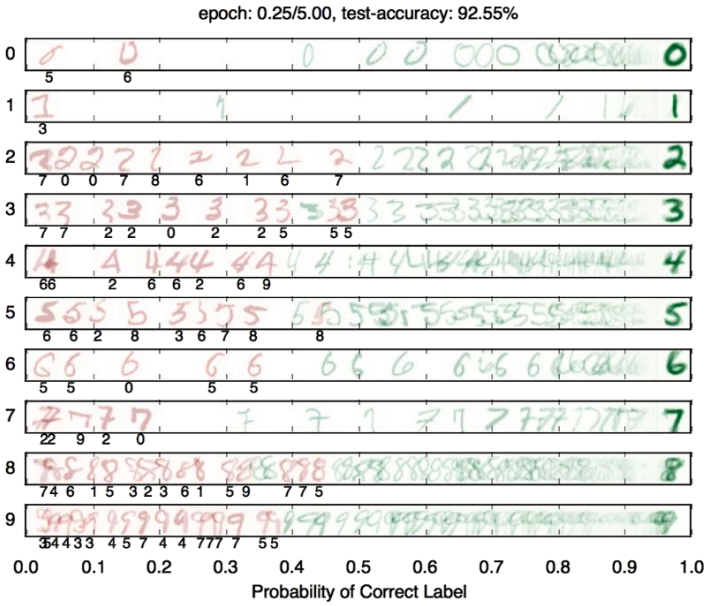
课程作业 4：机器学习（Machine Learning）



任务概述

在本次课程作业中，你将对机器学习算法进行初步探索，包括实现二进制感知机，非线性回归，以及构建神经网络来对数字进行分类。

本次课程作业需要安装如下两个依赖库（最好在 conda 虚拟环境下）：

```
pip install numpy
pip install matplotlib
```

已提供的代码包括一个神经网络迷你库（`nn.py`）和一组数据集（`backend.py`），具体用法示例可以参考：
https://inst.eecs.berkeley.edu/~cs188/sp24/projects/proj5/#original-project-provided-code-part-i 和
https://inst.eecs.berkeley.edu/~cs188/sp24/projects/proj5/#provided-code-part-ii-original-only

作业的程序包位于 QQ 群文件：课程作业\machinelearning.zip。

所有你自己的算法实现都位于 `models.py` 中相应任务的类和函数下面。

作业内容

任务 1：感知（Perceptron）

该任务需要实现一个二进制感知机，即**完成 `models.py` 中 `PerceptronModel` 类的实现。**

这里感知机的输出标签为 1 或者-1。样本数据已包含偏置特征，因此不需要单独的偏置（bias）参数。

需要完成的任务包括：

- 实现 `run(self, x)`方法，计算所存储的权重向量和给定输入的点积，并返回 `nn.DotProduct` 对象；
- 实现 `get_prediction(self, x)`，如果点积非负返回 1，否则返回-1。需要使用 `nn.as_scalar` 将标量 `Node` 转换为 Python 浮点数；
- 编写 `train(self)`方法。需要重复循环数据集并在错误分类的样本上进行更新。使用 `nn.Parameter` 类的 `update` 方法来更新权重。 当跑完整个数据集且没有任何错误时，训练准确率已达到 100%，训练可以终止；
- 这里更改参数值的唯一方法是调用 `parameter.update(direction, multiplier)`，该调用将执行权重更新：

$$\text{weights} \leftarrow \text{weights} + \text{direction} \cdot \text{multiplier}$$

为了测试你的代码,运行：

```
python autograder.py -q q1
```

神经网络小技巧（任务 2 和任务 3）：
https://inst.eecs.berkeley.edu/~cs188/sp24/projects/proj5/#neural-network-tips

任务 2：非线性回归（Non-linear Regression）

该任务需要训练一个神经网络来近似在[−2π,2π]上的 sin(x)，即**完成 `models.py` 中 `RegressionModel` 类的实现**。 对于这个任务，一个相对简单的架构就够用了（有关架构设计，请参阅神经网络小技巧）。使用 `nn.SquareLoss` 作为损失函数。

需要完成的任务包括：

- 实现 `RegressionModel.__init__`方法以进行初始化；
- 实现 `RegressionModel.run` 方法以返回模型的预测；
- 实现 `RegressionModel.get_loss` 方法以返回给定输入和目标输出的损失；
- 实现 `RegressionModel.train` 方法以使用基于梯度的更新来训练模型。

该任务只有一个数据集划分（即只有训练数据，没有验证数据或测试集）。 要求数据集中所有样本的平均损失≤0.02。可以使用训练损失来确定何时停止训练（使用 `nn.as_scalar` 将损失节点转换为 Python 数字）。模型训练的时间正常应该是数分钟。

为了测试你的代码，运行：

```
python autograder.py -q q2
```

任务 3：数字分类（Digit Classification）

该任务需要训练一个网络来对 MNIST 数据集中的手写数字进行分类。每张数字图片的大小为 28 x 28 像素，其值存储在 784 维浮点数向量中。输出目标为一个 10 维向量，与数字正确类别相对应的位置为 1，其他所有位置为 0，即是对数字类别的 one-hot 编码。

完成 `models.py` 中 `DigitClassificationModel` 类的实现。

`DigitClassificationModel.run()`的返回值应该是一个包含分数的 `batch_size x 10` 的节点，其中分数越高表示数字属于对应类别(0-9)的概率越高。使用 `nn.SoftmaxLoss` 作为损失函数。注意不要将 ReLU 激活函数置于网络的最后一个线性层。

该任务除了有训练数据之外，还有验证数据和测试集。

可以使用 `dataset.get_validation_accuracy()`来计算模型在验证数据上的准确率，以决定是否停止训练。测试集将在 autograder 中使用。

所实现的模型需要在测试集上的准确率应≥97%。需要注意的是，autograder 是在测试集上计算模型的准确率，而在模型训练过程中只能获得验证数据上的准确度，因此对验证准确度设置稍高的停止阈值可能会对通过测试有所帮助，例如 97.5%或者 98%。

为了测试你的代码，运行：

```
python autograder.py -q q3
```

各个任务的更详细描述可以参考：
https://inst.eecs.berkeley.edu/~cs188/sp24/projects/proj5/

作业报告

本次作业**可以两人为一组来完成**，需要提交报告和代码。对于以上任务，**报告需分别详细介绍代码的实现并分析实验结果**。使用 QQ 群文件中的报告模版（课程作业\报告模版.doc）撰写实验报告。如果作业由两人完成，**需要在报告中体现两人的具体分工。**

作业提交

将作业报告存储为PDF文件，并与相应的源码打包为zip文件，命名为学号1_学号2.zip，例如 221900001_221900002.zip。

上传到百度网盘：
https://pan.baidu.com/disk/main#/transfer/send?surl=AAgAAAAAABLnSg

注意：与之前的课程作业是不同的网址。

提交截止日期: 12 月 19 日 23:59:59

学术诚信

允许同学之间的相互讨论，但是**署你名字的工作必须由你自己独立完成**。

如果发现作业之间高度相似将被判定为互相抄袭行为，**抄袭和被抄袭双方的成绩都将被取消**。

应项目开发者的要求，**严禁将作业答案发布在网上**。