

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

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 para a resolução de problemas práticos.

1.1 Abordagem 1: LeNet e AlexNet com Imagens Normalizadas

A primeira abordagem de treinamento considerou o uso dos modelos 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 ??.

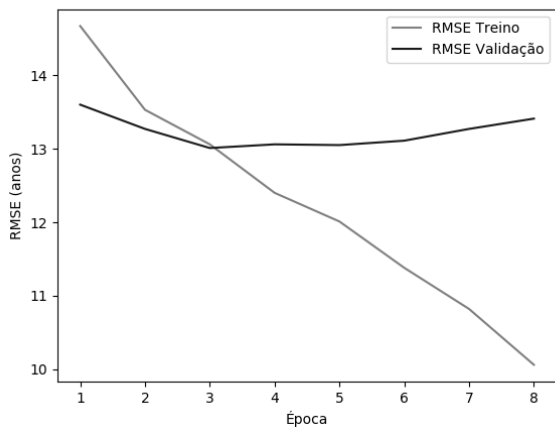
As imagens da base de dados foram normalizadas antes de serem apresentadas às redes. Todos os valores dos pixels componentes das imagens foram escalonados para o intervalo $[0, 1]$ por meio de uma divisão por 255. A prévia normalização das imagens antes da apresentação às CNNs colabora para uma melhor execução do gradiente descendente e diminui a variância

nos pesos.

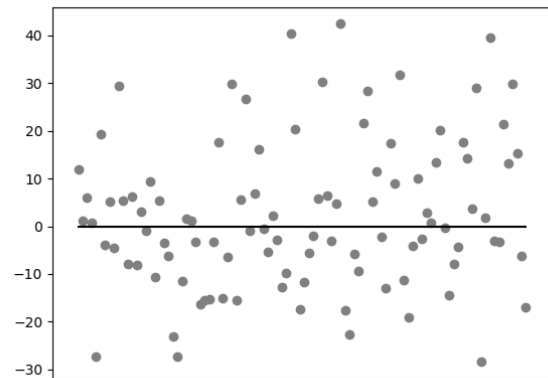
Os treinamentos destas duas arquiteturas duraram aproximadamente 16 e 12 horas respectivamente, em uma instância do Google Compute Engine com 4 CPUs virtuais e 15 GB de RAM. Os gráficos de treinamento e as retas zero obtidas a partir da apresentação do conjunto de teste aos modelos consolidados podem ser vistos na Figura 1.8. É possível notar que ambas as redes sofreram *overfitting* e obtiveram grande margem de erro. No entanto, a rede LeNet que utilizou *Leaky ReLU* como função de ativação obteve um desempenho mais satisfatório.

Figura 1.1: Resultados do treinamento e teste da CNN LeNet.

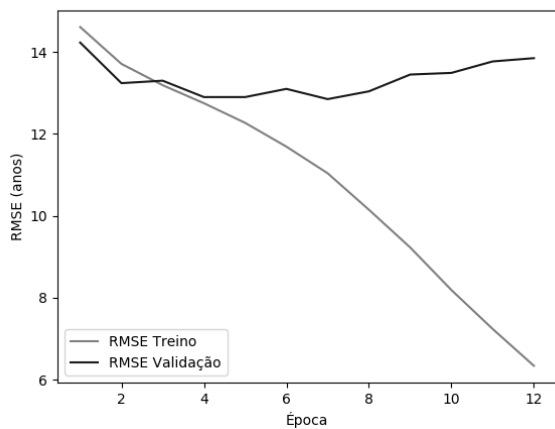
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



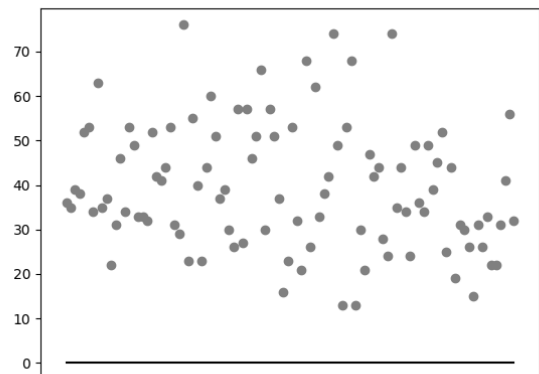
(b) Reta-0 LeNet *ReLU*.



