

Desenvolvimento de uma CNN para classificar o Nível de Retinopatia Diabética através do Exame de Fundo do Olho

Thiago Oliveira Lima – e10.thiago@gmail.com

Resumo—Este artigo apresenta alguns conceitos de Inteligência Artificial e algumas técnicas subáreas das mesmas, além de apresentar conceito sobre Retinopatia Diabética. Além disso, mostra resultados práticos da tentativa de desenvolver uma CNN para classificar o nível de Retinopatia Diabética de um paciente baseado em imagens do Exame do Fundo do Olho.

Palavras-Chave—*Inteligência Artificial, Deep Learning, Redes Neural Convolucional, Diabetes Mellito, Retinopatia Diabética.*

Abstract— This article presents some concepts of Artificial Intelligence and some subarea techniques, and presents the concept of Diabetic Retinopathy. It also shows the practical results of trying to develop a CNN to classify a patient's level of diabetic retinopathy based on fundus images.

Keywords—*Artificial Intelligence, Deep Learning, Convolutional Neural Networks, Diabetes Mellitus, Diabetic Retinopathy.*

I. INTRODUÇÃO

A importância de manter a integridade funcional da visão tem sido cada vez mais reconhecida como requisito para uma boa qualidade de vida, sobretudo durante a terceira idade, quando é comum a alteração de diversos aspectos da visão, seja por condições fisiológicas ou patológicas (West et al, 2002).

A Retinopatia Diabética é considerada uma das complicações microvasculares mais importantes do Diabetes Mellitus, além disso, é uma das principais causas de cegueira no ocidente. Pesquisas indicam que, no Brasil, 50% dos pacientes diagnosticados com Diabetes Mellitus são afetados pela Retinopatia Diabética. Especialistas recomendam que todo paciente com Diabetes Mellitus consulte o seu oftalmologista uma vez por ano, devido ao fato de que os estágios iniciais da Retinopatia Diabética não apresentam sintomas e, além disso, com um diagnóstico precoce, existem tratamentos capazes de impedir a progressão da Retinopatia Diabética e até a recuperação de parte da visão [2][6].

A Retinopatia Diabética pode ser diagnosticada através de exames oftalmológicos, que incluem desde o histórico do paciente, fotografia do fundo do olho e alguns outros testes que avaliam a saúde da retina. Um dos exames especialmente útil é a Angiografia fluoresceínica, ela é utilizada para avaliar clinicamente a presença e o grau/tipo da Retinopatia Diabética [6].

A Inteligência Artificial é um campo da computação que visa imitar alguns aspectos humanos, como o processo de pensamento, a capacidade de aprendizagem e o

armazenamento de conhecimento. Devido a isso, a Inteligência Artificial tem sido muito utilizada na área de análise de imagens e previsões de forma a auxiliar diversas áreas da vida humana.

Com a alta velocidade dos avanços tecnológicos, a grande quantidade de dados que se tem gerado e processado nos últimos anos, algumas técnicas de Inteligência Artificial que já existiam vem ganhando cada vez mais relevância. Um exemplo disso é o Deep Learning ou Deep Neural Network, que consiste em uma subárea do Aprendizado de Máquina baseado em Redes Neurais Artificiais que tem sido muito utilizado hoje em dia e se mostrado muito eficiente para resolver problemas complexos, como por exemplo os de processamento de imagens [7].

Esse trabalho tem o objetivo de utilizar técnicas de Deep Learning, mais especificamente uma Rede Neural Convolucional para auxiliar nos diagnósticos da Retinopatia Diabética assim como avaliar o grau/tipo de Retinopatia Diabética presente naquele indivíduo através de imagens do Exame de Fundo de Olho.

II. FUNDAMENTOS TEÓRICOS

A. Retinopatia Diabética

A Retinopatia Diabética (RD) é uma complicação do Diabetes Mellitus na qual os capilares responsáveis pela nutrição da retina são afetados, fazendo com que a visão do paciente fique comprometida.

