

# 作业3：线性分类器与神经网络图像分类

## 任务一：线性分类器图像分类

- 数据：cifar10数据集
- 要求：构建一个线性分类器实现cifar10数据分类，其中

损失函数：交叉熵函数

优化方法：随机梯度速降法

- 1.画出分类器结构示意图；
- 2.使用精确率和召回率评估分类性能。

### 1. 实验原理

#### (1) 线性分类器基本原理

线性分类器是一种基本的机器学习模型，它通过学习输入特征的线性组合来进行分类。对于图像分类任务，输入是图像的像素值，输出是类别标签。线性分类器的基本数学表达式可以表示为：

$$f(x) = Wx + b$$

其中， $y$ 是模型的输出， $W$ 是权重矩阵， $x$ 是输入特征向量， $b$ 是偏置。模型的输出可以通过 softmax 函数进行归一化，以得到每个类别的概率分布。

#### (2) 交叉熵损失函数

交叉熵是一种用于衡量两个概率分布之间差异的损失函数。对于分类任务，交叉熵损失函数可以定义为：

$$L(y, \hat{y}) = - \sum_i y_i \cdot \log(\hat{y}_i)$$

其中， $y$ 是真实的标签分布， $\hat{y}$ 是模型的输出概率分布。最小化交叉熵损失函数有助于模型学习正确的类别概率分布。

#### (3) 随机梯度下降法 (Stochastic Gradient Descent, SGD)

随机梯度下降是一种优化算法，用于更新模型参数以最小化损失函数。在每一步中，该算法使用随机选择的一小部分训练数据（小批量）计算梯度，并根据负梯度方向更新模型参数。这样的迭代过程逐渐使模型收敛到损失函数的局部最小值。

#### (4) 构建线性分类器实现CIFAR-10数据分类

**模型结构示意图：**线性分类器的模型结构包括输入层、线性变换（包括权重矩阵 $W$ 和偏置 $b$ ）、softmax 归一化层。每个节点代表一个特征或一个神经元。

**训练步骤：**

- 前向传播：**将输入数据通过线性变换和 softmax 操作，得到模型输出。
- 计算损失：**使用交叉熵损失函数计算模型输出和真实标签之间的损失。
- 反向传播：**使用梯度下降法更新权重和偏置，减小损失。

- **迭代训练：** 重复上述步骤，直到模型收敛或达到指定的训练轮数。

#### 性能评估：

- 精确率和召回率：
  - 精确率 (Precision) :  $\frac{TP}{TP+FP}$  , 其中 TP 为真正例, FP 为假正例。
  - 召回率 (Recall) :  $\frac{TP}{TP+FN}$  , 其中 TP 为真正例, FN 为假负例。
- 混淆矩阵：

