Algoritmo Híbrido de Otimização por Enxame de Partículas para o Aprendizado Federado de Redes Neurais Artificiais

Eliezer T. S. Sanhá * **Fabricio Javier Erazo-Costa** *,** **Frederico Gadelha Guimarães** ***

* *Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais (UFMG), Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, Brasil, (e-mail: eliezer-sanha@ufmg.br)*
** *Departmento de Engenharia Elétrica, Universidade Federal de Ouro Preto (UFOP), (e-mail: fabricioerazo@ufop.edu.br)*
*** *Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG), (e-mail: fredericoguimaraes@ufmg.br)*

Abstract: The aggregation algorithms of Federated Learning (FL) often overlook the balance between performance and communication cost. Therefore, in this study, we propose a new federated training method called FLPSO-SGD, based on the hybrid Particle Swarm Optimization-Stochastic Gradient Descent (PSO-SGD) algorithm, aiming to address these aspects. In contrast to classical FL training techniques, our method collects errors from the clients to the server, instead of model parameters, while the PSO-SGD algorithm conducts training on the client side. The algorithms were evaluated on classification problems using the UC Irvine datasets for the PSO-SGD, and the CIFAR-10 dataset for the FLPSO-SGD. The results highlight the promising performance of the PSO-SGD algorithm. Furthermore, the FLPSO-SGD algorithm demonstrated superior accuracies in global training compared to the FedAvg and FedPSO techniques. The results suggest that FLPSO-SGD is an effective alternative for FL training, particularly in applications with restricted bandwidth communication networks on clients.

Resumo: Os algoritmos de gerenciamento do treinamento de Aprendizado Federado (FL) comumente desconsideram o equilíbrio entre desempenho e custo de comunicação. Portanto, neste estudo propomos um novo método de treinamento federado denominado FLPSO-SGD que é baseado no algoritmo híbrido de otimização por enxame de partículas-Gradiente Descendente Estocástico (PSO-SGD) visando considerar esses aspectos. Em contraste com as técnicas clássicas de treinamento de FL, nosso método recebe os erros dos clientes em vez dos parâmetros dos modelos, enquanto o algoritmo PSO-SGD realiza o treinamento no cliente. Os algoritmos foram avaliados em problemas de classificação utilizando as bases de dados da UC Irvine para o PSO-SGD, e a base de dados CIFAR-10 para o FLPSO-SGD. Os resultados destacam o desempenho promissor do algoritmo PSO-SGD. Além disso, a metodologia federada FLPSO-SGD demonstrou acurácias superiores no treinamento global em comparação com as técnicas FedAvg e FedPSO. Os resultados sugerem que FLPSO-SGD é uma alternativa eficaz para treinamentos FL, especialmente em aplicações com redes de comunicação com largura de banda restrita nos clientes.

Keywords: Machine Learning; Neural Networks; Convolutional Neural Networks; Federated Learning; Particle Swarm Optimization.

Palavras-chaves: Aprendizado de Máquina; Redes Neurais Artificiais; Redes Neurais Convolucionais; Aprendizado Federado; Otimização por Enxame de Partículas.

1. INTRODUÇÃO

Os avanços tecnológicos têm possibilitado que uma variedade de dispositivos registrem dados em diversos contextos, desde ambientes industriais e veiculares, até setores financeiros, hospitalares, residências, incluindo diversos dispositivos de borda. Nesse cenário, a utilização de ferramentas como a Aprendizagem de Máquina (ML, do inglês *Machine Learning*) torna-se indispensável para a manipulação, processamento e extração de informações úteis a partir de uma grande quantidade de dados, muitas vezes complexos (Alpaydin, 2020). Essas técnicas impulsionam a tomada de decisão, realizando tarefas como classificação, regressão, previsão, entre outras.

⋆ O presente trabalho foi realizado com o apoio financeiro das agências de fomento CAPES, FAPEMIG, CNPq, processo no. 312991/2020-7; e da Fundação de Desenvolvimento da Pesquisa (Fundep) por meio de Projeto de PD&I "Programa de Aprendizado de Máquina Federado para Veículos Conectados".

O treinamento convencional de ML com vários dispositivos envolve a transmissão e coleta centralizada de dados em um ponto específico, onde um servidor utiliza estas informações para calcular um modelo global. Apesar da eficácia deste método, ele apresenta desafios em termos de privacidade de dados, segurança de dados e eficiência computacional. Para resolver estes problemas, Konečnỳ et al. (2016) apresentaram o conceito de Aprendizado Federado (FL, do inglês *Federated Learning*), uma nova configuração de ML que permite que o treinamento seja colaborativo e com dados distribuídos. Neste paradigma, os clientes utilizam exclusivamente dados existentes no próprio dispositivo para calcular modelos localmente. Ao contrário do treinamento convencional, apenas os parâmetros dos modelos locais são transmitidos ao servidor para gerenciar o treinamento global utilizando as informações enviadas por cada cliente. Este processo é realizado por um algoritmo de treinamento em FL para encontrar um modelo global (técnicas de agregação). Nesta abordagem, os clientes mantêm a privacidade dos dados, porque estes não são mais transmitidos e são processados localmente. Apesar dos benefícios de FL, ainda existem desafios a serem superados, como o custo de comunicação (transmissão de uma grande quantidade de parâmetros), a latência no treinamento e sobrecarga da rede (Hamer et al., 2020).

