COMPARAÇÃO DE OTIMIZADORES PARA CLASSIFICAÇÃO DE IMAGENS DE COLONOSCOPIA COM MOBILENETV2 E *TRANSFER LEARNING*

Rodrigo Luís Gasparino Lucatelli[1], Reginaldo José da Silva[2], Angela Leite Moreno[3]

[1]Universidade Federal de Alfenas, Alfenas, Brasil
(rodrigo.lucatelli@sou.unifal-mg.edu.br)
[2]Universidade Estadual Paulista "Júlio de Mesquita Filho", Ilha Solteira, Brasil
[3]Universidade Federal de Alfenas, Alfenas, Brasil

Resumo: O treinamento de Redes Neurais Convolucionais depende de diversos hiperparâmetros que afetam o desempenho do modelo. Um desses parâmetros é o algoritmo otimizador, responsável por orientar a convergência da rede durante o treinamento. Neste trabalho, quatro otimizadores são comparados quando aplicados a um conjunto de imagens colonoscópicas, com 50 sementes distintas para cada algoritmo, totalizando 200 modelos treinados. A análise baseia-se em intervalos de confiança de 95% calculados sobre as métricas obtidas. Conclui-se que, para este conjunto de dados, os otimizadores com taxa de aprendizado adaptativa, Adam e RMSprop, apresentaram desempenho superior ao SGD e ao SGDM em todas as métricas avaliadas.

Palavras-chave: Câncer Colorretal; Diagnóstico Auxiliado por Computador; *Deep Learning*; Otimizadores; Redes Convolucionais.

INTRODUÇÃO

O câncer colorretal (ou câncer de cólon e reto) figura atualmente como uma das neoplasias malignas mais incidentes do mundo. Em 2022, foram estimados aproximadamente 1,9 milhão de casos novos e 900 mil mortes pela doença em escala global, o que a posiciona entre as principais causas de morte por câncer (Bray et al., 2024). A maior parte dos casos se desenvolve a partir de pólipos adenomatosos, lesões precursoras que evoluem lentamente, ao longo dos anos, até assumirem características malignas. Essa progressão relativamente lenta abre uma janela de oportunidade para o diagnóstico precoce, que exerce papel determinante na sobrevida do paciente: a identificação e a remoção desses pólipos em estágios iniciais reduzem a incidência e a mortalidade da doença de forma expressiva, como demonstrado no estudo de acompanhamento de longo prazo conduzido por Winawer et al. (1993) com pacientes submetidos a polipectomia colonoscópica.

Na prática clínica, a colonoscopia é reconhecida como o método padrão-ouro para o rastreamento de pólipos e demais lesões precursoras do câncer colorretal, permitindo tanto a visualização direta da mucosa intestinal quanto a remoção imediata das lesões encontradas. A precisão do exame, porém, não depende apenas da tecnologia do aparelho: está sujeita também à experiência do gastroenterologista e a fatores humanos como a fadiga, tempo de retirada do colonoscópio e variação na percepção visual entre os examinadores. Rex et al. (1997), em um estudo com colonoscopias realizadas em sequência pelo mesmo paciente, estimaram taxas de não detecção de adenomas da ordem de 24% para pólipos menores que 5 mm, evidenciando que mesmo exames bem executados podem deixar passar lesões clinicamente relevantes.

Para mitigar essas limitações, sistemas de diagnóstico auxiliado por computador baseados em Redes Neurais Convolucionais (CNNs) vêm sendo incorporados à rotina endoscópica como ferramenta de apoio à decisão. Wang et al. (2018) desenvolveram e validaram clinicamente um algoritmo de aprendizado profundo capaz de detectar pólipos em tempo real durante o exame, reduzindo a taxa de lesões não detectadas em um ensaio randomizado. Resultado semelhante foi reportado por Mori et al. (2018), que avaliaram o uso de inteligência artificial na identificação de pólipos diminutos durante a colonoscopia, observando ganhos de eficiência sem aumento expressivo do tempo de exame. Esses sistemas dividem-se, de modo geral, em duas categorias: os voltados à detecção de lesões, denominados CADe (Computer-Aided Detection), responsáveis por sinalizar a presença de anomalias durante o exame, e os voltados à classificação de patologias já localizadas, denominados CADx (Computer-Aided Diagnosis), que é a categoria de problema trata neste trabalho.

