

PyTorch 框架班汇总作业（第二期）

笔记整理人：天国之影-Relph（2019 年 11 月 29 日）

说明

1. 本课程作业的所有代码都要基于 Python3，在 Jupyter Notebook 上完成。
2. Pytorch 中文文档地址：

<https://github.com/zergtant/pytorch-handbook>

3. 训练营代码下载（代码资料和训练营进度同步更新）：

链接：<https://pan.baidu.com/s/1Fy4rsoKEO3vBloVnfnMBJg>

提取码：dldl

我的作业 GitHub 地址：<https://github.com/Relph1119/Pytorch-Camp>

- （1）src 文件夹下面为每周的作业内容，用于记录我的作业完成情况，所有 ipynb 文件均带注释。
- （2）PyTorch_Tutorial 文件夹中为余霆嵩老师的课程代码。

1 第 1 周

1.1 PyTorch 简介及环境配置，PyTorch 基础数据结构——张量

学习时长：10/10—10/13

任务简介：安装 PyTorch 依赖环境学习 PyTorch 中的数据结构 Tensor 和 Variable
详细说明：

（1）本节第一部分介绍 pytorch 及 pytorch 作为深度学习框架的优势，并且基于 Windows 系统进行安装 Pycharm、Anaconda、Cuda、cudnn 和 pytorch，环境配置好后会进行 demo 演示，测试 pytorch 可以正常使用。

（2）本节第二部分介绍 pytorch 中的数据结构——Tensor，Tensor 是 PyTorch 中

最基础的概念，其参与了整个运算过程，因此本节将介绍张量的概念和属性，如 `data`, `device`, `dtype` 等，并介绍 `tensor` 的基本创建方法，如直接创建、依数值创建和依概率分布创建等。

视频：

- (1) `pytorch` 简介与安装
- (2) 张量简介与创建

作业名称（详解）：

- (1) 安装 Anaconda、Pycharm、CUDA+CuDNN（可选），虚拟环境，`pytorch`，并实现 `hello pytorch` 查看 `pytorch` 的版本
- (2) 张量与矩阵、向量、标量的关系是怎么样的？
- (3) `Variable`“赋予”张量什么功能？
- (4) 采用 `torch.from_numpy` 创建张量，并打印查看 `ndarray` 和张量数据的地址；
- (5) 实现 `torch.normal()` 创建张量的四种模式。

作业提交形式： PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）代码实现题将程序中 `print` 的结果复制上来即可。

打卡截止时间： 10/13

1.2 张量操作与线性回归；计算图与动态图机制

学习时长： 10/14—10/15

任务简介： 学习张量的基本操作与线性回归模型的实现；学习计算图概念，理解动态图和静态图的差异

详细说明：

(1) 本节将介绍张量的基本操作，如张量拼接切分、索引和变换，同时学习张量的数学运算，并基于所学习的知识，实现线性回归模型的训练，以加深知识点的认识。

(2) 本节第二部分介绍 `pytorch` 最大的特性——动态图机制，动态图机制是

PyTorch 与 TensorFlow 最大的区别，该部分首先介绍计算图的概念，并通过演示动态图与静态图的搭建过程来理解动态图与静态图的差异。

视频：

- (1) 张量操作与线性回归
- (2) 计算图与动态图机制

作业名称（详解）：

- (1) 调整线性回归模型停止条件以及 $y = 2 * x + (5 + \text{torch.randn}(20, 1))$ 中的斜率，训练一个线性回归模型；
- (2) 计算图的两个主要概念是什么？
- (3) 动态图与静态图的区别是什么？

作业提交形式： PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

打卡截止时间： 10/16

1.3 autograd 与逻辑回归

学习时长： 10/17—10/18

任务简介： 学习 pytorch 的自动求导系统——autograd；通过 autograd 训练逻辑回归模型

详细说明：

本节对 pytorch 的自动求导系统中常用的两个方法 `torch.autograd.backward` 和 `torch.autograd.grad` 进行介绍，并演示一阶导数，二阶导数的求导过程；理解了自动求导系统，以及数据载体——张量，前向传播构建计算图，计算图求取梯度过程，这些知识之后，就可以开始正式训练机器学习模型。这里通过演示逻辑回归模型的训练，学习机器学习回归模型的五大模块：数据、模型、损失函数、优化器和迭代训练过程。这五大模块将是后面学习的主线。

