Cins Face Recognition using Deep Learning Machine Learning Homework3

Erdoğan Abacı
150315025

erdoganabaci97@gmail.com

Computer Science Engineering
Celal Bayar University – Manisa/Turkey

June 01, 2019

Keywords: All image pixel, Face recognition, Face Dataset, Face estimation accuracy, Gradient Descent Solution(ADAM) , Image classification

Abstract

CNN(Convolutional Neural Network) with ANN(Artificial neural networks) are a machine learning method that can be used to solve a multitude of different tasks. In this report we strive to discuss some of the theoretical basics behind this technology and apply it to a practical problem in the field of digital image classification: Recognizing face from scanned face dataset.

We cover the entire process from finding, labelling and processing the data, over building the networks themselves, to optimizing their performance with different training techniques.

The Python-based implementation uses frameworks such as Keras(Background TensorFlow) and OpenCV to facilitate this process, and the output is presented in the format of faceJSON, a leading standard for exchanging musical notation. The data set was taken from the my webcam and faceset dataset.

1 Introduction

Face recognition has often been considered for surveillance environments as it can generally be conducted in a visible light camera environment. However, in a surveillance environment, most cases
involve a camera capturing images in a downward direction from above and people do not look directly at the camera. Thus, it is generally difficult to capture front facial images, and in such cases, facial recognition accuracy is greatly reduced. To address this issue, 105 composite geometrical descriptors for 3D face analysis based on 3D face data captured by a laser scanner were presented in a previous study.

Since the deep learning has dramatically drawn attention in the computer vision community, face recognition has been one of the most extensively studied topics to come up with the challenging of the recognition problem. Face recognition has used in several applications in both private and public sectors such as surveillance system, person authentication, etc.
Conventional face recognition method aims to develop the faster algorithm and more robust. Whereas, accurate recognition depends on high resolution of face image and with no occlusion. Good face image should be discriminative to the change of face identity while remains robust to the intra-personal variation. Although face recognition algorithms have reached the accuracy level under certain condition, the face recognition algorithm still affected by the external and internal variation such as the illumination, pose, facial expression and occlusion. The key challenge of face recognition is to develop effective feature representation to improve the accuracy in the different scenarios. Recently, deep learning has achieved great success on computer vision research with significant improving the state-of-art in classification and recognition problems.

2 **CNN**

Convolutional Neural Networks are very similar to ordinary Neural Networks Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.

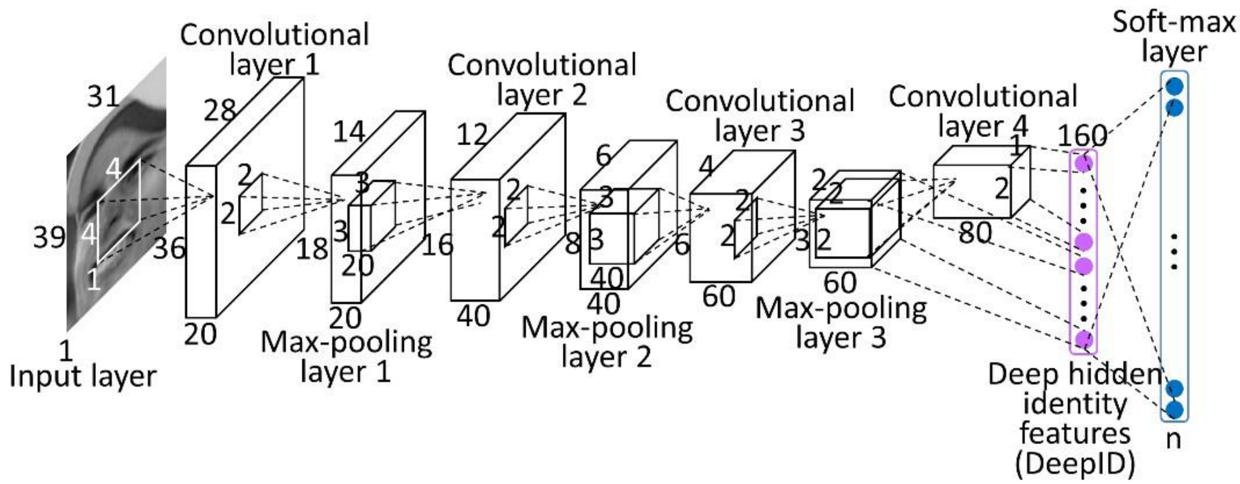


Figure 1 Convolution Layers

2-1 Activation Function

An activation function is a node that you add to the output layer or between two layers of any neural network. It is also known as the transfer function. It is used to determine the output of neural network layer in between 0 to 1 or -1 to 1 etc. When building a model and training a neural network, the selection of activation functions is critical. Experimenting with different activation functions for different problems will allow you to achieve much better results. I chose **relu** activation function because the derivative is easy to take

