

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Análise oncológica de mamografias por machine learning

Guilherme Freitas de Araujo

Proposta Projeto Final II

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Engenharia da Computação

Rio de Janeiro, Junho de 2019



Guilherme Freitas de Araujo

Análise oncológica de mamografias por machine learning

Proposta de Projeto de Conclusão de Curso, apresentado ao curso de **Engenharia de Computação** da PUC-Rio como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientadora:

Prof. Marley Maria B. R. Vellasco

Coorientador:

Prof. Italo de Oliveira Matias

Rio de Janeiro

Junho de 2019

Agradecimentos

À minha orientadora Marley Vellasco pelos ensinamentos, apoio e interesse ao longo do desenvolvimento desse projeto.

Ao meu co-orientador Italo Matias pela ajuda, acompanhamento e auxílio na busca de soluções, material e recursos necessários para o desenvolvimento do projeto.

Aos meus amigos e familiares que me apoiaram durante toda minha graduação.

Resumo

Araujo, Guilherme; Vellasco, Marley; Matias, Italo. Análise oncológica de mamografias por machine learning. Rio de Janeiro, 2019. 33p. Proposta de Projeto de Conclusão de Curso - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho tem como objetivo desenvolver uma aplicação utilizando machine learning para análise de tumores em imagens mamográficas.

Palavras-chave

Machine learning; Inteligência artificial; Aprendizagem profunda;

Abstract

Araujo, Guilherme; Vellasco, Marley; Matias, Italo. Analysis of mammograms by machine learning. Rio de Janeiro, 2019. 33p. Capstone Project Report – Department of Informatics. Pontifical Catholic University of Rio de Janeiro.

This Project aims the development of an application using machine learning for the analysis of tumors in mammographic images.

Keywords

Machine Learning; Artificial Intelligence; Deep Learning;

Sumário

1	Introdução	7
2	Situação Atual	8
3	Propostas e Objetivos do trabalho	9
4	Atividades Realizadas	10
4.1	Estudos preliminares	10
4.2	Estudos conceituais e de tecnologia	10
4.3	Testes e protótipos	17
4.4	Método	18
5	Projeto e especificação do problema	19
5.1	Base de dados.....	19
5.2	Aplicação.....	21
5.3	Ferramentas auxiliares	22
6	Implementação e avaliação	23
6.1	Teste da arquitetura.....	24
6.2	Teste de epochs	28
6.3	Teste de batch size.....	28
6.4	Testes finais	28
7	Considerações finais	30
8	Referências	31

1 Introdução

Câncer de mama é o segundo tipo de câncer com mais ocorrência em mulheres nos Estados Unidos da América (EUA) [1] e estima-se que por volta de 268,000 novos casos serão diagnosticados no ano de 2018 nos EUA e que mais ou menos 41,000 pessoas morrerão devido a essa patologia nessa mesma época [1]. A análise clínica de câncer de mama normalmente envolve o uso de diagnóstico por imagem e a mamografia é umas das técnicas utilizadas na detecção precoce do câncer de mama [2]. Mamografia é um exame de rastreio por imagem utilizando raio-x que permite visualmente analisar o tecido mamário [2]. Recomenda-se que mulheres acima dos 40 anos, ou consideradas dentro do grupo de alto risco de câncer mamário, façam exames de mamografia anualmente, pois a detecção precoce desse câncer permite um tratamento eficiente para cura do mesmo [3]. Diferentes características de tumores, tecidos e o erro humano podem causar diagnósticos errados dentro da área de oncologia [4].

O objetivo desse projeto é desenvolver uma aplicação para análise de câncer de mama em mulheres. Essa ferramenta utilizará redes convolucionais com aprendizagem profunda para classificar uma ou mais mamografias dentro da categoria de benigno ou maligno.

O projeto foi desenvolvido para Desktop (PC) para Windows porem pode ser facilmente portado para Linux, foi desenvolvido em Python e a principal biblioteca utilizada foi o Tensorflow da Google[13]. A aplicação foi desenvolvida a partir de uma aplicação já existente que utiliza redes convolucionais para classificação de câncer de mama a partir de imagens histológicas[14], a ideia foi adaptar tal aplicação para análise de mamografias.

O projeto envolveu muito estudo de tecnologias não profundamente vistas durante a graduação como machine learning, redes convolucionais e processamento digital de imagens, além do estudo multidisciplinar biológico para entendimento do domínio de mamografias e câncer de mama.

Seria muito interessante a continuidade desse trabalho para o desenvolvimento de uma ferramenta robusta que utilize o estudo e a aplicação desenvolvida para auxiliar os profissionais de medicina na análise clínica de exames de mama. Com uma interface de fácil uso e aperfeiçoamento da aplicação em questão.

2 Situação Atual

Não existe nenhum exame ou grupo de exames que possa assegurar que a mulher não tenha câncer de mama, os exames clínicos avaliam características teciduais diferentes que serão utilizadas pelo profissional da área para interpretação [2]. Há diversos componentes passivos de análise pelo profissional durante um exame de imagem da mama, são eles: nódulos, calcificações, distorção arquitetural, assimetria, linfonodo intramamário, lesão de pele e ducto único dilatado [2]. A análise de todos esses componentes segue uma orientação complexa e resulta em uma categorização baseada em probabilidade com diferentes condutas que podem causar estresse desnecessário para o paciente ou a prorrogação de um tratamento que já deveria ter sido realizado. Estima-se que uma a cada cinco mamografias resultam em falsos negativos [1] e que 7% a 9% de mulheres que realizam mamografias anualmente receberam um resultado falso-positivo recomendando a etapa de biopsia [5].

Categorização de imagens por meio de redes convolucionais é um processo que tem evoluído muito ao longo dos anos e possui diversas ferramentas para facilitar o desenvolvimento de soluções robustas e assertivas relacionadas a diferentes áreas. O estudo que obteve a menor taxa de erro no desafio de classificação que utiliza o CIPHAR-10 (Canadian Institute For Advanced Research) foi o AutoAugment [6], que usou de técnicas de transformações das imagens da base de dados para aumentar a precisão de seu algoritmo.

Diversos estudos são feitos relacionando biologia e medicina com inteligência artificial, pois questões dentro dessa área comumente contém informações complexas e numerosas demais para uma análise humana ou baseada em software comum.

A utilização de ferramentas de auxílio ao profissional de medicina pode ter um impacto enorme em custos relacionados à saúde evitando tratamentos desnecessários além de proporcionar um tratamento mais rápido e assertivo ao paciente. Esse projeto estuda e desenvolve uma aplicação que poderá ser utilizada como base do desenvolvimento de tal ferramenta.

3 Propostas e Objetivos do trabalho

O projeto tem como objetivo desenvolver uma aplicação que classifique diferentes imagens oriundas de mamografias no formato DICOM (Digital Imaging and Communications in Medicine) [21] em benignas e malignas.

Será utilizada uma rede convolucional, ideal para classificação de imagens, pois dispensa boa parte do pré-processamento da base de dados, devido à inteligência do algoritmo de aprender quais filtros devem ser aplicados a imagem antes de passar para a camada de rede neural tradicional para classificação.

Algumas arquiteturas diferentes serão estudadas para se decidir qual deve ser abordada no projeto, utilizando benchmarks de classificação de imagens como CIPHAR-10, CIPHAR-100 e ILSVRC (ImageNet Large Scale Visual Recognition Challenge). Diferentes configurações dos parâmetros da rede e da base de dados utilizada para treinamento serão testadas visando aperfeiçoar a classificação resultante da rede. Dependendo do tempo necessário para se computar o treinamento da rede desenvolvido poderá ser interessante utilizar múltiplas GPUs (Graphic Processing Units) e/ou um cluster para acelerar tal etapa do projeto.

Deseja-se ter ao final do projeto uma aplicação que possa servir como base para o desenvolvimento de uma ferramenta para uso clínico real auxiliando profissionais de medicina no dia-a-dia, e ter uma rede robusta que sirva como fonte de estudo para futuro desenvolvimento relacionando oncologia e processamento de imagens com inteligência artificial.

4 Atividades Realizadas

A primeira fase do projeto foi de estudo teórico da área de câncer de mama e de redes convolucionais para machine learning com imagens. A segunda fase envolveu a decisão de que linguagens, ferramentas e bibliotecas seriam utilizadas no desenvolvimento do projeto. A terceira fase envolveu testes, adaptação do projeto. A quarta e última fase envolveu a escrita deste relatório, documentando todo o projeto e suas conclusões.

4.1 Estudos preliminares

O conhecimento e a experiência do graduando em questão em relação ao tema de machine learning, inteligência artificial e processamento digital de imagens foram obtidos de duas formas antes do desenvolvimento desse projeto.

A primeira foi durante o curso da matéria ENG1456 – Inteligência Computacional Aplicada, cuja professora é a orientadora desse projeto. Durante o desenvolvimento da disciplina o aluno teve seu primeiro contato com inteligência artificial e redes neurais, criando uma base de entendimento para idealização desse projeto.

