PSI5790 Aprendizado Profundo para Visão Computacional
Primeiro período de 2024 Exercício-programa
Data de entrega: 16/06/2024 (domingo) até 23:59 horas

1. Introdução

Kaggle (www.kaggle.com) é um plataforma de competição de ciência de dados e uma comunidade online de profissionais de aprendizado de máquina. Kaggle permite que os usuários encontrem e publiquem conjuntos de dados, explorem e construam modelos em um ambiente baseado na web, trabalhem com outros cientistas e participem de competições para resolver desafios de ciência de dados [Wikipedia]. Kaggle fornece um ambiente semelhante ao Google Colab com computadores com GPU/TPU.

Em Kaggle, há um conjunto de dados de raio-x de pulmão, COVID-QU-Ex dataset:
https://www.kaggle.com/datasets/anasmohammedtahir/covidqu
Todas as imagens estão no formato .PNG com 256×256 pixels. Esse conjunto contém imagens de acordo com a tabela 1.

Tabela 1: Quantidade de imagens de BD COVID-QU-Ex.

	Train	Val	Test	Total
Covid-19	7658	1903	2395	11956
Non-Covid	7208	1802	2253	11263
Normal	6849	1712	2140	10701
Total	21715	5417	6788	33920

A figura 1 mostra algumas imagens desse BD. O objetivo deste EP é classificar as imagens de validação/teste em Covid19 ou Não-Covid19. Não-Covid19 pode significar paciente sadio (classe Normal) ou paciente com alguma infecção viral/bacteriana que não seja Covid19 (classe Non-Covid).

O desafio é conseguir o maior desempenho em duas condições:

1. Fazendo o treino "do zero", com os pesos da sua rede neural inicializados aleatoriamente (50% da nota).
2. Fazendo "transfer learning" a partir de algum modelo treinado no ImageNet (50% da nota).

Para avaliar o desempenho do seu modelo, utilize ROC-AUC. Quem obtiver ROC-AUC maior, receberá maior nota. Anexe o gráfico do ROC e também informe a taxa de acerto no ponto de "equal error rate".

Nota: Usando rede "inspirada em LeNet", obtive AUC=0,98, EER=5,29% (16 épocas com batch=16). Treino: 10 minutos.
Nota: Fazendo transfer learning e fine tuning com EfficientNetB0 obtive AUC=0.998987, EER=1.115%. 8 épocas com batch=16 tanto para transfer learninge como para fine tuning. Treino: 30 minutos.
Solução privada em: https://colab.research.google.com/drive/1s3d-9q7ndoddBaCIenxoMPGI6hYQU7OK#scrollTo=QpQBnfizfPgS

Nota: Treinando EfficientNetB0 sem peso do zero. AUC=0.9953, EER=3,1%. Tempo de treino 30 minutos.
Solução privada em: https://colab.research.google.com/drive/1pop3kyQjpxX7RaQQmvzs6fdxyko-uAuG#scrollTo=f_5ER3_AS3Oo