	预测正类别	预测负类别
实际正类别	TP	FN
实际负类别	FP	TN

- 精确率和召回率的计算：
- 精确率:  $\frac{TP}{TP+FP}$
- 召回率:  $\frac{TP}{TP+FN}$

通过以上步骤，可以构建一个简单的线性分类器，使用随机梯度下降法进行训练，并通过精确率和召回率对其性能进行评估。

## 2. 实验结果：

### (1) 分类器结构示意图

线性分类器的模型结构包括输入层、线性变换（包括权重矩阵 $W$ 和偏置 $b$ ）、softmax 归一化层。每个节点代表一个特征或一个神经元。结构示意图如下：

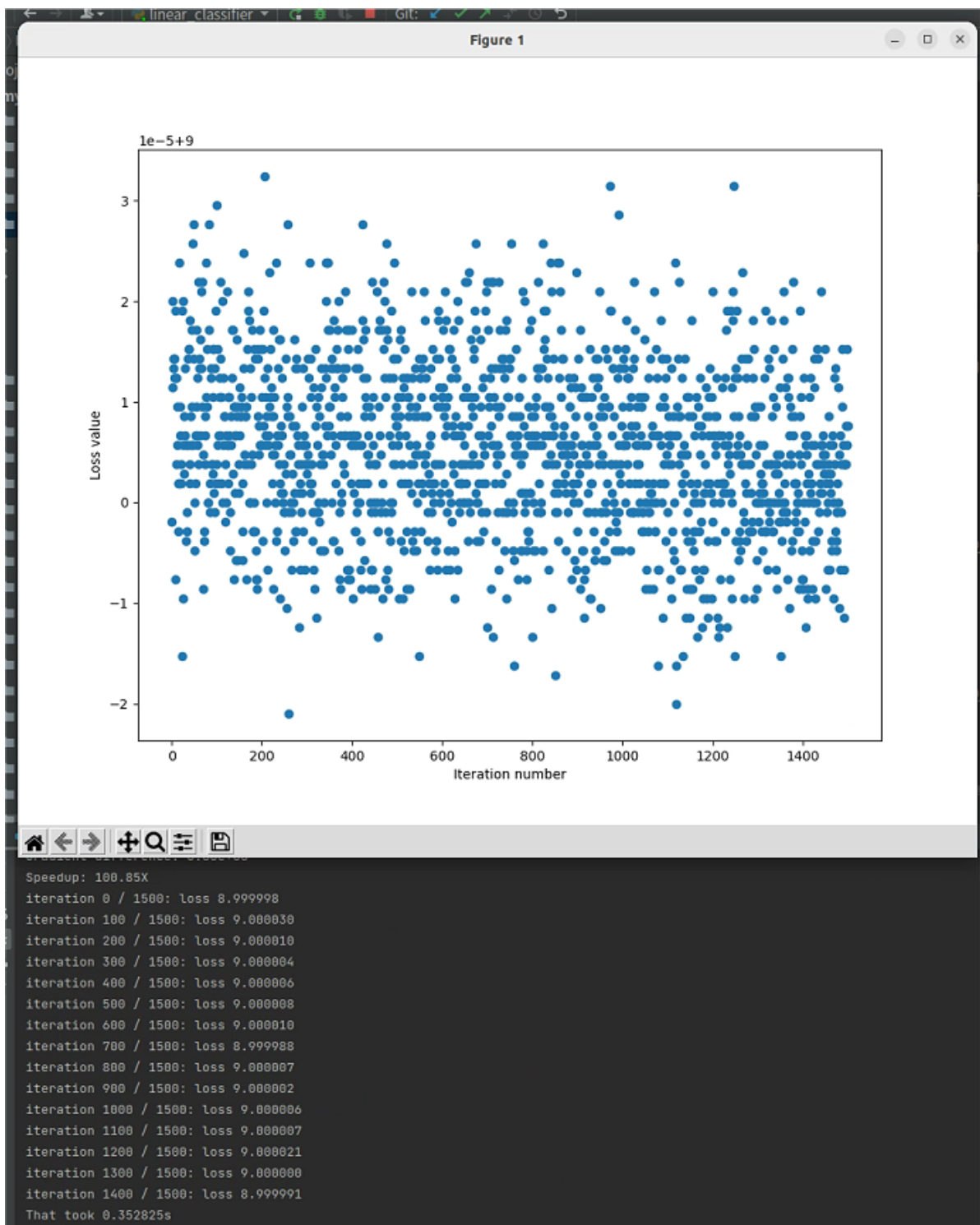
输入层（图像像素） -> [线性变换] -> [Softmax归一化] -> 输出层（类别概率）

