

Lab. 07 (Parte 2): Duelo de Hipóteses e Telemetria MLOps

Aléssio Miranda Júnior

Agosto - 2026/2

Lab. 07 (Parte 2): Duelo de Hipóteses e Telemetria MLOps

★ Objetivo da Sessão 2

Na Parte 1, você construiu um classificador com Transfer Learning que atingiu mais de 95% de acurácia com facilidade. Mas na engenharia de computação de alto nível, **um número alto sem auditoria empírica é perigoso**. Hoje você colocará essa solução sob prova de fogo: enfrentará três armadilhas clássicas da visão computacional (o *Shortcut Learning* de redes treinadas do zero, a destruição de pesos pelo descongelamento precoce e a explosão de parâmetros gerada pelo Flatten). Você implementará **Telemetria MLOps**, registrando o histórico de experimentos em CSV e emitindo sua conclusão técnica formal.

1 Parte 1: O Desafio de Auditoria e o Setup de MLOps (5 min)

A Missão de Auditoria Técnica

A diretoria de engenharia de TechCity questionou a entrega do Lab 07 (Parte 1):

“Vocês usaram um modelo do Google congelado e afirmam que ele é o melhor. Mas onde estão as evidências empíricas? E se uma CNN simples treinada do zero for suficiente? E se descongelarmos o modelo para aprender mais? E se usarmos Flatten em vez de GlobalAveragePooling?”

Para responder a essas perguntas com rigor corporativo, você executará quatro baterias controladas de testes, salvando métricas, parâmetros e tempo em um arquivo de governança: `historico_experimentos.csv`.

A Dinâmica do Laboratório

Você trabalhará iterando na função de treinamento e registrando cada hipótese no painel de controle acumulativo:

Cenário	Hipótese Testada	Configuração no Código	Comportamento Esperado
O	Aprender tudo do zero	CNN simples Conv2D + MaxPool	Ilusão de acurácia via Shortcut Learning
A	Padrão Ouro (Baseline)	MobileNetV2 congelada + GAP	Referência de acurácia (> 95%) e estabilidade
B	Descongelamento precoce	<code>base_model.trainable = True</code>	Catastrophic Forgetting (queda de acurácia)
C	Explosão dimensional	Substituir GAP por Flatten()	Salto de 20k para 184k parâmetros treináveis

2 Parte 2: Implementando a Telemetria de Experimentos (8 min)

Abra o arquivo `src/transfer_learning.py`. Adicionaremos o mecanismo de registro automático de experimentos em CSV:

```
1 import csv
2 import datetime
3
4 def registrar_e_exibir_historico(descricao, modelo, base, loss_t,
5                                acc_t, epocas,
6                                arquivo_csv="historico_experimentos.csv"):
7     """Registra metricas no CSV de governanca MLOps e exibe o painel
8     acumulado."""
```

```

7     t_params = sum(tf.keras.backend.count_params(w) for w in modelo.
           trainable_weights)
8     c_params = sum(tf.keras.backend.count_params(w) for w in modelo.
           non_trainable_weights)
9
10    if acc_t >= 0.95:
11        status = "Nivel Ouro"
12    elif acc_t >= 0.80:
13        status = "Aprovado"
14    else:
15        status = "Reprovado"
16
17    existe = os.path.exists(arquivo_csv)
18    with open(arquivo_csv, mode='a', newline='', encoding='utf-8') as
        f:
19        writer = csv.writer(f)
20        if not existe:
21            writer.writerow([
22                "Timestamp", "Descricao", "Base_Congelada",
23                "Params_Treinaveis", "Params_Congelados",
24                "Epocas", "Loss_Teste", "Acuracia_Teste", "Status"
25            ])
26        writer.writerow([
27            datetime.datetime.now().strftime("%H:%M:%S"),
28            descricao,
29            not base.trainable if hasattr(base, 'trainable') else
                False,
30            t_params, c_params, epocas,
31            f"{loss_t:.4f}", f"{acc_t:.2%}", status
32        ])
33
34    print("\n" + "="*88)
35    print("PAINEL DE EXPERIMENTOS ACUMULADOS (historico_experimentos.
        csv)")
36    print("="*88)
37    with open(arquivo_csv, mode='r', encoding='utf-8') as f:
38        reader = list(csv.reader(f))
39        print(f"{'#':<3} | {'Descricao':<24} | {'Treinaveis':<11} |
            {'Epocas':<6} | {'Loss':<6} | {'Acuracia':<8} | {'Status'}
            ")
40        print("-" * 88)
41        for i, row in enumerate(reader[1:], 1):
42            ts, desc, cong, treina, cong_p, ep, l_val, a_val, st =
                row
43            print(f"{'i':<3} | {desc:<24} | {int(treina):<11,d} | {ep
                :<6} | {l_val:<6} | {a_val:<8} | {st}")
44    print("="*88)

