最终报告

张子扬 无06 2020010790

1. 模型改善方式
2. 将验证集数据并入训练集，并对图片进行随机裁剪进行数据增强；
3. 将BaseModel更换为自己编写的ResNet18网络；
4. 尝试使用其他更新学习率的函数。
5. 实验结果

本报告中的以下所有实验的实验环境设置如下表所示。

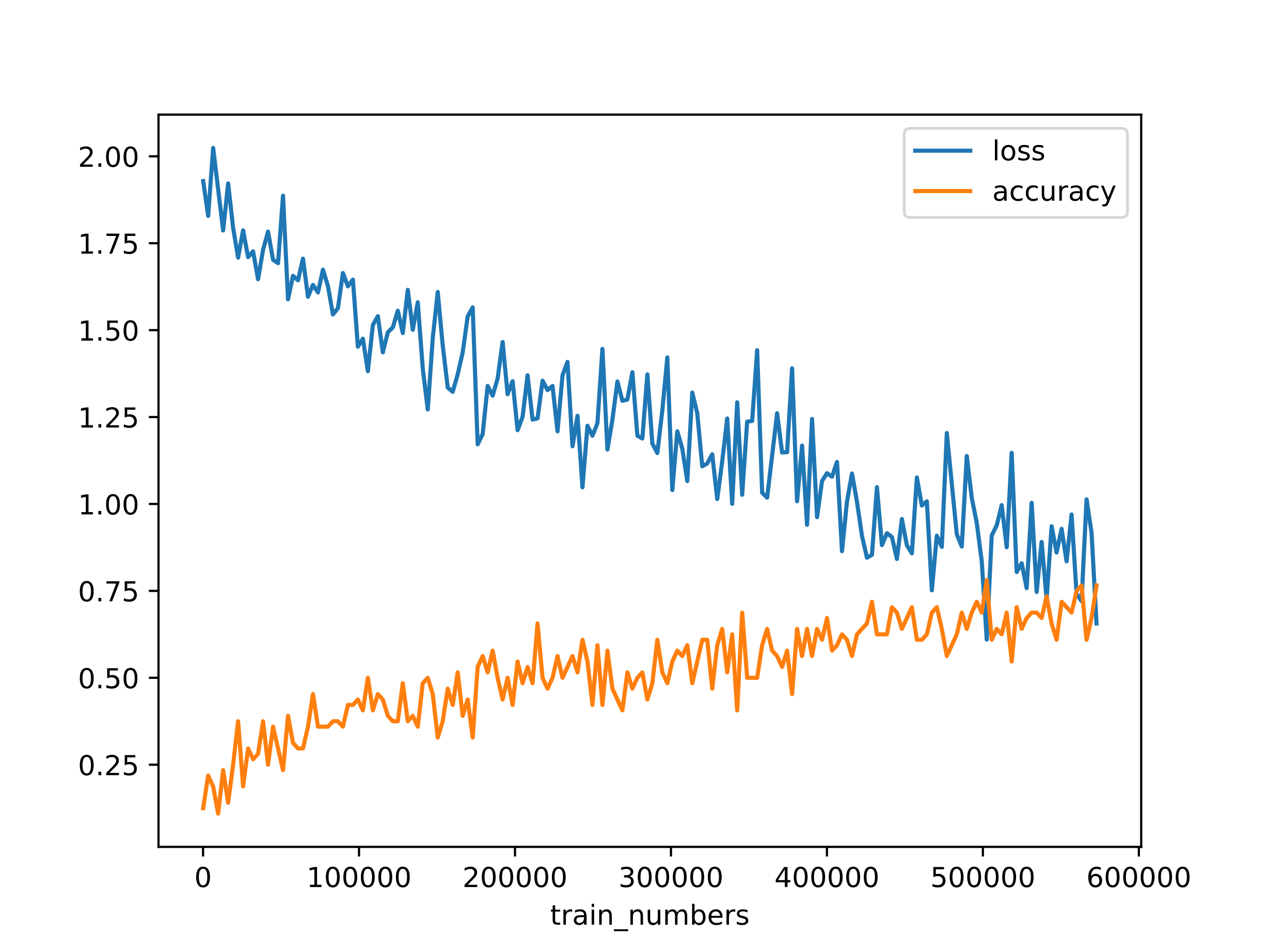
|  |  |
| --- | --- |
| 操作系统 | Linux Ubuntu |
| python版本 | 3.9.7 |
| CUDA Version | 11.2 |
| Driver Version | 460.39 |
| torch | 1.9.1+cu111 |
| torchvision | 0.10.1+cu111 |
| numpy | 1.19.5 |
| numpy-base | 1.20.3 |
| sklearn | 0.0 |
| matplotlib | 3.4.3 |

