Capstone Project Proposal, Udacity

Age-Gender Classifier

By Kouassi Konan Jean-Claude

I- Domain Background

Deep Learning is a recent advance in the field of Machine Learning which allows or even overpasses human-like performances. It is applied to perform several tasks in Artificial Intelligence, Self-Driving Car, Robotics, etc.
Deep Learning is also used in computer vision where nowadays it overpasses human vision in several domains such as age and gender guessing from pictures.
Some researches are still done on the subject. We could relate for example the interesting recent research work of Gil Levi and Tal Hassner from the Open University of Israel (*Age and gender classification using convolutional neural networks*), in the field of Computer Vision and Pattern Recognition (CVPR), in June 2015. They performed a separate Age and Gender Classification using Convolutional Neural Networks. Their process was built around the Adience benchmark, a public Age and Gender Classification Dataset which assembles 26,580 photos of 2,284 subjects.

My main goal with this project is to unify the prediction process in one step for both predictions, age and gender on the same Adience benchmark. For each image of this benchmark, I should generate its One-Hot Label containing its age and gender. But this time I will perform the work using a different Deep Neural Network architecture, trying to get or overpass their accuracies (86.8 ± 1.4 for Gender and 84.7 ± 2.2 for Age).

At the end of the project, I should also provide some simple and easy to use tools for dataset preprocessing on Adience benchmark in Python, considering different Machine Learning Frameworks requirements and process unification too. And, as an option, users should be able to install the application on their smartphone and perform age and gender classification on unseen faces.

II- Problem Statement

Nowadays Deep Learning technology is used in Computer Vision to provide or overpass human-like level for Age and Gender classification. This could be very useful for ***Identification and Authentication processes*** for many infrastructure systems.
We have a need to secure our infrastructures so that only trusted persons could have access.
On the other hand, we also know that smartphones are expanded through the world. Everybody has a smartphone and a solution which includes it should be very practical in term of accessibility and easiness. So why not use it for Age and Gender classification?
We think the combination of both, Deep Learning technologies for Age and Gender classification, and smartphones should bring significant revolutions in the domain of Identification. For example using a simple smartphone, we should control the access to a room to teenagers or children.

So, with this project, basis on the previous public Adience benchmark, I am going to create an app which can do Age and Gender classification using latest high-level computing technologies such as Fully-Connected Networks in a unique step. During the process, the intended input should be target faces images and as output, we should have a label with the correct information.

III- Datasets and Inputs

As for any machine learning problem, we need more data, relevant data to compute the Age and Gender classification with the highest accuracy.
Indeed, we need an accurate dataset which provides correct age and gender labels on face images. An error on the dataset labels could lead to biases on the final application.
Build such dataset is a very hard work because of availability of people, accurate information on them, the amount of data required (several dozens of thousand), data collection cost and time.
So, we will work with the publicly available Adience benchmark which had already been used for several research works.
It is a dataset of 26,580 photos of 2,284 subjects, with separate age and gender labels. There is one model with age label only and another with gender label only.

During the data pre-processing, we will unify the data labels. For each image of this Adience benchmark which has a both labels, we should generate its One-Hot Label containing its age and gender. And so we will have a new ***Unified Dataset*** (saved in a variable).
So, *for the training process,* we will use the *Unified Dataset* ***(the inputs will be image ids of the Unified DataSet and the outputs are One-Hot Labels with correct information)***.
For the Evaluation process, we will use the *Unified Test Set* ***(the inputs will be image ids of the Unified Test Set and the outputs are One-Hot Labels with correct information)***.
For the Prediction process, we will use the ***Unified Test Set, other unseen face images and non-face images as inputs. The outputs stay One-Hot Labels with correct information***.

The intended Deep Neural Network used for the Network Model should be a ***Fully-Connected Networks***.

IV- Solution Statement

Our intended solution for the Age and Gender classification problem should be the building of a ***Fully-Connected Networks model***.
Then we will apply the Unified DataSet to the model to teach him how to recognize age and gender on images; especially through the training, evaluation and prediction processes defined above.
Our solution model will be evaluated using its accuracy on the Adience benchmark.
This accuracy is in fact how well the model predicts the one-hot labels on the Adience benchmark. Indeed, each one-hot label of the Unified Dataset is a tuple of 10 dimensions (8 digits for the age classes and the last 2 digits for the gender classes). Initially, all the digits are set to zero, and the model will put only the correct classes' values to 1.

