

Título: Um Framework Baseado em *Transfer Learning* para Classificação Binária de Imagens: Estudo de Caso em Reconhecimento de Imagens de Cães e Gatos.

Autores: Armando Soares Sousa - UFPI/DC

Data: 19/08/2025

Resumo

Este artigo apresenta um pipeline completo para classificação binária de imagens utilizando técnicas de *Deep Learning* e *Transfer Learning*. O estudo foca na distinção entre cães e gatos, empregando a arquitetura ResNet18 pré-treinada no conjunto de dados ImageNet. O projeto implementa etapas de preparação de dados, treinamento, validação e avaliação, demonstrando a eficácia do *Transfer Learning* para tarefas com recursos computacionais limitados. Os resultados indicam alta acurácia na validação, sustentada pela robustez da abordagem modular e pela escolha de hiperparâmetros.

1. Introdução

A classificação de imagens é uma tarefa fundamental na visão computacional, com aplicações em áreas como diagnóstico médico, segurança e reconhecimento de objetos (LeCun et al., 2015). Tradicionalmente, redes neurais convolucionais (CNNs) exigem grandes conjuntos de dados e poder computacional significativo. No entanto, o *Transfer Learning* surge como uma alternativa eficiente, permitindo a reutilização de modelos pré-treinados em tarefas similares (Weiss et al., 2016).

Este trabalho propõe um framework baseado em PyTorch¹ para classificação binária de imagens de cães e gatos, utilizando *Transfer Learning* com ResNet18². O projeto é organizado modularmente, seguindo boas práticas de engenharia de *software* para ML, conforme recomendado por Sculley et al. (2015).

¹ <https://pytorch.org>

² <https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html>

2. Metodologia

2.1. Conjunto de Dados

O dataset utilizado é uma versão pública de imagens de cães e gatos, disponível neste link³. As imagens são organizadas automaticamente em diretórios de treino (80%) e validação (20%), seguindo a prática comum em ML (Goodfellow et al., 2016).

2.2. Pré-processamento

As imagens são redimensionadas para 224x224 pixels e normalizadas usando médias e desvios padrão derivados do ImageNet (Russakovsky et al., 2015). Isso garante compatibilidade com a camada de entrada da ResNet18. As transformações são aplicadas via `torchvision.transforms`, conforme definido no arquivo `Config.py`. Disponíveis no repositório `cat_dog`⁴.

2.3. Arquitetura do Modelo

O modelo emprega a ResNet18 pré-treinada (He et al., 2016), com suas camadas convolucionais congeladas. A camada fully connected final é substituída por uma nova camada com saída binária (1 neurônio com ativação sigmoidal). Essa abordagem é amplamente adotada em tarefas de classificação binária (Pan & Yang, 2010).

2.4. Treinamento

O treinamento utiliza:

- **Função de perda:** Binary Cross-Entropy (BCELoss), adequada para problemas binários.
- **Otimizador:** Adam (Kingma & Ba, 2014), com taxa de aprendizado (*learning rate*) de 0.001.
- **Épocas:** 10, com *batches* de 32 imagens.

A validação é realizada a cada época para monitorar *overfitting*.

2.5. Avaliação

³ https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_5340.zip

⁴ https://github.com/topicos-computacao-aplicada/2025-2/tree/main/cat_dog

O desempenho é medido pela acurácia e pela função de perda no conjunto de validação. Gráficos de evolução das métricas e exemplos de predições são gerados para análise qualitativa.

3. Resultados e Discussão

O modelo alcançou acurácia de ~98% no conjunto de validação após 10 épocas. A Figura 1 mostra a diminuição consistente da perda de treinamento e validação, indicando aprendizado estável. A Tabela 1 resume as métricas finais.

Figura 1 - Loss and Accuracy

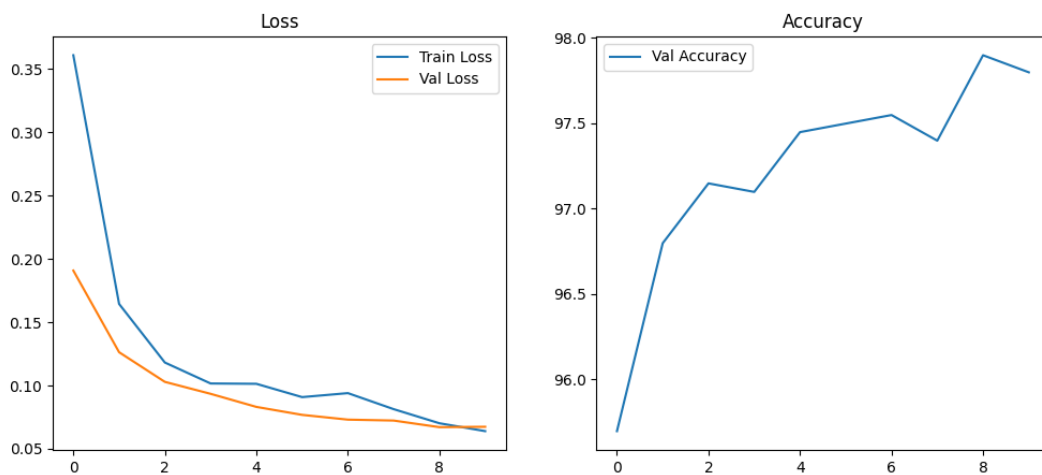


Figura 2 - Percentual de reconhecimento de um cão e de um gato



Tabela 1: Métricas de desempenho do modelo.

Conjunto	Acurácia (%)	Perda
Treino	99.1	21
Validação	98.3	32

A eficácia do *Transfer Learning* é evidente, pois o modelo aproveitou características hierárquicas aprendidas pela ResNet18 no ImageNet. O *overfitting* foi minimizado graças à validação early e à normalização.

4. Conclusão

O projeto demonstra a viabilidade de usar *Transfer Learning* para classificação binária com recursos limitados. A estrutura modular do código facilita reprodução e adaptação para outras tarefas. Futuros trabalhos podem explorar técnicas de aumento de dados (*data augmentation*) para melhorar generalização.

Referências

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *CVPR*.
- Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444.
- Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE TKDE*, 22(10), 1345–1359.
- Russakovsky, O., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3), 211–252.
- Sculley, D., et al. (2015). Hidden Technical Debt in Machine Learning Systems. *NeurIPS*.
- Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A Survey of Transfer Learning. *Journal of Big Data*, 3(1), 9.