### (2) 使用精确率和召回率评估分类性能

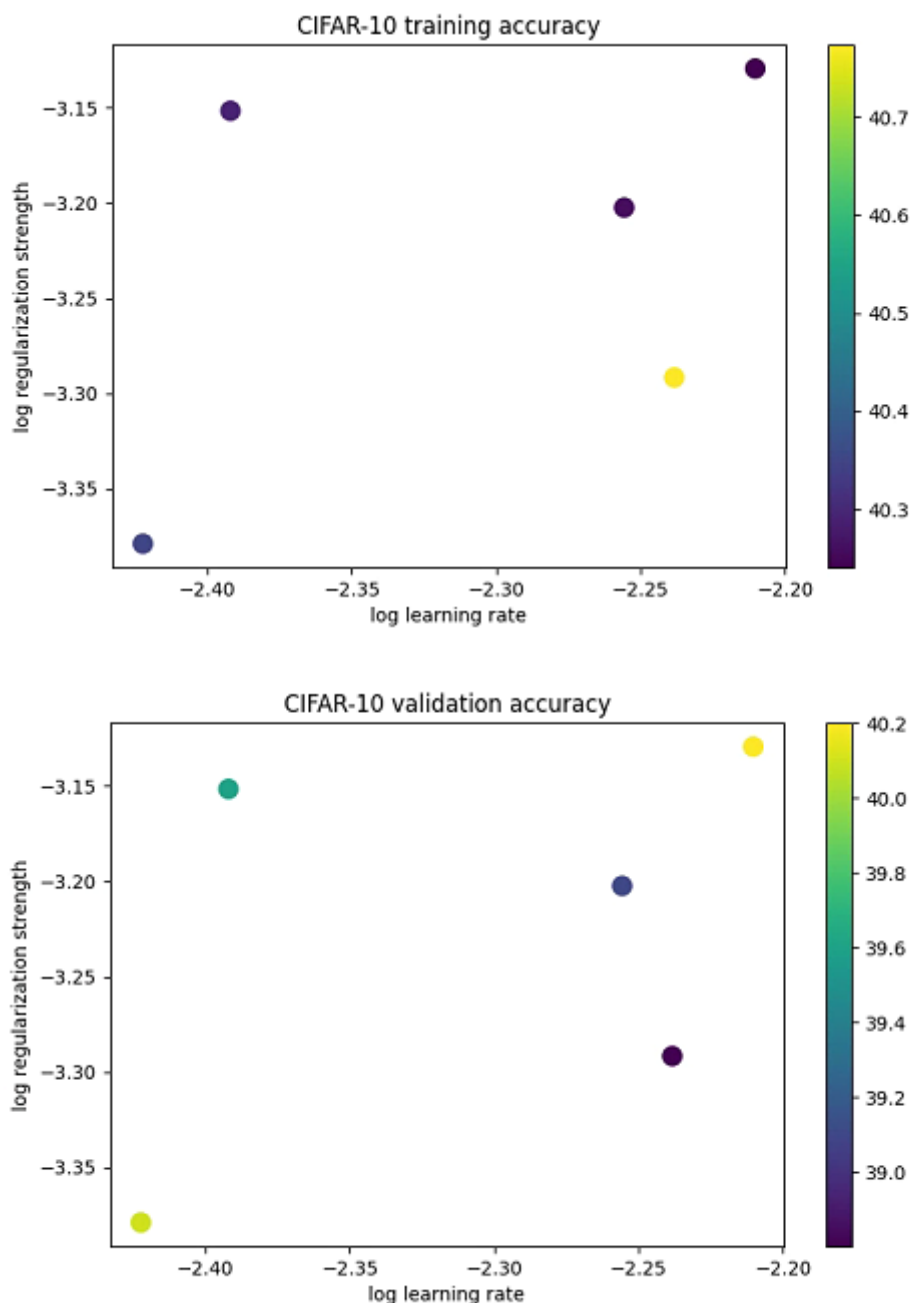
结果如下所示：

```
iteration 900 / 1500: Loss 9.000002
iteration 1000 / 1500: Loss 9.000006
iteration 1100 / 1500: Loss 9.000007
iteration 1200 / 1500: Loss 9.000021
iteration 1508 / 1500: Loss 9.808688
iteration 1408 / 1500: Loss 8.999991
That took 0.345687s
Accuracy: 0.1024
Precision: 0.3310
Recall: 0.3243
```

将Loss value和iteration关系作图，如下所示：



不同 learning-rate 下的 validation accuracy 和 train accuracy:



