数据科学与工程学院

统计方法与机器学习课程实验报告

统计方法与机器学习课程 Final Project：人脸识别

---

github.com/QiushiSun

2021 年 6 月 14 日

摘要

人脸识别是较为常见的机器学习在生物特征识别中的应用。本次实验的数据集为卡内基 · 梅隆大学的 Face Images Data Set，其中包含 640 张由 20 个人的不同表情构成的人脸照片。在实验过程中，采取神经网络算法进行人脸分类识别，再分别使用神经网络算法和聚类算法完成面部特征发现。本实验在 Python3.7(Jupyter-Lab) 环境下进行，并且对一些重要指标与分类精度进行了可视化展示。本报告使用 LaTeX 进行排版，本次实验所用的代码全部保存在 *Project3-code.ipynb/pdf* 文件中，本报告中主要展示可视化分析结果和模型性能。

关键词: 神经网络，聚类，人脸识别，表情识别

目录

第 1 章　项目概述与数据概览

本实验的目标在于使用神经网络算法对卡内基 · 梅隆大学的 Face Images Data Set 数据集进行人脸识别分类，随后再分别使用聚类算法和神经网络算法发现相似表情，并比较二者性能。

1.1　神经网络算法

本次实验仅采用一个简单的全连接神经网络结构，如图 1.1 所示

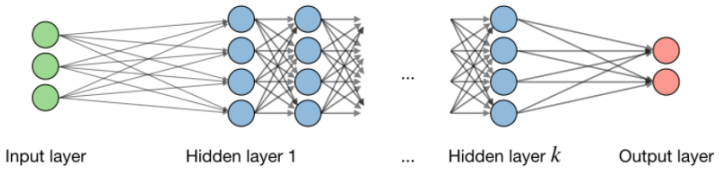

图 1.1: Simple Fully Connected Neural Net

该网络拥有：

- 输入层（Input Layer）：外界信息输入，不进行任何计算，仅向下一层节点传递数据
- 隐层（Hidden Layer）：接收上一层节点（可以是输入层，也可以是隐层）的输入，进行隐层内的计算，并将计算结果传到下一层
- 输出层（Output Layer）：接收上一层节点的输入，进行计算，并将结果输出（如概率形式的分类结果）

在人脸识别和表情分类问题上，主要是修改输出层的结构和对应标签，此部分详见代码

1.2　K-means 聚类算法

在相似表情发现问题上，除了神经网络算法外，我们还采用 K-means 聚类（K-means clustering）算法。K-means 是我们最常用的基于欧式距离的聚类算法，算法流程如图 1.2 所示，将数据（人脸）视作空间中随机分布的点，计算其与质心（Centroid）的距离来作为分类标准，再反复迭代直至收敛。

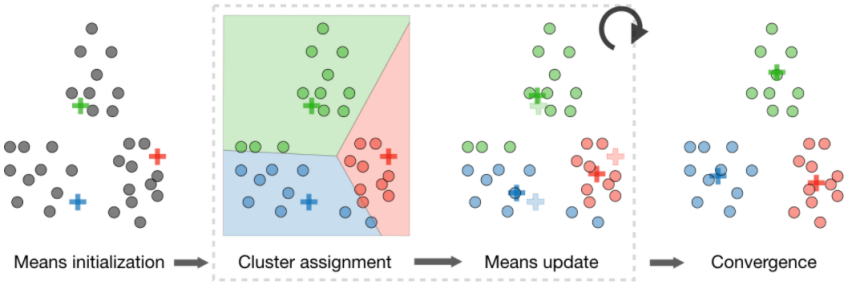

图 1.2: K-means Algorithm

1.3 CMU Face Images Data Set 数据集

本次实验所用的数据集为卡内基 · 梅隆大学 Face Images Data Set 数据集，数据集内包含 20 个不同的人且表情不同的人脸数据，共 300 条训练集，324 条测试集。初看觉得这个数据量非常的少，不仅类别数少，条目数量也非常少，但是要注意到这是 1999 年的数据集，对于那个年代的机器学习任务来说，这个数量级其实已经算是比较大了。

值得我们注意的是，这份年代久远的数据集使用的数据格式为 *.pgm* 格式，PGM 是 Portable Gray Map 的缩写。它是灰度图像格式中一种最简单的格式标准，在使用 python 读取时需要一些预处理。