```

3 Parte 3: Experimento 0 — A Dor de Aprender do Zero (10 min)

Vamos testar o que acontece se ignorarmos o Transfer Learning e construirmos uma CNN simples do zero:

```
1 # Configure a descricao do experimento
2 DESCRICAO_EXPERIMENTO = "Cenario 0 (Rede Basica do Zero)"
3
4 def criar_modelo_cenario_0():
5     """CNN simples com 2 camadas convolucionais treinada do zero."""
6     modelo = Sequential([
7         Conv2D(16, (3,3), activation='relu', input_shape=(96, 96, 3))
8         ,
9         MaxPooling2D(2,2),
10        Conv2D(32, (3,3), activation='relu'),
11        MaxPooling2D(2,2),
12        Flatten(),
13        Dense(16, activation='relu'),
14        Dense(1, activation='sigmoid')
15    ])
16    base_ficticia = modelo # Apenas para compatibilidade de chamada
17    modelo.compile(optimizer='adam', loss='binary_crossentropy',
18                  metrics=['accuracy'])
19    return modelo, base_ficticia
20
21 # Execute o treinamento do Cenario 0 e registre no historico
22 modelo, base = criar_modelo_cenario_0()
23 # Treine por 10 epocas e chame registrar_e_exibir_historico()
```

• Teoria: Reflexão Crítica: A Ilusão dos 85%–90%

Ao rodar o Cenário 0, muitos alunos notam que o modelo atinge $\approx 85\%$ a 90% no CIFAR-10. **Ele aprendeu a morfologia dos veículos?** Não! A rede aprendeu um atalho espúrio (*Shortcut Learning*): 90% dos aviões têm céu azul no topo, enquanto carros têm asfalto cinza. Em tarefas sem viés de fundo (ex: Pássaro vs. Avião), a CNN simples cai para 52% (chute aleatório).

4 Parte 4: Experimento A — O Padrão Ouro como Linha de Base (5 min)

Restaure o modelo da Parte 1 (MobileNetV2 congelada + GAP + Dense 16 + Dropout + Sigmoid):

```
1 DESCRICAO_EXPERIMENTO = "Cenario A (Padrao GAP + Congelado)"
2 modelo, base = criar_modelo_transfer() # MobileNetV2 com trainable=
   False
```

```
3 # Treine e registre no historico
```

Observe no terminal o salto de qualidade: acurácia superior a 98%, com perda reduzida para menos de 0.05 e treinando meros 20.513 parâmetros.

5 Parte 5: Experimento B — A Armadilha do Descongelamento (8 min)

Agora teste a tentação ingênua: “E se descongelarmos a rede inteira para aprender tudo?”

```
1 DESCRICAO_EXPERIMENTO = "Cenario B (Descongelado Precoce)"
2
3 modelo, base = criar_modelo_transfer()
4 # O ERRO FATAL: Descongelar o backbone logo no inicio do treino!
5 base.trainable = True
6
7 modelo.compile(optimizer='adam', loss='binary_crossentropy', metrics
8               =['accuracy'])
8 # Treine e registre no historico
```

△ Importante: O Fenômeno do Catastrophic Forgetting

Observe o resultado no CSV: a acurácia cai para $\approx 91\%$ (pior que o Padrão Ouro!) e o tempo de época quadruplica. A cabeça densa inicializada aleatoriamente emite gradientes de erro violentos que descalibram e corrompem os 2,2 milhões de pesos pré-treinados do Google.