视频：

autograd 与逻辑回归

作业名称（详解）：

（1）逻辑回归模型为什么可以进行二分类？

（2）采用代码实现逻辑回归模型的训练，并尝试调整数据生成中的 `mean_value`，将 `mean_value` 设置为更小的值，例如 1，或者更大的值，例如 5，会出现什么情况？再尝试仅调整 `bias`，将 `bias` 调为更大或者负数，模型训练过程是怎么样？

作业提交形式： PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

（1）文字要求最少 20 字

（2）图片要求最少 2 张，调整 `mean_value`,`bias` 分别截取所训练的逻辑回归模型

打卡截止时间：10/19

1.4 本周学习任务简单总结

学习时长： 10/20——10/20

任务简介： 温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

本周所学的几乎不涉及机器学习，深度学习概念，都是 `pytorch` 的基础概念知识，包括数据载体——张量，前向传播搭建计算图的概念，利用计算图和自动求导系统反向传播求取梯度，最后介绍机器学习模型训练的步骤。

视频：

（1） `pytorch` 第一周作业讲解-1

（2） `pytorch` 第一周作业讲解-2

（3） `pytorch` 第一周作业讲解-3

作业名称（详解）：

请用文字描述本周所学知识的重点，并且为每节课列出关键词，将这些本周

的关键词以序号进行排序；

作业提交形式： PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

打卡截止时间： 10/20

2 第 2 周

2.1 PyTorch 数据读取机制 Dataloader 与 Dataset；数据预处理

transforms 模块机制

学习时长： 10/21—10/22

任务简介： 学习 PyTorch 数据读取机制中的两个重要模块 Dataloader 与 Dataset；熟悉数据预处理 transforms 方法的运行机制

详细说明：

（1）本节第一部分介绍 `pytorch` 的数据读取机制，通过人民币分类实验来学习 `pytorch` 是如何从硬盘中读取数据的，并且深入学习数据读取过程中涉及到的两个模块 Dataloader 与 Dataset

（2）第二部分介绍数据的预处理模块 `transforms` 的运行机制，数据在读取到 `pytorch` 之后通常都需要对数据进行预处理，包括尺寸缩放、转换张量、数据中心化或标准化等等，这些操作都是通过 `transforms` 进行的，所以本节重点学习 `transforms` 的运行机制并介绍数据标准化(Normalize)的使用原理。

视频：

- （1） Dataloader 与 Dataset
- （2） transforms 与 normalize

作业名称（详解）：

（1）采用步进(Step into)的调试方法从 `for i, data in enumerate(train_loader)` 这一行代码开始，进入到每一个被调用函数，直到进入 `RMBDataset` 类中的 `__getitem__` 函数，记录从 `for` 循环到 `RMBDataset` 的 `__getitem__` 所设计的类与函数？（文字简答）

例如：

第一步：for i, data in enumerate(train_loader)

第二步：DataLoader 类，__iter__函数

第三步：***类， ***函数

第 n 步：RMBDataset 类，__getitem__函数

（2）训练 RMB 二分类模型，熟悉数据读取机制，并且从 kaggle 中下载猫狗二分类训练数据，自己编写一个 DogCatDataset，使得 pytorch 可以对猫狗二分类训练集进行读取。

数据下载：

<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data>

百度网盘链接：

链接：<https://pan.baidu.com/s/16pAXRJtKsOMrKx85Ay0QxA>

提取码：[wnn6](#)

打卡要求：提供 DogCatDataset 代码的实现截图或代码文字。

打卡截止时间：10/23

2.2 学习二十二种 transforms 数据预处理方法；学会自定义 transforms 方法

学习时长：10/24—10/26

任务简介：pytorch 提供了大量的 transforms 预处理方法，在这里归纳总结为四大类共二十二种方法进行一一学习；学会自定义 transforms 方法以兼容实际项目；

详细说明：

（1）本节第一部分介绍 transforms 方法中的裁剪、翻转和旋转，共 8 种方法，同时介绍如何将输入进模型的数据进行反 transforms 操作，变为可以可视化的图片；

（2）本节第二部分介绍 transforms 中的 8 种图像变换，3 种 transforms 方法的选

择方法；除了 pytorch 提供的预处理方法，还教大家自定义 transforms 方法，先通过熟悉 transforms 的运行机制，总结自定义 transforms 方法的注意事项，然后制定自定义 transforms 方法的基本结构，最后通过椒盐噪声的 transforms 实例来学习自定义 transforms 方法

