Deep Learning Lab

Open Ended Assignment

'Age & Gender Classification'

Using Convolutional Neural Network

Performed by

Jyotika Mane – 3824
Naina Dhingra – 3840
Aparajita Sarkar – 3853

Under the guidance of
Prof. S. Potdar

DEPARTMENT OF
ELECTRONICS AND TELECOMMUNICATION ENGINEERING
MKSSS's Cummins College of Engineering for Women, Pune
(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

TITLE: 'Age & Gender Classification' Using Convolutional Neural Network

OBJECTIVE:

The objective of this project was to develop deep learning models to accurately classify people into age categories and predict their gender. This report will discuss the methods used to pre-process the dataset, the models employed, and the results obtained from the models.

INTRODUCTION TO 'AGE-GENDER.CSV':

This dataset contains 23,705 samples wherein each sample has the following attributes – age, gender, ethnicity, name of picture, pixel values.

The age values range from 0 to 60, gender is split into male and female.

(Ethnicity is not being considered for this project as it is a sensitive subject and we didn't want our model to have the ability to create a bias on the basis of the ethnicity of an individual.)

Each picture has 2,304 pixels.

DISCUSSION ON THE DATASET:

The histograms for age and gender were plotted which gave us the following insights:

1. Most of the participants have age 26. The age distribution is a unimodal uniform distribution.
2. 12.3 k samples are of males and 11.3 k samples are of females.
3. The high intensity pixels' values may cause the model to be biased, so the pixel values are normalized (with respect to maximum pixel intensity – 255)

Otherwise, we see that the data is not too biased towards any point and there are no outliers or missing values, which eliminates the need for other data pre-processing techniques at this stage.

BRIEF DISCUSSION ON THE DEEP LEARNING MODEL IMPLEMENTED:

We used a Convolutional Neural Network with the following specifications:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 batch_normalization (BatchN  (None, 46, 46, 32)       128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 23, 23, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 21, 21, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 10, 10, 64)       0
 2D)

 flatten (Flatten)           (None, 6400)              0

 dense (Dense)               (None, 64)                409664

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 428,673
Trainable params: 428,609
Non-trainable params: 64
_________________________________________________________________
```

Other details:

1. Splitting dataset:
   - ~ 78% training (including 10% for validation)
   - ~ 22% testing
2. Activation functions:
   - ~ ReLU – for all the hidden layers
   - ~ Sigmoid – for the final densely connected layer as we require the output as either 0 or 1.
3. Number of epochs:
   Decided according to the validation error. Model is trained till validation error becomes less than 0.27.

These algorithms were chosen because they are commonly used in multi-class classification problems and have been shown to perform well on similar datasets.

OPTIMIZATION TECHNIQUES USED:

We trained the same CNN having above mentioned specifications with the following optimization techniques so as to have a comparative understanding.

1. **Stochastic Gradient Descent (SGD):** SGD algorithm is an extension of the Gradient Descent and it overcomes some its disadvantages. In the SGD algorithm derivative is computed taking one point at a time. The same learning rate is used for the whole duration of model training.

2. **Adaptive Gradient Descent (AdaGrad):** The key idea of AdaGrad is to have an adaptive learning rate for each of the weights. It performs smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequently occurring features.

3. **Adaptive Moment Estimation (Adam):** Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. Adam computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients, similar to momentum.

RESULTS:

Optimization technique	Epochs	Validation accuracy
1. SGD	14	88.3%
2. AdaGrad	8	88.8%
3. Adam	4	89.4%

CONCLUSION:

- The accuracy of the models used with all the optimization techniques is almost similar.
- The key takeaway is that to achieve the same validation loss, the time required by all the 3 models is very different.
- SGD takes the greatest number of epochs (and time), followed by AdaGrad, whereas Adam is the fastest in terms of model training.