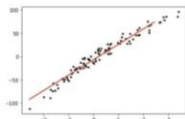


深度学习小作业（40 分）

小作业：配置环境+调用软件包

- 回归问题



- 分类问题



- 拼图问题



本课程的小作业旨在是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，难度逐步增加。

第一次小作业：回归问题（10 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. 配置工作环境
3. 训练一个简单的模型，目标是拟合一个多项式函数

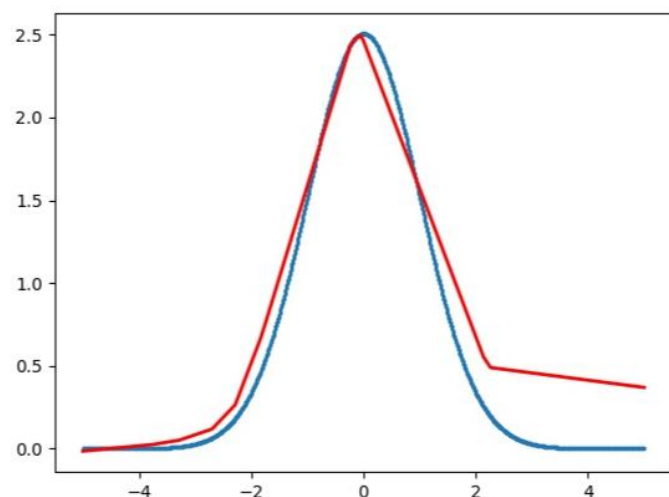
安装教程：

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：

自行给定一个函数 $f(x)$ (例如正态分布函数，三次函数)，自行确定定义域 x 的范围，在这一区间内随机采样 1000 个点，其中 800 个数据点作为训练集，200 个数据点作为测试集。

自行设计一个深度学习模型（例如多层感知机）尝试拟合 $f(x)$ ，并将结果作图以供展示。下面展示了一个函数拟合的例子：



提交作业时上传全部的 python 代码，与拟合的效果图。在 DDL 前发送至 canvas 上。

第二次小作业：分类问题（10 分）

目标：

1. 下载并学习使用 CIFAR-10 数据集
2. 搭建深度学习训练框架 Jittor, Mindspore 或 Paddle（三选一，推荐 Jittor），并设计深度学习模型
3. 在给定的要求下改进模型

CIFAR-10 数据集下载：<https://www.cs.toronto.edu/~kriz/cifar.html>

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。我们不会对得到的精度做出要求，可以选择任意网络进行实现。实现初步的图像分类任务后，我们额外增加了一个关于数据分布不均匀情况下的任务，具体要求如下：

1. 构建神经网络（具体网络不限，如 CNN 等）在 CIFAR-10 数据集上训练测试（自

行划分训练集测试集), 并汇报结果。

2. 将训练集进行划分, 具体做法为所有类别为 (0, 1, 2, 3, 4) 的图像仅保留 10%, 剩余部分不变。
3. 在新的训练集上训练神经网络, 这时结果相较于原先会有下降, 那么, 提出可能的解决方案, 尝试改进你的模型。
4. 将你在新的训练集上的分析, 模型设计改进, 连同实验结果一起, 写成 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)

训练集切分方式示例 :

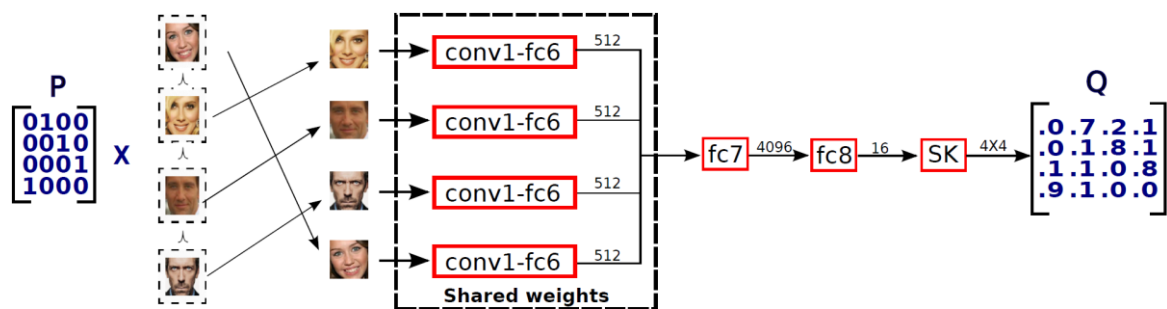
```
if i % 10 != 0:
    mask = (labels >= 5)
else:
    mask = (labels >= 0)
labels = labels[mask]
images = images[mask]
```

第三次小作业 : 拼图问题 (20 分)

目标 : 在 CIFAR-10 数据集上进行进一步研究, 在图像分类任务以外研究图片拼接这一任务, 即, 对于一张图片, 我们对其进行一定的切分, 例如水平竖直分为四份 (具体切分方式可以自行定义), 接着, 我们将这几张子图片打乱顺序作为输入, 目标就是设计一个神经网络可以根据这些乱序的子图片还原出他们本来的排列位置。

相关参考论文 : DeepPermNet: Visual Permutation Learning

<https://arxiv.org/abs/1704.02729>



任务：(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 张子图片后打乱作为输入，并将他们的正确排列作为输出，输出格式采用排列阵 P ，第 i 行第 j 列若为 1 则代表图片 i 应该排在位置 j ，每行每列最多有一个元素为 1。具体切分方式可以自行定义，但需要有相应的可视化效果说明自己的切分设计。
2. 设计一套针对图片拼接的神经网络，可以参考 DeepPermNet 的设计，即对每一个子图片过一个 CNN，将得到的特征进行拼接后过若干全连接网络后得到一个 $N \times N$ 的矩阵，通过 Sinkhorn 算法使其变成双随机矩阵 Q （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 Loss 后用于更新神经网络。
3. CNN 和 FC 网络等可以套作业二中的代码，也可以自行设计新的网络。Sinkhorn 算法我们推荐使用 pygtools 中的 sinkhorn（后面大作业可能也需要用到 pygtools，可以提前熟悉一下）。
4. 在实现了图片拼接问题的网络以后，尝试能否把图片拼接作为一种预训练方式来更好的辅助作业二中的图片分类任务。
5. 本作业同样对精度不做要求，是一个偏探索性质的开放性题目，没有固定的做法，言之有理即可。最终要求提交一份 pdf 报告，在 DDL 前与代码一起上传到 Canvas 上。（报告限制 pdf 三页）