Reconstrução Digital de Imagens com Redes Neurais Convolucionais

Marcelo G. de Andrade
Insper
São Paulo, Brasil
marceloga1@al.insper.edu.br

***Resumo*—Esse estudo explora implementações de diferentes redes neurais convolucionais com objetivo de realizar a reconstrução de imagens com espaços faltantes. As implementações testadas são Autoencoders Convolucionais e Deep Convolutional Generative Adversial Networks (DCGANs).**

***Index Terms*—machine learning, image inpainting, autoencoders, deep learning, convolution networks, generative adversial networks**

I. Introdução

A utilização de redes neurais convolucionais em aplicações de Machine Learning teve um grande crescimento na última década, principalmente para problemas que envolvem imagens, áudio e textos. Com finalidade didática, esse estudo explora as diferentes implementações dessas redes para resolver o problema de reconstrução de imagens.

II. O Problema

Como abordagem para o estudo de redes neurais convolucionais, foi usado o problema da reconstrução de imagens. Partindo de uma imagem com um espaço faltante, como pode-se completar essa imagem de modo que o resultado final seja o mais parecido possível com a imagem original?

Esse problema não é tão atual quanto redes neurais convolucionais, há diversas soluções procedurais utilizando características das partes completas da imagem em questão. A solução mais utilizada desse modo é a ferramenta Content-Aware do software Photoshop da Adobe.

Figura 1. Exemplo de reconstrução de imagem utilizando a ferramenta Content-Aware do Photoshop

No entanto, essa solução clássica e procedural tem dois principais pontos negativos:

1) A reconstrução nem sempre é fiel com a realidade. Usando a Figura 1 como exemplo, percebe-se que a reconstrução é muito abstrata, não fazendo sentido para olhos humanos que reconhecem imagens realistas.
2) Caso o espaço faltante contenha informações da imagem que não estão presentes em outro lugar da imagem, a reconstrução não preencherá esse espaço com essa informação. Um exemplo disso é a reconstrução de um rosto visto na Figura 2.

Figura 2. Exemplo de reconstrução falha utilizando o Content-Aware do Photoshop

O ser humano, por conhecimento prévio, sabe que no espaço faltante da Figura 2 estão os olhos da pessoa. No entanto, isso só acontece, pois já vimos imagens semelhantes e sabemos que se trata de um rosto de uma pessoa.

Pode-se agrupar os dois problemas citados acima em uma única falha em comu: o fato da solução clássica se basear apenas na própria imagem sem conhecimento prévio.

Para solucionar esse problema, criou-se um modelo de aprendizado de máquina que utiliza um dataset de imagens como treinamento, e aprende a reconstruir imagens em sua predição.

III. Trabalhos Relacionados

A utilização de Machine Learning, ou mais especificamente, redeus neurais convolucionais, para reconstrução de imagens é um tema que gerou pesquisas com ótimos resultados nos últimos cinco anos. Além das redes neurais convolucionais, em 2014 foi proposta um novo framework de redes neurais chamado de Generative Adversial Network (GAN), detalhado em [2]. As possíveis aplicações das GANs geraram um grande número de pesquisas na área de geração de imagens. A subdivisão de GANs para redes neurais convolucionais é chamada de Deep Convolutional Generative Adversial Networks

(DCGANs). Alguns exemplos de pesquisas sobre reconstrução de imagens com DCGANs são: [4], [7], [5], [6] e [1].

IV. Metodologia

Foram feitas duas implementações de redes neurais para a reconstrução das imagens. A primeira é um Autoencoder Convolucional e a segunda uma DCGAN.

Usou-se como dataset para as implementações o cifar10 [3]. Esse dataset consiste em 60.000 imagens de tamanho 32x32 pixels. Esse pequeno tamanho de imagem facilita na construção da rede e no tempo de treinamento da mesma.

Usou-se apenas as imagens do dataset, sem suas labels. Para cada imagem, foi cortado um quadrado de tamanho 16x16 no centro da mesma. A imagem com o espaço faltante se tornou a entrada do modelo, e o espaço cortado o valor desejado.

Essa operação pode ser visualizada na Figura 3.

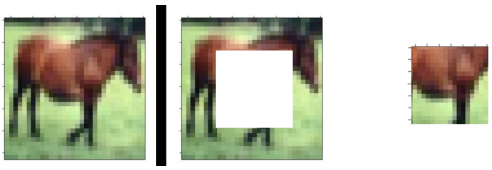


Figura 3. Visualização da operação de *crop* da imagem inicial

A. Autoencoder

A primeira implementação da rede de reconstrução de imagens foi um simples autoencoder convolucional. Esse autoencoder é constituído de camadas convolucionais com strides, diminuindo o tamanho da entrada e aumentando o número de features. No meio, uma camada chamada de *bottleneck*, com uma camada densa, e por fim, uma sequência de camadas *deconvolucionais*, tornando a imagem do tamanho de saída desejado.

