

Numpy 搭建三层神经网络

数据集介绍

Fashion MNIST 是一个流行的数据集，常用于训练各种机器学习算法识别服装。这个数据集由 Zalando（一家德国的在线时尚零售商）发布，目的是作为传统的 MNIST 数据集的一个替代品，后者包含了手写数字的图片。Fashion MNIST 设计的初衷是提供一个稍微更具挑战性的问题，同时避免 MNIST 数据集的一些问题，如数据集过于简单而不适合现代机器学习算法。

Fashion MNIST 数据集包含 10 个类别的服装图片，如 T 恤/上衣、裤子、套头衫、裙子、外套、凉鞋、衬衫、运动鞋、包和踝靴。每个类别有 6,000 张训练图像和 1,000 张测试图像，图像的分辨率为 28x28 像素，为灰度图。这使得该数据集在处理和使用时与原始的 MNIST 数据集相似，可以无缝地用于测试不同的机器学习模型和算法。

Fashion MNIST 数据集的使用非常广泛，适合入门级的机器学习项目，是理解图像分类任务的一个很好的起点。同时，由于其相对简单但又比数字识别更接近实际应用，因此也适用于更高级的机器学习和深度学习课程和研究。

模型架构

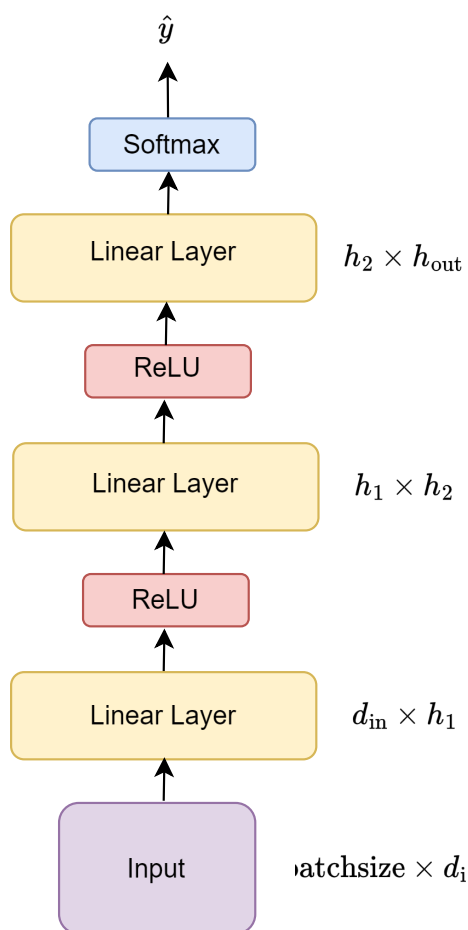


Figure 1: Architecture of the Multi-Layer Perceptron

实现细节

repo 地址 [here](#)

主要实现在文件夹 `./MLP/common/` 下

- `functions.py` 实现了 `softmax` 和交叉熵损失函数

- layers.py 实现了 Sigmoid, ReLU, Affine, Softmax, SoftmaxWithLoss 层的正向和反向传播
- module.py 实现了 Module 类, 作为一般多层神经网络的基类, 实现了以下方法
 - compute_accuracy 计算模型在给定数据集上的准确率
 - compute_macro_micro_avg 计算模型在给定数据集上的宏平均和微平均
 - save_parameters 保存模型参数
 - load_parameters 加载模型参数
- optimizer.py 实现了随机梯度下降(SGD), 支持学习率衰减和 l_2 -正则化. 此外还实现了 Momentum 和 Adam 优化器
- trainer.py 实现 Trainer 类
 - [load\save]_best_metrics 加载\保存最优指标
 - fit 训练神经网络. 每个 epoch 结束时计算一次在训练集和验证集上的指标. 根据当前 epoch 相应指标与历史最佳指标对比决定是否保存当前权重
 - plot 绘制训练集和验证集上的损失函数和分类准确率变化