O segundo momento de contato com inteligência artificial e processamento digital de imagens foi durante os 8 meses de iniciação científica realizada com o Italo Matias, co-orientador desse projeto, no laboratório de Bio-Design da PUC-Rio. Durante a iniciação foi realizado um projeto de segmentação de imagens médicas para análise de tumores pulmonares. Nesse tempo houve um acúmulo de conhecimento muito grande relacionado a processamento digital de imagens.

4.2 Estudos conceituais e de tecnologia

O início do projeto envolveu o estudo teórico da área de câncer de mama e mamografias, para isso foi utilizado o Atlas BI-RADS[2] que tem como objetivo definir, padronizar o processo de análise de imagens de mama, para melhorar a qualidade e segurança das análises clínicas.

Mamografia é uma imagem raio-x da mama, o exame para se obter tal imagem envolve colocar o peito da paciente entre duas superfícies firmes, logo em seguida as superfícies apertam o seio deixando-o mais reto o que facilita a obtenção da imagem. Após o exame as imagens são analisadas por um especialista na procura dos seguintes achados:

- **Nódulos:** Também chamados de massas são volumes tridimensionais classificados de acordo com sua forma, margem e densidade. A alta densidade do nódulo está significativamente associada a malignidade do mesmo.
- **Calcificações:** Classificadas de acordo com a forma, tamanho e distribuição. Calcificações benignas são normalmente maiores enquanto as malignas tendem a ser muito pequenas.
- **Distorção arquitetural:** Diz respeito a análise das linhas do tecido da mama, podendo estar relacionada a malignidade.
- **Assimetria:** Significa a assimetria das fibras do tecido, pode indicar lesões e malignidade.
- **Linfonodo intramamário:** São nódulos circunscritos tipicamente benignos.
- **Lesão na pele:** Lesão que se projeta sobre a mama, tipicamente benigna.
- **Ducto único dilatado:** Estrutura tubular normalmente rara e benigna.

Todos esses achados devem ser analisados objetivamente seguindo o sistema de laudos, que envolve uma breve descrição da indicação do exame, descrição sucinta da composição geral da mama, descrição objetiva de quaisquer achados importantes, comparação com exames anteriores caso o radiologista julgue necessário, avaliação de acordo com a escala BI-RADS, e a conduta que deve ser tomada.

A classificação do BI-RADS é dividida em seis categorias, sendo a quarta subdividida em três subcategorias, gerando um total de nove categorias cada categoria está atrelada à uma conduta e uma probabilidade de câncer. Tentando dividir essas categorias em benignas e malignas baseado na probabilidade de câncer temos cinco categorias provavelmente benignas (de 1 a 4B) benignas e duas malignas (4C e 5).

Avaliação	Conduta	Probabilidade de câncer
Categoria 0: incompleta	Reconvocação para obtenção de mais imagens ou comparação com exames anteriores	N/A
Categoria 1: negativa	Rastreamento mamografico de rotina	Essencialmente 0%
Categoria 2: achados benignos	Rastreamento mamografico de rotina	Essencialmente 0%
Categoria 3: achados provavelmente benignos	Seguimento de curto prazo (6 meses) ou acompanhamento periodico	0 a 2%
Categoria 4A: achados com baixa suspeita de malignidade	Diagnostico tecidual	2 a 10%
Categoria 4B: achados com moderada suspeita de malignidade	Diagnostico tecidual	10 a 50%
Categoria 4C: achados com alta suspeita de malignidade	Diagnostico tecidual	50 a 95%
Categoria 5: achado altamente sugestivo de malignidade	Diagnostico tecidual	> 95%
Categoria 6: malignidade comprovada por biopsia	Cirurgia quando apropriada	N/A

Figura 1: Tabela de classificação BI-RADS

As visões padrões relacionadas as mamografias são a craniocaudal (CC) e a mediolateral-oblíqua (MLO), o motivo de ter essas duas incidências são o aumento da sensibilidade das áreas cegas, diminuição dos falsos positivos.

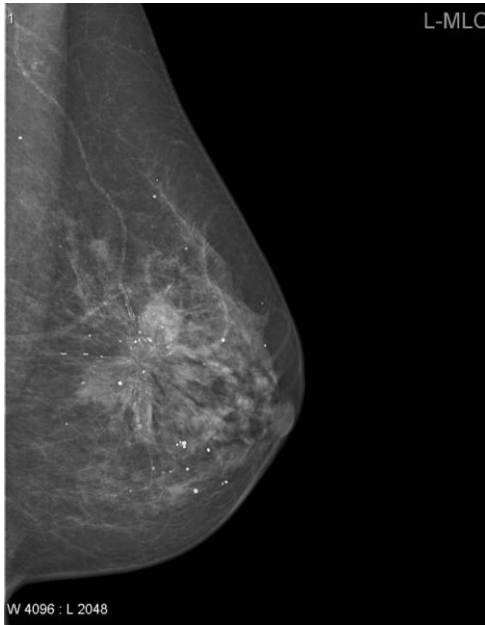


Figura 2: Imagem de visão MLO [7]

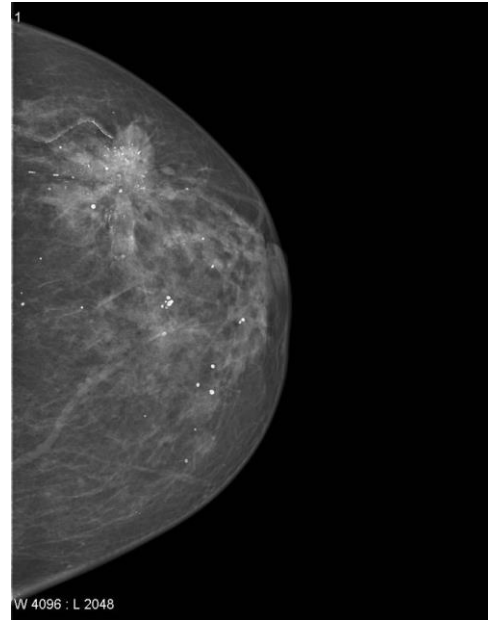


Figura 3: Imagem de visão CC [7]

Após o estudo teórico relacionado a mamografia, foi feito um estudo mais aprofundado sobre redes neurais e machine learning na classificação de imagens.

Redes neurais artificiais surgiram com o intuito de copiar certas capacidades do cérebro humano, como classificação, generalização e reconhecimento de padrões, ela é composta por unidade de processamento simples denominadas neurônios ou nós[11]. Um neurônio possui entradas, um corpo que realiza alguma operação matemática e produz uma saída[11].

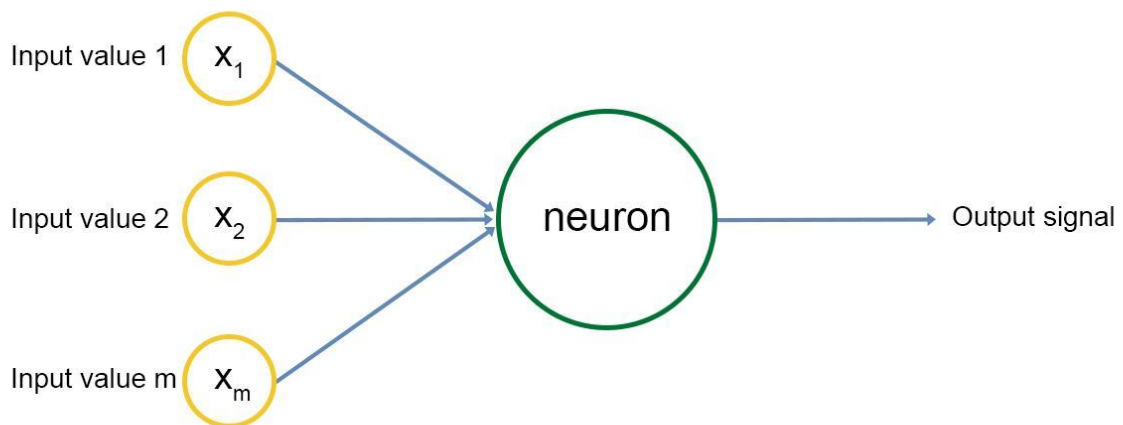


Figura 4: Modelo básico de um neurônio

Um dos primeiros modelos matemáticos desenvolvidos para formular o neurônio foi o modelo do Perceptron [11], nesse modelo cada entrada é binária (zero ou um) e possui um peso associado a ela, também foi adicionada uma constante como entrada denominada bias e a saída passa uma função de ativação,

que no caso do Perceptron é a função degrau.

