

# Um Estudo Comparativo Entre Cinco Métodos de Otimização Aplicados Em Uma RNC Voltada ao Diagnóstico do Glaucoma

Athyron M. Ribeiro<sup>1</sup>, Francisco de Paula S. Araujo Junior<sup>1</sup>

<sup>1</sup>Universidade Estadual do Piauí (UESPI)

Av. Nossa Senhora de Fátima, S/N, Bairro de Fátima, CEP: 64202-220

athyrsonmr@gmail.com, pjhatata@hotmail.com

## ABSTRACT

In this work a comparative study was made between five optimization methods (Adam, Adagrad, RMSprop, SGD with Momentum and Adabound) that were applied in a convolutional neural network implemented with Inception V3 architecture aimed at identifying glaucoma through images. The images used on this work were taken from five public image datasets available on the internet. The results show that, between the evaluated methods, the method Adabound achieves the best results during and after the training process.

## Keywords

Glaucoma; Convolutional Neural Networks; Optimization Methods.

## RESUMO

Neste trabalho foi realizada uma comparação entre cinco métodos de otimização (Adam, Adagrad, RMSprop, SGD com Momentum e Adabound) aplicados em uma rede neural convolucional implementada com a arquitetura Inception V3 voltada a identificação de glaucoma por meio de imagens. As imagens utilizadas nesta pesquisa foram extraídas de 5 bancos de imagens públicos disponíveis na internet. É observado que, dentre os métodos testados, Adabound foi o método que obteve melhores resultados durante e após o treinamento da rede.

## Palavras-Chave

Glaucoma; Redes Neurais Convolucionais; Métodos de Otimização.

## 1. INTRODUÇÃO

Glaucoma é o nome dado a uma classe de doenças que afetam o nervo óptico causando danos. Esta é a segunda patologia que mais causa perda de visão no mundo [8]. Segundo dados da Organização Mundial de Saúde, existem cerca de 60 milhões de glaucomatosos em todo o mundo, e a cada ano, mais 2,4 milhões de casos aparecem [3]. De acordo com [12] em 2020, cerca de 80 milhões de pessoas terão essa doença. A perda de visão causada pelo glaucoma é irreversível, porém com o tratamento precoce, o avanço desta doença pode ser significativamente atrasado. Por ser uma doença assintomática nos seus estágios iniciais, a detecção desta patologia só pode ser realizada por meio de exames clínicos. Sob a vista disso, tecnologias que auxiliam na identificação do glaucoma utilizando imagens do nervo óptico já vem sendo utilizadas por alguns profissionais da área, e grandes bancos de dados vem sendo construídos.

Redes Neurais Convolucionais (RNCs) são soluções estado da arte para a uma enorme quantidade de problemas. Desde 2014,

devido às várias vitórias alcançadas em diversas competições de classificação, como por exemplo a ImageNet competition, as RNCs vem se popularizando [17]. Dentre os usos possíveis dessa tecnologia destaca-se o auxílio no diagnóstico precoce de diversas doenças, dentre elas o glaucoma. Entretanto a precisão dos resultados obtidos pelas atuais RNCs voltadas para diagnósticos de doenças em muitos casos ainda não é a ideal para que elas sejam implementadas como ferramenta de auxílio em uma grande quantidade de consultórios médicos. As RNCs se apresentam em diversas arquiteturas e variedade de parâmetros.

Dentre os parâmetros que mais influenciam no desempenho de uma rede neural está a escolha do otimizador, função que tem como objetivo diminuir o erro entre os resultados obtidos por uma rede em comparação com os resultados desejados. De modo geral, quanto menor o erro obtido por uma rede neural, maior a acurácia dos resultados e consequentemente melhor o desempenho da rede.

Esse trabalho almeja avaliar a influência da escolha do otimizador na acurácia de uma RNC voltada a identificação do glaucoma utilizando imagens do disco óptico, comparando cinco diferentes métodos de otimização e verificando qual dentre os métodos testados contribui mais para o aumento da acurácia da rede. Para este trabalho foram escolhidos os seguintes métodos de otimização para comparação de desempenho, conhecidos como Adam [7], Adagrad [4], RMSprop [6], SGD com momentum [16] e Adabound [10].

Uma das principais contribuições deste trabalho é a validação do desempenho do método Adabound na classificação de imagens em duas categorias (imagens normais e imagens com glaucoma), sendo este um método de otimização proposto recentemente e que por isso ainda não foi amplamente testado, apesar de ser indicado por seus autores como um método de ótimo desempenho.

## 2. REFERENCIAL TEÓRICO

### 2.1 Redes Neurais Convolucionais

Redes neurais artificiais (RNAs) são modelos de processamento de dados inspirados no funcionamento das redes neurais biológicas e lembram o cérebro humano na forma como funcionam. Elas são capazes de realizarem tarefas como reconhecer objetos, classificar imagens, aprender padrões, etc. De acordo com [15], RNAs podem ser definidas como um conjunto de unidades de processamento ligadas entre si por meio de uma grande quantidade de conexões. Assim como nas redes biológicas, as unidades de processamento das RNAs são os neurônios, elementos básicos de sua constituição e as conexões entre os neurônios são conhecidas por sinapses artificiais. De acordo com [2], para um computador uma imagem colorida, com 3 canais de cores, vermelho, verde e azul, com e 250 pixels de altura e largura

é interpretada como uma matriz de 187.500 números. Essa matriz é utilizada como entrada pelas RNAs ao processarem imagens; onde a intensidade de cada pixel é representada por esses números, que podem variar de 0 até 255. Cada neurônio de uma camada é conectado a todos os neurônios da camada anterior. Logo a quantidade de pesos para a rede ajustar apenas em uma única camada seria de 187.500.