Vários algoritmos de treinamento de FL têm sido propostos com o objetivo de encontrar um modelo global a partir das informações enviadas pelos clientes no treinamento local. O algoritmo Média Federada (FedAvg, do inglês *Federated Averaging*) foi a primeira técnica a ser desenvolvida visando o cálculo do modelo global pela média ponderada dos parâmetros enviados pelos dispositivos. Outros métodos foram propostos a partir de FedAvg, como o FedProx que permite trabalhar com dados independentes e identicamente distribuídos (Li et al., 2020) e SCAFFOLD que aprimora os algoritmos anteriores acelerando a convergência e diminuindo o desvio do cliente (resultados globais ineficientes gerados devido ao erro ou inadequação do treinamento local de um ou mais clientes) (Karimireddy et al., 2020). Estas técnicas exigem o envio dos parâmetros dos modelos treinados localmente para o servidor, e apesar de amplamente utilizadas, estas técnicas possuem altos custo de comunicação, uma limitação principalmente em ambientes de redes de comunicação com largura de banda restrita entre clientes e servidor (Hamer et al., 2020). Para tratar este problema, Park et al. (2021) apresentaram FedPSO, uma nova abordagem de treinamento para FL. Este método emprega o algoritmo meta-heurístico Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*), no qual os clientes enviam as métricas de acurácias ou erros para o servidor, em lugar dos parâmetros. O servidor identifica o cliente com melhor desempenho e solicita exclusivamente a este dispositivo seus parâmetros adquiridos no treinamento local usando Gradiente Descendente Estocástico (SGD, do inglês *Stochastic Gradient Descent*). Esses parâmetros são considerados como globais e, em seguida, transmitidos para atualizar os parâmetros dos clientes.

O algoritmo de otimização SGD e suas variantes são amplamente adotadas na otimização de parâmetros de redes neurais artificiais (RNAs) devido à sua eficiência e escalabilidade. Esta característica os torna a escolha predominante para abordagens de treinamento local em FL. No entanto, o SGD pode encontrar dificuldades de convergência em mínimos locais, especialmente em funções não diferenciáveis. Em um contexto federado, isso pode levar ao surgimento do desvio de cliente e diminuir o desempenho dos modelos globais. Apesar de FedPSO oferecer uma solução na redução no custo de comunicação para aplicações de FL, esta metodologia utiliza PSO exclusivamente para gerenciar o treinamento federado global, enquanto o treinamento local está baseado (da mesma forma que FedAvg) no SGD. Para superar estas limitações, este trabalho propõe uma nova metodologia de FL denominada FLPSO-SGD que visa melhorar tanto o desempenho do algoritmo federado quanto considerar a redução do custo de comunicação. Com foco em ampliar tanto a eficiência quanto a versatilidade do processo de otimização no treinamento local nos clientes, um novo algoritmo híbrido denominado PSO-SGD é introduzido, onde SGD se torna uma componente da velocidade da partícula no PSO, enquanto o treinamento de FL utiliza o paradigma de envio das métricas (erro ou acurácia).

O método proposto foi avaliado em duas fases e comparado com técnicas do estado da arte. Inicialmente, o algoritmo híbrido PSO-SGD foi validado em uma base de dados para classificação utilizando RNAs. Em seguida, a metodologia federada FLPSO-SGD foi testada em uma base de dados para classificação de visão computacional utilizando redes neurais convolucionais. Os resultados demonstram um desempenho promissor do algoritmo PSO-SGD e da metodologia federada. Esta abordagem oferece uma nova perspectiva no treinamento federado, proporcionando melhorias significativas no desempenho e redução do custo de comunicação.

O restante do artigo está organizado da seguinte forma: A Seção 2 apresenta a fundamentação teórica e trabalhos relacionados. A Seção 3 descreve os métodos propostos neste trabalho. Os experimentos computacionais são abordados na Seção 4, enquanto a Seção 5 apresenta e discute os resultados obtidos nos experimentos.

2. FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

2.1 Otimização por Enxame de Partículas

No algoritmo meta-heurístico PSO (Kennedy and Eberhart, 1995), cada partícula representa uma solução candidata que é iterativamente ajustada no espaço de busca em direção aos valores ótimos. Esses ajustes são influenciados pelo que é conhecido como comportamento cognitivo, que corresponde à melhor posição encontrada pela própria partícula, e pelo comportamento social, que é determinado pela melhor posição encontrada por qualquer partícula em sua vizinhança. A atualização da velocidade da partícula é governada pela Equação (1), enquanto a atualização de sua posição no espaço de busca é regida pela Equação (2).

$$v(t+1) = wv(t) + c_1 r_1(t)(pbest(t) - x(t)) + c_2 r_2(t)(gbest(t) - x(t)) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

Onde, w é coeficiente da inércia, v representa a velocidade da partícula, r_1 e r_2 são números aleatórios entre [0, 1], c_1 e c_2 são coeficientes de aceleração cognitivo e social, respectivamente. A variável x é a posição da partícula, *pbest* é a melhor posição encontrada pela partícula e *gbest* é a melhor posição encontrada entre as partículas.

2.2 Aprendizado Federado

O Aprendizado Federado (FL, do inglês *Federated Learning*) é uma abordagem no campo do Aprendizado de Máquina que está revolucionando a forma como os modelos são treinados. Sua premissa central é permitir que múltiplos participantes, como dispositivos ou entidades (também conhecidos como clientes), colaborem no treinamento de modelos globais (gerenciados por um Servidor) de forma independente, eliminando a necessidade de compartilhar dados sensíveis localmente (McMahan et al., 2017).

O paradigma FL está composto essencialmente por dois elementos principais: O Cliente e o Servidor. O Cliente possui dados locais não compartilhados e realiza treinamento localmente, enquanto o Servidor coordena o treinamento global utilizando informações (parâmetros) provenientes dos Clientes. Além disso, o Servidor atualiza os Clientes com parâmetros globais calculados, formando uma rede de aprendizado distribuído e colaborativo. Por essa razão, outros elementos, como algoritmos de agregação de modelos para FL e protocolos de comunicação eficientes, também são considerados importantes (Li et al., 2021).

A técnica média federada (FedAvg, do inglês *Federated Averaging*) é um algoritmo de agregação de modelos para FL com uma arquitetura centralizada (inicialmente baseada no SGD), que permite trabalhar com dados distribuídos. Este algoritmo estabeleceu de forma simplificada o treinamento das RNAs em sistemas de FL (McMahan et al., 2017).

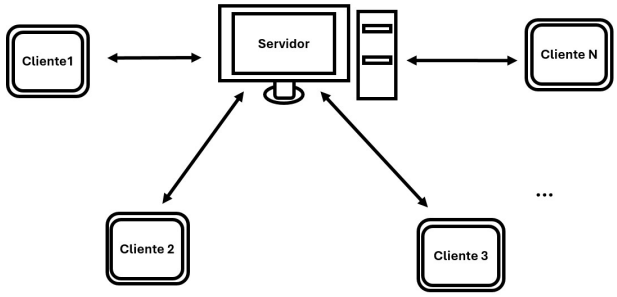

Figura 1. Configuração básica do treinamento com FedAvg em FL.

A Figura 1 ilustra a dinâmica básica de FedAvg. Em cada rodada, o Servidor manda os parâmetros globais para o grupo S_t de clientes selecionados. Enseguida, os clientes utilizam a equação (3) para atualizar o parâmetros globais com base em seus dados locais. Os parâmetros atualizados são enviados de volta ao Servidor, que utiliza a equação (4) para calcular a média ponderada das contribuições de todos os parâmetros locais recebidos, gerando assim novos parâmetros globais (modelo).

$$x_{ki} = x_i - \eta \nabla L(x_{Dki}; b) \tag{3}$$

$$x_i = \sum_{k=1}^{S_t} \frac{D_k}{D_{S_t}} x_{ki} \tag{4}$$

Onde, k é o k-ésimo Cliente na i-ésima iteração, x são os parâmetros ou modelos (neste caso pesos e vieses), D_k são os dados do k-ésimo cliente, e D_{S_t} são os dados de todos os clientes selecionados pelo Servidor.

2.3 Trabalhos Relacionados

O FedPSO (Park et al., 2021), tem como foco principal diminuir a quantidade de parâmetros transmitidos do cliente para o servidor, o que é alcançado por meio de uma avaliação dos valores dos erros ou acurácias obtidos nos treinamentos locais dos clientes. Apenas os parâmetros do cliente com melhor acurácia ou menor erro são enviados para o servidor, que por sua vez, redistribui esses parâmetros como atualização da rodada seguinte para os clientes. Embora essa abordagem reduza a quantidade de parâmetros totais que são enviados para o servidor, o que diminui custo de comunicação comparado ao FedAvg, não proporciona ganhos significativos de desempenho, uma vez que os clientes realizam treinamentos locais usando SGD da mesma forma que FedAvg. A Tabela 1 proporciona o resumo das abordagens para os algoritmos FedAvg e FedPSO.

Tabela 1. Abordagens de algoritmos de otimização utilizando Aprendizado Federado

Autor(es) (ano)	Algoritmo(s)	Síntese de abordagens
McMahan et al. (2017)	FedAvg	As atualizações são feitas mediante uma média ponderada.
Park et al. (2021)	FedPSO	Compara os *pbest* dos clientes, e estabelece $w(i+1)$ do cliente selecionado como atualização do SGD.