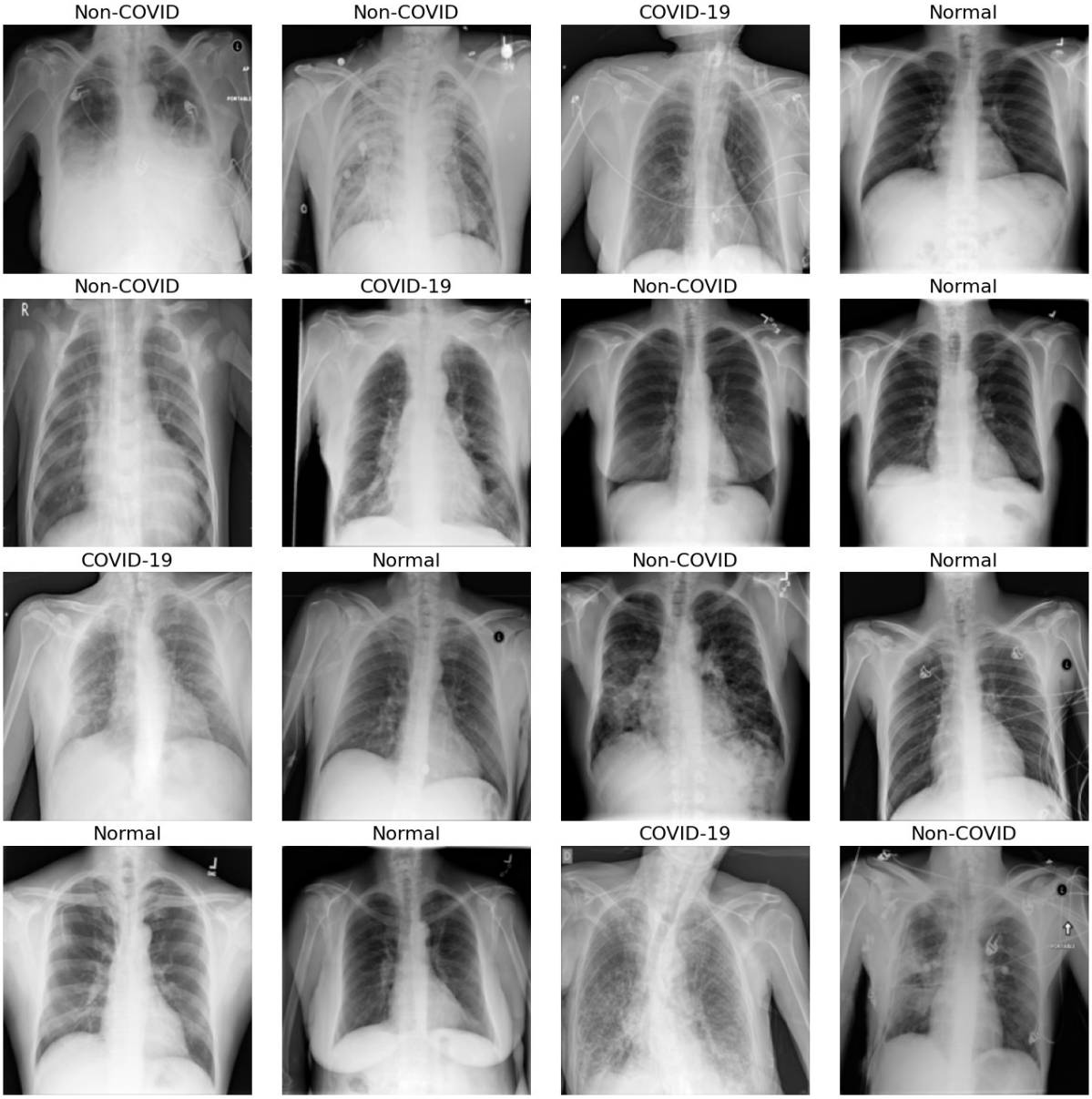

Figura 1: Exemplos de imagens de COVID-QU-Ex.

2. Fazer download do conjunto de imagens da Kaggle

Vou escrever o enunciado supondo que vocês vão usar o ambiente Colab. Se quiserem, podem usar o seu computador local, o ambiente da própria Kaggle ou qualquer outro ambiente. Neste caso, precisam fazer algumas adaptações ao enunciado.

O site abaixo descreve como baixar um conjunto de dados da Kaggle no Colab ou no seu computador local:

https://www.geeksforgeeks.org/how-to-import-kaggle-datasets-directly-into-google-colab/

Resumindo as informações desse site, você primeiro precisa criar uma conta na Kaggle:

https://www.kaggle.com/

Depois de criar a conta, na página principal da Kaggle, clique em "Account" que levará à página "Settings". Nessa página, clique em "Create New Token". Isso baixará um arquivo "kaggle.json", onde você obterá o seu *username* e uma *key*, algo como:

```
{"username":"Joao da Silva","key":"9e9f84b521cd43657418455fce99120b"}
```

Agora, no seu ambiente Colab, copie o conteúdo da célula 1.

```
#https://www.geeksforgeeks.org/how-to-import-kaggle-datasets-directly-into-google-colab/
!pip3 install -q kaggle
!mkdir ~/.kaggle
!echo '{"username":"Joao da Silva","key":"9e9f84b521cd43657418455fce99120b"}' > ~/.kaggle/kaggle.json
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets download anasmohammedtahir/covidqu
!unzip -u covidqu.zip
```

Célula 1: Célula para baixar o BD da Kaggle. Substitua parte amarela pelas suas informações.

Você deve substituir o conteúdo destacado em amarelo pelas suas informações obtidas de "kaggle.json". A execução dessa célula irá baixar o BD no seu diretório default Colab (/content) e o descompactará.

