lecture 3(dft,dct,dwt) ,基础pytorch操作 and 天池学习来的简单cnn模型

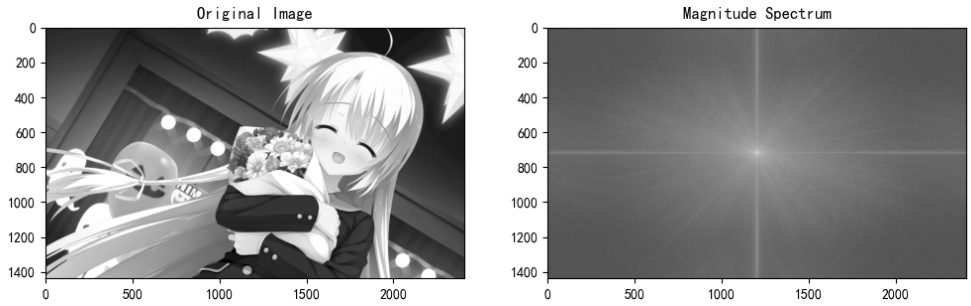
dft,dct,dwt

频域与时域

时域就是我们生活中的坐标空间，例如二维坐标，三维坐标等，频域是由现实坐标由正余弦组成

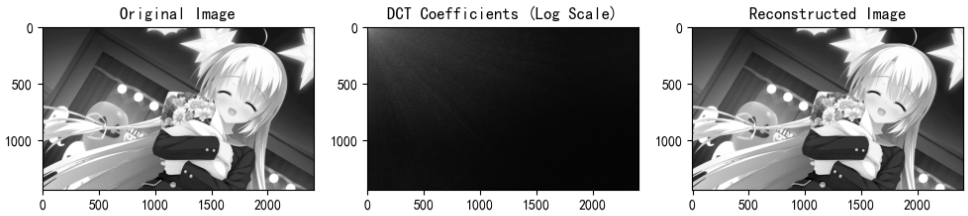
dft

在图像处理的运用方面，DFT主要用于去除杂质成分对图像造成的干扰，如图像去噪等。图像经过离散傅里叶变换之后从时域信息变成了频域信息进而分为高频部分和低频部分，对高频噪声进行滤波即可去除掉在时域中难以区分的噪声及杂质成分。



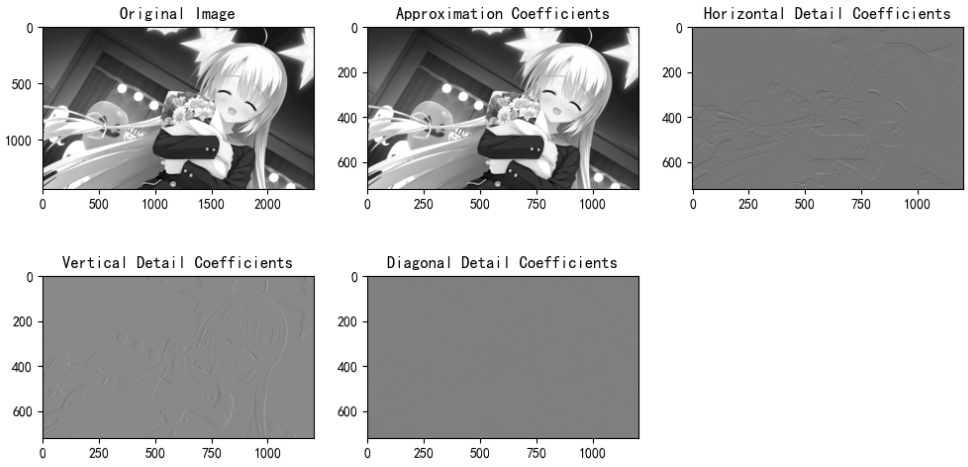
dct

DCT是DFT的一种特殊的形式，DCT是对实偶函数进行转换的，它相当于一个长度大概是其两倍的傅里叶变换，并且变换之后得到的函数仍然是实偶函数。DCT则主要用于图像的压缩。图像经过离散余弦变换后其基本信息主要集中在左上角，因此可以去除除左上角之外的其他数据也能很好的将图像复原成原始的样子，因此能够在误差可接受的范围内将图像进行压缩储存。



dwt

DWT和DCT的区别在于图像进行DWT变换后其小波域分为四个子带，每个子带不仅包括图像的频域成分还包括其空域成分。并且其包含图像主要信息的左上角子带(LL子带)能够再次不断的进行DWT变换从而将其连续分解成许多不同分辨率的信号成分，这意味着我们可以通过控制小波变换的层数来实现不同的压缩率目标。



pytorch基本操作入门

卷积

我们之前就接触过各种算子（滤波器就是算子的一部分），算子的核心就是各种卷积，服从高斯分布（正态分布）的高斯卷积，二阶微分的拉普拉斯算子，腐蚀膨胀的最大最小卷积操作(gpt说是类似，传统卷积是通过加权求和的)，中值卷积，平均卷积