下图为该数据集的人脸数据展示（读取数据部分请见代码），这是 Andrew Ng 20 年前的照片，在该数据集中被标注为 an2i，具体信息和该数据集由来可以看这篇文章 Andrew Ng in CMU Face Images。

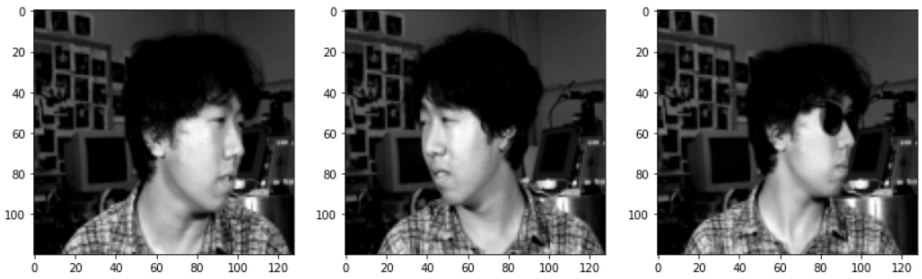

图 1.3: CMU Face Images Data Set

该数据集可在卡内基 · 梅隆大学计算机系主页的以下地址获得:
http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/faces.html
(也可以从 UCI Machine Learning Repository 下载)

Remark：该数据集有两个版本，一个是完整版的原始黑白图像数据，一个是 one-quarter size 的压缩版数据集，二者在数据处理上没有太大的差别，本次实验采用了后者作为数据源。

第 2 章　算法

2.1　神经网络算法流程

该算法的损失函数定义为：

$$C\left(w, b = \frac{1}{2n}\sum_{x}\|y(x) - a\|^2\right) \tag{2.1}$$

如公式 2.1 所示，使用均方误差（MSE）损失函数，其中可学习参数 w 代表权重向量，n 为训练集数据个数，a 表示输入为 x 时输出的向量。

2.1.1　梯度下降算法

梯度下降算法是常用的学习策略，如图 2.1 和公式 2.3 所示

$$w \longleftarrow w - \alpha\frac{\partial L(z, y)}{\partial w} \tag{2.2}$$

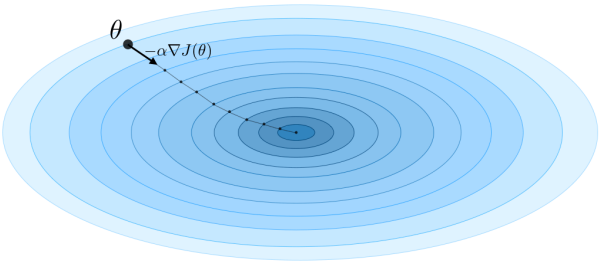

图 2.1: Gradient Descent

梯度下降算法的更新规则如公式 2.3 所示。

$$\begin{aligned} w_k' &= w_k - \eta\frac{\partial C}{\partial w_k} \\ b_l' &= b_l - \eta\frac{\partial C}{\partial b_l} \end{aligned} \tag{2.3}$$

在实际使用中我们一般不使用原始的梯度下降法，而是 *mini-batch* 梯度下降算法，如公式 2.4 所示

$$\begin{aligned} w_k' &= w_k - \eta/m\frac{\partial C_{X_j}}{\partial w_k} \\ b_l' &= b_l - \eta/m\frac{\partial C_{X_j}}{\partial b_l} \end{aligned} \tag{2.4}$$

2.1.2 激活函数、优化方法与权重初始化

激活函数采取传统的 Sigmoid 激活函数

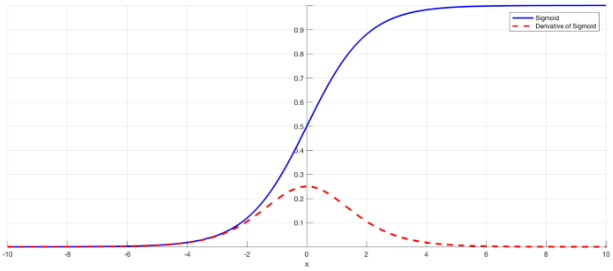

图 2.2: Sigmoid

在反向传播的梯度计算中，其导数可以用其自身表示，如公式 2.5 所示

$$S'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = S(x)(1-S(x)) \tag{2.5}$$

