CSE676 Project 2:
Age Estimation for Facial Images with CNN and Autoencoder

Team 5:
ShihChia Chen 50207079
Jiawei Zhao 50207069
Date: 11/30/2017

1. Abstract

- Task Type: classification
- Task Description:
  For Project1, we applied CNN to learn an age estimation model with approximately 25,000 images, and then use this model to estimate the ages for given test images (approximately 5,000 images), and test the accuracy by compare the guess with the true label. For this project, we add Autoencoder to this process, hoping it can first do a feature extraction for us and then using the extracted and extended image as the input for the training of our model and in the end reach a better result.

- Machine Learning Method
  - Model: CNN and Autoencoder
  - Architecture: CNN has two convolutional layer, two max pooling layer and one fully connected layer. Autoencoder has five encodes and five decodes and the output has the same resolution with the input data.
  - Cost Function: cross_entropy for CNN and square error for Autoencoder
  - Optimizer: AdamOptimizer

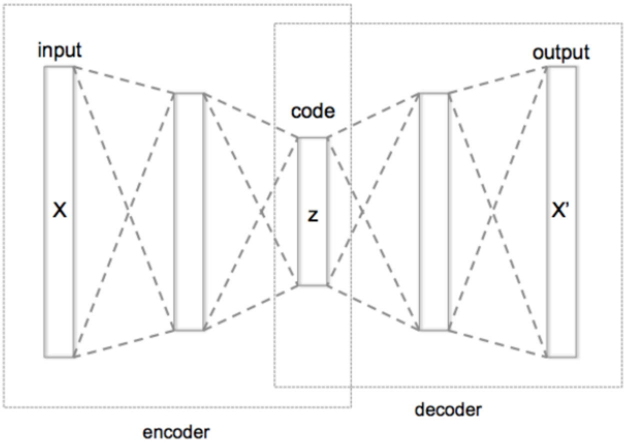
2. IMDB-WIKI Dataset

- Dataset Description:
  One of the largest publicly available dataset of facial images with gender and age labels for training. All information is stored in a *.mat* file. We can easily access to these information by using scipy(a python library), all information are stored as dictionary and can be retrieved easily. More details are available in the link below.

- Dataset Link: https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

3. Introduction to convolutional neural network(CNN) and Autoencoder

- Autoencoder
  An autoencoder is an artificial neural network used for unsupervised feature learning. The aim of autoencoder is to learn a representation for a set of data.



- Convolutional Neural Network

  CNN is a class of deep, feed-forward artificial neural network. The network learns the filters that in traditional algorithms were hand-engineered. Independence of priori knowledge of the data is a major advantage. The hidden layer of CNN typically consists of convolutional layer, pooling layer and fully connected layer.

4. Main idea behind this project

Our dataset images are very large, and the face may just consist part of the whole image, just as figure1 indicate, which means that there could be lots of noise as we process these data, and as a result of that, the accuracy may be jeopardized by those unnecessary information, and here is where the autoencoder comes in, autoencoder could be a good feature extractor, by applying the raw input through an autoencoder and the decode output as the input of our old CNN, hopefully the autoencoder could extract valuable information and discard unnecessary information, which would make the data denser and as a result help us to achieve a better accuracy.

Figure1, Sample data

5. Performance Measure and Expected Results

- Performance Measure

  Using age estimation accuracy as the indicator of the performance of our network. Also, we lower the requirement by considering the estimation to be correct if the guess lies within a range of the true age of the figure. That's because we have a large range of output value, from 0 to 99, which makes exact guess very difficult.

- Expected Results

  The objective of this project is to compare the result with the first project, because we have done some feature extraction before feeding them into the neural network, we can anticipate that there will be some degree of increment in the accuracy we get, and that depends very much on the performance of the autoencoder we apply.

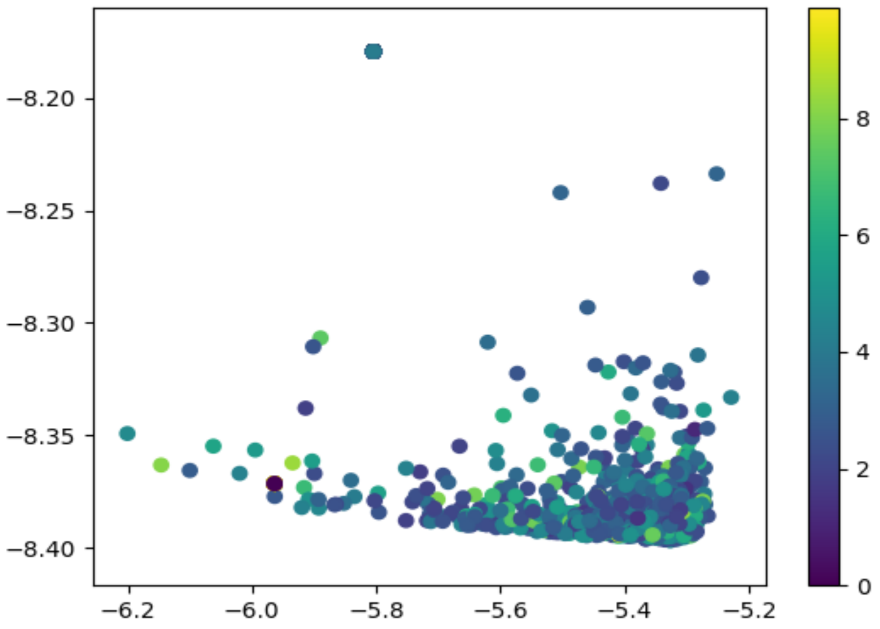
6. Results

All results are ran on google cloud platform with its computing instances.
Official website: cloud.google.com