视频：

(1) transforms (一)

(2) transforms (二)

作业名称（详解）：

(1) 将介绍的 transforms 方法一一地，单独地实现对图片的变换，并且通过 plt.savefig 将图片保存下来

打卡要求：截图，不少于 10 张不一样的数据增强变换的图片，如裁剪，缩放，平移，翻转，色彩变换，错切，遮挡等等

(2) 自定义一个增加椒盐噪声的 transforms 方法，使得其能正确运行

打卡要求：文字简答，复制 YourTransforms 类的代码

(3) 用手机将自己钱包中的 100 元人民币正面进行拍照，并且放到 hello pytorch/lesson/lesson-09/test_data/100 文件夹下，通过修改不同的数据增强方法，使得模型在 10 个 epoch 之后能正确区分你钱包中的 100 元

打卡要求：截图，将正确分类的 100 元的 plt.show 进行截取

打卡截止时间：10/26

2.3 本周学习任务简单总结

学习时长：10/27

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

本周所学的知识点都是属于机器学习训练模块中的数据模块，对数据进行读取，

预处理, 增强, 在其中包含了大量的方法函数, 如 `Dataloader`, `Dataset`, `transforms` 系列等等, 大家需要对其中各个函数的作用和运行机制要有一个清晰的认识, 认真总结本周所讲解的数据模块中的知识点。

作业名称 (详解):

请用文字描述本周所学知识的重点, 并且为每节课列出关键词, 将这些本周的关键词以序号进行排序;

作业提交形式: PPT 截图或手写拍照, 打卡提交.

打卡内容: (可以只是文字提交, 或图片提交, 或组合都行)

打卡截止时间: 10/27

3 第 3 周

3.1 nn.Module 与网络模型构建步骤; 模型容器与 AlexNet 构建

学习时长: 10/28—10/29

任务简介: 学习 `nn.Module` 类以及搭建网络模型步骤; 熟悉搭建网络模型时常用的模型容器

详细说明:

(1) 本节第一部分介绍网络模型的基本类 `nn.Module`, `nn.Module` 是所有网络层的基本类, 它拥有 8 个有序字典, 用于管理模型属性, 本节课中将要学习如何构建一个 `Module`。然后通过网络结构和计算图两个角度去观察搭建一个网络模型需要两个步骤: 第一步, 搭建子模块; 第二步, 拼接子模块。

(2) 本节第二部分介绍搭建网络模型常用的容器, 如 `Sequential`, `ModuleList`, `ModuleDict`, 然后学习 `pytorch` 提供的 `Alexnet` 网络模型结构加深对模型容器的认识。

视频:

- (1) 模型创建步骤与 `nn.Module`
- (2) 模型容器与 `AlexNet` 构建

作业名称 (详解):

(1) 采用步进(Step into)的调试方法从创建网络模型开始(`net = LeNet(classes=2)`)进入到每一个被调用函数, 观察 `net` 的 `_modules` 字段何时被构建并且赋值, 记录其中所有进入的类与函数

例如:

第一步: `net = LeNet(classes=2)`

第二步: `LeNet` 类, `__init__()`, `super(LeNet, self).__init__()`

第三步: `Module` 类,

第 `n` 步: 返回 `net`

打卡有求: 文字简答

(2) 采用 `sequential` 容器, 改写 `Alexnet`, 给 `features` 中每一个网络层增加名字, 并通过下面这行代码打印出来:

```
print(alexnet._modules['features']._modules.keys())
```

打开要求: 截图或文字

3.2 学习网络层中的卷积层, 池化层, 全连接层和激活函数层

学习时长: 10/31——11/2

任务简介: 学习网络模型中采用的神经网络层, 包括卷积层, 池化层, 全连接层和激活函数层, 学会如何区分二维卷积和三维卷积;

详细说明:

(1) 本节第一部分学习卷积神经网络中最重要的卷积层, 了解卷积操作的过程与步骤, 同时学会区分一维/二维/三维卷积, 最后学习转置卷积 (`Transpose Convolution`) 的由来以及实现方法;

(2) 本节第二部分学习池化层, 全连接层和激活函数层, 在池化层中有正常的最大值池化, 均值池化, 还有图像分割任务中常用的反池化——`MaxUnpool`, 在激活函数中会学习 `Sigmoid`, `Tanh` 和 `Relu`, 以及 `Relu` 的各种变体, 如 `LeakyReLU`, `PReLU`, `RReLU`

