



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

MATEUS FRANCO RODRIGUES

**GERAÇÃO DE DADOS SINTÉTICOS UTILIZANDO REDES NEURAIAS
ARTIFICIAIS**

RUSSAS

2021

MATEUS FRANCO RODRIGUES

GERAÇÃO DE DADOS SINTÉTICOS UTILIZANDO REDES NEURAIAS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de software.

Orientadora: Prof. Dra. Tatiane Fernandes Figueiredo

RUSSAS

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R614g Rodrigues, Mateus Franco.

Geração de dados sintéticos utilizando redes neurais artificiais / Mateus Franco Rodrigues. – 2021.
35 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas,
Curso de Engenharia de Software, Russas, 2021.

Orientação: Profa. Dra. Tatiane Fernandes Figueiredo.

1. Geração de dados. 2. Redes adversárias generativas. 3. aumento de dados. I. Título.

CDD 005.1

MATEUS FRANCO RODRIGUES

GERAÇÃO DE DADOS SINTÉTICOS UTILIZANDO REDES NEURAIAS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de software.

Aprovada em:

BANCA EXAMINADORA

Prof. Dra. Tatiane Fernandes
Figueiredo (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bonfim Amaro Júnior
Universidade Federal do Ceará(UFC)

Prof. Dr. Pablo Luiz Braga Soares
Universidade Federal do Ceará(UFC)

AGRADECIMENTOS

Aos meus pais, por todo amor que me foi dado e dedicação para que eu pudesse realizar mais esse sonho, até nas maiores dificuldades sempre fizeram de tudo para que eu conseguisse chegar até aqui. sou grato por tudo que me proporcionaram.

A minha namorada Dávila Lima, por me apoiar em todas as minhas escolhas e estar comigo em todos os momentos. Seu amor e paciência sempre fizeram a minha jornada ser mais leve. Eu a amo e agradeço de mais.

A professora Tatiane Fernandes por ter confiado e me orientado nesses últimos passos da minha pesquisa. Seus conselhos e incentivos foram essenciais.

Ao meu amigo e professor Pablo Soares por estar me ajudando desde o início da jornada, sou muito grato por todas as portas e caminhos que me ajudou a trilhar, mas sou mais grato ainda por a nossa amizade.

Aos meus amigos da Universidade Federal do Ceará (UFC) Cleiton Monteiro, Artur Sampaio, Susana Moreira, Marlo Oliveira, Herverson de Sousa e Ana Cibele, vocês sempre me inspiraram a ser uma pessoa melhor e sempre estiveram do meu lado durante toda essa jornada, agradeço a vocês também pelas aventuras e bons momentos juntos.

Aos meus amigos de Francisco Santos (FS) Agnel Junior, Paulo Victor, Wesley Henrique, Julio Cesar, Rafael Max e Daniel Sousa, vocês sempre estiveram do meu lado compartilhando bons momentos com brincadeiras e companheirismo, cada momento com vocês foi importante na minha vida, obrigado por tudo meus amigos.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

À cidade e ao campus de Russas por tão bem terem me recebido.

Para finalizar, a todos aqueles que contribuíram de forma direta e indiretamente para a minha graduação.

“Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas. Se você se conhece mas não conhece o inimigo, para cada vitória ganha sofrerá também uma derrota. Se você não conhece nem o inimigo nem a si mesmo, perderá todas as batalhas”

(Sun tzu)

RESUMO

A geração de dados sintéticos tem se intensificado bastante nos últimos anos, visto que muitos cientistas e pesquisadores têm utilizado cada vez mais técnicas de aprendizagem de máquina, das mais diversas formas, para resolver tanto problemas da comunidade científica quanto problemas empresariais. Desse modo, o objetivo deste trabalho é a criação de um modelo para a geração de imagens em escala de cinza, visando obter a melhor performance com base nos resultados da técnica de pontos de percepção e utilizando o menor número possível de recursos. O modelo criado pode ser utilizado como gerador de recursos para entradas em outros trabalhos que envolvam aprendizado de máquina e que estejam em um estado de carência de dados para o treinamento do modelo. Além disso, este trabalho também é composto por uma série de passos que facilitaram a replicação do modelo para outros casos com performance semelhante. Como resultado, as imagens geradas pelo modelo foram analisadas sob dois tipos de métricas: uma mais objetiva que resulta em valores concretos, que foram comparados com outros métodos para geração de dados; e outro método na qual foram feitas análises visuais das imagens geradas pelo modelo em comparação com as imagens originais já existentes no banco de dados em estudo. Nesse contexto, foram vistos resultados eminentes tanto se tratando da quantidade de recursos que foram utilizados para a geração de imagens, quanto a qualidade das imagens geradas em ambas as formas de avaliação.

Palavras-chave: Geração de dados. Geração de Imagens. Aumento de dados. Rede adversaria generativa Condicional.

ABSTRACT

The generation of synthetic data has intensified a lot in recent years, as many scientists and researchers have increasingly used machine learning techniques in the most diverse ways to solve both scientific communities and business problems. This work aims to create a model for the generation of grayscale images, aiming to obtain the best performance with perceptual using the least possible number of resources. The model created can be used as a resource generator for entries in other jobs involving machine learning and in a state of lack of data for model training. In addition, this work is also composed of a series of steps that facilitate the replication of the model to other cases with similar performance. As a result, the images generated by the model were analyzed under two types of metrics: a more objective one that results in concrete values, which were compared with other methods for data generation; and another method in which visual analyzes were made of the images generated by the model in comparison with the original images already existing in the database under study. In this context, outstanding results were seen regarding both the number of resources used for the generation of images, and the quality of the images generated in both forms of evaluation.

Keywords: Data generation. Image generation. Data augmentation. Conditional adverse-generative network.

LISTA DE FIGURAS

Figura 1 – Neurônio de uma Rede Neural Artificial (RNA)	16
Figura 2 – Arquitetura de uma Rede Neural Convolutacional (RNC)	19
Figura 3 – Rede Adversária Generativa (RAG)	20
Figura 4 – Rede Adversária Generativa Condicional (RAGC)	21
Figura 5 – Metodologia	24
Figura 6 – Rede Adversária Discriminadora	27
Figura 7 – Rede Adversária Geradora	28
Figura 8 – Amostra do Banco Mnist	29
Figura 9 – Comparação das médias do pontuação de percepção	31
Figura 10 – Comparação dos desvios padrões do pontuação de percepção	31
Figura 11 – Imagens do banco de dados	31
Figura 12 – Imagens geradas pela RAGC	31

LISTA DE TABELAS

Tabela 1 – Combinação de hiperparâmetros	25
--	----

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizagem de Máquina
AP	Aprendizagem Profunda
IA	Inteligência Artificial
PMC	Perceptron Multi-Camada
RAG	Rede Adversária Generativa
RAGC	Rede Adversária Generativa Condicional
RNA	Rede Neural Artificial
RNC	Rede Neural Convolutacional

SUMÁRIO

1	INTRODUÇÃO	12
2	OBJETIVOS	14
2.1	Objetivos gerais	14
2.2	Objetivos específicos	14
3	FUNDAMENTAÇÃO TEÓRICA	15
3.1	Geração de dados sintéticos	15
3.2	O aprendizado de uma rede neural artificial	15
3.3	Hiperparâmetros	17
3.4	Arquiteturas de RNA	18
3.4.1	<i>Redes neurais convolucionais (RNC)</i>	18
3.4.2	<i>Rede adversária generativa (RAG)</i>	19
3.4.3	<i>Rede adversária generativa condicional (RAGC)</i>	20
3.4.4	<i>Pontuação de percepção</i>	21
4	TRABALHOS RELACIONADOS	22
4.1	Modelo para geração de imagens gerais com Rede Adversária Generativa Condicional (RAGC)	22
4.2	Modelo para geração de imagens de cômodos residenciais com Rede Adversária Generativa (RAG)	22
4.3	Criação de Modelo para geração de imagens de tomografias com RAG	22
5	METODOLOGIA	24
5.1	Criação da arquitetura da RNA	24
5.2	Configuração de hiperparâmetros	25
5.3	Estrutura da Rede neural	26
5.4	Captura de dados para execução	28
5.5	Aplicação do modelo para geração de dados	29
6	RESULTADOS	30
6.1	Análise dos testes realizados	30
7	CONCLUSÕES E TRABALHOS FUTUROS	33
7.1	Trabalhos Futuros	33
	REFERÊNCIAS	34