Devido a sua arquitetura, as RNCs são capazes de realizar diversos tratamentos nos dados de entrada, que ajudam a rede a obter uma maior eficácia no tempo consumido pela etapa de treinamento, assim como uma maior acurácia na classificação dos dados. O funcionamento das RNCs é baseado no processo biológico de processamentos de dados visuais [19]. Os principais tipos de camada de uma RNC são convolução, pooling e classificação.

### 2.1.1 Camada de Convolução

De acordo com [2], uma camada de convolução tem como objetivo aprender de forma eficiente características importantes sobre a imagem. Para isso é utilizada a técnica de conectividade local, que também é utilizada pelo córtex cerebral, onde o campo receptivo do neurônio é chamado de filtro. A tarefa do filtro é processar a imagem ou a camada anterior, realizando a soma do produto de todos os valores de seu campo receptivo. Essa tarefa é executada em todas as posições de sua entrada.

O filtro recebe uma matriz com 3 dimensões, composta por altura, largura e profundidade, para realizar a tarefa, no qual a profundidade representa o número de canais da imagem ou camada anterior e a largura e a altura, o tamanho do campo receptivo. É possível detectar características independente de sua posição na imagem, graças ao fato que todos os neurônios formando o campo visual utilizam o mesmo filtro [2].

Ainda segundo [2], mapas de características armazenam os locais onde o filtro detectou uma característica importante na imagem. Eles são gerados pela convolução dos filtros nas entradas da rede. Os filtros localizados mais próximos da entrada da rede tem a tendência de guardar características mais simples como linhas e curvas. As próximas camadas que realizam o mesmo processo utilizarão esses mapas, porém os filtros passarão a buscar por características mais complexas como por exemplo formas geométricas. Até o final da rede, os filtros conterão representações mais próximas do objeto desejado.

### 2.1.2 Camada de Pooling

De acordo com [2] a camada de pooling tem a função de reduzir dimensionalmente cada mapa de característica, mantendo somente as informações mais relevantes. Essa técnica se refere a deslizar um filtro, sem que ele sobreponha regiões, com tamanho X por toda a área do mapa e aplicar uma função que irá comprimir a informação da região que será observada pelo filtro. A técnica mais utilizada é a que usa o valor máximo encontrado na região como resultado, conhecida como max pooling.

Segundo [2], a camada de pooling é a responsável por aumentar a robustez da rede em relação a ruídos. Como a camada de pooling escolhe os valores mais altos da região, pequenas variações e distorções prejudicam menos a acurácia da rede, o que beneficia a detecção em imagens de diferentes tamanhos. E aumenta a performance da rede ao reduzir a quantidade de parâmetros da mesma e assim reduzindo o sobreajuste, ou *overfitting*, que ocorre quando temos uma quantidade grande de parâmetros para uma quantidade inferior de dados. O que leva a rede a começar a decorar as entradas ao invés de aprender características gerais. A

rede com *overfitting* terá uma alta acurácia na etapa de treinamento, mas quando testada em entradas diferentes das utilizadas em sua fase de treino a acurácia será bastante inferior. A capacidade de generalização da rede é melhorada com a redução de parâmetros pela camada de pooling.

### 2.1.3 Camada de Classificação

A camada de classificação é a última camada da rede e é responsável por categorizar a entrada em uma das classes disponíveis. Geralmente essa camada é constituída por neurônios totalmente conectados, aqueles vistos em redes neurais simples, que interpretam os últimos mapas de característica da rede. Essa interpretação ocorre devido aos neurônios desta camada, estarem totalmente conectados com todos os mapas de características da camada anterior, sendo capaz de juntar essas informações para que uma visão geral da imagem possa ser obtida [2].

## 2.2 Arquitetura Inception V3

Em seu trabalho [17], afirma que embora o aumento do tamanho do modelo (isto é, aumentar o número de camadas da rede) e até do próprio custo computacional gerem ganhos de qualidade de uma rede neural em várias tarefas, a eficiência computacional e baixa quantidade de parâmetros ainda são fatores muito importantes para a aplicação de um modelo de rede neural em vários casos. A arquitetura Inception V3, proposta por [17], tem como diferencial a presença de módulos chamados inception. Segundo [9], esses módulos são os extratores de características convolucionais que tem como função aprender com poucos parâmetros representações ricas de informação. Tradicionalmente uma camada convolucional aprende filtros em um espaço 3D por meio das dimensões altura, largura e dimensão de canal. A idéia por trás do bloco Inception é facilitar o processo de mapeamento entre canais e correlações espaciais ao fatorar explicitamente a série de operações que examinam de forma independente essas correlações, ao contrário do modelo tradicional em que um único kernel é responsável por essas tarefas. Na Figura 2.1 temos uma ilustração da arquitetura Inception V3.

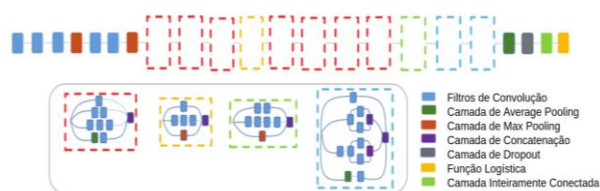


Figura 2.1. Arquitetura Inception V3. (Fonte: Adaptado de Ding et al., 2018)

## 2.3 Métodos de Otimização

Segundo [18], uma rede neural precisa que as variáveis de cada camada sejam alteradas de forma que esta funcione melhor no processo de classificação. Para esta tarefa é importante que o desempenho da rede seja continuamente medido através da comparação entre o resultado obtido e o resultado esperado. Sendo o objetivo da otimização minimizar o resultado da função cross-entropy (entropia cruzada), que é sempre positivo e se torna zero quando o resultado obtido é exatamente o mesmo resultado esperado. De acordo com [13], o algoritmo Gradiente Descendente é um dos métodos mais populares utilizados na tarefa de otimizar redes neurais e cada biblioteca de aprendizado de máquina estado da arte contém diversas implementações de versões otimizadas desse algoritmo. Esse método é uma maneira na qual se pode minimizar uma função objetivo  $J(\theta)$  pelos

parâmetros  $\theta \in \mathbb{R}^d$  de um modelo através da atualização desses parâmetros na direção oposta do gradiente da função objetivo  $\nabla_{\theta} J(\theta)$  com relação aos parâmetros. A taxa de aprendizado  $\eta$  determina o tamanho dos passos para atingir o mínimo local.