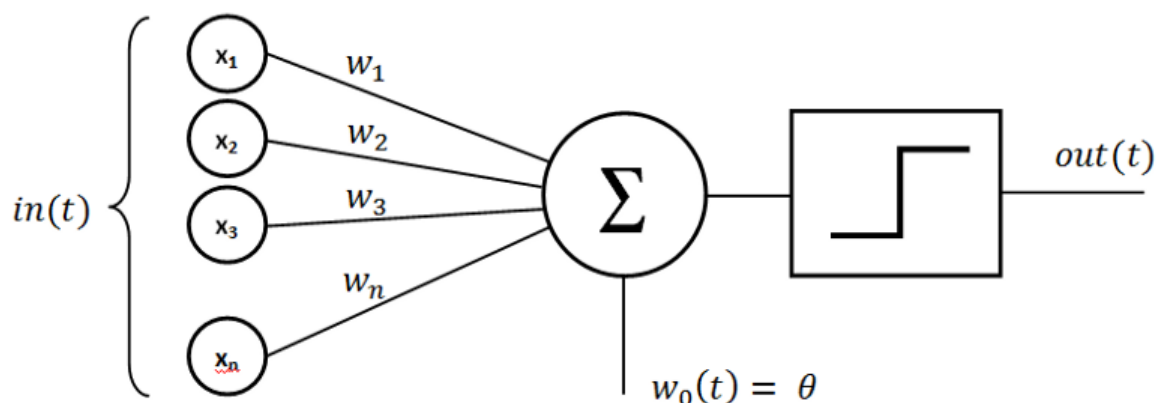


Figura 5: Perceptron

A função de ativação degrau faz com que a saída da soma das entradas associadas aos seus respectivos pesos gera um valor binário (zero ou um) que baseado em um *threshold* (limite). Esse modelo limita matematicamente as entradas e a saída devido a sua restrição binária.

Com essa limitação outros modelos matemáticos para definir neurônios surgiram, como a *Sigmoid* e a *Rectified Linear Unit* (ReLU), ambos permitindo valores reais como entradas e gerando valores reais como saída [15].

Esses neurônios são conectados para formar uma rede neural, essa estrutura formada pelos neurônios permite que os sinais de entrada de rede sejam processados pelos neurônios que se comunicam entre si, alterando os pesos de cada entrada adaptando os neurônios a diferentes estímulos proporcionados pelos sinais de entrada. É comum que esses neurônios sejam distribuídos em camadas, classificadas em camada de entrada, onde os padrões são apresentados a rede, camadas intermediárias ou escondidas, onde é feita a maior parte do processamento através das conexões ponderadas pelos pesos, e a camada de saída, onde o resultado final é concluído e apresentado [15]. Assim é possível que a rede “aprenda”, simulando o processo de aprendizado do cérebro humano.

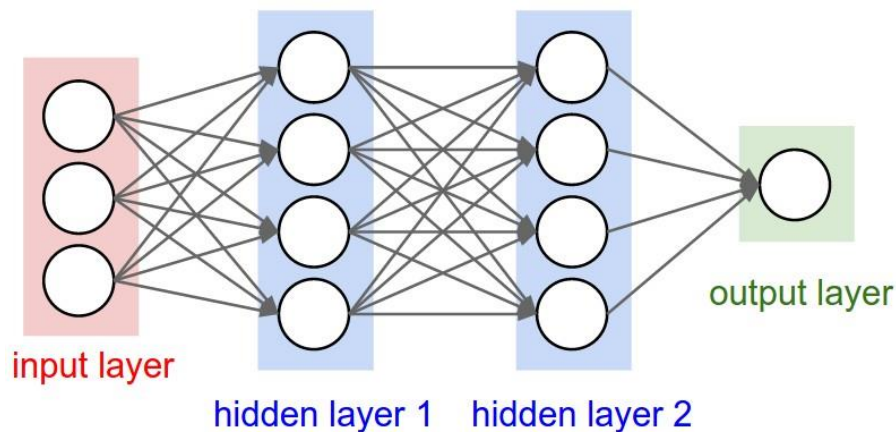


Figura 6: Camadas da rede neural

Para que esse aprendizado ocorra três métodos de treinamento foram desenvolvidos, aprendizado supervisionado, não supervisionado e por reforço. No aprendizado supervisionado existe um agente externo que indica à rede a resposta desejada para o padrão de entrada, ou seja, a saída foi pré-definida externamente à entrada. No aprendizado não supervisionado não existe um agente externo indicando a resposta desejada aos padrões de entrada. No aprendizado por reforço a rede também não possui a informação de resposta porém recebe um reforço positivo ou negativo dependendo da resposta. Denomina-se ciclo de uma aprendizagem quando todos os pares entrada e saída do conjunto de treinamento são apresentados para a rede.

Backpropagation é um algoritmo de aprendizado supervisionado, em que é calculado um erro após o processamento da rede do sinal de entrada até a saída, e esse erro é propagado na direção inversa da rede, ou seja, da saída para entrada, atualizando os pesos de cada neurônio de acordo com esse erro calculado [11].

As redes perceptron multicamadas (MLP) se caracteriza por ter todos seus neurônios conectados a um neurônio da próxima camada [10], a figura 6 é um exemplo de uma MLP. Essa rede utiliza do mecanismo de *backpropagation* para realizar o seu aprendizado.

Redes convolucionais [15] são versões adaptadas de MLP usadas principalmente para classificação e análise de imagens, essas redes recebem esse nome devido ao uso do processo de convolução, que no sentido de redes convolucionais é uma operação linear que envolve a multiplicação de um grupamento de entradas com pesos e um filtro estabelecido, gerando na saída um mapa de ativação daquele filtro, com isso a rede aprende quais filtros são ativados quando certas características são detectadas em uma posição espacial da entrada. Assim a própria rede aprende os filtros que antes eram aplicados no pré-processamento de imagens, evitando que seja necessário um conhecimento prévio das características da imagem relevantes para sua classificação [10][12].

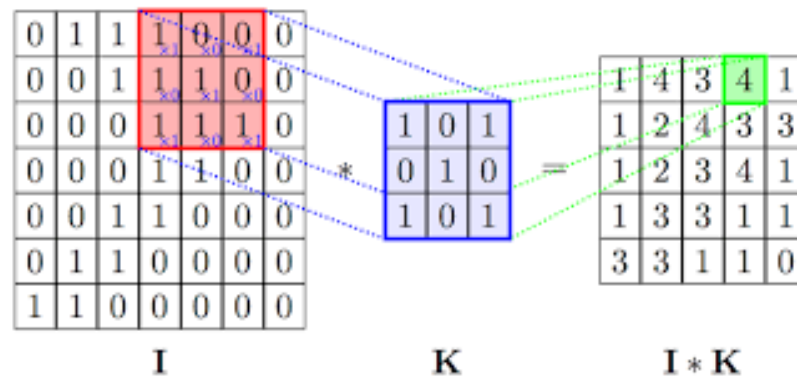


Figura 7: Filtro K sendo convolucionado na imagem I gerando mapa de características.

Para lidar com entradas de matrizes de grandes dimensões como imagens, é impraticável conectar neurônios com todos os neurônios da camada anterior como ocorre na MLP. Redes convolucionais utilizam da correlação da informação espacial para conectar os neurônios, ou seja, cada neurônio só está conectado à uma pequena região da entrada, aumentando desempenho e separando a imagem em frames.

Além da camada de convolução temos a camada de *pooling* que tem como objetivo diminuir progressivamente o tamanho espacial da representação para reduzir a quantidade de parâmetros para serem computados pela rede, evitando o *overfitting* que é um problema causado pela grande quantidade de parâmetros que a rede deve processar, causando uma grande variância no resultado da classificação[15]. *Pooling* é normalmente realizado através do max pooling que obtém o maior valor de uma certa região da matriz para propagar. Essa camada de pooling normalmente é inserida logo após uma camada de convolução. Outra técnica para se combater o *overfitting* é a adição de uma camada de *Dropout* que tem como objetivo descartar neurônios da rede diminuindo a propagação de ruídos na mesma [21].

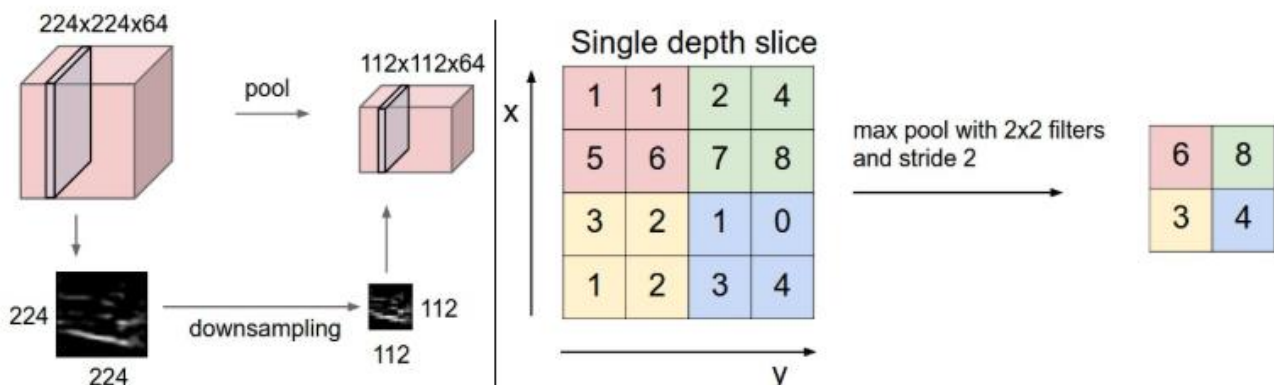


Figura 8: Processo de *pooling*

Outras camadas que podem ser adicionadas a rede são de ReLU que aplica a função de ativação removendo valores negativos do mapa de ativação, transformando-os em zeros, com isso os aspectos de decisão e não-linearidade são melhorados. E a camada de perda (*loss layer*) que determina como o treino penaliza a variação entre o que foi predito e o valor real, é comum serem utilizadas funções de *softmax* e *sigmoid cross-entropy* nessa camada.