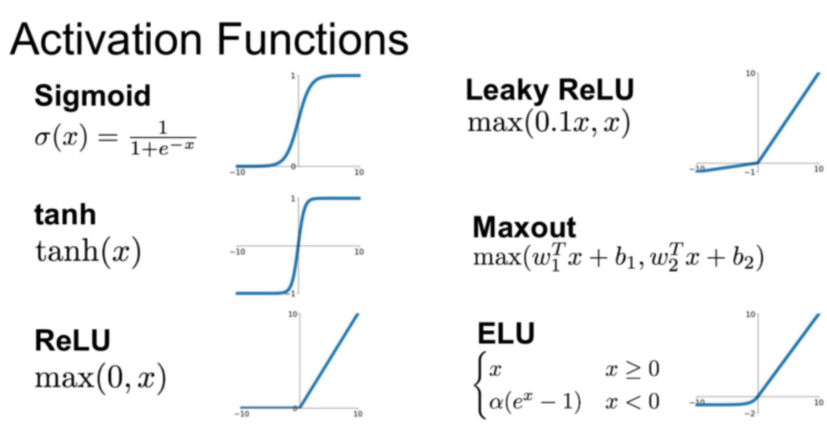


Figure 2 Activation Functions

2-2 Pooling Layer

The Pooling layer can be seen between Convolution layers in a CNN architecture. This layer basically reduces the amount of parameters and computation in the network, controlling overfitting by progressively reducing the spatial size of the network. There are two operations in this layer; Average pooling and Maximum pooling. Only Max-pooling will be discussed in this post.

Max-pooling, like the name states; will take out only the maximum from a pool. This is actually done with the use of filters sliding through the input; and at every stride, the maximum parameter is taken out and the rest is dropped. This actually down-samples the network.

Unlike the convolution layer, the pooling layer does not alter the depth of the network, the depth dimension remains unchanged.Max pooling purpose is take **the largest element** from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

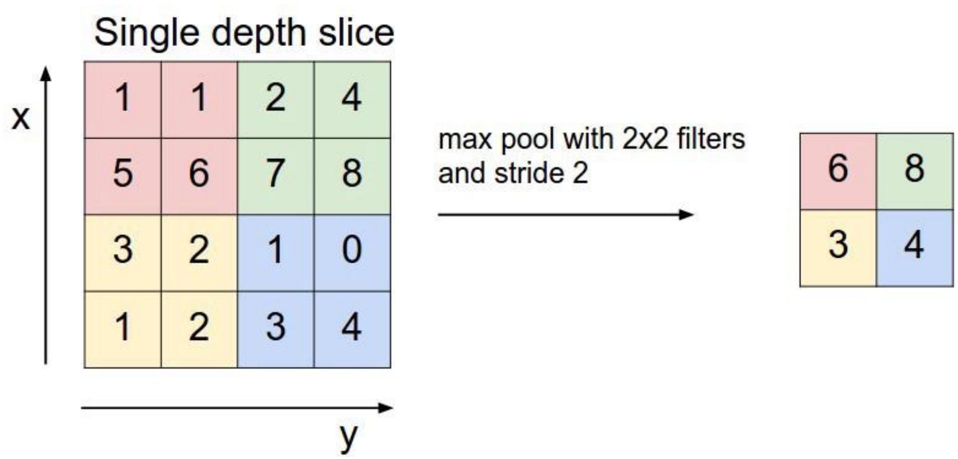


Figure 3 Max Pooling

2-3 Learning Rate

Learning rate is defined in the context of optimization, and minimizing the loss function of a neural network. You define a cost function for a neural network, and the goal is to minimize this cost function. For this optimization problem, we use gradient descent or other variants of it where the model paramaters (here weights and biases in the network) are updated in a way to decrease the cost function.

I use **Adam(adaptive momentum)** learning optimization rate because adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it’s called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network.

