

人工智能中的编程第六次作业

本作业借鉴和参考了Needle项目 <https://github.com/dlsyscourse/lecture5>，拟基于上次作业hw5实现的自动微分框架来实现一个简单的优化器。

我们将会在手写数字识别任务(MNIST)上进行测试。你需要写一个softmax-loss函数来对于输入的图片进行分类的loss进行评估，然后使用梯度下降法来进行优化。

首先是Softmax函数会将logits 转换成概率分布。定义如下的分类loss，需要计算logits，其中 k 为类别数， z_i 为第 i 个类别的logits， z_y 为正确类别的logits。

$$\ell_{\text{softmax}}(z, y) = \log \sum_{i=1}^k \exp(z_i - z_y)$$

对于双层Linear层，激活函数选用ReLU的分类器，logits可以写成 $z = W_2^T \text{ReLU}(W_1^T x)$ ，其中 m 是数据数目， W_1 和 W_2 是两个Linear层的权重， x 是输入的图片。对于其他的网络结构，可以通写为 $z = \Theta^T x$ ，于是优化的目标为：

$$\text{minimize}_{\Theta} \frac{1}{m} \sum_{i=1}^m \ell_{\text{softmax}}(\Theta^T x^{(i)}, y^{(i)})$$

接下来我们把训练集写成 $X \in \mathbb{R}^{m \times n}$ ， $y \in \{1 \dots k\}^m$ 的形式，经过推演可以得到梯度的计算公式为：

$$\nabla_{\Theta} \ell_{\text{softmax}}(X\Theta, y) = \frac{1}{m} X^T (Z - I_y)$$

其中 $Z = \text{normalize}(\exp(X\Theta))$ ， I_y 是一个one-hot向量，对于正确的类别为1，其他为0。你可以利用这个梯度来对参数矩阵进行更新。接着你可以尝试加上动量和学习率衰减等优化策略来提高优化器的性能。(具体可以参考Adam部分代码的注释)

实现步骤：

- 读取数据集，你可以使用 `torchvision` 对于MNIST的数据进行下载和进行标准化的预处理，注意区分train和test数据集。
- 定义你的网络结构。
- 完成softmax-loss函数。
- 对于上述loss进行梯度计算，然后在 `task1_optimizer.py\SGD_epoch` 内用SGD算法更新网络参数。
- 在SGD的基础上，实现Adam优化策略。

文件说明

- 主要文件：`task1_optimizer.py` 中给出了进行优化的框架代码，你需要填写对应函数
- 基础环境：`task0_operators.py` 中定义了常用的运算符的正向和反向操作。在 `task0_autodiff.py` 中定义了进行自动微分的步骤。你可以将hw5的对应代码复制到这

里。

- 工具函数：在 `utils.py` 中提供了一些可能会用到的工具函数。为了避免交叉引用，可以根据需要将这些函数复制到你的代码中。

Tips

- 你可以在training的时候加上对于学习率调整的函数，来提高你的模型的性能。
- 不同网络结构会影响分类效果，我们将检测你的优化器实现是否正确来评分。
- 注意区分train和test数据集。

作业提交

- 本次作业总分5分，会根据完成情况按部分给分。我们会根据代码实现进行评分。
- 请于2025.12.4晚23:59前于 course.pku.edu.cn 提交代码和一份简短的报告说明。