### 2.3.1 SGD

O método SGD, proposto por [16] executa uma atualização de parâmetros para cada amostra de treino  $x^{(i)}$  e cada rótulo  $y^{(i)}$ :  $\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})$ . SGD executa uma atualização de parâmetro por vez. O que torna ele geralmente um algoritmo muito rápido e pode ser usado para aprendizado online. Esse algoritmo realiza atualizações com uma alta variância que leva a função objetivo sofrer grandes flutuações [13]. Momentum é um método que ajuda a acelerar o método SGD a encontrar a direção correta de convergência mais rapidamente e apresentar menos oscilações. Isso ocorre através da adição de uma fração  $\gamma$  do vetor de atualização do passo anterior ao atual vetor de atualização [13].

### 2.3.2 ADAM

Adam, proposto por [7], é um método de otimização que calcula taxas de aprendizado adaptativas para cada parâmetro. Segundo [18], Adam mantém uma média exponencial decrescente dos gradientes anteriores  $m_t$ , sendo  $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$  e  $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ , onde  $m_t$  e  $v_t$  são estimações do primeiro momento, a média, e o segundo momento, a variação não centralizada, dos gradientes respectivamente,  $\beta_1$  e  $\beta_2$  são as taxas de decaimento e elas terminam em 1. A regra de atualização fica:  $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$ .

### 2.3.3 ADAGRAD

Adagrad [4] é proposto como um algoritmo de otimização que incorpora dinamicamente conhecimento sobre os dados observados em interações anteriores a fim de executar um aprendizado que permite ao algoritmo identificar características raras mas predizíveis sobre os dados trabalhados. Segundo [13], Adagrad é um algoritmo baseado na descida do gradiente que adapta a taxa de aprendizado  $\eta$  para os parâmetros, executando grandes atualizações para parâmetros infrequentes e atualizações menores para os parâmetros mais frequentes. Adagrad altera a taxa de aprendizado  $\eta$  a cada intervalo de tempo  $t$  para cada parâmetro  $\theta_i$  se baseando nos gradientes anteriores calculados para este parâmetro de acordo com  $\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$ , onde  $G_t$  é uma matriz diagonal onde cada elemento  $i$ ,  $i$  é a soma dos quadrados dos gradientes com respeito a  $\theta_i$  no intervalo  $t$ , enquanto  $\epsilon$  é um termo de suavização que evita divisão por zero e  $g_{t,i} = \nabla_{\theta} J(\theta_i)$ ,

sendo  $g_{t,i}$  o gradiente da função objetivo com respeito ao parâmetro  $\theta_i$  no intervalo  $i$ .

### 2.3.4 RMSProp

RMSProp, proposto por [6], é um método de taxa de aprendizagem adaptativa que combina a idéia de utilizar apenas o sinal do gradiente com a idéia de adaptar separadamente o tamanho de passo para cada peso. Segundo [18], a média de execução  $E[g^2]_t$  num tamanho de passo  $t$  depende da média anterior e do gradiente atual onde  $E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$  e  $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t$ . Esse algoritmo divide a taxa de aprendizado por uma média exponencial decrescente do quadrado dos gradientes (Ruder, 2016).

### 2.3.5 Adabound

Adabound, proposto por [10], é um algoritmo que tem como proposta ser tão rápido como o método Adam e tão eficiente como o método SGD. Em seu trabalho os autores demonstram que taxas de aprendizado muito grandes podem levar a um desempenho ruim do modelo. O algoritmo Adabound emprega limites dinâmicos nas taxas de aprendizado para alcançar uma transição gradual e suave de um método adaptativo para o SGD. Adabound é inspirado pela técnica gradient clipping (gradiente cortado), que usada na prática remove os gradientes maiores que um limiar estabelecido para evitar que os gradientes da função acumulem e fiquem muito grandes. A proposta do Adabound é a aplicação dessa técnica no método Adam.

## 3. METODOLOGIA

### 3.1 Base de Dados

Para o treinamento da RNC utilizada neste trabalho foram utilizadas imagens de cinco bancos de imagens públicos disponibilizados na internet que contém imagens obtidas de pacientes diagnosticados com glaucoma e imagens obtidas de pacientes diagnosticadas sem essa doença. Os bancos de imagens selecionados foram RIM-ONE R1, RIM-ONE R2 e RIM-ONE R3 [5], HRF [11] e Drishti [14]. Os dois primeiros bancos são os únicos que contém apenas imagens do nervo óptico. Sendo que o banco RIM-ONE R1 possui 40 imagens diagnosticadas com glaucoma e 118 normais, e o banco RIM-ONE R2 possui 200 imagens diagnosticadas com glaucoma e 255 normais. O banco RIM-ONE R3 contém imagens do olho diagnosticado do paciente e possui 74 imagens diagnosticadas com glaucoma e 85 normais. O banco HRF é composto por imagens do fundo do olho do paciente, e contém 15 imagens diagnosticadas com glaucoma, 15 imagens normais e 15 imagens com retinopatia diabética, que foram desconsideradas para este trabalho. O banco Drishti possui também imagens do fundo do olho do paciente, sendo 70 imagens diagnosticadas com glaucoma e 31 imagens normais. A Figura 3.1 apresenta exemplos de imagens encontradas em cada banco.