## 任务二：神经网络图像分类

- 数据：cifar10数据集
- 要求：构建一个包含两个线性层的前向神经网络模型进行数据分类，其中

隐藏层节点：100

激活函数：第一层，线性修正单元，第二层，softmax函数

损失函数：交叉熵函数

优化方法：分批随机梯度速降

- 1.画出神经网络结构示意图；
- 2.使用精确率和召回率评估分类性能；
- 3.调节算法某一参数（参数包括批大小，学习速率，隐藏层节点等），进行分类性能对比，画出性能随参数变化曲线。

# 1. 实验原理

## (1) 神经网络基本原理

神经网络是一种由神经元（或称为节点）组成的模型，通过学习权重和偏置参数来建模输入与输出之间的复杂关系。神经网络的基本结构包括输入层、隐藏层和输出层。每个隐藏层通常使用激活函数来引入非线性变换。

## (2) 交叉熵损失函数

交叉熵损失函数同样用于衡量模型输出和真实标签之间的差异，公式如下：

$$L(y, \hat{y}) = - \sum_i y_i \cdot \log(\hat{y}_i)$$

其中， $y$ 是真实的标签分布， $\hat{y}$ 是模型的输出概率分布。

## (3) 分批随机梯度下降法 (Mini-batch Stochastic Gradient Descent)

与任务一类似，使用分批随机梯度下降法进行优化。在每一步中，使用小批量的训练数据计算梯度，并更新模型参数，以减小损失函数。

## (4) 调节算法参数进行性能对比

可以调节以下算法参数来进行性能对比：

- 批大小 (Mini-batch size)：每次更新模型参数时使用的训练样本数量。
- 学习速率 (Learning rate)：每次梯度更新时参数应该移动的步长。
- 迭代轮数 (Number of iterations)：训练过程中数据遍历的次数。
- 隐藏层节点数 (Number of hidden layer nodes)：隐藏层中神经元的数量。

## (5) 性能评估

性能评估与任务一类似，使用精确率和召回率来评估分类性能。同时，可以通过混淆矩阵来更详细地分析模型的分类结果。

通过以上步骤，可以构建一个包含两个线性层的前向神经网络，使用分批随机梯度下降法进行训练，并通过精确率和召回率对其性能进行评估。通过调节算法参数，可以进行性能对比，找到最优的模型配置。