3. MÉTODOS PROPOSTOS

Esta seção apresenta o método colaborativo proposto para otimização de parâmetros das RNAs com dados distribuídos. Para um melhor entendimento do método de Aprendizado Federado proposto, que é baseado em Otimização por Enxame de Partículas hibridizado com o gradiente Descendente Estocástico (FLPSO-SGD), será dividido em duas partes, primeiro mostrando o funcionamento da estratégia do servidor e posteriormente a estratégia de treinamento dos clientes que utiliza a abordagem híbrida sugerida (PSO-SGD). A Figura 2 mostra N dispositivos (clientes), cada um com seu próprio treinamento local e com conjunto de dados diferentes que tem comunicação com um Servidor central.

3.1 Servidor

O processo de gerenciamento do treinamento FL do método proposto se encontra resumido no Algoritmo 1. Para

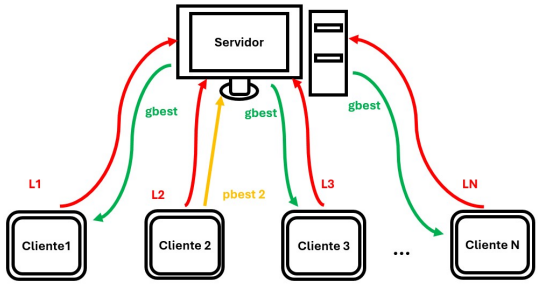

Figura 2. FLPSO-SGD. Os clientes enviam os erros L_k (setas vermelhas) e exclusivamente o de melhor desempenho envia os parâmetros (seta amarela). O Servidor atualiza esses parâmetros para os clientes (setas verdes).

cada rodada de comunicação definida (linha 1), o servidor recebe os erros L_k enviados pelos N clientes (setas vermelhas na Figura 2) e avalia, qual é o cliente com menor erro (linha 2) utilizando as equações 5 e 6 .

$$L_{\min} = \min(L_1, L_2, \ldots, L_N) \tag{5}$$

$$k^* = \arg\min_k L_k \tag{6}$$

Posteriormente, o servidor solicita o *pbest* (seta amarela na Figura 2) do cliente com o melhor valor *fitness* (menor erro), e em seguida atribui estes valores como *gbest* (linha 3). Finalmente, um subconjunto aleatório S_t de clientes é selecionado para receber *gbest* do servidor (setas verdes na Figura 2), e atualizar os seus parâmetros aplicando a função de atualização do cliente (linha 5-7). Este processo ocorre paralelamente para todos os clientes pertencentes a subconjunto S_t. A atualização do cliente (linha 6) se refere a aplicação do algoritmo híbrido utilizado para treinamento local (pseudocódigos do Algoritmo 2) que será explicado na próxima subseção.

Após o servidor fornecer o *gbest* para o grupo de clientes selecionados (S_t), inicia-se o treinamento local de cada cliente. Durante esse processo, novos conjuntos de parâmetros são iterativamente atualizados e refinados. Em seguida, os erros obtidos são enviados de volta ao servidor. Esse ciclo de atualização e comunicação é repetido até que o número predeterminado de rodadas seja alcançado.

Algorithm 1 FLPSO-SGD

1: **Para** cada rodada $t = 1, 2, 3...$ **faça**
2: $L_{min} \longleftarrow \min(L_1, L_2, \ldots, L_N)$
3: Solicita o *pbest* do Cliente k^*
4: servidor escolhe subconjunto S_t de clientes
5: **para** cada cliente $k \in S_t$ em paralelo
6: $L_k \longleftarrow$ Atualização_Cliente(k, g_{best})
7: **Fim Para**
8: **Fim Para**

3.2 Cliente

Para o treinamento local, o algoritmo PSO-SGD é proposto utilizando a hibridização dos algoritmos PSO e SGD. Para este objetivo, a taxa de aprendizado (η) associada ao gradiente descendente da função de custo ∇L é agregada como uma componente da velocidade da partícula (cliente) de PSO, conforme descrito na equação (7).

$$\begin{aligned} v_{i+1} = wv_i + c_1 r_1 (pbest_i - x_i) \\ + c_2 r_2 (gbest - x_i) - \eta \nabla L(x_{Dki}; b) \end{aligned} \tag{7}$$

Onde x representa uma configuração dos parâmetros (pesos e bias) da RNA, que se atualiza minimizando a função do erro L, *pbest* são os parâmetros ótimos pessoais do cliente e *gbest* parâmetros ótimos globais (geralmente fornecidos pelo servidor). O índice k representa $k-$ésimo cliente na sua $i-$ésima época de treinamento, com η como taxa de aprendizado e $\nabla L(x_{Dki}; b)$ representando o gradiente da função de custo correspondente ao mini-lote b relativos aos dados Dki do cliente.