Como o Diabetes Mellitus é uma síndrome que afeta uma porção considerável da população e a visão é um sentido essencial que gostaríamos de preservar, é comum que pacientes afetados por Diabetes Mellitus façam exames oftalmológicos com certa periodicidade para evitar o desenvolvimento da Retinopatia Diabética.

A Retinopatia Diabética é classificada em: Retinopatia Diabética Não Proliferada (RDNP) e Retinopatia Diabética Proliferada (RDP). Sendo que a RDNP pode ser dividida em leve, moderada, grave e muito grave e a RDP pode ser inicial, de alto risco e avançado. Em todos os estágios da Retinopatia Diabética pode estar presente o edema macular.

Uma das principais formas de detecção da RD é a Angiografia fluoresceínica ou Angiofluoresceinografia que consiste em um exame onde é injetado na veia do braço um “corante” chamado fluoresceína sódica e após alguns segundos já é possível observar essa substância circulando no interior dos olhos. Após esse processo e da dilatação da

pupila, são retiradas algumas fotos do fundo do olho que são analisadas por um especialista.

B. Inteligência Artificial

A Inteligência Artificial (IA) é um campo de estudo da computação surgido na década de 50 que tem como objetivo principal estudar e criar sistemas com a capacidade de se comportar de forma semelhante aos seres humanos, ou seja, que seja capaz de aprender, entender, lembrar, e reconhecer coisas no mundo real [5].

Tecnologias com princípios de IA vem a cada dia ganhando mais força e se popularizando em diversas áreas da vida humana. Podemos dizer que alguns dos principais motivos que levou a essa crescente popularização da IA nos últimos anos são: a grande conectividade do mundo atual, a grande quantidade de dados que vem sendo disponibilizados, avanços em algoritmos de IA e principalmente a grande quantidade e redução de preço da infraestrutura necessária para o processamento [5].

A capacidade de processamento é um dos pontos importantes para IA pois está relacionado diretamente com a velocidade de tomada de decisão. Quanto mais as tecnologias de processamento avançam e se tornam mais baratas, melhores ficam os sistemas de IA devido ao fato que eles conseguem resolver problemas mais complexos e tomar decisões em tempo hábil. Além disso, vivemos em uma época em que a quantidade de dados que geramos e armazenamos é gigante e com uma grande variação de natureza de informação e formas de captação, essa grande quantidade de dados produzidos e informações disponíveis também é um ponto chave para o avanço da IA, principalmente em uma das suas subáreas, o Aprendizado de Máquina [5].

C. Aprendizado de Máquina

O Aprendizado de Máquina (AM), do inglês, Machine Learning é uma subárea da IA que pesquisa métodos computacionais relacionados à aquisição de novos conhecimentos, novas habilidades e novas formas de organizar o conhecimento já existente (Russel e Norvig, 2013).

Basicamente, o AM desenvolve técnicas de aprendizado e também sistemas computacionais com a capacidade de adquirir conhecimento de forma automática. Um sistema de aprendizado pode ser considerado um programa que consegue tomar decisões diante de um problema apresentado, baseado no conhecimento acumulado por soluções corretas de problemas anteriores. Difere-se do termo IA pois é mais focado na capacidade humana de aprender, enquanto a IA é mais abrangente [5].

É possível criar soluções de AM utilizando várias técnicas diferentes, todas essas técnicas consistem em procurar padrões nos dados através de algoritmos diferentes, por isso é muito comum que AM seja confundido com o nome da técnica utilizada para resolver um problema específico de aprendizado. Algumas das técnicas que podem ser utilizadas são: Rede Neural Artificial (RNA) e Deep Learning (DL).