视频:

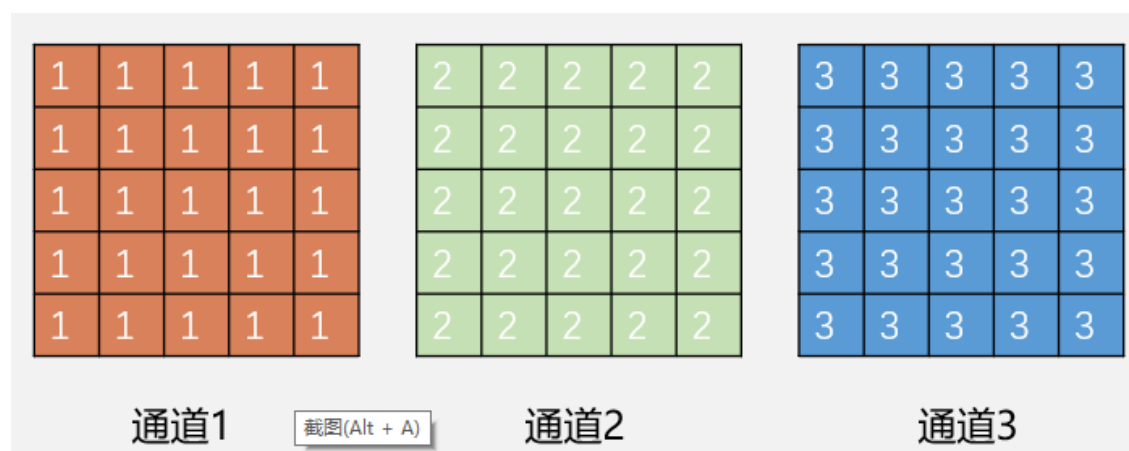
- (1) nn 网络层-卷积层
- (2) 网络层-池化-线性-激活函数层

作业名称（详解）：

(1) 深入理解二维卷积，采用手算的方式实现以下卷积操作，然后用代码验证

1) 采用 2 个尺寸为 3*3 的卷积核对 3 通道的 5*5 图像进行卷积，padding=0，stride=1，dilation=0

其中 input shape = (3, 5, 5)，数据如下



kernel size = 3*3， 第一个卷积核所有权值均为 1， 第二个卷积核所有权值均为 2， 计算输出的 feature map 尺寸以及所有像素值

2) 接 1) 题，上下左右四条边均采用 padding，padding=1，填充值为 0，计算输出的 feature map 尺寸以及所有像素值

打卡要求：代码输出结果截图，打印输出 feature map 的值

(2) 对 lena 图进行 3*3*3d 卷积，提示：padding= (1, 0, 0)

```
# ===== 3d
# flag = 1
flag = 0
if flag:
    conv_layer = nn.Conv3d(3, 1, (1, 3, 3), padding=(1, 0, 0))
    nn.init.xavier_normal_(conv_layer.weight.data)
```

```
# calculation  
img_tensor.unsqueeze_(dim=2)  # B*C*H*W to B*C*D*H*W  
img_conv = conv_layer(img_tensor)
```

打卡截止日期：11/2

3.3 本周学习任务简单总结

学习时长：11/3

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

本周所学的知识点都是属于机器学习训练模块中的模型模块，详细讲解了，模型的搭建步骤，模型基本类 `nn.Module`，以及常用的网络层，1d/2d/3d 卷积层，最大池化层，反池化层，平均池化层，线性层，Sigmoid 激活函数，Tanh 激活函数，ReLU 及其各种变种 LeakyReLU，PReLU，RReLU 等知识点，内容比较多，大家要认真总结本周所讲解的知识点。

作业名称（详解）：请用文字描述本周所学知识的重点，并且为每节课列出关键词，将这些本周的关键词以序号进行排序；

作业提交形式：PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

打卡截止时间：11/3

4 第4周

4.1 权值初始化；损失函数（一）

学习时长：11/4—11/5

任务名称：权值初始化；损失函数（一）

任务简介：学习权值初始化的原理；介绍损失函数、代价函数与目标函数的关系，

并学习交叉熵损失函数

详细说明:

(1) 本节第一部分讲解权值初始化的必要性, 首先分析神经网络中权值的方差过大导致梯度爆炸的原因, 然后从方差一致性原则出发分析 Xavier 初始化方法与 Kaiming 初始化方法的由来, 最后介绍 pytorch 提供的十种初始化方法。

(2) 本节第二部分介绍损失函数、代价函数与目标函数的联系与不同之处, 然后学习人民币二分类任务中使用到的交叉熵损失函数, 在讲解交叉熵损失函数时补充分析自信息、信息熵、相对熵和交叉熵之间的关系, 最后学习四种损失函数:

1. nn.CrossEntropyLoss
2. nn.NLLLoss
3. nn.BCELoss
4. nn.BCEWithLogitsLoss

视频:

- (1) 权值初始化
- (2) 损失函数-1

作业名称 (详解):

(1) Lossfunction 依旧属于网络层的概念, 即仍旧是 Module 的子类, 为了对 lossfunction 有一个更清晰的概念, 需要大家采用步进(Step into)的调试方法从 `loss_functoin = nn.CrossEntropyLoss()` 语句进入函数, 观察从 `nn.CrossEntropyLoss()`到 `class Module(object)`一共经历了哪些类, 记录其中所有进入的类及函数。

例如:

第一步: CrossEntropyLoss 类, `super(CrossEntropyLoss, self).__init__`

第二步:

第三步:

第 n 步: 进入 Module 类

打卡有求: 文字简答或截图

(2) 损失函数的 **reduction** 有三种模式，它们的作用分别是什么？

当 **inputs** 和 **target** 及 **weight** 分别如以下参数时，**reduction='mean'** 模式时，**loss** 是如何计算得到的？

```
inputs = torch.tensor([[1, 2], [1, 3], [1, 3]], dtype=torch.float)
target = torch.tensor([0, 1, 1], dtype=torch.long)
weights = torch.tensor([1, 2])
```

打卡要求：文字简答以及代码截图

4.2 pytorch 的 14 种损失函数

学习时长：11/7—11/8

任务简介：学习 pytorch 中剩下的 14 种损失函数；学习优化器 **optimizer** 的基本属性、基本方法和作用。

详细说明：

(1) 本节第一部分学习 pytorch 的另外 14 种损失函数：

5. nn.L1Loss

6. nn.MSELoss

7. nn.SmoothL1Loss

8. nn.PoissonNLLLoss

9. nn.KLDivLoss

10. nn.MarginRankingLoss

11. nn.MultiLabelMarginLoss

12. nn.SoftMarginLoss

13. nn.MultiLabelSoftMarginLoss

14. nn.MultiMarginLoss

15. nn.TripletMarginLoss

16. nn.HingeEmbeddingLoss

17. nn.CosineEmbeddingLoss

18. nn.CTCLoss

(2) 本节第二部分学习优化器的基本概念，了解 pytorch 中 optimizer 的基本属性和方法

视频：

(1) 损失函数-2

(2) 优化器-1

作业名称（详解）：

(1) 总结所学习的 18 种损失函数，制作思维导图或总结表等

打卡要求：文字简答或截图

4.3 torch.optim.SGD

学习时长：11/9

任务简介：学习最常用的优化器， optim.SGD

详细说明：深入了解学习率和 momentum 在梯度下降法中的作用，分析 LR 和 Momentum 这两个参数对优化过程的影响，最后学习 optim.SGD 以及 pytorch 的十种优化器简介。

视频：

优化器-2

作业名称（详解）：

(1) 优化器的作用是管理并更新参数组，请构建一个 SGD 优化器，通过 add_param_group 方法添加三组参数，三组参数的学习率分别为 0.01, 0.02, 0.03，momentum 分别为 0.9, 0.8, 0.7，构建好之后，并打印优化器中的 param_groups 属性中的每一个元素的 key 和 value（提示：param_groups 是 list，其每一个元素是一个字典）

打卡要求：代码或者文字截图

4.4 本周学习任务简单总结

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。本周所学的知识点非常多，涉及十种权值初始化，十八种损失函数以及十种优化器，在这三个概念中，又分别涉及，梯度爆炸，方差一致性原则，损失函数与代价函数和目标函数的关系，信息熵、相对熵与交叉熵的关系，梯度下降法，指数加权平均等等概念知识，希望大家课后认真总结学习。

