva como inicialização (isto é, base ou referência) do problema e possa ser aprendido por uma rede neural profunda. O segundo passo consiste em utilizar o modelo definido para gerar um conjunto de soluções otimizadas que sirvam como exemplos de treinamento para a rede neural. Depois, a DNN será treinada, através de ajustes recursivos nos pesos Θ existentes em cada neurônio, gerando assim um modelo capaz de aproximar essas soluções. As subseções 3.1, 3.2 e 3.3 exploram cada uma dessas fases de execução.

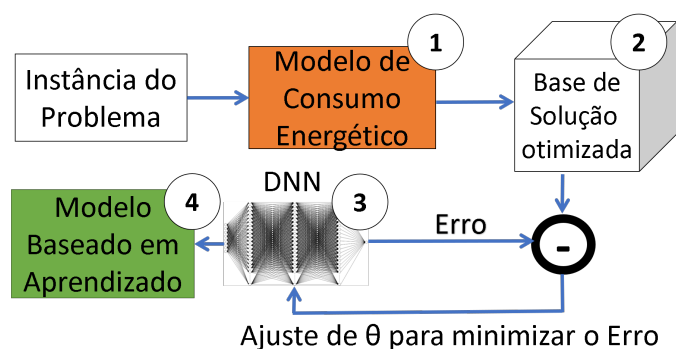


Figura 1. Etapas do método proposto: 1) Definição do modelo de consumo energético; 2) Conjunto de soluções otimizadas para formação do conjunto de treinamento e teste; 3) DNN utilizada para aproximação das soluções; 4) Modelo de consumo baseado em aprendizado de máquina.

3.1. Modelo de Consumo de Energia

Nesta fase se definiu um modelo de algoritmo de otimização clássica que seja capaz de otimizar simultaneamente parâmetros multiobjetivos conflitantes como potência e taxa de transmissão, tamanho do pacote de dados e a distância entre os nós para servir como base de aprendizado pela rede neural.

A realização desta fase do trabalho objetiva a modelagem do consumo de energia para redes *ad hoc slotted* Aloha lineares com saltos equidistantes, proposta por [Da Silva and De Moraes 2018]. Partindo do estudo de [Zhang and Gorce 2008], tem-se que a probabilidade de transmitir com sucesso um pacote contendo N_b bits é

$$pr_s(\gamma) = (1 - BER(\gamma))^{N_b}, \quad (1)$$

sendo $BER(\gamma)$ a taxa de erro de bit em função da relação sinal-ruído $\gamma = C_2 P_t \left(\frac{d}{i}\right)^{-\alpha} R^{-1}$, onde $C_2 = \frac{G_T G_R \lambda^2}{(4\pi)^2 N_0}$ (ver Tabela 1 e [Da Silva and De Moraes 2018]).

No entanto, como se deseja inserir o efeito do protocolo Aloha, e se partiu do fato de que a comunicação se dá através da realização de i saltos, os autores realizaram uma alteração na Eq. (1). Sabendo que o número de nós (n) neste cenário é uma relação direta de i , e portanto representado por $n = i + 1$, a probabilidade de uma estação transmitir com sucesso passou a ser representada por

$$Pr_s = pr_s(\gamma)\delta = \frac{(1 - BER(\gamma))^{N_b} \left(\frac{i}{i+1}\right)^{i-1}}{i+1}, \quad (2)$$

em que $\delta = \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-2}$ representa a probabilidade de apenas uma estação transmitir com sucesso no protocolo *slotted* Aloha. Ademais, a energia média total por bit, a partir da realização de i saltos entre fonte e destino, passou a ser dada, então, como sendo

$$\bar{E}_{ihop} = \bar{E}_{1hop} \cdot i = \frac{(E_c + K_1 P_t)}{(1 - BER(\gamma))^{N_b}} \cdot \frac{(i+1)^i}{i^{i-2}}, \quad (3)$$

em que, K_1 é a constante $\frac{\beta_{amp}}{R}$, em que R é a taxa de transmissão e β_{amp} uma constante de proporcionalidade, $E_c = \frac{2T_{start}P_{start}}{N_b} + \frac{P_{txElec} + P_{rxElec} + \alpha_{amp}}{R}$ e P_t é a potência de transmissão, sendo T_{start} , P_{start} , P_{txElec} , P_{rxElec} e α_{amp} parâmetros relacionados ao consumo do rádio dos dispositivos [Zhang and Gorce 2008].