1 INTRODUÇÃO

Aprendizagem de Máquina (AM) e Aprendizagem Profunda (AP) são duas subáreas da Inteligência Artificial (IA) que tem tido recentemente, uma ampla eclosão na academia através de diversas pesquisas, tais como carros autônomos (AMMOUR *et al.*, 2017), detectores de objetos (LI *et al.*, 2017), entre outros. Embora, entre 1969 e 1986, período conhecido como inverno da IA, essas subáreas não obtivessem seu devido reconhecimento, pois o *perceptron*, um dos principais métodos de aprendizagem profunda da época, era capaz de solucionar somente problemas lineares. Com o surgimento de novos métodos, como o Perceptron Multi-Camada (PMC), a capacidade de solucionar problemas mais complexos (RUMELHART *et al.*, 1988) trouxe uma nova ascensão a área. Desta forma, a evolução das redes neurais passou a ser crescente, sendo atualizada com mais camadas escondidas, que continham uma ampla capacidade de aprendizagem quando treinadas corretamente (HINTON; SALAKHUTDINOV, 2006).

Após o surgimento das PMC, foram propostas outros tipos de redes neurais, tais como, as Redes Neurais Convolucionais (RNC) que tem melhor capacidade de aprendizagem quando os dados são imagens (CONCHA *et al.*, 2019) ou vídeos (FERREIRA, 2017) e a Rede Adversária Generativa (RAG), onde suas principais funções são a criação e a simulação de dados (KARACAN *et al.*, 2016; XIONG *et al.*, 2017). Estas e outras redes tem sido amplamente aplicadas para solucionar problemas de classificação, regressões e clusterização, principalmente em imagens (ZUCHINI *et al.*, 2003; BRAGA; COELHO, 2018). Com a resolução de alguns problemas considerados desafiadores no passado, como por exemplo, a necessidade de pré-treinamentos, já resolvidos devido a possibilidade de passagem de conhecimento entre redes neurais (CHEN *et al.*, 2016), surgiram outras lacunas antes não exploradas. Uma dessas lacunas é a dificuldade, dependendo do problema, de se encontrar dados base para que as redes possam de fato adquirir um conhecimento considerado razoável (WANG; PEREZ, 2017; CUI *et al.*, 2015).

Segundo Wong *et al.* (2016), a falta de dados se tratando de imagens, é um problema relativamente atual, e tem prevalecido sobre outras questões, tais como, ajuste de parâmetros e otimização da Rede Neural Artificial (RNA) (ZANCHETTIN, 2008). De acordo com Horta *et al.* (2011), a literatura tem utilizado métodos inteligentes para aumento e balanceamento de dados como uma tentativa de contornar este problema, no entanto nenhum demonstrou ser 100% eficaz. Motivados pelas questões apresentadas, este trabalho busca por novos métodos que possam solucionar o problema de ausência de muitas imagens em uma base de dados de forma razoável. Para tal, uma possível solução foi construída a partir de um estudo de geração de imagens,

através de uma RAG, tendo sido utilizada uma base de dados sintética como semente da geração. Desta forma, este trabalho tem como principal função a criação de um modelo para gerar dados sintéticos no formato de imagens, buscando ampliar as possibilidades de escolha na geração de dados. Os resultados poderão ser utilizados para melhorar a performance dos algoritmos de aprendizagem de máquina fazendo com que os mesmos tenham uma melhor chance de acerto.

A organização desse trabalho é mostrada da seguinte forma. No Capítulo 2 estão os objetivos gerais e específicos do trabalho. O Capítulo 3 encontra-se a fundamentação teórica que tem o objetivo de ajudar o leitor a entender todos os termos mostrados durante o trabalho. Já o Capítulo 4 será dedicado aos trabalhos relacionados. Além disso, as metodologias e os processos que serão utilizados para se alcançar os objetivos desse trabalho se encontram no Capítulo 5. No capítulo 6 é apresentado os resultados e por fim no capítulo 7 se encontram conclusões sobre o trabalho e trabalhos futuro.

2 OBJETIVOS

2.1 Objetivos gerais

Criação de um modelo com base na arquitetura de Rede Adversária Generativa Condicional (RAGC) para a geração de imagens em escala de cinza visando auxiliar modelos de classificação de AM.

2.2 Objetivos específicos

Visando a conclusão do objetivo principal deste trabalho foi feito um pequeno conjunto de objetivos específicos:

- Definir os dados que vão ser utilizados;
- Definir o processo necessários como treinamento e configuração de hiperparâmetros até a geração dos dados;
- Seleção das redes neurais para geração dos modelos utilizados para criação das novas imagens;
- Comparar e avaliar as imagens geradas com a pontuação de percepção.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo vamos expor os princípios gerais desse trabalho e explicá-los. Ademais também serão explicados noções básicas sobre redes neurais e suas arquiteturas.

3.1 Geração de dados sintéticos

De acordo com Clarke (1976) a geração de dados sintéticos é definida como um problema que pode ser dividido em dois subproblemas distintos. O primeiro subproblema consiste na necessidade de gerar dados com característica condizentes com a realidade, conseqüentemente esses influenciarão diretamente nas próprias características do objeto representado. Outro detalhe importante é que os dados não devem ser iguais aos dados passados como entrada. O segundo subproblema consiste nas validações de características lógicas, pois algumas características podem influenciar diretamente na existência do objeto, exemplo: imagine que é necessário representar o dia a dia de uma pessoa, então dados como: quantas horas a pessoa descansa; trabalha; e se diverte são gerados. Caso a soma desses dados ultrapassem 24 horas, esse objeto não existiria, pois isso implica dizer que o dia dessa pessoa tem mais de 24 horas o que logicamente é uma afirmação falsa.

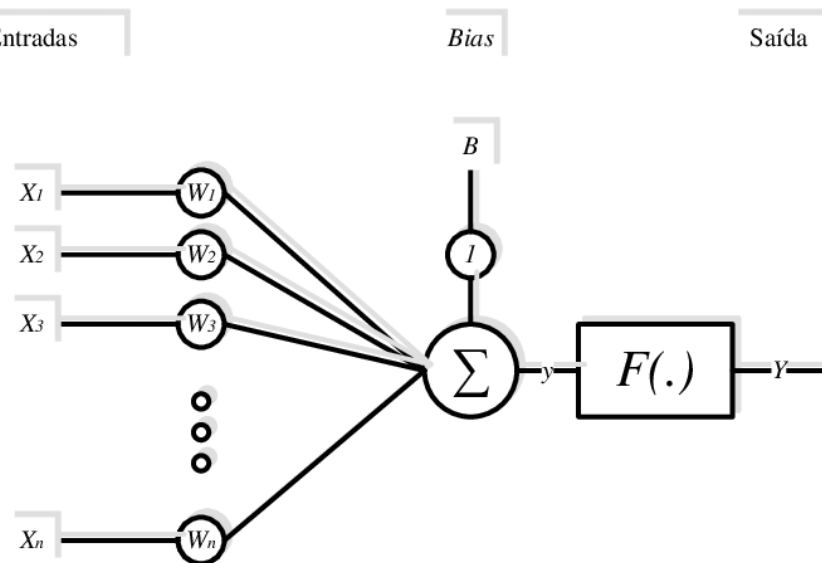
3.2 O aprendizado de uma rede neural artificial