下面将展示5次实验以及调研结果。

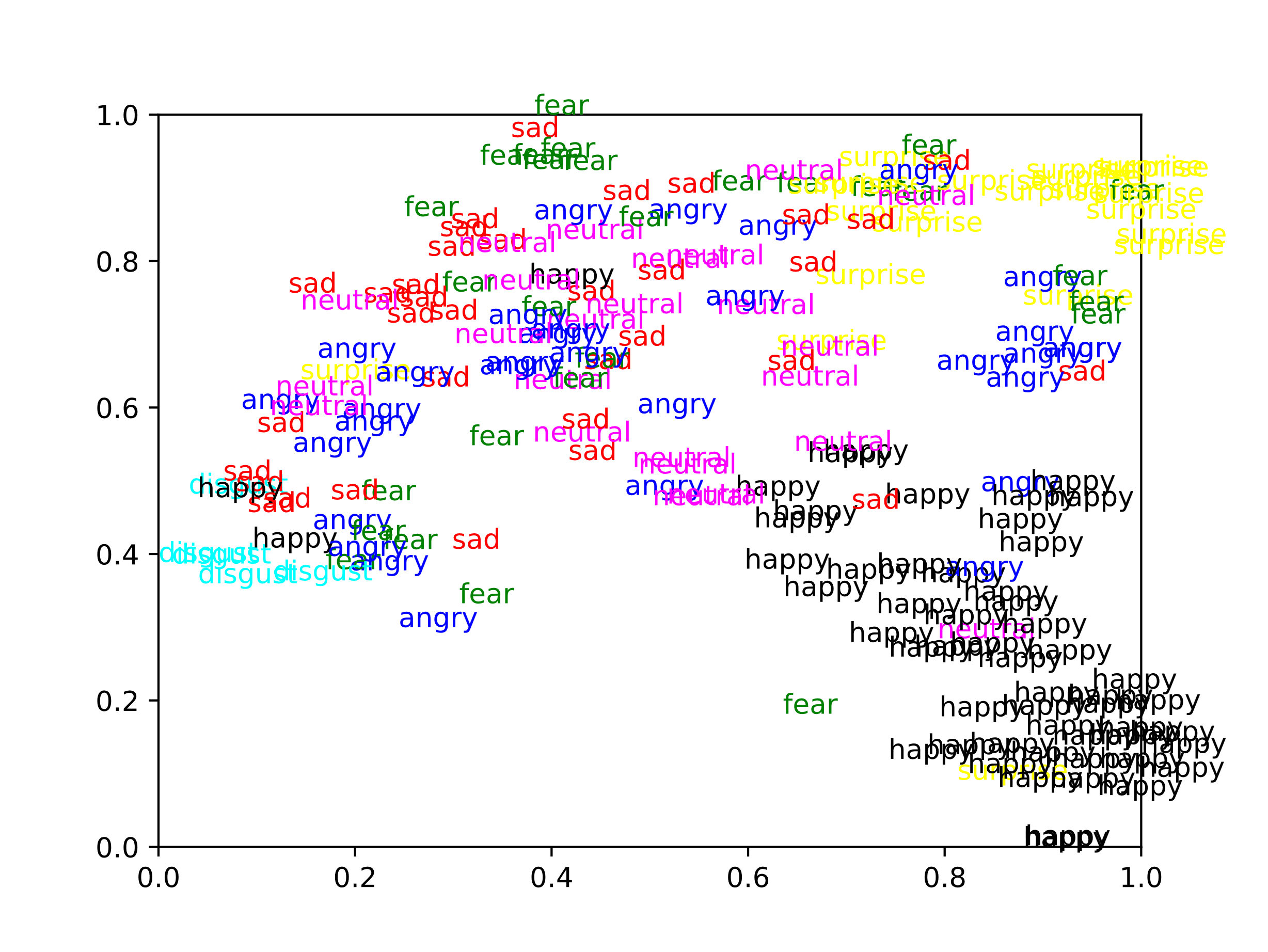
|  |  |
| --- | --- |
| image\_size | 48 48 |
| batch\_size | 64 |
| learning\_rate | 0.1 |
| weight\_decay | 5e-4 |
| epochs | 30 |
| milestones | 20 25 |