D. Redes Neurais Artificiais

A Rede Neural Artificial (RNA), do inglês, Artificial Neural Network é um modelo matemático baseado na

arquitetura do cérebro biológico, logo o modelo possui representações de neurônios, sinapses e das ligações químicas responsáveis pela propagação das informações entre os neurônios. Sua estrutura é organizada em camadas (entrada, saída e ocultas), onde cada camada possui pelo menos um neurônio. As saídas dos neurônios são ponderadas pelos pesos (sinapses), combinadas através da função de ativação e passadas como entrada para os neurônios da camada seguinte. Esse processo repete-se até a camada de saída, onde os dados serão representados como o resultado da rede. As camadas que não são nem de entrada e nem de saída são denominadas de camadas ocultas [3].

Nas RNAs, cada neurônio possui um peso (representação das sinapses) que o liga aos neurônios da próxima camada. Os pesos são ajustados através de um processo chamado de treinamento, que consiste na apresentação de vários dados e respostas esperadas à rede, a fim de que ela processe e compare sua resposta com a esperada, através disso ela pode “aprender” os melhores pesos para cada neurônio a fim de que erre menos quando apresentado a um dado nunca visto antes[3].

A apresentação de dados nas RNAs consiste na passagem dos valores aos neurônios de entrada. Esses valores são multiplicados pelos pesos e passados para os respectivos neurônios da próxima camada, com os quais existe uma conexão direta. Este tipo de rede normalmente é unidirecional, ou seja, o fluxo da informação segue apenas uma direção e não possui ciclo interno. Por causa dessa característica, esse tipo de rede é chamado de feed-forward [3].

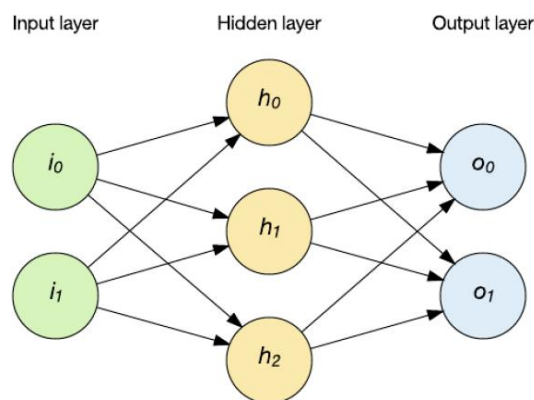


Figura 1 - Redes Neurais Feed-forward [3]

Quando uma RNA possui muitas camadas ocultas na sua arquitetura, normalmente ela é chamada de Rede Neural Profunda ou Deep Learning.

E. Rede Neural Convolucional

A Rede Neural Convolucional, do inglês, Convolutional Neural Network (CNN) é um tipo de RNA que possui muitas camadas ocultas e, portanto, é uma técnica de Deep Learning. Esse tipo de rede tem sido muito utilizado para processamento e análise de imagens digitais.

As CNNs possuem arquitetura semelhante às RNAs tradicionais, com a diferença que, nas camadas iniciais, os pesos aos quais os neurônios são ponderados representam filtros que serão aplicados sobre uma imagem. Através disso,

as camadas iniciais conseguem identificar elementos básicos de uma imagem, como por exemplo: linhas, formas, cores, textura. A utilização de mais camadas ocultas resulta no aprendizado de outros tipos de características mais complexas, até algumas não esperadas, mas que a própria rede percebeu que era importante para a resolução do problema.

Ao fim de uma CNN, normalmente existe uma RNA totalmente conectada, essa etapa da CNN é responsável por juntar todas as características reconhecidas pelas camadas convolucionais e gerar uma resposta para o problema, seja ele para classificação ou predição.

