

**NICOLI PINHEIRO DE ARAUJO**

**ESTIMAÇÃO INTELIGENTE DE IDADE DE TELESPECTADORES PARA  
APLICAÇÕES DE SUGESTÃO DE CONTEÚDO EM *SMART* TVS**

Trabalho de Conclusão de Curso apresentado  
à banca avaliadora do Curso de Engenharia  
de Computação, da Escola Superior de  
Tecnologia, da Universidade do Estado do  
Amazonas, como pré-requisito para obtenção  
do título de Engenheira de Computação.

Orientador(a): Profa. Dra. Elloá Barreto Guedes da Costa

Manaus – Novembro – 2018

**Universidade do Estado do Amazonas - UEA**

**Escola Superior de Tecnologia - EST**

*Reitor:*

**Cleinaldo de Almeida Costa**

*Vice-Reitor:*

**Cleto Cavalcante de Souza Leal**

*Diretor da Escola Superior de Tecnologia:*

**Roberto Higino Pereira Da Silva**

*Coordenador do Curso de Engenharia de Computação:*

**Salvador Ramos Bernardino Da Silva**

*Coordenador da Disciplina Projeto Final:*

**Marcia Sampaio Lima**

*Banca Avaliadora composta por:*

*Data da Defesa: 30/11/2018.*

**Profa. Dra. Elloá Barreto Guedes da Costa (Orientadora)**

**Prof. M.Sc. Mario Augusto Bessa De Figueiredo**

**Prof. Dr. Carlos Mauricio Serodio Figueiredo**

### **CIP – Catalogação na Publicação**

**L864a**

ARAUJO, Nicoli Pinheiro de

Estimação Inteligente de Idade de Telespectadores para Aplicações de Sugestão de Conteúdo em *Smart TVs* / Nicoli Araujo; [orientada por] Profa. Dra. Elloá Barreto Guedes da Costa – Manaus: UEA, 2018.

**240 p.: il.; 30cm**

Inclui Bibliografia

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação).  
Universidade do Estado do Amazonas, 2018.

CDU: \_\_\_\_\_

NICOLI PINHEIRO DE ARAUJO

**ESTIMAÇÃO INTELIGENTE DE IDADE DE TELESPECTADORES PARA  
APLICAÇÕES DE SUGESTÃO DE CONTEÚDO EM *SMART* TVS**

Trabalho de Conclusão de Curso apresentado  
à banca avaliadora do Curso de Engenharia  
de Computação, da Escola Superior de  
Tecnologia, da Universidade do Estado do  
Amazonas, como pré-requisito para obtenção  
do título de Engenheiro de Computação.

**Aprovado em: / /2018**

BANCA EXAMINADORA

---

**Profa. Dra. Elloá Barreto Guedes da Costa**  
*UNIVERSIDADE DO ESTADO DO AMAZONAS*

---

**Prof. Mario Augusto Bessa De Figueiredo, M.Sc.**  
*UNIVERSIDADE DO ESTADO DO AMAZONAS*

---

**Prof. Carlos Mauricio Serodio Figueiredo, Dr.**  
*UNIVERSIDADE DO ESTADO DO AMAZONAS*



# Resumo

Este trabalho apresenta uma proposta para estimação de idade de telespectadores para aplicações de sugestão de conteúdo em *Smart TVs* utilizando técnicas de *deep learning*. Tal ferramenta pode ser utilizada de diversas maneiras, incluindo para facilitar a coleta de informações que contribuam para melhor experiência de provimento de conteúdo, criação e controle de configurações personalizadas e para a implementação de estratégias de controle parental mais eficientes.

**Palavras Chave:** Estimação de Idade, Redes Neurais Artificiais Convolucionais, *Deep Learning*.

# ***Abstract***

This work presents a proposal for estimating the age of viewers for content suggestion applications on *Smart* TVs using machine learning techniques. Such a tool can be used in a variety of ways, including to facilitate the collection of information that contributes to a better content delivery experience, to the creation and control of custom settings, and to the implementation of more efficient parental control strategies.

***Keywords:*** *Age Estimation, Convolutional Neural Networks, Deep Learning.*

# Agradecimentos

(...) Agradeço também à Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) que, por meio do Projeto PROTI Pesquisa 11/2017, colaborou para a consolidação da infraestrutura física e tecnológica do Laboratório de Sistemas Inteligentes da Escola Superior de Tecnologia da Universidade do Estado do Amazonas. Este trabalho de conclusão de curso é um dos produtos deste projeto, pois foi desenvolvido no referido laboratório, fez uso dos recursos computacionais ali disponíveis e foi melhorado graças às discussões e interações com o grupo de pesquisa nele sediado.

**Epígrafe**

*Sua frase.*

Autor da frase



# Sumário

Lista de Tabelas	ix
Lista de Figuras	xi
<b>1 Resultados e Discussão</b>	<b>1</b>
1.1 Abordagem 1: LeNet e AlexNet com Imagens Normalizadas . . . . .	1
1.2 Abordagem 2: Introduzindo <i>Data Augmentation</i> . . . . .	4
1.3 Abordagem 3: Introduzindo Equalização de Histograma . . . . .	7
1.4 Abordagem 4: Utilizando MAE para o Cálculo da Perda . . . . .	9
1.5 Abordagem 5: LeNet Apenas com Normalização da Entrada . . . . .	11
1.6 Abordagem 6: VGG-16 e Dados Normalizados . . . . .	12
1.7 Abordagem 7: VGG-16 com <i>Data Augmentation</i> e Equalização de Histograma .	13
1.8 Abordagem 8: VGG-16 com <i>Leaky ReLU</i> . . . . .	14
1.9 Abordagem 9: SqueezeNet . . . . .	14
1.10 Sumarizando os Resultados . . . . .	16

# Lista de Tabelas

1.1	Resultados do treino e teste dos modelos propostos na Abordagem 1. . . . .	4
1.2	Resultados do treino e teste dos modelos propostos na Abordagem 2. . . . .	6
1.3	Resultados do treino e teste dos modelos propostos na Abordagem 3. . . . .	8
1.4	Resultados do treino e teste dos modelos propostos na Abordagem 4. . . . .	10
1.5	Resultados do treino e teste dos modelos propostos na Abordagem 5. . . . .	12
1.6	Resultados do treino e teste dos modelos propostos na Abordagem 6. . . . .	13
1.7	Resultados do treino e teste dos modelos propostos na Abordagem 6. . . . .	13
1.8	Resultados do treino e teste dos modelos propostos na Abordagem 6. . . . .	14
1.9	Resultados do treino e teste dos modelos propostos na Abordagem 6. . . . .	16
1.10	Sumário dos resultados obtidos de todas as abordagens conduzidas. . . . .	16

# Lista de Figuras

1.1	Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 1.	2
1.2	Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem	
1.	.....	3
1.3	Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 2.	5
1.4	Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem	
2.	.....	6
1.5	Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 3.	8
1.6	Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem	
3.	.....	9
1.7	Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 4.	10
1.8	Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 5.	11
1.9	Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem	
6.	.....	12
1.10	Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem	
7.	.....	14
1.11	Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem	
8.	.....	15
1.12	Resultados do treinamento e teste da CNN SqueezeNet de acordo com a Abor-	
dagem 9.	.....	15

# Capítulo 1

## Resultados e Discussão

Considerando a estratégia descrita na solução proposta, os resultados da execução das CNNs aplicadas ao problema de estimação de idade a partir de uma imagem de face são apresentados a seguir. Estes resultados estão organizados segundo abordagens sequenciais, que contemplam desde as técnicas mais elementares, e que vão aumentando o grau de complexidade conforme uso de estratégias específicas da prática de DL visando obter melhores resultados.