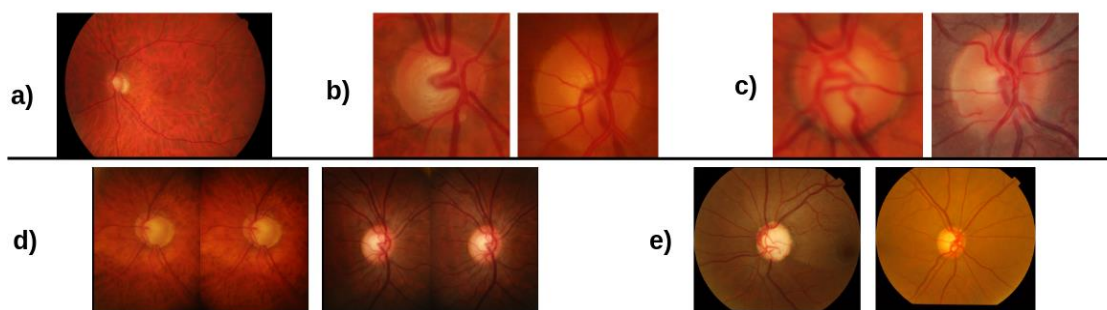


Figura 3.1. Algumas das imagens encontradas nos bancos utilizadas neste trabalho. a) HRF. b) RIM-ONE R1. c) RIM-ONE R2. d) RIM-ONE R3. e) Drishti-GS. Fonte: Autoria Própria.

### 3.2 Equipamento Utilizado

Neste trabalho todos os arquivos foram armazenados no Google Drive e para o treinamento da RNC foi utilizada a plataforma Google Colaboratory, também conhecida por Colab, um ambiente de desenvolvimento gratuito em nuvem baseado em Jupyter notebooks que conta com aceleração por GPU e integração fácil com o Google Drive. O trabalho de [1] verificou que o serviço da Google apresenta um ótimo desempenho em termos de processamento, em comparação com GPUs locais e apontou ainda que o Colab é um ótimo meio para compartilhar conhecimento entre pesquisadores.

O hardware utilizado nesta pesquisa opera com as seguintes especificações:

- Plataforma: Linux-4.14.79+-x86\_64-with-Ubuntu-18.04-bionic
- CPU: Intel(R) Xeon(R) CPU @ 2.30GHz
- GPU: Tesla K80 , 2496 CUDA cores, compute 3.7, 12GB GDDR5 VRAM
- Armazenamento: 358 GB

### 3.3 Ferramentas

Nesta pesquisa foi utilizado o framework Pytorch 1.1.0, idealizado para aprendizado de máquina, escrito em Python, e com suporte a API CUDA. As principais arquiteturas de redes neurais e os otimizadores mais conhecidos e utilizados pela comunidade de pesquisadores já estão implementados no framework Pytorch. O otimizador Adabound [10], proposto ainda em 2019 também já possui uma implementação desenvolvida por seu autor em Pytorch. Sendo esse o principal motivo da escolha deste Framework. Para o aumento de dados foi utilizada a biblioteca Keras, também escrito em Python, a qual possui funções dedicadas ao processamento de imagens.

### 3.4 Detecção do Glaucoma

Para o treinamento da RNC Inception V3 foram utilizadas apenas imagens do nervo óptico. Foi implementado um script para recortar apenas essa região das imagens dos bancos RIM-ONE R3, HRF e Drishti. Esse processo não foi necessário nos bancos RIM-ONE R1 e RIM-ONE R2, pois ambos contém apenas imagens dessa região. As coordenadas necessárias para o recorte foram obtidas manualmente com o auxílio do programa BBox-Label-Tool-master. Como o banco RIM-ONE R3 continha duas regiões do nervo óptico por imagem, optou-se por recortar ambas as regiões de cada imagem. A Figura 3.2 apresenta exemplos das imagens obtidas desses bancos após o recorte.

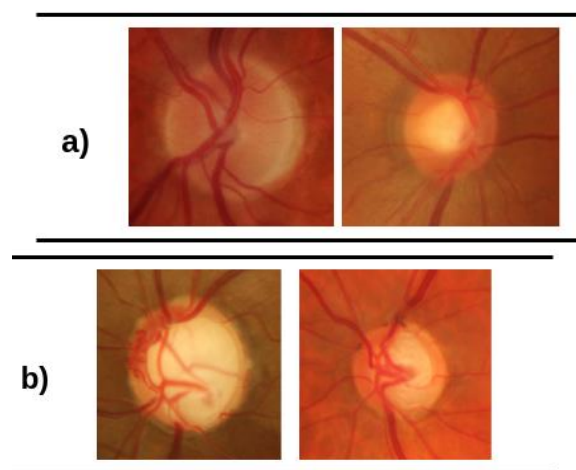


Figura 3.2. Exemplos de imagens após a aplicação do algoritmo de recorte a) imagens normais b) imagens com glaucoma. Fonte: Autoria Própria.