Uma vez que o consumo energético em [Da Silva and De Moraes 2018] foi modelado tendo como variáveis o tamanho do pacote N_b e a taxa de transmissão R , foi realizada a derivação da Eq. (3) em função de P_t a fim de obter uma potência ótima de transmissão $P_0(R, N_b)$, em função dessas variáveis, que minimizasse o consumo energético em redes *ad hoc slotted* Aloha. Fazendo as devidas substituições, a energia média gasta por bit para se transmitir com sucesso, considerando a realização de i saltos equidistantes, passou a ser [Da Silva and De Moraes 2018]

$$\bar{E}_{ihop}(P_t = P_0(R, N_b)) = \frac{C_3 R + (C_4 + \beta_{amp} P_0(R, N_b)) N_b}{R N_b \left(1 - \frac{\alpha_m}{2\beta_m C_2 P_0(R, N_b) \left(\frac{d}{i}\right)^{-\alpha} R^{-1}}\right)^{N_b}} \cdot \frac{(i+1)^i}{i^{i-2}}, \quad (4)$$

em que $C_3 = 2T_{start}P_{start}$ e $C_4 = P_{txelec} + P_{rxelec} + \alpha_{amp}$.

Foi considerado ainda pelos autores a utilização da modulação BPSK (do inglês *Binary Phase Shift Keying*) e, por este motivo, na Eq. (4) as constantes α_m e β_m são, respectivamente, 1 e 2 [Da Silva and De Moraes 2018]. Neste contexto, o coeficiente de atenuação (α) assume dois valores distintos, 2 e 4, para representar o espaço livre e um cenário com maior perdas, respectivamente.

Como se busca realizar uma análise da modelagem proposta por [Da Silva and De Moraes 2018] por meio de uma DNN, neste trabalho foram usados os principais

parâmetros apresentados no trabalho dos autores, os quais são apresentados na Tabela 1. Estes parâmetros objetivam representar as características físicas do transceptor μ AMPS-1 [Karl and Willig 2007]. Vale ressaltar ainda que foi utilizada a representação em intervalos da taxa de transmissão, do tamanho do pacote e da distância entre fonte e destino para que se pudesse aumentar o espaço de busca durante o treino da DNN.

Tabela 1. Parâmetros para o modelo de consumo de energia [Karl and Willig 2007] e [Zhang and Gorce 2008]

Parâmetro	Descrição	Valor
P_{rxElec}	Potência do circuito de recepção	$279mW$
P_{start}	Potência para inicialização	$58,7mW$
T_{start}	Tempo de inicialização	$446\mu s$
P_{txElec}	Potência do circuito de transmissão	$151mW$
α_{amp}	Nível de potência	$174mW$
β_{amp}	Constante de proporcionalidade	5
N_0	Densidade espectral de ruído	$-154dBm/Hz$
f_c	Frequência da portadora	$2,4GHz$
c	Velocidade da luz	$3 \times 10^8 m/s$
G_T	Ganho da antena de transmissão	1
G_R	Ganho da antena de recepção	1
R	Taxa de transmissão	50 a $10^7 bps$
N_b	Número de bits por quadro	1 a 8×10^6
α	Coefficiente de atenuação	2 e 4
d	Distância entre fonte e destino	10 a $250m$
i	Número de saltos até o destino	1, 2, 4, 8