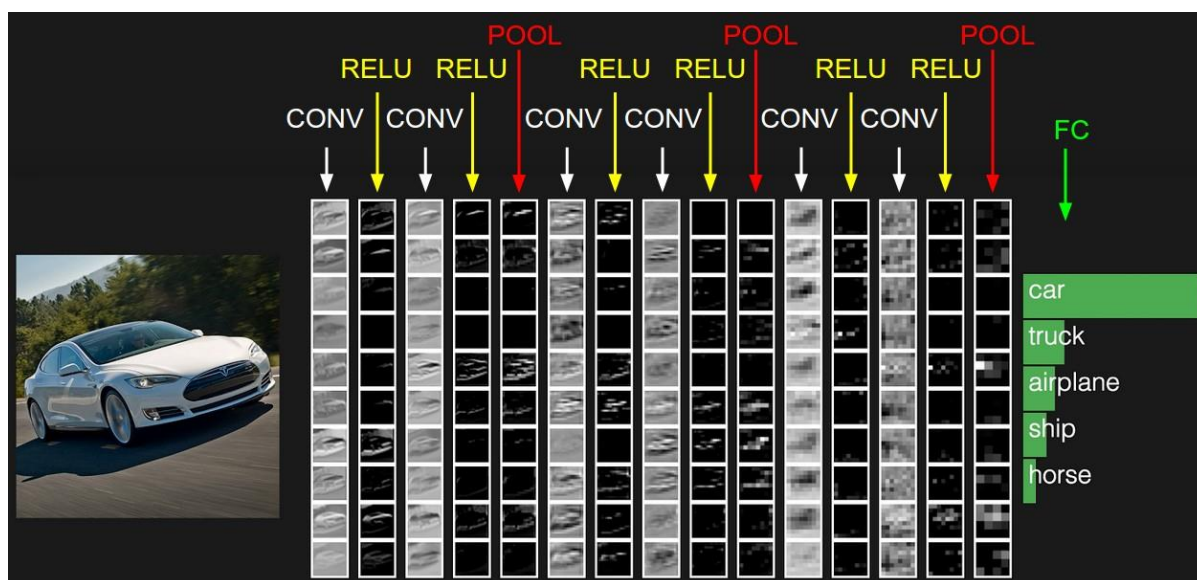


Figura 9: Exemplo de arquitetura de rede convolucional

Necessariamente no final de rede convolucional temos uma rede MLP totalmente conectada visto na figura 9, realizando o processamento das entradas e classificando a saída.

4.3 Testes e protótipos

Python foi linguagem escolhida para o desenvolvimento da rede neural utilizada, pelo fato de ser uma linguagem que possui uma grande quantidade de ferramentas relacionadas a processamento de imagens e redes neurais. A principal biblioteca utilizada foi o Tensorflow, um framework para machine learning open source criado pela Google. Além disso o autor do projeto já possuía experiência com Python o que facilitou o desenvolvimento do projeto.

No início a biblioteca Tensorflow foi estudada através da própria documentação oficial além de vídeos tutoriais que explicavam um pouco mais a fundo o uso da biblioteca [13].

Depois dessa introdução ao Tensorflow foi realizada uma pesquisa para descobrir projetos já desenvolvidos que poderiam servir como base para o

desenvolvimento da solução proposta pelo projeto. E uma das soluções encontradas foi o projeto Classifying Cancer [15] que é um projeto em Python que utiliza Tensorflow para classificar imagens de tumores de mama. Esse projeto possui uma rede neural profunda que classifica os tumores em benignos ou malignos dependendo de medidas já mensuradas e guardadas em um arquivo .csv (Comma Separated Values), e uma rede neural convolucional que foi testada em histogramas de tumores.

Após alguns testes para comprovar funcionamento do projeto e análise do código para entendimento do que foi desenvolvido e necessidades básicas para execução do software foi decidido utilizar esse projeto como base para o desenvolvimento da solução proposta, visto que seria fácil adapta-lo alterando parâmetros de treino e a arquitetura da rede em questão. É importante ressaltar que o código está sob a licenciado sob a GNU General Public License v3.0 que permite modificação, uso comercial, distribuição e uso privado [16].

4.4 Método

O projeto totalizou dois semestres de desenvolvimento, o primeiro (2018.1) para o desenvolvimento da parte I e o segundo (2019.1) para o desenvolvimento da parte II. Foi decidido esperar até 2019.1 para finalizar o projeto para alinhar com a formatura do autor e em 2018.2 o autor deu uma pausa no desenvolvimento do projeto. Assim foi necessário atualizar o cronograma proposto no relatório I para refletir a realidade do projeto.

Cronograma 1:

Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
Estudo teórico de redes neurais e cancer de mama								
	Estudo do escopo do projeto e das ferramentas a serem utilizadas							
		Testes das ferramentas						
			Elaboração do Relatório Final 1					
				Desenvolvimento da rede neural				
					Teste, treinamento e aperfeiçoamento da rede			
					Desenvolvimento da aplicação final			
						Elaboração do Relatório Final 2		
							Eventuais correções gerais	
								Defesa do Projeto Final

Cronograma 2:

Abril	Maio	Junho	Julho
Estudo teórico redes neurais e câncer de mama			
	Estudo do escopo do projeto e das ferramentas		
		Testes das ferramentas	
			Relatório Final 1
2019.1			
Desenvolvimento da solução			
	Teste		
		Relatório Final 2	
			Defesa

Não foi utilizada uma metodologia de desenvolvimento ágil como Scrum, Kanban, XP. Apenas foram considerados os prazos propostos no cronograma para se finalizar o projeto dentro do prazo determinado pelo departamento.

5 Projeto e especificação do problema

5.1 Base de dados

Primeiramente foi necessário pesquisar uma base de dados de mamografias que pudesse ser densa o suficiente para que a rede possa aprender a ainda testar o “conhecimento” adquirido.

A base de dados utilizada foi a CBIS-DDSM (Curated Breast Imaging Subset of DDSM) que é uma versão atualizada e padronizada da DDSM (Digital Database for Screening Mammography) [18] de 1997. Essa base de dados possui 2.620 casos de estudo de mamografias, contendo imagens salvas no formato DICOM[20], tanto em seu tamanho original quanto em forma de segmentação ROI (Region Of Interest) [19]. As imagens dessa base de dados são oriundas do Hospital Geral de Massachusetts, Escola Medicinal Universitaria de Wake Forest, da Escola de Medicina de Washington University of St Louis e do US Army Research and Material Command.

As imagens são divididas em dois grupamentos: calcificação e massa, e cada grupo tem imagens classificadas em normais, benignas e malignas. A base de dados original expõe as informações de classificação entre outra em forma de metadados, mas para facilitar os desenvolvedores da base padronizada extraíram essas informações e as juntaram em arquivos .csv

Para garantir a consistência dos dados da base de dados os desenvolvedores contrataram um médico para analisar alguns casos questionáveis e os mesmos foram removidos da base atual.

As imagens segmentadas ROI foram melhoradas através de um algoritmo de segmentação de lesões utilizado para delineação da massa do tumor do tecido em volta, esse algoritmo só foi aplicado as imagens de massa e está evidenciado na figura 10. Devido à baixa precisão alcançada utilizando as imagens não segmentadas, foi decidido utilizar as imagens segmentadas ROI para aprendizado e teste da rede, o algoritmo de segmentação não foi implementado nesse projeto e fica como uma possível melhoria para o futuro caso esse projeto seja revisitado.

