

深度学习课程作业拟定（小作业 20 分）

本课程的小作业旨在使同学们熟悉并入门国产的开源深度学习框架（Jittor, Paddle, Mindspore），包括但不限于安装框架，配置深度学习环境，设计实现简单的深度学习模型，按照问题需求对模型进行修改等。目前计划安排小作业三次，难度会逐步增加。第一个小作业时让同学们初步入门国产深度学习框架；第二个小作业是让同学们学会使用国产深度学习框架实现一些常见的判别式神经网络，并在一些常见任务上进行训练与优化；第三个小作业则是让同学们在一些较为复杂的应用场景中探索，使用学习到的深度学习知识解决问题。同学们需要在 DDL 前将小作业相关的文件（包括报告和代码）一起打包上传到 Canvas 上。

注意事项：

- 请使用 Jittor、Mindspore 或 Paddle 中的一种国产深度学习框架完成作业。
- 请在实验过程中遵循良好的代码规范和文档撰写规范。
- 请保持实验报告的简明扼要，清晰地展示实验过程和结论。
- 请尽量在 DDL 以内提交作业。若在超过 DDL 十天以内提交，晚交会扣除一定的分数，每晚交一天扣一分，作业超过十天则不再收。如有特殊情况，请即时与老师和助教沟通。

第一次小作业：国产框架入门（4 分）

目标：

1. 安装国产深度学习框架 Jittor, Mindspore 或 Paddle （三选一，推荐 Jittor）
2. 对国产深度学习框架进行初步的了解与尝试使用，在此基础上配置工作环境
3. 训练一个简单的模型，目标是实现简单的线性回归模型用于拟合线性数据

安装教程：

1. Jittor（计图）：<https://cg.cs.tsinghua.edu.cn/jittor/download/>

2. Mindspore (昇思) : <https://www.mindspore.cn/install>
3. Paddle (飞桨) : <https://www.paddlepaddle.org.cn/install/quick>

任务:

1. 了解线性回归模型的基本要素

线性回归是一种简单而有效的模型，它假设输入特征与输出之间存在线性关系。在我们的例子中，我们假设房价（输出）与房屋面积和房龄（输入特征）之间存在线性关系。模型可以表示为：

$$Price = w_{area} \cdot area + w_{age} \cdot age + b$$

其中， w_{area} 和 w_{age} 是模型的权重参数， b 是偏置参数。

假设我们收集了一系列的真实数据，包括多栋房屋的实际售价和它们对应的面积和房龄。我们将这些数据分为训练集和测试集，用训练集来训练模型，用测试集来评估模型的性能。这里真实的数据由大家自行生成，可以用一个线性函数加随机噪音的方式批量生成数据。

我们需要一个衡量模型预测值与真实值之间差距的函数，即损失函数。在线性回归中，常用的损失函数是平方损失函数，表示为每个样本误差的平方和的平均值：

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (\hat{y}^i - y^i)^2$$

其中 \hat{y}^i 是模型对第 i 个样本的预测， y^i 是第 i 个样本的真实标签， n 是样本数量。

为了最小化损失函数，我们使用优化算法来调整模型的参数。在线性回归中，常用的优化算法是随机梯度下降（SGD）。它的基本思想是沿着损失函数的负梯度方向更新参数，以减小损失。具体来说，更新参数的过程如下：

$$(w, b) \leftarrow (w, b) - \eta \nabla_{(w, b)} L(w, b)$$

其中 η 是学习率, $\eta \nabla_{(w,b)} L(w, b)$ 是损失函数关于模型参数 w 和 b 的梯度。

2. 利用国产深度学习框架实现线性回归模型, 可以参考 Jittor, Mindspore 或 Paddle 的官方文档用于上手
 - 1) 随机初始化模型参数 w 和 b
 - 2) 在训练数据上进行迭代
 - 3) 重复步骤 2) 直到收敛, 在迭代更新参数的过程中, 通过监控损失函数的变化来判断模型是否收敛。
3. 将你本次作业的实现经过, 线性模型搭建, 实验结果与分析, 写成两页 pdf 报告的形式, 在 DDL 前与代码一起上传到 Canvas 上。 (报告限制 pdf 两页)