Se quiser executar a célula 1 no computador local, basta executar os comandos da célula no seu terminal, sem o símbolo "!".

3. Leitura do conjunto de imagens

Crie obrigatoriamente a função:

```
def leImagens(wildcards, classes, nl, nc):
```

que irá ler as imagens especificadas na lista "wildcards" e associar a elas classes especificadas na lista "classes". Por exemplo, a célula 2 deverá ler todas as imagens de treino e armazená-las em matrizes numpy *ax* (imagens) e *ay* (com rótulos 1=covid, 0=non-covid, 0=normal); também lerá todas as imagens/rótulos de validação/teste em matrizes *qx* e *qy*.

Nota: As classes non-covid e normal recebem o rótulo "0" pois são "não-covid19".

```
wildcards=['./Lung Segmentation Data/Lung Segmentation Data/Train/COVID-19/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Train/Non-COVID/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Train/Normal/images/*.png']
classes=[1, 0, 0]
ax,ay = leImagens(wildcards, categorias, nl=224, nc=224)
wildcards=['./Lung Segmentation Data/Lung Segmentation Data/Test/COVID-19/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Test/Non-COVID/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Test/Normal/images/*.png']
classes=[1, 0, 0]
qx,qy=leImagens(wildcards,categorias,nl=224,nc=224)
wildcards=['./Lung Segmentation Data/Lung Segmentation Data/Val/COVID-19/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Val/Non-COVID/images/*.png', './Lung Segmentation Data/Lung Segmentation Data/Val/Normal/images/*.png']
classes=[1, 0, 0]
vx,vy=leImagens(wildcards,categorias,nl=224,nc=224)
```

Célula 2: Código para ler todas as imagens de treino, teste e validação em matrizes numpy *ax*, *ay*, *qx*, *qy*, *vx* e *vy*.

Nota: 33.920 imagens 224×224 em níveis de cinza ocupam 33920 × 224 × 224 = 1.701.969.920 bytes (1,7 GB). Como Colab básico fornece espaço RAM de 12 GB no computador (além de 16 GB de memória no GPU Tesla T4), as imagens cabem tranquilamente na memória do computador (se forem armazenadas com um canal e usando elementos do tipo *uint8*). Pode não vai caber se:

1) Armazenar pixels como *float32* ou *float64* ou *int*.
2) Armazenar imagens com 3 bandas (imagens coloridas).
3) Criar várias cópias dos tensores.

Para trabalhar com números em *float32* dentro da rede neural, basta converter somente o lote de imagens atual para *float32* (talvez com 3 bandas) ao alimentar a rede neural ou fazer a conversão de *uint8* para *float32* (talvez com 3 canais) dentro da rede.

Nota: Relatórios e vídeos melhor elaborados receberão notas maiores.

Obs. 1: Este exercício deve ser resolvido individualmente.

Obs. 2: Em princípio, deve usar o mesmo ambiente de programação utilizado em aula (Python, Keras, OpenCV). Se quiser utilizar outro ambiente, converse antes com o professor.

Obs. 3: Entregue um documento PDF de no máximo 4 páginas (relatorio.pdf) descrevendo o funcionamento dos seus programas, os resultados obtidos e as suas conclusões. O envio do relatório é obrigatório (veja o anexo B). O relatório é um documento Word/Latex/Writer convertido para PDF. Um notebook Python anotado não será aceito como relatório.

Obs. 4: Entregue um vídeo de no máximo dois minutos explicando o funcionamento dos seus programas e os resultados obtidos. No vídeo deve aparecer em algum momento o seu rosto e um documento seu. É necessário que o vídeo contenha áudio, pois é muito difícil entender um vídeo sem a explicação falada. O vídeo não pode ter sido acelerado para diminuir a duração, pois dificulta entender a fala. Você pode enviar o vídeo em si (.mkv, .mp4, .avi, etc.) ou um link para o vídeo (youtube, google drive, etc). No segundo caso, deve assegurar que o professor (hae.kim@usp.br) tenha acesso ao seu vídeo.

Obs. 5: Entregue os programas-fontes ou link para notebook Google Colab. Não entregue o conjunto de dados.

Obs. 6: Não se esqueça de escrever o seu nome no relatório, no início do vídeo e no início dos programas-fontes.

Obs. 7: Envie o material através de edisciplinas. Dentro do prazo, você pode substituir o material anterior por um novo.