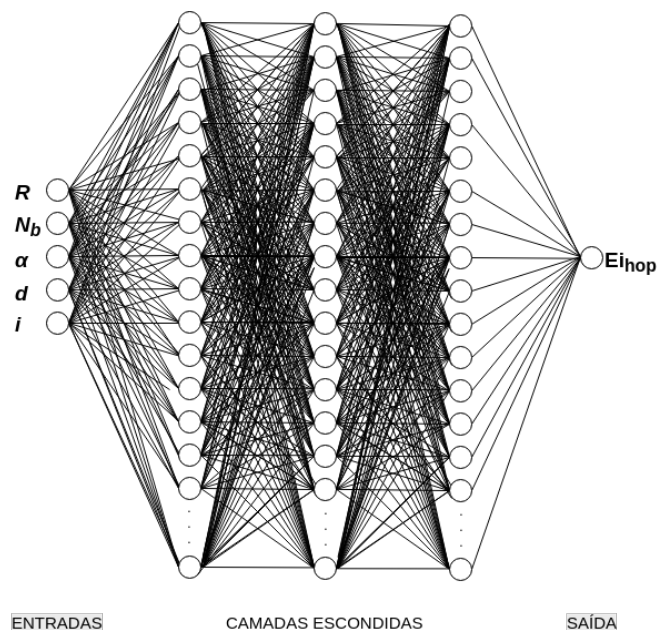


Figura 2. Arquitetura da rede neural profunda (DNN).

3.2. Formação do Conjunto de Treinamento

Esta fase consiste em criar uma base de dados para treinamento da DNN combinando todas variáveis envolvidas na Eq. (4), utilizando os parâmetros da Tabela 1, na tentativa de maximizar a capacidade de aprendizado da rede neural. Neste sentido, foi necessária a criação de um algoritmo de busca exaustiva, Algoritmo 1, capaz de realizar todas as combinações dos intervalos exibidos na Tabela 1 para as variáveis de entrada: R , N_b , α , d e i com saltos equidistantes, encontrando as soluções ótimas globais para a saída E_{ihop} . O mencionado algoritmo gerou um total de 38.506 amostras representativas para o problema que foram usadas para alimentar a rede neural, Figura 2, durante seu treinamento e testes.

Algorithm 1: Busca exaustiva das melhores configurações

Entrada: $\Delta_R, \Delta_{N_b}, \Delta_\alpha, \Delta_d, \Delta_i$ (faixas de variações de R, N_b, α, d e i)

Saída: *ConjuntoTreinamentoDNN*

início

$k = 0;$

para cada α em Δ_α **faça**

para cada d em Δ_d **faça**

para cada i em Δ_i **faça**

$j = 0;$

$E_{ihop} = MAX_{Valor};$

para cada R em Δ_R **faça**

$E_{ihop} = MAX_{Valor};$

para cada N_b em Δ_{N_b} **faça**

$TempE_{ihop} = Calcule_{E_{ihop}}(Eq.4);$

se $E_{ihop} > TempE_{ihop}$ **então**

$E_{ihop} = TempE_{ihop};$

$ConjuntoTreinamentoDNN[k, j] \leftarrow$

$E_{ihop}, R, N_b, \alpha, d, i;$

fim

fim

fim

$j = j + 1;$

fim

$k = k + 1;$

fim

fim

fim

3.3. Aproximação do Modelo de Consumo de Energia via DNN

Para aproximar a modelagem proposta por [Da Silva and De Moraes 2018] por meio de uma rede neural foi necessário identificar e salvar as melhores configurações da DNN para os hiper-parâmetros: quantidade de camadas escondidas da rede, número de épocas para o treinamento, taxa de aprendizado, algoritmo para a ativação dos neurônios, tamanho do *batch* entre outros [Huan et al. 2016], [Mohammadi et al. 2018].

De forma a elaborar essa análise, a construção de uma rede neural para aproximar funções algorítmicas iterativas necessita de algumas etapas analíticas. As etapas incluem:

a construção de redes neurais simples, consistindo em unidades de ativação e unidades binárias, aproximando assim operações de multiplicação e divisão; a composição dessas pequenas redes neurais para aproximar uma função racional representando uma iteração do algoritmo; a concatenação de tais funções racionais para aproximar todo o algoritmo e a limitação do erro propagado Θ da primeira iteração até a última [Sun et al. 2017].

Para a execução dessas tarefas, utilizou-se uma rede neural profunda do tipo MLP (do inglês *MultiLayer Perceptron*) com 5 neurônios de entradas, 3 camadas escondidas, em que cada camada possui 100 neurônios e um neurônio de saída. Para sua implementação foi utilizado o conjunto de bibliotecas scikit-learn para a linguagem Python.