优化方法采取 Nesterov 动量方法，如公式 2.6 所示，动量方法帮助通过局部极小点 (Local minima) 和鞍点 (Saddle point)，并加神经网络的快收敛速度；

$$\begin{aligned} v_{t+1} &= \rho v_t - \alpha \nabla f\left(\tilde{x}_t\right) \\ \tilde{x}_{t+1} &= \tilde{x}_t - \rho v_t + (1+\rho) v_{t+1} \\ &= \tilde{x}_t + v_{t+1} + \rho\left(v_{t+1} - v_t\right) \end{aligned} \tag{2.6}$$

权重初始化方法采用 Xavier 初始化，这种方法的优点为保证前向传播和反向传播时每一层的方差一致，推导如公式 2.7 所示，详细解释可参考 Stanford cs231n 课程的 Weight Initialization 章节。

$$\begin{aligned} \operatorname{Var}\left(y_i\right) &= \operatorname{Din}^* \operatorname{Var}\left(x_i w_i\right) \\ &= \operatorname{Din}^*\left(E\left[x_i^2\right] E\left[w_i^2\right] - E\left[x_i\right]^2 E\left[w_i\right]^2\right) \\ &= \operatorname{Din}^* \operatorname{Var}\left(x_i\right)^* \operatorname{Var}\left(w_i\right) \end{aligned} \tag{2.7}$$

2.2 K-means 聚类算法流程

K-means 聚类（k-means clustering）算法是一种易于理解的无监督学习算法。

随机初始化聚类中心 $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ （个数依任务而定，如本次相似表情发现中的表情个数），然后反复执行公式 2.8 和公式 2.9 直至收敛为止。

$$c^{(i)} = \arg\min_j \left\| x^{(i)} - \mu_j \right\|^2 \tag{2.8}$$

$$\mu_j = \frac{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^{m} 1_{\{c^{(i)}=j\}}} \tag{2.9}$$

以公式 2.10 的数值收敛为迭代终止条件。

$$J(c,\mu) = \sum_{i=1}^{m} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2 \tag{2.10}$$

第 3 章　实验过程与结果

本实验的过程主要包括数据预处理，训练模型、结果分析和数据的可视化展示等几个主要步骤。这部分代码请见 *Project3-code.ipynb* 文件。

3.1　人脸识别任务

虽然在本章节的实验中同时使用聚类算法和神经网络算法，但是因为人脸识别任务是 20 分类，在如此少的训练集数据下使用 Kmeans 聚类显然无法获得好效果，所以对于人脸识别任务仅使用神经网络，在后续的实验中再加入聚类算法。

由于网络的搭建过程代码比较冗长，请直接参考附带的 *Project3-code.ipynb/pdf* 文件（我已将 ipynb 转换为 pdf 格式方便查看），在本报告中主要展示可视化分析结果以及模型性能。

如图 3.1 所示，在训练过程中训练集能随迭代轮次的增加而完全拟合，测试集损失也能快速下降到收敛，

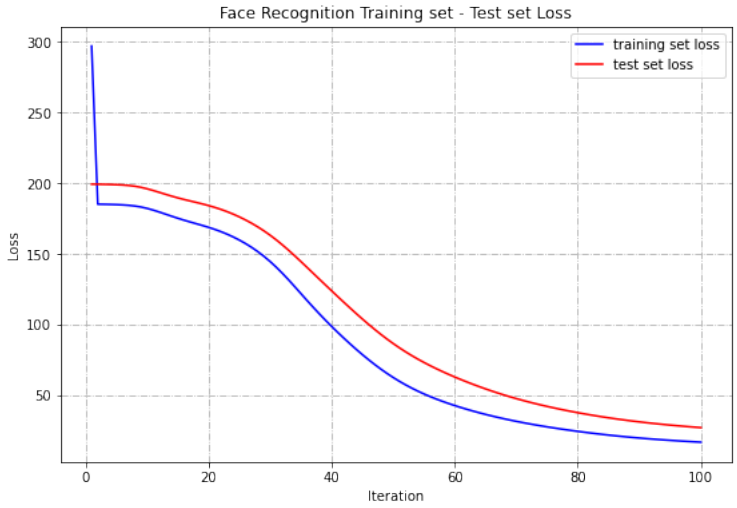

图 3.1: 人脸识别任务损失下降曲线

训练过程中，训练集和测试集的准确率如图 3.2 所示，训练起初有一些小的波动，但随后测试集准确率一直上升到收敛，最终性能为训练集准确率: 0.996，测试集准确率为 0.981。