最终得到的测试集准确率是58.15%。

训练过程中的 loss 以及准确率的变化如下图所示。



样本的特征分布如下图所示。

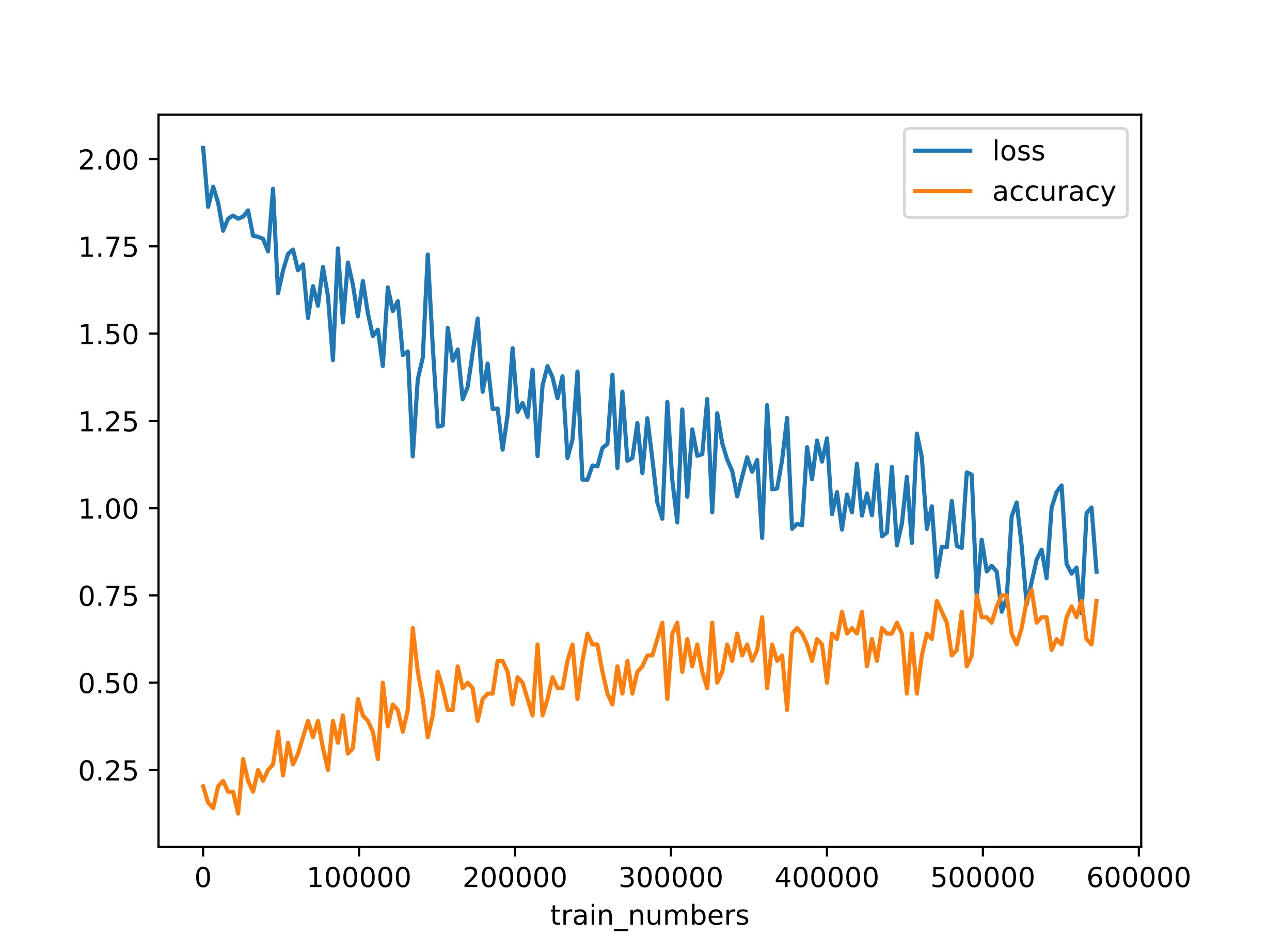


可见happy、disgust、surprise的特征分布较为良好，其余特征分布稍显分散。

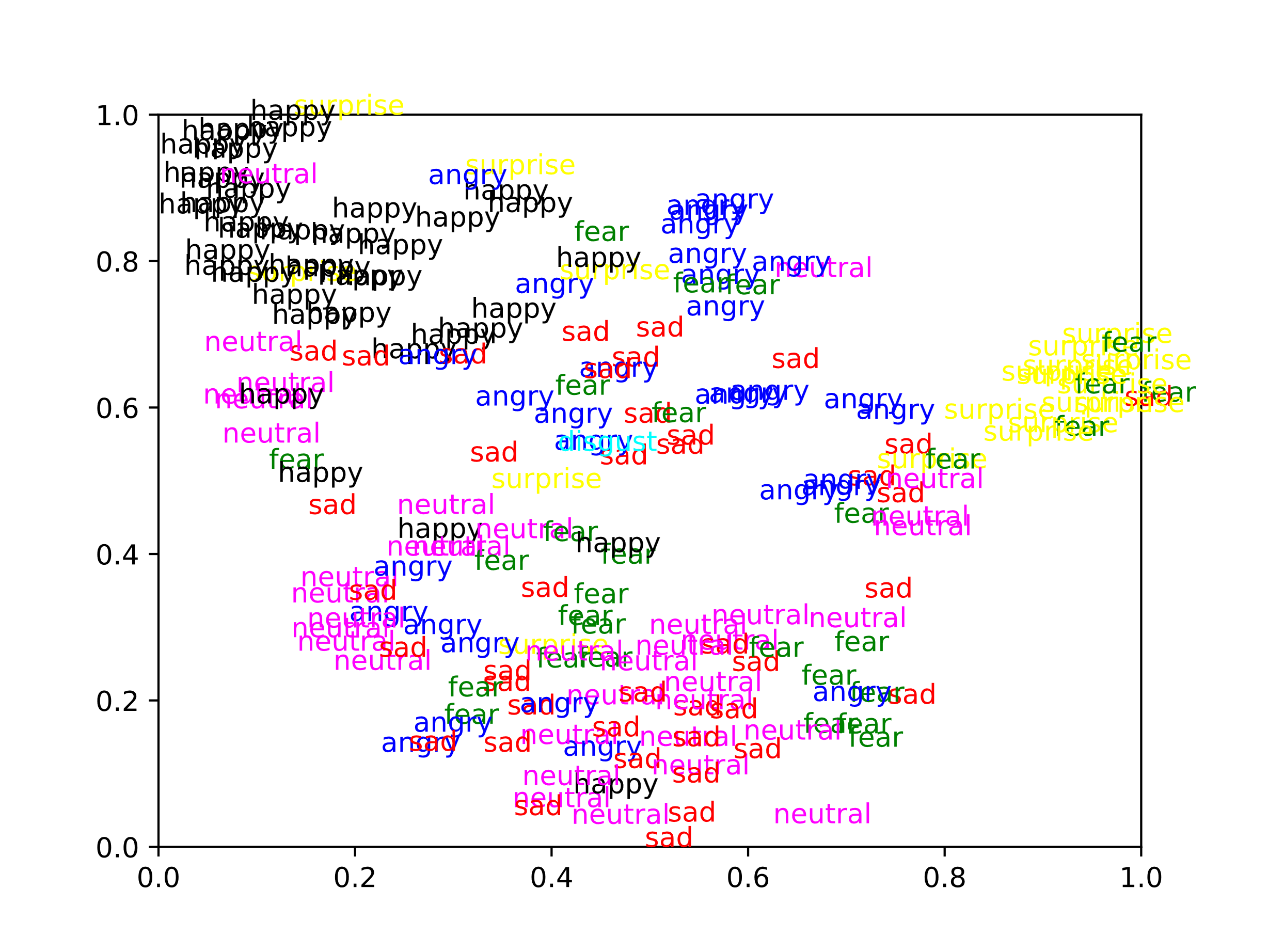
|  |  |
| --- | --- |
| image\_size | 48 48 |
| batch\_size | 64 |
| learning\_rate | 0.05 |
| weight\_decay | 5e-4 |
| epochs | 30 |
| milestones | 20 25 |