作业名称（详解）：请用文字描述本周所学知识的重点，并且为每节课列出关键词，将这些本周的关键词以序号进行排序；

作业提交形式：PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

打卡截止时间：11/11

5 第5周

5.1 学习率调整；TensorBoard 简介与安装

任务时长：11/11

任务简介：熟悉 pytorch 的学习率调整策略；安装可视化工具 TensorBoard

详细说明：

（1）本节第一部分讲解 pytorch 中提供的学习率调整策略，首先介绍基类 `_LRScheduler` 基本属性与方法，然后逐个学习率方法进行讲解使用，分别 `Step`、`MultiStep`、`Exponential`、`CosineAnnealing`、`ReduceLROnPlateau` 和 `Lambda`，一共六种学习率调整策略；

（2）本节第二部分讲解可视化工具 TensorBoard 的运行机制与安装过程，TensorBoard 是强大的可视化工具，起初为 TensorFlow 的副产品，但目前 PyTorch 已支持 TensorBoard 的使用。目前，TensorBoard 支持 Scalars, Images, Audio, Graphs,

Distributions, Histograms, Embeddings, Text 等数据的可视化。

视频:

- (1) 学习率调整策略
- (2) TensorBoard 简介与安装

作业名称 (详解):

- (1) 熟悉 TensorBoard 的运行机制, 安装 TensorBoard, 并绘制曲线 $y = 2 * x$

打卡要求: 截图

打卡截止时间: 11/12

5.2 TensorBoard 使用

任务时长: 11/14-11/15

任务简介: 学习 TensorBoard 中 scalar 与 histogram 的使用; 学习 TensorBoard 中 Image 与 PyTorch 的 make_grid 使用

详细说明:

(1) 本节第一部分学习 TensorBoard 的 SummaryWriter 类的基本属性, 然后学习 add_scalar, add_scalars 和 add_histogram 的使用, 最后采用所学函数实现模型训练过程中的 Loss 曲线, Accuracy 曲线的对比监控, 同时对参数及其梯度的分布进行可视化。

(2) 本节第二部分学习 TensorBoard 的 add_image 方法, 并学习 PyTorch 的 make_grid 函数构建网格图片, 对批量图片进行可视化, 最后采用所学函数对 AlexNet 网络卷积核与特征图进行可视化分析。

视频:

- (1) TensorBoard 使用-1
- (2) TensorBoard 使用-2

作业名称 (详解):

(1) 可视化任意网络模型训练的 Loss，及 Accuracy 曲线图，Train 与 Valid 必须在同一个图中

打卡要求：截图

(2) 采用 `make_grid`，对任意图像训练输入数据进行批量可视化

打卡要求：截图

打卡截止日期：11/15

5.3 hook 函数与 CAM(class activation map, 类激活图)

任务时长：11/16

任务简介：学习 pytorch 的 hook 函数机制以及 CAM 可视化算法

详细说明：深入学习了解 pytorch 的 hook 函数运行机制，介绍 pytorch 中提供的 4 种 hook 函数，分别为：

1. `torch.Tensor.register_hook(hook)`
2. `torch.nn.Module.register_forward_hook`
3. `torch.nn.Module.register_forward_pre_hook`
4. `torch.nn.Module.register_backward_hook`

最后，介绍 CAM 可视化及其改进算法 Grad-CAM

视频：

hook 函数与 CAM 可视化

作业名称：

(1) 采用 `torch.nn.Module.register_forward_hook` 机制实现 AlexNet 第一个卷积层输出特征图的可视化，并将 `/torchvision/models/alexnet.py` 中第 28 行改为：

`nn.ReLU(inplace=False)`，观察

`inplace=True` 与 `inplace=False` 的差异。

打卡要求：截图

打卡截止时间：11/16

5.4 本周学习任务简单总结

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

作业名称（详解）：

请用文字描述本周所学知识的重点，并且为每节课列出关键词，将这些本周的关键词以序号进行排序；

作业提交形式：PPT 截图或手写拍照,打卡提交.