And once all is done we could build the mobile application using our deep learning implementation.

V- Benchmark Model

Our ***Age and Gender Classification Model*** is built using a ***Fully-Connected Network*** trained on the Unified Dataset.
Using this model, we could generate for each face image a one-hot label with its correct age and gender information.
Comparatively to the "*Age and gender classification using convolutional neural networks*" paper where we have to perform the prediction for age or gender separately, now with our ***Age and Gender Classification Model*** we could do both operations in one step.

So, our final solution will be evaluated on the performance of our benchmark model (our ***Age and Gender Classification Model***) to predict good labels.

VI- Evaluation Metrics

Our work is based on the **Adience benchmark.**
The **Adience benchmark** is a dataset of 26,580 photos of 2,284 subjects, with separate age and gender labels.

As Evaluation Metrics we will use:
1- *The **Loss** and **Accuracy** of our benchmark model (Age and Gender Classification Model) on the **Unified Test Set (computed from the Adience benchmark)***.

==> We could consider the ***Loss*** as the difference between an estimation made by the model ($\hat{f}$) and its true value in the dataset (f); $\boxed{L(f,\hat{f}) = \|f-\hat{f}\|_2^2}$, representing the price paid for inaccuracy of predictions in Machine Learning problems. So we expect the Loss to be very small, as small as possible.

In Deep Learning, using weights (ω) and biases (b) applied on the input vector:
Loss = AVERAGE CROSS ENTROPY

$$\boxed{\mathcal{L} = \frac{1}{N}\sum_i D(S(\omega X_i + b), L_i)},$$

S = *A Softmax function which turns scores into probalities (**the predictions**)*
L_i: Local minima which will become the One-Hot Label (***the true values in the dataset***)
D = Derivative *or* ***Distance***
Cross Entropy = Distance between the two probabilities vectors (S and L_i)
The **Training Loss** is a measure of the distance average over the entire training set for all inputs and all the labels that are available.