Apesar do sucesso dessas arquiteturas de aprendizado profundo, seu desempenho final não depende apenas da escolha da rede neural. Bottou, Curtis e Nocedal (2018), em revisão sobre métodos de otimização para aprendizado de máquina em larga escala, destacam que o algoritmo de otimização influencia diretamente a velocidade de convergência, a estabilidade do treinamento e a capacidade de generalização do modelo final, podendo levar a resultados bastante distintos mesmo quando a arquitetura da rede permanece inalterada. Essa observação é reforçada por Ruder (2016), que descreve as principais limitações de algoritmos como o SGD em paisagens de perda irregulares e justifica o desenvolvimento de variantes adaptativas como o Adam e o RMSprop. Ainda assim, poucos trabalhos sobre classificação de imagens colonoscópicas comparam sistematicamente o efeito de diferentes otimizadores sobre o mesmo modelo e o mesmo conjunto de dados, controlando as fontes de variação estocástica do treinamento.

Diante dessa lacuna, este trabalho analisa e compara quatro algoritmos de otimização aplicados ao treinamento de um modelo de Rede Neural Convolucional sobre o conjunto de dados CRCCD_V1 (Dash et al., 2024), composto por imagens de exames de colonoscopia. O objetivo consiste em avaliar o comportamento de cada otimizador frente às fontes de variação estocástica do treinamento, identificando diferenças de desempenho e de estabilidade entre os algoritmos e discutindo suas implicações para a escolha de hiperparâmetros em sistemas de diagnóstico auxiliado por computador voltados a dispositivos com recursos computacionais limitados.

REFERENCIAL TEÓRICO

Quatro otimizadores são analisados neste trabalho: *Stochastic Gradient Descent* (SGD), *Stochastic Gradient Descent with Momentum* (SGDM), *Root Mean Square Propagation* (RMSprop) e *Adaptive Moment Estimation* (Adam). Cada um desses algoritmos possui abordagem matemática própria para a atualização dos pesos da rede neural durante o processo de retropropagação, influenciando a velocidade de convergência e a estabilidade do treinamento. A seguir, descrevem-se os mecanismos de funcionamento de cada algoritmo, na ordem em que foram propostos historicamente.

Stochastic Gradient Descent (SGD)

O SGD é um algoritmo de otimização direto e computacionalmente econômico (Robbins e Monro, 1951). Diferente do Gradiente Descendente convencional, que calcula os gradientes utilizando todo o conjunto de dados de uma só vez, o SGD realiza a atualização dos parâmetros, pesos e viés, a cada iteração, baseando-se em um único exemplo de treino ou, em uma variação comum, em um pequeno lote (*mini-batch*). A equação fundamental de atualização do SGD é dada por:

$$w_{t+1} = w_t - \eta \cdot \nabla L(w_t) \quad (1)$$

em que ω(t) representa os pesos no instante t, η é a taxa de aprendizado e ∇L(ω(t)) é o gradiente da função de perda em relação aos pesos. Por calcular o gradiente com base em poucos dados por vez, o SGD introduz alta variância nas atualizações dos pesos, gerando flutuação na função de perda. Essa flutuação pode auxiliar o modelo a escapar de mínimos locais, mas também dificulta a convergência para o mínimo global, fazendo o modelo oscilar continuamente ao redor do ponto ideal e aumentando o tempo total de treinamento.

Stochastic Gradient Descent with Momentum (SGDM)

Para atenuar as oscilações excessivas causadas pelo SGD, especialmente em superfícies de perda com gradientes assimétricos, emprega-se o SGD com momentum (Polyak, 1964; Sutskever et al., 2013), aqui abreviado como SGDM. O método simula um efeito físico de inércia, no qual uma esfera pesada acumula velocidade ao rolar pela inclinação da função de perda. O algoritmo introduz uma variável de velocidade v(t), que retém uma média móvel exponencial dos gradientes passados:

$$v_t = \gamma \cdot v_{t-1} + \eta \cdot \nabla L(w_t) \quad (2)$$

$$w_{t+1} = w_t - v_t \quad (3)$$

em que γ é o coeficiente de momentum, que determina a fração da velocidade anterior mantida a cada passo. O momentum acelera a convergência na direção correta e amortece oscilações irrelevantes, auxiliando a superar mínimos locais. A inércia acumulada, contudo, pode fazer o algoritmo ultrapassar o mínimo global, exigindo mais iterações para estabilizar no ponto ótimo.