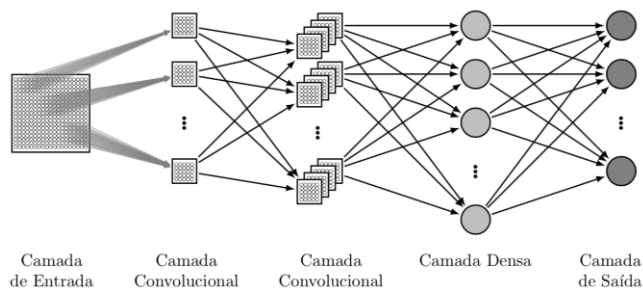


Figura 2 - Exemplo de Arquitetura CNN

III. METODOLOGIA

A. Definição do Problema e Objetivos

O problema que esse trabalho tem como objetivo solucionar é a classificação do nível de Retinopatia Diabética quando analisada a imagem do fundo do olho. Essa análise é feita especificamente pelo oftalmologista e é fundamental para determinar o tratamento que deve ser seguido para que a Retinopatia Diabética não continue se desenvolvendo e, em alguns casos, podendo melhorar a visão do paciente.

O objetivo desse trabalho é desenvolver uma CNN que seja capaz de classificar corretamente qual o nível de Retinopatia Diabética quando uma nova imagem de fundo de olho for apresentada para a mesma, fazendo com que o processo de diagnosticar um paciente se torne mais eficiente. As classes desse problema são:

Níveis de RD	Classe do Problema
Sem RD	0
Suave	1
Moderado	2
Severa	3
RD Proliferada	4

Tabela 1 - Níveis de RD

B. Conjunto de Dados e Ferramentas Tecnológicas

Os dados que serão utilizados nesse trabalho foram obtidos através da plataforma Kaggle, e foi fornecido para duas competições, a APTOS 2019 e APTOS 2015. Se tratam de imagens de exames do fundo do olho (**Figura 3,4**).

Dos dados de 2019 foram utilizados as 3662 que são fornecidas como imagens de treino pois elas são rotuladas e

também um arquivo CSV onde são encontrados o nome da imagem e o rótulo da classe a que ela pertence.



Figura 3 - APTOS 2019

Dos dados de 2015 foram utilizados 38.788 imagens do exame do fundo do olho rotuladas, e um arquivo CSV onde possui todos os nomes das imagens e seus rótulos.

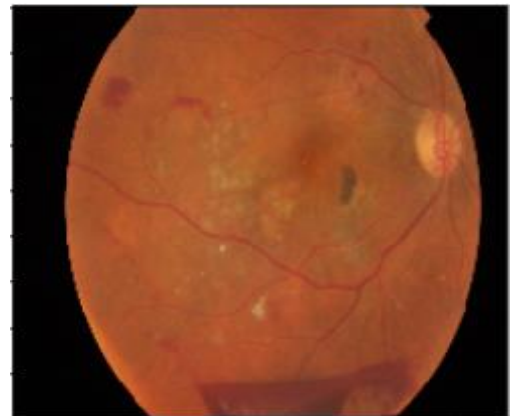


Figura 4 - APTOS 2015

A linguagem Python será a utilizada para o desenvolvimento dos algoritmos da CNN, pois a mesma possui bibliotecas voltadas para Deep Learning que facilitam e tornam o desenvolvimento mais prático, além disso será utilizado também a plataforma de desenvolvimento Google Colaboratory.

C. Desenvolvimento

Primeiramente, foi desenvolvido no Google Colaboratory um código que permitia fazer o download dos dados da plataforma Kaggle na máquina virtual disponibilizada pelo Google. Além disso, foi implementado uma rotina que fazia a leitura e algumas adequações das imagens (padronização de tamanho e quantidade de canais desejadas) para a utilização na CNN, para isso foi utilizado as bibliotecas CV2 e CSV do Python. Após isso, foi feita a normalização e separação das imagens em conjunto de treinamento e conjunto de validação.

Posteriormente, uma rotina foi criada pra substituir o download do conjunto de dados, pois ele era necessário a todo momento que se fosse reutilizar o código, através dessa rotina foi gerado um arquivo no formato PKL com os dados de entrada, esse arquivo foi salvo no Google Drive, para que

todo os momentos que fosse utiliza-los bastava acessar localmente.