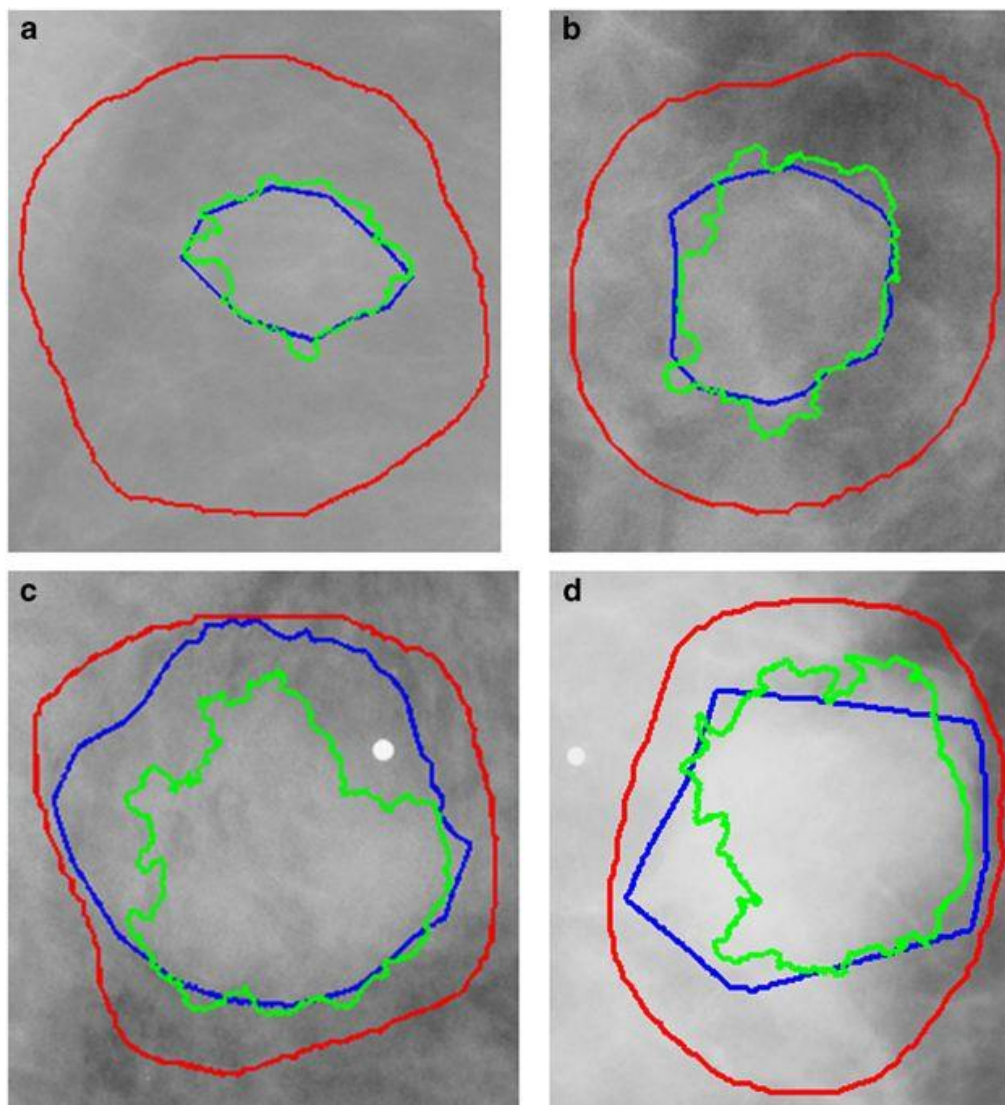


Figura 10: Quatro casos de imagens de mamografias. A linha vermelha delimita a área segmentada previamente, a linha azul delimita a área segmentada manualmente por um especialista e a linha verde delimita a área segmentada pelo algoritmo.

A base de dados já foi dividida em partições padronizadas para treinamento e teste baseados na escala BI-RADS [2], tendo 20% dos casos para teste e o resto para treinamento da rede. Cada caso dos subsets de massa e calcificação possui a imagem craniocaudal (CC) e mediolateral-obliqua (MLO) que são imagens padrões resultantes de mamografias [19] vistos nas figuras 2 e 3.

5.2 Aplicação

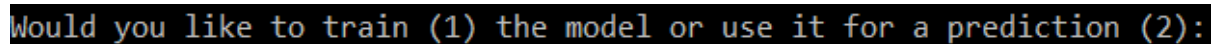
Para o desenvolvimento da aplicação foi feito um *fork* do projeto original no Github o que cria uma cópia do repositório para que as alterações necessárias possam ser feitas em alterar o projeto original. O projeto alterado estará público para que qualquer pessoa possa ver o código porem apenas o autor terá poder de contribuição para o projeto, podendo abrir o mesmo no futuro. O projeto utiliza das seguintes bibliotecas:

- **Tensorflow:** já mencionado nesse documento é a principal biblioteca utilizada na aplicação, que nos permite criar a utilizar redes neurais de forma fácil e simples. Foi utilizada a ferramenta Tensorboard do Tensorflow para gerar visualizações da rede.
- **Numpy:** Biblioteca para computação científica em Python que permite criar arrays N-dimensionais para serem facilmente computados. Foi utilizada para guardar informações das imagens.
- **Scikit-learn:** Biblioteca de machine learning em Python que foi utilizada para fazer permutações aleatórias em coleções, no caso nossa base de dados para gerar aleatoriedade na escolha das imagens a serem treinadas e testadas.
- **Open CV:** Biblioteca de visão computacional utilizada para carregar as imagens e redimensiona-las.
- **Glob:** Biblioteca que permite utilizar o padrão de caminhos do sistema operacional no estilo Unix. Utilizada para se obter caminho de salvamento e carregamento de arquivos.

Algumas outras bibliotecas mais comuns também foram utilizadas para simples programação da lógica em Python.

O funcionamento da aplicação está bem descrito na documentação do Github, aqui apenas será feita uma descrição superficial de como rodar a aplicação.

Para executar a aplicação basta executar o `main.py` contido no projeto. No caso o autor utilizou a ferramenta Anaconda para facilitar instalação de bibliotecas, e dentro da própria ferramenta realizou a chamada do `main.py` por linha de comando, também é possível configurar o projeto dentro de uma IDE (Ambiente Integral de Desenvolvimento) e executar o programa pela interface da mesma, para desenvolver o autor utilizou o PyCharm e o Spyder contido na própria Anaconda. Ao executar a aplicação o usuário deverá escolher se ele deseja treinar a rede ou usa-la para classificação, como mostra a imagem a seguir:



```
Would you like to train (1) the model or use it for a prediction (2):
```

Figura 11: Mensagem exibida ao usuário ao executar a aplicação

Ao selecionar a opção 1 o modelo será criado caso ainda não exista e treinado a partir das imagens contidas na pasta *images* dentro do projeto. Essa pasta está subdividida em duas: *train* e *predict* que deve conter as imagens utilizadas para treino e classificação respectivamente, e cada uma dessas está subdividida em duas: *benign* e *malignant*, que deve conter as imagens já organizadas em benignas e malignas. O projeto teve como objetivo apenas classificar os tumores em benignos e malignos mesmo que a divisão da base de dados seja normal, benigno e maligno, o interesse futuro é realizar essa classificação dentro da escala BI-RADS[2].

Após o treino a rede/modelo fica salvo dentro de uma pasta *tmp* e pode ser utilizado para análise pelo Tensorboard e para a classificação. Para classificar as imagens basta executar a aplicação novamente e escolher a opção 2, nesse caso o programa carregará as imagens dentro da pasta *predict* e realizará a classificação, mostrando para o usuário a porcentagem de assertividade no final da execução, como será melhor descrito nos testes.

5.3 Ferramentas auxiliares

As imagens da base de dados estão disponíveis no formato DICOM[20], porém para serem utilizadas pela aplicação elas precisam estar no formato PNG (Portable Networks Graphics). Além de que as imagens não estão disponíveis em pastas similares a necessitada pela aplicação.

Para resolução dessas questões foram desenvolvidos dois *scripts* em Python: `readCsv.py` e `convertDicomToPng.py`. O primeiro tem como objetivo ler o `.csv` disponível junto a base de dados para separar as imagens em benignas e malignas em duas pastas diferentes. O segundo tem como objetivo converter as

imagens DICOM[20] em PNG simplesmente alterando a extensão das mesmas. Os scripts foram desenvolvidos com caminhos absolutos e devem ser alterados para rodar na máquina do usuário, seria interessante alterar esses *scripts* para que essa parte do processo seja automática, facilitando o processo em si. E depois de executar os scripts ainda é necessário passar as imagens manualmente para as pastas da aplicação.

6 Implementação e avaliação

Diversas alterações foram feitas na aplicação principal para se adequar ao domínio do problema, e diversos testes foram feitos para validar mudanças estruturais, como arquitetura da rede e de parâmetros de aprendizado para garantir maior assertividade final da solução. Inicialmente a primeira mudança foi alterar a quantidade de canais de cores utilizada como parâmetro do programa de três (RGB das imagens histológicas) para um (cinza monocromático das imagens DICOM), essa mudança tem grande alteração na definição da arquitetura que será exposta em seguida. Algumas outras simples mudanças de alteração de caminhos de arquivos e chamadas das bibliotecas foram feitas para adequar o código e por fim foram iniciados os testes da aplicação.

