

实验一

实验要求

使用 pytorch 或者 tensorflow 手写一个前馈神经网络，用于近似函数：

$$y = \sin(x) + \cos(x) + \sin(x) \cos(x), \quad x \in [0, 2\pi)$$

并研究网络深度、学习率、网络宽度、激活函数对模型性能的影响。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。若你需要使用 GPU，可能还需安装 CUDA 驱动。本次实验仅利用 CPU 也可以完成，但仍强烈推荐大家安装 GPU 版本，以满足后续实验需求。
2. **数据生成**：本次实验的数据集仅需使用程序自动生成，即在 $[0, 2\pi)$ 范围内随机 sample 样本作为 x 值，并计算 $y = \sin(x) + \cos(x) + \sin(x) \cos(x)$ 作为 y 值。要求生成三个**互不相交**的数据集分别作为**训练集、验证集、测试集**。训练只能在训练集上完成，实验调参只能在验证集上完成。
3. **模型搭建**：采用 pytorch 或 tensorflow 所封装的 module 编写模型，例如 `torch.nn.Linear()`, `torch.nn.ReLU()` 等，无需手动完成底层 forward、backward 过程。
4. **模型训练**：将生成的训练集输入搭建好的模型进行前向的 loss 计算和反向的梯度传播，从而训练模型，同时也建议使用网络框架封装的 optimizer 完成参数更新过程。训练过程中记录模型在训练集和验证集上的损失，并绘图可视化。
5. **调参分析**：将训练好的模型在验证集上进行测试，以 Mean Square Error(MSE) 作为网络性能指标。然后，对网络深度、学习率、网络宽度、激活函数等模型超参数进行调整，再重新训练、测试，并分析对模型性能的影响。
6. **测试性能**：选择你认为最合适的（例如，在验证集上表现最好的）一组超参数，重新训练模型，并在测试集上测试（注意，这理应是你的实验中**唯一**一次在测试集上的测试），并记录测试的结果（MSE）。

实验提交

本次实验截止日期为 **4月2日 23:59:59**，需提交代码源文件及实验报告到邮箱：proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 **【学号】- 姓名 - exp1.zip**

2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。
3. 代码中提供一个可以直接运行的并输出结果的 main.py，结果包括训练集损失、验证集损失随 epoch 改变的曲线（保存下来）和测试集的 MSE。
4. 代码中提供一个描述所有需依赖包的 requirements.txt，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对超参数的**实验分析**，最优超参数下的训练集、验证集**损失曲线**以及测试集上的**实验结果**。

参考资料

[Learning PyTorch with Examples — PyTorch Tutorials 2.0.0+cu117 documentation](#)

对 Pytorch 不熟悉的同学，可以看一下 Pytorch 官方的一个样例。这个样例使用三次函数拟合 $\sin(x)$ 函数，我们的实验与这个有点类似。

[GitHub - hehaha68/USTC_2022Spring_Introduction-to-Deep-Learning: 中科大2022春《深度学习导论》课程资源](#)

这是往年深度学习导论课程实验代码，同学们可以参考一下。（这里面没有实验一的代码）

实验二

实验要求

使用 pytorch 或者 tensorflow 实现卷积神经网络，在 ImageNet 数据集上进行图片分类。研究 dropout, normalization, learning rate decay, residual connection, network depth 等超参数对分类性能的影响。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。（如果前面实验已经安装过，则这个可以跳过）
2. **数据集**：这次实验使用 Tiny-Imagenet-200 数据集，包含 200 个类，每个类有 500 张训练图像，50 张验证图像和 50 张测试图像。由于测试图像没有标签，因此使用数据集中的验证集当作测试集，并从训练集中手动划分新的训练集和测试集。下载链接：
<http://cs231n.stanford.edu/tiny-imagenet-200.zip> 或 <https://rec.ustc.edu.cn/share/4bfe7ba0-cfab-11ed-9877-5f868bfa864a>
3. **模型搭建**：采用 pytorch 或 tensorflow 所封装的 module 编写模型，例如 `torch.nn.Linear()`, `torch.nn.ReLU()` 等，无需手动完成底层 forward、backward 过程。
4. **模型训练**：将生成的训练集输入搭建好的模型进行前向的 loss 计算和反向的梯度传播，从而训练模型，同时也建议使用网络框架封装的 optimizer 完成参数更新过程。训练过程中记录模型在训练集和验证集上的损失，并绘图可视化。
5. **调参分析**：将训练好的模型在验证集上进行测试，以 **Top 1 Accuracy(ACC)** 作为网络性能指标。然后，对 dropout, normalization, learning rate decay, residual connection, network depth 进行调整，再重新训练、测试，并分析对模型性能的影响。
6. **测试性能**：选择你认为最合适的（例如，在验证集上表现最好的）一组超参数，重新训练模型，并在测试集上测试（注意，这理应是你的实验中**唯一**一次在测试集上的测试），并记录测试的结果（ACC）。