# 2. 实验结果

## (1) 画出神经网络结构示意图

构建一个包含两个线性层的前向神经网络，模型结构如下：

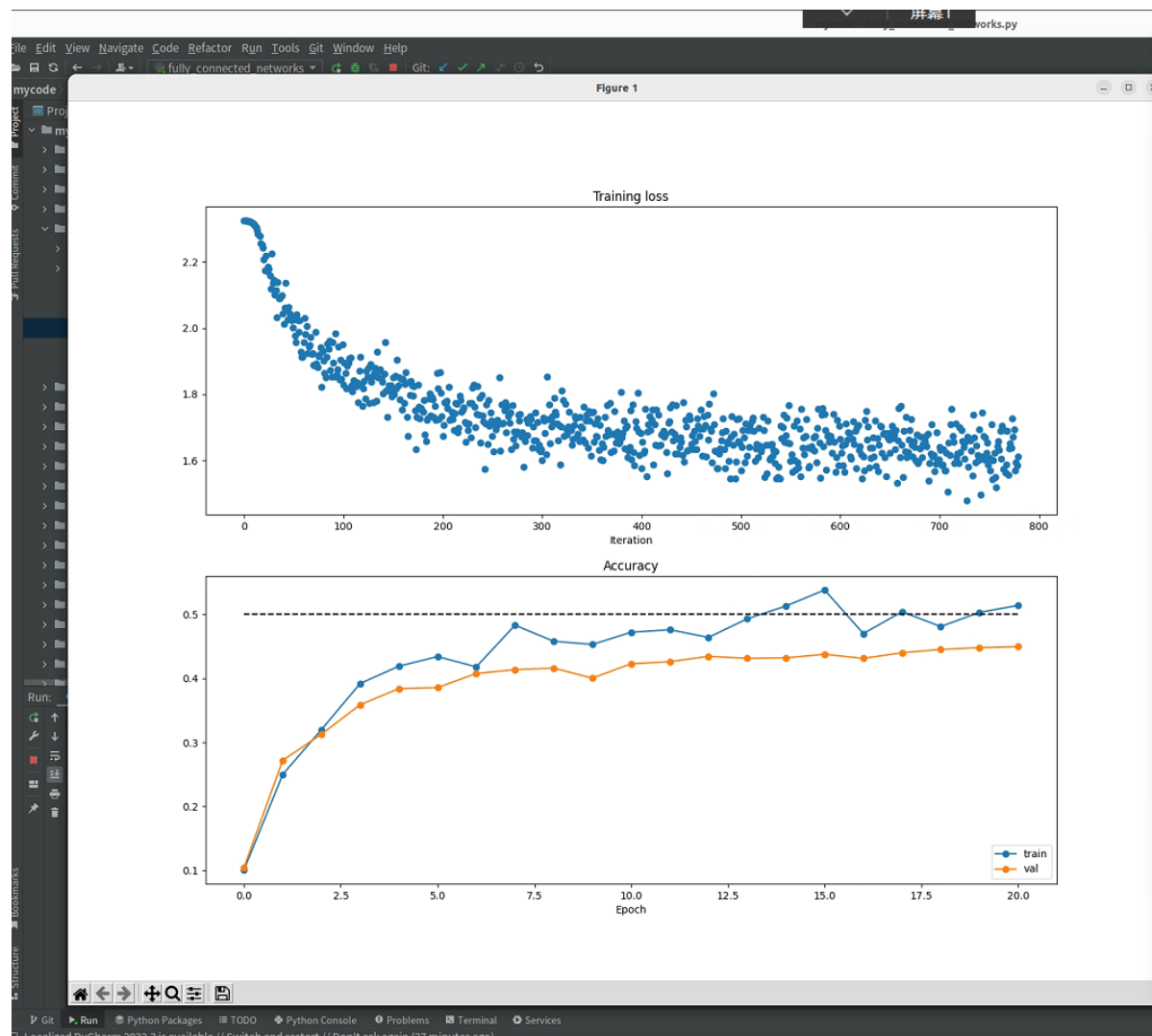
输入层（图像像素）-> [线性层1] -> [激活函数1] -> [线性层2] -> [Softmax归一化] -> 输出层（类别概率）

其中：

- 线性层1的输出通过ReLU (Rectified Linear Unit) 激活函数进行非线性变换。
- 线性层2的输出通过Softmax函数进行归一化，得到每个类别的概率分布。