第二次小作业：判别式神经网络实践 (8 分)

判别式神经网络是一类常见的神经网络模型, 其主要任务是学习输入数据与其对应标签之间的映射关系, 从而实现对不同类别的数据进行分类。判别式神经网络通过学习从输入到输出的映射, 以便最大化或最小化某个目标函数, 从而实现对数据的判别。这类神经网络包括多种结构, 如感知机、多层感知机、卷积神经网络 (CNN)、循环神经网络 (RNN) 等。判别式神经网络在图像处理中的应用涵盖了图像分类、目标检测、语义分割、图像生成、图像超分辨率等多个方面, 为图像处理任务的自动化和高效处理提供了强大的工具。最典型的应用之一就是图像分类, 即将输入的图像分为不同的类别。卷积神经网络 (CNN) 是最常用的判别式神经网络, 通过一系列卷积层和池化层来提取图像特征, 并通过全连接层进行分类。在本次小作业中, 我们希望同学们可以学习使用 CNN 和 RNN 这两种常见的判别式神经网络, 并根据具体的图像分类场景进行对应的优化与改进。

目标:

1. 下载图像分类 [CIFAR-10](#) 数据集
2. 搭建国产深度学习训练框架 Jittor, Mindspore 或 Paddle (三选一, 推荐 Jittor), 并设计判别式神经网络模型 (CNN, RNN)。

3. 在给定的要求下改进模型

任务：

在国产深度学习框架 Jittor, Mindspore 或 Paddle （三选一，推荐 Jittor）中，利用神经网络解决 CIFAR-10 图像分类这一任务。我们鼓励同学们在本次作业中进一步了解国产深度学习框架，通过对 CNN 和 RNN 这两种常见的判别式生成网络的搭建与实现，达到可以熟练使用国产深度学习框架解决具体的任务问题的能力。

在图像分类这一经典任务中，卷积神经网络是较为常用的做法，卷积神经网络（CNN）是一种专门用于处理图像数据的深度学习模型，常用于解决图像分类问题。通过一系列的卷积层、池化层和全连接层构成的网络结构，CNN 可以自动学习并提取图像中的特征，从而对图像进行准确的分类。在使用 CNN 解决图像分类问题时，首先需要收集并预处理图像数据集，然后构建 CNN 模型并进行训练。训练过程中，CNN 通过反向传播算法不断调整模型参数，使得模型的预测结果与真实标签尽可能接近。

除了 CNN 以外，循环神经网络（RNN）也可以用于图像处理。用 RNN 解决分类问题的思路有很多，最直观的方法是将图片（28*28）视为 28 个一维数据组成的，通过 RNN 网络学习这 28 行数据，来输出一个分类，并与 label 进行 loss 计算并更新神经网络。同学们如果有自己的设计也欢迎尝试。大家在完成作业的过程中也可以思考一下 CNN 与 RNN 的差异与特性。

要求：

1. 构建卷积神经网络（CNN）在 CIFAR-10 数据集上训练测试，并汇报结果。
2. 构建循环神经网络（RNN）在 CIFAR-10 数据集上训练测试，并汇报结果。
3. 将训练集进行划分，具体做法为所有分类为（0, 1, 2, 3, 4）的图像仅保留 10%，剩余部分不变。
4. 在新的训练集上训练神经网络，这时结果相较于原先的结果会有下降，那么，提出可能的解决方案，尝试改进你的模型。
5. 将你对神经网络（CNN,RNN）的设计与实现，在新训练集上的分析，模型设计改进，连同实验结果一起，写成四页 pdf 报告的形式，在 DDL 前与代码一起上传到 Canvas 上。（报告限制 pdf 四页）

训练集切分方式示例：

```
if i % 10 != 0:

    mask = (labels >= 5)

else:

    mask = (labels >= 0)

labels = labels[mask]

images = images[mask]
```