Inicialmente o treinamento foi baseado apenas nos dados da APTOS 2019 e os resultados eram bons em relação a perda e acurácia, porém, não mostravam o desempenho de uma rede inteligente quando a matriz confusão era analisada. Posteriormente, foi percebido que o conjunto de imagens desse dataset possuía um desequilíbrio muito grande, com isso surgiu a ideia de tentar contorná-lo através de técnicas da biblioteca scikit-learn ou encontrar um outro dataset que possuía uma semelhança para que pudesse juntá-los e resolver o desequilíbrio, com isso foi utilizado os dados da APTOS 2015.

Quando os dois conjuntos de dados foram mesclados, foi percebido que o desequilíbrio continuava muito grande, porém, a quantidade de imagens da classe com menos exemplos era de 1003. Portanto foi decidido limitar todas as classes em 1003 exemplos de modo que se pudesse fazer uma tentativa de aprendizado com as classes da forma mais equilibrada possível.

Tiveram tentativas de treinamento com vários tipos de redes diferentes, inclusive foi aplicado técnicas de Transfer Learning com as redes VGG e ResNet50, também se aplicou imagens de 3 tamanhos diferentes (640x480, 224x224, 256x256) e com cores em RGB e níveis de cinza. Alguns resultados e discussões podem ser apreciados no próximo capítulo.

IV. RESULTADOS E DISCUSSÕES

Ao treinar o primeiro classificador criado, pode-se perceber que o mesmo possui um resultado relativamente bom para o problema. Os resultados foram:

Loss	Acc	Val_loss	Val_acc
0.3571	0.9141	0.5216	0.8558

Porém, foi verificado que na predição do conjunto de validação, o resultado observado na matriz de confusão era que o classificador estava “chutando” apenas duas classes.

Ao analisar o conjunto de dados, foi constatado que o mesmo possuía um desbalanceamento na quantidade de imagens para cada classe do problema, como pode ser constatado na **Tabela 2**, e que o classificador apenas chutava nas classes que possui mais exemplos (0 e 2).

Classe do Problema	Qtd. De Exemplos
0	1805
1	370
2	999
3	193
4	295

Tabela 2 - Qtd. Exemplos em cada classe

Com o intuito de resolver esse problema, foi utilizado um parâmetro do método fit do sklearn, chamado `compute_class_weight`, a introdução desse parâmetro no método fit, representa um peso para cada classe da imagem que é calculado de acordo com a proporção de quantidade de exemplos, na prática, funciona como um mecanismo de atenção ou punição da rede, serve para adequar o treinamento a um problema desbalanceado.

Utilizando o mesmo classificador anterior, com a mesma quantidade de épocas de treinamento (50), porém com o parâmetro `compute_class_weight`, percebemos uma leve piora no desempenho da rede, os resultados são:

Loss	Acc	Val_loss	Val_acc
0.5171	0.8241	1.0216	0.7558

Quando aplicada a predição e feita a matriz de confusão, percebemos que o classificador passa a prever uma classe a mais que a anteriormente (**Figura 5**). Mesmo com a melhora observada, isso não é suficiente, pois as classes do problema com maior gravidade são a 3 e a 4, nesses casos o classificador criado simplesmente não conseguia prever.

[343	6	2	0	0]
[12	23	33	0	0]
[9	17	187	0	0]
[2	2	32	0	0]
[3	7	55	0	0]]

Figura 5 - Matriz Confusão Simples

Ainda na tentativa de resolver esse problema, foi feita uma mescla com as imagens das competições APTOS 2015 e APTOS 2019, a nova quantidade total de imagens obtidas pode ser observada na **Tabela 3**.