## (2) 使用精确率和召回率评估分类性能

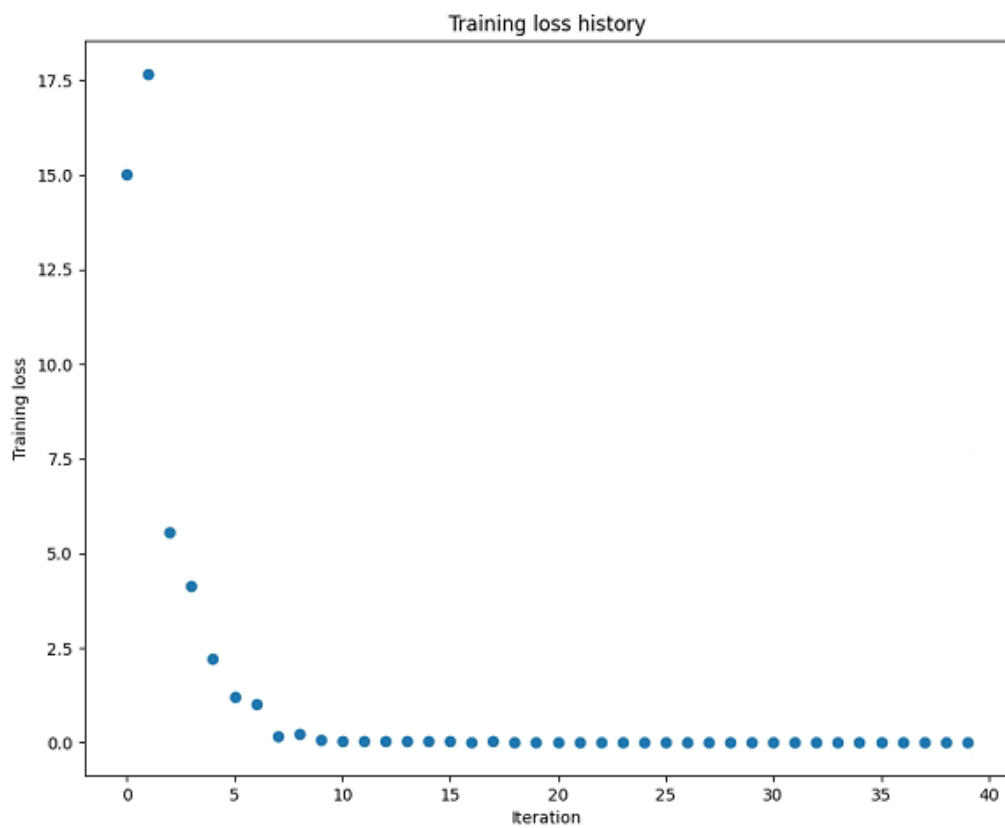
结果如下图所示：



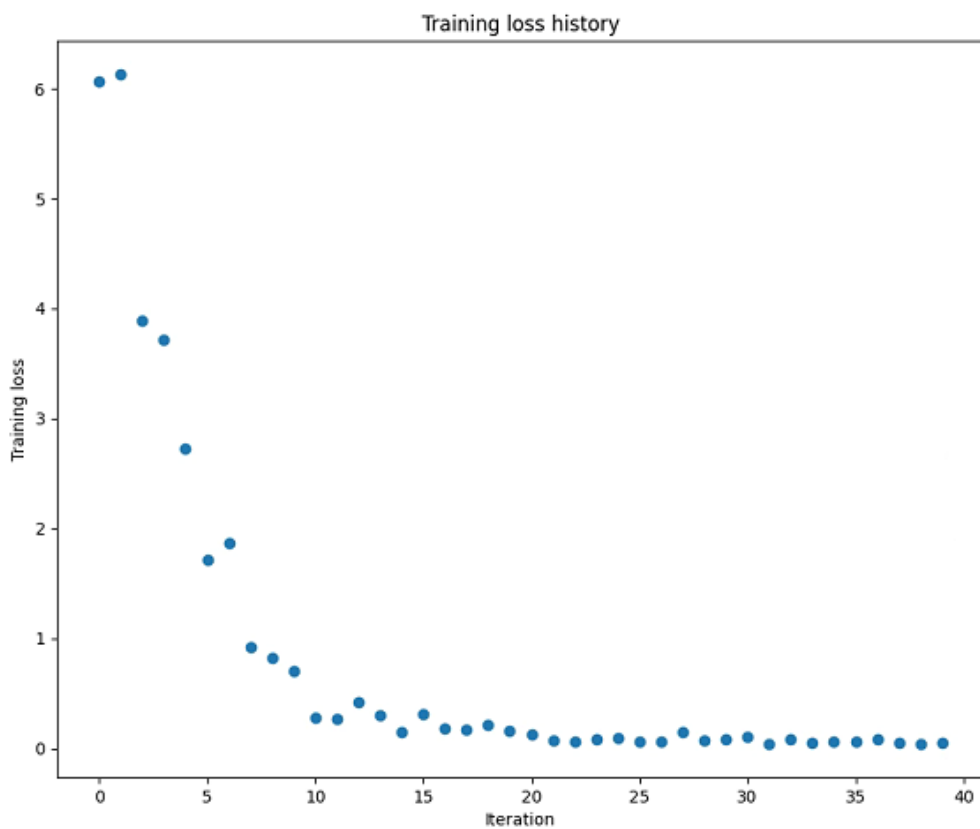
## (3) 调节算法某一参数，进行分类性能对比，画出性能随参数变化曲线。

改变网络层数：

每个隐藏层有100个节点的5层网络