第三次小作业：图像复原任务（8 分）

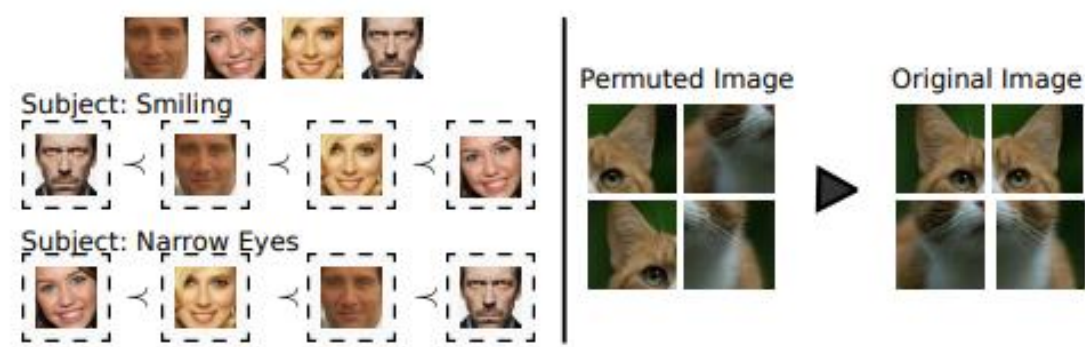
本次小作业聚焦于深度学习领域中的“视觉”与“图”两大重点研究方向，探索图像复原任务在图像处理领域的应用。图像复原任务是指从被打乱的子图片中恢复原始图片，及其被打乱的排列顺序。例如，对于一张图片，将其切分为若干张子图片并打乱顺序，然后设计一个模型来还原这些子图片的正确排列位置。这一任务在图像处理和计算机视觉领域具有重要的应用价值，能够帮助我们理解图像的结构和特征。

目标：

本次小作业的目标是利用 CIFAR-10 数据集作为实验数据集，探索图像复原任务在图像处理领域的应用。具体而言，我们将对 CIFAR-10 数据集中的图像进行切分，然后对切分后的子图片进行打乱顺序。最终目标是设计一个深度学习模型，能够根据乱序的子图片还原出它们的正确排列位置。通过这一任务，我们将深入了解深度学习在图像处理中的应用，并探索图像复原任务的解决方案。

参考论文：

参考论文为《DeepPermNet: Visual Permutation Learning》，该论文提出了一种视觉排列学习模型，用于图像复原任务。论文链接：[DeepPermNet: Visual Permutation Learning](#)



任务步骤：（这里只是推荐的实现步骤，如果有不同的设计也可以自行实现，最终我们将依据 pdf 报告中展示的完成质量综合考虑进行打分）

1. 修改作业二中的图片分类的代码框架，将输入图片分割为 N 张子图片，打乱它们的顺序作为输入，并将他们的正确排列作为 label，输出格式采用排列阵 P ，其中第 i 行第 j 列为 1 则代表第 i 个子图片应该排在第 j 个位置。注意，每行与每列均最多有一个元素为 1。具体切分方式可以自行定义，分割数量不少于 4 份。
2. 设计一个针对图片拼接的神经网络模型，可以参考 DeepPermNet 的设计，即将每一个子图片输入一个神经网络（CNN 或 RNN）进行特征提取，将得到的特征进行拼接后通过若干全连接网络得到一个 $N \times N$ 的矩阵，使用 Sinkhorn 算法使其变成双随机矩阵 Q （即每行每列的和固定为 1）作为神经网络的预测值。最后将预测值 Q （双随机矩阵）与真值 P （排列阵）作为损失函数的输入，计算 Loss 后用于更新整个神经网络。
3. 神经网络可以套作业二中的代码，也可以自行设计新的网络。Sinkhorn 算法推荐使用 pygtools 中的 [Sinkhorn](#)，这是我们实验室自行研发的开源平台。

本作业侧重探索性质，对精度不做要求，没有固定的解决方案，言之有理即可。最终要求提交一份四页 pdf 报告，在 DDL 前与代码一起上传到 Canvas 上。（报告限制 pdf 四页）