实验提交

本次实验截止日期为 **4 月 16 日 23:59:59**，需提交代码源文件及实验报告到邮箱：
proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 **【学号】 - 姓名 - exp2.zip**
2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。

3. 代码中提供一个可以直接运行的并输出结果的 `main.py`，**结果包括训练集损失、验证集损失随 epoch 改变的曲线（保存下来）和测试集的 ACC。**
4. 代码中提供一个描述所有需依赖包的 `requirements.txt`，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对超参数的**实验分析**，最优超参数下的训练集、验证集**损失曲线**以及测试集上的**实验结果**。

参考资料

往届同学的实验代码和报告：https://github.com/hehaha68/USTC_2022Spring_Introduction-to-Deep-Learning

提供的 Lab2_demo.ipynb

实验三

实验要求

使用 pytorch 或者 tensorflow 的相关神经网络库编写基于 BERT 的预训练语言模型，利用少量的训练数据，微调模型用于文本情感分类。并和直接用 RNN/Transformer 训练的文本分类器进行对比，研究训练数据量变化对性能带来的影响。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。这个实验还需要安装 transformers 库，方便调用预训练模型。（如果前面实验已经安装过，则这个可以跳过）
2. **数据准备**：本次实验统一使用指定的 IMDB 公开数据集 "Large Movie Review Dataset"。该数据集分别包含 25,000 条电影评论作为训练集和测试集。任务为二分类任务。数据下载地址为：[Sentiment Analysis \(stanford.edu\)](#)
3. **数据预处理**：你需要通过 pytorch 或 tensorflow 所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如 torch.utils.data.Dataset 等。这个数据集是非常常见的公开数据集，你可以参考一些公开代码片段。
4. **语言模型**：搭建 BERT 模型并加载大语料库上预训练的模型参数，推荐的预训练参数来源为 [BERT \(huggingface.co\)](#)；RNN 模型；Transformer 模型。
5. **情感分类**：情感分类模型包含一个语言模型和一个分类器（MLP）。首先，将一个句子中的每个单词对应的词向量输入语言模型，得到句子的向量表征。然后将句向量作为分类器的输入，输出二元分类预测，然后进行 loss 计算和反向梯度传播训练，这里的 loss 是分类 loss，如交叉熵 loss。
6. **研究训练数据量**：对于 IMDB 中的 train 数据，采用不同比例的数据（如25%，50%，75%，100%）用于划分训练集和验证集以训练/微调情感分类模型。剩下的train中数据不使用。然后使用训练好的模型测试在 test 数据上的表现。

(所以在整个实验中需要调整的参数就是不同的**语言模型**和**训练数据的比例**)

实验提交

本次实验截止日期为 **5月23日 23:59:59**，需提交代码源文件及实验报告到邮箱：proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 学号- 姓名 - exp3.zip

2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。
3. 代码中提供一个可以直接运行的并输出结果的 main.py，**结果包括训练集损失、验证集损失随 epoch 改变的曲线（保存下来）和测试集的评价指标。**
4. 代码中提供一个描述所有需依赖包的 requirements.txt，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对训练数据量和不同语言模型的**实验分析**。

参考资料

往届同学的实验代码和报告：https://github.com/hehaha68/USTC_2022Spring_Introduction-to-Deep-Learning