Root Mean Square Propagation (RMSprop)

O RMSprop (Tieleman, 2012) é um algoritmo de otimização adaptativo desenvolvido para resolver o problema do decaimento prematuro da taxa de aprendizado encontrado no AdaGrad (Duchi et al., 2011). Em vez de aplicar uma taxa de aprendizado estática para todos os parâmetros, o RMSprop ajusta a taxa de aprendizado de cada peso de maneira individual e dinâmica, dividindo a taxa pela raiz quadrada da média móvel exponencial dos quadrados dos gradientes passados:

$$v_t = \beta \cdot v_{t-1} + (1-\beta) \cdot [\nabla L(w_t)]^2 \quad (4)$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t} + \epsilon} \cdot \nabla L(w_t) \quad (5)$$

em que β é o hiperparâmetro de decaimento, geralmente fixado em 0,9, e ε = 10^{-8} é uma constante que evita divisões por zero. Ao normalizar os gradientes, o RMSprop equaliza o tamanho dos passos de atualização: parâmetros com gradientes grandes têm seus passos reduzidos, enquanto parâmetros com gradientes pequenos recebem passos maiores. Isso torna o algoritmo eficiente para problemas não-estacionários e CNNs profundas, embora exija o ajuste cuidadoso do hiperparâmetro β.

Adaptive Moment Estimation (Adam)

O Adam (Kingma e Ba, 2014) combina os princípios do SGDM e do RMSprop. O algoritmo calcula taxas de aprendizado adaptativas para cada parâmetro, armazenando tanto a média móvel exponencial dos gradientes, o primeiro momento, correspondente à média, quanto a média móvel exponencial dos quadrados dos gradientes, o segundo momento, correspondente à variância não-centrada. As equações de atualização incorporam uma correção de viés para as iterações iniciais, dado que as médias são inicializadas em zero:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla L(w_t) \quad (6)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot [\nabla L(w_t)]^2 \quad (7)$$

A atualização final dos pesos, após a correção de viés, é expressa por:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \quad (8)$$

Os valores padrão amplamente utilizados são $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ e ε = 10^{-8} (Kingma e Ba, 2014). O Adam é considerado o otimizador padrão na maior parte das arquiteturas de *deep learning* atuais, por sua convergência rápida e baixa necessidade de ajuste manual de hiperparâmetros. Estudos da literatura de aprendizado de máquina apontam, contudo, que em casos específicos de classificação de imagens o Adam pode deixar de convergir para o melhor ótimo global em comparação ao SGD quando este é configurado corretamente (Reddi, Kale e Kumar, 2018), o que justifica a análise comparativa realizada neste trabalho com diferentes sementes.

MATERIAL E MÉTODOS

Conjunto de dados

O conjunto de dados utilizado foi o CRCCD_V1 (Dash et al., 2024), composto por imagens de exames de colonoscopia rotuladas para a tarefa de classificação de lesões. Do conjunto original, duas classes foram removidas deste conjunto (ULCERATIVE_COLITIS e ESOPHAGITIS), por não corresponderem a patologias colorretais neoplásicas de interesse para a tarefa de classificação. As imagens remanescentes foram redimensionadas e normalizadas conforme o padrão de entrada exigido pela MobileNetV2, ou seja, uma resolução de 224 por 224, com 3 canais de entrada (RGB). Além disso, foram aplicadas técnicas de aumento de dados (*data augmentation*) às imagens do conjunto de treino, introduzindo variações aleatórias a cada época para reduzir o risco de sobreajuste. Essas alterações são: rotação da imagem em ±30°, movimentação da imagem nos eixos horizontais e verticais de, no máximo, 20% do tamanho do eixo da imagem, variação de escala (zoom) de até 20% e espelhamento horizontal e vertical aleatório das imagens.

Divisão dos dados

O conjunto de dados foi particionado, uma única vez com semente fixa (seed = 22), em subconjuntos de treino, validação e teste na proporção 60/20/20, totalizando 8.750 imagens distribuídas entre 14 classes: 5.250 para treino, 1.750 para validação e 1.750 para teste. A partição foi criada uma única vez com semente fixa (seed = 22) e permanece idêntica para todas as 200 execuções. As variações de desempenho entre execuções de um mesmo otimizador refletem, portanto, a aleatoriedade na inicialização dos pesos das camadas densas adicionadas ao topo da rede e na ordem de apresentação dos lotes de treino, e não diferenças na composição dos subconjuntos de dados. Os pesos das camadas convolucionais da MobileNetV2 partem sempre dos valores pré-treinados no ImageNet, fixos e idênticos para todas as execuções.

