

Compressão de Imagens com Perdas usando Redes Neurais

Raphael Soares Ramos

Universidade de Brasília

raphael.soares@nubank.com.br

27 de outubro de 2019

Overview I

- 1 Introdução
 - Definições
 - Motivação
 - Compressão de Imagens
 - Hipótese
 - Objetivos
- 2 Fundamentação Teórica
 - JPEG
 - Redes Neurais
- 3 Metodologia
 - Bases de Dados
 - Modelos Desenvolvidos
- 4 Experimentos
 - JPEG
 - Modelo 1

Overview II

- Modelo 2
- Modelo 3
- JPEG x Modelo 3

5 Conclusão

Introdução

Codificação de Dados

- Transformação feita nos dados para atingir um certo objetivo.
- Compressão (redução do comprimento) vs. Criptografia (proteger sigilo ou integridade do que os dados significam).

Compressão de Dados

- Processo de codificar uma determinada informação utilizando uma menor representação.
- Arte ou ciência de representar informação de forma compacta [Sayood et al. 2017].
- Representações compactas são criadas identificando e usando estruturas existentes nos dados para que seja possível extrair redundâncias nos dados [Sayood et al. 2017].
- Dois tipos: com perda ($x \neq \hat{x}$) e sem perda ($x = \hat{x}$).

Porquê comprimir

- Geração e uso cada vez maior de dados digitais (muitos são redundantes e irrelevantes para determinadas aplicações).
- Representar digitalmente 1 segundo de vídeo sem compressão usando o formato CCIR 601 requer mais de 20 *megabytes* de armazenamento ou 160 megabits para transmissão [Sayood et al. 2017].
- Imagem Monocromática com resolução 512×512 :
$$Taxa = \frac{512 \cdot 512 \cdot 8}{10^6} = 2.097 Mbits.$$
- Aumentar capacidade de armazenamento e de transmissão de um sistema.
- Para serviços de streaming de mídia como Netflix, não usar compressão não é uma opção.

Medidas de Desempenho em Compressão de Imagens

- Representar com o menor número possíveis de bits, preservando a qualidade e a inteligibilidade necessárias à sua aplicação.
- Taxa vs. Distorção.
- *Mean Squared Error* é uma medida de distorção.

$$MSE = \frac{1}{n} \sum_{n=1}^N (x(n) - \hat{x}(n))^2 \quad (1)$$

Algoritmos de Compressão de Imagens

- Exploram características imperfeitas da nossa percepção e propriedades estatísticas para fornecer resultados superiores quando comparados com métodos de compressão de dados genéricos.
- Examinada durante anos por pesquisadores e times como o *Joint Pictures Experts Group*.
- Duas fases: modelagem (descrever redundâncias em forma de um modelo) e codificação (codificar descrição do modelo e de como os dados diferem do modelo).
- Alguns métodos de compressão:
 - ① JPEG [G. K. Wallace et al. 1993]
 - ② JPEG2000 [C. Christopoulos et al. 2000]
 - ③ BPG [Bellard et al. 2019]
 - ④ WebP [Google 2019]

Hipótese

- Se modelos baseados em *autoencoders* convolucionais recorrentes são competitivos com os clássicos codecs *JPEG* e *JPEG2000* para imagens com muito conteúdo de alta frequência;
- Se o codificador de entropia utilizado no latente será capaz de comprimir em proporções semelhantes para todos os níveis de resíduos;
- Se usar imagens que codificadores clássicos têm dificuldade para comprimir para treinamento do modelo é benéfico para os resultados dele;
- Se há um grande impacto nos resultados ao usar funções de perdas variadas;
- Se os resultados dos *autoencoders* serão melhores ao trabalhar com um diferente domínio para os dados de entrada, um que beneficie o processo de aprendizagem via atualização de gradientes. Ou seja, se é benéfico para o desempenho do modelo aplicar uma transformação nos dados de entrada que evite problemas como gradientes muito

Objetivos gerais e específicos

- Deseja-se construir e avaliar o desempenho de um framework de compressão de imagens ponta a ponta (modelagem e codificação) usando autoencoders empilhados convolucionais recorrentes com o auxílio de um codificador de entropia para comprimir o código.