最大池化

最大池化操作通过在输入特征图上滑动一个固定大小的窗口，并在每个位置计算该窗口内所有值的最大值，从而生成一个新的缩小后的特征图。具体步骤如下：

1. **选择窗口大小**：通常使用 (2 *2) 或 (3 * 3) 的窗口。

2. **选择步幅（stride）**：步幅决定了窗口滑动的距离。常见的步幅值是与窗口大小相同，例如步幅为2。
3. **滑动窗口**：从输入特征图的左上角开始，按照指定步幅逐一移动窗口。
4. **计算最大值**：对于每个窗口，计算其中的最大值，并将该最大值放入输出特征图的相应位置

最大池化的优点

1. **降维**：通过减少特征图尺寸，有助于降低计算开销。
2. **平移不变性**：保留特征图中的显著特征，使得模型对输入图像的小幅位移不敏感。
3. **减轻过拟合**：通过减少参数数量，有助于减轻模型的过拟合问题。

最大池化与其他池化操作

除了最大池化，还有其他类型的池化操作，如平均池化（Average Pooling），它计算窗口内所有值的平均值而不是最大值。不同的池化方法适用于不同的任务和数据集。

综上所述，最大池化是卷积神经网络中常用的一种技术，通过提取局部区域的最大值来缩小特征图尺寸，简化模型并提高其鲁棒性

线性层(全连接层/密集层)

在神经网络中的作用

线性层通常出现在神经网络的最后几层，用于将来自卷积层或其他层的高维特征转换为目标输出，例如：

- **分类问题**：线性层将特征映射到类别数量的输出节点，每个节点代表一个类的得分或概率。
- **回归问题**：线性层将特征映射到实际值或一组实际值。

实际应用场景

这种情况通常出现在卷积神经网络（CNN）的最后一层或者在图像特征提取之后。例如，你可能已经从图像数据中提取了一些特征，然后将这些特征通过全连接层来进行分类或者回归任务。

优化器

常见优化器

1. **随机梯度下降（SGD）**
2. **Adam**
3. **RMSprop**
4. **Adagrad**

优化器的基本使用步骤

1. **定义模型和损失函数**
2. **初始化优化器**
3. **在训练循环中使用优化器更新模型参数**

```python
    # 反向传播和优化
    optimizer.zero_grad()  # 清除梯度
    loss.backward()   # 计算梯度
    optimizer.step()  # 更新参数
```

优化器（Optimizer）是用于调整神经网络参数（如权重和偏置）的算法，通过最小化损失函数来改进模型的性能

损失函数

损失函数（Loss Function）是用于衡量模型预测值与实际值之间差异的函数。它在模型训练过程中扮演着至关重要的角色，因为优化器通过最小化损失函数来调整模型参数，从而提升模型的性能和准确性

常见损失函数

1. **均方误差（MSE Loss）**
2. **交叉熵损失（Cross Entropy Loss）**
3. **二元交叉熵损失（Binary Cross Entropy Loss）**
4. **平滑L1损失（Smooth L1 Loss）**

选择合适的损失函数

选择损失函数通常取决于具体的任务类型：

- **回归任务**：预测连续值，常用均方误差损失。
- **分类任务**：预测离散类别，常用交叉熵损失（二分类或多分类）。
- **目标检测等复杂任务**：可能需要组合多个损失函数，如平滑L1损失等。

非线性激活(ReLU和Sigmoid)

非线性激活就是用特定函数拟合，常见的有：relu和sigmoid，

1.relu是利用不同原函数组合的分段函数，通过max,min等组合的分段函数

2.sigmoid是类似傅里叶级数的思路，使用一些特点函数拟合的原不规则函数

正向传播与反向传播(backward())

正向传播是指从输入数据开始，通过神经网络的前向计算，将输入数据逐层传递至输出层的过程。在正向传播过程中，每一层的输入经过加权求和和激活函数处理后，作为下一层的输入，直到输出层得到最终的预测结果。

反向传播是指计算神经网络中每个参数对损失函数的导数，从而根据导数对参数进行更新的过程。其目标是调整神经网络的参数，使得预测值与实际值之间的误差最小化。利用的是**链式法则**

反向传播利用链式法则（Chain Rule）来计算每个参数的梯度。具体过程如下：

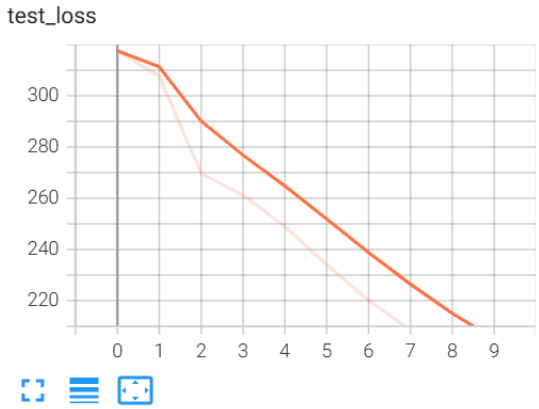
1. 根据当前参数的值，进行正向传播，得到预测值。
2. 计算预测值和实际值之间的损失（通常使用均方误差等损失函数）。
3. 从输出层开始，根据损失函数计算参数的梯度。
4. 使用梯度下降等优化算法，更新参数的值。
5. 重复上述步骤，直到达到停止条件（如达到最大迭代次数或损失函数收敛）。