Modelo e configuração de treinamento

Para o treinamento, foi adotado um modelo pré-treinado com *transfer learning* da MobileNetV2 (Sandler et al., 2018), arquitetura com número reduzido de parâmetros, projetada para execução em dispositivos móveis e capaz de atingir bom desempenho com custo computacional baixo, característica relevante para sistemas de apoio diagnóstico voltados a equipamentos com recursos limitados.

O treinamento foi conduzido com lote de 64 imagens e limite máximo de 500 épocas, divididas em duas fases. Na primeira fase, de 10 épocas, as camadas da MobileNetV2 permaneceram congeladas, permitindo ao modelo adaptar apenas o classificador adicionado ao topo da rede, composto por uma camada densa de 128 neurônios com ativação ReLU, uma camada de *Dropout* de 0,5 e uma camada de saída com ativação *softmax*. Na segunda fase, de até 490 épocas, as 30 últimas camadas foram descongeladas, com exceção das camadas de *batch normalization*, mantidas congeladas para preservar as estatísticas de normalização aprendidas durante o pré-treinamento na ImageNet. Na segunda fase, foram aplicados parada antecipada e redução da taxa de aprendizado, ambos com paciência de 10 épocas, monitorando acurácia de validação e perda de validação, respectivamente.

Esses valores foram definidos empiricamente para equilibrar tempo de treinamento e qualidade da convergência, evitando tanto o encerramento prematuro quanto o sobreajuste.

Cada otimizador foi configurado com taxas de aprendizado distintas para cada fase do treinamento, conforme a Tabela 1.

Tabela 1. Taxas de aprendizado por otimizador por fase de treinamento.

Otimizador	Fase 1	Fase 2
Adam	0,001	0,0001
RMSprop	0,001	0,0001
SGD	0,01	0,001
SGDM	0,001	0,0001

A taxa de aprendizado do SGD na primeira fase foi fixada em um valor mais alto que o dos demais algoritmos, 0,01 contra 0,001, por se tratar de um otimizador com taxa fixa para todos os parâmetros, que exige um passo inicial maior para compensar a ausência de adaptação individual presente nos otimizadores adaptativos. O coeficiente de momentum do SGDM foi fixado em $\gamma = 0{,}9$.

Número de simulações

Cada um dos quatro otimizadores foi treinado com 50 sementes distintas, totalizando 200 modelos treinados em condições diferentes. A semente controla a inicialização dos pesos das camadas do classificador adicionado ao topo da MobileNetV2 (*Dense* de 128 neurônios, *Dropout* e camada de saída) e a ordem de apresentação dos lotes durante o treinamento. Os pesos das camadas convolucionais da MobileNetV2 partem sempre dos valores pré-treinados no ImageNet, fixos e idênticos para todas as execuções. Como a partição dos dados também é fixa, as variações de desempenho entre execuções de um mesmo otimizador refletem exclusivamente a aleatoriedade na inicialização das camadas densas e na ordem dos lotes, e não diferenças na composição dos subconjuntos de treino, validação e teste. O número de repetições foi definido por critério prático: com n = 50 execuções, a distribuição amostral da média converge para a normal pelo Teorema Central do Limite, o que valida o uso de intervalos de confiança para estimativa do desempenho médio de cada otimizador; 50 repetições oferecem margem adicional de estabilidade para essa estimativa sem tornar o custo computacional total impraticável dado o hardware disponível.

Intervalos de confiança

Para cada métrica de cada otimizador, foi calculado um intervalo de confiança de 95% com base nos 50 valores obtidos nas diferentes sementes, assumindo distribuição t de Student dado o tamanho amostral finito. A comparação entre otimizadores foi realizada por meio da sobreposição dos intervalos: pares cujos intervalos não se sobrepõem fornecem evidência descritiva de diferença entre os grupos, enquanto pares com sobreposição são considerados inconclusivos quanto à existência de diferença. A ausência de sobreposição não equivale a um teste de hipótese formal, e as conclusões comparativas apresentadas neste trabalho têm caráter descritivo e exploratório.

Métricas de avaliação

As métricas utilizadas para comparar os otimizadores foram calculadas a partir dos valores de Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN) da matriz de confusão. A acurácia representa a proporção total de predições corretas em relação ao total de imagens avaliadas no conjunto de teste, sendo expressa por:

$$Acur = \frac{VP + VN}{VP + VN + FP + FN} \tag{9}$$

Embora seja a métrica mais intuitiva, sua interpretação isolada pode mascarar limitações do modelo, pois um modelo pode ter uma alta acurácia classificando apenas a classe majoritária e ignorando a minoritária.