A Figura 3 ilustra a dinâmica do algoritmo híbrido proposto para o treinamento local dos clientes, destacando a modificação na equação de atualização de velocidade do PSO. Nesta figura, a soma vetorial v_{i+1}, representada pelo vetor azul, corresponde a uma atualização da velocidade utilizando o algoritmo do PSO original. A modificação proposta por este trabalho consiste na agregação da componente do SGD com a taxa de aprendizado, resultando na atualização da velocidade v_{i+1}, representado pelo vetor preto. Em seguida, os parâmetros são atualizados para $x(i+1)$, somando x_i e v_{i+1}. É importante destacar que o *gbest*, uma componente do vetor v_{i+1}, é fornecido pelo servidor aos clientes selecionados aleatoriamente (S_t).

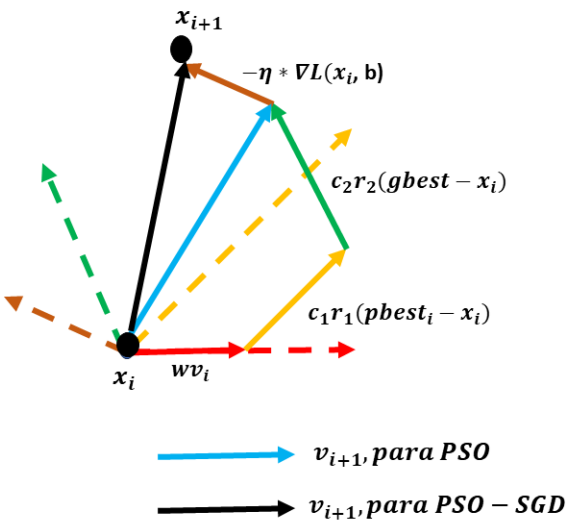

Figura 3. Representação vetorial do algoritmo híbrido PSO-SGD proposto para a otimização dos parâmetros locais de RNAs.

O Algoritmo 2 descreve o processo proposto para o treinamento local dos clientes. Primeiramente, os dados do cliente são divididos em lotes pequenos e os parâmetros do cliente k são inicializados. Se o *gbest* inicial do cliente for

um vetor nulo (sem fornecimento inicial do *gbest* pelo servidor) os parâmetros de x_k são inicializados aleatoriamente e a velocidade inicial da partícula (cliente) é zero, o que faz com que só o último termo (SGD) da equação (7) inicializa a primeira época do treinamento do cliente. Os valores de erros são comparados, caso a condição de atualização (linha 13) seja cumprida o *pbest* é atualizado (linha 14), este processo é repetido de acordo com a quantidade de épocas definidas para o treinamento local e quantidades de mini-lotes que o cliente possui. Após atingir o número de épocas de treinamento especificadas o último erro obtido neste treinamento local é enviado ao servidor (linha 18), para ser avaliado junto aos erros de outros clientes.

Algorithm 2 Função de Atualização do Cliente

1: **Função** Atualização_Cliente $(k, gbest)$
2: Divisão dos dados do cliente em B lotes
3: **Se** $gbest = 0$, **Então**
4: inicializa x_k aleatoriamente
5: $v_k \longleftarrow$ vetor nulo
6: $p_{\text{best}} \longleftarrow x_k$
7: $g_{\text{best}} \longleftarrow x_k$
8: **Fim Se**
9: **Para** cada época do cliente $i = 1, 2, ...$ **faça**
10: **Para** cada lote $b \in B$ **faça**
11: execução da equação (7)
12: $x_{i+1} \longleftarrow x_i + v_{i+1}$
13: **Se** $L(D_k, x_{i+1}) < L(pbest)$, **Então**
14: Atualiza $pbest \longleftarrow x_{i+1}$
15: **Fim Se**
16: **Fim Para**
17: **Fim Para**
18: **retorna** $L(D_k, x_{i+1})$
19: **Fim da Função**

4. EXPERIMENTOS COMPUTACIONAIS

Os experimentos foram conduzidos em duas fases distintas. Na primeira fase, a validação do método proposto PSO-SGD é realizada utilizando uma base de dados para classificação por meio de RNAs. Na segunda fase, a metodologia federada FLPSO-SGD é testada em problemas de classificação de visão computacional, empregando Redes Neurais Convolucionais.

4.1 Base de dados

Para a primeira fase do experimento (validação de PSO-SGD) foram utilizados 7 diferentes conjuntos de dados *(datasets)*, correspondentes aos problemas clássicos de classificação, que foram obtidos através do Repositório para Aprendizado de Máquina da *UC Irvine*. A Tabela 2 mostra os detalhes principais dos conjuntos de dados (número de instâncias, número de atributos e número de classes).