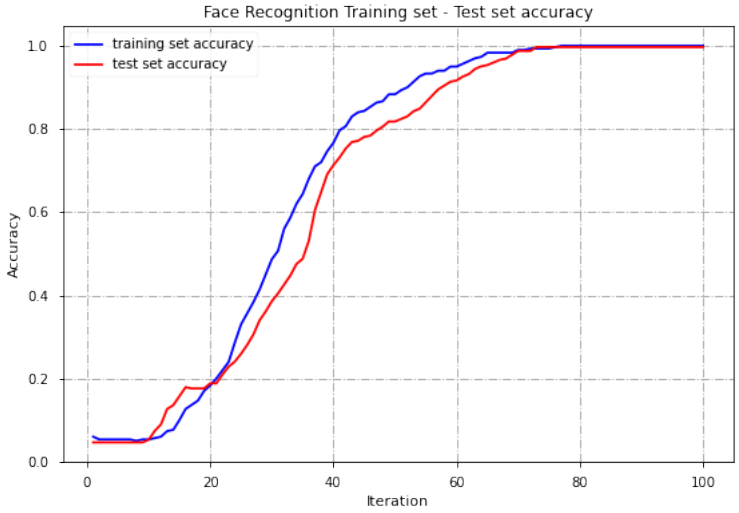

图 3.2: 人脸识别任务准确率曲线

3.2 相似人脸特征发现

该数据集中的人脸有多种特征，如人的情绪，人脸朝向，是否带墨镜等。经实验，两种算法在人脸情绪识别上的表现非常不理想，对比两种算法的性能没有太大的意义。算法在人脸情绪识别上的表现不佳的可能原因在本章节末尾做出了解释，此部分的解释请见 3.2.3 节，在此使用聚类算法表现稍好的人脸朝向来进行与神经网络算法性能的对比。