Após o recorte, as imagens dos bancos RIM-ONE e Drishti foram todas reunidas, e organizadas de acordo com sua classificação. Em seguida, para a geração de nova imagens a partir das originais foram aplicadas aleatoriamente distorções como variação do brilho, rotação e espelhamento horizontal e vertical das imagens, utilizando o script para aumento de dados desenvolvido utilizando as funções da Biblioteca Keras. Devido a existência de uma quantidade de imagens originais normais superior a quantidade de imagens originais com glaucoma, para balancear a quantidade de imagens com glaucoma e normais, a quantidade de imagens com glaucoma foi aumentada numa taxa de 10 novas imagens para cada imagem original, enquanto essa taxa foi de apenas 8 para as imagens que não possuíam glaucoma. Porém, para o banco de imagens HRF a taxa de 8 novas imagens para cada imagem original foi utilizada para ambas as classes de imagens. A taxa de aumento das imagens foi escolhida empiricamente e a base foi assim aumentada de 1050 para 9335 imagens. Como as imagens de alguns bancos estavam em formatos de compressão diferentes, após a aplicação do script de aumento de dados, todas as imagens foram convertidas para o formato JPG e em seguida foram embaralhadas. Finalmente 75% de todas as imagens obtidas foram separadas para o treinamento, 20% para validação, e 5% para teste, conforme a Tabela 1.

**Tabela 1. Divisão das imagens em Treino, Teste e Validação**

Diagnóstico	Treino	Validação	Teste	Total
glaucoma	3500	900	249	4649
normal	3500	900	286	4686

A rede neural convolucional Inception V3 utilizada para o treinamento foi implementada com as configurações padrões disponibilizadas pelo framework Pytorch. Antes de serem enviadas para o treinamento as imagens passaram por um pré-processamento disponibilizado pelo framework Pytorch, no qual as imagens foram redimensionadas para 299 pixels, que é o tamanho de imagem com o qual o modelo Inception trabalha, e em seguida foi realizada uma normalização com média [0.485, 0.456, 0.406] para os canais vermelho, azul e verde respectivamente e desvio padrão de [0.229, 0.224, 0.225] para respectivamente os mesmo canais de cores, em todas as imagens, procedimento necessário para que o treinamento da rede requiera menos processamento computacional. O valor para cada canal foi escolhido empiricamente.

### 3.5 Métricas de Avaliação

Durante cada treinamento foram obtidos os dados da acurácia de treino, acurácia de validação, erro de treino, e erro de validação obtidos em cada época para todos os modelos, sendo um modelo cada versão da rede Inception V3 que utiliza um dos cinco otimizadores selecionados para esse trabalho.

Para uma melhor avaliação dos modelos foi elaborada uma matriz de confusão, que é uma tabela que relaciona os resultados esperados com os resultados obtidos pelo modelo, após treinamento, ao classificar as imagens reservadas para teste. Sendo verdadeiros positivos (VP), os resultados positivos identificados corretamente; falso positivos (FP), os resultados negativos, que foram identificados como positivos; verdadeiros negativos (VN), os resultados negativos identificados corretamente e falso negativos (FN) os resultados positivos que foram identificados como negativos. Onde positivo se refere às imagens com glaucoma e negativo às imagens normais. Como exemplificado na Figura 3.3.

Original	Glaucoma	VP	FP
	Normal	FN	VN
		Glaucoma	Normal
		Predição	

**Figura 3.3. Matriz de confusão. Fonte: Autoria Própria.**

A partir da matriz de confusão podemos extrair diferentes métricas de avaliação, como:

- acurácia (ACC), dada pela equação 3.1, que mede a razão entre as imagens classificadas corretamente em relação ao número total de imagens;

$$ACC = \frac{VP + VN}{VP + FP + VN + FN} \quad (Eq. 3.1)$$

- sensibilidade (SEN), dada pela equação 3.2, que mede a razão entre as imagens com glaucoma identificadas corretamente em relação ao número total de imagens com glaucoma;

$$SEN = \frac{VP}{VP + FN} \quad (Eq. 3.2)$$

- especificidade (ESP), dada pela equação 3.3, que mede a razão entre as imagens normais corretamente identificadas, em relação ao número total de imagens;

$$ESP = \frac{VN}{VN + FP} \quad (Eq. 3.3)$$

- valor preditivo positivo (VPP), dado pela equação 3.4, que mede a razão entre as imagens corretamente identificadas com glaucoma em relação ao total de imagens identificadas com glaucoma;

$$VPP = \frac{VP}{VP + FP} \quad (Eq. 3.4)$$

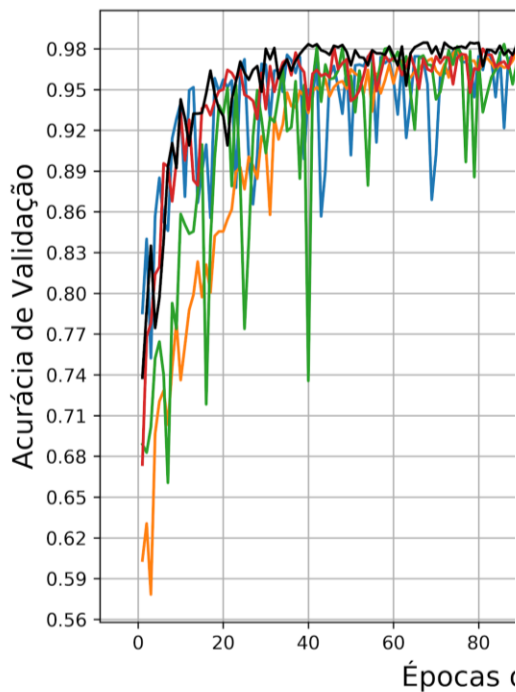
- valor preditivo negativo (VPN), dado pela equação 3.5, que mede a razão entre as imagens corretamente identificadas como normais em relação ao total de imagens identificadas como normais.

$$VPN = \frac{VN}{VN + FN} \quad (Eq. 3.5)$$

## 4. RESULTADOS

O treinamento de uma rede neural geralmente é dividido em épocas, procedimento em que a rede salva todos os pesos obtidos após processar todas as imagens separadas para essa etapa. Após cada etapa de treino geralmente a rede utiliza esses pesos para classificar todas as imagens disponíveis que não foram vistas pela mesma durante o treino. Chamamos essa etapa de validação, sendo o resultado dessa etapa comparado com aquele das épocas anteriores para que a rede possa escolher a melhor combinação de pesos e se atualizar a fim de diminuir a taxa de erro obtida na classificação [2].