Classe do Problema	Qtd. De Exemplos
0	25810
1	2443
2	5292
3	873
4	708

Tabela 3 - Qtd. Exemplos em cada classe (após mescla)

Nesse caso, a decisão tomada foi de juntar os dois datasets e separar aleatoriamente as imagens de forma que a quantidade de exemplos de cada classe fosse igual. Com isso, foi obtido 1003 exemplos de cada classe.

Retomando o treinamento com o dataset balanceado, várias redes diferentes foram submetidas a esses dados, os resultados não eram satisfatórios em relação ao aprendizado ou então a rede sofria de Overfitting. Após um tempo de dedicação pra tentar resolver esse tipo de problema (alguns resultados podem ser vistos no **Anexo 1**), foi percebido que o método inicial criação das redes foi equivocado.

A forma mais adequada de criação de uma CNN pra resolver qualquer problema consiste na criação de camadas convolucionais (Conv2d + MaxPooling) utilizando filtros de tamanho adequados ao tamanho das suas imagens de entrada, além disso, utilização de uma camada Flatten para fazer a redução da dimensionalidade e conectar as camadas convolucionais com as camadas totalmente conectadas, para que no fim a resposta seja qual a classe que se tem mais probabilidade de uma certa imagem ser.

Caso o desenvolvimento de uma rede criada da forma descrita acima apresente um comportamento de Overfitting, é acrescentado formas de regularização (BatchNormalization, Dropout, Reg. L1, Reg.L2, Data Augmentation) para tentar resolver o problema.

Porém, as redes desenvolvidas para tentar resolver o problema desse artigo já eram constituídas de conjuntos de camadas convolucionais, BatchNormalization, camadas densas e dropout.

A solução mais prática foi retomar o desenvolvimento do zero com o primeiro classificador criado e retirando todas as camadas de regularização existentes. Os resultados obtidos foram interessantes (**Figura 6,7,8 e Tabela 4**) o resultado obtido depois desse procedimento foi o melhor de todas as tentativas posteriores e anteriores.

Tentando evoluir a rede, foi utilizado o conjunto de dados balanceado para fazer alguns testes, os mesmos não apresentaram um desempenho bom. Além disso, foi aplicado Data Augmentation, esse processo não mostrou uma melhora e sim um resultado ligeiramente próximo, porém a matriz confusão não estava interessante, além disso, não foi possível explorar muito devido à demora no tempo de treinamento. Redes com tamanhos maiores e menores e com outros tipos de função de perda também não mostraram resultados melhores.

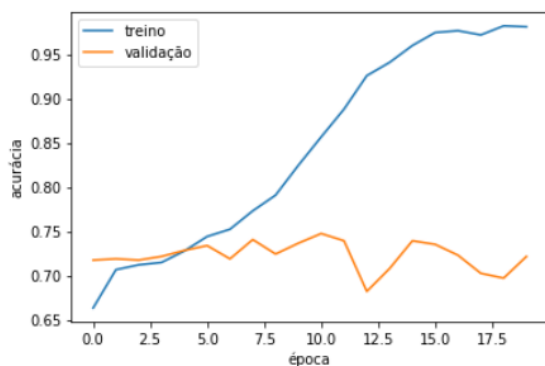


Figura 6 - Acurácia x Época (Melhor Resultado)

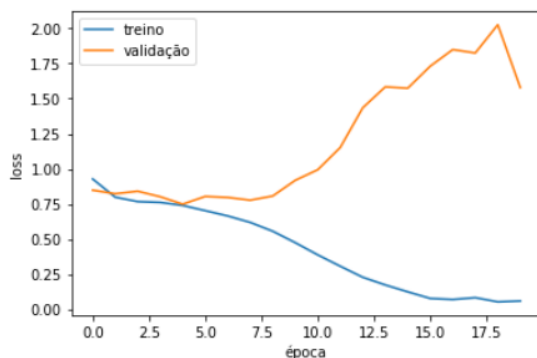


Figura 7 - Loss x Época (Melhor Resultado)