(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

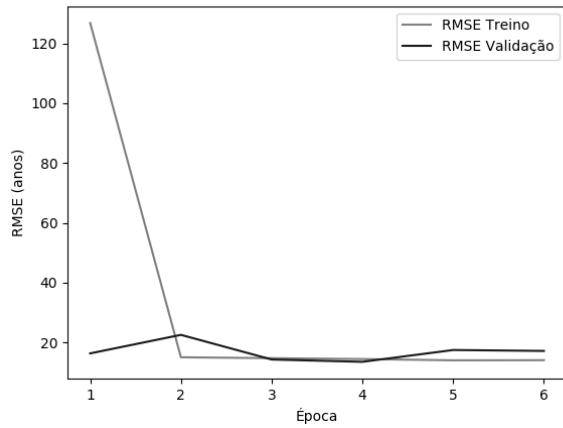
desta abordagem encontram-se sintetizados na Tabela 1.5.

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

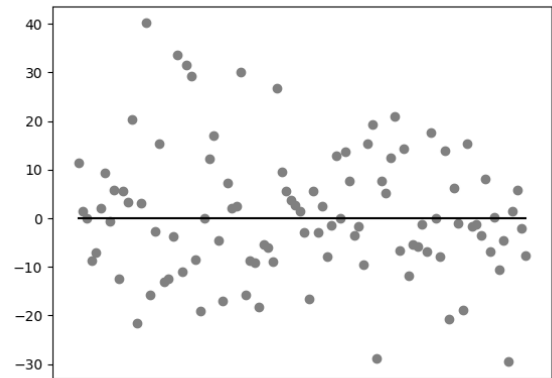
Rede	Função de ativação	Épocas	MAE Teste	RMSE Teste
LeNet	<i>ReLU</i>	4	10.53	13.55
LeNet	<i>Leaky ReLU</i>	8	38.33	40.82
AlexNet	<i>ReLU</i>	5	11.03	13.76
AlexNet	<i>Leaky ReLU</i>	5	39.27	41.97

Figura 1.2: Resultados do treinamento e teste da CNN AlexNet.

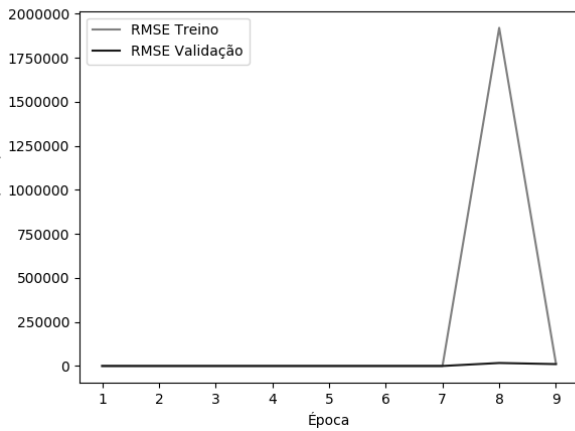
(a) Treinamento AlexNet *ReLU*.



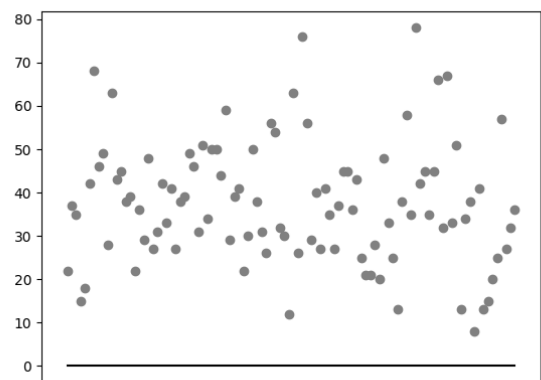
(b) Reta-0 AlexNet *ReLU*.



(c) Treinamento AlexNet *Leaky ReLU*.