实验结果

尝试了以下参数

```
hidden_sizes = [64, 128, 256]
lrs = [0.01, 0.001, 0.0001]
l2_regs = [0.0001, 0.001, 0.01]
```

hidden_sizes1 和 hidden_sizes2 取遍 hidden_sizes 中的值. 得到最优(以准确率为依据)参数组合为

```
{'hidden_size1':256, 'hidden_size2':64, 'lr':0.001, l2_reg':0.001}
```

其在训练集和验证集上损失函数值和准确率变化如下:

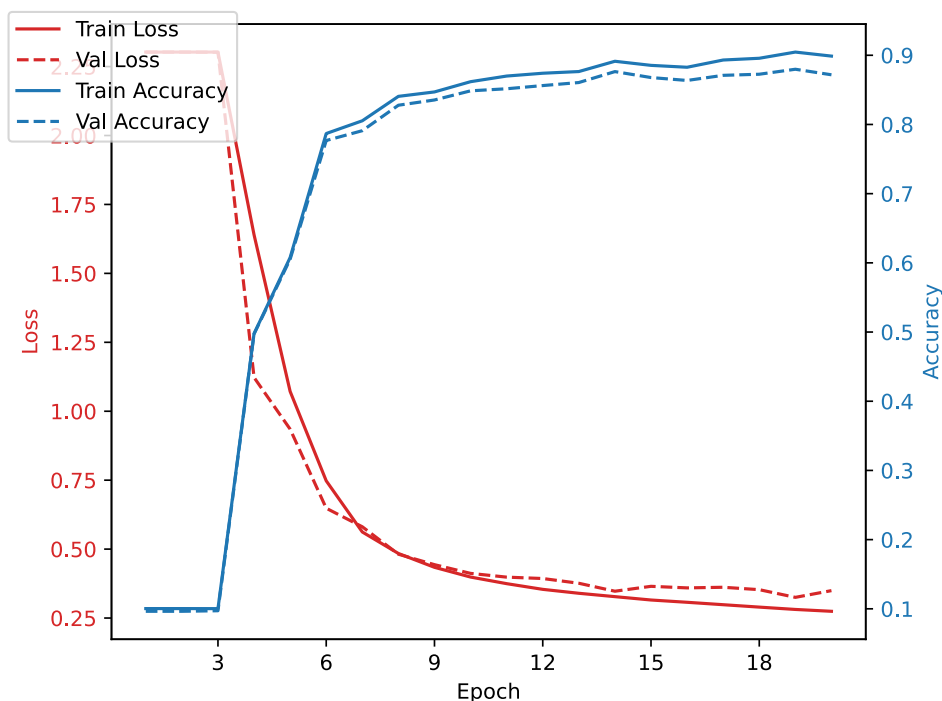


Figure 2: Loss and Accuracy Change on Training and Validation Set

在测试集上的准确率为86.7%.

可视化其权重得

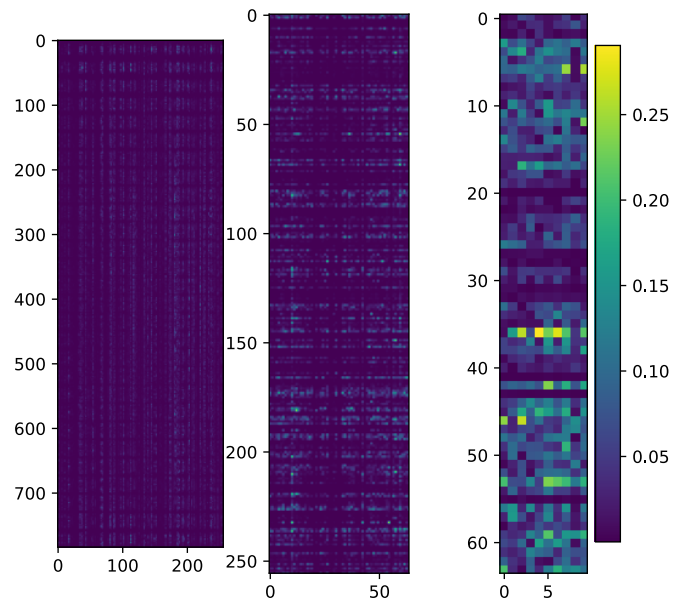


Figure 3: Visualization of Weights of Each Layer