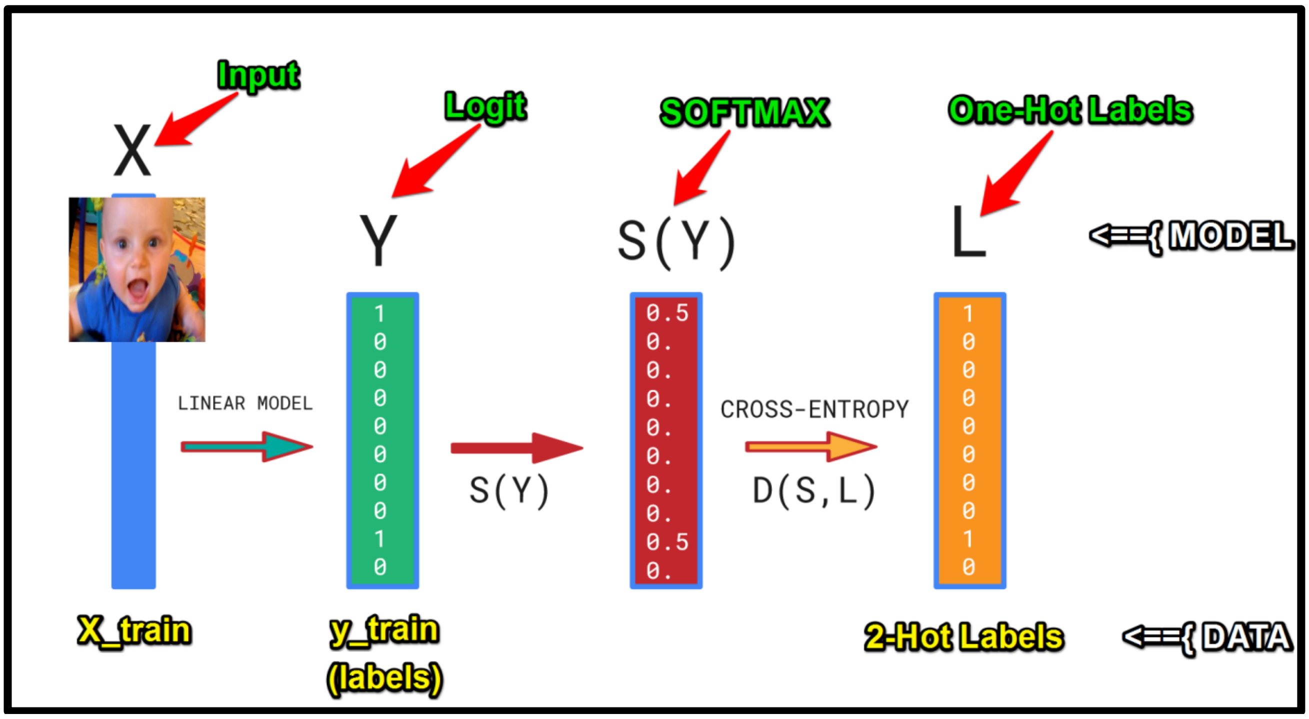

Figure 1: The ***Training Loss Process*** computation on the Unified Training Set

There are several types of Loss functions according to the Machine Learning task to perform (regression, classification, decision tree, etc.). But ***as we are working on a Supervised Classification problem we will only be interested in Loss functions for classification: Square loss, Logistic loss, Hinge loss, Cross-entropy loss***. Indeed, these losses work better on classification problems.

==> We could consider the ***Accuracy*** as the number of correct predictions made by the model among a set of input data (for example the *Unified Test Set* images).

$$\boxed{Accuracy = \frac{\textbf{Number of well predicted One} - \textbf{Hot Labels}}{\textbf{Total number of images in the Unified Test Set}}}$$

***2- The Accuracy** of our Age and Gender Classification Model on the **Adience benchmark** and also on **other unseen face images**.*

Indeed, first, we will evaluate and validate the performance of our benchmark model *(Age and Gender Classification Model)* on its capacity to produce ***low Loss*** and ***high Accuracy*** during the training process (How well it learns?). *Then, we will use the **validated benchmark model*** to evaluate its Accuracy on *the **Adience benchmark** and also on **other unseen face images***.

VII- Project Design

The goal of this project is to build an Age and Gender Classifier and then deploy the final improved model into a mobile application to allow live Age and Gender Classification such as the digit recognition of the Deep Learning project.