6 Parte 6: Experimento C — A Explosão do Flatten (7 min)

Mantenha a base congelada (`trainable = False`), mas substitua `GlobalAveragePooling2D()` por `Flatten()`:

```
1 DESCRICAO_EXPERIMENTO = "Cenario C (Flatten sem GAP)"
2
3 def criar_modelo_cenario_c():
4     base_model = MobileNetV2(weights='imagenet', include_top=False,
5                               input_shape=(96, 96, 3))
6     base_model.trainable = False
7
8     modelo = Sequential([
9         base_model,
10        Flatten(), # <-- SUBSTITUICAO DO GAP
11        Dense(16, activation='relu'),
12        Dropout(0.3),
```

```

12         Dense(1, activation='sigmoid')
13     ])
14     modelo.compile(optimizer='adam', loss='binary_crossentropy',
15                   metrics=['accuracy'])
16     return modelo, base_model
17
17 modelo, base = criar_modelo_cenario_c()
18 # Treine e registre no historico

```

Observe na tabela CSV: os parâmetros treináveis explodem de ≈ 20.000 para mais de **184.000** parâmetros ($9\times$ mais conexões!), tornando o modelo mais pesado e vulnerável a sobreajuste.

7 Parte 7: Síntese e Conclusão Técnica do Aluno (5 min)

Ao término das quatro baterias, o seu terminal exibirá o painel acumulado:

#	Descrição do Experimento	Treináveis	Épocas	Acurácia	Status MLOps
1	Cenário O (Rede Básica do Zero)	553.233	10	$\approx 85.0\%$	Risco de Shortcut
2	Cenário A (Padrão GAP + Congelado)	20.513	8	98.1%	Nível Ouro
3	Cenário B (Descongelado Precoce)	2.278.497	6	$\approx 91.2\%$	Catastrophic Forgetting
4	Cenário C (Flatten sem GAP)	184.336	7	$\approx 95.5\%$	Explosão Dimensional

Vá até o final do script `src/transfer_learning.py` e preencha a variável `CONCLUSAO_ALUNO` respondendo com fundamentação técnica:

- 1 Qual foi o impacto empírico de congelar vs. descongelar a base (Cenário A vs. B)?
- 2 Qual foi a diferença matemática de parâmetros entre usar GAP e Flatten (Cenário A vs. C)?
- 3 Por que o Cenário A é considerado a melhor prática de engenharia de software e IA?

8 Parte 8: Restauração para o Padrão Ouro e Push Final (5 min)

Para garantir que o pipeline de integração contínua do GitLab aprove seu repositório:

- 1 Certifique-se de que o script `src/transfer_learning.py` execute a arquitetura final do **Cenário A (Padrão Ouro)**.
- 2 Execute os testes unitários:

```
1 pytest tests/test_transfer.py -v -s
```

3 Realize o commit e o envio oficial para o GitLab com a telemetria anexada:

```
1 git add src/transfer_learning.py historico_experimentos.csv
  transfer_learning.png
2 git commit -m "Lab 07 finalizado: Duelo de Modelos e Telemetria
  MLOps"
3 git push origin main
```

Conclusão da Sessão 2 & Fechamento do Módulo 1

✓ Takeaways

- Você comprovou empiricamente que o Transfer Learning congelado supera amplamente redes do zero.
- Presenciou o *Catastrophic Forgetting* ao descongelar precocemente o extrator de características.
- Demonstrou por que o GlobalAveragePooling2D é o padrão industrial, economizando 90% dos pesos em relação ao Flatten.
- Praticou governança de MLOps com o arquivo de telemetria `historico_experimentos.csv`.

◇ Informação

Gancho para a Aula 08 (Fechamento do Módulo 1): Seu classificador atinge 98% de acurácia. Mas o que acontece se colocarmos esse modelo em produção? Como servir previsões em tempo real para milhares de usuários? Na **Aula 08**, empacotaremos esse modelo em uma **API REST com FastAPI**, criaremos uma imagem containerizada com **Docker** e aprenderemos a analisar métricas críticas de negócio: **Precision**, **Recall** e a **Matriz de Confusão** no lançamento do seu **Trabalho Prático 1 (TP1)**!