A segunda fase dos experimentos (teste de FLPSO-SGD) utilizou a base de dados de Visão Computacional CIFAR-10 (*Canadian Institute for Advanced Research, 10 classes*), que pertence ao conjunto de dados conhecidos como *"Tiny Images"*, ou seja, a uma grande coleção de imagens variadas, de baixa resolução que frequentemente são usadas em problemas de Aprendizado de Máquina, sobretudo, no

Tabela 2. Detalhes de conjunto de dados usados na 1ª fase dos experimentos

Conjunto de dados	Instâncias	Atributos	Classes
Iris	150	4	3
Breast Cancer	569	31	2
Glass	214	9	6
Banknote	1372	5	2
Ionosphere	351	34	2
Sonar	208	60	2
Wine	178	13	3

campo de Visão Computacional. CIFAR-10 possui 60.000 amostras de imagens coloridas de 32x32 pixels, que são: avião, automóvel , pássaro, gato, veado, cachorro, sapo, cavalo, navio e caminhão. Existem 6000 imagens por classe, com 5000 imagens para o treinamento e 1000 imagens por classe para testar o modelo. A Figura 4 mostra algumas imagens aleatórias do conjunto de dados CIFAR-10.

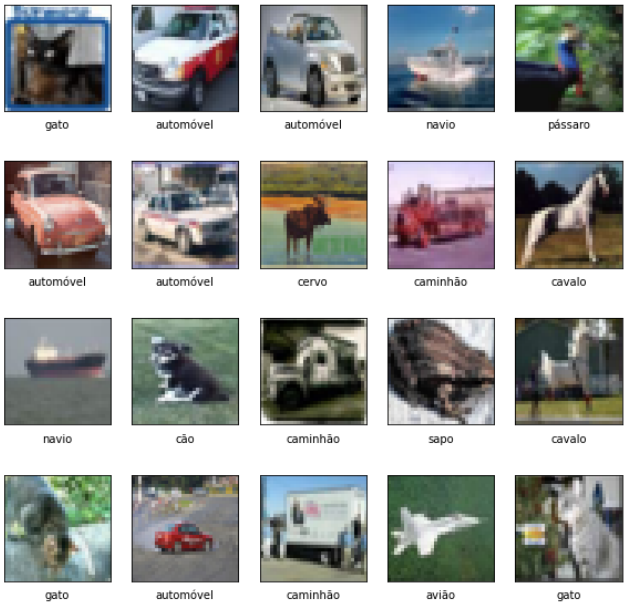

Figura 4. Conjunto de imagens aleatórias do CIFAR-10.

4.2 Configuração dos Experimentos

Para a primeira fase dos experimentos, configurou-se uma arquitetura de RNA com uma camada oculta contendo 20 neurônios, as camadas de entrada e saída foram configuradas de acordo com os atributos e classe dos problemas, respectivamente. Na camada oculta foram utilizadas ReLU como a função de ativação, na camada de saída foi usada sigmoide como classificador para os problemas de classificação binária, e para problemas de classificação multiclasse utilizou-se Softmax como classificador, usando a Entropia Cruzada como função de custo. Os hiper-parâmetros pertinentes foram configurados com os seguintes valores: A taxa de aprendizagem foi de 0,01 para todos os conjuntos de dados menos para *Glass* que foi de 0,001, essa mudança no valor da taxa de aprendizado, se deve a complexidade do problema *Glass* (problema com maior número de classes na primeira fase dos experimentos) e devido a eficiência experimental empírica, foi adotado o valor de 0,001 para o referido problema.

A população inicial de PSO para todos os conjuntos de dados foi de 25 partículas, com os parâmetros $c_1 = 0,8$, $c_2 = 0,5$, $E = 0,9$ e com 1000 iterações como critério de parada. Para todos os conjuntos de dados utilizados nesta fase experimental, utilizou-se 20% de dados para teste, os erros de validação foram obtidos aplicando a validação cruzada com 5-folds e 1000 iterações, isto é, o conjunto de treinamento foi subdividido em 5 partes, cada iteração usa 4 partes para o treinamento e 1 parte para validação, e depois das 1000 iterações, o erro de validação é obtido somando-se os erros obtidos em cada iteração, e então, divide-se o resultado pelo número total de iterações. Nesta primeira fase, os experimentos foram realizados em um computador com 32 GB de RAM instalada, e processador Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz.

Na segunda fase dos experimentos, onde se aplicou o algoritmo com Aprendizado Federado FLPSO-SGD, foi utilizado um problema de Visão Computacional (CIFAR-10), através de uma arquitetura simples de CNN com três camadas convolucionais, duas de Pooling e duas camadas densas. Para o aprendizado federado, considera-se o número de partículas definidas como o número de clientes. A taxa de aprendizagem foi de 0.001, com 100 épocas do Servidor (rodadas), 10 clientes (partículas) cada um com 5 épocas de treinamento, com os hiper-parâmetros $c_1 = 0,8$, $c_2 = 1,5$, $w = 0,9$ decaindo linearmente conforme a época e iteração do cliente, até atingir o valor mínimo possível $w = 0,4$. A função de custo utilizada foi a entropia cruzada, as camadas intermediárias usam ReLU como função de ativação, a camada de saída (última camada densa), utiliza Softmax como classificador. Foram utilizadas a validação cruzada k-folds, com $k = 5$ para determinar as métricas do treinamento locais do Cliente. Os Experimentos computacionais desta fase foram realizados no mesmo computador da fase anterior, em conjunto com a placa de vídeo GPU, NVIDIA GeForce RTX 3060, com 12 GB de VRAM, com 3584 núcleos CUDA de processamento.