A sensibilidade avalia a proporção de verdadeiros positivos dentre todos os casos reais positivos:

$$Sens = \frac{VP}{VP + FN} \tag{10}$$

sendo importante em sistemas de triagem médica por minimizar a ocorrência de Falsos Negativos, reduzindo o risco de que um paciente doente seja classificado como saudável. No contexto multiclasse deste trabalho, a sensibilidade é calculada como média macro entre as classes.

Complementar à sensibilidade, a especificidade avalia a capacidade do classificador de identificar corretamente os casos negativos:

$$Espec = \frac{VN}{VN + FP} \tag{11}$$

sendo fundamental para reduzir o índice de Falsos Positivos e evitar que pacientes saudáveis sejam submetidos a tratamentos e exames desnecessários.

O F1-Score é definido como a média harmônica entre a Precisão, proporção de positivos preditos que são realmente corretos, e a Sensibilidade, sendo calculado por:

$$F1\text{-}Score = \frac{2 \cdot VP}{2 \cdot VP + FP + FN} \tag{12}$$

Diferente da média aritmética, a média harmônica penaliza desequilíbrios entre essas duas métricas de forma mais acentuada, o que a torna especialmente útil para dados com classes desbalanceadas, embora sua

principal limitação resida em desconsiderar os Verdadeiros Negativos, concentrando-se exclusivamente no desempenho sobre a classe positiva.

A área sob a curva de Característica de Operação do Receptor (AUC-ROC) avalia a capacidade do modelo de distinguir entre as classes ao longo de múltiplos limiares de decisão, sendo obtida ao plotar a Taxa de Verdadeiros Positivos no eixo Y contra a Taxa de Falsos Positivos no eixo X. O valor varia de 0,5, desempenho equivalente a uma predição aleatória, a 1,0, separação perfeita entre as classes. Sua principal vantagem é a invariância em relação ao limiar de classificação e ao desbalanceamento de classes, embora em cenários de desbalanceamento extremo a métrica possa apresentar superestimação do desempenho real do modelo.

Por fim, o Coeficiente de Correlação de Matthews (MCC) foi adotado como métrica principal deste trabalho. Amplamente reconhecido por sua confiabilidade em cenários com grande desbalanceamento entre classes (Matthews, 1975; Chicco e Jurman, 2020), o MCC avalia a qualidade global da classificação integrando de forma igualitária as quatro categorias da matriz de confusão, ao contrário da acurácia, que pode mascarar resultados ruins em classes minoritárias, ou do F1-Score, que ignora os Verdadeiros Negativos. O coeficiente resulta em um valor entre −1 e +1, em que +1 representa classificação perfeita, 0 indica desempenho equivalente a uma predição aleatória e −1 indica discordância absoluta entre o previsto e o real, sendo calculado por:

$$MCC = \frac{VP \cdot VN - FP \cdot FN}{\sqrt{(VP+FP)(VP+FN)(VN+FP)(VN+FN)}} \quad (13)$$

Por considerar todas as células da matriz de confusão de forma igualitária, um modelo só atinge MCC próximo de 1 se identificar corretamente todas as classes, sem ignorar as de menor número de amostras, o que garante que mesmo em conjuntos desbalanceados um MCC elevado indique de fato boa capacidade discriminativa.

Hardware

O treinamento dos modelos foi realizado em um computador pessoal equipado com processador Ryzen 5 7600X, 16 GB de memória RAM e placa de vídeo dedicada Radeon RX 9060XT com 16 GB de VRAM, com sistema operacional Windows 11. O treinamento foi conduzido preferencialmente sem outros processos simultâneos, a fim de minimizar interferências no tempo de execução.

RESULTADOS E DISCUSSÃO

As métricas médias obtidas pelos quatro otimizadores ao longo das 50 sementes são apresentadas na Tabela 2.

Observa-se que o SGD e o SGDM obtiveram resultados semelhantes entre si, com o SGDM apresentando leve vantagem sobre o SGD, por exemplo, MCC médio de 0,9254 contra 0,9241. O Adam e o RMSprop, por sua vez, apresentaram desempenho superior em todas as métricas, com MCC médio de 0,9749 e 0,9744, respectivamente.

Isso sugere que o momentum tem impacto marginal neste conjunto de dados, enquanto a distinção mais relevante observada está entre os otimizadores com taxa de aprendizado fixa, SGD e SGDM, e os com taxa adaptativa, Adam e RMSprop, mecanismo originado no AdaGrad (Duchi et al., 2011) e incorporado a esses dois últimos algoritmos.