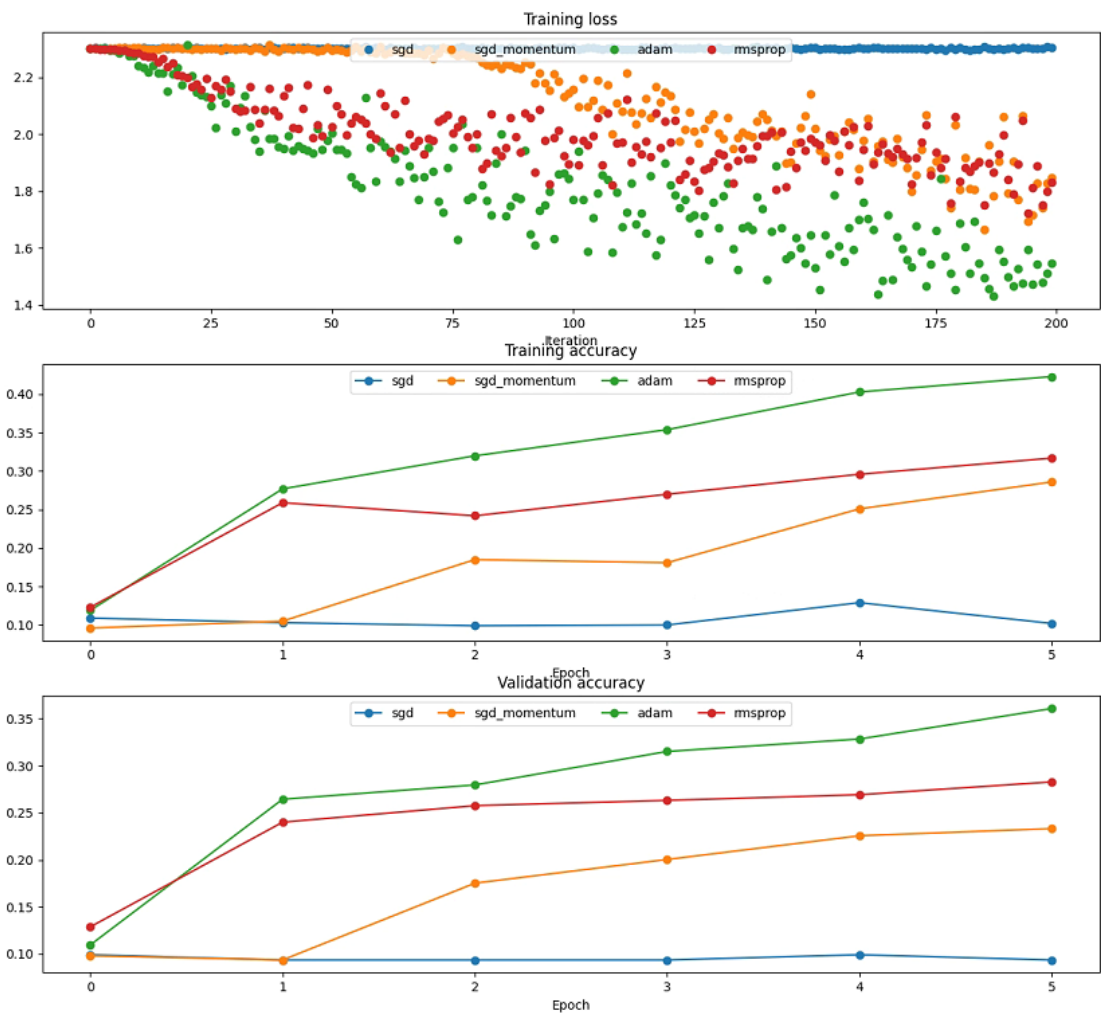


每个隐藏层有100个节点的3层网络



由上图可知，随着网络层数增加，模型准确度随着迭代次数的提高变快。

改变梯度下降算法：



由上图可知，rmsprop算法的效果最好。

### 改变迭代轮数见 (1)

由图可知，迭代次数增加，准确度变高，loss减小。