打卡内容：（可以只是文字提交，或图片提交，或组合都行）

6 第6周

6.1 正则化

任务时长：11/18-11/19

任务简介：了解正则化中 L1 和 L2（weight decay）；了解 dropout

详细说明：

（1）本节第一部分讲解正则化的概念，正则化方法是机器学习（深度学习）中重要的方法，它目的在于减小方差。常用的正则化方法有 L1 和 L2 正则化，其中 L2 正则化又称为 weight decay。在 pytorch 的优化器中就提供了 weight decay 的实现，本节课将学习 weight decay 的 pytorch 实现。

（2）本节第二部分讲解深度学习中常见的正则化方法——Dropout，Dropout 是简洁高效的正则化方法，但需要注意其在实现过程中的权值数据尺度问题。本节课将详细介绍 pytorch 中 Dropout 的实现细节。

视频：

- （1）正则化之 weight_decay
- （2）正则化之 Dropout

作业名称（详解）：

（1）weight decay 在 pytorch 的 SGD 中实现代码是哪一行？它对应的数学公式为？

打卡有求：文字或截图

（2）PyTorch 中，Dropout 在训练的时候权值尺度会进行什么操作？

打卡有求：文字

打卡截止时间：11/19

6.2 归一化 (Normalized)

任务时长：11/18—11/19

任务简介：学习深度学习中常见的标准化方法

详细说明：

（1）本节第一部分介绍深度学习中最重要的一個 Normalization 方法——Batch Normalization，并分析其计算方式，同时讲解 PyTorch 中 nn.BatchNorm1d、nn.BatchNorm2d、nn.BatchNorm3d 三种 BN 的计算方式及原理。

（2）本节第二部分介绍 2015 年之后出现的常见的 Normalization 方法——Layer Normalization、Instance Normalization 和 Group Normalization，分析各 Normalization 的由来与应用场景，同时对比分析 BN，LN，IN 和 GN 之间的计算差异。

视频：

（1）Batch Normalization

（2）Normalization_layers

作业名称（详解）：

（1）Batch Normalization 的 4 个重要参数分别是什么？BN 层中采用这四个参数对 X 做了什么操作？

打卡要求：文字描述

（2）课程结尾的“加减乘除”是什么意思？

打卡要求：文字描述

打卡截止时间：11/22

6.3 本周学习任务简单总结

任务时长：11/24

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

任务截止时间：11/24

7 第 7 周

7.1 模型保存与加载；Finetune

任务时长：11/25-11/26

任务简介：了解序列化与反序列化；了解 transfer learning 与 model finetune

详细说明：

(1) 本节第一部分学习 `pytorch` 中的模型保存与加载，也常称为序列化与反序列化，本节将讲解序列化与反序列化的概念，进而对模型的保存与加载有深刻的认识，同时介绍 `pytorch` 中模型保存与加载的方法函数。

(2) 本节第二部分学习模型微调（Finetune）的方法，以及认识 Transfer Learning（迁移学习）与 Model Finetune 之间的关系。

视频：

- (1) 模型保存与加载
- (2) 模型 finetune

作业名称（详解）：

- (1) 任意一个模型，将其内部权重赋值为 0.123，然后保存 `state_dict`；接着重新

创建一个网络 `net_load`，打印刚创建的 `net_load` 中的参数，接着加载保存好的 `state_dict`，并且 `load` 到 `net_load` 中，再次打印 `net_load` 中的参数，观察是否成功将保存的参数重新加载进来

打卡有求：赋值 `print` 的信息或截图重新加载进来的参数的数值

（2）一句话描述迁移学习研究内容

打卡要求：文字

打卡截止时间：11/26 23:59

7.2 GPU 的使用；PyTorch 中常见报错

任务时长：11/28—11/29

任务简介：学习使用 GPU 进行加速运算；学习常见报错信息，方便调试代码

详细说明：

（1）本节第一部分学习如何使用 GPU 进行加速模型运算，介绍 `Tensor` 和 `Module` 的 `to` 函数使用以及它们之间的差异，同时学习多 GPU 运算的分发并行机制

（2）本节第二部分学习 PyTorch 中常见的报错信息及其解决方法，为大家提供一个快速的调试文档，同时希望大家一同贡献该文档。

视频：

（1）GPU 的使用

（2）PyTorch 常见报错

作业名称（详解）：

（1）采用 `time` 函数与 `for` 循环，对比 `cpu` 上运算时间以及 `gpu` 上运算时间

打卡要求：文字描述 `cpu` 型号，`gpu` 型号，运算时间对比

（2）尝试贡献一个 `pytorch` 中的报错信息/存在的坑（选做）

打卡要求：石墨文档编辑

打卡截止时间：11/29 23:59

7.3 本周学习任务简单总结

任务时长：12/1

任务简介：温故而知新，简单回顾本周学到几个重要知识

详细说明：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

任务截止时间：12/1