### 1.1 Abordagem 1: LeNet e AlexNet com Imagens Normalizadas

A primeira abordagem de treinamento considerou o uso dos modelos LeNet e AlexNet de maneira canônica, isto é, tais como são definidos na literatura. Adotou-se as funções de ativação não-lineares *ReLU* e *Leaky ReLU* por serem simples de calcular e por satisfazerem os critérios de continuidade e diferenciação, requeridos pelo algoritmo de *backpropagation*, conforme discutido anteriormente na Seção ??.

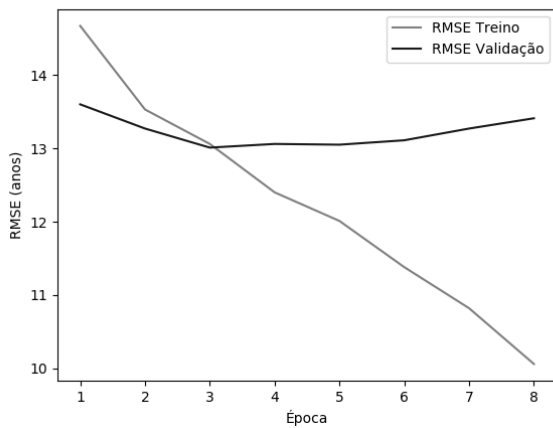
As imagens da base de dados foram normalizadas antes de serem apresentadas às redes, conforme a estratégia de pré-processamento previamente definida, isto é, todos os valores dos *pixels* componentes das imagens foram escalonados para o intervalo  $[0, 1]$  por meio de uma divisão por 255. Conforme mencionado, a prévia normalização das imagens antes da apresentação

às CNNs colabora para uma melhor execução do gradiente descendente e diminui a variância nos pesos.

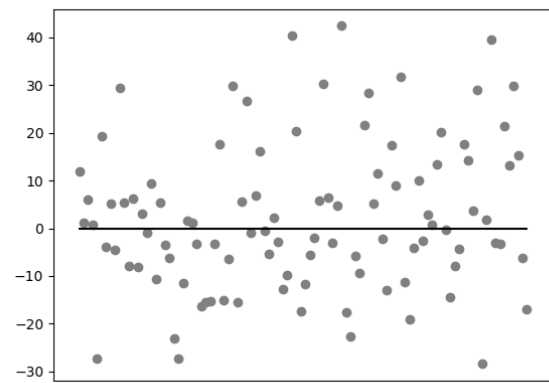
Os treinamentos destas duas arquiteturas duraram aproximadamente 16 e 12 horas respectivamente, em uma instância do Google Compute Engine com 4 CPUs virtuais e 15 GB de RAM. Os gráficos de treinamento e as retas zero obtidas a partir da apresentação do conjunto de teste aos modelos treinados podem ser vistos na Figura 1.1.

Figura 1.1: Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 1.

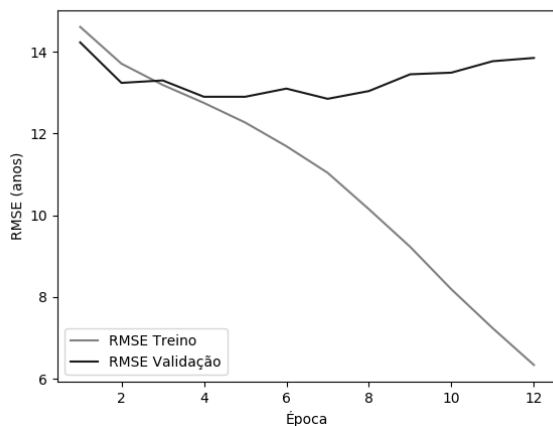
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



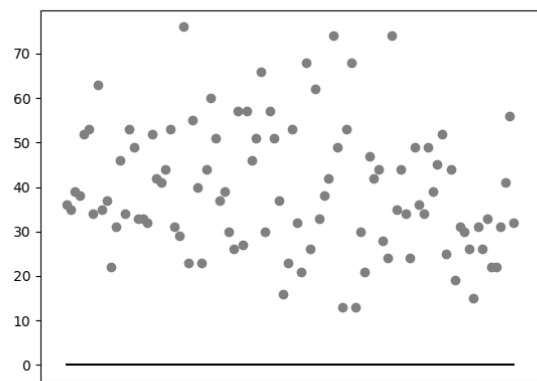
(b) Reta-0 LeNet *ReLU*.



(c) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *Leaky ReLU*.



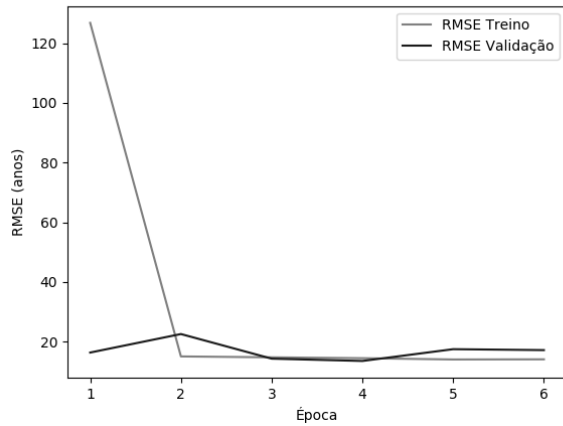
(d) Reta-0 LeNet *Leaky ReLU*.



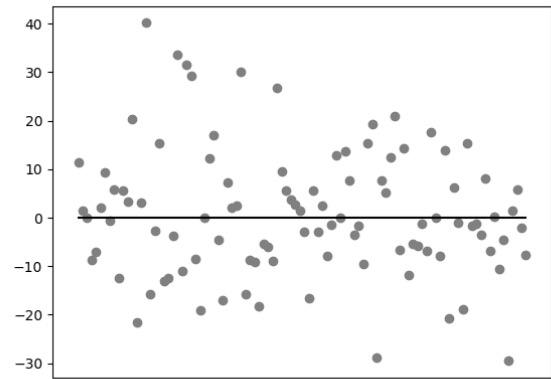
Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.1. De maneira geral, ambas as

Figura 1.2: Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem 1.