We measure the accuracy in four different ranges of age:

- Result differs from true label in less than or equal to one year
- Result differs from true label in less than or equal to three years
- Result differs from true label in less than or equal to five years

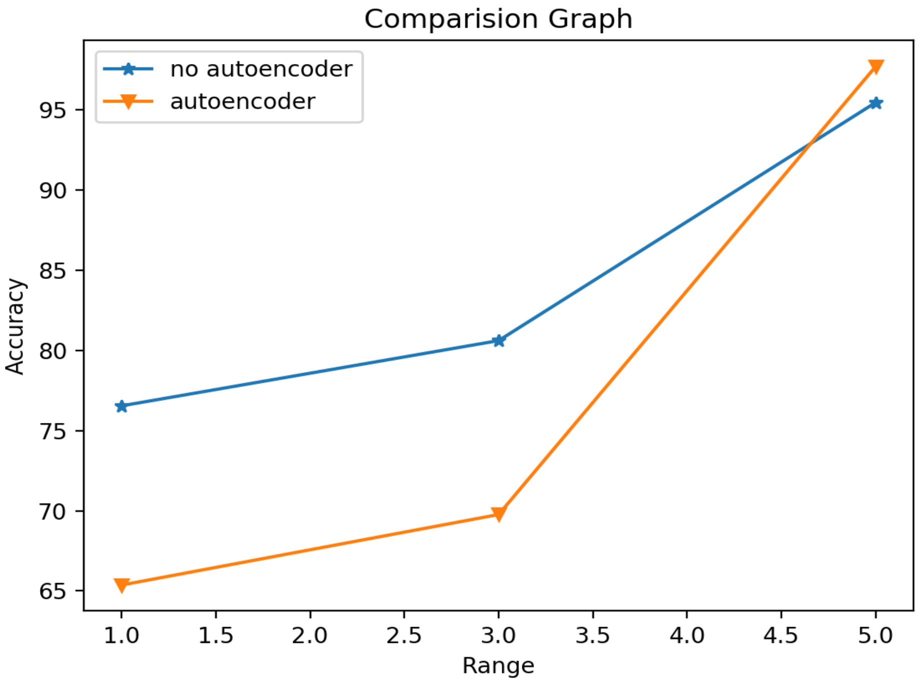
Encoder Result Visualization:



Accuracy Table:

Range of age	[age-1, age+1]	[age-3, age+3]	[age-5, age+5]
Accuracy (No autoencoder)	76.55%	80.62%	95.48%
Accuracy (with autoencoder)	65.37%	69.77%	97.73%

Comparison Graph:



7. Conclusion and Some Thoughts

First, let's look at the result visualization of the encoder process, ideally, for this unsupervised learning, we expect the encoding process to group the input data into 10 clusters, however, we can see that our encoder fails to fulfill this task greatly, we see that most of the data is located at the right-bottom corner, yet still they tend to be separated, and we also see that there're many point that goes far from that right-bottom corner. And the pattern for these points is that the points that located at left-bottom corner is lighter whereas the points lie at the right-upper corner is darker. In conclusion, we see that the autoencoder indeed is trying to cluster the points to different corners, however, because the data is big and many of them is noise, so it is hard for the autoencoder to differentiate and discard the proper feature sometimes, and resulting in a not very satisfactory encoding.

Then let's look at the prediction accuracy. Just as expected, after applying the autoencoder for the data, the data's features are extracted and hence we should get a better result, ie. higher accuracy. And with experiment of age ranging between true age-5 to true age+5, we get the expected result, however, for others, we can see that the accuracy actually goes down after we apply autoencoder. And that's because the autoencoder fails to extract useful features or has extracted too much misleading information. We should clearly understand the advantages and disadvantages of autoencoder before we choose it as our feature extractor.

- Pros:
  - Autoencoder network is handy and easy to build
  - Autoencoder is a great way for feature extraction
  - Autoencoder is an efficient network for dimention reduction
- Cons:
  - If the data isn't representative, then the autoencoder may obscure the information rather than clarify it
  - An autoencoder learns to capture as much information as possible rather than as relevant information as possible, which can lead to misleading results

In a nutshell, we think that it's good for us to employ autoencoder if our data is specific and doesn't have too much noise, and applying autoencoder is very likely to increase our neural network performance, on the other hand, if the data is big and general, or contains too much unnecessary information, it's not recommended to apply an autoencoder beforehand since that may result in loss of truly valuable information and aggregate unnecessary or even misleading information.