FERRAMENTAS PARA PRÉ-PROCESSAMENTO E REDUÇÃO DIMENSIONAL DE IMAGENS NO RECONHECIMENTO E DESVIO DE OBSTÁCULOS POR UM ROBÔ MÓVEL

Luís Eduardo de Souza Cintra

Luiz Afonso Batalha Marão

Prof. Dr. Glauco Augusto de Paula Caurin

Escola de Engenharia de São Carlos - USP

luis.cintradu@usp.br

Objetivos

O tratamento de quantidades significativas de dados é um tema extremamente importante dentro do panorama tecnológico. Vale evidenciar que a tarefa de processar grandes volumes de informações pode ser facilitada por meio das técnicas de redução dimensional: um grupo de metodologias matemáticas e computacionais capazes de comprimir um campo de dados mediante uma perda informacional mínima.

Assim, o objetivo desse projeto é explorar e qualificar dois dos mais relevantes destes métodos: PCA - *Principal Component Analysis* - e os Autoencoders; através de suas implementações no sistema de um robô móvel autônomo [1]. Essas ferramentas atuam no processamento das imagens capturadas por uma câmera frontal, visando incrementar a eficiência da rede neural de controle tanto em sua fase de treinamento, quanto na sua aplicação em tempo real. Posteriormente, verificam-se os ganhos e perdas obtidos analisando-se a performance do robô durante a realização de um percurso em um campo de obstáculos.

Métodos e Procedimentos

Com o intuito de ampliar o horizonte de possibilidades e construir um ambiente de análise completo, optou-se pelo uso de 4 *datasets*, cada um com 8076 imagens no formato *grayscale* nas seguintes dimensões: 32×32, 64×64, 128×128 e 256×256.

Para a implementação do PCA [2] foram determinados os componentes principais de cada *dataset* com base em uma variância de 99%, de forma que esses elementos são fixados e posteriormente aplicados na compressão de cada imagem individual obtida pelo Rover, fornecendo uma perspectiva de generalização para o método.

Para os Autoencoders [3], foram estipuladas padronizações durante a composição dos modelos, evitando irregularidades e mantendo o escopo do projeto bem delimitado. Consoante a essa determinação, as combinações de hiperparâmetros das redes neurais de convolução utilizadas foram investigadas por meio de um algoritmo de força bruta denominado *grid search*.

Em seguida, para que fosse possível reduzir o número de testagens, foi-se desenvolvido um algoritmo de pré-avaliação, o SCORE, com o objetivo de introduzir uma métrica de seleção que pudesse fornecer uma pontuação para cada rede com base em três variáveis: precisão da reconstrução, taxa de redução e número de FLOPS do decoder.

Como análise final, as experimentações foram comparadas com *baselines* visando analisar dois principais aspectos que foram condensados no Custo Performático (CP):

$$\text{CP} = \frac{\Delta\text{Taxa de sucesso}}{\Delta\text{Tempo de treinamento}} \qquad (1)$$

Resultados

Tabela 1: Custo Performático dos modelos em relação a seus respectivos baselines

DATASET 32	
MODELO	**CUSTO PERFORMÁTICO**
PCA	-1.21 ± 0.03
g32_2c_8b_128f_128a	-0.09 ± 0.29
g32_2c_8b_128f_128a	-0.25 ± 0.11
g32_2c_8b_128f_64a	-0.28 ± 0.10
g32_2c_4b_128f_32a	-0.34 ± 0.06
g32_2c_4b_128f_64a	-0.35 ± 0.08
g32_3c_8b_64f_128a	-0.41 ± 0.04
g32_3c_16b_64f_64a	-0.48 ± 0.04
g32_3c_8b_64f_64a	-0.59 ± 0.01
g32_3c_4b_64f_128a	-0.60 ± 0.01
g32_3c_4b_64f_16a	-0.67 ± 0.00