Fundamentação Teórica

Formato de Arquivos

- Arquivos comprimidos pelo método *JPEG* são normalmente descritos no formato *JPEG File Interchange Format (JFIF)*, que é uma limitação do padrão *JPEG* completo (muitos espaços de cores e modos de operação).
 - 1 Primeiro é feita conversão do espaço de cor para YCbCr.
 - 2 É usado um fator para reduzir a quantidade de pixels nos componentes de crominância. Normalmente é usado um fator de 2 nas duas direções.

Funcionamento

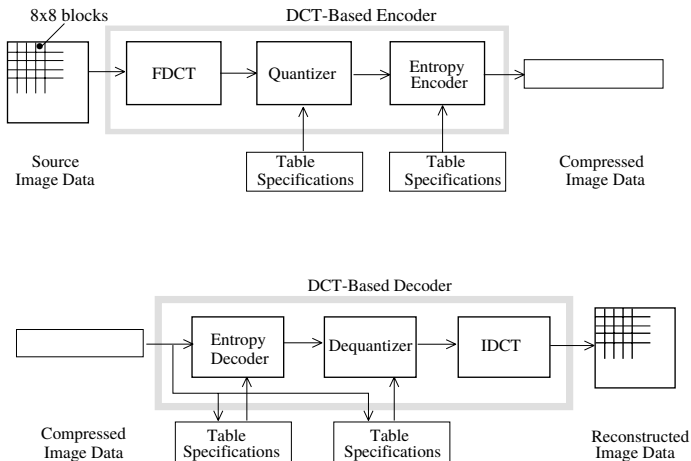


Figura: Diagrama geral que ilustra o funcionamento (codificação e decodificação) do método de compressão JPEG. Fonte: [G. K. Wallace et al. 1993]

Transformada Discreta de Cossenos (DCT)

- Cada bloco 8x8 pode ser replicado por 64 (8x8) ondas de cossenos.
- Analisa as frequências dos valores originais da imagem ao longo de cada linha e coluna usando um conjunto de ondas de cossenos oscilando em diferentes frequências e amplitudes. Cada bloco é representado usando estas ondas.

Transformada Discreta de Cossenos (DCT)

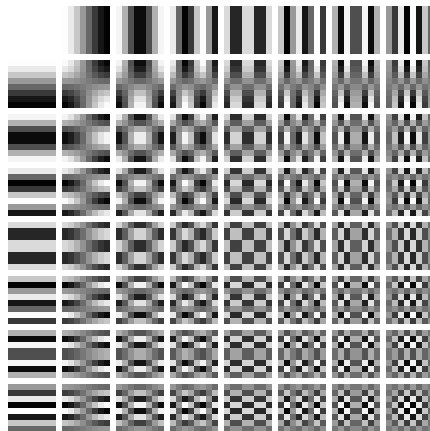


Figura: 64 ondas base de cossenos que produzem qualquer imagem 8x8. A DCT-II irá calcular os coeficientes - a contribuição de cada uma dessas ondas que somadas irão recriar a imagem perfeitamente. Fonte: [Wikimedia Commons 2015]

Quantização

- Para o *encoder* será usada a seguinte equação:

$$\left[F^Q(u, v) = \frac{F(u, v)}{Q(u, v)} \right] \quad (2)$$

- 1 Parte em que há maior perda de informação da imagem.
 - 2 Cada um dos 64 coeficientes DCT $F(u, v)$ são uniformemente quantizados em conjunto com uma tabela de quantização de 64 elementos, que é definida pelo nível de qualidade escolhido para a aplicação.
 - 3 Remove altas frequências definindo altos valores para $Q(u, v)$ (passo de quantização)
- No final, para cada bloco 8 por 8 da imagem original, existirá uma matriz quantizada.