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

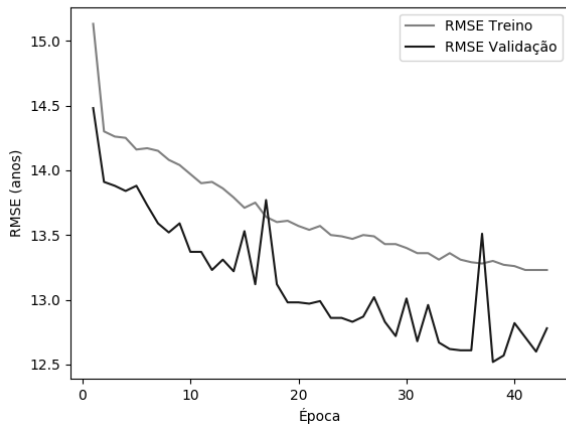


1.2 Abordagem 2

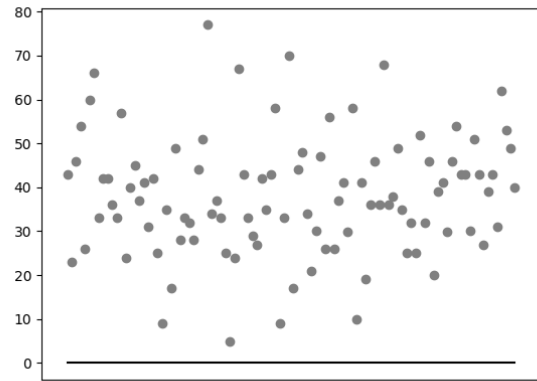
A segunda abordagem de treinamento adotada utilizou as imagens da base de dados normalizadas e com *data augmentation*, que inclui a probabilidade de uma rotação entre 0 e 20 graus, zoom de 0.8 a 1.2, chance de flip de 0.5, translate de 0.2.

Figura 1.3: Resultados do treinamento e teste da CNN LeNet.

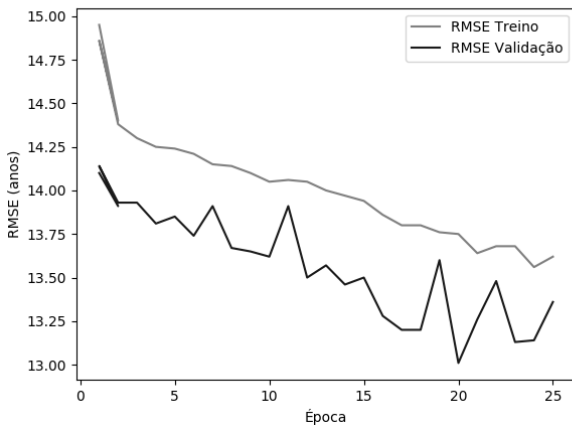
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



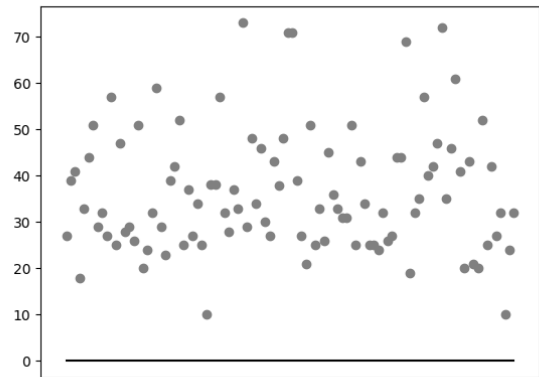
(b) Reta-0 LeNet *ReLU*.



(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 1.5.

Tabela 1.2: Resultados do treino e teste dos modelos propostos na Abordagem 1.

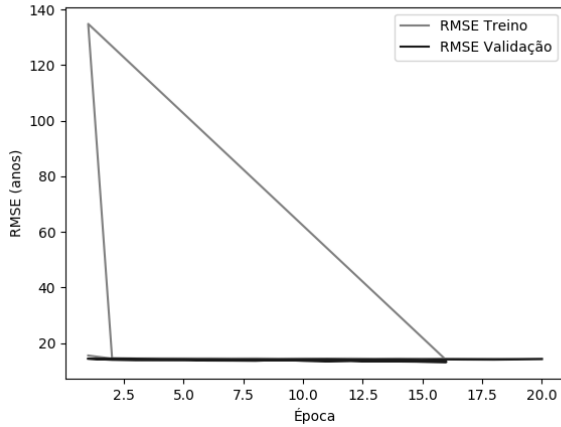
Rede	Função de ativação	Parâmetros	Épocas	Tempo de treinamento	MAE Teste	RMSE Teste
LeNet	<i>Leaky ReLU</i>	params	15	12 h	14.44	18.18
LeNet	<i>ReLU</i>	params	43	16 h	14.09	17.93
AlexNet	<i>ReLU</i>	58.286.145	10	15 h	38.63	41.22
AlexNet	<i>Leaky ReLU</i>	params	30	40 h	15.33	18.58