Tabela 2. Médias por otimizador.

Otimizador	Acur.	Sens.	Espec.	F1-Score	AUC-ROC	MCC
Adam	0,9769	0,9769	0,9979	0,9770	0,9996	0,9749
RMSprop	0,9764	0,9764	0,9979	0,9765	0,9996	0,9744
SGD	0,9302	0,9302	0,9937	0,9304	0,9953	0,9241
SGDM	0,9314	0,9314	0,9938	0,9316	0,9953	0,9254

Tabela 3. Métricas do melhor modelo (Adam, semente 342).

Otimizador	Acur.	Sens.	Espec.	F1-Score	AUC-ROC	MCC
Adam	0,9887	0,9887	0,999	0,9887	0,9998	0,9876

O modelo com os melhores resultados médios foi treinado com o Adam e semente 342, conforme a Tabela 3, atingindo métricas elevadas para o conjunto de dados embora treinado por apenas 66 épocas. O desempenho acima da média deve ser atribuído à inicialização dos pesos das camadas densas e à ordem dos lotes determinados por essa semente específica, que produziram uma trajetória de otimização mais eficiente durante o fine-tuning. A partição dos dados e os pesos pré-treinados da MobileNetV2 são fixos e idênticos para todas as execuções.

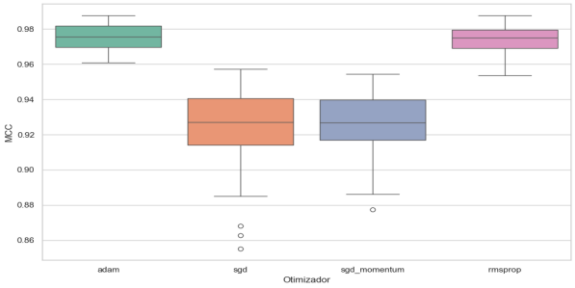

Figura 1. Distribuição do MCC por Otimizador.

A Figura 1 apresenta a distribuição do MCC obtido por cada otimizador ao longo das 50 sementes, na forma de gráfico de caixas. O Adam e o RMSprop concentram seus valores em uma faixa estreita, próxima de 0,97, com poucos pontos fora do intervalo interquartil, o que indica baixa dispersão entre execuções, refletida nos intervalos de confiança de ±0,0021 e ±0,0022, respectivamente. O SGD e o SGDM, por outro lado, exibem caixas mais alargadas e amplitude interquartil maior, sinalizando que a escolha da semente tem efeito mais pronunciado sobre o resultado final desses dois algoritmos. O SGD, em particular, apresenta a maior dispersão entre todos os otimizadores avaliados, além de alguns valores discrepantes abaixo da caixa, o que sugere que, em determinadas inicializações estocásticas dos pesos, esse otimizador pode convergir para soluções de qualidade inferior.

O número de épocas necessário para a convergência e o tempo de treinamento associado foram analisados em paralelo, pois comparar apenas o tempo de treinamento pode ser enganoso: o tempo por época varia conforme o custo computacional interno de cada otimizador, podendo levar a conclusões distorcidas sobre a eficiência real do algoritmo.

Na Figura 2, observa-se a relação entre o número de épocas treinadas e o MCC final de cada modelo. Os pontos referentes ao Adam e ao RMSprop concentram-se entre 40 e 70 épocas, atingindo valores de MCC já elevados nessa faixa, o que indica convergência relativamente rápida. Os pontos do SGD e do SGDM, por sua vez, distribuem-se em uma faixa mais ampla, entre 60 e 140 épocas, e exigem, em média, cerca de 90 épocas para atingir desempenho comparável, evidenciando uma trajetória de convergência mais lenta e irregular.

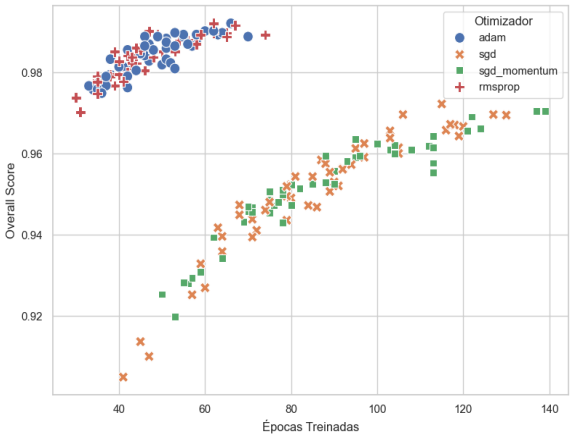

Figura 2. Distribuição do MCC por épocas treinadas.