3.2.1 使用聚类算法进行相似人脸特征发现

我们先使用 Kmeans 算法对该问题进行聚类，设置四个聚类中心，对单个簇挑选十个样本进行可视化展示，如图 3.3 和 3.4 所示

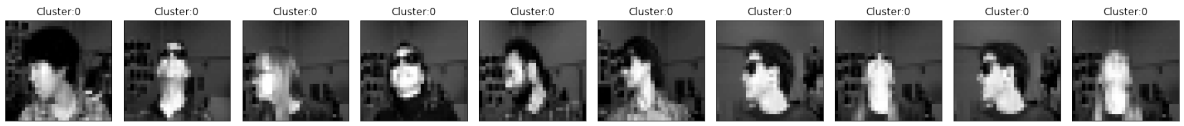

图 3.3: Face Direction Cluster 0

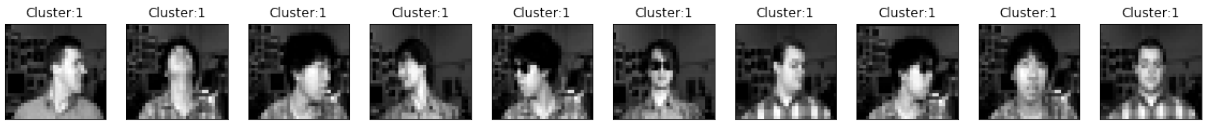

图 3.4: Face Direction Cluster1

在上述两张图中，可以明显看到第 0 个聚类簇重的抽样主要是向左侧的面部表情，

而第 1 个聚类簇重的抽样则没有明显特征，接下来对这些聚类簇的全部数据的分布进行分析

绘制第 0 个聚类簇中的分布柱状图，如图 3.5 所示可以看到该聚类簇可视为对于 left 这个方向的分类，准确度可认为有 59%

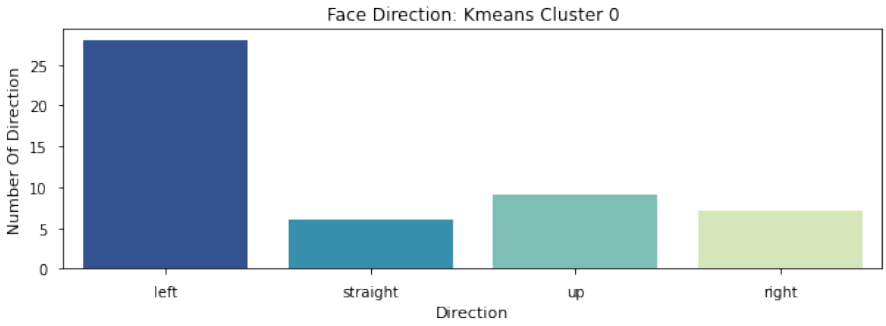

图 3.5: Face Direction Cluster0

绘制第 1 个聚类簇中的分布柱状图，如图 3.6 所示可以看到该聚类簇的分类比较均匀，没有明显按照脸部朝向分类的迹象，准确度不佳。

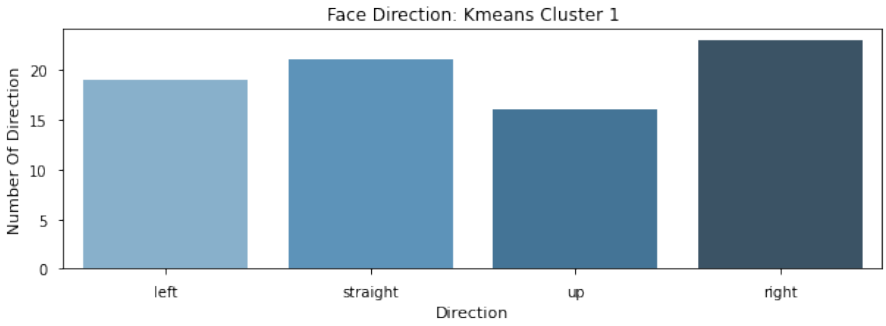

图 3.6: Face Direction Cluster1

接下来对数据进行降维，并进行对 K-means 算法表情发现任务的可视化展示。使用 K-means 算法对降维后的图片进行表情分类的聚类簇可视化如图 3.7 所示，图中的红色五角星为聚类中心。

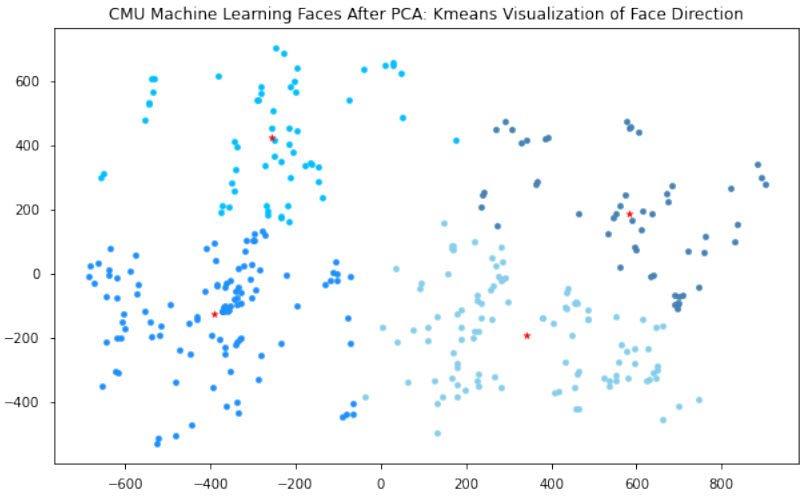

图 3.7: K-means 算法面部特征发现

以上是使用 Kmeans 聚类算法的全部过程，可见无监督学习在这个任务上的表现并不出色，具有很大的不确定性。

3.2.2 使用神经网络算法进行相似人脸特征发现

在上面的实验中可以发现，Kmeans 聚类算法并不能有效且稳定的解决面部特征识别任务，所以我们再次使用神经网络算法进行分类任务。从人脸识别任务到面部特征识别任务的迁移较为简单，主要修改的是算法的输出。

如图 3.8 所示，这个四分类问题的损失下降的非常平滑直至收敛。

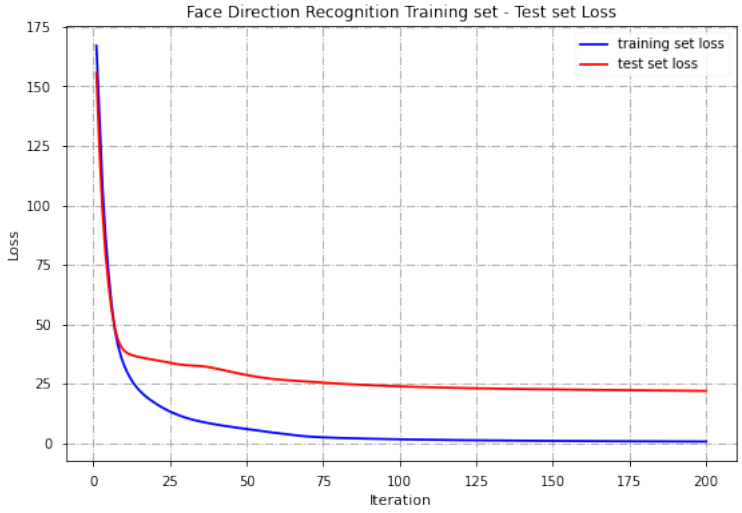

图 3.8: Face Direction Recognition Loss

再来看下这个问题使用神经网络算法分类的准确率，如图 3.9 所示，训练集可以完全收敛，准确率达到 1，测试集准确率最高达到 0.944。神经网络算法在该问题的准确