提供的 Lab4_demo.ipynb

实验数据下载链接：<https://rec.ustc.edu.cn/share/923bdc50-ee3b-11ed-9e6b-95f1dcc044f2>

实验四

实验要求

使用pytorch 或者 tensorflow 的相关神经网络库编写图卷积神经网络模型 GCN，并在相应的图结构数据集上完成节点分类和链路预测任务，最后分析自环、层数、DropEdge、PairNorm、激活函数等因素对模型的分类和预测性能的影响。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。这个实验还需要安装 torch_geometric。（如果前面实验已经安装过，则这个可以跳过）
2. **数据准备**：本次实验使用的数据包含三个常用的图结构数据集：Cora、Citeseer、PPI。下面分别进行介绍。
 - **Cora**：该数据集是由 2708 篇机器学习论文作为节点、论文间引用关系作为有向边构成的图数据。具体的数据描述见<https://relational.fit.cvut.cz/dataset/CORA>。数据集下载链接 <https://linqsdata.soe.ucsc.edu/public/lbc/cora.tgz>。另外，提供一个数据处理范例链接<https://graphsandnetworks.com/thecoradataset/>。请同学们仔细阅读相关材料，了解文件的具体结构和数据格式。
 - **Citeseer**：该数据集是由3312篇论文及相互引用构成的图数据集。数据集下载链接 <https://linqsdata.soe.ucsc.edu/public/lbc/citeseer.tgz>。文件的结构和数据格式与Cora类似。
 - **PPI**：PPI 网络是蛋白质相互作用（Protein Protein Interaction,PPI）网络的简称数描述可参考链接 <https://blog.csdn.net/ziqingnian/article/details/112979175>。数据集下载链接 <http://snap.stanford.edu/graphsage/ppi.zip>。
3. **数据预处理**：你需要通过pytorch 或 tensorflow所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如 torch.utils.data.Dataset 等。由于这三个数据集是非常常见的公开数据集，你可以参考一些公开代码片段，尤其是 github 上典型的GCN 教程级实现或相关论文的源码。
4. **图网络模型**：搭建GCN模型，这一步可以参考网络上公开的源码。
5. **节点分类**：在三个数据集上按照节点分类任务的需求自行划分训练集、验证集、测试集，并用搭建好的GCN 模型进行节点分类。
6. **链路预测**：在三个数据集上按照链路预测任务的需求自行划分训练集、验证集、测试集，并用搭建好的GCN 模型进行链路预测。
7. **测试性能**：选择你认为最合适的（例如，在验证集上表现最好的）一组超参数，重新训练模型，并在测试集上测试（注意，这理应是你的实验中 唯一一次在测试集上的测试），并记录测试的结果。

实验提交

本次实验截止日期为 **6月6日 23:59:59**，需提交代码源文件及实验报告到邮箱：
proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 学号- 姓名 - exp4.zip
2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。
3. 代码中提供一个可以直接运行的并输出结果的 main.py，**结果包括训练集损失、验证集损失随 epoch 改变的曲线（保存下来）和测试集的评价指标。**
4. 代码中提供一个描述所有需依赖包的 requirements.txt，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对超参数的**实验分析**，最优超参数下的训练集、验证集**损失曲线**以及测试集上的**实验结果**。

参考资料

往届同学的实验代码和报告：https://github.com/hehaha68/USTC_2022Spring_Introduction-to-Deep-Learning

提供的 Lab4_demo.ipynb

实验数据下载链接：<https://rec.ustc.edu.cn/share/3bab12a0-ee13-11ed-b34a-1d166b75eb33>

实验五(style transfer)

实验要求

使用pytorch 或者 tensorflow 的相关神经网络库复现一篇风格迁移 (style transfer) 论文[Image Style Transfer Using Convolutional Neural Networks](#)的效果。

请自行阅读论文和查阅网上资料学习其基本原理，然后动手实践复现论文的效果，如果能在充分理解论文基本原理的基础上，对其进行一些额外的探索、分析或改进，能够加分。

报告主要记录论文基本原理、实践过程、关键代码、实验结果、分析和总结等。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。并且依据参考的内容安装相应的库。
2. **数据准备**：自行下载相关图片数据和预训练好的特征提取网络模型。
3. **数据预处理**：你需要通过pytorch 或 tensorflow所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如 torch.utils.data.Dataset 等。
4. **模型训练**：训练风格迁移的目标图像 (target image)。注意无需训练网络参数。
5. **结果展示**：充分实验，展示和分析结果。

