Convolutional Neural Network for Gender and Age Analysis Based on Face Images

Yuehan Wang
University of Iowa
yuehanui@gmail.com

may 10, 2018

Abstract
In this paper, I build up a Convolutional Neural Network (CNN) for gender and age recognition based on face images. Gender and age recognition and analysis has broad application prospect. Such technology has been widely used in large-scale shopping malls and commercial streets for customer demographics.

1. Code Environment
The project is written using python 3. The following modules are used in the code and must be installed before the code could run:

- openCV
- Keras
- Tensorflow
- Numpy

2. Dataset
The IMDB-WIKI dataset as I know is the largest publicly available dataset of face image with age and gender labels. Due to the computing power of my machine, only the WIKI portion of the IMDB-WIKI dataset is used in this project. But The function written in this project allows the readers to train the network using the whole dataset in reach best result. The WIKI dataset (1) contains 62,308 images. The labels information of the images is stored in the .mat file. The format of the labels information in the .mat file is:

- Date of Birth
- Date when the photo was taken
- Path of the image
- Gender (binary)
- Name of the person in image
- Etc.

To train a convolutional neural network for gender and age analysis, we only need four of the information in the labels—paths of the image which allow us to load the image, gender labels for gender analysis training, date of birth and date when the photo was taken to calculate the age for age analysis training. The IMDB-WIKI dataset already provided a version of data with face only for download so we don't need to work with face recognition and cropping. The file process_data.py is written to preprocess the data before training. It use scipy.io to read the file wiki.mat and returns a list of image paths, a list of gender labels and a list of age labels. The dataset contains a large amount of low-grade images, which are either not only contains faces,

not faces at all or completely black or white images. If we keep these image, the training data will likely to be too noisy the training will be difficult to converge. Some of the low-grade images are excluded by functions in process_data.py and some of them are removed from the dataset manually. About eighty percent of the images labeled as male in the originally dataset. I delete some of the images which are labeled as male to make the image of male and female generally equal. The images then are cropped using openCV to the size of 64*64. The dataset then is split in the ratio of 8:1:1 of training set, valid set and testing set. After the modification and adjusts named above, the final dataset I use to train the network has the following specifications:

- Dataset total: 9701 images

- Gender composition:

Male	Female
53%	47%

- Age composition:

Age range	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
Percent	0.4%	6.7%	37.9%	21.6%	13.1%	9.0%	6.0%	2.9%	1.9%	0.4%

- Image size: 64*64*3

2. Methods

The training network is built based on Keras and using Tensorflow backend. The Network consists of the following 6 layers(Figure 1):

- Convolutional 2D layer with 128 filters of size 5x5 and with stride 4 with input size of 64*64*3 with Activation function LeakyReLU
- Convolutional 2D layer with 256 filters of size 5x5 with Activation function LeakyReLU
- Maxpooling layer with pool size 2x2
- Dropout layer with drop rate 0.25
- Convolutional 2D layer with 128 filters of size 5x5 with Activation function LeakyReLU
- Convolutional 2D layer with 256 filters of size 5x5 with Activation function LeakyReLU
- Maxpooling layer with pool size 2x2
- Dropout layer with drop rate 0.25
- Dropout layer with drop rate 0.25
- Fully connected layer with 1024 node
- Dropout layer with drop rate 0.25
- Softmax output layer with size 2(in gender prediction) or 100(in age prediction)