O aprendizado de uma RNA é feito de acordo com o problema que se quer resolver. Existem redes neurais com três tipos de aprendizado, sendo eles o supervisionado, não supervisionado e por reforço. Neste trabalho, iremos explorar o aprendizado supervisionado, pois todas as redes que serão mostradas neste trabalho utilizam esse método. O aprendizado supervisionado é um processo para treinar alguns tipos de RNA. Ele é relativamente simples, partindo do ponto que se tenha um conjunto de dados que são as entradas da rede neural e as respectivas saídas correspondentes a cada uma das entradas. No início, os dados são enviados para a entrada da RNA e nesse momento é executado o produto escalar na camada de entrada. Depois, os dados são enviados para as camadas intermediárias, comumente chamadas de camadas escondidas. Então, o mesmo processo ocorre como na camada de entrada. Adicionalmente, também é executada uma função de ativação para cada neurônio e novamente os dados são passados para a camada posterior até que os dados percorram todas as camadas e cheguem ao fim da RNA.

A primeira parte do aprendizado é um processo linear, em que as entradas são dadas

e multiplicadas por seus respectivos pesos e passadas aos neurônios, estes têm a função de somar quaisquer entradas e então enviar a saída para a próxima camada. A representação de cada neurônio na Figura 1 apresenta esta parte do processo. Outro ponto bastante importante é o *bias*, este é um parâmetro adicionado a RNA usado para ajustar a saída junto da multiplicação de pesos, ele faz com que a RNA se adapte melhor a problemas do mundo real fazendo com que o processo de treinamento consiga sair do ponto da origem.

Figura 1 – Neurônio de uma RNA



Fonte: www.researchgate.net (2010)

Antes de repassar o valor resposta de um neurônio para o subseqüente é necessário utilizar a função de ativação, que é escolhida de acordo com as características do problema, um exemplo de função de ativação é a função relu dada por:

$$\text{relu}(x) = \text{maiorEntre}(0, x).$$

Onde a função *maiorEntre* retorna o maior valor entre a entrada x e o valor 0, sendo $x \in \mathbf{R}$. Depois que os resultados são repassados de neurônio a neurônio até chegar a saída da RNA, o aprendizado avança para um novo procedimento: a checagem do erro. Este valor é o resultado entregue pela RNA em comparação com o resultado válido. A partir desta checagem, são efetuados cálculos para identificar a quantidade de alterações necessárias que devem ser feitas no treinamento da rede, até que se adquira uma determinada aproximação desejada. Existem várias funções para o processo de ajuste de acordo com o erro obtido, a função categórica cruzada mostrada logo abaixo, ela é uma das mais utilizadas na literatura e por este motivo também será

utilizada nesse trabalho. Como cada uma das entradas gerará um resultado, podemos fazer os cálculos dos respectivos erros, para posteriormente melhorar a acurácia da RNA na próxima iteração.

$$E = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}).$$

Onde E é o erro em relação aos valores esperados e obtidos, M é o número de classes, y é um indicador binário que corresponde a classificação correta da classe c em relação a entrada o . Desta forma, p representa a predição em probabilidade da entrada o ser da classe c .

A partir desse ponto, a RNA tem o valor necessário para modificar os pesos de modo que possa diminuir o erro na saída resultante da RNA. Com o erro total o próximo passo é fazer o processo inverso ao inicial, ou seja, retroceder em cada camada e atualizar os pesos. O ajuste de pesos é feito através da derivada parcial da função de ativação. Desta forma, temos:

$$W^{t_1} = W^{t_0} - \alpha \frac{\partial E}{\partial W}.$$

Onde W^{t_0} é o valor antigo do peso, α é o valor da taxa de aprendizagem e W^{t_1} é o novo valor atualizado do peso.

Depois de todo esse processo, uma iteração, também chamada de época, é finalizada. Em cada final de época são selecionados novas entradas que são novamente enviadas para a entrada da RNA. Esse ciclo é re-executado n vezes até que a RNA tenha uma acurácia desejável para o problema.

3.3 Hiperparâmetros

Os hiperparâmetros de uma rede neural são constantes que indicam parte da arquitetura e como vai ser coordenado o processo de estruturação e de treinamento da RNA como um todo. A importância na configuração desse tipo de constante pode ser notada na velocidade do treinamento da rede neural e no próprio desempenho da rede posteriormente. Alguns dos hiperparâmetros que devem ser ajustados são: a quantidade de épocas, o número de neurônios nas camadas escondidas, o tamanho do lote, a taxa de aprendizagem, o modo de inicialização dos pesos, dentre outros. Como objetivo maioria dos ajustes feitos nos hiperparâmetros buscam uma forma de otimizar a RNA fazendo com que os resultados da função de erro sejam os melhores possíveis.

3.4 Arquiteturas de RNA

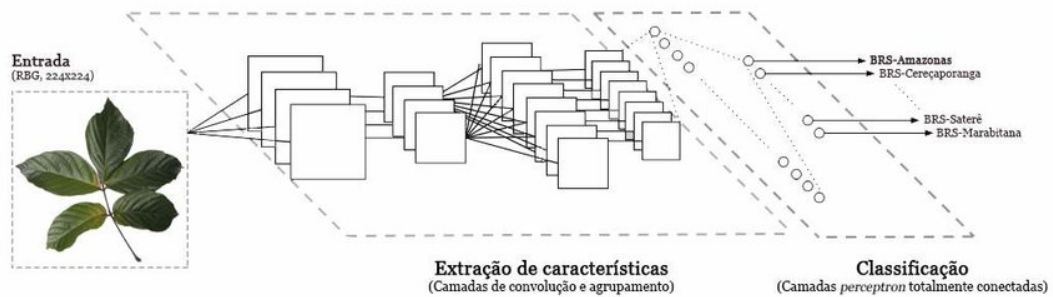
Uma RNA pode ser construída de diferentes formas, muitas escolhas são importantes na sua topologia como o tipo de neurônio usado, o comportamento dos neurônios, a comunicação entre eles e a organização das camadas da rede. Tudo isso faz parte e influencia no que é chamado hoje de arquitetura da RNA (VASUDEVAN *et al.*, 2019). Atualmente, existem mais de 20 tipos de arquiteturas de redes neurais, dentre os quais encontra-se os tipos primitivos, tais como *perceptron*, *perceptron* multi camadas e redes mais elaboradas tais como as RNC e RAG. Essas duas últimas arquiteturas de RNA são extensões de redes primitivas e serão necessárias para o entendimento e realização deste trabalho.

3.4.1 Redes neurais convolucionais (RNC)

Para LeCun *et al.* (1998) as RNC são um tipo de arquitetura de rede composto por duas estruturas: uma rede neural comum do tipo multicamada adicionada de um tipo de camada formada por filtros que são utilizadas para tratamento das entradas antes de serem passadas para a rede neural em si. Nesse tipo de rede, os filtros podem ser utilizadas para receber e tratar vários tipos de entrada como imagens, números, e até vídeos. Dependendo do problema abordado os principais tipos de filtros aplicados são de convolução, polimento máximo e achatamento.