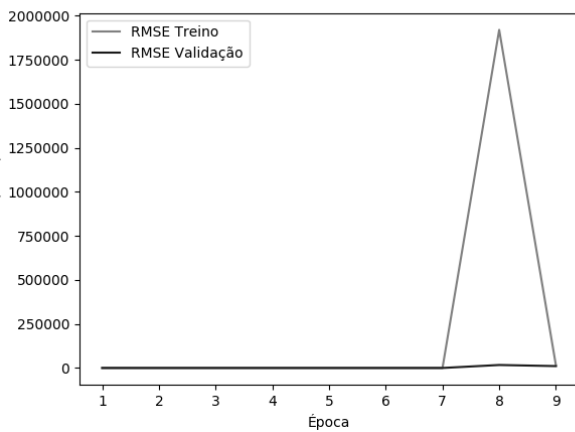
(a) Treinamento AlexNet *ReLU*.



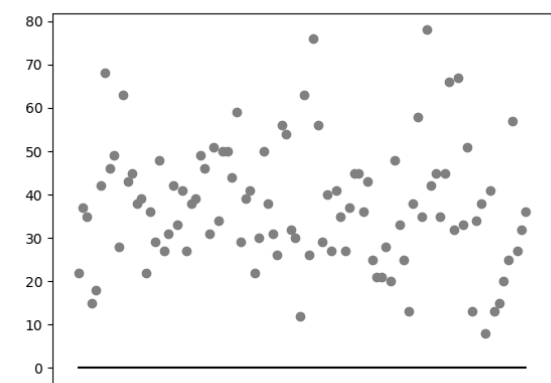
(b) Reta-0 AlexNet *ReLU*.



(c) Treinamento AlexNet *Leaky ReLU*.



(d) Reta-0 AlexNet *Leaky ReLU*.



redes obtiveram melhor desempenho com o uso da função de ativação *ReLU* em que, em particular, a arquitetura LeNet apresento resultados ligeiramente superiores. Apesar disso, nota-se que o treinamento não seguiu de maneira estável, com grande variação entre as métricas de treinamento e validação, ocasionando *early stopping* após um número relativamente curto de épocas. Tal cenário é sugestivo de *overfitting*, especialmente mais evidentes nos casos em que se utilizou a função de ativação *Leaky ReLU*.

Tabela 1.1: Resultados do treino e teste dos modelos propostos na Abordagem 1.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>ReLU</i>	4	10.53	13.55
LeNet	<i>Leaky ReLU</i>	8	38.33	40.82
AlexNet	<i>ReLU</i>	5	11.03	13.76
AlexNet	<i>Leaky ReLU</i>	5	39.27	41.97

## 1.2 Abordagem 2: Introduzindo *Data Augmentation*

A abordagem anterior consistiu essencialmente da utilização dos modelos tal como foram definidos e com uma simples operação de adequação dos dados de entrada por meio de normalização. Porém, em problemas de Visão Computacional, é comum aplicar técnicas de *data augmentation* com vistas a aumentar artificialmente o conjunto de dados, fazendo com que o modelo, em sua fase de treinamento, não seja exposto à mesma entrada em mais de uma ocasião. Esta estratégia de regularização colabora na mitigação do *overfitting* e costuma promover uma melhor generalização (CHOLLET, 2017).

As técnicas de *data augmentation* consideradas foram a rotação entre 0 e 20 graus no sentido horário ou anti-horário, zoom de 0.8 a 1.2 vezes, inversão horizontal com probabilidade de ocorrência de 0.5 ou translação com probabilidade igual a 0.2.

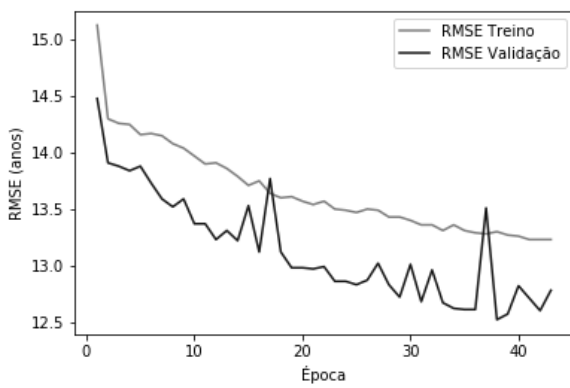
Desta feita, após o aumento artificial do conjunto de dados, partiu-se então para o treinamento e teste dos modelos conforme a metodologia adotada, considerando as arquiteturas LeNet e AlexNet e as funções de ativação *ReLU* e *Leaky ReLU*. Os gráficos das métricas de desempenho coletadas durante o treinamento e a reta-0 obtida a partir dos dados de teste em cada uma destas quatro configurações são ilustrados nas Figuras 1.3 e 1.4.

De maneira análoga, as métricas de desempenho coletadas encontram-se detalhadas na Tabela 1.2. Nota-se que o número de épocas no treinamento foi maior que a abordagem anterior, indicando que houve um cenário mais favorável para o aprendizado dos padrões nos dados. De maneira geral, as métricas obtidas não fornecem uma evidência forte de que esta segunda abordagem produz resultados mais significativos que a primeira mas, no caso da CNN AlexNet

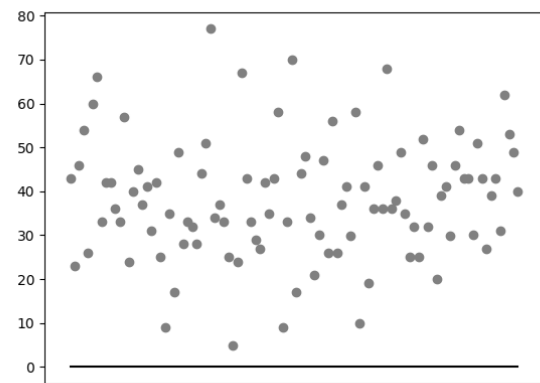
com *ReLU*, os resultados foram comparáveis. O efeito positivo esperado pelo *data augmentation* não se mostrou tão evidente quanto se esperava inicialmente. Porém, isto pode acontecer em razão dos valores dos hiperparâmetros e da necessidade de melhor pré-processamento das imagens antes da apresentação às CNNs, o que motivou a realização da abordagem a seguir.

Figura 1.3: Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 2.