O funcionamento adequado da MLP também depende das taxas e parâmetros utilizados para sua execução e aprendizado. Para este fim, o treinamento da MLP foi realizado em 1000 épocas. Esta quantidade de épocas foi suficiente para alcançar o erro quadrático médio (MSE, do inglês *Mean Squared Error*) estimado. A taxa de aprendizagem foi definida em 0,005.

Alguns dos problemas encontrados no treinamento de redes MLP são o conflito entre *underfitting*, a não convergência das soluções, e *overfitting*, que é o excesso de aprendizado ou ato de decorar os dados de treinamento. Tais fatos podem causar problemas durante o treinamento e na fase de inferência do modelo.

Para garantir que o modelo não sofresse com nenhum desses problemas, foi definido um fator de erro mínimo para a MLP não perder seu poder de generalização à dados não analisados durante o treinamento. Após vários testes empíricos, o valor mínimo de convergência do erro foi de 0,05% ou seja, acima dessa taxa o modelo deixava o aprendizado salutar e entrava em *overfitting*.

Outro parâmetro importante nessa etapa do processo de treinamento foram as funções de otimização e de ativação utilizadas. Para a otimização do fator de gradiente estocástico foi escolhido o algoritmo *Adam* por apresentar melhor ajuste nos parâmetros dos neurônios em relação as técnicas *Adagrad* e ao *RMSProp* [Kingma and Ba 2014].

Uma rede neural profunda possui maior capacidade de adquirir informações do que a rede neural tradicional. Portanto, sua capacidade de transformação dos dados que percorrem os neurônios e velocidade de aprendizado acompanham essa complexidade. Por este motivo, é de extrema importância determinar as transformações não-lineares que serão necessárias nos neurônios ocultos, construir uma estrutura de rede compatível com os dados a serem manipulados e analisar a capacidade de predição e a dependência dos valores iniciais [Tachibana and Otsuka 2018].

Por esse motivo, é essencial o uso de uma função de ativação capaz de garantir uma rápida convergência com um moderado custo computacional. Foi escolhida a função de ativação ReLU (do inglês *Rectified Linear Unit*) [Tachibana and Otsuka 2018], cuja principal vantagem é o controle de quais neurônios precisam ser ativados durante as iterações e baixa saturação em regiões não centrais de ativação. Além disso, esse tipo de função de ativação torna o aprendizado mais rápido que as funções de ativação tradicionais.

Para lidar com o tempo de treinamento e o custo computacional necessário no pro-

cesso de aprendizagem, a atualização dos neurônios em lotes paralelos é realizada a fim de melhorar a eficiência da DNN [Zheng et al. 2016]. Este trabalho utilizou a atualização em *mini-batch* com aprendizado paralelo de 200 neurônios por lote. Todas as definições dos hiper-parâmetros da MLP foram realizadas através de múltiplos testes baseados na análise empírica. As melhores configurações dentre os experimentos realizados estão resumidos na Tabela 2.

Tabela 2. Parâmetros para DNN-MLP

Descrição	Valor
Quantidade de camadas escondidas	3
Quantidade de neurônios em cada camada escondida	100
Número máximo de épocas para treinamento	1000
Taxa de aprendizado	0,005
Fator de erro mínimo	0,5
Algoritmo de otimização do gradiente estocástico	<i>Adam</i>
Função de ativação de neurônios	<i>ReLU</i>
Quantidade de batchs	200

4. Experimento

A rede neural proposta foi submetida a testes intensivos realizados em um notebook Intel CoreTM Dual Core de 2,4 GHz, com 4 GB de RAM, na tentativa de eliminar a criação de tendências nos resultados.

Para formação da base de dados de teste foi utilizado um conjunto composto por 7.701 amostras, exatamente 20% da base de treinamento, com valores de entradas não utilizadas durante o processo de treinamento. Estas entradas foram selecionadas dentro dos intervalos definidos na Tabela 1, conservando a distribuição individual dos parâmetros. A Figura 3 exibe a distribuição das amostras no conjunto de teste e no conjunto de treinamento, deixando claro que ambas seguem distribuições semelhantes.