A Figura 3 reproduz o mesmo padrão observado na Figura 2, agora em função do tempo de treinamento em minutos, o que confirma que a diferença não decorre apenas do número de épocas, mas também do tempo necessário para cada uma delas. O Adam e o RMSprop convergem em aproximadamente 7 minutos, enquanto o SGD e o SGDM demandam cerca de 15 minutos, praticamente o dobro do tempo de convergência dos otimizadores adaptativos nas condições de hardware utilizadas neste trabalho.

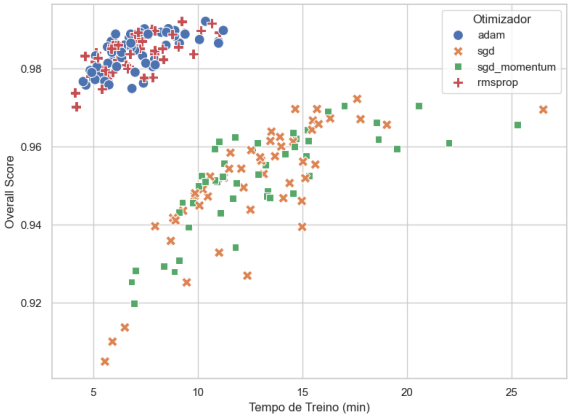

Figura 3. Distribuição do MCC por tempo em minutos.

Os intervalos de confiança de 95% calculados sobre as 50 sementes para cada métrica são apresentados na Tabela 4.

Os intervalos confirmam a baixa variabilidade já observada na Figura 1, sendo mais estreitos para Adam e RMSprop. Isso significa que, caso o experimento fosse replicado muitas vezes, 95% dos intervalos calculados conteriam o verdadeiro valor médio da métrica para aquele otimizador, e intervalos mais estreitos indicam maior estabilidade frente à variação das sementes.

Tabela 4. Intervalos de confiança de 95% por otimizador.

Otimizador	Acur.	Sens.	Espec.	F1-Score	AUC-ROC	MCC
Adam	0,9769 ± 0,0020	0,9769 ± 0,0020	0,9979 ± 0,0002	0,9770 ± 0,0019	0,9996 ± 0,0000	0,9749 ± 0,0021
RMSprop	0,9764 ± 0,0020	0,9764 ± 0,0020	0,9979 ± 0,0002	0,9765 ± 0,0020	0,9996 ± 0,0001	0,9744 ± 0,0022
SGD	0,9302 ± 0,0061	0,9302 ± 0,0061	0,9937 ± 0,0006	0,9304 ± 0,0061	0,9953 ± 0,0007	0,9241 ± 0,0066
SGDM	0,9314 ± 0,0048	0,9314 ± 0,0048	0,9938 ± 0,0004	0,9316 ± 0,0049	0,9953 ± 0,0006	0,9254 ± 0,0053

Os intervalos de confiança de Adam e RMSprop não se sobrepõem aos de SGD e SGDM em nenhuma das métricas avaliadas, fornecendo evidência descritiva de que os otimizadores adaptativos apresentaram desempenho superior neste conjunto de dados. Embora o Adam tenha obtido médias ligeiramente superiores às do RMSprop, a ampla sobreposição entre os intervalos dos dois algoritmos indica que não é possível distingui-los descritivamente com base apenas nesse recurso, o que vale igualmente para a comparação entre SGD e SGDM. A sobreposição dos intervalos entre esses dois últimos sugere que o impacto do momentum é menor do que o da taxa de aprendizado adaptativa neste conjunto de dados. Essa interpretação tem caráter exploratório e não substitui uma análise inferencial formal.

Cabe registrar, todavia, uma limitação metodológica relevante na comparação entre os otimizadores: durante o *fine-tuning* (fase 2), o SGD foi configurado com taxa de aprendizado de 0,001, ao passo que Adam, RMSprop e SGDM operaram com 0,0001, uma diferença de dez vezes. Essa assimetria nos hiperparâmetros pode ter contribuído para a maior instabilidade e dispersão observadas no SGD, tornando a comparação entre otimizadores parcialmente confundida por essa diferença de configuração. Experimentos futuros devem garantir a equivalência das taxas de aprendizado para isolar o efeito do algoritmo de otimização.

