Final Project

Mobile Eye Gaze Estimation with Deep Learning

Wenting Li

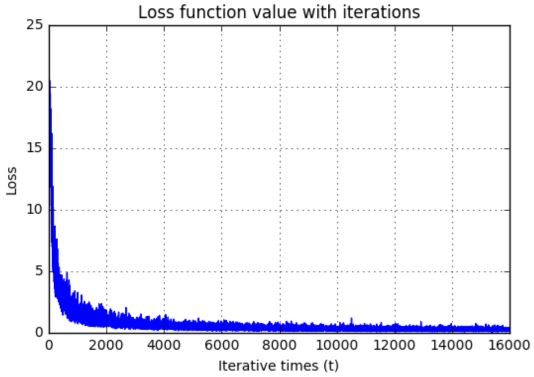
1. Parameters:

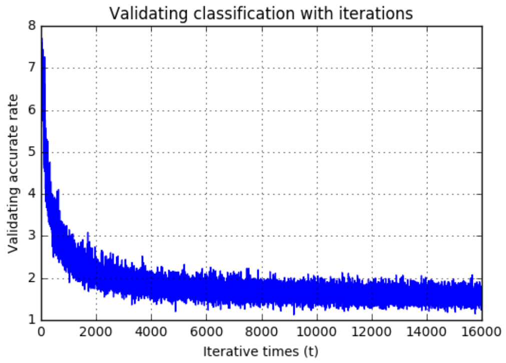
Learning_rate=0.001 (decay=0.8,momentum=0.0005);

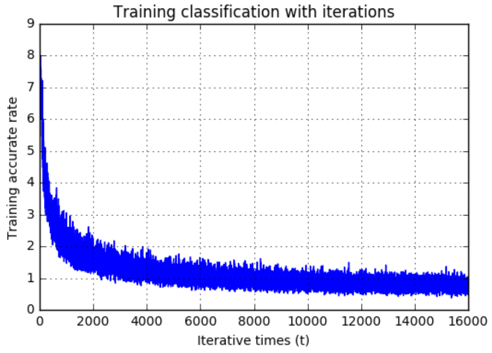
Batch_size=128 (due to the capacity of my computer only a small batch_size is selected);

Iteration=160000;

2. Loss and accuracy plots







3. Visualization of the model architecture

1. Four-pathway CNN model is constructed with the input of ‘left eye, right eye, face and face_mask’ training data, the specific parameters are shown in Table I and the visualization is in Figure 1.
2. For the input of ‘eye_left, eye_right, face’, there are four layers CNN for each of them. These four layers have the same architecture as follows:

Table I : Architecture parameters of four-pathway CNN model

Layer Name	Type	Kernel Size	Stride	Padding	Output Size
Conv1	Convolutional	5×5	2	0	64 @ 30×30
Pool1	Max pooling	3×3	2	0	64 @ 14×14
Conv2	Convolutional	5×5	1	2	64 @ 14×14
Pool2	Max pooling	3×3	2	0	64 @ 6×6
Conv3	Convolutional	3×3	1	1	128 @ 6×6
Conv4	Convolutional	1×1	1	0	64 @ 6×6
Fc1_e	Fully Connected	-	-	-	128
Fc1_f	Fully Connected	-	-	-	128
Fc2_f	Fully Connected	-	-	-	64
Fc1_fm	Fully Connected	-	-	-	256
Fc2_fm	Fully Connected	-	-	-	128
Fc1	Fully Connected	-	-	-	128
Fc2	Fully Connected	-	-	-	2