Para esse trabalho 75% das imagens disponíveis foram utilizadas para a etapa de treinamento e 20% foram utilizadas para a etapa de validação. Assim como no trabalho de [18], cada sessão de treinamento rodou por 200 épocas. Os testes com cada otimizador foram todos realizados utilizando 0.001 como tamanho de passo inicial, valor escolhido empiricamente. Com exceção deste, todos os parâmetros padrões de cada otimizador foram mantidos.

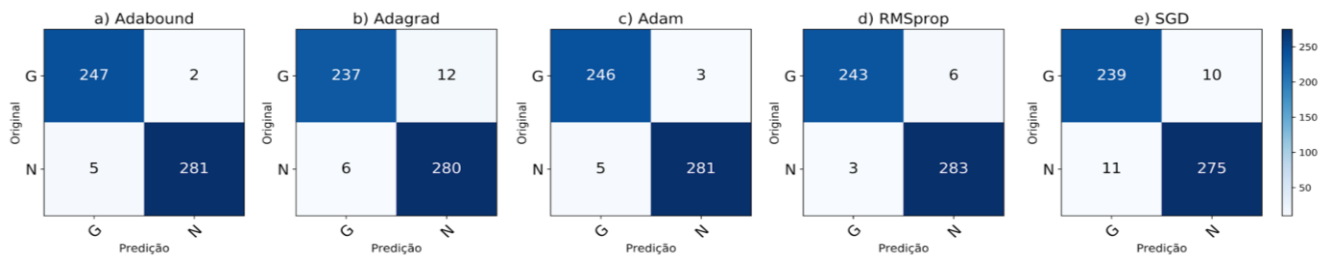


**Figura 4.1. Gráfico com o histórico da acurácia de validação ao longo do treinamento da rede para cada algoritmo. Fonte: Autoria Própria.**

Tem-se na Figura 4.1 o gráfico comparativo da acurácia obtida durante cada etapa de validação da rede para os cinco otimizadores selecionados para esse trabalho. E na Figura A.1 (APÊNDICE A) tem-se o gráfico comparativo do erro obtido também durante cada etapa de validação da rede.

Podem ser observados no gráfico da figura 4.1 que o otimizador Adabound foi o que obteve os melhores resultados durante a maioria das épocas. Sendo também aquele que demonstrou o desempenho mais estável, ao contrário dos métodos Adagrad e RMSprop que apresentaram desempenho ótimo em algumas épocas, porém em muitas outras tiveram um desempenho inferior

àqueles obtidos pelos próprios em épocas recentes anteriores. Após cada modelo da rede ser treinado por 200 épocas foi realizado um teste de classificação para cada modelo, no qual se utilizaram as 535 imagens que haviam sido separadas anteriormente para teste, 5% de todas as imagens segundo a tabela 1. Esse teste avaliou a capacidade da rede classificar corretamente as imagens com glaucoma e normais. Através dos resultados obtidos foi possível gerar uma matriz de confusão para cada modelo, ilustrados na Figura 4.2.

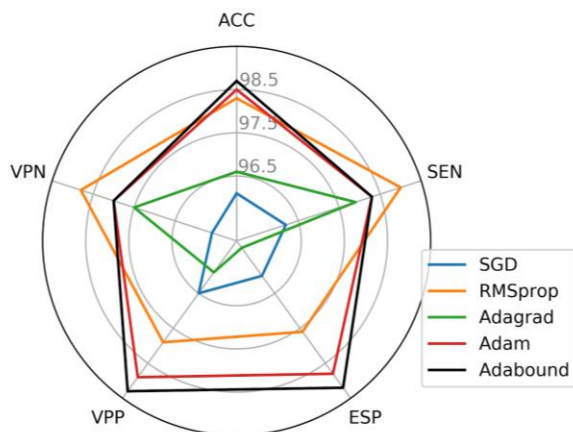


**Figura 4.2. Matriz de confusão gerada para cada otimizador. Fonte: Autoria Própria.**

Com a matriz de confusão foi possível extrair novamente métricas como a acurácia (ACC), sensibilidade (SEN), especificidade (ESP), valor preditivo positivo (VPP) e negativo (VPN) para cada otimizador na classificação de imagens com glaucoma e normais.

contém os valores obtidos por cada otimizador (Os melhores resultados para cada métrica estão em negrito).

No gráfico da Figura 4.3 está ilustrada a diferença de desempenho dos 5 otimizadores para cada uma dessas métricas e a Tabela 2



**Figura 4.3. Gráfico radar ilustrando o desempenho de cada otimizador após 200 épocas de treinamento. Fonte: Autoria Própria.**