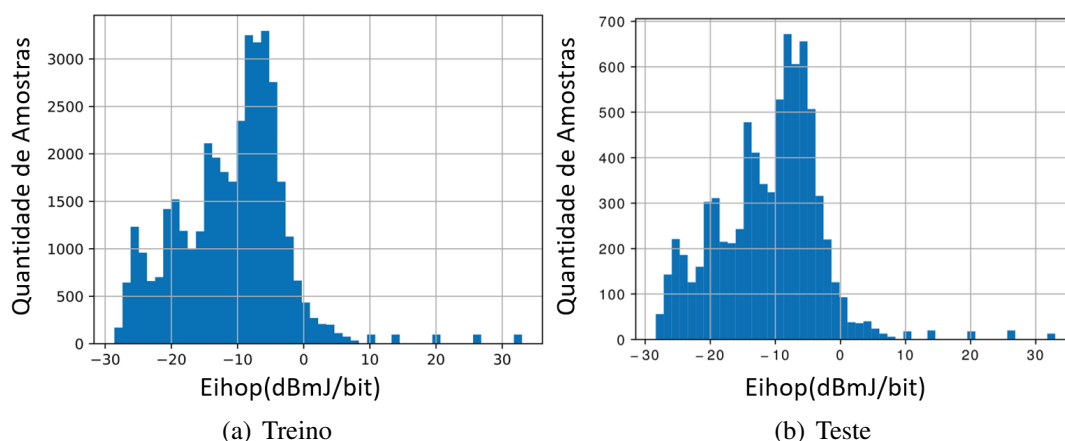


Figura 3. Distribuição da saída E_{ihop} nos conjuntos de treino e teste.

A utilização apenas dos parâmetros definidos nas Tabelas 1 e 2 não oferecem informações sobre o desempenho médio do modelo desenvolvido. Isso se deve ao fato de que grande parte das técnicas heurísticas, a exemplo da MLP, utilizarem de pesos

iniciados aleatoriamente, apresentando uma variação no desempenho ao final do processo de treinamento. Deste modo, passam a apresentar resultados que podem ser considerados *outliers* e a não retratarem o real desempenho médio do modelo.

Diante disso, a DNN proposta foi testada em diferentes cenários obtidos através de 30 testes com variações nos pesos iniciais dos perceptrons e no conjunto de amostras da base de testes verificando a escalabilidade e robustez real do modelo [Amin et al. 2017].

A fim de garantir a eficiência, os resultados foram apresentados como gráficos bidimensionais comparando a média dos resultados obtidos entre a saída real, Eq. (4), e a predita pelo modelo proposto. O resultado de [Da Silva and De Moraes 2018] foi utilizado como validação por se tratar de uma técnica que alcança as melhores soluções globais atualmente.

5. Resultados e Discussões

Para uma rede neural obter as soluções esperadas, mantendo seu poder de generalização, é preciso que se defina bem seus hiper-parâmetros. Variáveis como a quantidade ideal de épocas de treinamento, critério de parada do aprendizado e o método de comparação entre os estimadores durante a avaliação da evolução da rede definem o comportamento durante o treinamento do modelo, podendo gerar situações de *underfitting* ou *overfitting* [Luxhøj et al. 1997]. Por este motivo, foi utilizado como comparação de estimadores o MSE, que serviu como definidor da quantidade de épocas e de avaliação entre as técnicas durante as iterações.

A Figura 4 apresenta a evolução do MSE durante o processo de treinamento da DNN. Inicialmente, a rede apresenta uma queda acentuada, mantendo algumas variações ao decorrer das épocas de treinamento. Em todos os testes realizados a DNN convergiu ao MSE mínimo definido, precisando de aproximadamente 120 interações para alcançar este valor alvo. Em termos gerais, os resultados obtidos foram bastante satisfatórios, se comparados ao valor de início do MSE.

A Figura 5 apresenta um comparativo de resultados para a média de E_{ihop} encontrados pelos dois métodos, com a curva DNN apresentando os valores preditos para todo conjunto de testes e para a curva do modelo de energia [Da Silva and De Moraes 2018] que apresenta o resultado ótimo (Eq. (4)) encontrado para as mesmas entradas. A análise do gráfico demonstra que o MSE se manteve muito próximo do definido no treinamento, isto é 0,05, indicando convergência do método. Isso evidencia a capacidade da rede em aprender as relações não lineares do algoritmo comparado.