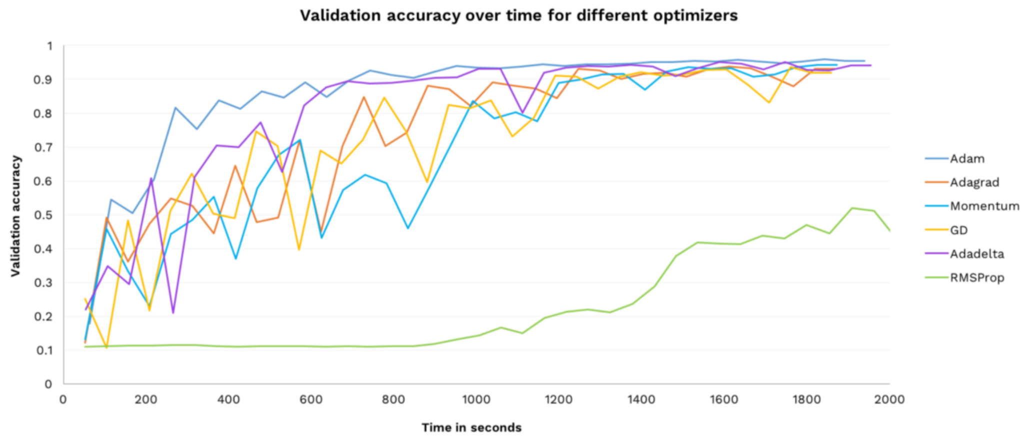
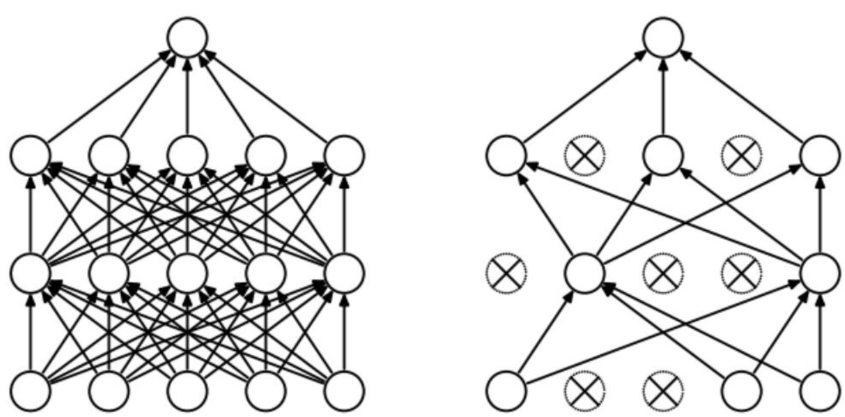


Figure 4 Adam Optimization Learning Rate

2-4 Dropout

Dropout is a regularization technique patented by Google for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units in a neural network.

Why do we need Dropout?.Because **prevent over-fitting.** In this reason I use dropout.



(a) Standard Neural Net (b) After applying dropout.

Figure 5 Dropout

2-5 Train and Loss

Loss function is the function that measures the error rate and performance of the designed model. The last layer of deep networks is the layer on which the loss function is defined. Since the loss function performs the error calculation by converting the problem into an optimization problem, the objective function, which is used in optimization terminology, is also defined by the names of cost function. The loss function basically calculates how different the model's estimate is from ground truth.

Therefore, if we do not create a well-predicted model, the difference between the ground value and the predicted value will be high, so the loss value will be high, and if we have a good model, the loss value will be less. If it is exactly the same, loss will be 0. We expect a good model to have a loss value close to 0. Why not 0? because **regularization**.