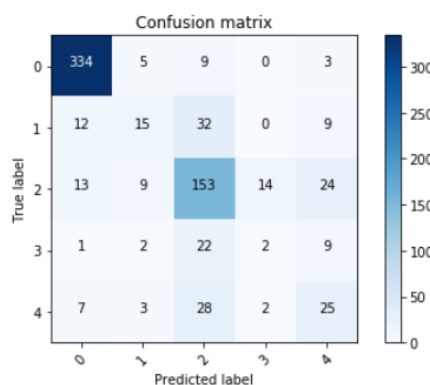


Figura 8 - Matriz Confusão (Melhor Resultado)

Tamanho Imagem		480x640x1
Conv2d	Camadas	4
	Neurônios	8,16,32,64
Densa	Camadas	3
	Neurônios	32,16,5
Épocas		20
Otimizador		Adam
Comentários		
Voltei para o dataset desbalanceado e utilizei o parâmetro <code>compute_class_weight</code> . Retirei todas camadas de BatchNormalization e Dropout. Todas funções de ativação Relu. Stride da última camada Convolutacional = (2,2) e as outras (1,1), o <code>kernel_size</code> utilizado foi (2,2).		

Tabela 4 - Informações Rede (Melhor Resultado)

V. CONCLUSÕES

Através desse artigo pode-se perceber o crescente aumento das mais diversas formas de Inteligência Artificial e sua importância para os vários âmbitos da vida humana, como por exemplo a medicina. Além disso, podemos concluir que o avanço em trabalho como o de classificar os níveis de Retinopatia Diabética possui grandes desafios no sentido de tornar um problema cada vez mais eficiente. Com a superação desses desafios teremos grandes avanços e uma eficiência muito boa no diagnóstico de um problema que é consequência de uma doença muito comum na humanidade.

Além disso, podemos perceber que uma análise superficial ou não variada dos resultados que uma rede proporciona pode levar a conclusões equivocadas, por exemplo, uma acurácia de 91% no conjunto de validação para o conjunto de dados desse artigo, não necessariamente seria um resultado bom devido a sua característica de desbalanceamento.

Outra ressalva importante é que o desenvolvimento de uma inteligência artificial aplicando todos os conceitos, sem o embasamento teórico da arquitetura proposta e de como funciona os termos teóricos para resolver possíveis problemas encontrados, muitas vezes não levará a um bom resultado.

VI. ANEXOS

Tamanho Imagem		480x640x3
Conv2d	Camadas	9
	Neurônios	16,32,32,64,64,128,128,256,256
Densa	Camadas	4
	Neurônios	256,128,32,5
Épocas		100
Otimizador		Nadam
Comentários		
Os valores de dropout foram alterados para 0.2. Rede apresentou overfitting (acc = 0.9840 , loss = 0.0462, acc_val=0.2901, loss_val = 5.2865). Mudanças no otimizador e função de ativação não mostrou melhoras.		

Tamanho Imagem		480x640x3
Conv2d	Camadas	7
	Neurônios	16,32,32,64,64,128,128

Densa	Camadas Neurônios	
		128,32,5
Épocas		50
Otimizador		Nadam
Comentários		
Começou a se fazer uma redução da rede para tentar encontrar a menor rede que desse Overfitting. Rede apresentou overfitting (loss: 0.0600 - acc: 0.9791 - val_loss: 4.7500 - val_acc: 0.2652). Mudanças no otimizador e função de ativação não mostrou melhoras.		