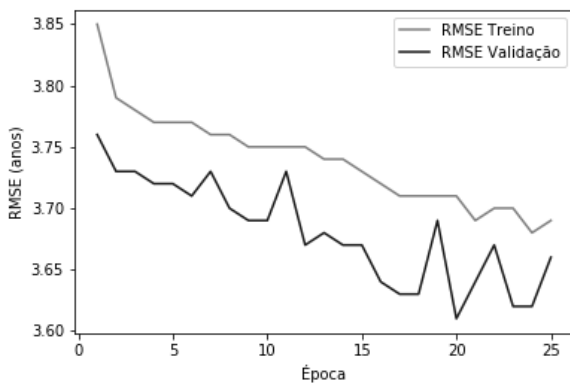
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



(b) Reta-0 LeNet *ReLU*.



(c) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *Leaky ReLU*.



(d) Reta-0 LeNet *Leaky ReLU*.

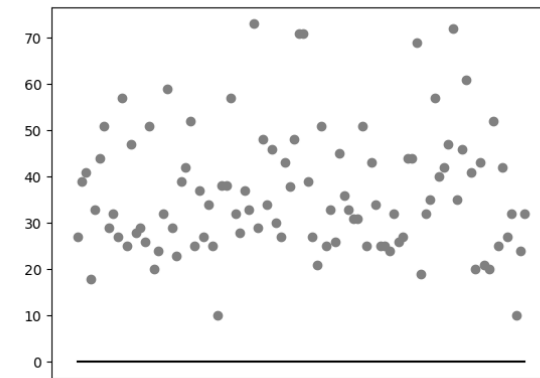
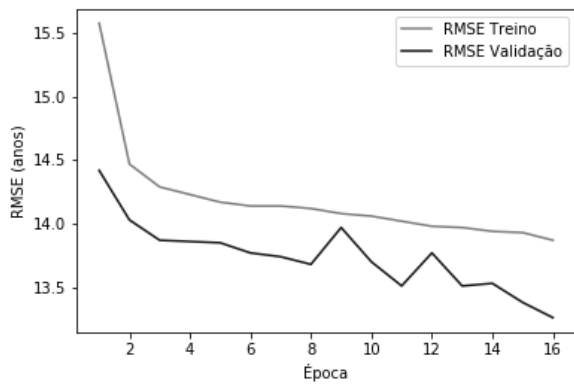


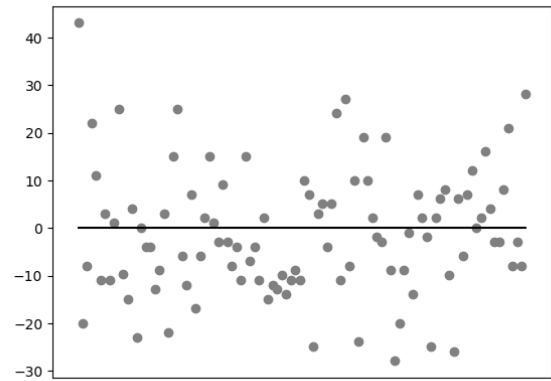


Figura 1.4: Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem 2.

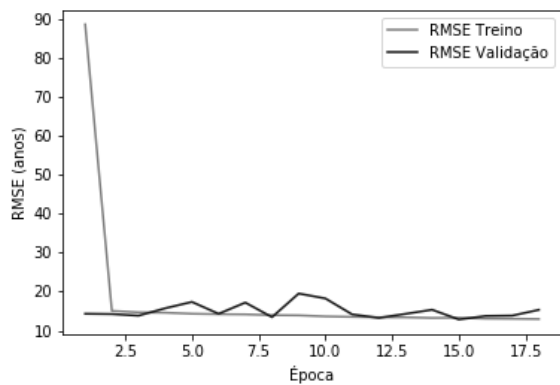
(a) Treinamento AlexNet *ReLU*.



(b) Reta-0 AlexNet *ReLU*.



(c) Treinamento AlexNet *Leaky ReLU*.



(d) Reta-0 AlexNet *Leaky ReLU*.

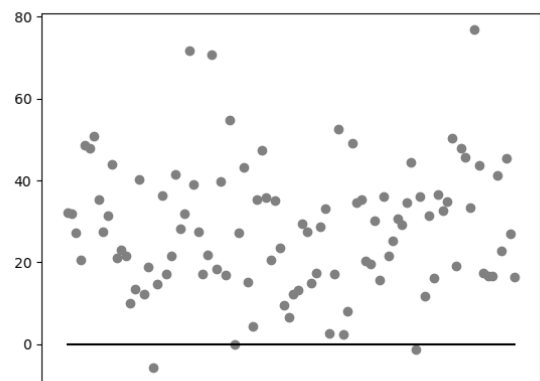


Tabela 1.2: Resultados do treino e teste dos modelos propostos na Abordagem 2.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>ReLU</i>	39	37.85	40.27
LeNet	<i>Leaky ReLU</i>	21	38.50	41.06
AlexNet	<i>ReLU</i>	16	11.59	14.59
AlexNet	<i>Leaky ReLU</i>	16	28.06	31.81

## 1.3 Abordagem 3: Introduzindo Equalização de Histograma

A terceira abordagem utilizou as imagens da base de dados normalizadas e técnicas de *data augmentation* previamente mencionadas. Além disto, visando melhorar as métricas de desempenho, introduziu-se o processo de equalização das imagens por histograma, que ajusta o contraste da imagem utilizando o histograma de cores. Conforme mencionado na Seção ??, este método aumenta o contraste global de imagens, especialmente quando os dados úteis da imagem são representados por cores próximas. No contexto da detecção de idade por meio da imagem da face de determinado indivíduo, a equalização por histograma reforça marcas de expressões e outras imperfeições (ACHARYA; RAY, 2005).

Considerado uma nova equalização do conjunto de dados segundo o histograma, somado à normalização anterior e à aplicação de *data augmentation*, consolidou-se então o treinamento e teste das redes LeNet e AlexNet com função de ativação *ReLU* e *Leaky ReLU* segundo a terceira abordagem. Os gráficos de treinamento e a reta-0 obtidos encontram-se nas Figuras 1.5 e 1.6, respectivamente. A partir destes gráficos, observa-se um treinamento mais consistente, com queda progressiva da perda à medida que se avançam as épocas. Apesar disso, nota-se também que houve retrocessos na métrica de perda da validação ao longo do treinamento, o que promoveu *early stopping*.