Para Silva *et al.* (2017) cada camada dentro de uma RNC tem uma função importante. As primeiras camadas são as que mais usam processamento dentro do modelo RNC. Denominadas camadas convolucionais, nessas camadas o dado passado é tratado como uma matriz. No caso de uma imagem, cada um dos seus pixels equivalem a uma posição na matriz de entrada, a função dessas camadas é somente processar uma serie de mapas de filtros entre cada conjunto de valores da matriz. Esse processo resulta em uma matriz que corresponde a matriz anterior com tamanho reduzido, sendo essa matriz reduzida a entrada para as próximas camadas. A Figura 2 mostra a arquitetura e o posicionamento das camadas de uma RNC.

Figura 2 – Arquitetura de uma RNC



Fonte: www.researchgate.net (2017)

Diferente de uma PMC, as camadas adicionais em uma RNC a tornam muito poderosa, principalmente na etapa de extração de características, fazendo com que seja a mais utilizada atualmente para tratamento de imagens e até objetos 3D.

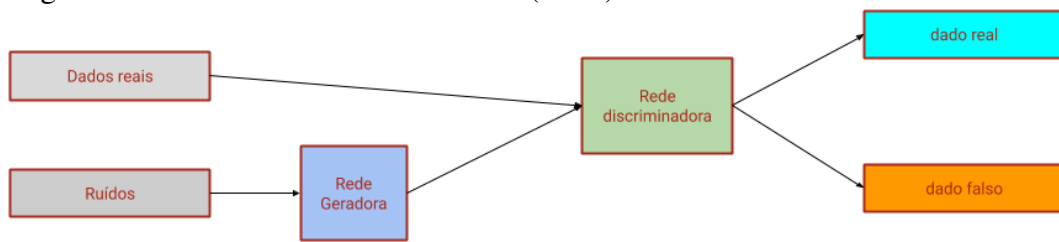
3.4.2 Rede adversária generativa (RAG)

Para Goodfellow *et al.* (2014), uma RAG é um *framework* para se treinar duas RNA simultâneas. Para tal, o treinamento é realizado através de um processo de aprendizagem adversária, de forma que as duas RNA aprendam ao mesmo tempo. O conceito deste tipo de rede, pode ser entendido como uso de duas redes neurais com diferentes funções para resolver um problema. Assim, as duas redes trabalham como adversárias para que ambas consigam aprender. Essa arquitetura foi pensada de forma que uma rede tivesse a função de gerar dados e outra de discriminar dados.

A rede geradora tem como entrada um conjunto de dados que podem ser ruídos ou dados reais, esses dados são lançados para a rede neural, que faz o processo de propagação até que se tenha um resultado ao final. Por fim, o objetivo dessa rede é gerar dados que sejam semelhantes aos dados reais esperados. Já a rede discriminadora recebe dois tipos de dados como entrada, o dado da rede geradora e os dados reais que se esperam de resposta pela rede. O seu objetivo é distinguir quais dados são da rede geradora e quais dados são reais.

A estrutura da RAG e como suas duas redes são organizadas pode ser vista Figura 3.

Figura 3 – Rede Adversária Generativa (RAG)



Fonte: Autoria Própria

O treinamento de uma RAG ocorre da seguinte forma. Primeiro, é realizada a passagem de dados, sejam eles aleatórios ou não, para a rede geradora. Nesse momento, a rede geradora faz os cálculos necessários como em uma PMC e a saída é guardada para ser repassada para o discriminador. O discriminador recebe como entrada os dados reais e os dados criados pela rede geradora, e exatamente como uma rede RNC, a rede discriminadora aprende a classificar os tipos de dados. Os dados que são reais, esperados pela rede em si e os dados que foram criados a partir da rede geradora. Assim, a rede geradora também aprende e tenta criar novos dados que consigam passar pela distinção da rede discriminadora.

3.4.3 Rede adversária generativa condicional (RAGC)

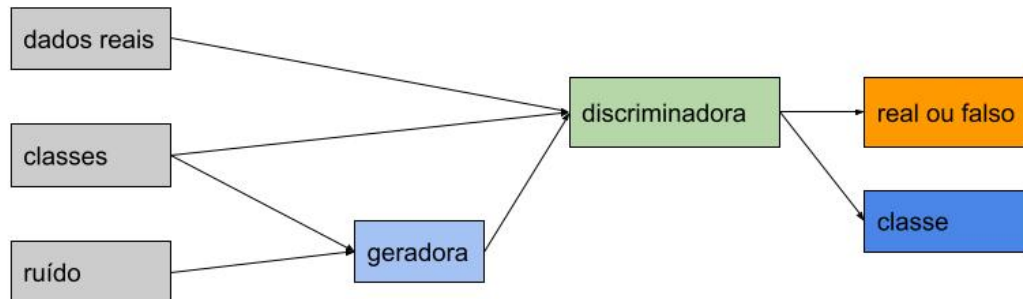
No trabalho de Odena *et al.* (2017), que foi descrito como uma melhoria para o treinamento das redes RAG, surgiu uma nova arquitetura de rede neural, a RAGC foi descrita como uma evolução da RAG comum considerando o quesito de geração de dados. Diferentemente da sua antecessora, foi adicionado a sua entrada apenas o conjunto de rótulos correspondente as classes dos dados passados para ambas as redes geradora e discriminadora, com isso a RAGC é capaz de definir que a sua RNA geradora gere dados correspondentes as classes passadas como entrada, possibilitando ao utilizador a escolha de qual tipo de classe deve ser gerado, além disso RAG comuns sofrem com um problema de convergência para um único tipo de dado caso sejam passadas diferentes classes na sua entrada, o que não ocorre na utilização de redes do tipo RAGC.

A arquitetura da RAGC é bem semelhante a uma RAG comum, utilizando as redes geradora e discriminadora para fazer a geração de dados, existindo apenas algumas diferenças. A primeira, é que para essa arquitetura não é passado somente um ruído para que a rede geradora faça a geração de dados, também é necessário o rótulo da classe, dizendo a RNA qual tipo de classe ela deve gerar. Enquanto na rede discriminadora também ocorre mudanças nos dados discriminados. Agora, além de responder se o dado é real ou falso, a rede discriminadora também

deve rotular qual tipo de classe aquele dado gerado pertence.

A Figura 4 apresenta a arquitetura para este tipo de RNA

Figura 4 – Rede Adversária Generativa Condicional (RAGC)



Fonte: Autoria Própria

3.4.4 Pontuação de percepção

Para Brownlee (2019), a pontuação de percepção nada mais é do que uma forma de fazer uma avaliação de uma rede neural do tipo RAG ou qualquer outra RNA que tenha como objetivo principal a geração de dados de forma automática a partir de ruídos passados para a rede. Esse método consiste em uma série de cálculos de probabilidade, utilizando uma RNA pré treinada juntamente com os dados gerados. Ao final, a pontuação de percepção retorna dois valores. Sendo eles a média e o desvio padrão. A média representa a qualidade das imagens e a quantidade de classes que a RNA é capaz de gerar. O desvio padrão mostra o quanto que as imagens são diferentes umas das outras em relação a sua própria classe.

É importante mencionar que apesar da pontuação de percepção ser um bom método para avaliar os dados gerados, ela é influenciada por diversos fatores, principalmente relacionados a RNA utilizada para realizar cálculos da média e do desvio padrão. Os valores obtidos a partir do resultado da pontuação de percepção podem sofrer alterações positivas ou negativas caso a RNA não tenha sido treinada suficientemente. Além disso, caso a RNA utilizada na pontuação de percepção não tenha sido treinada com os dados que a RAG tem o objetivo de gerar, existe uma grande probabilidade que o resultado retornado, ambos os valores da média e desvio padrão sejam números aleatórios e não representem fielmente o real aprendido da RAG.

4 TRABALHOS RELACIONADOS

Neste capítulo serão mostrados trabalhos, que de alguma forma tem um objetivo semelhante ao trabalho aqui proposto, os trabalhos englobam desde o objetivo mais específico que é a geração de dados até objetivos mais gerais como a utilização de IA para esse fim.