Tamanho Imagem		480x640x1
Conv2d	Camadas Neurônios	3
		8,16,32
Densa	Camadas Neurônios	3
		32,16,5
Épocas		50
Otimizador		Nadam
Comentários		
Foi constatado que o método de criação inicial estava equivocado. Já fui construindo a rede com camadas de regularização. A partir daqui comecei uma outra abordagem, onde tirei todas as camadas de regularização presente na rede. Voltei a utilizar a função de ativação Relu nas camadas densas. Apresentou Overfitting e nenhuma mudança alterou o estado de Overfitting (Figura 9,10)		

REFERÊNCIAS

- [1] M. Gualtieri, *Avaliação funcional da visão de pacientes diabéticos em estado de pré e pós retinopatia diabética*. 2009. 139f. Tese de Doutorado – Universidade de São Paulo, São Paulo, 2009.
- [2] M. C. Boelter, M. J. de Azevedo, J. L. Gross, J. Lavinsky. Fatores de risco para retinopatia de diabética. *Arq Bras Oftalmol* 2003;66:239-47.
- [3] L. C. Vieira, Redes Neurais Recorrentes aplicadas a predição de tráfego de redes de computadores. Universidade Federal do Espírito Santo.
- [4] A. V. Braga, A. F. Lins, L. S. Soares, L. G. Fleury, J. C. Carvalho, R. S. do Prado. *Inteligência Artificial na Medicina*.
- [5] C. Bigonha, *Inteligência Artificial e ética. Panorama setorial da internet, Ano 10, Nº 2, 1-20, Out 2018*.
- [6] <https://saude.novartis.com.br/doencas-da-visao/diagnostico-da-retinopatia-diabetica/>
- [7] <https://medium.com/data-hackers/deep-learning-do-conceito-%C3%A0s-aplica%C3%A7%C3%B5es-e8e91a7c7eaf>
- [8] P. Harry, C. Frans, B. Deborah, H. Simon, Z. Yalin. Convolutional Neural Networks for Diabetic Retinopathy. 2016. 200f. *Procedia Computer Science*.
- [9] A. Michael, L. Yiyue, E. Ali, C. Warren, A. Ryan, F. James, N. Meindert. Improved Automated Detection of Diabetic Retinopathy on a Publicly Available Dataset Through Integration of Deep Learning.
- [10] G. Varun, P. Lily, C. Marc, S. Martin. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*.

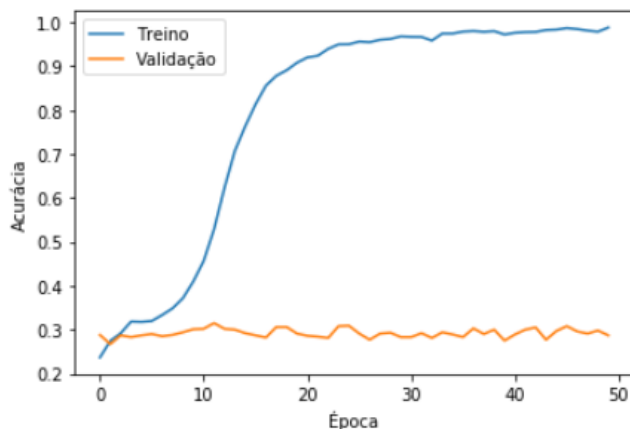


Figura 9 - Acurácia x Época

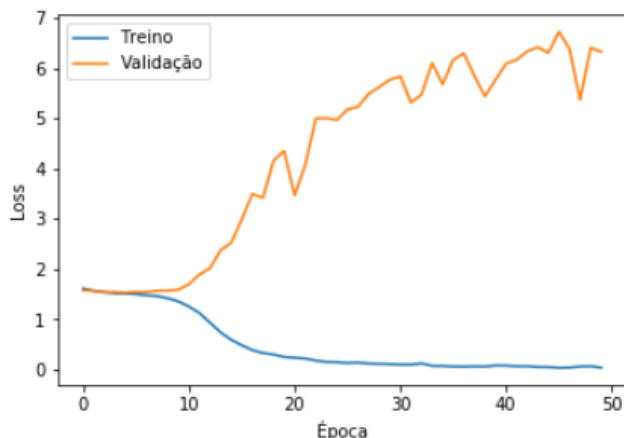


Figura 10 - Loss x Época