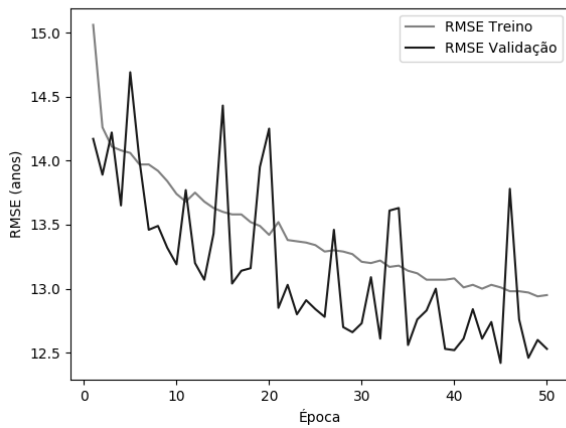
Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.3. É interessante notar que o aumento nas épocas do treinamento não foi efetivo em obter melhores métricas de testes. Tem-se que, embora as CNNs aprendam por mais épocas que nos cenários anteriores, este aprendizado é menor para fins de generalização. Apesar de estar havendo um esforço de pré-processamento por meio de *data augmentation* e de equalização por histograma, seguindo práticas tipicamente adotadas em trabalhos desta área (CHOLLET, 2017), as melhorias promovidas por estas estratégias são de cunho heurístico, podendo ser boas em determinados cenários, mas não garantidamente em todos, tal como ocorreu no contexto em questão.

Tabela 1.3: Resultados do treino e teste dos modelos propostos na Abordagem 3.

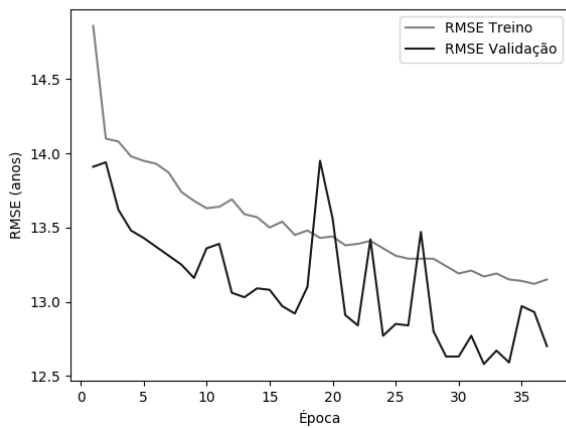
Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>ReLU</i>	46	38.66	41.20
LeNet	<i>Leaky ReLU</i>	38	38.26	40.85
AlexNet	<i>ReLU</i>	7	13.10	15.88
AlexNet	<i>Leaky ReLU</i>	18	35.25	38.04

Figura 1.5: Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 3.

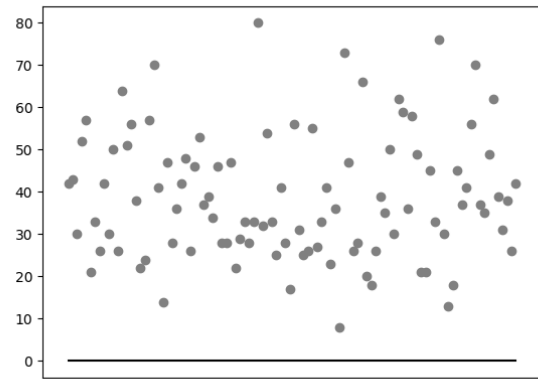
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



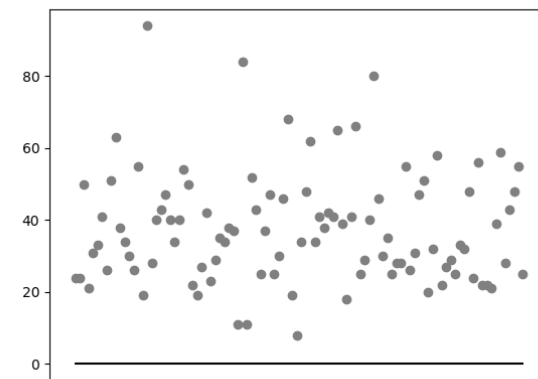
(c) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *Leaky ReLU*.



(b) Reta-0 LeNet *ReLU*.



(d) Reta-0 LeNet *Leaky ReLU*.



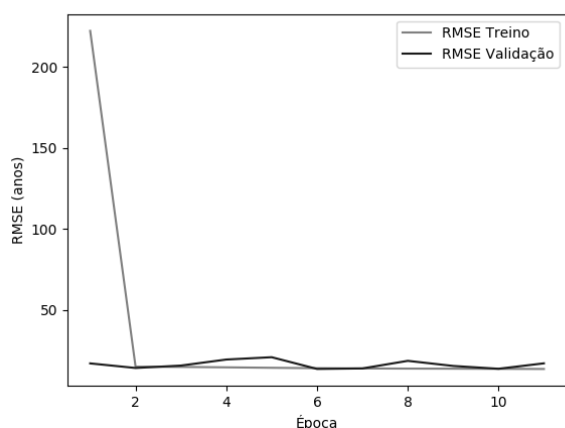
## 1.4 Abordagem 4: Utilizando MAE para o Cálculo da Perda

A análise dos gráficos de treinamento das redes anteriores levou à suposição de que a métrica utilizada para a atualização dos pesos RMSE estivesse trazendo instabilidade para o treinamento. Desta maneira, esta abordagem considera a rede com melhor desempenho verificado até então, LeNet com função de ativação *ReLU* sujeita à *data augmentation*, imagens normalizadas e com

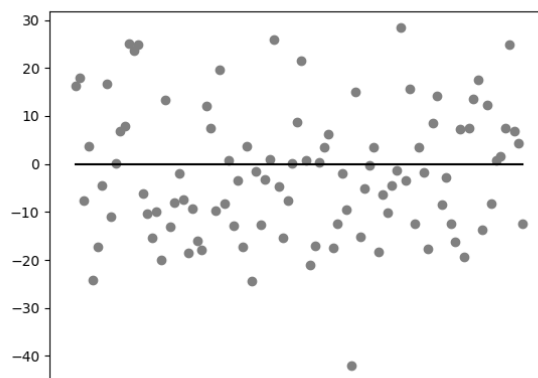
checar se a rede é esta mesmo

Figura 1.6: Resultados do treinamento e teste da CNN AlexNet de acordo com a Abordagem 3.

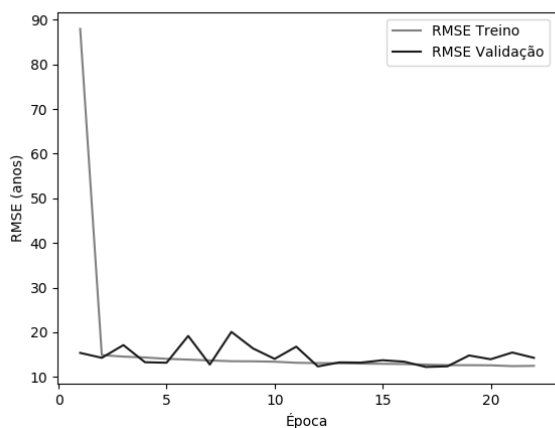
(a) RMSE de treinamento da arquitetura AlexNet utilizando funções de ativação *ReLU*.