1.3 Abordagem 3

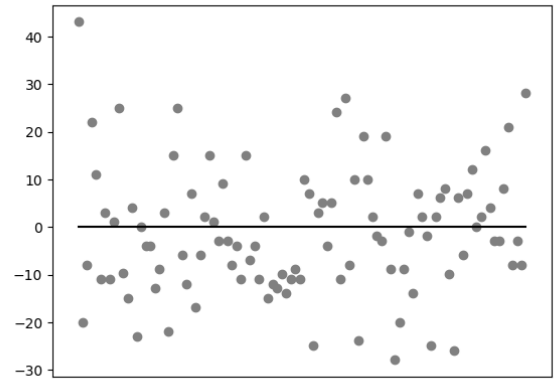
A terceira abordagem utilizou as imagens da base de dados normalizadas e com equalização de histograma de cores, além de técnicas de *data augmentation*, que inclui a probabilidade de uma rotação entre 0 e 20 graus, zoom de 0.8 a 1.2, chance de flip de 0.5, translate de 0.2.

Figura 1.4: Resultados do treinamento e teste da CNN AlexNet.

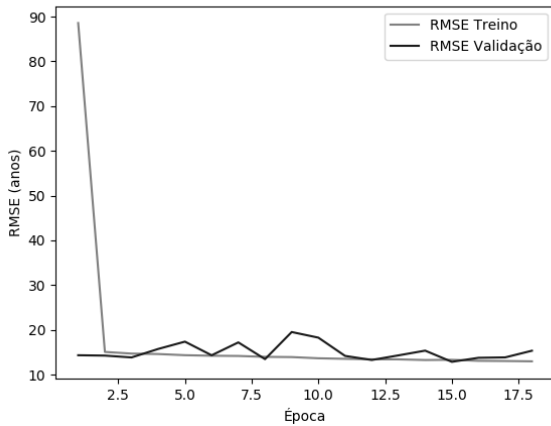
(a) Treinamento AlexNet *ReLU*.



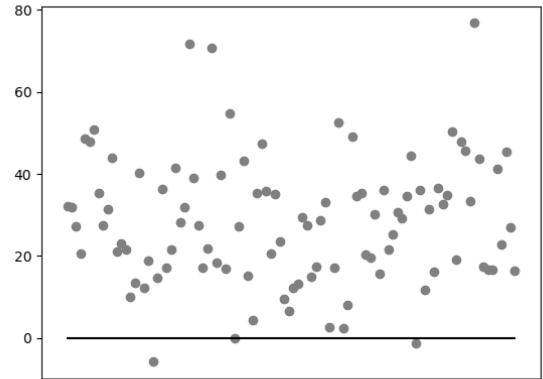
(b) Reta-0 AlexNet *ReLU*.



(c) Treinamento AlexNet *Leaky ReLU*.



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



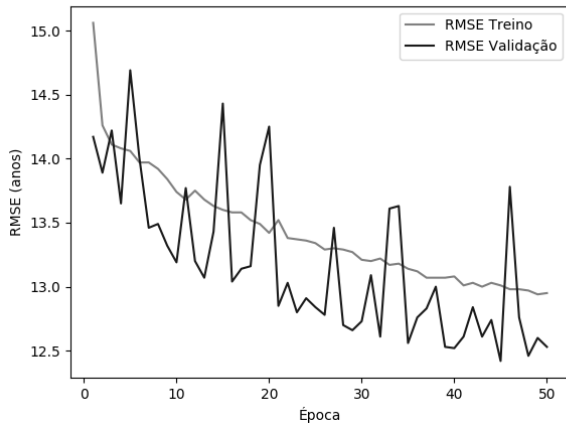
Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.5.

Tabela 1.3: Resultados do treino e teste dos modelos propostos na Abordagem 1.

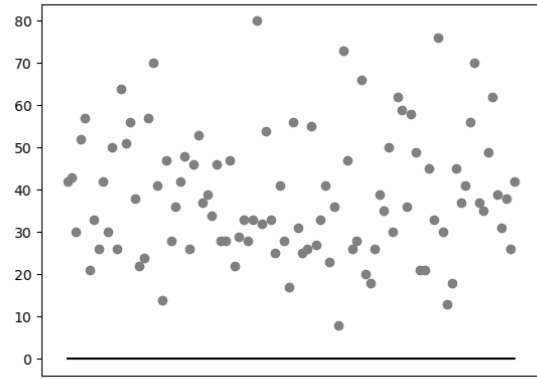
Rede	Função de ativação	Parâmetros	Épocas	Tempo de treinamento	MAE Teste	RMSE Teste
LeNet	<i>Leaky ReLU</i>	params	15	12 h	14.44	18.18
LeNet	<i>ReLU</i>	params	43	16 h	14.09	17.93
AlexNet	<i>ReLU</i>	58.286.145	10	15 h	38.63	41.22
AlexNet	<i>Leaky ReLU</i>	params	30	40 h	15.33	18.58

Figura 1.5: Resultados do treinamento e teste da CNN LeNet.