4.1 Modelo para geração de imagens gerais com RAGC

Realizado por Odena *et al.* (2017), neste trabalho os autores criaram dados sintéticos para realização da etapa de teste, a partir de um conjunto de dados menores como entrada. Este trabalho tem uma abordagem semelhante ao trabalho que esta sendo proposto, os autores criaram um modelo utilizando RAGC que consegue aprender e fazer a geração de várias imagens passadas, o banco utilizado foi o *imagenet*, que contém mil classes e mais de duas milhões de imagens. Como resultado, os autores informaram que várias classes de imagens foram geradas da forma correta, podendo até causar um condição de confusão, caso fossem misturadas às imagens geradas com as imagens do banco. Para a validação do trabalho foi utilizado a acurácia relativa da rede geradora e da rede discriminadora e uma avaliação mais subjetiva comparando as imagens geradas com as imagens do banco.

4.2 Modelo para geração de imagens de cômodos residenciais com RAG

Em Radford *et al.* (2015) foi proposto uma RNA do tipo RAG para geração de imagens, composta de várias camadas convolucionais. O resultado desse trabalho foi um modelo feito com a arquitetura RAG capaz de gerar imagens em forma de matriz. Os autores fizeram testes com alterações em toda a estrutura da rede neural, como a alteração de camadas e números de neurônios em cada camada, além de alterações na forma das camadas e algoritmos para aprendizagem, tentando maximizar o máximo possível os resultados obtidos. É importante mencionar que neste trabalho foram abordados imagens de vários tipos, os mais utilizados foram cômodos de uma residência e rostos de pessoas.

4.3 Criação de Modelo para geração de imagens de tomografias com RAG

Frid-Adar *et al.* (2018) apresentou uma rede RAG com o propósito de aumentar bases de dados de imagens, que foram utilizadas posteriormente em classificadores. A base

de dados de imagens em estudo foram de tomografias. Neste trabalho foi utilizado uma rede comum, sem camadas de convolução, para treinar a RNA e fazer a geração das imagens. Como principais resultados obtidos, os autores afirmam que a base de dados aumentada servirá como entrada de algoritmos classificadores em trabalhos futuros, além de ajudar na área de estudo sobre tumores com mais instâncias para análise.

5 METODOLOGIA

Neste capítulo será apresentado a metodologia utilizada para geração dos dados sintéticos. A metodologia está dividida em 5 etapas como mostra a Figura 5.

Figura 5 – Metodologia



Fonte: Autoria Própria

A Etapa 1 consiste na escolha da arquitetura e criação da RNA. Na Etapa 2 ocorre a configuração dos hiperparâmetros da RNA objetivando o melhor desempenho possível. Na Etapa 3 é feita a captura de dados que serão usados para o treinamento da RNA. Na Etapa 4, penúltima etapa, é aplicado o modelo, ou seja, é nesta etapa que ocorre o processo para geração de dados e por fim foi feita uma avaliação das imagens utilizando métodos objetivos e subjetivos.

5.1 Criação da arquitetura da RNA

Neste trabalho, utilizou-se a arquitetura de redes neurais do tipo RAGC. Os principais pontos considerados pelo autor para sua escolha foram:

- **Flexibilidade:** uma RAGC é uma combinação de 2 ou mais redes neurais distintas e por isso conseguem solucionar problemas mais complexos;
- **Fácil configuração:** ao contrário de outras redes, como as recorrentes geradoras que contém várias camadas a mais que uma RNA, a RAGC necessita de menos memória e menos dados para ser treinada;
- **Desempenho:** apesar de uma RAGC ser uma mistura de 2 ou mais redes neurais, essas redes neurais normalmente possuem poucas camadas, fazendo com que o processo de treinamento não seja tão custoso;
- **Não convergência:** diferentemente de uma RAG, uma RAGC não tem problema de convergência de classes. Isso significa que ela pode gerar vários tipos de classes diferentes, mesmo depois de ser treinada por bastante tempo.

5.2 Configuração de hiperparâmetros

A configuração de hiperparâmetros é uma etapa importante no aprendizado de uma rede neural, tendo em vista que alguns desses parâmetros influenciam na velocidade em que a rede irá aprender. Para geração da configuração de hiperparâmetros deste trabalho, aplicou-se um *afinamento de rede*. Essa técnica consiste em testar o aprendizado da rede com a maior quantidade de combinações possíveis combinações de hiperparâmetros.

Neste trabalho foram utilizadas 729 combinações de 5 hiperparâmetros, sendo eles a quantidade de neurônios na rede, a quantidade de camadas escondidas, o otimizador a ser utilizado, o tamanho do lote utilizado e a taxa de aprendizagem. Para realização do teste, executou-se o processo de treinamento, testando cada uma das combinações por um período de 2000, 4000 e 6000 épocas. Desta forma, os valores resultante foram comparados, sendo definido qual conjunto de hiperparâmetros seria a melhor opção para este trabalho. A Tabela 1 apresentam as 5 combinações que obtiveram os melhores resultados.

Tabela 1 – Combinação de hiperparâmetros

Neurônios	camadas escondidas	lote	otimizador	taxa de aprendizagem
80	3	100	<i>RMSprop</i>	0.00002
20	3	200	<i>Adam</i>	0.00002
80	3	200	<i>SGD</i>	0.0002
80	3	200	<i>RMSprop</i>	0.00002
150	3	200	<i>RMSprop</i>	0.00002

Fonte: Autoria Própria

Depois de realizado o processo de afinamento da rede temos a melhor opção de hiperparâmetros indicados acima em negrito, é importante destacar que esse método resulta na melhor configuração possível somente para os parâmetros usados na combinação, então o mais indicado é que se tente utilizar o maior numero de combinações possíveis para que se obtenha uma estrutura com um bom desempenho.

Nesse estágio do trabalho os hiperparâmetros foram escolhidos, levando em consideração o melhor desempenho possível para o pontuação de percepção, além disso foi feita uma avaliação subjetiva das imagens onde foram considerados os hiperparâmetros: 11000 para épocas e 100 para o tamanho do lote. Além disso, o otimizador que obteve melhor desempenho foi o *RMSprop* com uma taxa de aprendizagem para ambas as redes geradora e discriminadora de 0.00002.

5.3 Estrutura da Rede neural

As duas estruturas das RNA que compõem a RAGC resultante desse trabalho foram construídas utilizando 4 camadas convolucionais 2D para ambas as redes, 4 camadas *dropout* com 25% para esquecimento para a rede discriminadora, além de 2 e 3 camadas de *batch normalization* para a rede discriminadora e geradora respectivamente que ajudam a deixar os pesos de uma rede neural em um intervalo de $[0, 1]$ e aplicando a função de ativação *relu* em todas as camadas escondidas. Nas Figuras 6 e 7 podemos ver duas imagens que representam cada uma das estruturas das RNA respectivamente, tanto a discriminadora quanto a geradora. Pode-se notar a baixa quantidade de parâmetros utilizados na composição do modelo RAGC utilizado nesse trabalho. Para a rede que geradora das imagens foi necessário somente 773.521 parâmetros, já no caso da rede discriminadora houve a necessidade de usar somente 69.840 parâmetros, um valor bastante baixo se comparado a trabalhos semelhantes na literatura, que em sua maioria utilizam RAG para tentar gerar imagens em escala de cinza. Como exemplo podemos citar (KHANDELWAL, 2019) que utilizaram um total de 2.956.577 parâmetros.