A Figura 6 ilustra os intervalos definidos no trabalho de [Da Silva and De Moraes 2018] para o coeficiente de atenuação com a distância $\alpha = 2$, distância entre fonte e destino $d = 80\text{m}$ e número de saltos $i = 1, 2, 4$ e 8 com distâncias entre os saltos 80, 40, 20 e 10 metros, respectivamente. As curvas praticamente sobrepostas comprovam a eficiência da rede neural, obtendo uma média de 0,05 e um desvio padrão de 0,54 para o MSE de comparação entre a DNN e o Modelo de Consumo Energético proposto em [Da Silva and De Moraes 2018]. Deste modo, pode-se assegurar que a DNN pode replicar eficientemente as relações de entrada e saída desse modelo.

A Tabela 3 analisa a ordem de complexidade assintótica do modelo para o pior caso utilizando a notação Big O, a média do custo computacional e o tempo de proces-

samento gasto na obtenção das 7.701 melhores soluções em ambos os métodos. Neste aspecto, os custos totais para a execução da DNN e o seu desempenho na obtenção dos resultados quando treinada, são separados nesta análise.

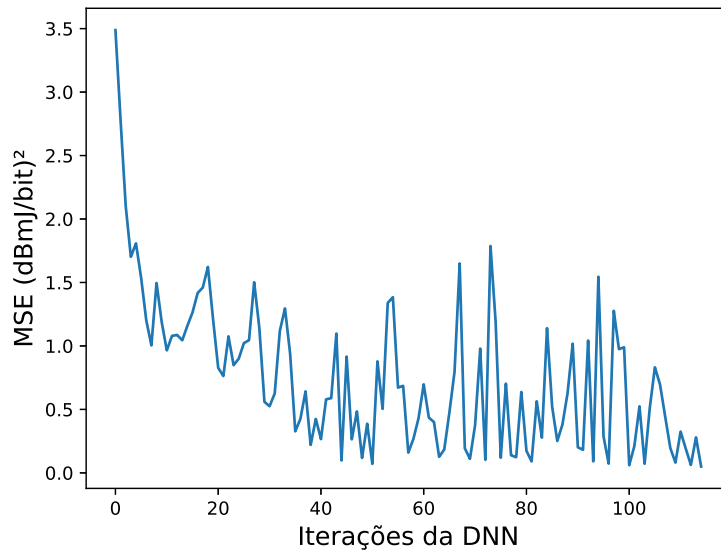


Figura 4. Evolução do Erro Quadrático Médio (MSE) da saída E_{ihop} durante o treinamento.

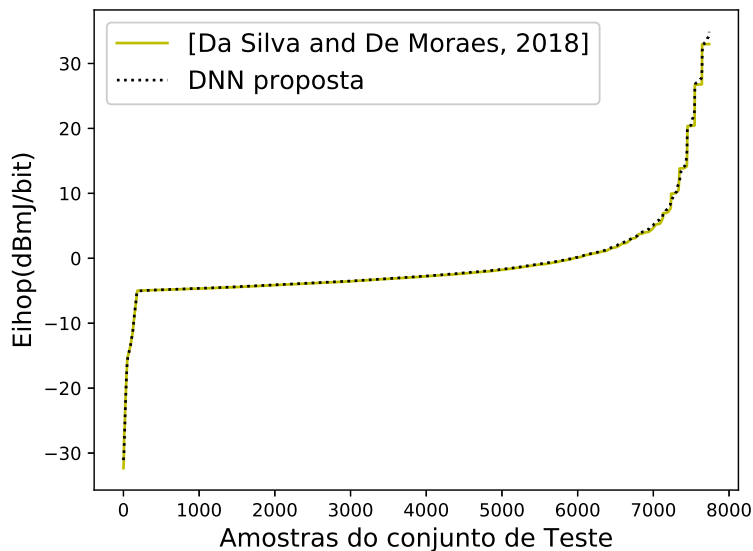


Figura 5. E_{ihop} obtido do modelo [Da Silva and De Moraes 2018] e da DNN para todo conjunto de teste.