Foi montada uma bateria de testes para se decidir qual arquitetura seria escolhida, quantidade de classificações que seriam feitas, ou seja, se os tumores seriam classificados em benignos e malignos ou benignos, malignos e normais, qual o tamanho que a imagem seria redimensionada, tamanho do batch (*batch size*), teste de quantidade de *epochs* e os testes finais para gerar os resultados da solução.

Batch size é a quantidade de amostras que serão inseridas na rede a cada iteração de treino, assim a rede é treinada por ciclos de treinamento, chamados de *epochs* e a cada ciclo uma quantidade determinada pelo *batch size* é utilizada para o aprendizado, dessa maneira a fase de treino/aprendizagem da rede é composto de vários *epochs* configuráveis [10]. Cada ciclo é composto por duas fases, a de transmissão dos sinais de entrada da rede para a saída e a de atualização dos pesos baseada na estimativa de erro. A variação desses parâmetros influencia na ocorrência ou não dos fenômenos de *overfitting* e *underfitting* [15] o primeiro já explicado na seção 4.2 relacionado a camada de *pooling* e o segundo sendo o contrário do primeiro, que faz com que seu modelo tenha pouca variância perante as entradas, sendo incapaz de interpretar as diferentes características das amostras, classificando-as incorretamente.

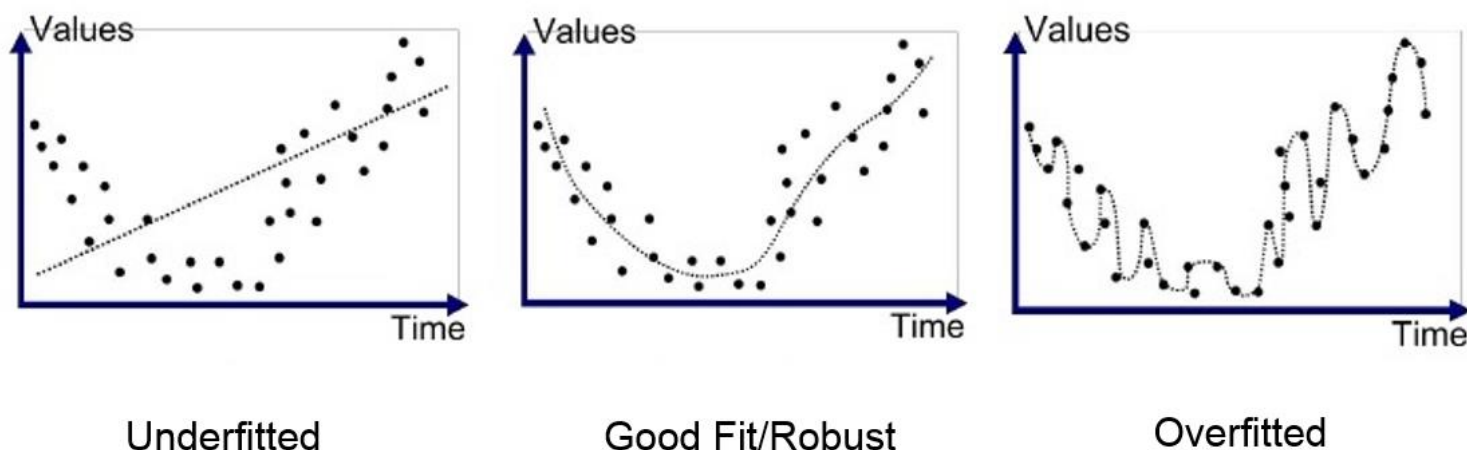


Figura 12: Gráficos mostrando diferença de uma rede *underfitted*, robusta e *overfitted*. A linha em pontilhado delimita as duas classificações das amostras representadas pelos pontos em azul.

6.1 Teste da arquitetura

Diversas arquitetura foram estudadas e testadas, dentre elas foram variados o *learning rate* e quantidade de *epochs* para se ter alguma métrica sobre aquela arquitetura. *Learning rate* é o parâmetro que determina como que novas informações adquiridas na fase de treinamento da rede sobrescreve aprendizados antigos [20], valores muito altos faz com que a rede não consiga aprender características novas, pois as atualizações de peso são bem grosseiras e faz com que o modelo tenha dificuldade de convergir, já valores muito baixos fazem com que o modelo demore muito para convergir.

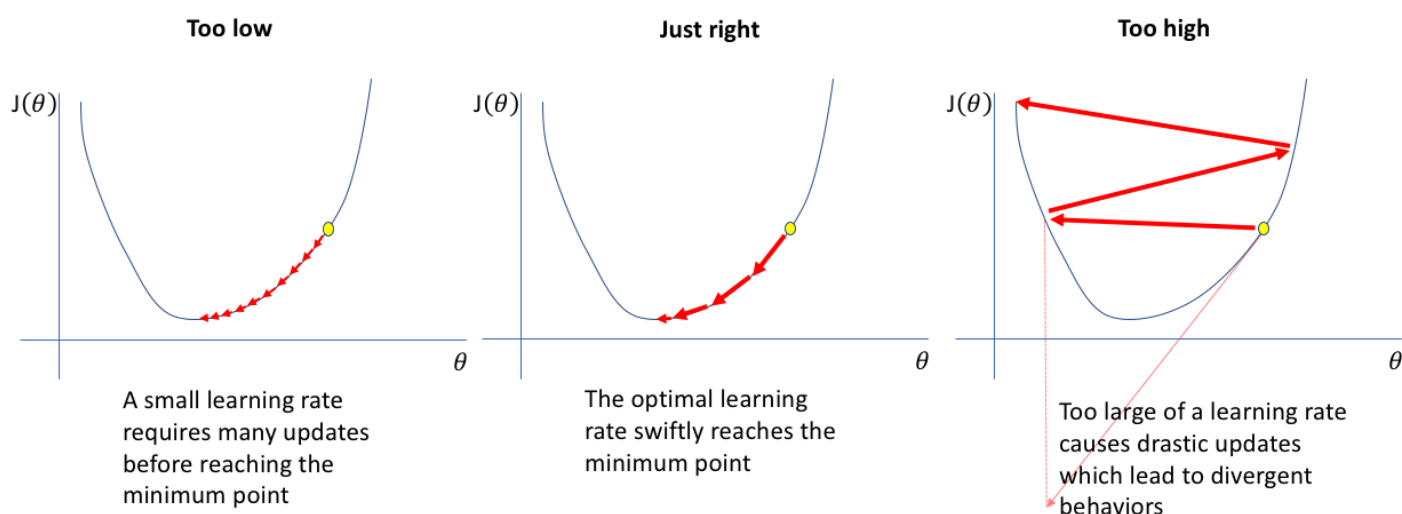


Figura 13: Gráfico da esquerda mostra exemplo de *learning rate* muito baixo, do centro mostra um valor robusto e o da direita mostra um valor muito alto.

Os testes para validar cada arquitetura foi utilizando as imagens segmentadas ROI de calcificação e massa, essas imagens estão sendo redimensionadas para 64x64 pixels, foram realizados testes com redimensionamento 128x128 que causou uma lentidão no processamento da rede e basicamente nenhum ganho pratico, e foram realizados testes com redimensionamento 32x32 que demonstrou uma perda significativa na capacidade de classificação da rede (menos de 60% de assertividade). A primeira arquitetura testada foi a desenvolvida pelo próprio autor da solução original, ela é composta pelas seguintes camadas:

- **Conv1:** Composta por uma camada convolucional com 64 filtros na dimensão 3x3, que receberá uma imagem 64x64, uma camada de *pooling* que utiliza o *max pooling* com filtros 2x2, e essa camada utiliza a função de ativação *Rectified Linear Unit* (ReLU).
- **Conv2:** Composta por outra camada convolucional com 128 filtros de dimensão 3x3, e as mesmas camadas de *pooling* e ReLU da Conv1.
- **Conv3:** Igual a camada Conv2.
- **Fully_Connected1:** Primeira camada totalmente conectada com 1024 neurônios com uma camada de ReLU.
- **Dropout:** Camada que tem como objetivo combater o *overfitting* com 50% de chance de descartar um neurônio.
- **Fully_Connected2:** Segunda camada totalmente conectada tendo em sua saída a quantidade de classificações, no caso 2. Nessa camada a função de ativação é a *Softmax* que normaliza os valores da saída para que sua soma seja 1.

A figura 18 mostra a arquitetura da rede gerada utilizando o Tensorboard para visualização, para isso em diversos pontos do código são coletados dados (*summaries*) que são exibidos de forma organizada em um servidor local, a forma de como executar esse utilitário está descrita no Github na documentação original do projeto e só precisa ser atualizado para o caminho específico da máquina em que se está sendo executado. Além da arquitetura é possível ver analisar problemas na sua rede através da coleta de métricas quantitativas da sua rede, como o *learning rate*. A figura 14 mostra os dados de teste dessa arquitetura da rede.