A arquitetura da rede pode ser vista a seguir:

Camada	Kernel	Strides	Ativação	Output
Input	-	-	-	32x32x3
Conv	3x3	2x2	ReLU	16x16x32
Conv	3x3	2x2	ReLU	8x8x64
Conv	3x3	2x2	ReLU	4x4x128
BatchNorm	-	-	-	4x4x128
Flatten	-	-	-	2048
Dropout	-	-	-	2048
Dense	-	-	ReLU	2048
Reshape	-	-	-	4x4x128
Deconv	3x3	2x2	ReLU	8x8x64
Deconv	3x3	2x2	ReLU	16x16x3
Output	-	-	-	16x16x3

Função de Perda	Otimização	Batch Size
Mean Squared Error	Adam	64

B. Deep Convolutional Generative Adversial Network (DCGAN)

A segunda implementação envolve um conceito mais abstrato que a primeira, as GANs. Por se tratar de uma implementação de redes neurais muito recente, ainda não há consenso em como utilizá-las, apenas algumas estratégias que deram certo em estudos específicos. Usou-se como principal base de estudo o artigo [7], pois o dataset de entrada era de imagens pequenas o suficiente para ser adaptado e a rede implementada estava bem documentada.

As GANs consistem em duas redes distintas. Uma chamada de *geradora* e a outra de *discriminadora*. A *geradora* é muito semelhante a implementação Autoencoder feita na seção IV-A, responsável por prever o conteúdo do espaço faltante a partir da imagem incompleta. A rede *discriminadora* é responsável por discriminar as verdadeiras imagens faltantes das imagens geradas pela rede *geradora*.

A ideia geral das GANs é treinar essas duas redes separadamente para que elas entrem em uma disputa para diminuir o seu erro. A *geradora* tenta enganar a *discriminadora*, enquanto a *discriminadora* tenta não ser enganada.

Mais detalhes do conceito por trás das GANs pode ser visto em sua definição, no artigo [2].

Utilizou-se algumas boas práticas para arquitetura de GANs descritas nos trabalhos relacionados listados na seção III. Essas são:

- Normalizar valores de entrada entre -1 e 1 e usar ativação tanh na última camada da rede *geradora*.
- Utilizar ativação LeakyReLU seguida de BatchNormalization nas camadas convolucionais.
- Atribuir um *learning rate* 10 vezes menor para a rede *discriminadora* em relação a *geradora*.
- Treinar imagens verdadeiras e preditas pela rede *geradora* em *batchs* separados na rede *discriminadora*.

A partir dos artigos citados e das boas práticas acima, foi arquitetada a rede *geradora* e a rede *discriminadora*

Tabela I
Rede Discriminadora

Camada	Kernel	Strides	Ativação	Output
Input	-	-	-	16x16x3
Conv	3x3	2x2	LeakyReLU	16x16x32
BatchNorm	-	-	-	16x16x32
Conv	3x3	2x2	LeakyReLU	8x8x64
BatchNorm	-	-	-	8x8x64
Conv	3x3	2x2	LeakyReLU	4x4x128
BatchNorm	-	-	-	4x4x128
Conv	3x3	2x2	LeakyReLU	2x2x256
BatchNorm	-	-	-	2x2x256
Flatten	-	-	-	1024
Dropout	-	-	-	1024
Dense	-	-	ReLU	1024
Dense	-	-	Sigmoid	1

Função de Perda	Otimização	Batch Size
Binary Cross Entropy	Adam(LR = 0.0002)	32

Tabela II
REDE GERADORA

Camada	Kernel	Strides	Ativação	Output
Input	-	-	-	32x32x3
Conv	5x5	2x2	LeakyReLU	16x16x64
BatchNorm	-	-	-	16x16x64
Conv	5x5	2x2	LeakyReLU	8x8x128
BatchNorm	-	-	-	8x8x128
Conv	3x3	2x2	LeakyReLU	4x4x256
BatchNorm	-	-	-	4x4x256
Conv	2x2	2x2	LeakyReLU	2x2x512
BatchNorm	-	-	-	2x2x512
Flatten	-	-	-	2048
Dropout	-	-	-	2048
Dense	-	-	ReLU	2048
Reshape	-	-	-	2x2x512
Deconv	3x3	2x2	LeakyReLU	4x4x256
BatchNorm	-	-	-	4x4x512
Deconv	5x5	2x2	LeakyReLU	8x8x128
BatchNorm	-	-	-	8x8x128
Deconv	5x5	2x2	LeakyReLU	16x16x64
BatchNorm	-	-	-	16x16x64
Deconv	5x5	1x1	Tanh	16x16x3
Output	-	-	-	16x16x3

Função de Perda	Otimização	Batch Size
Mean Squared Error	Adam(LR = 0.002)	32

A maior dificuldade de implementação de uma GAN é seu treinamento. Como descrito em [2], deve haver um equilíbrio entre o desempenho das duas redes para que haja convergência. Caso uma das redes acabe tendo um desempenho melhor que a outra, há uma grande probabilidade do modelo não convergir.

Para cada batch de imagens, é treinada a rede *discriminadora*, e em seguida, a rede completa, sendo essa a junção da rede *geradora* e da *discriminadora*. Por se tratar da junção de duas redes distintas, um hiper parâmetro importante a ser definido é o impacto de cada uma das funções de perda na rede completa. Nesse estudo, utilizou-se os mesmo parâmetros usados em [7]: 0.999 * PG(Perda Geradora) e 0.001 * PD(Perda Discriminadora).

