Projeto Aprendizado de Máquina Preditivo

CMC- 15 Inteligência Artificial
Profa. Ana Carolina Lorena
Trabalho em Grupo de Dois ou Três Alunos

1. Objetivo

Exercitar e fixar conhecimentos adquiridos sobre Aprendizado de Máquina (AM) preditivo, por meio do uso e da comparação de técnicas do paradigma de aprendizado supervisionado em um problema prático.

2. Descrição do Trabalho

O trabalho envolverá distinguir imagens de abelhas e formigas (*Hymenoptera dataset* - https://www.kaggle.com/datasets/thedatasith/hymenoptera). Primeiramente, os dados deverão ser estruturados (vetorizados) extraindo atributos com o uso de dois modelos de redes profundas pré-treinadas. Em seguida, os conjuntos obtidos serão processados considerando as etapas de: (i) Análise e pré-processamento dos dados; (ii) Treinamento de técnicas de AM supervisionadas; (iii) Comparação de resultados das técnicas de AM nos diferentes conjuntos produzidos.

Os dados estão em:
https://drive.google.com/drive/folders/1XJiizL57-UyNMC8W_es0NWbcKj_ca8mG?usp=sharing

Eles já estão divididos em três subconjuntos: treinamento, validação e teste. Os arquivos train.txt, validation.txt e test.txt contêm os caminhos das imagens e suas respectivas classes (0 para formigas e 1 para abelhas).

O que deve ser feito:

1) Extrair características das imagens usando **dois modelos de redes pré-treinadas**. Um pode ser o ResNet18, apresentado a seguir, outro deve ser escolhido por vocês a partir de pesquisa de modelos existentes (mencionar qual foi escolhido e por que). Mencione quantos atributos são gerados em cada caso.

A seguir é apresentado um exemplo de código para obter as características (atributos) de imagens do conjunto MNIST usando a rede pré-treinada ResNet18, que pode ser usada como base em sua vetorização (extração de atributos) das imagens do conjunto Hymenoptera:

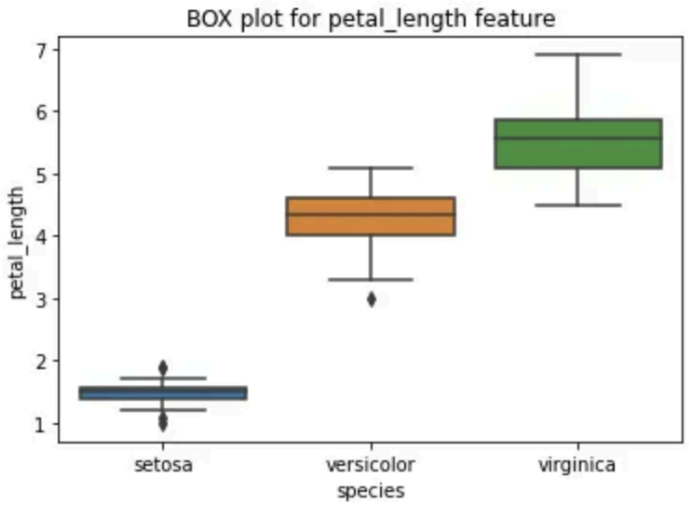
```
import torch
import torch.nn as nn
import torchvision.transforms as transforms
import torchvision.datasets as datasets
from torchvision.models import resnet18, ResNet18_Weights
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
```

```
device = torch.device("cpu")
batch_size = 16

transform = transforms.Compose([
    transforms.Resize(224),
    transforms.Grayscale(num_output_channels=3),
    transforms.ToTensor(),
    transforms.Normalize([0.5]*3, [0.5]*3)
])

mnist_dataset = datasets.MNIST(root='./data', train=True, download=True,
transform=transform)
mnist_loader = DataLoader(mnist_dataset, batch_size=batch_size, shuffle=False)

resnet = resnet18(weights=ResNet18_Weights.DEFAULT).to(device)
resnet.eval()

extracted_features = []

def hook_fn(module, input, output):
    extracted_features.append(output.detach().cpu())

# Registrando o hook na camada 'avgpool'(camada antes da camada de classificação)
hook_handle = resnet.avgpool.register_forward_hook(hook_fn)

all_labels = []
with torch.no_grad():
    for images, labels in mnist_loader:
        images = images.to(device)
        _ = resnet(images)
        all_labels.append(labels)

features = torch.cat(extracted_features, dim=0)
features = features.view(features.size(0), -1)
labels = torch.cat(all_labels)

print("Features extraídas:", features.shape)  # Deve ser [60000, 512]
print("Labels:", labels.shape)  # [60000]

hook_handle.remove()
```

2) A partir da vetorização anterior, vocês terão duas representações do conjunto de dados Hymenoptera. **Para cada uma delas**, faça o seguinte:

   a) Usando a partição de treinamento (train.txt), escolha os cinco atributos mais relevantes para distinguir abelhas e formigas (usando, por exemplo, `SelectKBest` de `sklearn.feature_selection`).

   b) Plote boxplots dos atributos selecionados, distinguindo os valores por classe, tal como no exemplo do conjunto iris a seguir:



   Discuta os resultados obtidos. É possível obter algum *insight*? Se sim, qual?

   c) Agora treine os seguintes modelos de AM sobre os dados train.txt, ajustando hiperparâmetros no conjunto validation.txt: k-vizinhos mais próximos (kNN, variando k entre 1, 3 e 5); árvores de decisão (AD, variando o max_depth entre 5, 10 e None); e Random Forests (RF, variando n_estimators entre 100, 200 e 500). Reportar quais foram os valores de hiperparâmetros escolhidos em cada caso.

   d) Os melhores modelos de kNN, CART e RF (com os melhores hiperparâmetros no conjunto de validação) devem ser então testados no conjunto test.txt e a acurácia preditiva deve ser medida.

3) Faça uma tabela como a seguir reportando os resultados em termos de acurácia no teste:

	kNN	**AD**	**RF**
Atributos ResNet18	<valor acurácia>	<valor acurácia>	<valor acurácia>
Atributos Rede Y	<valor acurácia>	<valor acurácia>	<valor acurácia>

4) Compare, analise e discuta os resultados alcançados. Alguma representação possibilitou alcançar melhores resultados preditivos? E que houve alguma técnica de AM consistentemente melhor nesses experimentos?

3. Material a ser entregue e prazo

Material: Notebook em formato de relatório com as implementações, resultados e discussões
Prazo de Entrega: 23/setembro/2025

Estrutura sugerida para o notebook:

Nomes dos Membros da Equipe
1. Extração de atributos e análise
    a) Rede ResNet18
    b) Rede <X>
2. Treinamento dos modelos
    a) kNN
    b) AD
    c) RF
3. Teste dos modelos
4. Discussões
5. Conclusões: Comentários e sugestões sobre o trabalho (complexidade/facilidade, sugestões, etc.).

Bom Trabalho!

Profa. Ana Carolina Lorena
aclorena@ita.br