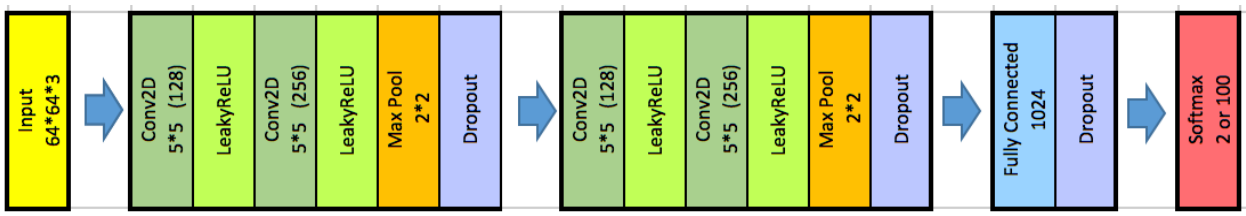
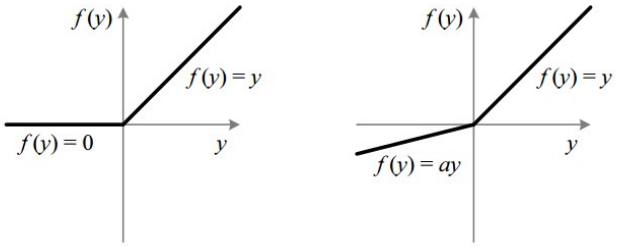


Figure 1 Network Structure.

The learning rate is fixed at 0.1 and with nesterov momentum of 0.9. The activation function is LeakyReLU with alpha = 0.01. The LeakyReLU is a leaky version of Rectified Linear Unit. It allows small gradient when the unit is not active and therefore it prevent the node from dying which happens in ReLU.
The equation of LeakyReLU (with alpha = 0.01) is:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$

The following graphs shows the difference between ReLU function and LeakyReLU function:



ReLU LeakyReLU

Figure 1. image from Towarddatascience.com

4. Result

The gender prediction model reached a train accuracy of 63% and test accuracy of 76% after 200 epochs. I expect the accuracy would continue to increase if we run more training epochs.
The age prediction model did not reach a good accuracy. After training for 200 epochs, the training accuracy is 6.2% and test accuracy is 0.3%. Compared with the gender prediction which has only 2 classes, the age prediction model has 100 classes so I assume it would require larger dataset, more complex network and more training epochs to converge the loss.

5. Files usage:

The size of the training data is too big to upload. The dataset must be downloaded manually before training. Small size of test dataset is attached in the file.

File	Usage
process_data.py	File to process the raw data download from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/static/wiki_crop.tar before training.
gender_train.py	Using the data to train the model of gender prediction, dataset must be downloaded before training.
age_train.py	Using the data to train the model of age prediction, dataset must be downloaded before training.
gender_test.py	Test the gender prediction model
age_test.py	Test the gender prediction model

5. Train using the whole IMDB-WIKI dataset.

I only train the model using a data size of 9701 images. The whole IMDB-WIKI dataset contains more than 500 thousand face images with labels. To train the model using the whole data set would result in better result but it requires powerful machine and time.

To train using the whole IMDB-WIKI dataset, download the face only images and metadata from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/static/wiki_crop.tar. Put the wiki_crop file under the same path as process_data.py. Then run the process_data.py file. A new folder named picture will appear. Then you can run either the gender_train.py or age_train.py to train either the model of gender prediction or age prediction using the whole WIKI dataset.

References:

[1] SHARMA, S. (2017). *Activation Functions: Neural Networks – Towards Data Science*. [online] Towards Data Science. Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 [Accessed 14 May 2018].

[2] Keras.io. (n.d.). *Advanced Activations Layers - Keras Documentation*. [online] Available at: https://keras.io/layers/advanced-activations/ [Accessed 14 May 2018].

[3] Ekmekji, A. (n.d.). *Convolutional Neural Networks for Age and Gender Classification*. [online] Cs231n.stanford.edu. Available at: http://cs231n.stanford.edu/reports/2016/pdfs/003_Report.pdf [Accessed 14 May 2018].

[4] GitHub. (2018). *How could we use Leaky ReLU and Parametric ReLU as activation function ? · Issue #117 · keras-team/keras*. [online] Available at: https://github.com/keras-team/keras/issues/117 [Accessed 14 May 2018].