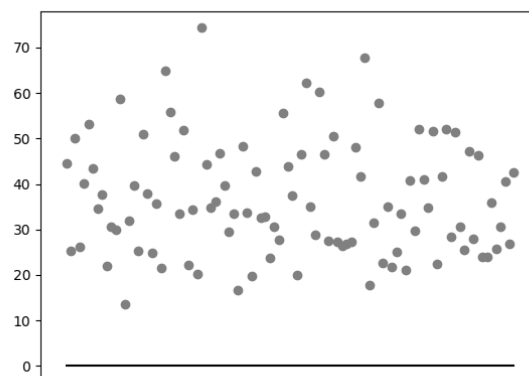
(b) Reta-0 AlexNet *ReLU*.



(c) RMSE de treinamento da arquitetura AlexNet utilizando funções de ativação *Leaky ReLU*.



(d) Reta-0 AlexNet *Leaky ReLU*.

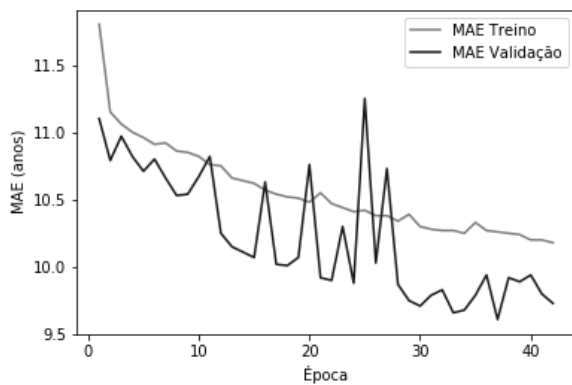


equalização por histograma, mas com a utilização do MAE para cálculo da perda.

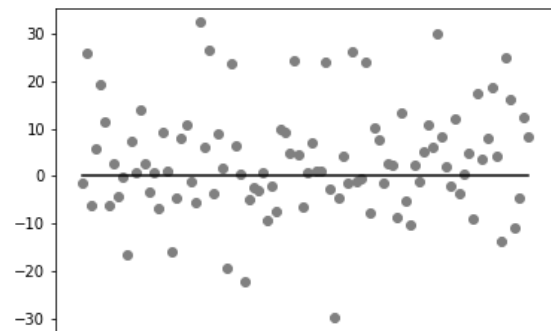
Esta abordagem foi considerada em virtude do treinamento com o otimizador *Adam* construir superfícies estocásticas de erro a partir da métrica de perda. Assim, após treinamento e teste, obteve-se os gráficos ilustrados nas Figuras 1.7.

Figura 1.7: Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 4.

(a) MAE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



(b) Reta-0 LeNet *ReLU*.



Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.4.

Tabela 1.4: Resultados do treino e teste dos modelos propostos na Abordagem 4.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>Leaky ReLU</i>	38	9.98	12.91

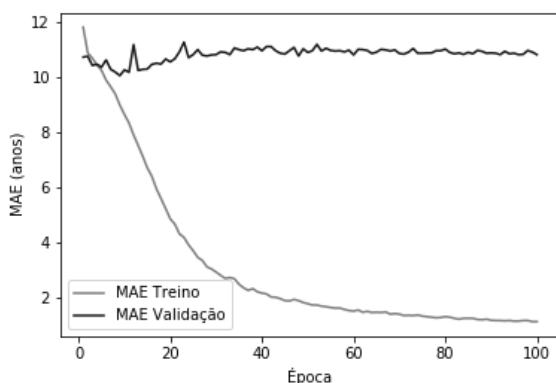
Os resultados obtidos evidenciam melhores métricas de desempenho até então, oferecendo diretrizes positivas a cerca das estratégias adotadas. Apesar da identificação de uma rede satisfatória até então, fica o questionamento se as melhorias obtidas em si foram introduzidas pelas heurísticas de pré-processamento ou se são intrínsecas do modelo adotado. Como a equalização por histograma e os processos de *data augmentation* introduzem ônus de pré-processamento aos dados, desejou-se verificar se o desempenho positivo desta rede seria mantido mesmo sem tais práticas, o que motivou a realização da abordagem seguinte.

## 1.5 Abordagem 5: LeNet Apenas com Normalização da Entrada

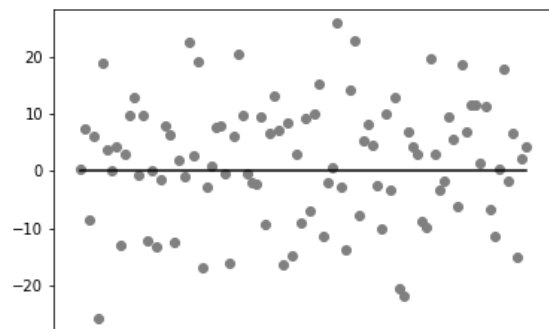
A quinta abordagem adotou a rede com melhor desempenho obtido até o momento, a LeNet com função de ativação *ReLU*, treinada apenas com imagens da base de dados normalizadas, mas sem equalização de histograma de cores nem tampouco técnicas de *data augmentation*. Seguiu-se utilizando a métrica MAE para o cálculo da perda e da atualização dos pesos. Buscando garantir maior estabilidade nas métricas de desempenho durante o treinamento, aumentou-se o tamanho do batch para 128, com vistas a estimar melhor a superfície de erro à medida que o algoritmo de otimização percorre o gradiente descendente.

Figura 1.8: Resultados do treinamento e teste da CNN LeNet de acordo com a Abordagem 5.

(a) MAE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



(b) Reta-0 LeNet *ReLU*.



Obedecendo ao método de validação cruzada *holdout*, os resultados desta abordagem encontram-se sintetizados na Tabela 1.5. É interessante notar que não houve melhorias na métrica de desempenho, pelo contrário, o erro aumentou em relação ao cenário anterior, sugerindo que as práticas de pré-processamento colaboram positivamente no melhor aprendizado das características da tarefa considerada. Uma vez que foram exploradas diferentes práticas de treinamento das redes LeNet e AlexNet, em que as melhorias alcançadas estagnaram após a quarta abordagem, partiu-se então para a avaliação de arquiteturas mais profundas aplicadas ao problema considerado, o que ensejou as abordagens a seguir.