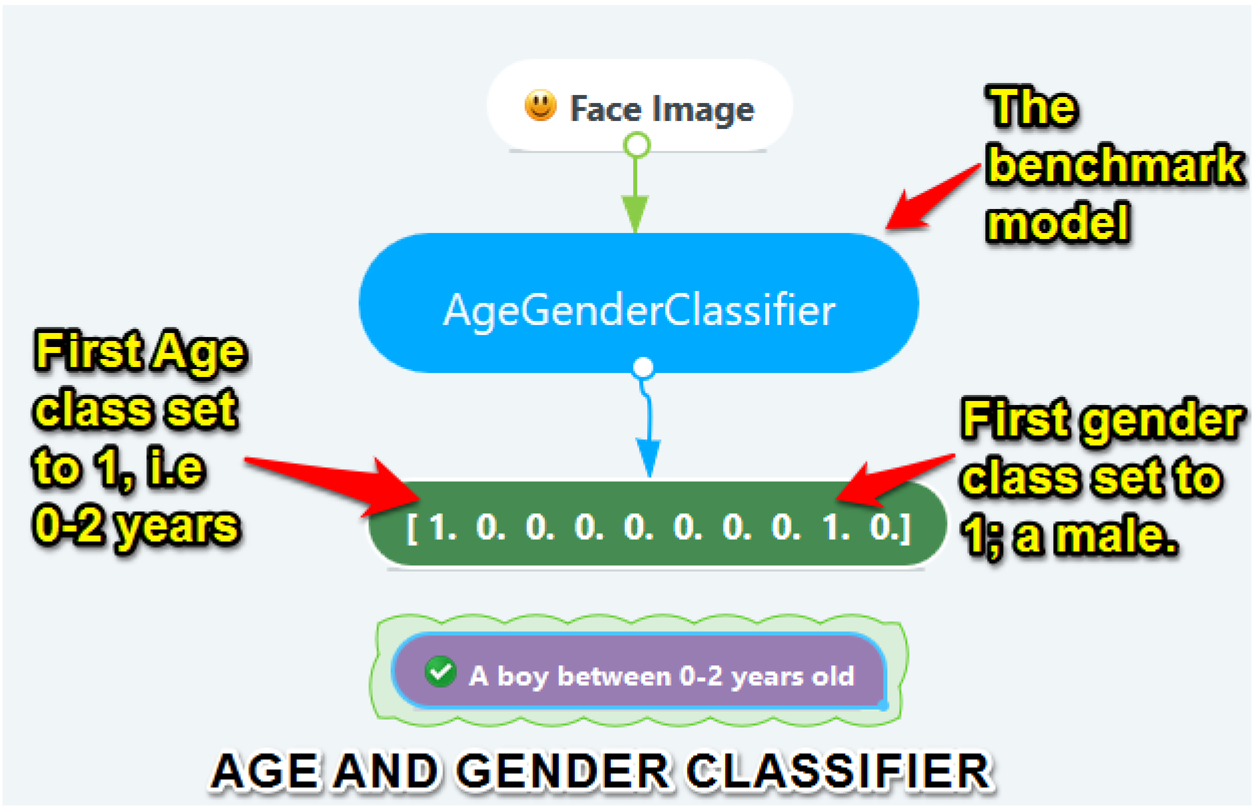
Face Image
The benchmark model
AgeGenderClassifier
First Age class set to 1, i.e 0-2 years
[1. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
First gender class set to 1; a male.
A boy between 0-2 years old
AGE AND GENDER CLASSIFIER

Figure 2: The benchmark model (Age and Gender Classification Model)

As shown in Figure 2, raw face images are sent to the benchmark model as input. Then they will be pre-processed into a numpy array to meet the classifier requirements.
The model itself (Age Gender Classifier) consist of a Fully-Connected Artificial Neural Network (FCN) which takes the images as input and output a 10-dimensional One-Hot Label (8 digits for the age classes and the last 2 digits for the gender classes).
***Note*: Here we keep the appellation One-Hot Label because it is done for each class (age or gender) separately, but the final result should be a 2-Hot Label.**

Two frameworks will be used for the FCN: TFLearn and Tensorflow Keras. Both admit numpy array and lists as input for the training process.
It is also possible to use Tensorflow *DNNClassifier* or *DNNRegressor* which are adapted for classification tasks, but in our case, they will require more data pre-processing operations, as they work only with dictionaries as input.
However, they have the advantage to produce a few line of code for the network model and easy visualization tools.

1- Contribution of this project

We define below **our contribution on the "Age and gender classification using convolutional neural networks" paper**.

The main contribution had been:

- ✓ Provide some simple and easy to use tools for dataset pre-processing, considering different Machine Learning Frameworks requirements.
- ✓ Assemble the prediction process in one step for both predictions, age and gender.
- ✓ Turning the implementation into video and mobile application
- ✓ Another plus is the publishing of a detailed step by step guide for implementation of this kind of solution.

2- Create a working Age-Gender Classifier
Below the milestone for the Age-Gender Classifier creation:

2.1- Data Preparation

Here, using the Adience benchmark we created a function (***age_gender_per_image***) to perform dataset unification. This function generates an age and gender dictionary of all images which have both information available. The key of the dictionary is the full path name of the images in the Adience benchmark.
Then, using the *age_gender_per_image* dictionary we created the ***One_Hot_Labels*** dictionary which also has the full path name of the images in the Adience benchmark as keys, and uses ***as values*** the *One-Hot Labels* generated from the *age_gender_per_image dictionary*.

The ***One_Hot_Labels dictionary*** is considered as the $\boxed{\textit{Unified DataSet}}$ as it has for each image, its age and gender information as a unique one-hot label (***2-hot label***), see *Figure 1*.
The size of the Unified Dataset is ***16156***. It contains only unique images that have both information, age and gender. We get the same size from the *aligned* and *non-aligned* datasets of the Adience Benchmark.