3. The weights and biases for the ‘eye_left’ and ‘eye_right’ are shared and after these four layers, the output ‘conv4_eye_left’ and ‘conv4_eye_right’ are concatenated in the dimension of size of kernel (or in the dimension of 64); then the size of this concatenation is 128 @ 6×6 following with a fully connected layer.
4. The output of the four layers CNN for the input of ‘face’ is firstly flattened to a fully connected layer and another fully connected layer with a half size;
5. The input ‘face_mask’ is directed flattened to a fully connected layer and then another fully connected layer to reduce the size to the half;
6. Each convolutional layer and fully connected layer is followed by a activation function ‘RELU=max(0,x)’;
7. The parameters of this architecture refer to the ALexNet in [1] and the architecture is based on the four-pathway in [2], but there are some differences between my model and the existed ones:
    a. The first convolutional layer has a smaller kernel size 5×5, and the stride is 2 smaller than 4 in the paper;
    b. The number of kernels in all of the convolutional layers are all smaller than that in the paper;
    c. The output of the fourth convolutional layer of the input of ‘eye_left’ and ‘eye_right’ has an activation function ‘RELU’, and then they are concatenated, but in the paper they concatenate the ‘eye_left’ and ‘eye_right’ before the activation function.
    d. In addition, the architecture in the paper exploits local response normalization to generalize from the training data, but I found adding this part cannot improve the final error and thus ignore it.
8. Although the graph is defined by my codes, there are two functions ‘strip_consts’ and ‘show_graph’ used in my codes to show the graph directly come from website as follows: http://stackoverflow.com/questions/38189119/simple-way-to-visualize-a-tensorflow-graph-in-jupyter.