Da Tabela 3, infere-se que uma rede neural além de produzir saídas passíveis de serem utilizadas em sistemas reais, realiza uma economia de 97,6% no uso da CPU e até 24.047,3s no tempo total de busca de soluções. Tal desempenho foi alcançado tendo em vista que uma DNN treinada se limita a utilizar operações simples como multiplicação de matrizes em seu funcionamento.

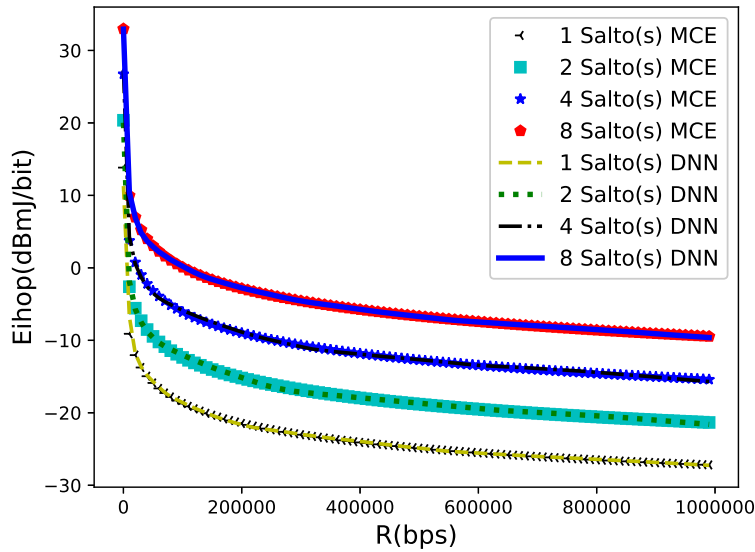


Figura 6. Comparação entre modelo de consumo energético (MCE) proposto por [Da Silva and De Moraes 2018] e DNN proposta neste trabalho, para Energia média por bit em função da taxa de transmissão (R) com tamanho do pacote N_b ótimo e $\alpha = 2$.

Tabela 3. Custo de processamento (em termos de uso de CPU e memória (MEM)) e tempo utilizado para convergência das técnicas

Técnica	Big O	Uso Médio	Tempo Total
DNN Treino e Teste	$O(n^4)$	346, 85% CPU; 2, 5% MEM	12, 94min
DNN Treinada	$O(n^2)$	2, 1% CPU; 0, 1% MEM	0, 678s
[Da Silva and De Moraes 2018]	$O(n^5)$	99, 7% CPU; 2, 2% MEM	6, 68h

Os códigos para reproduzir os experimentos apresentados neste artigo estão disponíveis em <https://github.com/pafil137/NN/tree/master/GColab-master/DProjectDEEP>.

6. Conclusões

A principal contribuição deste trabalho é um modelo baseado em aprendizagem profunda para o gerenciamento de redes *ad hoc* limitadas por recursos energéticos e de processamento, a exemplo das redes IoT. Deste modo, se utilizou o aprendizado de máquina e a otimização multi-objetiva de recursos para eficiência energética aplicado em redes reais. Ao contrário dos trabalhos existentes sobre aproximação de algoritmos de otimização, como aqueles que usam desdobramentos, a abordagem aqui empregada é comparada a um modelo teórico rigoroso.

Nos Experimentos numéricos extensivos, foram considerados dados simulados com parâmetros reais de redes *ad hoc slotted* Aloha. Os resultados apresentados neste artigo, mostram que o uso de uma DNN treinada permite alcançar resultados próximos dos ótimos com custo de processamento 97,6% menor. Essa vantagem também é observada na redução do tempo para obtenção das soluções alcançando uma economia na ordem de 24.047,3s. A partir desses resultados é possível inferir que o uso de técnicas

de *Deep Learning*, pode servir como base para criação de novos protocolos inteligentes para redes IoT, reduzindo o consumo energético de maneira mais eficiente que as técnicas tradicionais utilizadas atualmente.

Como trabalhos futuros, se pretende implementar a DNN proposta em kits práticos de IoT e avaliar seu desempenho e compatibilidade empregados à sistemas reais de redes de sensores. Para isso, pretendemos usar bibliotecas específicas como PyTorch QNN-PACK ou TensorFlow Lite que contêm otimizadores que permitem comprimir o tamanho das redes neurais profundas, deixando-as minimalistas para a implantação em dispositivos com recursos de CPU e memória limitados.