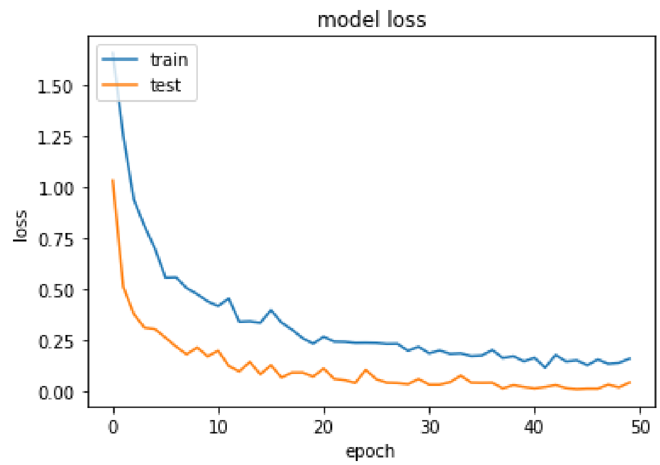


Figure 6 Model Loss

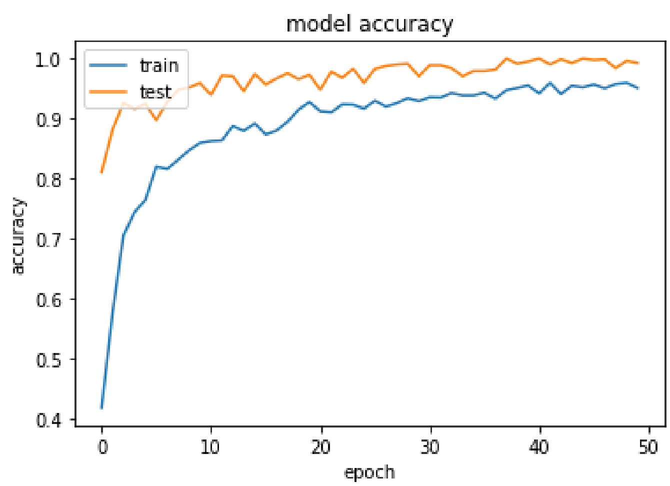


Figure 7 Model Accuracy

2-6 Backpropagation

As these assessments will change for each home, most houses are However, a neural network has to be trained before it can be used for classification. For this purpose, an algorithm called back-propagation is used. First, the errors that occurred in the output units are determined. Those errors are then passed backwards, first to the hidden layer and then to the input layer. This way, the amount of "blame" that every connection holds for the error, can be determined.

In the last step, a **gradient descent search** is performed to find better weights that minimize the error. When this has happened for every set of data in the training set, one training epoch has passed, and the performance can be evaluated again.

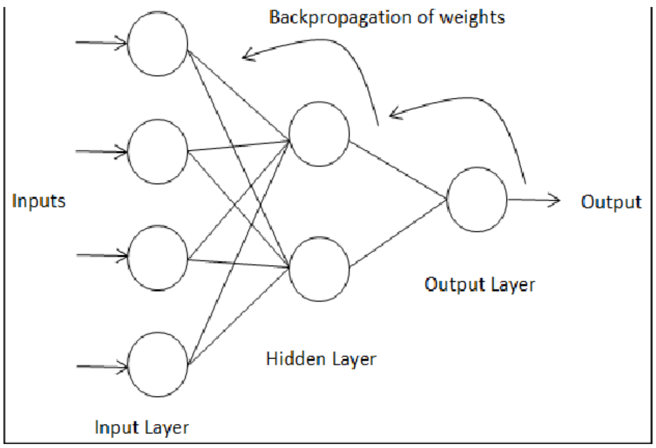


Figure 8 Backpropagation

3 The Dataset

While data on actual face recognition is the ideal information for estimate large data, The images were imported into Python using OpenCV. The resulting objects were NumPy 50 × 50 arrays that could easily be flattened to 2500 × 1 arrays and used to build Facedata set in the web and I take a picture and collect myself. 70% of the images were used to build the training set, the remaining 30% for the test set. However, the distorted images were added to the training set afterwards, increasing its size fourteenfold.

```
[[[[0.48235294]
   [0.46666667]
   [0.45882353]
   ...
   [0.43921569]
   [0.44313725]
   [0.44705882]]

  [[0.49803922]
   [0.4627451 ]
   [0.44705882]
```

Figure 9 Image Pixel

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

Figure 10 Y train categorical

3-1 Image Augmentation(image extend)