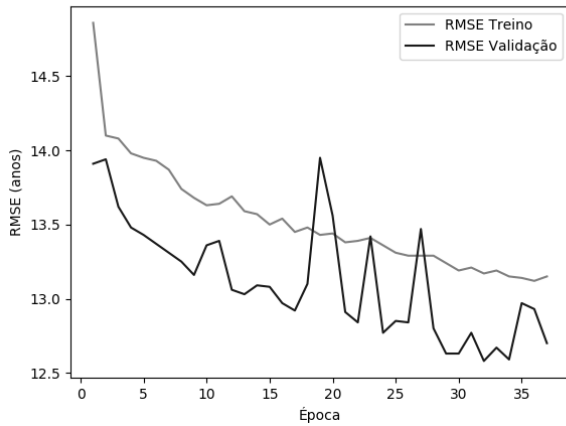
(a) RMSE de treinamento da arquitetura LeNet utilizando funções de ativação *ReLU*.



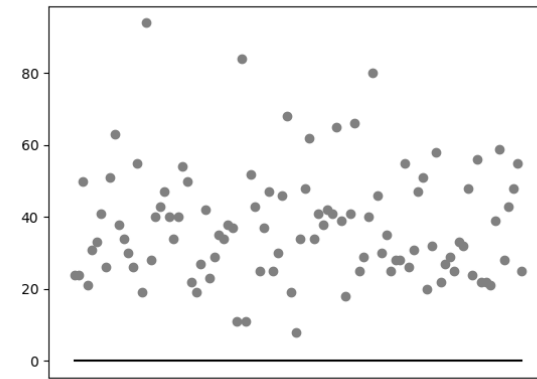
(b) Reta-0 LeNet *ReLU*.



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



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



1.4 Abordagem 4

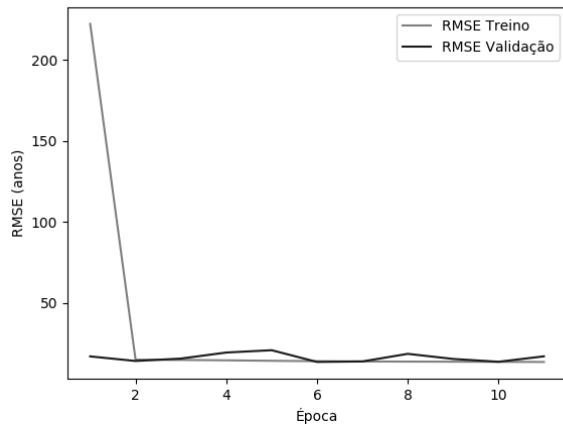
A quarta abordagem utilizou as imagens da base de dados normalizadas e com equalização de histograma de cores, além de técnicas de *data augmentation*, que inclui a probabilidade de uma rotação entre 0 e 20 graus, zoom de 0.8 a 1.2, chance de flip de 0.5, translate de 0.2. Porém, utilizou-se a métrica MAE para o cálculo da atualização dos pesos (como loss). Neste ponto, escolheu-se a dentre as treinadas nas abordagens anteriores, ou seja, a rede LeNet com função de ativação Relu.

critério

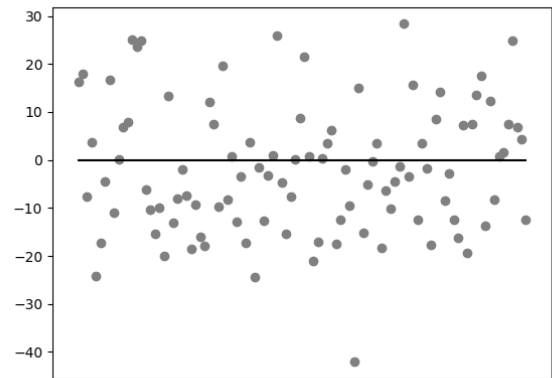
Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.5.