通过反向传播，神经网络可以根据损失函数的反馈逐渐调整参数，使得神经网络的预测结果更加准确。

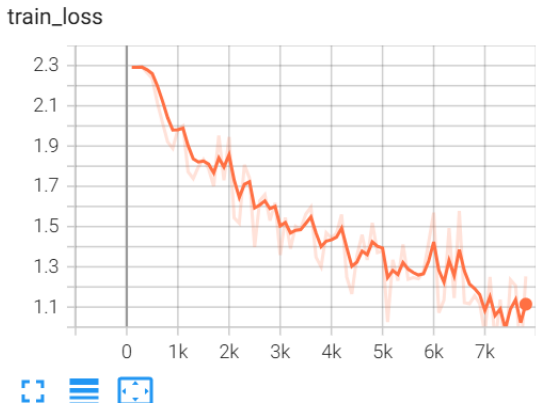
总结起来，正向传播是从输入到输出的信息传递过程，而反向传播是根据损失函数的反馈计算参数梯度并更新参数的过程。这两个步骤共同组成了神经网络的训练过程。

调用tensorboard的简单模型训练过程误差变化

test_loss



train_loss



其实我感觉上面的都不重要

个人感觉不如直接上数据集自己实操一个模型，但是自己去整理感觉贯通多了，之前一直尝试快速过渡基础部分，学的确实懵懵懂懂

cnn解决街景字符识别问题

cnn模型

CNN模型的组成部分：

1. **卷积层（Convolutional Layer）**：
   - 卷积层通过滤波器（或称为卷积核）在输入图像上进行滑动计算，提取局部特征。
   - 每个滤波器学习检测不同的特征，例如边缘、纹理等。
   - 多个滤波器形成多个输出通道，以捕捉丰富的特征信息。
2. **池化层（Pooling Layer）**：
   - 池化层通过减少特征图的空间尺寸，保留重要信息并减少计算量。
   - 最常见的池化操作是最大池化（Max Pooling），选取窗口内的最大值作为输出。
3. **非线性激活函数**：
   - 在卷积层和全连接层之间，通常使用非线性激活函数如ReLU，以增加模型的非线性能力。

4. **全连接层（Fully Connected Layer）**：
    - 全连接层接受前面卷积层和池化层的输出，将其展平并与权重矩阵相乘，生成最终的分类结果。
5. **其他技术**：
    - Dropout：随机失活部分神经元，有助于防止过拟合。
    - 批量归一化（Batch Normalization）：规范化每一层的输入，加速训练过程和提高模型稳定性。

CNN可以解决的问题：

1. **图像分类**：
    - CNN最经典的应用是图像分类，即给定一张图像，预测其属于哪个类别（如猫、狗等）。
2. **目标检测**：
    - CNN可以检测图像中多个对象的位置和类别，是许多计算机视觉任务（如自动驾驶、安防监控）的关键组成部分。
3. **语义分割**：
    - 将图像的每个像素分配到特定的类别，例如医学图像中的器官分割。
4. **人脸识别**：
    - CNN可以学习人脸图像的特征，进行个体识别或情感分析。
5. **图像生成**：
    - 利用生成对抗网络（GAN）结合CNN，可以生成逼真的图像，如艺术作品或虚拟现实场景。
6. **动作识别**：
    - 识别视频中的动作或行为，如体育比赛中的动作分类。

CNN由于其在处理图像和视频等网格化数据方面的优越性能和效率，成为计算机视觉领域最常用的深度学习模型之一，广泛应用于各种实际问题的解决方案中。

题目分析

数据集中带有图片的边框信息

Field	Description
top	左上角坐标X
height	字符高度
left	左上角最表Y
width	字符宽度
label	字符编码

首先题目中图片的数字图片是不定长的，我们可以进行字符串的填充，题目中说不超过6个字符长度，转化为6字符定长题，如：234*** ， 3456 * , * * * * *,123456

这样就可以利用cnn模型最典型的应用----图像分类

然后就确定好了方法，就是转化为定长数据集再进行cnn图像分类，然后与测试集检验，得到固定轮数后loss最小的模型，然后与test数据集求解得出最终误差loss

后续我的想法

继续学习机器学习，学习一些经典的朴素贝叶斯,随机决策树,K算法等打基础，顺便多打打类似天池,kaggle之类的比赛和找一些项目锻炼自己的实战能力

然后深入学习cnn,rcnn卷积神经网络，transfomer等经典深度学习模型，不过感觉gpu应该吃不消(只有一张英伟达4060哇)，然后就阅读一些论文什么的，学一下一些先进的idea，尝试进行论文的写作

最后研究自己选好的方向，学习这方面的论文，并尝试自己改造一些idea加一点自己的东西进去

不过说这么多都是虚假的，第一步就在原点走来走去找资源，走了几个星期原路

只能说慢慢来了