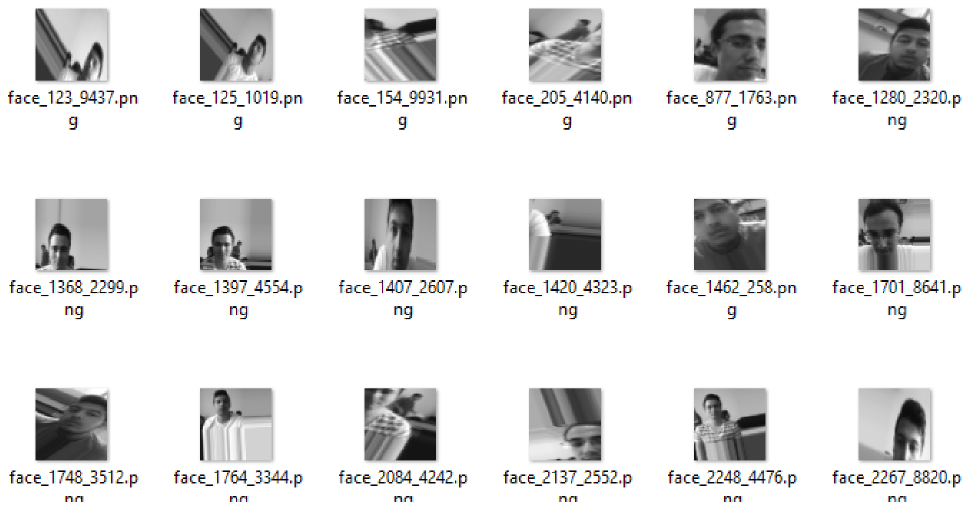


Figure 11 My Augmentation Data

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.I prevent to over-fitted

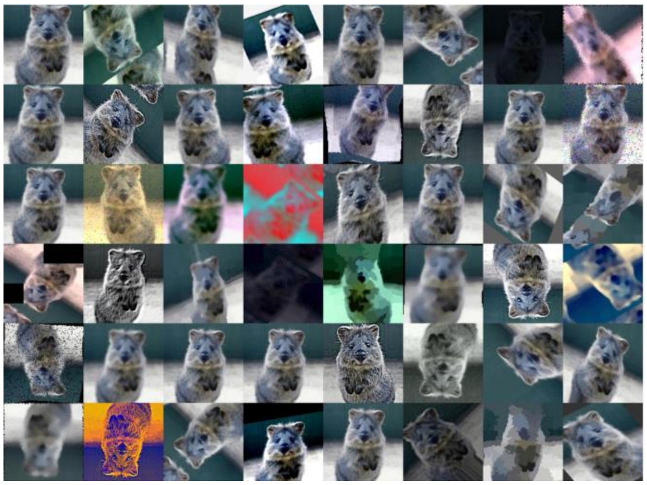


Figure 12 Image Augmentatio

4 Results

The cnn(deep learning) process uses tensorflow backend within the sample for tuning 50x50 pixels of the determine minimum weight, which allows for an ideal compromise between minimizing loss and accuracy.

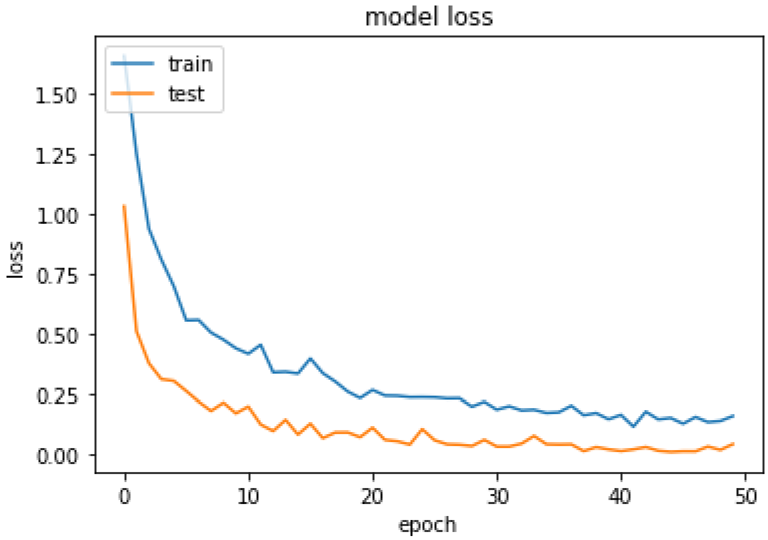


Figure 13 Model Loss

The results show good train and loss error, if we change cnn filter and epoch and batch size,it will give a better result.As I mention above,process the counting weight loss is decreasing.Activation function is important because when we make backpropagation to determine weight

,computer challenge with derivation every epoch.In this reason I use relu activation function and every step 32 image calculate which means that batch size.

```
Test Accuracy : 0.9925925922982487
Test Loss : 0.04021810648249991
```

Figure 14 Train Test Accuracy

When the program classified correctly me from others , program output say figure 13 ,otherwise acces denied,who are you?

```
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
```

Figure 15 I Recognize You

5 Conclusion and Discussion

In this paper, we proposed a framework for real-time multiple face recognition. The recognition algorithm based on CNN which is the optimization algorithm. The framework consists of the tracking technique and using the minimal weight of the model. This can reduce the processing time and network parameter to learn recognize multiple **face feature in real-time**. The framework is implemented into NVIDIA board which supports CUDA, the advantage of the board is able to run the process in parallel. As a result, the processing time is faster and suitable for real-time multiple **face recognition with acceptable recognition rate.**

References

Face Rec. (2019). Face Rec Web Site: http://www.face-rec.org/databases/

Github. (2019). Github Web Site: http://cs231n.github.io/convolutional-networks/

Machine Learning Mastery. (2019). Machine Learning Mastery Web Site: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

Medium. (2019). Medium Web Site: https://medium.com/octavian-ai/which-optimizer-and-learning-rate-should-i-use-for-deep-learning-5acb418f9b2

Towardsdatascience. (2019). towardsdatascience Web Site: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c