So, it (the Unified Dataset) will finally be used to produce ***age_gender_feature_list*** and ***age_gender_label_list*** which are the input for the ***Training and Test Sets*** creation, using the sklearn *train_test_split* tool. This tool accepts only *lists, numpy arrays, scipy-sparse matrices* or *pandas dataframes* as input.

If we need to provide a ***Validation dataset***, we will use this time the model estimators tools to use a part of the Training set as Validation set.

2.2- Building the Age-Gender Classifier

Here we created two Fully-Connected Artificial Neural Networks (FCNs); one with TFLearn and the other with Tensorflow Keras. Both admit numpy array and lists as input for the training process. Doing so, we will be able to compare their influences on the results and increase the model performance.

2.3- Models Evaluation

Here Loss and Accuracy are our Evaluation Metrics.

- ✓ With *TFLearn*: we use tensorboard to visualize the learning curves.

tensorboard --logdir='tflearn_logs' (in a terminal of the server where the notebook is executed)

- ✓ With *Tensorflow Keras*: as tensorboard is not included in the version 1.1 of Tensorflow, we printed the learning curves using the training history.

```
print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Tensorboard is available with *Tensorflow Keras in the* Tensorflow 1.2 branch, some updates are still ongoing with this branch, and it will be worth to try it.

At this point, we should have a working Age-Gender Classifier.

3- Making Predictions

The Prediction process for both frameworks (TFlearn and *Tensorflow Keras*) are made on:

- ✓ the Unified Test Set,
- ✓ other unseen face images and
- ✓ non-face images as inputs.

The outputs stay One-Hot Labels with correct information.
Here we focus on the accuracy of the model in each case to relate how well it performs.

4- Improvement

According to the previous results, we could perform some improvement to increase the accuracy with each framework using this checklist:

- ✓ ***The number of epoch:*** increase this number could give to the model more time to learn better.
- ✓ ***The optimizer functions*** *(Adam, RMSProp, Adadelta, SGD, etc.).*
- ✓ ***The activation functions*** *(sigmoid, relu, softplus, hard_sigmoid, softsign, etc.).*
- ✓ ***The loss functions*** *(categorical_crossentropy, mean_squared_error, squared_hinge, sparse_categorical_crossentropy, hinge, mean_squared_logarithmic_error, etc.).*
- ✓ ***The regularization functions*** *(L2 regularization, l1_l2, etc.).*
- ✓ ***The initializers*** *(he_normal, he_uniform, Xavier normal initializer, Xavier uniform initializer, LeCun uniform initializer, etc.).*
- ✓ ***The model:*** *when building the model we thought to another version of the output which could require less computation. The idea is to output a tuple (a, g) at the end of the network. Where a is the age range [0:7] and g, the gender (0 or 1). So the output and labels of the Figure1 would become (0, 1). 0 for the first class of age [0-2 years] and 1 for the gender as a male. But even if* ***it does not clearly appear****, we have still* ***10*** *classes (****8*** *possibilities for age and* ***2*** *for gender).*
  The computation cost during the learning process is:

  - First model: $\left(\overbrace{\{0,1\}}^{2},\overbrace{\{0,1\}}^{2},\dots,\overbrace{\{0,1\}}^{2}\right)$ ***==> [0. 0. 0. 0. 0. 1. 0. 0. 1. 0.] ==>*** $2^{10} = 1024$

  - Second model: $\left[\overbrace{\{0,1,2,3,4,5,6,7\}}^{8\ possibilities},\overbrace{\{0,1\}}^{2}\right]$ ***==> (3, 1) ==>*** $8 * 2 = 16$

  So, we could try the second model and fine tune our implementation. If we pick the second model, let's note that it is still possible to switch between the two kinds of output using a simple function (generate a 10-dimensional 2-hot label from a tuple).

- ✓ ***The size of the unified dataset:*** the Adience benchmark presents two datasets, non-aligned faces dataset (1.2Go) and aligned (2.6Go). We could see the difference of accuracy between the non-aligned dataset and the aligned one for the same image ids.
  We could also ***increase the sample of training data*** to improve the accuracy.