Outro desmembramento do trabalho pretende desenvolver a DNN para outros protocolos de controle de acesso ao meio mais sofisticados, a exemplo do CSMA/CA (do inglês *Carrier Sense Multiple Access with Collision Avoidance*), aplicando-o em cenários cujos nós se dispõem de maneira aleatória na rede a fim de reproduzir topologias que mais se aproxime de casos práticos.

Agradecimentos

Este trabalho foi financiado em parte pela Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

- Amin, H. H., Deabes, W., and Bouazza, K. (Jul. 2017). Hybrid spiking neural model for clustering smart environment activities. In *15th International Conference on Industrial Informatics (INDIN)*, Emden, Germany.
- Baligh, H., Hong, M., Liao, W.-C., Luo, Z.-Q., Razaviyayn, M., Sanjabi, M., and Sun, R. (2014). Cross layer provision of future cellular networks. *arXiv preprint arXiv:1407.1424*.
- Da Silva, B. A. and De Moraes, R. M. (Sep. 2018). Um estudo sobre o consumo de energia em redes ad hoc lineares aloha com saltos equidistantes. In *Anais do XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - Disponível em: https://www.dropbox.com/s/89k2t7swo064afk/Anais_SBrT_2018_v2.pdf?dl=0*, Campina Grande, PB.
- Fadlullah, Z., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., and Mizutani, K. (2017). State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455.
- Huan, Y., Qin, Y., You, Y., Zheng, L., and Zou, Z. (Sep. 2016). A multiplication reduction technique with near-zero approximation for embedded learning in iot devices. In *System-on-Chip Conference (SOCC), 2016 29th IEEE International*, Seattle, USA.
- Kapoor, C., Singh, H., and Laxmi, V. (Apr. 2018). A survey on energy efficient routing for delay minimization in iot networks. In *International Conference on Intelligent Circuits and Systems (ICICS)*, Phagwara, India.

- Karl, H. and Willig, A. (2007). *Protocols and architectures for wireless sensor networks*. John Wiley & Sons.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Luxhøj, J. T., Williams, T. P., and Shyur, H.-J. (1997). Comparison of regression and neural network models for prediction of inspection profiles for aging aircraft. *IIE transactions*, 29(2):91–101.
- Mamdouh, M., Elrukhsi, M. A., and Khattab, A. (Aug. 2018). Securing the internet of things and wireless sensor networks via machine learning: A survey. In *International Conference on Computer and Applications (ICCA)*, Beirut, Lebanon.
- Mao, Q., Hu, F., and Hao, Q. (2018). Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 20(4):2595–2621.
- Mohammadi, M., Al-Fuqaha, A., Sorour, S., and Guizani, M. (2018). Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys Tutorials*, 20(4):2923–2960.
- Nguyen, V.-T., Luong, T.-K., Le Duc, H., and Hoang, V.-P. (Sep. 2018). An efficient hardware implementation of activation functions using stochastic computing for deep neural networks. In *12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*, Hanoi, Vietnam.
- Samuel, A. and Sharma, D. K. (Jul. 2017). Summary generation using geo-coordinates and temporal data in microblogging environment. In *8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, India.
- Shi, Q., Razaviyayn, M., Luo, Z.-Q., and He, C. (May. 2011). An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic.
- Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., and Sidiropoulos, N. D. (Jul. 2017). Learning to optimize: Training deep neural networks for wireless resource management. In *18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan.
- Tachibana, K. and Otsuka, K. (Sep. 2018). Wind prediction performance of complex neural network with relu activation function. In *57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Nara, Japan.
- Zhang, R. and Gorce, J.-M. (Mar. 2008). Optimal transmission range for minimum energy consumption in wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, USA.
- Zheng, J., Ma, Q., and Zhou, W. (Oct. 2016). Performance comparison of full-batch bp and mini-batch bp algorithm on spark framework. In *8th International Conference on Wireless Communications & Signal Processing (WCSP)*, Yangzhou, China.