5. RESULTADOS E DISCUSSÃO

Os resultados experimentais são apresentados e discutidos nesta seção. Primeiramente, os resultados referentes ao algoritmo híbrido PSO-SGD e posteriomente ao método federado FLPSO-SGD.

A Tabela 3 apresenta os valores (média ± desvio padrão) de 15 execuções experimentais do tempo de treinamento em segundos $t(s)$, erro de validação L e da acurácia do modelo Acc, utilizando os conjuntos de dados descritos na (Tabela 2), mostrando a comparação dos algoritmos de otimização *Adam* (Kingman and Ba, 2015), e o algoritmo híbrido PSO-SGD proposto neste trabalho para o treinamento local do cliente. O tempo médio de processamento foi menor em 4 conjuntos de dados para PSO-SGD, e dos três que *Adam* foi menor, em dois conjuntos de dados (*Glass* e *Banknote*) a diferença entre as técnicas foi menor a três centésimas de segundo. Os experimentos também foram realizados com os algoritmos de otimização *Adagrad* (Dean et al., 2012) e *RMSprop* (Tieleman and Hinton, 2012). No entanto, como a técnica *Adam* demonstrou o melhor desempenho, a Tabela 3 apresenta exclusivamente os valores obtidos com *Adam* e PSO-SGD.

Tabela 3. Resultados da primeira fase dos experimentos

		Adam	PSO-SGD
Iris	*t (s)*	4,507 ±0,106	**4,338** ±0,080
	L	1,387E-01 ±0,219	**1,238E-05** ±1,50E-05
	Acc.	0,929 ±0,120	**0,973** ±0,034
B. Cancer	*t (s)*	4,564 ±0,070	**4,426** ±0,066
	L	3,748E-04 ±4,30E-04	**6,183E-06** ±7,02E-06
	Acc.	**0,963** ±0,009	0,931 ±0,011
Glass	*t (s)*	**4,637** ±0,046	4,668 ±0,076
	L	2,783E-01 ±0,180	**1,147E-02** ±0,009
	Acc.	0,676 ±0,079	**0,758** ±0,066
Banknote	*t (s)*	**4,845** ±0,062	4,871 ±0,141
	L	4,806E-03 ±0,004	**4,173E-07** ±7,90E-07
	Acc.	0,998 ±0,009	**0,999** ±0,002
Ionosphere	*t (s)*	4,781 ±0,084	**4,731** ±0,137
	L	4,226E-02 ±0,036	**7,790E-06** ±1,10E-05
	Acc.	**0,897** ±0,045	0,895 ±0,026
Sonar	*t (s)*	4,770 ±0,055	**4,731** 0,137
	L	9,603E-06 ±4,13E-06	**4,648E-14** ±1,71E-013
	Acc.	**1,000** ±0,000	**1,000** ±0,000
Wine	*t (s)*	**4,384** ±0,064	4,566 ±0,063
	L	9,926E0-05 ±1,70E-05	**5,069E-06** ±5,32E-05
	Acc.	**0,998** ±0,007	0,996 ±0,009

A Figura 5, estende a comparação em termos da acurácia para outros dois proeminentes algoritmos da literatura: AdaGrad (Dean et al., 2012) e RMSprop (Hinton et al., 2012). Pode-se verificar que os resultados do algoritmo híbrido PSO-SGD proposto para o treinamento local do cliente, obteve resultados de acurácia, em alguns conjunto de dados melhores que os outros otimizadores, em outros conjunto houve empate, e na maioria das vezes os resultados são melhores que os resultados de AdaGrad e RMSprop, ou seja, a validação da hibridização PSO-SGD proposto neste trabalho foi satisfatória, com resultados bastantes competitivos em relação ao atuais algoritmos de otimização de RNAs do estado-da-arte.

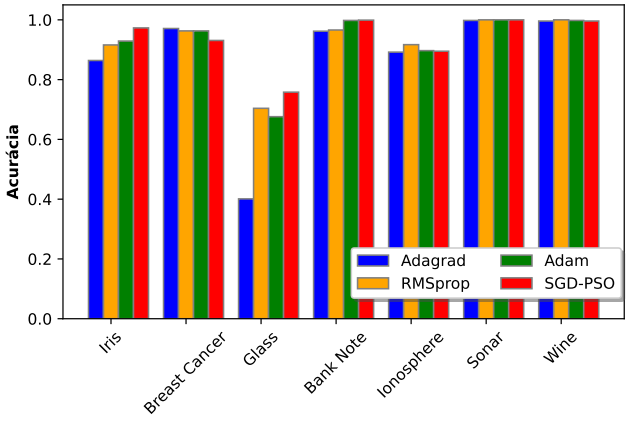

Figura 5. Comparação da acurácia dos modelos de 4 algoritmos de otimização, aplicados na primeira fase dos experimentos computacionais.