最终得到的测试集准确率是60.28%。

训练过程中的 loss 以及准确率的变化如下图所示。



样本的特征分布如下图所示。

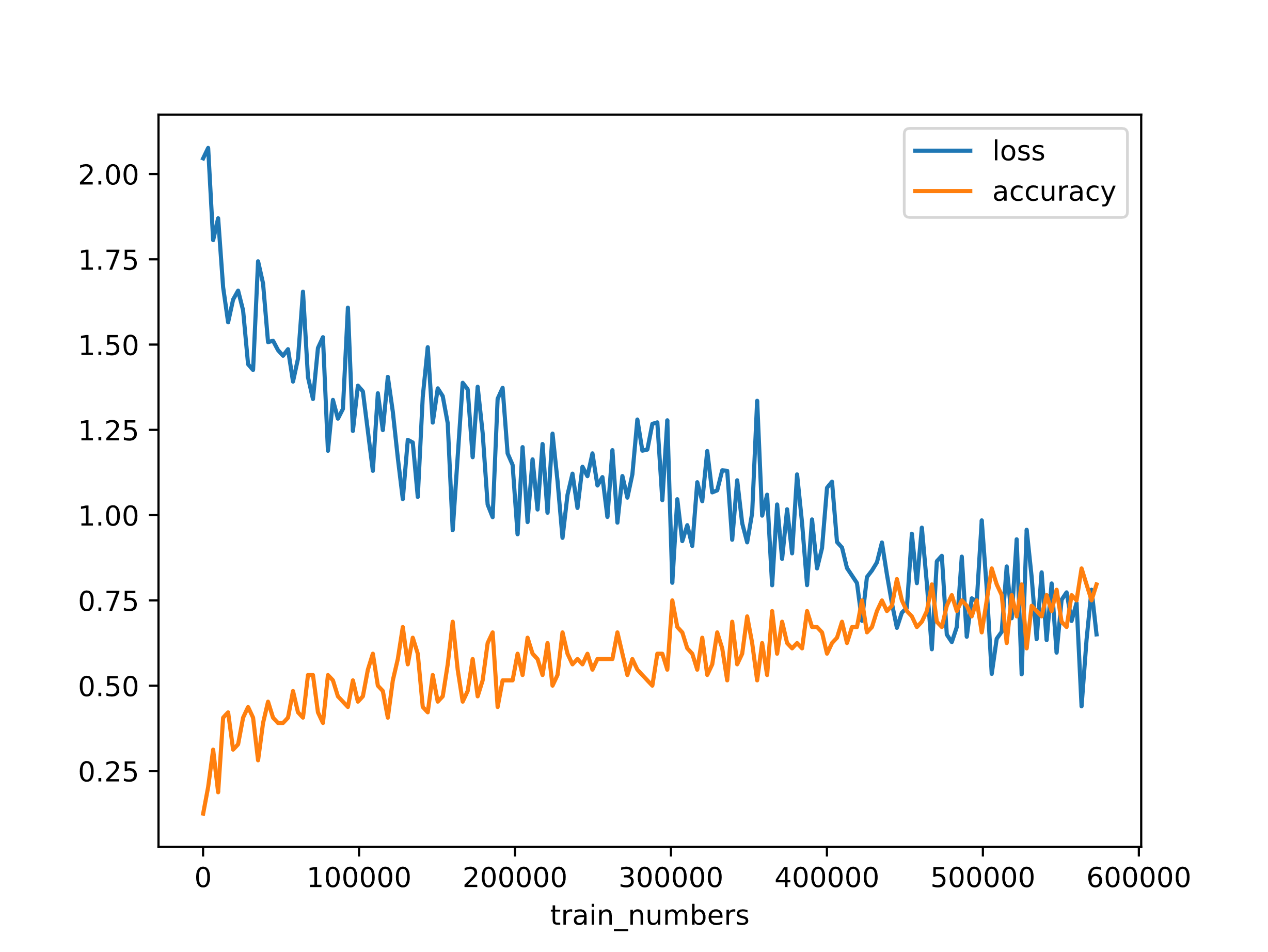


可见happy、surprise的特征分布较为良好，其余特征分布稍显分散。

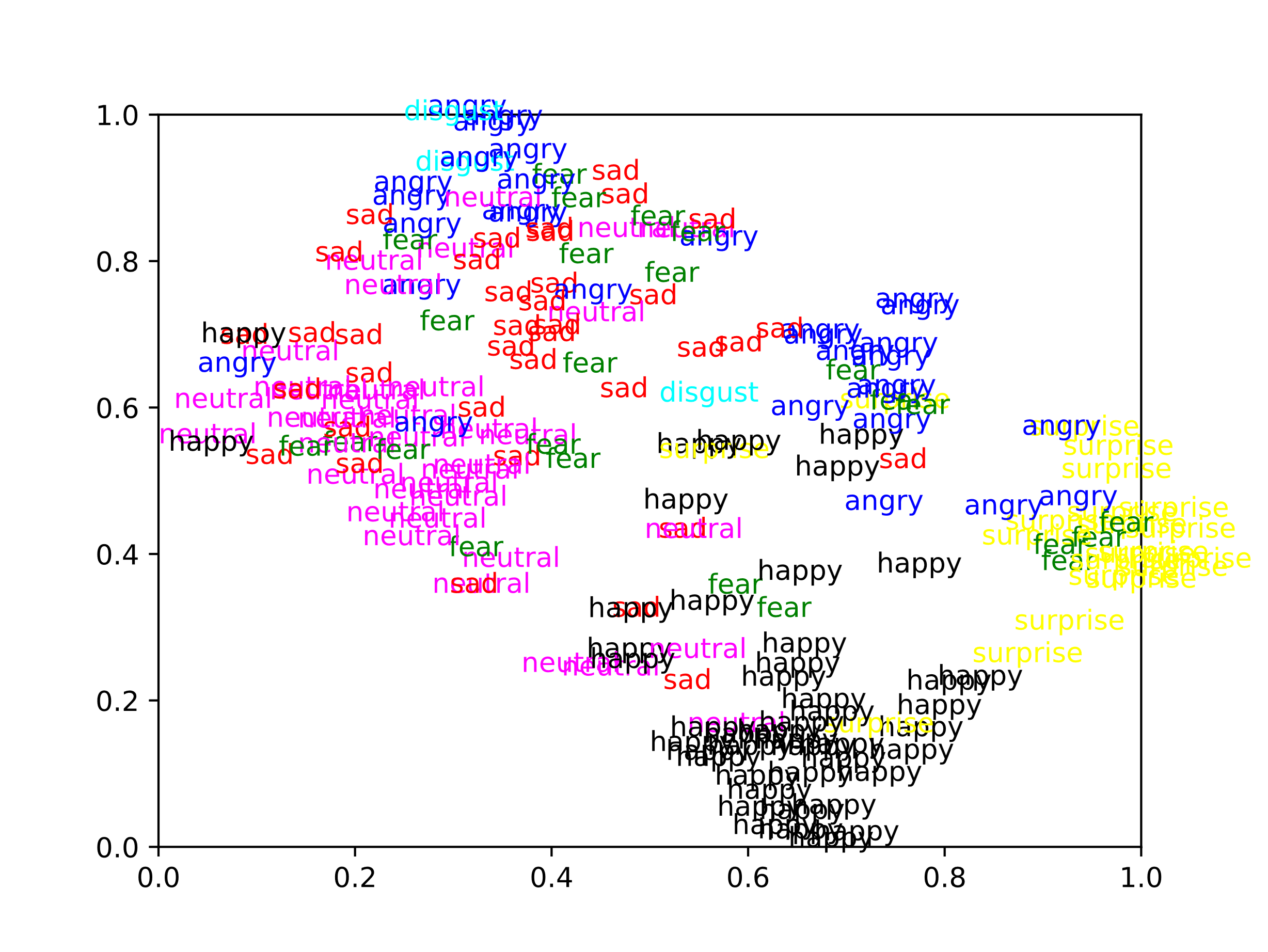
|  |  |
| --- | --- |
| image\_size | 48 48 |
| batch\_size | 64 |
| learning\_rate | 0.02 |
| weight\_decay | 5e-4 |
| epochs | 30 |
| milestones | 20 25 |