Codificação

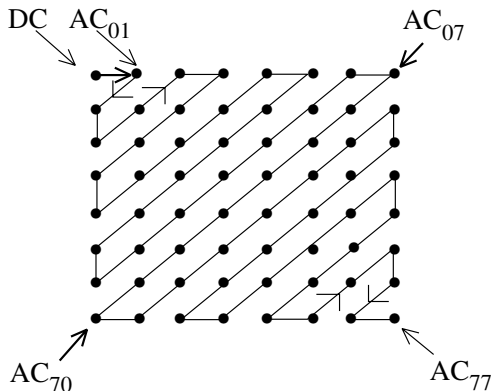
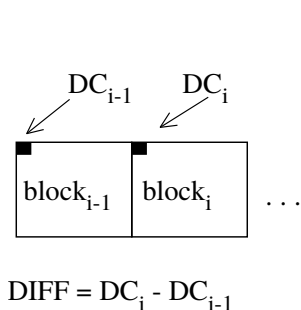


Figura: Codificação dos Direct Current (DC) coeficientes (esquerda) e sequência zig-zag usada para codificar todos os coeficientes (direita).

Fonte: [G. K. Wallace et al. 1993]

Conceitos Básicos

- Para muitas tarefas em inteligência artificial, é necessário extrair *features* (características) dos dados.
 - ① As vezes é difícil saber quais *features* devem ser extraídas.
 - ② Solução: aprender a representação dos dados (*representation learning*).
- *Deep learning* resolve o problema de *representation learning* introduzindo representações que são expressadas em termos de outras representações mais simples.

Convoluções

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (3)$$

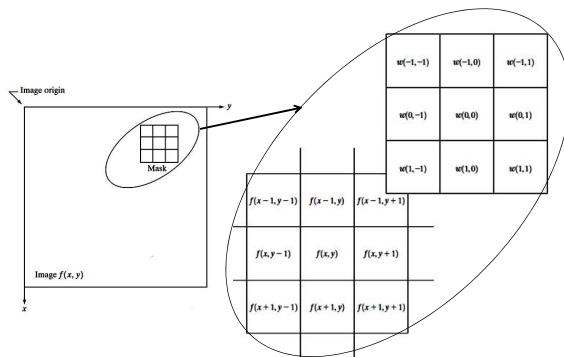


Figura: Operação de convolução. Fonte: [Zaghetto et. al 2018]

Autoencoder

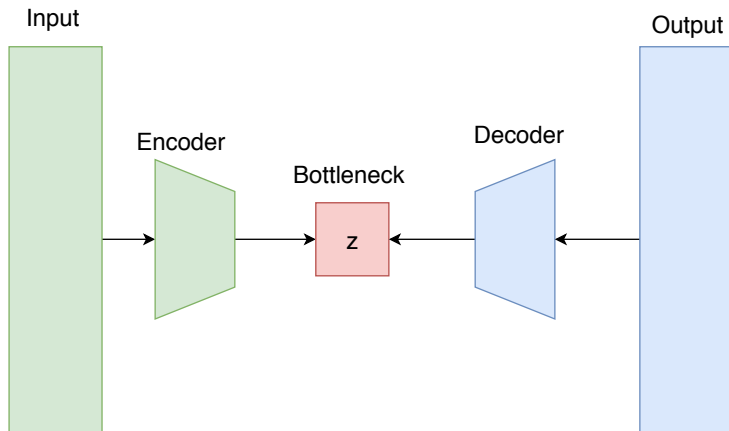


Figura: Ilustração de um *autoencoder*.

Compressão de Imagens com Taxa Variável usando Redes Neurais Recorrentes [Toderici et. al 2015]

- Função de binarização $b(x)$:

$$b(x) = x + \epsilon \in \{-1, 1\}, \epsilon \sim \begin{cases} 1 - x & \text{com probabilidade } \frac{1+x}{2}, \\ -1 - x & \text{com probabilidade } \frac{1-x}{2}, \end{cases} \quad (4)$$

- Binarizador:

$$B(x) = b(\tanh(W^{bin}x + b^{bin})) \quad (5)$$

- Em tempo de teste b é substituída por:

$$b^{inf}(x) = \begin{cases} -1, & \text{se } x < 0, \\ 1, & \text{caso contrário.} \end{cases} \quad (6)$$