Após validar o algoritmo proposto para o treinamento local do cliente (primeira fase dos experimentos computacionais), a segunda fase de apresentação e discussão dos resultados envolve as aplicações experimentais do algoritmo de Aprendizado Federado FLPSO-SGD desenvolvido no presente trabalho, para a resolução do problema de Visão Computacional CIFAR-10.

Os resultados da acurácia para o problema CIFAR-10 dos algoritmos federados se encontram na Figura 6. A evolução das acurácias ao longo das 100 rodadas do servidor são mostradas para FLPSO-SGD (azul), FedPSO (laranja) e FedAvg (verde). A Figura 6 indica que FLPSO-SGD tem uma convergência mais rápida para resultados ótimos de acurácia comparado com as outras duas técnicas federadas implementadas, e no final das 100 rodadas, FLPSO-SGD obteve a maior acurácia com 0,7451, enquanto que para FedPSO e FedAvg, os resultados finais foram de 0,7158 e 0,6867 respectivamente.

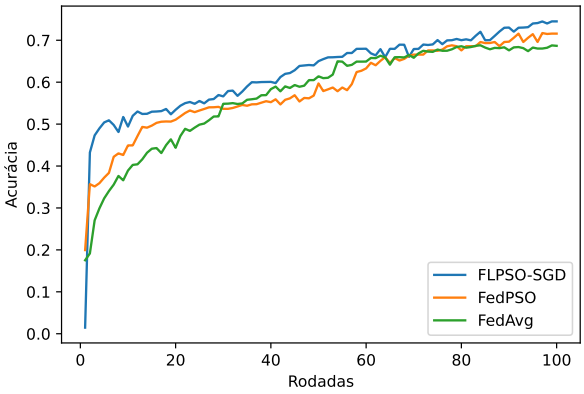

Figura 6. Acurácia de classifação da base de dados CIFAR-10 ao longo de 100 rodadas, utilizando FLPSO-SGD (azul), FedPSO (laranja) e FedAvg (verde)

6. CONCLUSÃO

As abordagens de FL emergem como uma solução robusta para treinamentos colaborativos e com dados distribuídos, assegurando a privacidade e segurança dos dados dos dispositivos. Este estudo apresentou o FLPSO-SGD como uma nova metodologia para o treinamento federado, utilizando o método híbrido PSO-SGD como algoritmo local nos clientes. Os resultados destacam o desempenho promissor do algoritmo PSO-SGD, evidenciado por seu tempo de treinamento eficiente, baixo erro de validação e alta acurácia. Além disso, a metodologia federada FLPSO-SGD demonstrou acurácias superiores no treinamento global em comparação com as técnicas FedAvg e FedPSO. Consequentemente, esses achados sugerem que a nova abordagem federada não melhora apenas o custo de comunicação, mas também eleva o desempenho global do sistema, consolidando-se como uma alternativa viável e eficaz para o treinamento distribuído em ambientes federados, especialmente em locais com redes de comunicação com largura de banda restrita nos clientes.

Este estudo avaliou exclusivamente a acurácia como métrica de FLPSO-SGD em um ambiente centralizado. Os trabalhos futuros estão direcionados em adicionar o cálculo do custo de comunicação, redes de treinamentos locais mais complexos e de oferecer uma nova solução FLPSO-SGD descentralizada.

AGRADECIMENTOS

Este trabalho teve o apoio de (i) Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Bolsa no. 312.991/2020-7; (ii) Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por meio do Programa de Excelência Acadêmica (PROEX) - Código de Financiamento 001; (iii) Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); (iv) Convênio de PD&I entre Fundep e UFMG – "Programa de Aprendizado de Máquina Federado para Veículos Conectados".

REFERÊNCIAS

Alpaydin, E. (2020). *Introduction to machine learning.* MIT press.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al. (2012). Large scale distributed deep networks. *Advances in neural information processing systems*, 25.

Hamer, J., Mohri, M., and Suresh, A.T. (2020). Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, 3973–3983. PMLR.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2.

Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A.T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, 5132–5143. PMLR.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization (pso). In *Proc. IEEE international conference on neural networks, Perth, Australia*, volume 4, 1942–1948.

Kingman, D. and Ba, J. (2015). Adam: A method for stochastic optimization. conference paper. In *3rd International Conference for Learning Representations*, volume 2015.

Konečnỳ, J., McMahan, H.B., Ramage, D., and Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4), 3347–3366.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2, 429–450.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Park, S., Suh, Y., and Lee, J. (2021). Fedpso: Federated learning using particle swarm optimization to reduce communication costs. *Sensors*, 21(2), 600.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 6.