  Here are some Datasets which could be used to perform this task. A specific script should be written for each dataset to add its content to the Adience benchmark (the Unified dataset):

  - FaceTracer Database
  http://www.cs.columbia.edu/CAVE/databases/facetracer/
  15,000 faces in the database

  - Color FERET Database
  https://www.nist.gov/itl/iad/image-group/color-feret-database
  Contains 1564 sets of images for a total of 14,126 images that includes 1199 individuals and 365 duplicate sets of images.

  - Karolinska Directed Emotional Faces (KDEF)
  http://www.emotionlab.se/resources/kdef
  A set of totally 4900 pictures of human facial expressions of emotion.

- Chicago Face Database
http://faculty.chicagobooth.edu/bernd.wittenbrink/cfd/download/download.html
Version 2 of the database includes high-resolution photographs of 597 male and female targets of varying ethnicity. File size is approx. 1.5 GB.

- Agingmind Stimuli Face Database
http://agingmind.utdallas.edu/download-stimuli/face-database/
A database of 575 individual faces ranging from ages 18 to 93.

5- Build the mobile application

This is the final step of our Age-Gender Classifier building.
Once we have created an Age-Gender Classifier on our computer which works well, we must now convert it into a beautiful mobile application in python with graphic user interfaces.

5.1- Several possibilities to run the App on videos:

- ✓ MoviePy, Python module for video editing. It works on Windows, Mac, and Linux, with Python 2 or Python 3.
- ✓ Python Live Video Streaming Example
- ✓ Applying the functions to openCV for video processing as in the [Udacity-Self-Driving-Car-Preview]; look at the video part.
- ✓ The Tensorflow 1.2 branch has some video processing functions [VIDEO-QA IMPLEMENTATION WITH TENSORFLOW 1.2RC0]

5.2- On Mobile:

- ✓ Deploying a TensorFlow model to Android
- ✓ Creating Custom Model For Android Using TensorFlow
- ✓ Tensorflow codelabs:
  - ✓ [TensorFlow for Poets 1]
  - ✓ [TensorFlow for Poets 2: Optimize for Mobile]

- ✓ Develop Your First Android Application in Python
- ✓ Python for Android Tutorial
- ✓ TensorFlow Android Camera Demo
- ✓ Programming Foundations with Python
- ✓ Developing Scalable Apps in Python
- ✓ Android Development for Beginners
- ✓ Android Basics: Multi-screen Apps

5.3- Below the intending functionalities of the App:

1- Select a picture or take a video with your phone
2- The App give the gender and age of the picture (For instance, "*It seems that we have a 30 years old Woman in this picture*")

6- Future work

As a futurist technology, the Age-Gender Classifier should be continuously improved. Here are some ideas of future work on the subject:

- It would be worth to apply the same technique on other types of Artificial Neural Networks such as GAN, as it is a recent field always in exploration, ***mainly the Wasserstein GAN*** which produces accurate results.
- Propose a ***universal and easy method*** for deployment of notebook implementations on a Mobile App. Indeed, the actual methods are fine but limited to some frameworks only.
  Now, Tensorflow could be deployed on Android and iOS.
  But other methods are welcome, ***mostly universal methods*** (cross-platform, supporting several Deep Learning frameworks).
  For instance, a way to deploy Tensorflow, TFLearn and Keras implementations on Android, iOS, Windows Mobile, etc. with a single platform.

VIII- References

1- Age and gender classification using convolutional neural networks (The research paper) | Gil Levi, Tal Hassner

2- The Adience benchmark, a public Age and Gender Classification Dataset of the Open University of Israel (Face Image Project)

3- Face Aging with Conditional Generative Adversarial Networks | Grigory Antipov, Moez Baccouche, Jean-Luc Dugelay

4- https://www.tensorflow.org

5- http://tflearn.org

6- https://keras.io

7- https://en.wikipedia.org/wiki/Loss_functions_for_classification

8- http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/

9- MoviePy

10- Deploying a TensorFlow model to Android

11- Creating Custom Model For Android Using TensorFlow

12- TensorFlow for Poets 2: Optimize for Mobile