最终得到的测试集准确率是62.35%。

训练过程中的 loss 以及准确率的变化如下图所示。



样本的特征分布如下图所示。

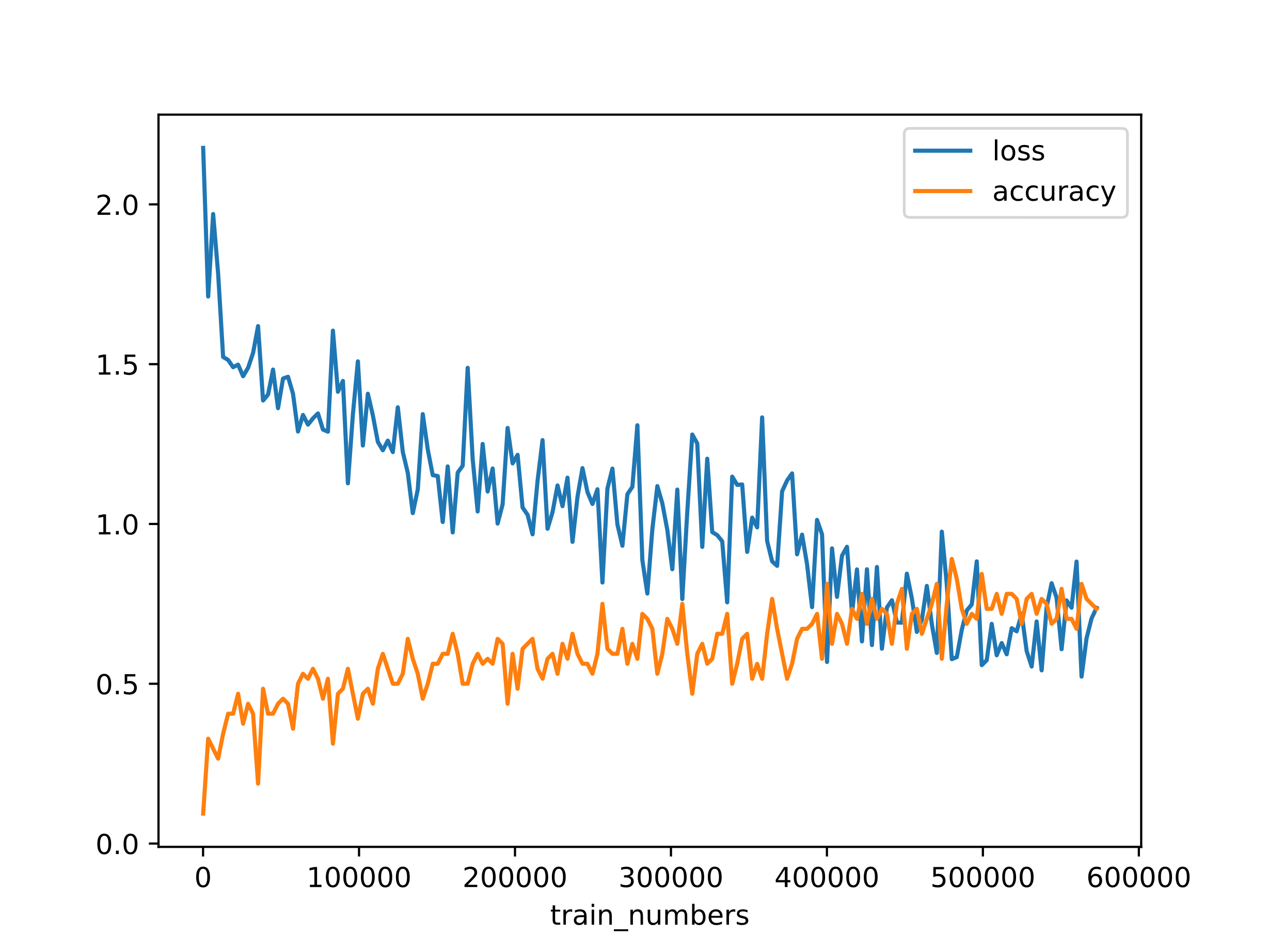


可见happy、surprise、neutral、sad、disgust的特征分布均较为良好，angry与fear的特征分布稍显分散。

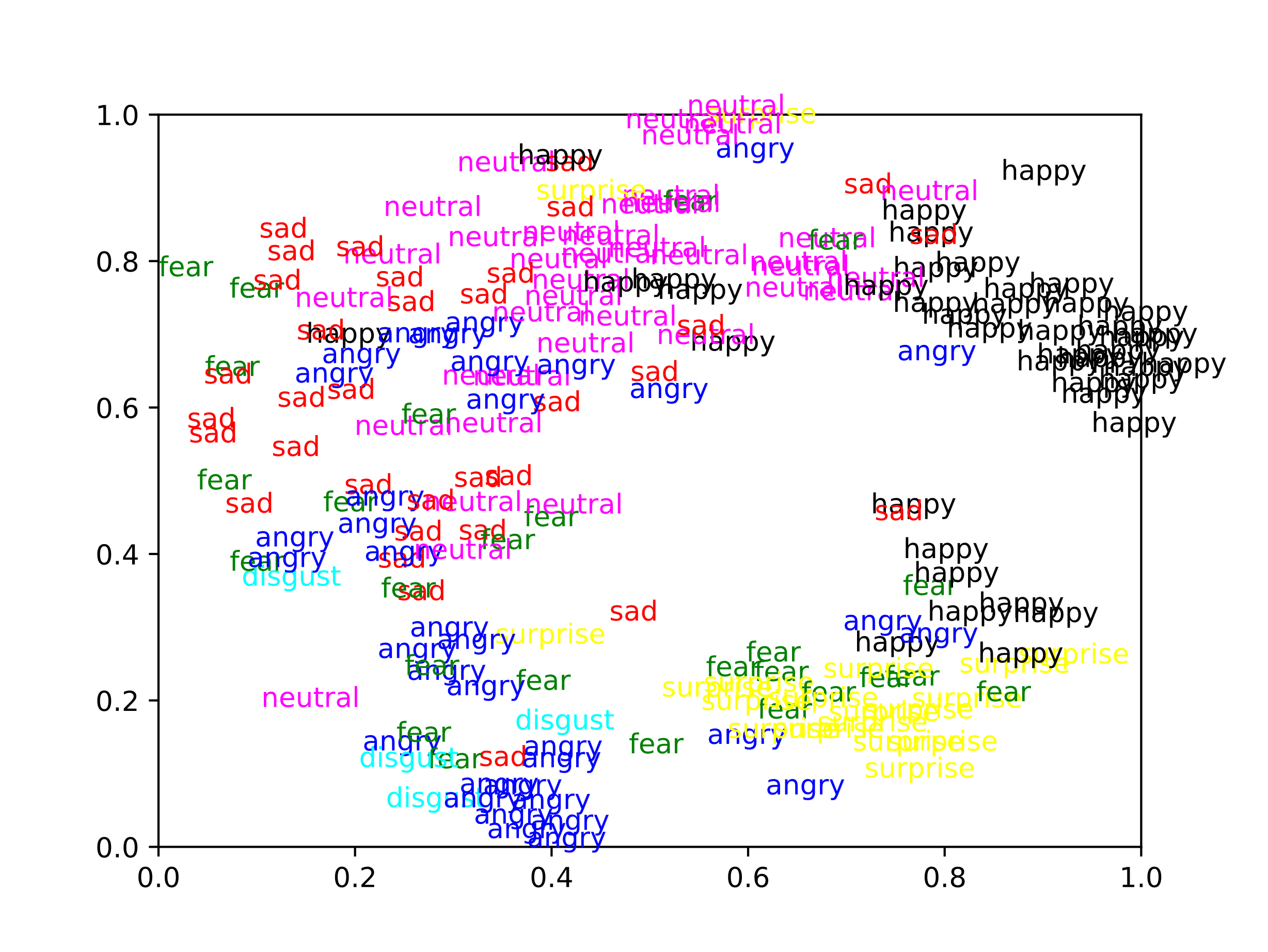
使用StepLR函数更新学习率，设置为每10个周期学习率乘以0.5，其余超参数如下表所示：