Anexo: Relatórios dos exercícios programas

O mais importante numa comunicação escrita é que o leitor entenda, sem esforço e inequivocamente, o que o escritor quis dizer. O texto ficar "bonito" é um aspecto secundário. Se uma (pseudo) regra de escrita dificultar o entendimento do leitor, essa regra está indo contra a finalidade primária da comunicação. No site do governo americano [1], há regras denominadas de "plain language" para que comunicações governamentais sejam escritas de forma clara. As ideias por trás dessas regras podem ser usadas em outros domínios, como na escrita científica. Resumo abaixo algumas dessas ideias.

(1) Escreva para a sua audiência. No caso do relatório, a sua audiência será o professor ou o monitor que irá corrigir o seu exercício. Você deve enfocar na informação que o seu leitor quer conhecer. Não precisa escrever informações que são inúteis ou óbvias para o seu leitor.

(2) Organize a informação. Você é livre para organizar o relatório como achar melhor, porém sempre procurando facilitar o entendimento do leitor. Seja breve. Quebre o texto em seções com títulos claros. Use sentenças curtas. Elimine as frases e palavras que podem ser retiradas sem prejudicar o entendimento. Use sentenças em ordem direta (sujeito-verbo-predicado).

(3) Use palavras curtas e simples. Escreva usando menos palavras possível. Use o tempo verbal o mais simples possível. Evite "verbos ocultos" (por exemplo, substitua "precisamos realizar uma revisão das contas da Agência" para "precisamos rever as contas da Agência"; substitua "fiz o pagamento do seu salário" por "paguei o seu salário"). Evite cadeia longa de nomes, substituindo-os por verbos (em vez de "desenvolvimento de procedimento de proteção de segurança de trabalhadores de minas subterrâneas" escreva "desenvolvendo procedimentos para proteger a segurança dos trabalhadores em minas subterrâneas"). Minimize o uso de abreviações (para que o leitor não tenha que decorá-las). Use sempre o mesmo termo para se referir à mesma realidade (pode confundir o leitor se usar termos diferentes para se referir a uma mesma coisa). O relatório não é obra literária, não tem problema repetir várias vezes a mesma palavra.

(4) Use voz ativa. Deixe claro quem fez o quê. Se você utilizar oração com sujeito indeterminado ou na voz passiva, o leitor pode não entender quem foi o responsável (Ex: "Criou-se um novo algoritmo" - Quem criou? Você? Ou algum autor da literatura científica?). O site diz: "Passive voice obscures who is responsible for what and is one of the biggest problems with government writing."

(5) Use exemplos, diagramas, tabelas, figuras e listas. Ajudam bastante o entendimento.

O relatório deve conter pelo menos as seguintes informações:

Identificação

Seu nome, número USP, nome da disciplina, etc.

Breve enunciado do problema

Apesar do enunciado do problema ser conhecido ao professor/monitor, descreva brevemente o problema que está resolvendo. Isto tornará o documento compreensível para alguma pessoa que não tem o enunciado do EP à mão.

Técnica(s) utilizada(s) para resolver o problema

1 https://plainlanguage.gov/guidelines/

Descreva quais técnicas você usou para resolver o problema. Se você mesmo inventou a técnica, descreva a sua ideia, deixando claro que a ideia foi sua. Se você utilizou alguma técnica já conhecida, utilize o nome próprio da técnica (por exemplo, filtragem Gaussiana, algoritmo SIFT, etc.) juntamente com alguma referência bibliográfica onde a técnica está descrita. Use elementos gráficos como imagens intermediárias e diagramas, pois ajudam muito a compreensão. Não "copie-e-cole" código-fonte, a não ser que seja relevante. Use preferencialmente o pseudo-código.

Ambiente de desenvolvimento utilizado

Em qual plataforma você desenvolveu o programa? Como o professor/monitor pode compilar o programa? Você utilizou que bibliotecas?

Operação

Como o professor/monitor pode executar o programa? Que argumentos são necessários para a execução do programa? Há parâmetros que devem ser configurados? Quais arquivos de entrada são necessários? Quais arquivos de saída são gerados?

Resultados Obtidos

Descreva os resultados obtidos. Qual é o tempo de processamento típico? O problema foi resolvido de forma satisfatória?

Referências

Descreva o material externo utilizado, como livros/artigos consultados, websites visitados, etc.