Na criação do modelo foram utilizadas varias ferramentas, *python*¹ foi a linguagem de programação escolhida por sua simplicidade e facilidade para trabalhar com AP, além disso também foi utilizado auxiliares como *numpy*² para o agrupamento e organização dos dados e o *matplotlib*³ para visualizar e mostrar os dados, além disso para a criação do modelo proposto neste trabalho foi utilizado o *framework tensorflow*⁴ que contem varias funções já prontas que auxiliaram bastante na criação do modelo.

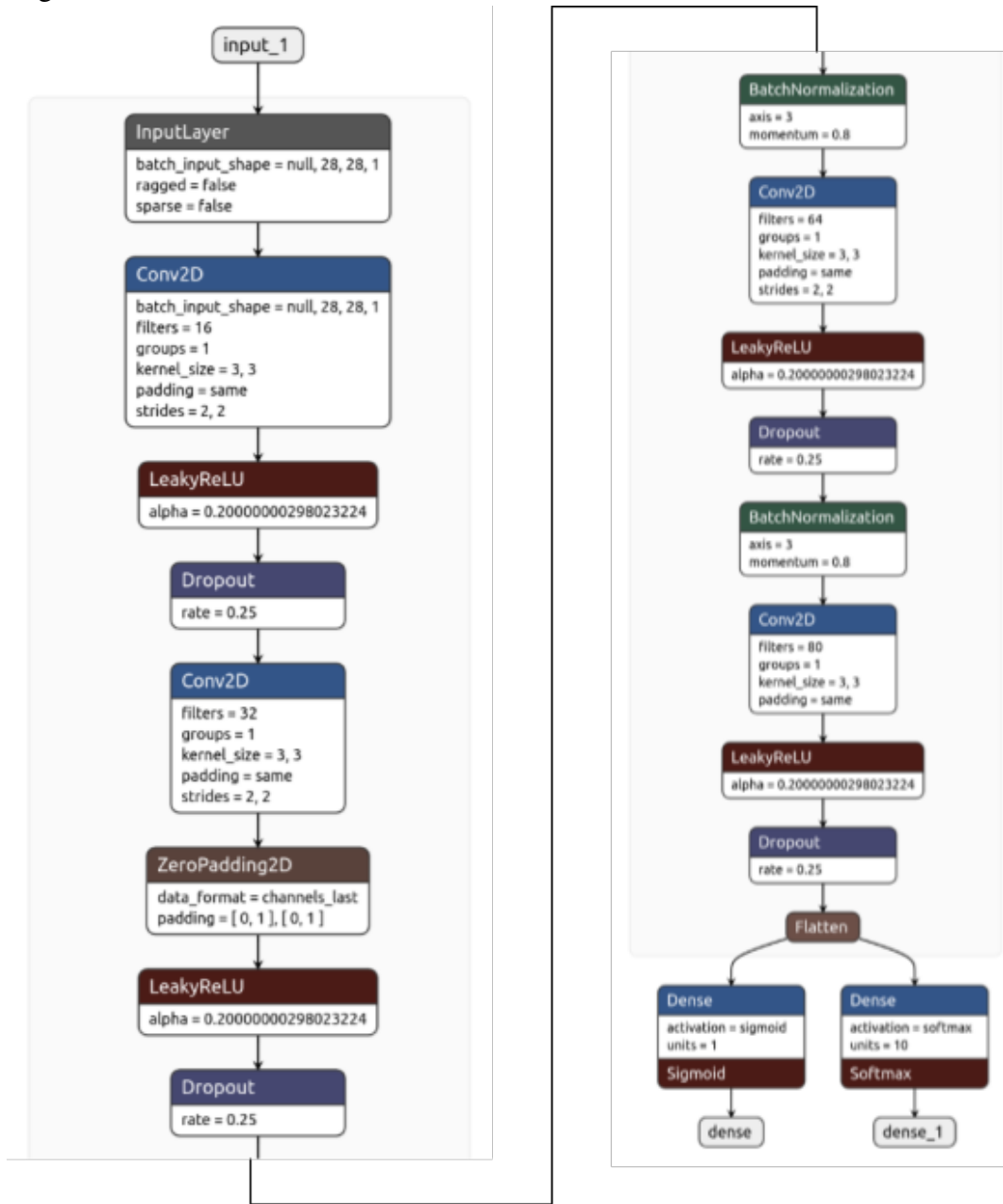
¹ <https://www.python.org/>

² <https://numpy.org/>

³ <https://matplotlib.org/>

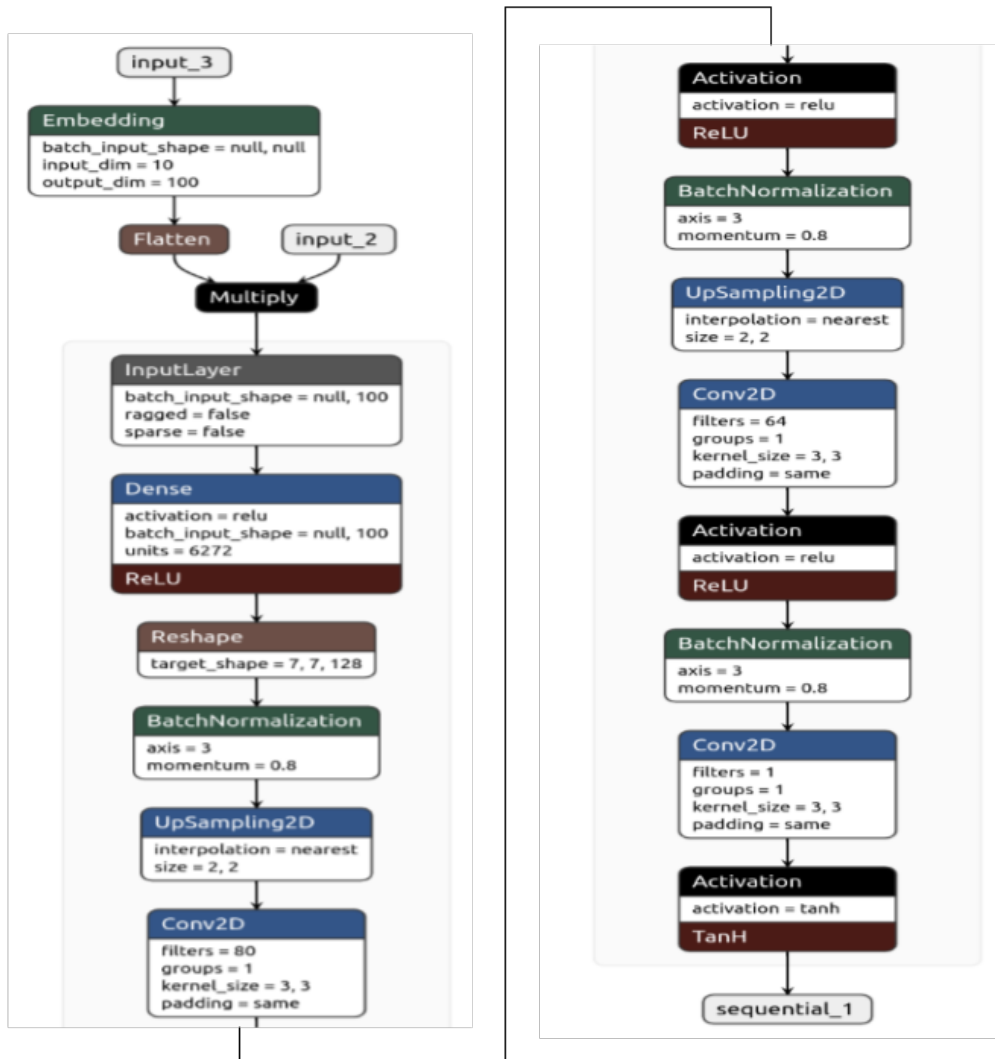
⁴ <https://www.tensorflow.org/?hl=pt-br>