- Autoencoder: $D(B(E(x)))$

Metodologia

Bases de Dados

- Foram pegas imagens de CLIC, DIV2K e EYE.
- Foram gerados 6,231,440 *patches* com 32 pixels de largura e altura. Estes *patches* foram divididos em 5 bases:
 - 1 **BD0**: formada por 1248978 de *patches* que pertencem ao grupo dos 20% com menor entropia;
 - 2 **BD1**: formada por 1251421 de *patches* que pertencem ao grupo dos que estão na faixa 40% à 60% (porcentagem dada pelo *patch* com maior entropia);
 - 3 **BD2**: formada por 1248725 de *patches* que pertencem ao grupo dos 20% com maior entropia;
 - 4 **BD3**: formada por 1247033 de *patches* pegos de forma aleatória. Correspondem à 20% do total.
 - 5 **BD4**: formada por 1246698 de *patches*. 20% do total retirados aleatoriamente dos 50% de *patches* com maior entropia.

Bases de Dados

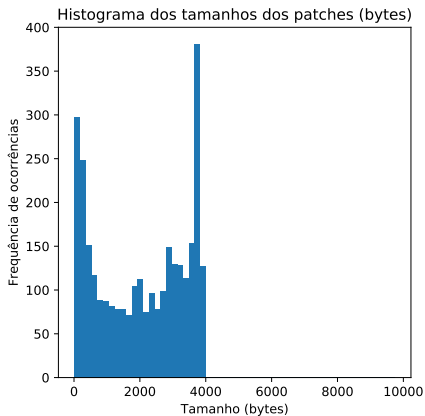


Figura: Histograma de todas as bases de dados.

Bases de Dados

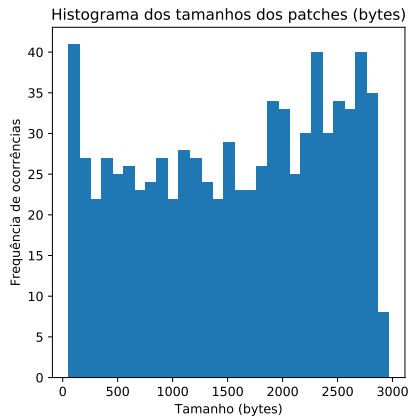


Figura: Histograma da BD2.

Modelo 1

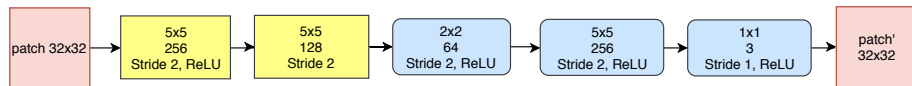


Figura: Ilustração do Modelo 1

Modelo 2

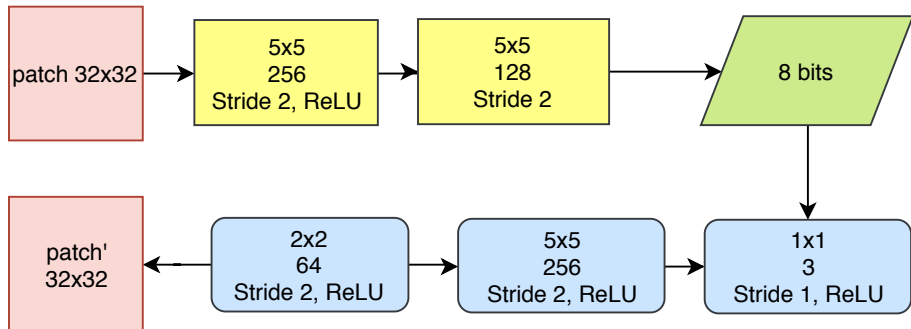


Figura: Ilustração do Modelo 2.

Modelo 3

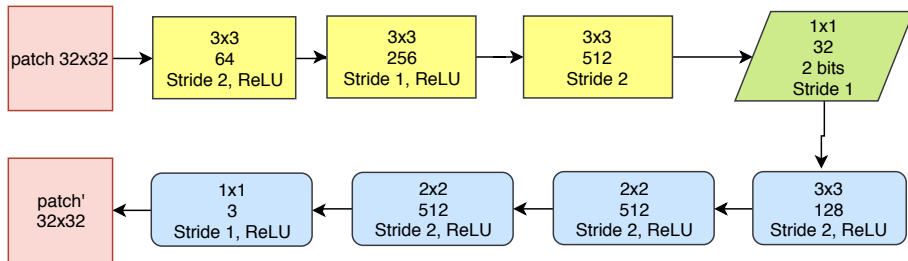


Figura: Ilustração do Modelo 3.