CONCLUSÃO

Diante dos resultados obtidos, conclui-se que, dos otimizadores analisados, o Adam e o RMSprop atingiram desempenho superior em menos tempo, enquanto o SGD e o SGDM apresentaram desempenho inferior e demandaram aproximadamente o dobro do tempo de convergência. Esse comportamento está associado à taxa de aprendizado adaptativa, mecanismo originado no AdaGrad (Duchi et al., 2011) e incorporado primeiro ao RMSprop e, na sequência, ao Adam. A adaptação individual da taxa de aprendizado por parâmetro permite que esses algoritmos se ajustem mais rapidamente às características do conjunto de dados. Para uma dada semente, todos os otimizadores partem da mesma inicialização: pesos da base MobileNetV2 provenientes do pré-treinamento na ImageNet (fixos) e pesos das camadas densas gerados estocasticamente pela semente, o que garante que as diferenças observadas se devem ao algoritmo de otimização e não ao ponto de partida.

A complexidade adicional do cálculo com momentum não se justifica diante da melhora marginal observada entre SGD e SGDM. A principal diferença identificada reside, portanto, na capacidade de adaptação da taxa de aprendizado durante o treinamento, mecanismo presente no Adam e no RMSprop e ausente no SGD e no SGDM.

Como limitação deste trabalho, destaca-se a ausência de testes estatísticos formais para confirmar as diferenças observadas pelos intervalos de confiança, a restrição a um único conjunto de dados, e a assimetria nas taxas de aprendizado do fine-tuning: o SGD operou com taxa de 0,001, enquanto os demais otimizadores utilizaram 0,0001, o que pode ter influenciado os resultados comparativos e limita a atribuição das diferenças ao algoritmo de otimização em si. Para trabalhos futuros, propõe-se replicar o experimento com taxas de aprendizado equivalentes entre todos os otimizadores; testar os algoritmos sobre outros conjuntos de dados de colonoscopia, a fim de avaliar a generalização dos resultados; incluir otimizadores mais recentes, como AdamW e RAdam; e aplicar testes estatísticos formais, como Wilcoxon e Friedman, capazes de confirmar com rigor inferencial as tendências observadas neste trabalho de forma apenas descritiva.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Brasil, processo nº BIS-00732-25.

REFERÊNCIAS

Bottou, L.; Curtis, F. E.; Nocedal, J. Optimization methods for large-scale machine learning. SIAM Review, v. 60, n. 2, p. 223–311, 2018.

Bray, F. et al. Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians, v. 74, n. 3, p. 229–263, 2024.

Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, v. 21, n. 6, 2020.

Dash, A. et al. CRCCD V1 (Colorectal Cancer Classification and Detection). Mendeley Data, Version 2, 2024. doi: 10.17632/2pybr7f7yc.2.

Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, v. 12, n. 61, 2011.

Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. International Conference on Learning Representations, v. 3, 2015.

Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA): Protein Structure, v. 405, n. 2, p. 442–451, 1975.

Mori, Y. et al. Real-time use of artificial intelligence in identification of diminutive polyps during colonoscopy. Annals of Internal Medicine, v. 169, n. 6, p. 357–366, 2018.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, v. 4, n. 5, p. 1–17, 1964.

Reddi, S. J.; Kale, S.; Kumar, S. On the convergence of Adam and beyond. International Conference on Learning Representations, v. 6. 2018.

Rex, D. K. et al. Colonoscopic miss rates of adenomas determined by back-to-back colonoscopies. Gastroenterology, v. 112, n. 1, p. 24–28, 1997.

Robbins, H.; Monro, S. A stochastic approximation method. The Annals of Mathematical Statistics, v. 22, n. 3, p. 400–407, 1951.

Ruder, S. An overview of gradient descent optimization algorithms. arXiv preprint, arXiv:1609.04747, 2016.

Sandler, M. et al. MobileNetV2: inverted residuals and linear bottlenecks. IEEE/CVF Conference on Computer Vision and Pattern Recognition, p. 4510–4520, 2018.

Sutskever, I. et al. On the importance of initialization and momentum in deep learning. International Conference on Machine Learning, v. 28, p. 1139–1147, 2013.

Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, v. 4, n. 2, p. 26–31, 2012.

Wang, P. et al. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. Nature Biomedical Engineering, v. 2, n. 10, p. 741–748, 2018.

Winawer, S. J. et al. Prevention of colorectal cancer by colonoscopic polypectomy. The National Polyp Study Workgroup. The New England Journal of Medicine, v. 329, n. 27, p. 1977–1981, 1993.