Figura 1.6: Resultados do treinamento e teste da CNN AlexNet.

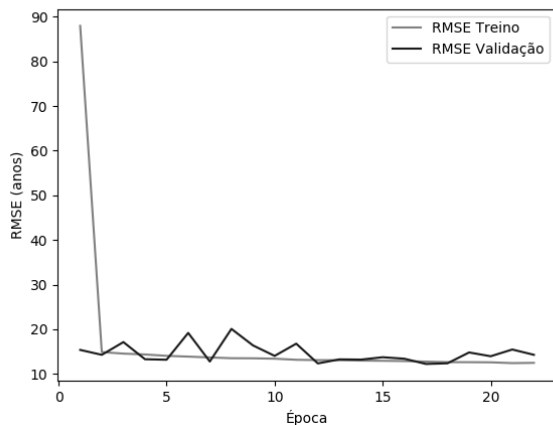
(a) Treinamento AlexNet *ReLU*.



(b) Reta-0 AlexNet *ReLU*.



(c) Treinamento AlexNet *Leaky ReLU*.



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

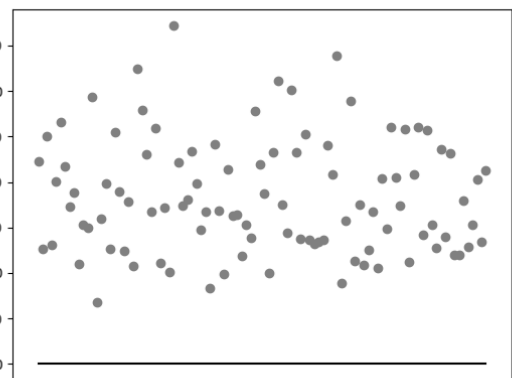


Tabela 1.4: Resultados do treino e teste dos modelos propostos na Abordagem 1.

Rede	Função de ativação	Parâmetros	Épocas	Tempo de treinamento	MAE Teste	RMSE Teste
LeNet	<i>Leaky ReLU</i>	params	15	12 h	14.44	18.18
LeNet	<i>ReLU</i>	params	43	16 h	14.09	17.93
AlexNet	<i>ReLU</i>	58.286.145	10	15 h	38.63	41.22
AlexNet	<i>Leaky ReLU</i>	params	30	40 h	15.33	18.58

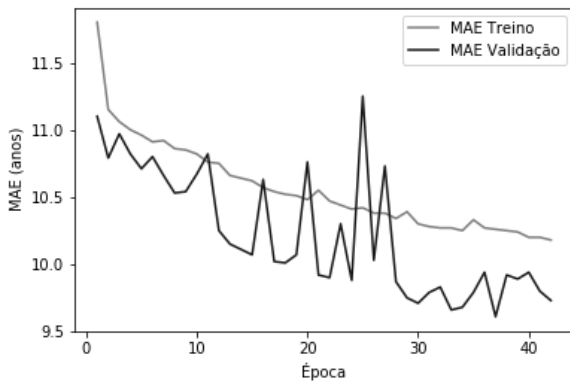
1.5 Abordagem 5

A quinta abordagem utilizou as imagens da base de dados normalizadas, mas sem equalização de histograma de cores, e técnicas de *data augmentation*. Porém, seguiu-se utilizando a métrica MAE para o cálculo da atualização dos pesos (como loss), e aumentou-se o tamanho do batch para 128, haja vista a característica instável do treinamento mostrada nas abordagens anteriores. Utilizou-se somente da arquitetura LeNet ReLU sem histograma e data augmentation por ser a que alcançou menor RMSE dentre as treinadas nas abordagens anteriores.

Obedecendo ao método de validação cruzada *holdout* previamente mencionado, os resultados desta abordagem encontram-se sintetizados na Tabela 1.5.

Figura 1.7: Resultados do treinamento e teste da CNN LeNet.

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



(b) Reta-0 LeNet *ReLU*.

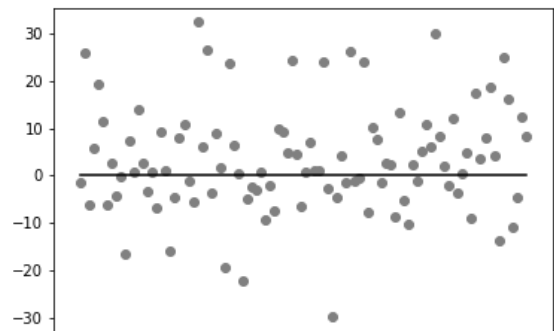


Tabela 1.5: Resultados do treino e teste dos modelos propostos na Abordagem 1.