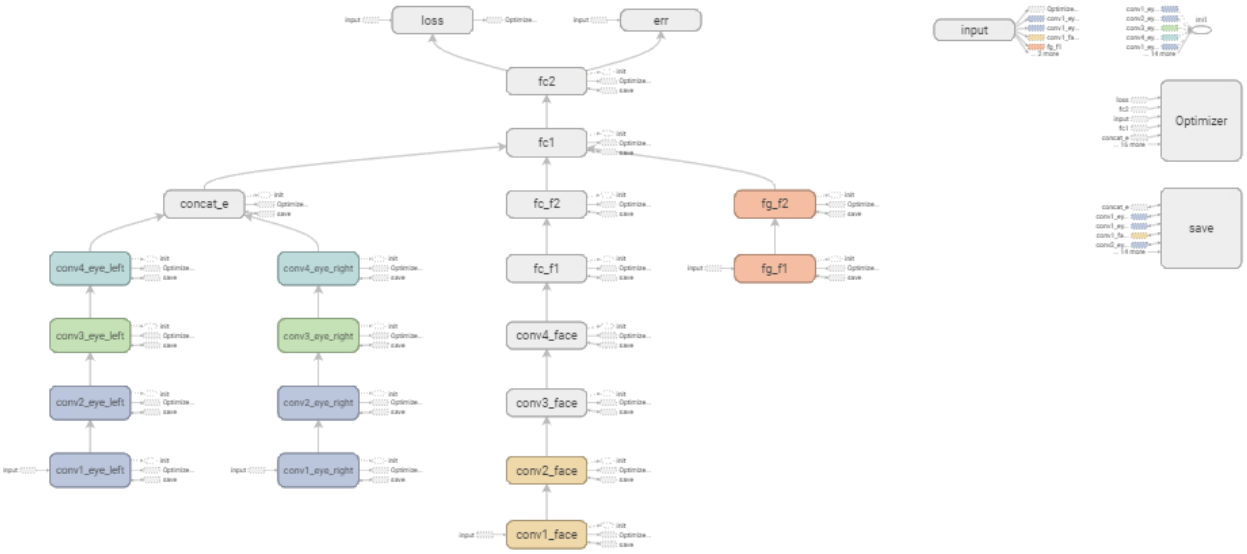


Figure 1: Visualization of the four-pathway architecture CNN

4. Explanation and justification

1. **The final error** of this model after 16000 iterations is **1.54**;
2. The reasons are as follows:
   a. Due to the size of input image is 64×64, I choose a small size of kernel 5×5 instead of 11×11 for the first convolutional layer, meanwhile, the stride is also reduced from 4 to 2;
   b. The second convolutional layer has the same kernel size 5×5 with the first convolutional layer to further suppress the useless parts for gaze prediction; This is one critical improvement. The idea behind is based on the architecture of VGG: using many smaller but same size kernels instead of a larger one. This can improve the nonlinearity of the kernels, since more RELU can be employed. This architecture work well especially for location issue [3].
   c. There are two maximum pooling layers to reduce the redundancy. In fact, I also tested the results if using average pooling and found max pooling can have a better result;
   d. The third and fourth convolutional layers using smaller kernel size to reduce the spatial convolution.
   e. Four-pathway CNN has a better performance than one-path. The outputs of the convolutional layers of the left and right eye, face and face grid are concatenated together to improve the accuracy. The reason why this concatenation can improve the accuracy is that these four types of inputs are correlated and concatenating them together can extract more information. Details about the comparison of the one-pathway and four-pathway architectures is in Section 1.5.
   f. For the three types of inputs (eye_left, eye_right and face), there are four CNN layers, but for the input of 'eye_left' and 'eye_right', they share the same weight and biases, since they are likely to focus on the same direction;

g. For the input of face grid, there is no need to add CNN layers, since the binary image is too simple. Only two connected layers together with the activation function RELU are able to suppress the not important parts;

5. Architecture comparison

5.1 Common Parameters between one-pathway and four-pathway CNN architecture:

Learning rate=0.001;
Decay=0.8;
Momentum=0.0005;
Batch_size=128;

5.2 Architecture visualization of the one-pathway CNN

One-pathway with four layers CNN is constructed with the input of 'left eye' training data, the specific parameters are shown in Table II.

Table II: Architecture parameters of the one-pathway CNN

Layer Name	Type	Kernel Size	Stride	Padding	Output Size
Conv1	Convolutional	5×5	2	0	64 @ 30×30
Pool1	Max pooling	3×3	2	0	64 @ 14×14
Conv2	Convolutional	5×5	1	2	64@ 14×14
Pool2	Max pooling	3×3	2	0	64@ 6×6
Conv3	Convolutional	3×3	1	1	128 @ 6×6
Conv4	Convolutional	1×1	1	0	64 @ 6×6
Fc1_e	Fully connected	-	-	-	$6 \times 6 \times 64$
Fc1	Fully connected	-	-	-	2