Experimentos

Desempenho JPEG

Tabela: Tabela contendo médias obtidas pelo **JPEG** em cada uma das bases de teste utilizadas

Bases	BPP	PSNR	SSIM	MSSIM	Quality
CLIC Mobile test (patches 32)	8	44.23	0.98	0.99	94.06
CLIC Mobile test	2	39.58	0.96	0.99	88.69
Kodak	2	36.77	0.95	0.99	85.91
BD0	8	57.85	0.99	0.99	99.18
BD1	8	42.94	0.97	0.99	95.94
BD2	8	32.31	0.95	0.99	83.69
BD3	8	43.84	0.96	0.99	93.94
BD4	8	36.72	0.96	0.99	89.68

Treino e Teste em todas as BDs

Tabela: Tabela contendo o valor da *PSNR*, em decíbeis, dos testes do **Modelo 1** com o uso do otimizador *Adam* e *learning rate* fixa. As linhas denotam a base de treino utilizada. As colunas denotam as bases de teste usadas para avaliação do modelo. O índice “todas” se refere ao uso de todas as imagens de todas as bases **BD** para treino

Treino (linhas) x Teste (colunas)	BD0	BD1	BD2	BD3	BD4
BD0	49.66	38.07	26.34	38.05	31.13
BD1	47.57	44.08	34.54	42.66	38.69
BD2	53.69	51.16	44.07	50.06	47.14
BD3	50.62	47.88	39.76	46.59	43.25
BD4	47.30	46.98	41.10	45.63	43.75
Todas	46.77	46.35	43.94	45.88	45.08

Resultados

Tabela: Tabela contendo o valor da *PSNR*, em decíbeis, dos testes do Modelo 1 com o uso do otimizador *Adam* e *learning rate* fixa.

Treino (linhas) x Teste (coluna)	CLIC Mobile test
BD0	34.77
BD1	44.11
BD2	48.97
BD3	45.34
BD4	42.42
Todas	51.27
Todas + CLIC Mobile train	55.78
Todas + Clic Mobile train + Clic Professional train	47.26
CLIC Mobile Train	46.63

Resultados

Tabela: Tabela contendo os resultados do Modelo 2 para as métricas visuais *PSNR*, *SSIM* e *MS-SSIM* a uma taxa nominal de 8 bits por pixel

Bases de Treino e Teste	BPP	PSNR	SSIM	MS-SSIM	Épocas
CLIC Mobile test	8	35.03	0.94	0.98	30
BD1	8	35.26	0.93	0.98	30
BD2	8	29.10	0.95	0.98	30

Resultados

Tabela: Tabela contendo os melhores resultados do Modelo 3

Bases	BPP	PSNR	SSIM	MS-SSIM
CLIC Mobile test (patches 32)	2	33.75	0.92	0.97
Kodak (patches 32)	2	31.46	0.88	0.96
BD0	2	40.24	0.97	0.99
BD1	2	35.00	0.91	0.98
BD2	2	27.53	0.91	0.97
BD3	2	33.27	0.91	0.97
BD4	2	30.16	0.90	0.97

Curva Distorção vs. Taxa

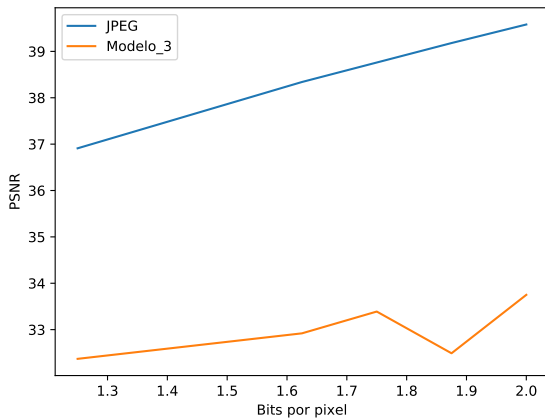


Figura: Comparação Modelo 3 e *JPEG* na métrica PSNR.