V. RESULTADOS

Como explicado anteriormente, foi utilizado o dataset cifar10 [3] para treinamento e teste das duas implementações feitas.

A. Autoencoder

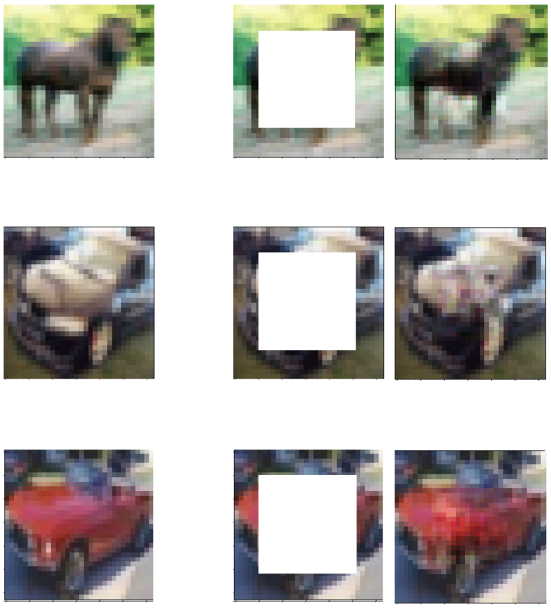
Com *batchs* de 64 imagens, a rede foi treinada por 50.000 *epochs*. O erro convergiu para 0.009. Os resultados para três imagens aleatórias do conjunto de testes foi o seguinte:

A primeira imagem é a imagem original do dataset, a segunda imagem é com o espaço faltante, e a terceira a imagem reconstruída com o espaço predito pela rede.

Os resultados já são razoáveis, claramente tendo preenchido de modo coerente às imagens anteriores. No entanto, percebe-se que o espaço preenchido é embaçado, turvo. Isso acontece, pois é apenas otimizado o erro médio dos pixels individualmente, prejudicando o contexto dos pixels conjuntos. Esse problema também foi apresentado em [6]. Uma forma de solucionar o problema da reconstrução turva é a segunda implementação, as GANs.

B. DCGAN

Por conta de uma rede muito maior e do limite de memória da GPU, foi utilizado um *batch* de 32 imagens para treinar a GAN. A rede foi treinada por 80.000 *epochs*, momento em que o erro médio da rede *geradora* convergiu para 0.008. Os resultados das três mesmas imagens do conjuntos de testes foi o seguinte:



Embora o erro médio final ser similar ao do Autoencoder, percebe-se que a utilização de uma rede *discriminadora* resolveu o problema da geração de imagens turvas. Os resultados finais da GAN, além de coerentes em termos de cor e posição, também consideraram o contexto dos pixels, tornando o resultado final mais realista que o Autoencoder.

VI. Conclusão

A partir das implementações feitas de reconstrução de imagens usando diferentes tipos de redes neurais convolucionais, conclui-se que a utilização dessas estruturas são válidas e promissoras para a solução desse problema. A construção dessas redes também tem um alto valor didático, apresentando diversos temas atuais do *Deep Learning* e incentivando a busca por soluções alternativas, como as GANs.

Além disso, as implementações mostram que a utilização de DCGANs na área de reconstrução de imagens é promissora, conseguindo solucionar problemas de falta de contexto dos pixels conjuntos, tendo um resultado extremamente realista, reafirmando as conclusões apresentadas em [4], [7], [5], [6] e [1].

A primeira melhoria futura para o estudo seria a adaptação das redes para um dataset maior. No entanto, com datasets mais complexos, a dificuldade de treinamento das GANs cresce, como explicado em [6], o ponto de equilíbrio entre as redes se torna mais complicado de ser encontrado.

Por fim, uma segunda melhoria seria a aplicação de GANs em diferentes problemas que envolvem geração de imagens realistas, o que ratificaria a importância e eficácia das GANs em problemas atuais.

Referências

[1] Kevin J. Shih Ting-Chun Wang Andrew Tao-Bryan Catanzaro Guilin Liu, Fitsum A. Reda. Image inpainting for irregular holes using partial convolutions. 2018.

[2] Mehdi Mirza Bing Xu David Warde-Farley-Sherjil Ozair Aaron Courville Yoshua Bengio Ian J. Goodfellow, Jean Pouget-Abadie. Generative adversarial nets. 2014.

[3] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[4] Teck Yian Lim Alexander G. Schwing Mark Hasegawa-Johnson Minh N. Do Raymond A. Yeh, Chen Chen. Semantic image inpainting with deep generative models. 2017.

[5] Gozde Unal Rur Demir. Patch-based image inpainting with generative adversarial networks. 2018.

[6] Hiroshi Ishikawa Satoshi Iizuka, Edgas Simo-Serra. Globally and locally consistent image completion. 2017.

[7] Deepak Pathak Philipp Krahenb ¨ uhl Jeff Donahue Trevor Darrell Alexei A. Efros. Context encoders: Feature learning by inpainting. 2016.