Figura 6 – Rede Adversária Discriminadora



Fonte: Aatoria Própria

Figura 7 – Rede Adversária Geradora



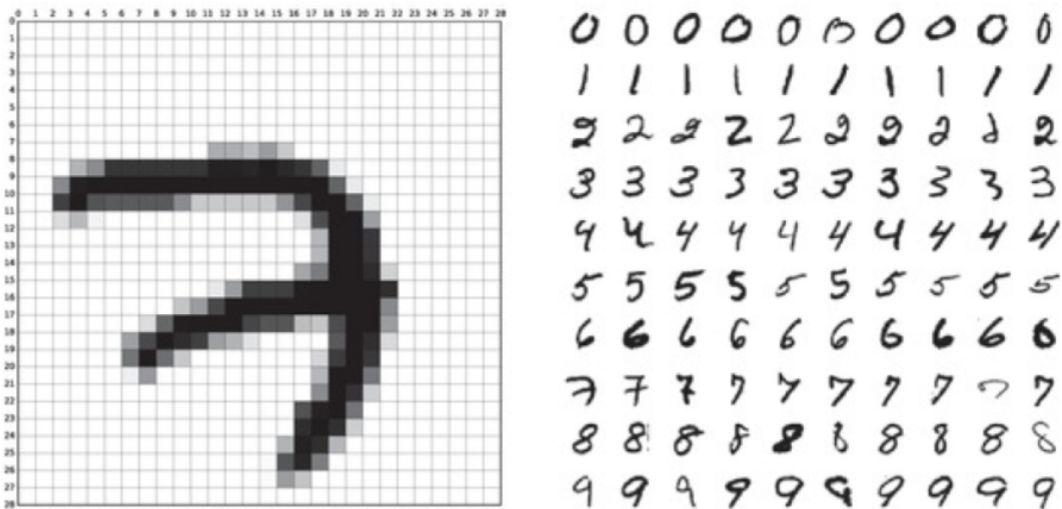
Fonte: Autoria Própria

5.4 Captura de dados para execução

O banco de dados utilizado foi o *Mnist*⁵, contendo 60000 instâncias de imagens de dígitos escritos a mão. Os dígitos estão dentro do intervalo $[0, 9]$. As imagens do *Mnist* possuem 28 pixels de largura e 28 pixels de altura, sendo imagens com apenas um canal, ou seja, contém somente uma camada de cor. A Figura 8 apresenta um exemplo de imagem do banco uma amostra do banco *Mnist*.

⁵ <http://yann.lecun.com/exdb/mnist/>

Figura 8 – Amostra do Banco Mnist



Fonte: Autoria Própria

5.5 Aplicação do modelo para geração de dados

O treinamento do modelo foi realizado utilizando os hiperparâmetros definidos conforme definido na Seção 5. Ao final do treinamento foram salvos tanto a estrutura da rede, quanto o pesos gerados, evitando assim a necessidade da realização de retreinamentos da RNA. Após a rede treinada, efetuou-se a ultima etapa de teste que consistiu na geração das imagens sintéticas e a avaliação e comparação das imagens geradas com as imagens de uma arquitetura de RAG, além disso foi feita também uma comparação com as imagens de teste do próprio banco de dados, essa avaliação se deu tanto na forma subjetiva na qual foram colocadas imagens lado a lado, e também da forma objetiva por meio da pontuação de percepção.

6 RESULTADOS

Neste capítulo é apresentado os resultados pertinentes ao trabalho e aos seus objetivos. Para validação do trabalho foi executado a metodologia proposta na Seção 5. Todo o processo foi realizado utilizando o banco *Mnist*. A avaliação da qualidade do modelo criado, utilizou-se o algoritmo de pontuação de percepção, um dos métodos mais utilizados para avaliar RAG. Para cada uma das redes neurais, uma série de hiperparâmetros foram testados, sendo escolhido a configuração de hiperparâmetros visando desempenho em tempo e qualidade do dados gerados.

6.1 Análise dos testes realizados

Para validação dos resultados obtidos, utilizou-se o método pontuação de percepção, que consiste em medir a qualidade das imagens, mensurando a diferença entre uma imagem gerada através de uma RAG e a uma RAGC.

O valor da pontuação de percepção para o modelo do tipo RAGC desenvolvido nessa pesquisa obteve uma média de 5.562 e desvio padrão de 0.432. Este valores podem ser considerados medianos se comparados aos obtidos quando aplica-se a pontuação de percepção ao próprio banco de dados, obtendo-se uma média de 9.588 e desvio padrão de 0.107. Já quando comparados com o as imagens geradas por uma RAG, pode-se afirma que o modelo proposto nesta pesquisa obtém um valor superior, uma vez que a pontuação de percepção da RAG obteve uma média 1.000 e desvio padrão de aproximadamente 0.0000001. As Figuras 9 e 10 apresentam respectivamente a comparação das médias e desvios padrões de forma gráfica.

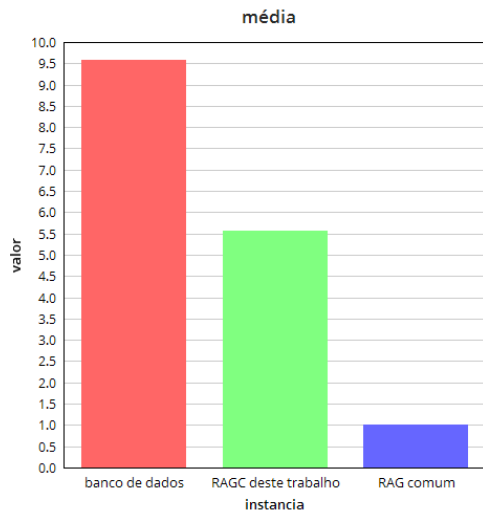


Figura 9 – Comparação das médias do pontuação de percepção

Fonte: Autoria Própria

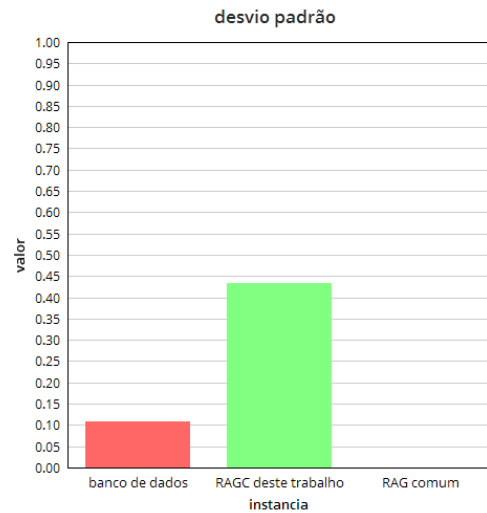


Figura 10 – Comparação dos desvios padrões do pontuação de percepção

Fonte: Autoria Própria

Buscando também apresentar uma análise visual e subjetiva das imagens geradas, as Figuras 12 e 11 apresenta respectivamente algumas imagens geradas com o modelo proposto neste trabalho, comparando-as com as imagens reais do banco de dados utilizado.