Tabela 1.5: Resultados do treino e teste dos modelos propostos na Abordagem 5.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>ReLU</i>	9	10.09	13.04

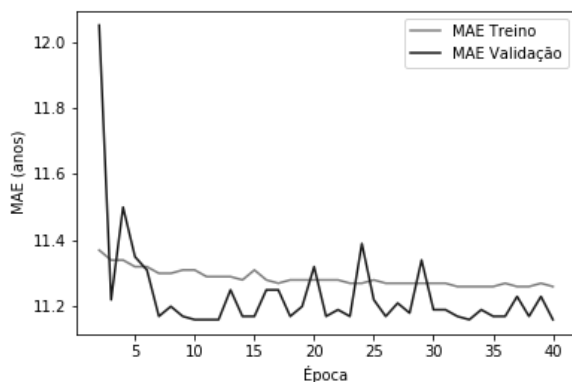
## 1.6 Abordagem 6: VGG-16 e Dados Normalizados

A sexta abordagem considerou a utilização da arquitetura VGG-16, previamente apresentada na Seção ???. Para utilização neste contexto, de maneira análoga aos cenários anteriores, removeu-se a camada de saída, tipicamente utilizada para fins de classificação, e adicionou-se uma camada densa com função de ativação *ReLU*. Embora tipicamente esta rede seja utilizada com *transfer learning* do conjunto de dados ImageNet, optou-se por seguir a prática adotada nas demais redes, com inicialização aleatória dos pesos e treinamento integral com o conjunto de dados apenas sujeito à normalização.

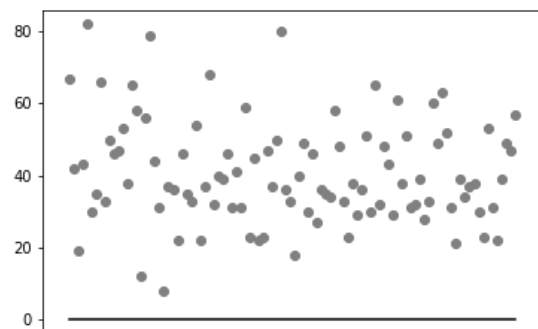
Em decorrência do treinamento e teste nos mesmos moldes dos cenários anteriores, obteve-se os gráficos ilustrados na Figura 1.9. As métricas de desempenho obtidas são elencadas na Tabela 1.6.

Figura 1.9: Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem 6.

(a) MAE de treinamento da arquitetura VGG-16 utilizando funções de ativação *ReLU*.



(b) Reta-0 VGG-16 *ReLU*.



É interessante notar que a rede VGG-16 é mais profunda em termos de camadas que as redes LeNet e AlexNet, o que impacta em uma maior quantidade de parâmetros livres a serem calculados e que acaba por exigir mais recursos computacionais para seu treino. Segundo as

Tabela 1.6: Resultados do treino e teste dos modelos propostos na Abordagem 6.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
VGG-16	<i>ReLU</i>	11	40.90	38.35

ideias de DL, imagina-se que uma rede com um número maior de camadas hierárquicas seja capaz de produzir melhores representações, impactando positivamente no aprendizado da tarefa. Apesar disso, teve desempenho aquém. Uma suposição para este resultado é que a rede sofreu *dying ReLU problem*, pois apenas produziu saídas iguais a zero, maximizando o erro aferido no teste. Para contornar este problema, esta arquitetura foi novamente testada, mas introduzindo também *data augmentation* e normalização por histograma de frequência.

Esta rede previa o valor 37.047264 para todas as imagens de entrada.

## 1.7 Abordagem 7: VGG-16 com *Data Augmentation* e Equalização de Histograma

A rede VGG-16 utilizada nesta abordagem foi instanciada e treinada com os mesmos parâmetros descritos na Abordagem 6, porém, passou-se a utilizar *data augmentation* e equalização de histograma, obedecendo às mesmas práticas de configuração utilizada nas Abordagens 2 a 4 para as redes LeNet e AlexNet e dos resultados positivos observados.

Os resultados obtidos do treinamento e teste encontram-se ilustrados nos gráficos do treinamento e reta-0 da Figura 1.12 e detalhados na Tabela 1.9. É interessante notar que as técnicas para mitigar *overfitting* não foram efetivas para contornar o *dying ReLU problem* previamente verificado, impactando em métricas análogas de erro.

Tabela 1.7: Resultados do treino e teste dos modelos propostos na Abordagem 6.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
VGG-16	<i>ReLU</i>	23	40.99	38.39

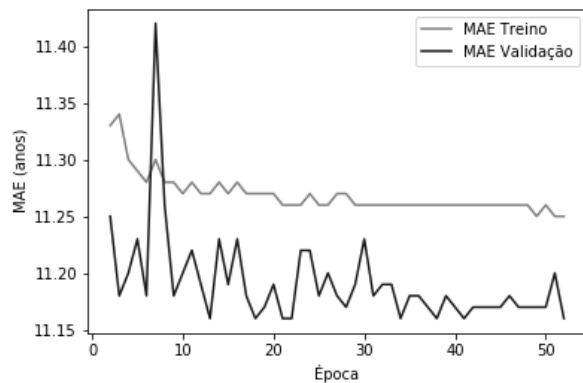
Esta rede previa apenas o valor 37.01601 para todas as entradas.

Veja que ao longo do texto há termos diferentes para denotar esta mesma coisa: normalização por histograma, equalização por histograma, normalização por histograma de frequência e equalização de histograma. Sugiro padronizar, a banca pegará no pé.

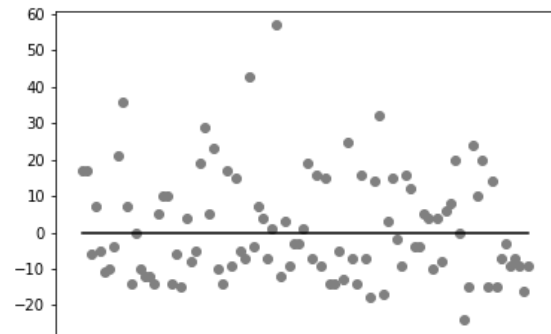


Figura 1.10: Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem 7.

(a) MAE de treinamento da arquitetura VGG-16 utilizando funções de ativação *ReLU*.



(b) Reta-0 VGG-16 *ReLU*.



## 1.8 Abordagem 8: VGG-16 com *Leaky ReLU*

A abordagem agora considerada tenta endereçar de maneira mais objetiva a superação do *dying ReLU problem*, pois considera a adoção da função de ativação *Leaky ReLU* que não está sujeita à este fenômeno. Além disto, tentou-se utilizar uma taxa de aprendizado inicial maior, igual a 0.003, tentando favorecer uma caminhada mais estável no gradiente descendente em função do método de otimização utilizado. É importante salientar que, embora possa implicar em uma melhor estabilidade na eventual convergência, esta taxa de aprendizado torna o treinamento consideravelmente mais demorado.

