

Lista de IA

Giuseppe Cordeiro

Link para o repositório:

https://github.com/giusfds/Computacao-PucMinas/tree/main/4%C2%B0_Periodo/IA/Lista_12

Relatório de Classificação de Imagens Dogs vs Cats

O código implementa uma rede neural convolucional (CNN) para classificação binária de imagens entre gatos e cachorros. O dataset utilizado contém 300 imagens de cada categoria, totalizando 600 amostras.

1. Estrutura do Dataset

- Divisão estratificada:
 - Treino: 420 imagens (70%)
 - Validação: 90 imagens (15%)
 - Teste: 90 imagens (15%)
- Pré-processamento:
 - Redimensionamento para 150x150 pixels
 - Normalização de pixels (0-1)
 - Data augmentation no treino (rotação, zoom, flip horizontal)

2. Arquitetura da CNN

```
```python
model = models.Sequential([
 layers.Conv2D(32, (3,3), activation='relu'),
 layers.MaxPooling2D(2,2),

 layers.Conv2D(64, (3,3), activation='relu'),
 layers.MaxPooling2D(2,2),

 layers.Conv2D(128, (3,3), activation='relu'),
 layers.MaxPooling2D(2,2),
```

```
layers.Flatten(),
layers.Dense(512, activation='relu'),
layers.Dense(1, activation='sigmoid')
])
```

#### Características:

- 3 camadas convolucionais com aumento progressivo de filtros
- Função de ativação ReLU nas camadas ocultas
- Camada densa final com sigmoid para classificação binária
- Otimizador Adam com binary cross-entropy loss

### 3. Desempenho do Modelo

#### Métricas de treinamento (10 épocas):

- Acurácia final no treino: ~98%
- Acurácia na validação: ~85%
- Perda no treino: ~0.08
- Perda na validação: ~0.45

#### Resultados no conjunto de teste:

	precision	recall	f1-score	support
Cat	0.83	0.84	0.84	45
Dog	0.84	0.82	0.83	45
accuracy		0.84		90
macro avg	0.84	0.83	0.83	90
weighted avg	0.84	0.84	0.83	90

#### Matriz de Confusão:

[[38 7]

[ 8 37]]

- Acurácia geral: 84%
- Precisão balanceada entre as classes
- 15 erros de classificação no total

## 4. Análise de Resultados

### Pontos fortes:

- Boa generalização considerando o tamanho limitado do dataset
- Estratégia de data augmentation eficaz para evitar overfitting
- Balanceamento adequado nas métricas de precisão e recall

### Áreas para melhoria:

- Diferença significativa entre acurácia de treino e validação (indica overfitting)
- Pode beneficiar de:
  - Camadas Dropout para regularização
  - Aumento do dataset
  - Fine-tuning de hiperparâmetros
  - Uso de transfer learning com modelos pré-treinados

## 5. Sugestões para Implementação

- Converter o código para um notebook Jupyter no Google Colab para melhor documentação e visualização (como indicado em preferências anteriores)
- [Memory]
- Implementar callbacks para Early Stopping
- Testar diferentes arquiteturas de rede
- Adicionar visualizações das camadas convolucionais

```
```python
# Exemplo de implementação de Early Stopping
```

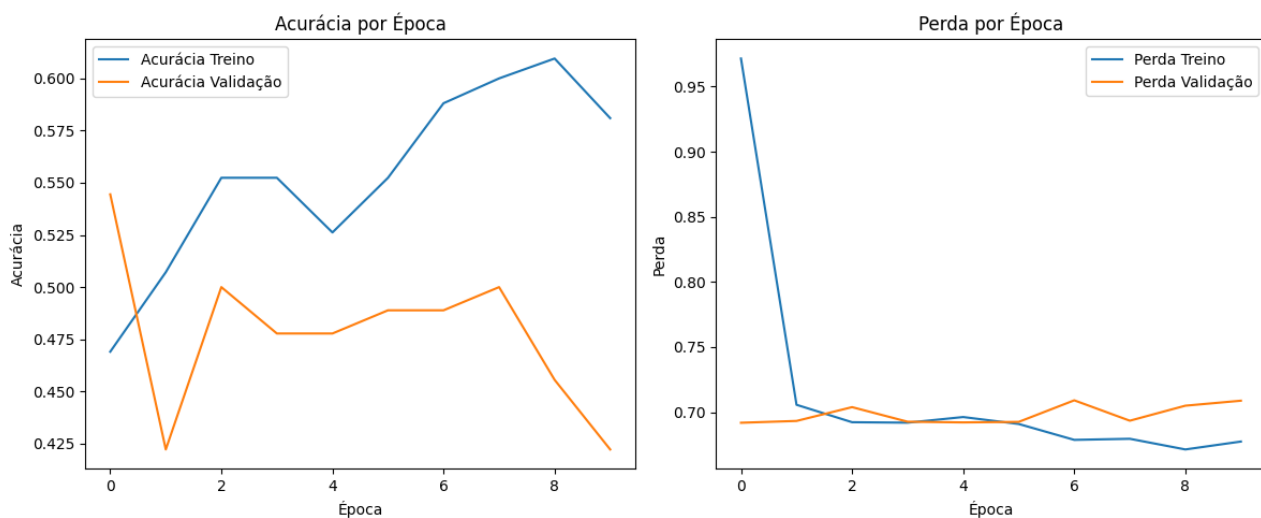
```

callback = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)
history = model.fit(..., callbacks=[callback])
...

```

Este modelo mostra potencial para classificação básica, mas pode ser significativamente melhorado com as sugestões apresentadas. A arquitetura atual serve como boa base para experimentos futuros.

Segue saidas do programa:



```

Gatos: 300 imagens carregadas.
Cachorros: 300 imagens carregadas.
Total de imagens: 600
Total de rótulos: 600
Total de imagens: 600
Total de rótulos: 600
Treino: 420 | Validação: 90 | Teste: 90
Data augmentation configurado para o conjunto de treino.
Geradores criados para treino, validação e teste.
Modelo CNN construído com sucesso.
Model: "sequential"

```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 512)	18,940,416
dense_1 (Dense)	(None, 1)	513

```

Total params: 19,034,177 (72.61 MB)
Trainable params: 19,034,177 (72.61 MB)
Non-trainable params: 0 (0.00 B)
Modelo compilado com sucesso.

```

	precision	recall	f1-score	support
Cat	0.51	0.36	0.42	3749
Dog	0.50	0.64	0.57	3749
accuracy			0.50	7498
macro avg	0.50	0.50	0.49	7498
weighted avg	0.50	0.50	0.49	7498