度上相较 Kmeans 聚类算法有非常显著的提升。

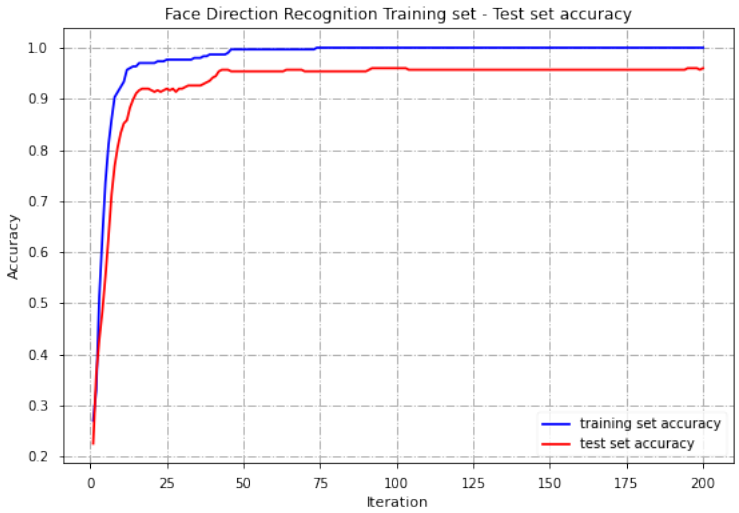

图 3.9: Face Direction Recognition Accuracy

3.2.3 解释与改良方案

对表情分类任务的实验数据如下所示，如下图所示，训练的效果非常差，测试集准确率和盲猜几乎没有区别。

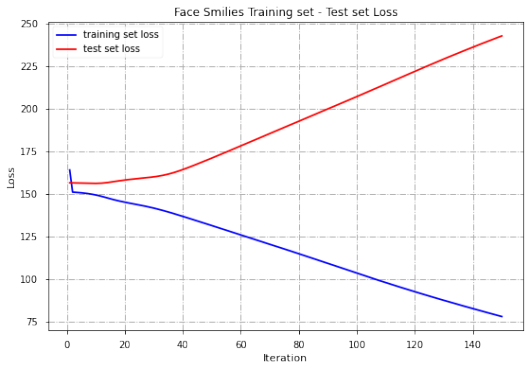

图 3.10: 表情分类损失值变化

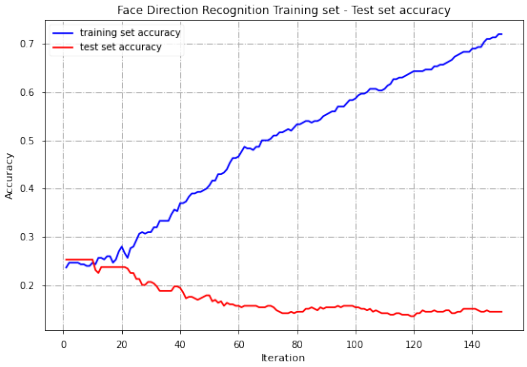

图 3.11: 表情分类准确率变化

我们所使用的卡内基 · 梅隆大学 Face Images Data Set 数据集是一个相当老的数据集，不仅样本数量少，数据的质量也有待商榷。我认为无法获得较好的情绪分类结果原因有以下两个

- 该人脸数据集没有去除背景，如图 1.3 所示，该数据集的人脸图不仅包含人脸，还包含了与人脸无关的背景（如电脑），这对聚类问题而言会是严重的噪声。
- 数据集规模小，且全部为黑白图片，损失了大量真实世界的人脸的特征。

第 4 章　参考资料

(1)《统计学习方法（第二版）》李航. 清华大学出版社

(2) Stanford cs229: Deep Learning Cheatsheet

(3) Stanford cs231n: Weight Initialization

(4) Understanding the difficulty of training deep feedforward neural networks