Tabela 1.8: Resultados do treino e teste dos modelos propostos na Abordagem 6.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
VGG-16	<i>ReLU</i>	23	40.99	38.39

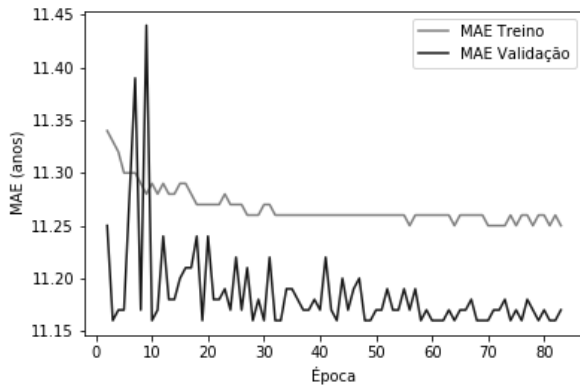
Esta rede previa apenas a média 36.999367 para todas as entradas.

## 1.9 Abordagem 9: SqueezeNet

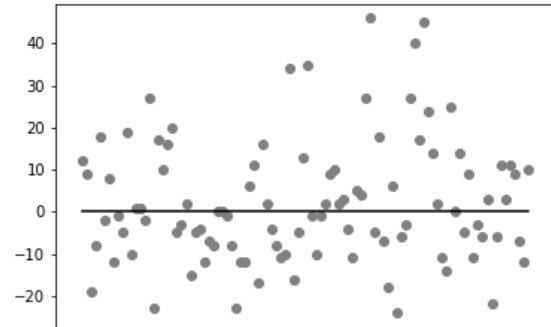
A última abordagem considerada neste trabalho, buscando sempre a perspectiva de melhores métricas e uma boa adequação ao contexto final de utilização do modelo, tratou do treinamento

Figura 1.11: Resultados do treinamento e teste da CNN VGG-16 de acordo com a Abordagem 8.

(a) MAE de treinamento da arquitetura VGG-16 utilizando funções de ativação *Leaky ReLU*.



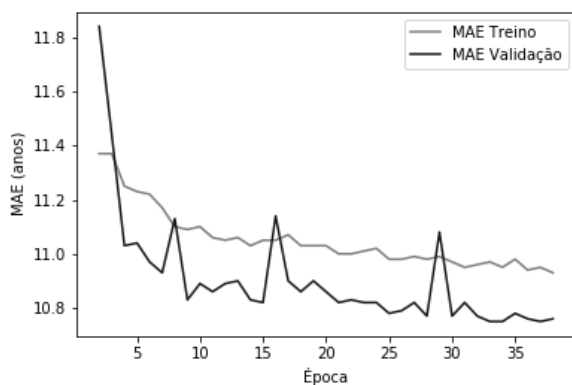
(b) Reta-0 VGG-16 *Leaky ReLU*.



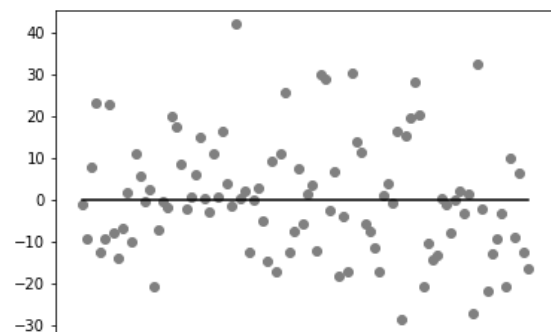
e teste da rede SqueezeNet. Dada a proposição recente deste modelo na literatura, ainda há poucos guias práticos sobre sugestões de uso e inicialização. Assim, esta arquitetura foi mantida tal como inicialmente proposta, mas considerou-se o uso de *data augmentation* e equalização por histograma, objetivando os mesmos ganhos verificados nas abordagens anteriores melhor sucedidas até então.

Figura 1.12: Resultados do treinamento e teste da CNN SqueezeNet de acordo com a Abordagem 9.

(a) MAE de treinamento da arquitetura SqueezeNet utilizando funções de ativação *ReLU*.



(b) Reta-0 LeNet *ReLU*.



(...) Esta rede foi capaz de gerar valores diferenciados para cada imagem de entrada, ou seja, não ficou presa em algum valor para todas as entradas.

Tabela 1.9: Resultados do treino e teste dos modelos propostos na Abordagem 6.

Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
VGG-16	<i>ReLU</i>	23	40.99	38.39

## 1.10 Sumarizando os Resultados

Considerando todas as abordagens conduzidas, a Tabela 1.10 sintetiza todos os treinamentos e testes realizados e os resultados obtidos da métrica de desempenho MAE adotada para esta tarefa. Pode-se identificar então que, para a tarefa elencada, a rede com melhor desempenho observado foi a X sujeita à treinamento com \_\_\_\_\_

Tabela 1.10: Sumário dos resultados obtidos de todas as abordagens conduzidas.

Abordagem	Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
1	LeNet	<i>ReLU</i>	4	10.53	13.55
1	LeNet	<i>Leaky ReLU</i>	8	38.33	40.82
1	AlexNet	<i>ReLU</i>	5	11.03	13.76
1	AlexNet	<i>Leaky ReLU</i>	5	39.27	41.97
2	LeNet	<i>ReLU</i>	39	37.85	40.27
2	LeNet	<i>Leaky ReLU</i>	21	38.50	41.06
2	AlexNet	<i>ReLU</i>	16	11.59	14.59
2	AlexNet	<i>Leaky ReLU</i>	16	28.06	31.81
3	LeNet	<i>ReLU</i>	46	38.66	41.20
3	LeNet	<i>Leaky ReLU</i>	38	38.26	40.85
3	AlexNet	<i>ReLU</i>	7	13.10	15.88
3	AlexNet	<i>Leaky ReLU</i>	18	35.25	38.04
4	LeNet	<i>Leaky ReLU</i>	38	9.98	12.91
5	LeNet	<i>ReLU</i>	9	10.09	13.04
6	VGG-16	<i>ReLU</i>	11	40.90	38.35
7	VGG-16	<i>ReLU</i>	23	40.99	38.39

Sumarizar a rede, a abordagem e suas características.

É interessante notar que o ajuste de parâmetros e hiperparâmetros para as redes neurais convolucionais segue as mesmas dificuldades das redes neurais *multilayer perceptron*, em que a superfície de erro é hiperdimensional e complexa, e que, quanto maior o número de pesos na rede, maior é o espaço de busca pelos parâmetros ideais, dificultando, por conseguinte, a tarefa de encontrá-los (BRAGA; CARVALHO; LUDERMIR, 2007). No caso das CNNs, em

especial, soma-se isto ao fato do aprendizado destas redes ser naturalmente mais oneroso em virtude do número de operações necessárias para realização das convoluções e do algoritmo de *backpropagation*.

# Referências Bibliográficas

ACHARYA, T.; RAY, A. K. *Image processing: principles and applications*. [S.l.]: John Wiley & Sons, 2005.

BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. 2. ed. Rio de Janeiro: LTC, 2007.

CHOLLET, F. *Deep Learning with Python*. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2017. ISBN 1617294438, 9781617294433.