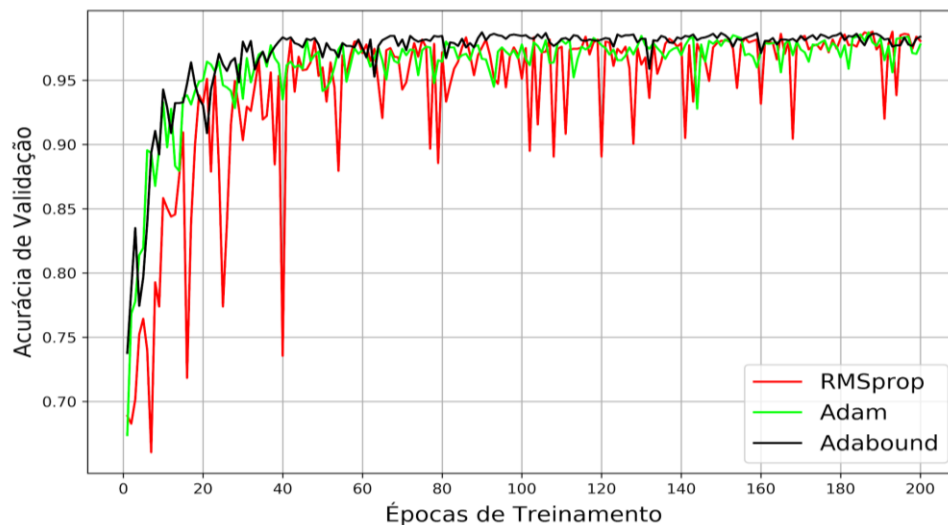
Pode-se notar ao examinar a Figura 4.3 e a Tabela 2 que o otimizador Adabound teve os melhores resultados nas métricas acurácia, especificidade e valor preditivo positivo, sendo o melhor em classificar corretamente imagens com glaucoma, superando o otimizador Adam, o mais popular recentemente dentre os métodos de otimização utilizados para treinamento de redes neurais, que também alcançou ótimos resultados em todos os avaliadores de desempenho. Adagrad e SGD, obtiveram os piores resultados em todas as métricas de avaliação, ao passo que o método RMSprop obteve o melhor desempenho na classificação correta de imagens normais.

**Tabela 2. Desempenho de cada otimizador após 200 épocas de treinamento**

	ACC	SEN	ESP	VPP	VPN
<b>SGD</b>	96,1%	96,2%	96,0%	96,5%	95,6%
<b>RMSprop</b>	98,3%	<b>99,0%</b>	97,6%	97,9%	<b>98,8%</b>
<b>Adagrad</b>	96,6%	97,9%	95,2%	95,9%	97,5%
<b>Adam</b>	98,5%	98,3%	98,8%	98,9%	98,0%
<b>Adabound</b>	<b>98,7%</b>	98,3%	<b>99,2%</b>	<b>99,3%</b>	98,0%

A Figura 4.4 apresenta novamente o gráfico comparativo de acurácia de validação. Porém optou-se agora por exibir apenas o desempenho dos otimizadores que obtiveram os melhores resultados nas métricas da tabela 2. Pode-se verificar com esse gráfico que apesar de não ter obtido o primeiro lugar em nenhuma das métricas de avaliação, o otimizador Adam pode ser

considerado mais confiável nesta tarefa do que o método RMSprop, que apresentou um desempenho inferior aos demais durante a maioria das épocas de treinamento. Esse gráfico também deixa claro o desempenho mais estável e ligeiramente superior do otimizador Adabound em relação aos demais, ao longo de todo o treinamento.



**Figura 4.4. Gráficos da Acurácia de Validação para os Otimizadores Adam, RMSprop e Adabound. Fonte: Autoria Própria.**

## 5. CONSIDERAÇÕES FINAIS

O propósito deste trabalho foi comparar a influência da escolha do otimizador na performance de uma rede neural convolucional, com o objetivo de identificar o método mais adequado à classificação de imagens. Esse trabalho teve como foco a detecção do glaucoma utilizando a arquitetura de rede Inception V3. Apesar de existir uma grande variedade de arquiteturas de redes neurais convolucionais disponíveis atualmente e uma enorme quantidade

de problemas que podem ser solucionados com RNCs, os resultados deste trabalho podem servir, não apenas como uma base para futuros algoritmos de detecção de glaucoma utilizando imagens, mas também como um pontapé inicial na construção de modelos de RNCs.

Quando se realiza o treinamento de uma rede neural geralmente a maior dificuldade se encontra na escolha correta dos parâmetros,

sendo muitas vezes necessário a realização de vários testes a fim de verificar com qual parâmetro a rede apresenta o melhor desempenho. A partir dos resultados obtidos por esse trabalho podemos verificar que a escolha do método de otimização tem um impacto bastante significativo no comportamento da rede.

Os métodos RMSprop e Adagrad demonstraram-se pouco confiáveis, uma vez que o desempenho apresentado pelos mesmos é bastante irregular durante o treinamento. Os métodos SGD com momentum e Adam, os mais utilizados recentemente como escolha de otimizador inicial no treinamento das atuais RNCs, se verificaram como métodos estáveis, apresentando quedas menos drásticas de desempenho ao longo do treinamento. Porém aquele que mais se destacou positivamente tanto em estabilidade como em acurácia foi o otimizador Adabound, o mais recentemente proposto dentre os métodos testados neste trabalho. O diferencial deste método está no fato de que, apesar de todos os otimizadores estudados neste trabalho trabalharem com taxas de aprendizado adaptativas, o método Adabound vai aumentando o valor deste parâmetro ao longo do treinamento.

Não se pode afirmar com os resultados obtidos neste trabalho qual o melhor método de otimização para treinamento de RNCs. Alguns métodos são mais adequados para lidar com alguns tipos de dados do que outros. Como citado por [7], Adam é um método direcionado ao treinamento de grandes bancos de dados, ao passo que Adagrad é mais indicado para lidar com gradientes esparsos e o método RMSprop lida melhor com objetivos não estacionários. Porém podemos concluir que o método Adabound tem grande potencial para problemas envolvendo classificação de imagens e é um método indicado quando se quer evitar constante configuração de parâmetros.

Trabalhos futuros podem ser realizados utilizando outras arquiteturas de RNCs, ou variações da rede Inception utilizando uma diferente escolha de parâmetros, assim a aplicação desses testes em outros bancos de dados. Há também a intenção de se desenvolver um aplicativo móvel que seja capaz de identificar imagens com glaucoma. Sendo o diagnóstico online de doenças um dos prováveis caminhos a serem trilhados num futuro próximo entre aprendizado de máquina e medicina.