DATASET 64	
MODELO	**CUSTO PERFORMÁTICO**
PCA	-1.80 ± 0.03
g64_4c_32b_32f_64a	-0.17 ± 0.24
g64_2c_4b_128f_8a	-0.33 ± 0.10
g64_3c_8b_64f_128a	-0.45 ± 0.04
g64_3c_16b_64f_128a	-0.49 ± 0.05
g64_3c_8b_64f_64a	-0.50 ± 0.06
g64_2c_4b_128f_128a	-0.51 ± 0.03
g64_3c_16b_64f_16a	-0.52 ± 0.05
g64_3c_4b_64f_32a	-0.60 ± 0.01
g64_3c_4b_64f_128a	-0.70 ± 0.00
g64_4c_8b_32f_128a	-1.02 ± 0.04

DATASET 128	
MODELO	**CUSTO PERFORMÁTICO**
PCA	-0.92 ± 0.01
g128_4c_8b_32f_64a	-0.12 ± 0.22
g128_4c_16b_32f_32a	-0.15 ± 0.14
g128_4c_8b_32f_32a	-0.15 ± 0.15
g128_3c_4b_64f_128a	-0.18 ± 0.13
g128_2c_4b_128f_64a	-0.21 ± 0.11
g128_3c_8b_64f_128a	-0.22 ± 0.11
g128_3c_4b_64f_32a	-0.24 ± 0.10
g128_2c_4b_128f_32a	-0.32 ± 0.07
g128_4c_16b_32f_128a	-0.36 ± 0.05
g128_3c_8b_64f_8a	-0.41 ± 0.00

DATASET 256	
MODELO	**CUSTO PERFORMÁTICO**
PCA	-0.83 ± 0.01
g256_4c_8b_32f_32a	-0.17 ± 0.09
g256_3c_4b_64f_128a	-0.28 ± 0.05
g256_4c_16b_32f_64a	-0.30 ± 0.05
g256_4c_32b_32f_64a	-0.32 ± 0.05
g256_3c_8b_64f_64a	-0.33 ± 0.06
g256_4c_4b_32f_32a	-0.35 ± 0.05
g256_4c_4b_32f_128a	-0.37 ± 0.03
g256_2c_4b_128f_128a	-0.38 ± 0.05
g256_2c_4b_128f_32a	-0.45 ± 0.05
g256_4c_32b_32f_32a	-0.50 ± 0.03

Tabela 2: Significado dos elementos que compõe o código de descrição das redes

g	Dimensão da imagem grayscale
c	Quantidade de camadas
b	Número de neurônios no bottleneck
f	Número de filtros na camada mais interna
a	Número de neurônios na camada auxiliar

A Tabela 1 demonstra os resultados finais obtidos por uma série de 40 modelos que obtiveram maior SCORE. Nota-se que os Autoencoders apresentaram um CP menor em módulo quando comparados ao PCA, um indício que o uso de reduções lineares trazem um prejuízo maior ao sistema de controle.

Por fim, cabe ao experimentador estabelecer um limite máximo para o custo performático que respeite os parâmetros da aplicação final.

Conclusões

É possível visualizar que os resultados obtidos foram positivos para a aplicação dos Autoencoders na tarefa de redução dimensional, obtendo em sua grande maioria um CP, em módulo, inferior a 1. Por outro lado, o PCA demonstrou ser um método mais custoso para esta aplicação, motivado principalmente por quedas significativas na taxa de sucesso.

Os modelos proporcionaram uma economia volumosa no tempo de treinamento quando comparados às linhas de base, demonstrando que o projeto foi bem-sucedido na geração de eficiência.

Referências Bibliográficas

[1] Marão, L.A.B. Deep Reinforcement Learning Multi-Sensor Based Navigation and Control. 82 p. Qualificação de doutorado. Escola de Engenharia de São Carlos, Universidade de São Paulo, 2020

[2] JOLIFFE, Ian. Principal Component Analysis. 2 ed. Springer, 2002

[3] GOODFELLOW, Ian. et al. Deep Learning. MIT Press, 2016. Disponível em: http://www.deeplearningbook.org