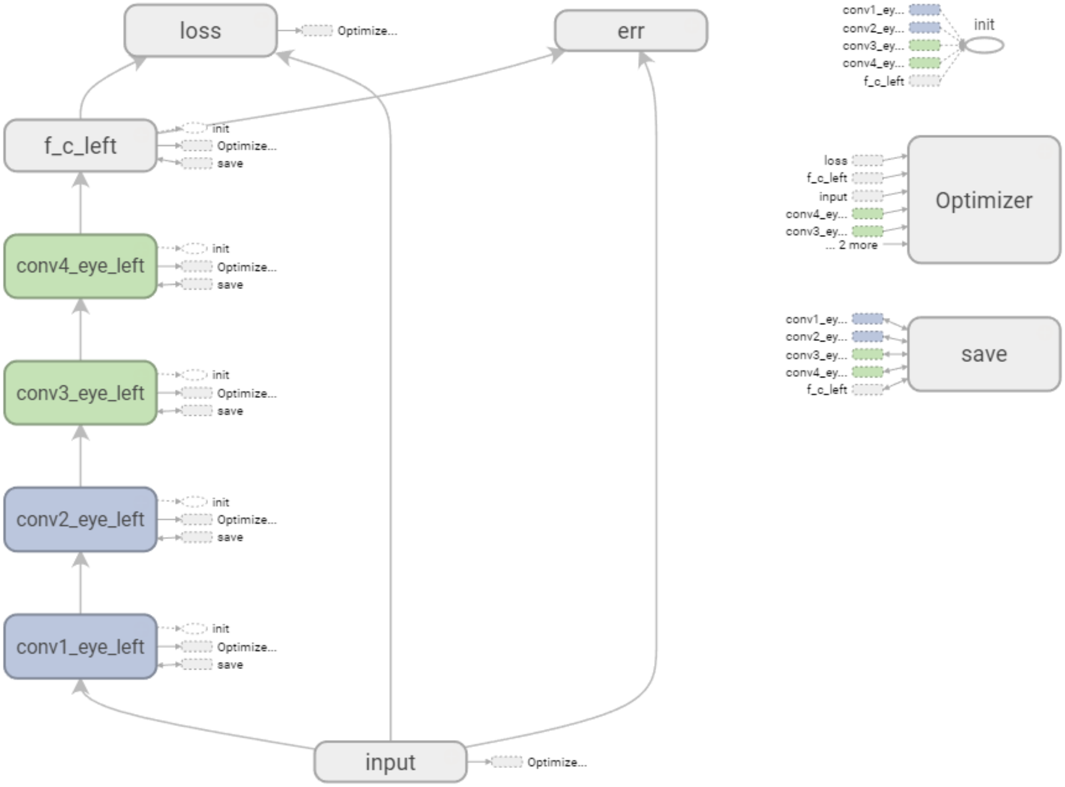
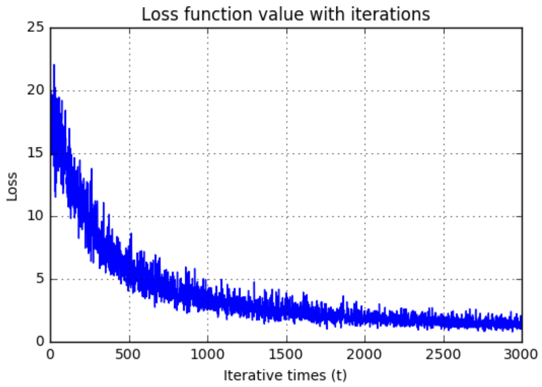
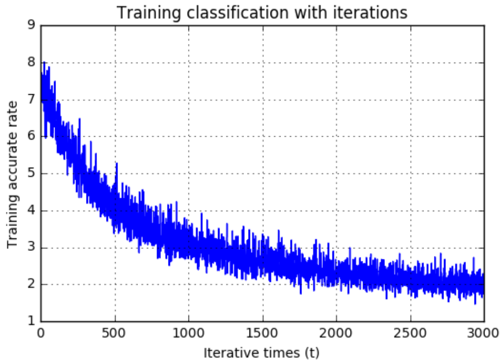
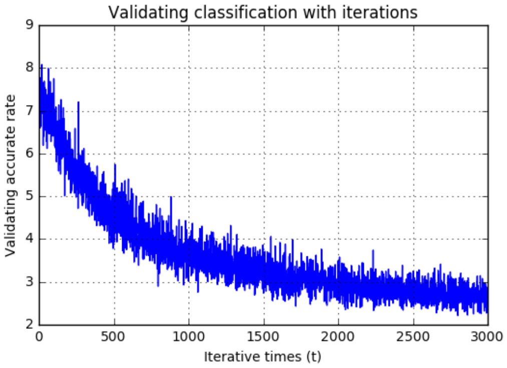


Figure 2 One-pathway graph tensorboard visualization

5.3 Loss and accuracy plots of one-pathway CNN







5.4 Performance Comparison

Table III Comparison of one-pathway and four-pathway CNN for eye tracking

	One-pathway CNN	Four-pathway CNN
Best accuracy	2.63	1.54
Computation time	40min /3000 (0.013 min)	512min /16000 (0.032 min)
Computer configuration	Dell i7 CPU with 16G RAM	Dell i7 CPU with 16G RAM

For the one-pathway CNN, after 3000 steps, the loss is converged and reaches a final error 2.63, but four-pathway CNN can achieve a lower estimation error 1.54 after 16000 iterations, although the time is much longer. Actually, another test experiments of four-pathway CNN reaches a final error of 1.61 after 7000 iterations. The time for per iteration is 0.014 min (100min /7000). Thus after a long time the computation speed declines gradually. Generally, the speed of four-pathway CNN is satisfactory. Therefore, the four-pathway CNN performance better in the whole performance.

Reference

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012.

[2] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., & Torralba, A. (2016). Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2176-2184).

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).