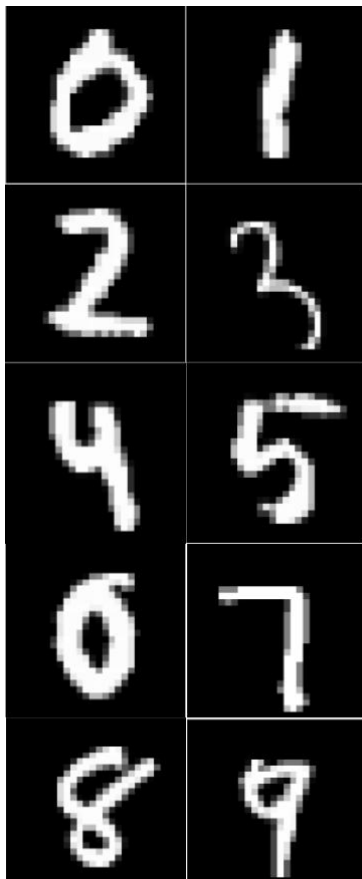


Figura 11 – Imagens do banco de dados
Fonte: Autoria Própria



Figura 12 – Imagens geradas pela RAGC
Fonte: Autoria Própria

Note que as algumas imagens geradas através do modelo desenvolvido nesse trabalho podem ser confundidas com as imagens do próprio banco de dados, por exemplo as classes 0, 1, 3, 5, 6 e 9 foram as que obtiveram o melhor resultado, já as classes 7 e 4 tiveram um desempenho intermediário enquanto isso as classes 8 e 2 tiveram os piores resultados. É importante mencionar que uma RAGC pode aprender a geração das imagens de forma não uniforme, então é esperado que algumas classes sejam geradas com melhor qualidade que outras.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propõe a recriação de uma arquitetura de rede neural artificial, gerando assim um modelo inteligente capaz de gerar imagens em escala de cinza, que poderão ser utilizadas nas mais diferentes aplicações.

Para analisar a qualidade do modelo gerado, assim como as imagens criadas, utilizou-se a técnica de pontuação de percepção. Os resultados obtidos demonstram que o modelo RAGC desenvolvido nesta pesquisa tem um desempenho razoável para geração de imagens, considerando o banco *Mnist*, obtendo uma pontuação de percepção consideravelmente superior, ou seja uma melhor performance, quando comparado a um modelo utilizando uma RAG comum. Acredita-se também que o modelo aqui apresentado possa obter resultados semelhantes quando utilizado para geração de imagens considerando outros bancos de dados que contenham divisões de classes semelhantes ao *Mnist*. Por fim, concluímos que apesar de alguns rótulos de determinadas imagens apresentarem maior nível de ruído que outras, ainda assim é possível identificar a maioria das imagens geradas pelo modelo.

7.1 Trabalhos Futuros

Como sugestão para trabalhos futuros a serem pesquisados:

- Geração de dados no formato de números para servirem de entrada para outros trabalhos de AM;
- Geração de dados no formato de objetos 3D utilizando RAGC;
- Geração de vídeos utilizando RAG.

REFERÊNCIAS

- AMMOUR, N.; ALHICHRI, H.; BAZI, Y.; BENJDIRA, B.; ALAJLAN, N.; ZUAIR, M. Deep learning approach for car detection in uav imagery. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 9, n. 4, p. 312, 2017.
- BRAGA, H. C. dos R.; COELHO, C. J. Reconhecimento de padrões em imagem empregando aprendizagem profunda. **CIPEEX**, v. 2, p. 2629–2631, 2018.
- BROWNLIE, J. **How to Implement the Inception Score (IS) for Evaluating GANs**. 2019. Disponível em: <<https://machinelearningmastery.com/how-to-implement-the-inception-score-from-scratch-for-evaluating-generated-images/>>.
- CHEN, T.; GOODFELLOW, I.; SHLENS, J. **Net2Net: Accelerating Learning via Knowledge Transfer**. 2016.
- Clarke, L. A. A system to generate test data and symbolically execute programs. **IEEE Transactions on Software Engineering**, SE-2, n. 3, p. 215–222, Sep. 1976.
- CONCHA, T. *et al.* Multi-stream convolutional neural networks for action recognition in video sequences based on spatio-temporal information: Redes neurais convolucionais de múltiplos canais para reconhecimento de ações em sequências de vídeos baseado em informações espaço-temporais. [sn], 2019.
- CUI, X.; GOEL, V.; KINGSBURY, B. Data augmentation for deep neural network acoustic modeling. **IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)**, IEEE Press, v. 23, n. 9, p. 1469–1477, 2015.
- FERREIRA, A. E. T. Estimação do ângulo de direção por vídeo para veículos autônomos utilizando redes neurais convolucionais multicanais. 2017.
- Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using gan for improved liver lesion classification. In: **2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)**. [S.l.: s.n.], 2018. p. 289–293.
- GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. **Generative Adversarial Networks**. 2014.
- HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.
- HORTA, R. A. M.; BORGES, C. C. H.; CARVALHO, F. A. A. d.; ALVES, F. J. D. S. Previsão de insolvência: uma estratégia para balanceamento da base de dados utilizando variáveis contábeis de empresas brasileiras. **Sociedade, Contabilidade e Gestão**, v. 6, n. 2, 2011.
- KARACAN, L.; AKATA, Z.; ERDEM, A.; ERDEM, E. Learning to generate images of outdoor scenes from attributes and semantic layouts. **CoRR**, abs/1612.00215, 2016. Disponível em: <<http://arxiv.org/abs/1612.00215>>.
- KHANDELWAL, R. **Generative Adversarial Network(GAN) using Keras**. 2019. Disponível em: <<https://medium.datadriveninvestor.com/generative-adversarial-network-gan-using-keras-ce1c05cfd3>>.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998.

LI, J.; LIANG, X.; WEI, Y.; XU, T.; FENG, J.; YAN, S. Perceptual generative adversarial networks for small object detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 1222–1230.

ODENA, A.; OLAH, C.; SHLENS, J. **Conditional Image Synthesis With Auxiliary Classifier GANs**. 2017.

Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. **arXiv e-prints**, p. arXiv:1511.06434, Nov 2015.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *et al.* Learning representations by back-propagating errors. **Cognitive modeling**, v. 5, n. 3, p. 1, 1988.

SILVA, C.; WELFER, D.; GIODA, F. P.; DORNELLES, C. Cattle brand recognition using convolutional neural network and support vector machines. **IEEE Latin America Transactions**, IEEE, v. 15, n. 2, p. 310–316, 2017.

VASUDEVAN, V.; ZOPH, B.; SHLENS, J.; LE, Q. V. **Neural architecture search for convolutional neural networks**. [S.l.]: Google Patents, 2019. US Patent App. 16/040,067.

WANG, J.; PEREZ, L. The effectiveness of data augmentation in image classification using deep learning. **Convolutional Neural Networks Vis. Recognit**, 2017.

WONG, S. C.; GATT, A.; STAMATESCU, V.; MCDONNELL, M. D. Understanding data augmentation for classification: when to warp? In: IEEE. **2016 international conference on digital image computing: techniques and applications (DICTA)**. [S.l.], 2016. p. 1–6.

WWW.RESEARCHGATE.NET. **Neuron in neural network**. 2010. Disponível em: <https://www.researchgate.net/figure/Figura-321-Neuronio-artificial-de-McCulloch-e-Pitts_fig1_276279246>. Acesso em: 23 set. 2019.

WWW.RESEARCHGATE.NET. **Convolutional Neural Network model**. 2017. Disponível em: <https://www.researchgate.net/figure/Figura-1-Modelo-de-Rede-Neural-Artificial-Convolutacional-LeNet-e-suas-fases-de_fig1_326631979>. Acesso em: 04 set. 2019.

XIONG, W.; LUO, W.; MA, L.; LIU, W.; LUO, J. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. **CoRR**, abs/1709.07592, 2017. Disponível em: <<http://arxiv.org/abs/1709.07592>>.

ZANCHETTIN, C. Otimização global em redes neurais artificiais. Universidade Federal de Pernambuco, 2008.

ZUCHINI, M. H. *et al.* Aplicações de mapas auto-organizáveis em mineração de dados e recuperação de informação. [sn], 2003.