Curva Distorção vs. Taxa

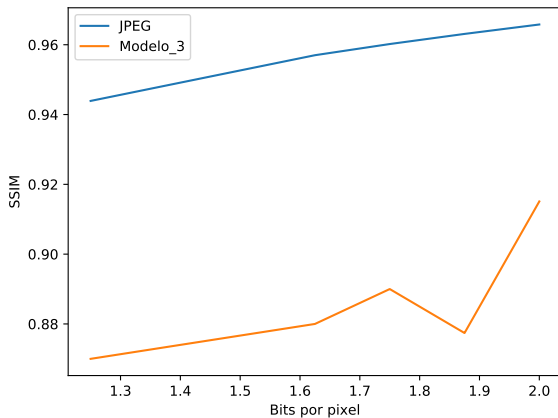


Figura: Comparação Modelo 3 e *JPEG* na métrica SSIM.

Conclusão

Limitações do Trabalho

- Custo computacional superior aos codecs clássicos.
- Algoritmos flexíveis, porém quantização é uma operação não diferenciável o que dificulta treinamento de redes neurais.

Trabalho de Graduação 2

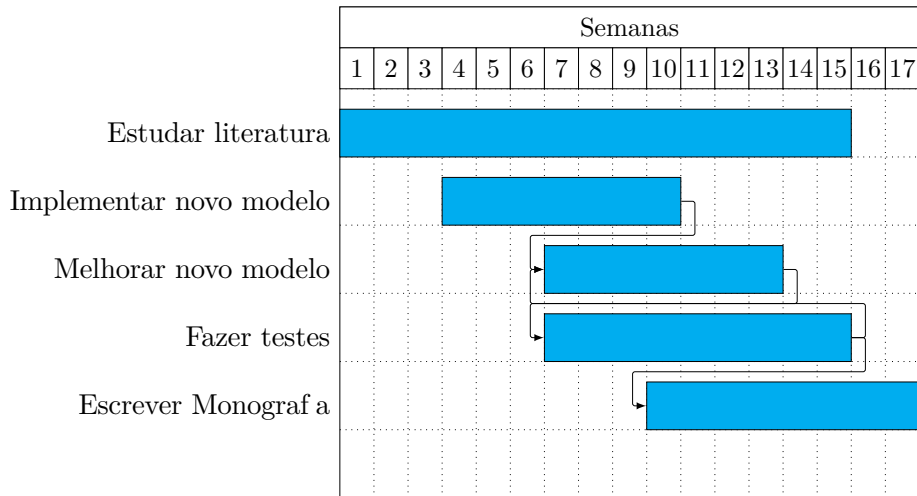


Diagrama de Gantt

References I



Khalid Sayood

Introduction to Data Compression

Morgan Kaufmann



Gregory K. Wallace

The JPEG Still Picture Compression Standard

In: IEEE Transactions on Consumer Electronics



C. Christopoulos and A. Skodras and T. Ebrahimi

The JPEG2000 Still Image Coding System: An Overview

In: IEEE Transactions on Consumer Electronics



Fabrice Bellard

BPG image format

In: BPG

References II



WebP

Compression Techniques

In: [Compression](#)



Michael A. Nielsen

Neural Networks and Deep Learning

In: [Determination Press](#)



Ian Goodfellow and Yoshua Bengio and Aaron Courville

Deep Learning

In: [MIT Press](#)



Alexandre Zaghetto

Introduction to Image Processing

In: [GitHub Repository](#)

References III



[Wikimedia Commons](#)

Wikimedia Commons, The Free Media Repository

In: [DCT Figure](#)



[Leslie N. Smith](#)

Cyclical Learning Rates for Training Neural Networks

In: [Proceedings - IEEE Winter Conference on Applications of Computer Vision](#)



[Brad Kenstler](#)

Cyclical Learning Rate (CLR)

In: [GitHub Repository](#)



[CS231n](#)

CS231n Convolutional Neural Networks for Visual Recognition

In: [CS231n](#)

References IV



Toderici, George and Vincent, Damien and Johnston, Nick and Jin Hwang, Sung and Minnen, David and Shor, Joel and Covell, Michele

Variable Rate Image Compression with Recurrent Neural Networks

In: ICLR 2016