Batch size	Epochs	Learning rate	Acurácia
64	20	0.0001	61.35%
64	20	0.001	65.03%
64	50	0.0001	65.54%
64	50	0.001	57.62%

Figura 14: Resultados dos testes da arquitetura original.

A primeira variação da arquitetura foi aumentando e diminuindo a quantidade de camadas convolucionais da rede, no caso foram testadas 2, 3, 4 e 5 camadas e o seu impacto na rede, essas camadas possuem a mesma estrutura de Conv2 e Conv3, foi mantido um learning rate de 0.0001 por causa dos resultados anteriores. Resultados estão evidenciados na figura 15.

Conv layers	Batch size	Epochs	Learning rate	Acurácia
2	64	20	0.0001	66.25%
2	64	50	0.0001	64.72%
4	64	20	0.0001	63.49%
4	64	50	0.0001	62.57%
5	64	20	0.0001	64.41%
5	64	50	0.0001	64.72%

Figura 15: Resultados dos testes da variando número de camadas convolucionais.

Com esses resultados foi decidido se manter 2 camadas convolucionais. A segunda variação envolveu a adição de uma camada totalmente conectada, e seu resultado foi calculado como mostra a figura 16.

Full Conn layers	Batch size	Epochs	Learning rate	Acurácia
3	64	20	0.0001	62.88%
3	64	50	0.0001	63.42%

Figura 16: Resultados dos testes com mais uma camada totalmente conectada.

Visto que a adição de uma camada totalmente conectada não atribuiu acurácia ao sistema, foram mantidas duas camadas que era a configuração inicial. A última variação foi a adição de uma camada de *dropout* após a primeira camada convolutiva devido à suspeita do efeito de *overfitting* estar ocorrendo de acordo com a validação de erro em cada etapa de treino. Os resultados estão exibidos na figura 17.

Dropout layer	Batch size	Epochs	Learning rate	Acurácia
2	64	20	0.0001	61.96%
2	64	50	0.0001	64.48%

Figura 17: Resultados dos testes com mais uma camada de *dropout*.

Concluindo esses testes não foi adicionada a camada de *dropout*. As figuras 18 e 19 mostram a arquitetura original e a nova respectivamente.

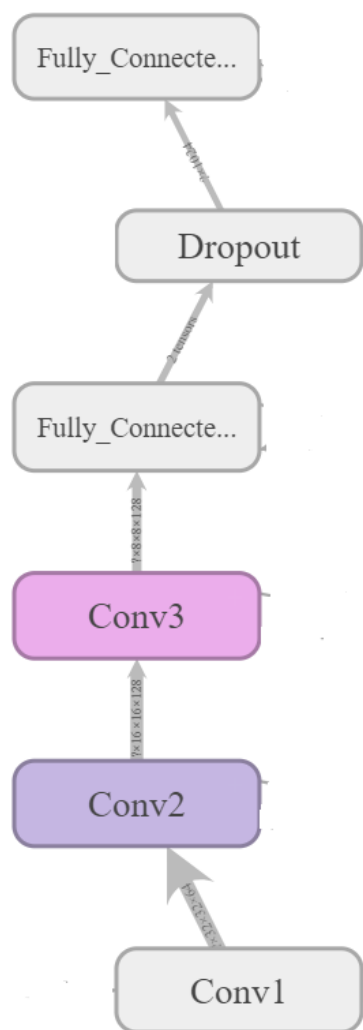


Figura 18: Arquitetura original.

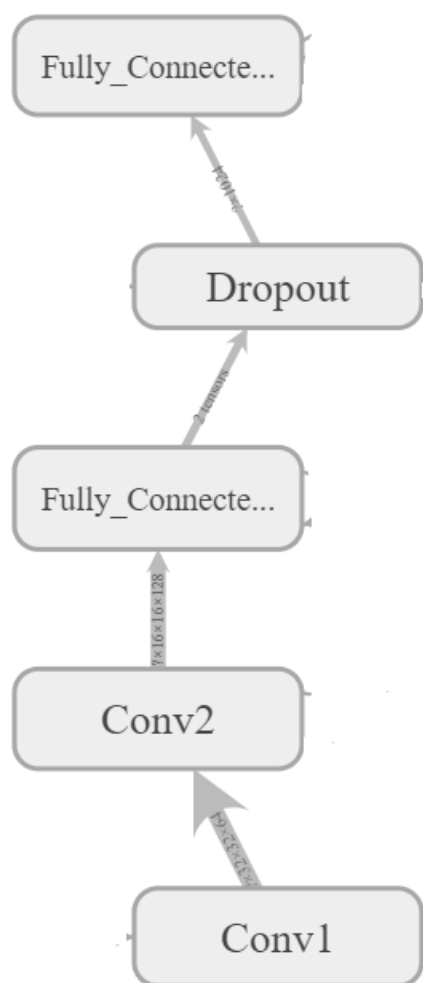


Figura 19: Arquitetura nova.

Após a validação da arquitetura foram alterados alguns parâmetros de treino para tentar melhorar a assertividade da solução.

6.2 Teste de epochs

A variação de *epochs* seguiu o padrão 10, 20, 30, 50, 80, 100, para teste da acurácia, *learning rate* ficou fixo em 0.0001 e *batch size* em 64. Resultados exibidos na figura 20

Batch size	Epochs	Learning rate	Acurácia
64	10	0.0001	60.73%
64	20	0.0001	66.25%
64	30	0.0001	65.70%
64	50	0.0001	64.72%
64	80	0.0001	62.68%
64	100	0.0001	55.52%

Figura 20: Resultados da variação de *epochs*.

Conforme o resultado foi mantido para testes variação de 20 e 30 *epochs*.

6.3 Teste de batch size

A variação de *batch size* seguiu o padrão 8, 16, 32, 64, 128, para teste da acurácia, *learning rate* ficou fixo em 0.0001 e *epochs* em 20 e 30.

Batch size	Epochs	Learning rate	Acurácia
8	20	0.0001	58.34%
8	30	0.0001	57.26%
16	20	0.0001	60.68%
16	30	0.0001	60.03%
32	20	0.0001	64.48%
32	30	0.0001	64.01%
64	20	0.0001	66.25%
64	30	0.0001	65.70%
128	20	0.0001	63.45%
128	30	0.0001	63.19%

Figura 21: Resultado dos testes variando *batch size*.

Conforme o resultado foi utilizado o *batch size* com 64 amostras.

6.4 Testes finais

O resultado final de acurácia não ficou satisfatório (66.25%) após os testes de arquitetura, *batch size* e *epochs*. Então foi analisada a base de dados e concluída que havia a possibilidade da mesma estar muito escassa para

treinamento, tínhamos 548 imagens de calcificações malignas, 997 imagens de calcificações benignas, 629 imagens de massas malignas e 689 imagens de massas benignas, totalizando 2.863 imagens.

Sendo assim foi realizado o processo de *data augmentation* [22] que tem o objetivo de aumentar sua base de dados aplicando certas transformações nas suas imagens que mantem as características que as classificam. Considerando que a rede convolucional seja robusta para classificar as imagens, diferentes orientações e posições dos atributos que caracterizam a classe não devem diferenciar o resultado, garantindo a invariância da rede. Normalmente essas transformações são feitas alterando ponto de vista, tamanho, iluminação ou aplicando translação.

Na base de dados foi aplicada a operação de espelhamento das imagens [23] para tentar melhorar o aprendizado da rede. Para isso foi criado mais um *script* em Python usando a biblioteca *Pillow* (Python Image Library), com isso a base de imagens duplicou, processo exemplificado na figura 22.