实验提交

本次实验截止日期为 **7月1日 23:59:59**，需提交代码源文件及实验报告到邮箱：proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 **【学号】- 姓名 - exp5.zip**
2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。
3. 代码中提供一个可以直接运行的并输出结果的 main.py。
4. 代码中提供一个描述所有需依赖包的 requirements.txt，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**基本原理、实验过程、关键代码、实验结果、实验分析**。

参考资料

参考论文：[Image Style Transfer Using Convolutional Neural Networks](#)

pytorch开源代码库参考：<https://github.com/enomotokeji/pytorch-Neural-Style-Transfer>

tensorflow 开源代码库参考：<https://github.com/MingtaoGuo/Style-transfer-with-neural-algorithm>

实验五(Text2Image)

实验要求

使用 pytorch 或者 tensorflow 的相关神经网络库编写 text2image 代码，并在相应的数据集上实验。本次 text2image 实验主要使用两种方法，一种是基于 Conditional GAN 的方法，另一种是基于 Conditional Diffusion Model 的方法。基于 Conditional GAN 的方法需要自己训练，基于 Conditional Diffusion Model 的方法则可以使用训练好的模型。基于 Conditional GAN 的方法可以参考论文 [RAT](#) 和其开源代码库 [RAT-GAN]([GitHub - senmaoy/RAT-GAN: A conditional GAN for text-to-image](#))。基于 Conditional Diffusion Model 的方法可以参考 [StableDiffusion]([Stability-AI/stablediffusion: High-Resolution Image Synthesis with Latent Diffusion Models \(github.com\)](#))，可以直接调用其训练好的模型。本次实验以动手实践为主，报告主要记录实践过程和关键代码以及实验结果。

实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。并且依据参考的内容安装相应的库。
2. **数据准备**：本次实验使用的数据包含一个常用的简单的 text2image 数据集
 - **Oxford-102**：该数据集包含8189张102种花的图片。每张图片包括10种文字描述。可以从 [text_flower](#) 中下载文字描述，从 [flower](#) 中下载图片。使用其中82种类别当作训练集，剩下 20 种类别当作测试集。
3. **数据预处理**：你需要通过pytorch 或 tensorflow所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如 torch.utils.data.Dataset 等。
4. **模型训练**：基于 Conditional GAN 的模型需要自己训练，因此需要绘制 loss 随着训练进行的图像。
5. **模型测试**：对于测试集中的数据。使用自己训练好的模型(CGAN)生成图片，并展示部分实验结果（包括原始图片，原始文字描述以及生成后的图片）。同时还需使用开源的 StableDiffusion 模型生成图片，并展示实验结果。两个模型使用相同的文本描述。

实验提交

本次实验截止日期为 **7月1日 23:59:59**，需提交代码源文件及实验报告到邮箱：proton00@mail.ustc.edu.cn，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 学号- 姓名 - exp5.zip
2. 代码仅包含 .py 文件，如果有多个文件，放在 src/ 文件夹内。对于自己训练好的模型权重，可以上传到睿课网的云盘，并在实验报告中提供共享的地址。

3. 代码中提供一个可以直接运行的 `main.py`，可以根据输入的文本，生成文本所描述的图片。
4. 代码中提供一个描述所有需依赖包的 `requirements.txt`，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括**实验过程**和**关键代码**展示，训练集损失图以及生成的实验结果展示（挑出较好的结果）。

参考资料

实验数据下载：

[text_flower](#)

[flower](#)

参考论文：

[RAT](#)

[2112.10752.pdf \(arxiv.org\)](#)

参考代码：

[RAT-GAN]([GitHub - senmaoy/RAT-GAN: A conditional GAN for text-to-image](#))

[Stability-AI/stablediffusion: High-Resolution Image Synthesis with Latent Diffusion Models \(github.com\)](#)

[Stable unCLIP \(huggingface.co\)](#)

深度学习导论大作业提交要求

在 8月 15 日 23:59:59 前把报告和代码打包提交到邮箱 yingheustc@mail.ustc.edu.cn 中。

报告格式为 pdf 文件，内容应包括：1) 队伍名称，队员名字、学号、分工及贡献。2) 所选比赛的**链接**、简要问题介绍、问题分析。3) 实验所用方案技术，实验结果及分析。4) 最后分数和排名（需注明总参赛队伍数目）。

代码仅包括源代码。