Rede	Função de ativação	Parâmetros	Épocas	Tempo de treinamento	MAE Teste	RMSE Teste
LeNet	<i>Leaky ReLU</i>	params	15	12 h	14.44	18.18
LeNet	<i>ReLU</i>	params	43	16 h	14.09	17.93
AlexNet	<i>ReLU</i>	58.286.145	10	15 h	38.63	41.22
AlexNet	<i>Leaky ReLU</i>	params	30	40 h	15.33	18.58

1.6 Abordagem x

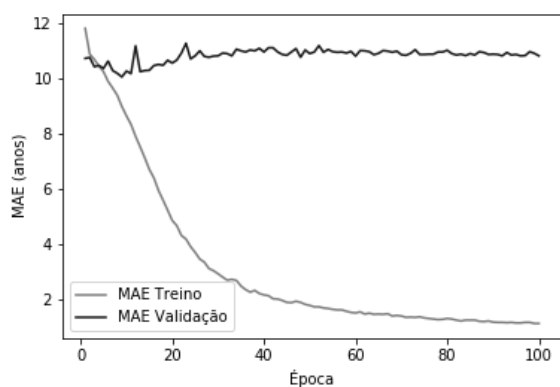
- Mesmas redes - Normalização das imagens, equalização por histograma -> o que é - data augmentation -> mais técnicas de data augmentation

1.7 Abordagem x+1

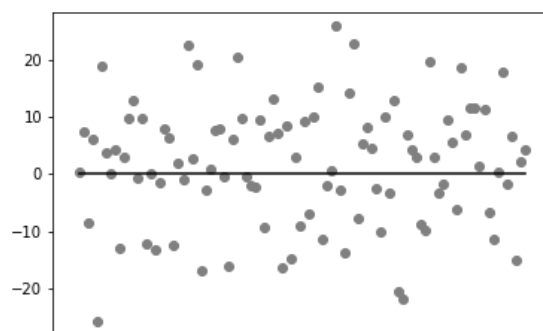
Outras arquiteturas VGG com transfer learning 1. Retirar última camada (softmax) e adicionar leaky relu 2. Retirar duas últimas camadas (dense e softmax) e adicionar leaky relu

Figura 1.8: Resultados do treinamento e teste da CNN LeNet.

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



(b) Reta-0 LeNet *ReLU*.



Capítulo 2

Considerações Finais

O objetivo deste trabalho consiste em elaborar estratégias inteligentes para estimação de idade de telespectadores de *Smart* TVs a partir de suas respectivas fotografias faciais. Para este fim, foram propostos, treinados e testados em caráter preliminar dois modelos de CNNs já bem estabelecidos na literatura, a LeNet e AlexNet, com dois perfis de hiperparâmetros cada um.

Com isto, observou-se uma melhora significativa na performance da AlexNet, enquanto o RMSE da LeNet não sofreu grandes mudanças. Quanto às saídas das redes, a LeNet exibiu valores positivos e negativos próximos de zero, e a AlexNet forneceu previsões que giravam em torno da média dos dados. Estes resultados são preliminares e certamente outros modelos e parâmetros serão investigados conforme previsto na metodologia e cronograma deste trabalho de conclusão de curso.

Nos próximos meses, os esforços estarão concentrados em pesquisar e adotar estratégias que possam minimizar os problemas identificados, como substituir as funções de ativação das camadas ocultas por outras variantes da *ReLU*, adotar métodos específicos de inicialização de pesos, normalização de *batch*, entre outros. Planeja-se também a proposição, o treinamento e teste de outras redes inspiradas em outros modelos canônicos.

O problema em questão é importante do ponto de vista prático para o desenvolvimento de diversas soluções de recomendação de conteúdo e controle parental em *Smart* TVs, auxiliando no desenvolvimento destas soluções tecnológicas. Considerando a formação de uma bacharelada em Engenharia de Computação, endereçar este problema permite a prática de diversos conceitos

vistos ao longo do curso, em especial relacionados às disciplinas de Inteligência Artificial, Redes Neurais, Processamento Digital de Imagens, *Machine Learning* e Sinais e Sistemas.

Referências Bibliográficas