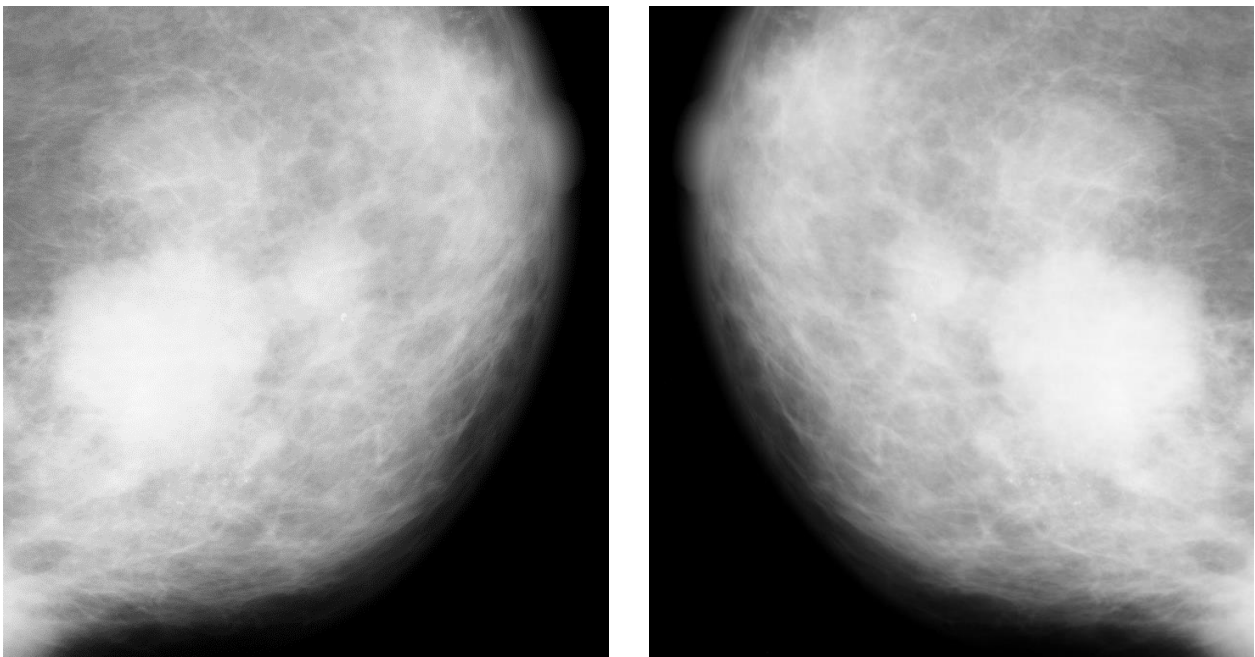


Figura 22: Imagem original da esquerda e imagem espelhada na direita.

Com a adição dessas imagens na base de dados foram realizados novos testes utilizando a arquitetura já definida, *batch size* de 64 amostras e variando o *epoch* com os valores 20, 30 e 50. E o melhor resultado encontrado foi com 30 *epochs*, 73.68%, evidenciado na figura 23.

Porcentagem = 73.6842105263158

Figura 23: Resultado final de assertividade.

A partir dos testes concluiu-se que o resultado final não foi satisfatório para ter a possibilidade de desenvolver uma aplicação que seja usada na área médica a partir dessa rede. Considerando a criticidade das decisões tomadas na área medicinal, qualquer *software* que tenha como objetivo auxiliar o profissional de medicina na sua tomada de decisão no dia-a-dia deve conter o máximo de acurácia possível. Algo em torno de 90% seria interessante para já ser testado em prática na medicina.

7 Considerações finais

No final acredito que o trabalho contribuiu para a área de pesquisa de análise de mamografias com auxílio da tecnologia, acredito que estudos que relacionam redes convolucionais e processamento digital de imagens na área médica evoluirão bastante nos próximos anos e esse projeto poderá ser revisitado e melhorado. O resultado final de acurácia não foi satisfatório considerando o nível de assertividade necessário na área médica.

O projeto resultou em um aprendizado enorme para o autor, tanto na área tecnológica quanto biológica e foi muito importante ver como software aplicado em uma área tão crítica como medicina pode resultar em uma solução para questões de saúde da população no futuro. O conhecimento obtido sobre redes neurais convolucionais e sobre processamento digital de imagens será carregado pelo resto da carreira profissional do autor.

Se o projeto começasse agora a rede seria aperfeiçoada e seria desenvolvida uma interface para que o usuário pudesse utilizar o software de maneira fácil, automática e intuitiva. Todos os processos ainda não automatizados completamente seriam automatizados e o produto final do projeto seria um software que poderia ser utilizado pelos médicos no auxílio do diagnóstico de tumores de mama. Também seria estudada a possibilidade de classificar as imagens dentro da escala BI-RADS[2] tornando a classificação mais granular o que poderia ser de interesse ao usuário final. Além disso mais imagens seriam procuradas para incrementar a base de dados e seria estudada a possibilidade de utilizar GPUs para realizar o processamento da rede ou até realizar o processamento na nuvem.

Próximos passos para o projeto caso alguém queira dar continuidade seriam

os já citados: aperfeiçoamento da rede, desenvolvimento de uma interface para utilização do software, busca de mais imagens para complementar a base de dados, automação de todos os processos necessários para preparar a execução da aplicação principal incluindo a segmentação ROI, classificar as imagens dentro da escala BI-RADS[2], melhorar o processamento e por fim até homologar e certificar uma aplicação que possa servir como auxílio ao profissional de medicina no seu dia-a-dia, na análise de tumores de mama.

8 Referências

- [1] SPIEGEL, Rebecca L.; MILLER, Kimberly D.; JEMAL, Ahmedin. **Cancer Statistics, 2018**. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.3322/caac.21442>>. Acesso em: 20 jun. 2018.
- [2] D'ORSI, C.J.; SICKLES, E.A.; MENDELSON, E.B.; MORRIS, E.A. **ACR BI-RADS® Atlas, Breast Imaging Reporting and Data System**. Reston, VA, American College of Radiology; 2013.
- [3] OEFFINGER, K.C.; FONTHAM, E.T.; ETZIONI R. **Breast cancer screening for women at average risk: 2015 guideline update From the American Cancer Society**. *JAMA*. 2015. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.3322/caac.21442>>. Acesso em: 20 jun. 2018.
- [4] GIESS, Catherine; FROST, Elisabeth; BIRDWELL, Robyn. **Difficulties and errors in diagnosis of breast neoplasms**. 2012. Disponível em: <<https://www.ncbi.nlm.nih.gov/pubmed/22824119>>. Acesso em: 20 jun. 2018.
- [5] HUBBARD R.A.; KERLIKOWSKI, K.; FLOWERS, C.I.; YANKASKAS, B.C.; ZHU, W.; MIGLIORETTI, D.L. **Cumulative Probability of False-Positive Recall or Biopsy Recommendation After 10 Years of Screening Mammography: A Cohort Study**. *Ann Intern Med*. 2011. doi: 10.7326/0003-4819-155-8-201110180-00004. Disponível em: <<http://annals.org/aim/article-abstract/474984/cumulative-probability-false-positive-recall-biopsy-recommendation-after-10-years?doi=10.7326%2f0003-4819-155-8-201110180-00004>>. Acesso em: 20 jun. 2018.
- [6] CUBUK, Ekin D.; ZOPH, Barret; MANE, Dandelion; VASUDEVAN, Vijay; LE,

Quoc V. **AutoAugment: Learning Augmentation Policies from Data**. eprint arXiv:1805.09501. 2018. Disponível em: <<https://arxiv.org/abs/1805.09501>>. Acesso em: 20 jun. 2018.

[7] Imagens de caso do A.Prof Frank Gaillard, **Radiopedia.org**, rID: 12608

[8] ROSENBLATT, Frank (1958), **The Perceptron--a perceiving and recognizing automaton**. Report 85-460-1, Cornell Aeronautical Laboratory.

[9] MCCULLOCH, Warren; PITTS, Walter (1943), **A Logical Calculus of the ideas immanent in nervous activity**, Bulletin of Mathematical Biology, Vol. 52, No. 1/2.

[10] NIELSEN, Michael A. **“Neural Networks and Deep Learning”**, Determination Press, 2015.

[11] SATHYANARAYANA, Shashi. **“A Gentle Introduction to Backpropagation”**. July 22, 2014.

[12] CIRESAN, Dan C., MEIER Ueli, MASCI Jonathan, GAMBARDILLA Luca M. SCHMIDHUBER Jorgen, **“Flexible, High Performance Convolutional Neural Networks for Image Classification”**

[13] **Tensorflow**. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 20 jun. 2019.

[14] **Classifying Cancer**. Disponível em: <<https://github.com/jhole89/classifying-cancer>>. Acesso em: 20 jun. 2019.

[15] **CS231n Convolutional Neural Networks for Visual Recognition**. Disponível em: <<http://cs231n.github.io/convolutional-networks/>>. Acesso em: 20 jun. 2019.

[16] **GNU General Public License** Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.en.html>>. Acesso em: 20 jun. 2019.

[17] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy & Daniel L. Rubin. **A curated mammography data set for use in computer-aided detection and diagnosis research**. Scientific Data volume 4, Article number: 170177 (2017).

- [18] Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M., & Rubin, D. L., et al. (2017). **A curated mammography data set for use in computer**. *Scientific Data* 64(2), p. 10-12.
- [19] **DICOM**. Disponível em: <<https://www.dicomstandard.org/patent/>>. Acesso em: 20 jun. 2019.
- [20] Hafidz Zulkifli. **"Understanding Learning Rates and How It Improves Performance in Deep Learning"**. *Towards Data Science*. 15. Fev. 2019.
- [21] Srivastava, Nitish; C. Geoffrey Hinton; Alex Krizhevsky; Ilya Sutskever; Ruslan Salakhutdinov. **"Dropout: A Simple Way to Prevent Neural Networks from overfitting"**. *Journal of Machine Learning Research*. 2014.
- [22] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. **High-performance neural networks for visual object classification**. Arxiv preprint arXiv:1102.0183, 2011
- [23] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le, **Auto Augment: Learning Augmentation Policies From Data**, arXiv:1805.09501, 2018