## 6. REFERÊNCIAS

- [1] Carneiro, T., Medeiros Da Nobrega, R., Nepomuceno, T., Bian, G., De Albuquerque, V. and Filho, P. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating
- [2] Cerentini, A. (2018). Identificação do glaucoma em imagens do fundo do olho utilizando aprendizagem profunda.
- [3] Claro, M., Santos, L., Silva, W., AraUjo, F., Dalilia, A., & Santana, A. (2015). Automatic detection of glaucoma using disc optic segmentation and feature extraction. In 2015 Latin American Computing Conference (CLEI) (pp. 1-7). IEEE.
- [4] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul), 2121-2159.
- [5] Fumero, F., Alayón, S., Sanchez, J. L., Sigut, J., & Gonzalez-Hernandez, M. (2011, June). RIM-ONE: An open retinal image database for optic nerve evaluation. In 2011 24th international symposium on computer-based medical systems (CBMS) (pp. 1-6). IEEE.
- [6] Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14, 8.
- [7] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [8] Kingman, S. (2004). Glaucoma is second leading cause of blindness globally. *Bulletin of the World Health Organization*, 82, 887-888.
- [9] Leonardo, M. M., & Faria, F. A. (2017). Um Sistema de Reconhecimento de Espécies de Moscas-das-Frutas. *Revista Eletrônica de Iniciação Científica em Computação*, 17(2).
- [10] Luo, L., Xiong, Y., Liu, Y., & Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint arXiv:1902.09843.
- [11] Odstrcilik, J., Kolar, R., Budai, A., Hornegger, J., Jan, J., Gazarek, J., ... & Angelopoulou, E. (2013). Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database. *IET Image Processing*, 7(4), 373-383.
- [12] Quigley, H. A., & Broman, A. T. (2006). The number of people with glaucoma worldwide in 2010 and 2020. *British journal of ophthalmology*, 90(3), 262-267.
- [13] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
- [14] Sivaswamy, J., Krishnadas, S. R., Joshi, G. D., Jain, M., & Tabish, A. U. S. (2014, April). Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation. In 2014 IEEE 11th international symposium on biomedical imaging (ISBI) (pp. 53-56). IEEE.
- [15] Sousa, F. M. M. M. D. (2017). Uma Otimização do Perceptron Multicamadas Utilizando Aproximação Quadrática e Método de Ponto Proximal.
- [16] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147).
- [17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
- [18] Taqi, A. M., Awad, A., Al-Azzo, F., & Milanova, M. (2018). The impact of multi-optimizers and data augmentation on TensorFlow convolutional neural network performance. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 140-145). IEEE.
- [19] Vargas, A. C. G., Paes, A., & Vasconcelos, C. N. (2016, July). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In *Proceedings of the XXIX Conference on Graphics, Patterns and Images* (pp. 1-4).



## APÊNDICE A - GRÁFICO DE ERRO DE VALIDAÇÃO

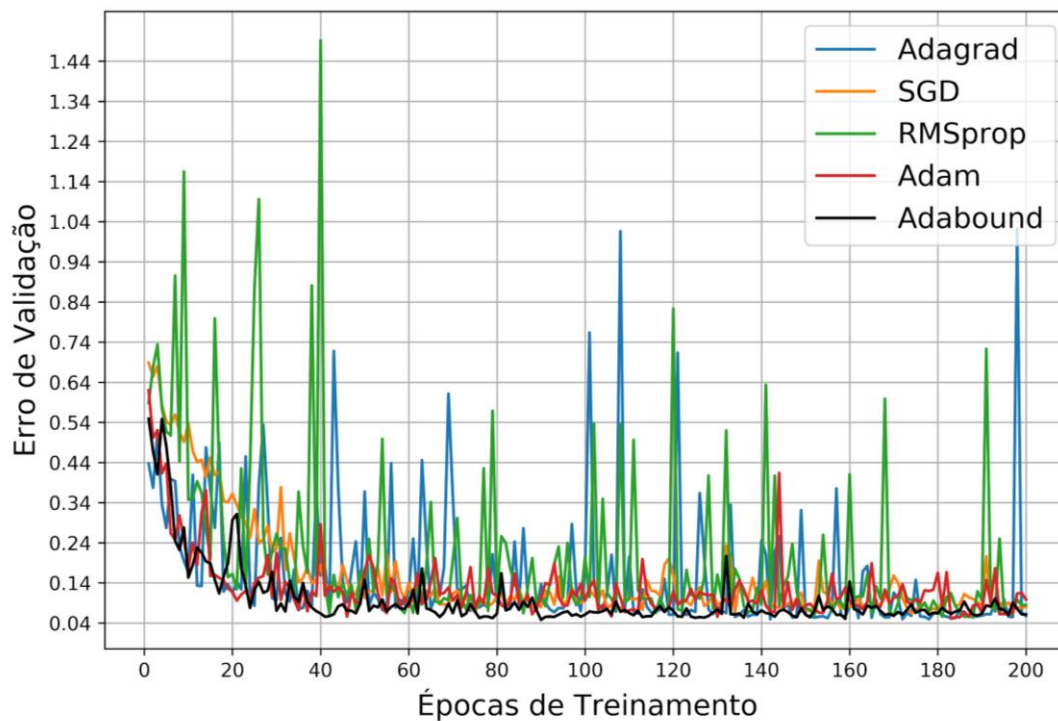


Figura A.1. Gráfico com o histórico do erro de validação ao longo do treinamento da rede para cada algoritmo. Fonte: Autoria Própria.