|  |  |
| --- | --- |
| image\_size | 48 48 |
| batch\_size | 64 |
| learning\_rate | 0.02 |
| weight\_decay | 5e-4 |
| epochs | 30 |

最终得到的测试集准确率是60.68%。



样本的特征分布如下图所示。

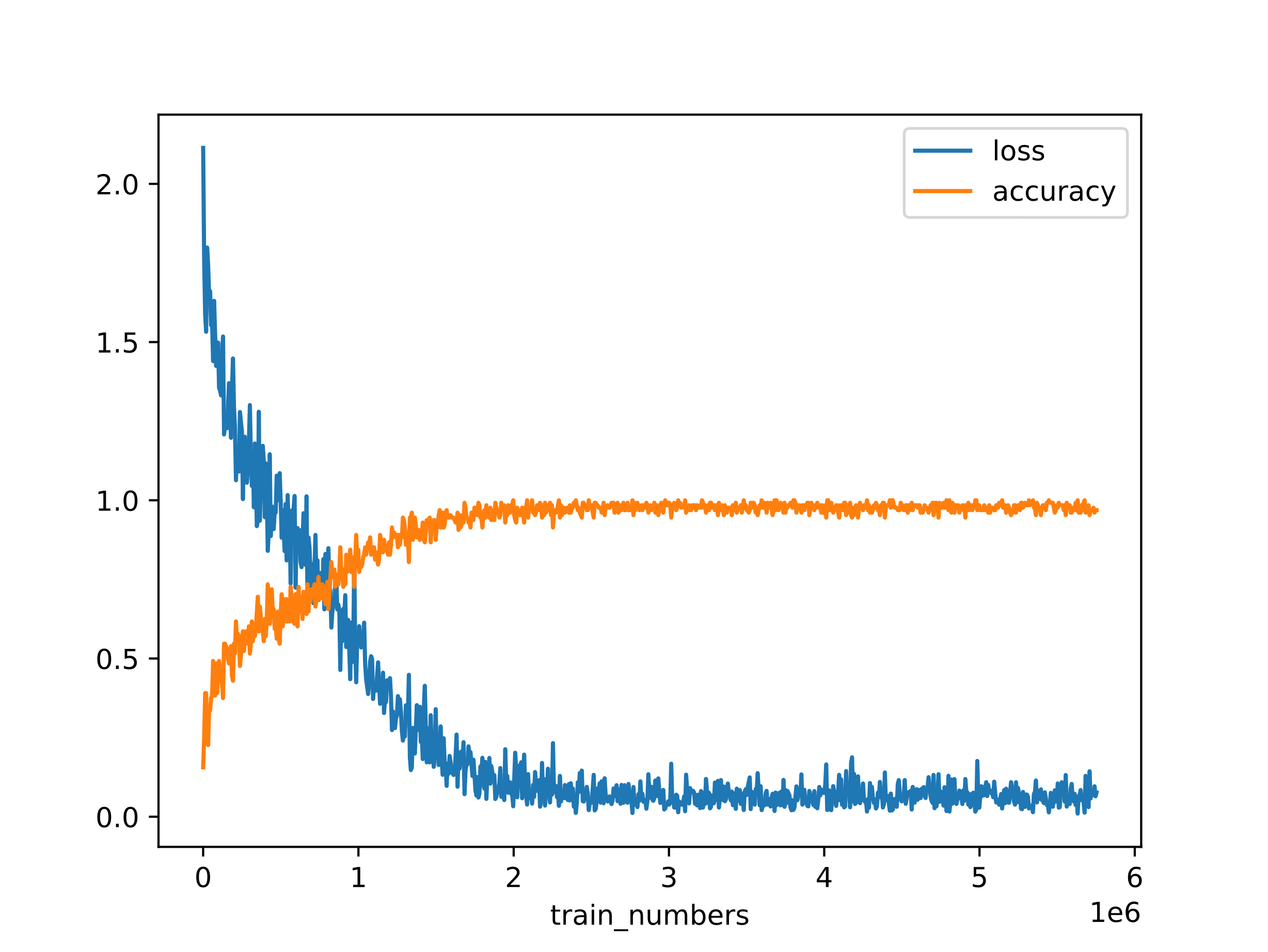


可见happy、surprise、neutral的特征分布较为良好，其余特征分布稍显分散。

使用ReduceLROnPlateau函数更新学习率，该函数的作用是动态调整学习率，我的设置是当测试集准确率不再上升的时候即减小学习率。其余超参数如下表所示：

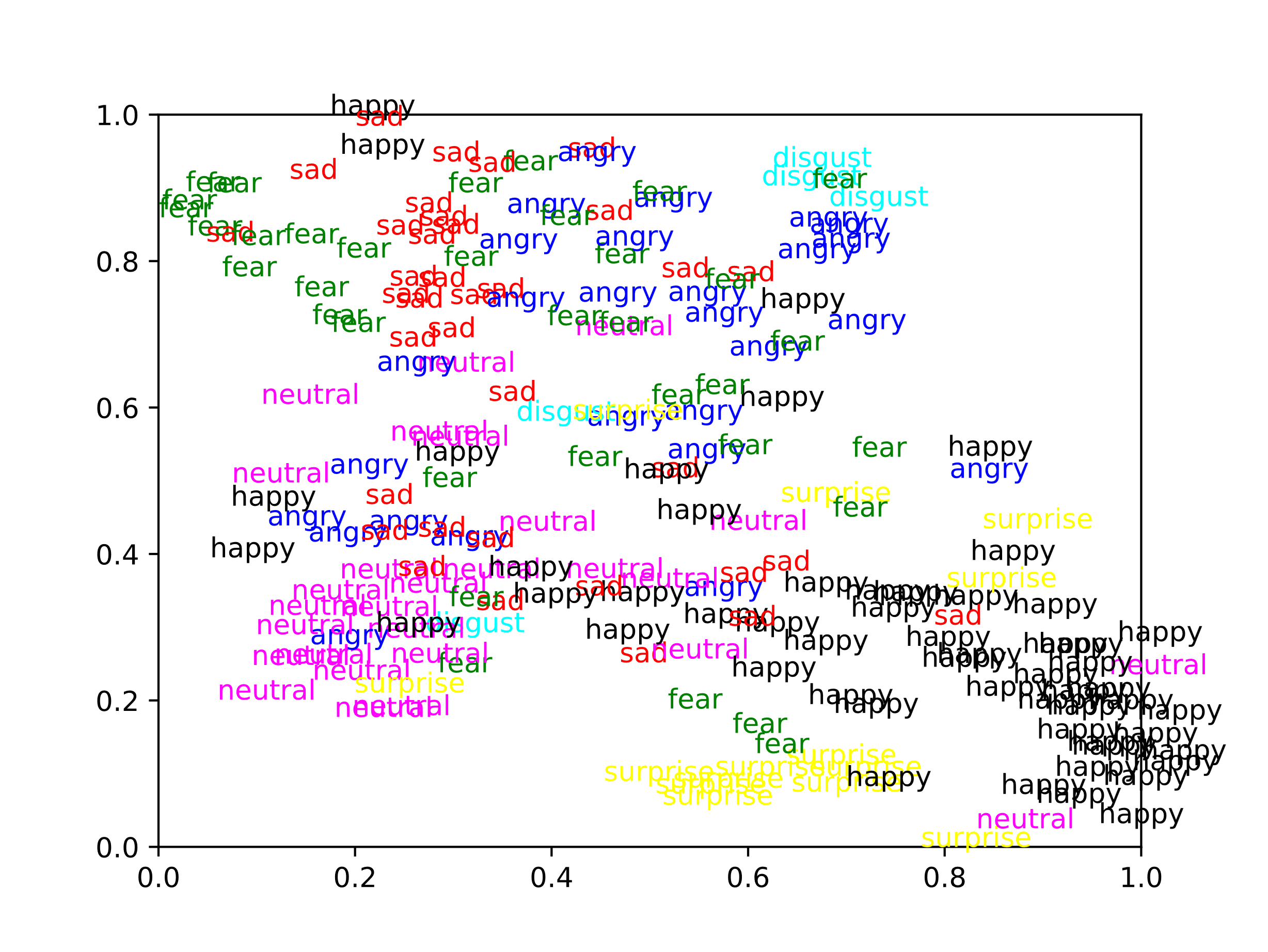
|  |  |
| --- | --- |
| image\_size | 48 48 |
| batch\_size | 128 |
| learning\_rate | 0.1 |
| weight\_decay | 1e-4 |
| epochs | 300 |

最终得到的测试集准确率是61.49%。



由上图可以看出最终训练集准确率逼近100%，loss趋于0，但是测试集准确率并没有提升，可见模型最终出现了过拟合。

样本的特征分布如下图所示。



可见happy、neutral的特征分布较为良好，其余特征分布稍显分散。

1. 实验总结

总结以上5次具有代表性的实验，不难看出模型的测试集准确率卡在60%左右无法突破。这不禁让人想要探究一下：在不使用其他数据集数据的情况下，在该数据集上最高能够达到多大的准确率呢？我在<https://paperswithcode.com/sota/facial-expression-recognition-on-fer2013>上找到了该数据集的SOTA，惊奇地发现只有76.82%（不过我想是这是因为没有人去刷这个数据集的准确率）。其中没有使用其他数据集、且只使用了CNN架构的SOTA为73.70%。我仔细阅读了作者的代码（见<https://github.com/LetheSec/Fer2013-Facial-Emotion-Recognition-Pytorch>），发现作者使用的也是ResNet18网络，最终通过各种trick把测试集准确率调到了73.70%。经过对比，我发现自己的数据增强较为简陋，对各种trick也知之甚少，最终只能感叹自己调参的水平还有待提高呀！