SACCADE GAZE PREDICTION USING A RECURRENT NEURAL NETWORK

Thuyen Ngo, B.S. Manjunath

Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA
{thuyen, manj} @ ece.ucsb.edu

ABSTRACT

We present a model that generates close-to-human gaze sequences for a given image in the free viewing task. The proposed approach leverages recent advances in image recognition using convolutional neural networks and sequence modeling with recurrent neural networks. Feature maps from convolutional neural networks are used as inputs to a recurrent neural network. The recurrent neural network acts like a visual working memory that integrates the scene information and outputs a sequence of saccades. The model is trained end-to-end with real-world human eye-tracking data using back propagation and adaptive stochastic gradient descent. Overall, the proposed model is simple compared to the state-of-the-art methods while offering favorable performance on a standard eye-tracking data set.

Index Terms— Eye tracking, scanpath, gaze, fixations

1. INTRODUCTION

Due to sensory and computational limitations, humans and many other animals employ visual attention as the strategy to actively explore the environment. Human visual system only has high visual acuity in a small region, the fovea, and the photoreceptor density drops rapidly when moving away from the fovea. When a human observer gazes at a point, the fixated region is projected onto the fovea and sampled with highest density. The peripheral on the other hand is perceived with low resolution. This helps reduce the amount of information the brain needs to process at a given time but it requires the eyes to move constantly to integrate the information of the entire scene. The mechanisms which control such eye movements have been extensively studied in psychology and neuroscience [1].

A similar computational bottleneck exists in computer vision where processing the entire image might be prohibitively expensive. For example, the popular deformable part model for object detection [2] takes a few seconds to process a single image as it uses scanning windows over the whole image. In face recognition or image classification with convolutional neural networks, the input image normally needs to be cropped so that objects are aligned roughly at the center of the image. It could be advantageous in these cases to have an attentional model to select meaningful regions to process. Toward this, recent work in computer vision has focused on (1) saliency models, which predict the probability map of fixations (a comprehensive review of saliency models can be found in [3]); and (2) objectness measures [4, 5], where potential regions containing objects are selected. However, these models ignore the sequential nature of visual attention, which could be valuable information for visual search and large scale image analysis.

Predicting the fixation sequence is quite challenging and has not received much attention in computer vision. The first model [6] simply uses winner-take-all and inhibition of return schemes to output a sequence of winners from a static saliency map. In [7, 8] authors use stochastic models for scanpath based on Levy distribution of saccades magnitudes. [9] exploits three factors that might guide sequential eye movements: reference sensory responses, fovea periphery resolution discrepancy, and visual working memory. The first learning based method is introduced in [10] by integrating semantic information along with Levy flight and saliency map into a Hidden Markov Model. And recently reinforcement learning has been applied to learn the induced reward function [11] or the fixation policy [12] to obtain state of the art performance.

In this work, we utilize both CNN and RNN's power and simplicity to model fixation sequences. Compared to the state of the art [12] and other previous work [10, 3], our model offers several advantages. First, it does not assume any prior knowledge about the data. Most previous works integrate some understanding about eye tracking data in the models. For example, the center bias and Levy flight distribution of eye movements have been used either as features or priors in [12, 10, 3]. The model in [12] is even trained with different set of features for different eye tracking datasets. Thanks to end-to-end training, we expect our model to learn such knowledge from the data itself. Second, the proposed method unifies the feature extraction steps into a single pass through a CNN. Other methods normally need to extract low level features such as edges or color features and then compute semantic features using object detectors.

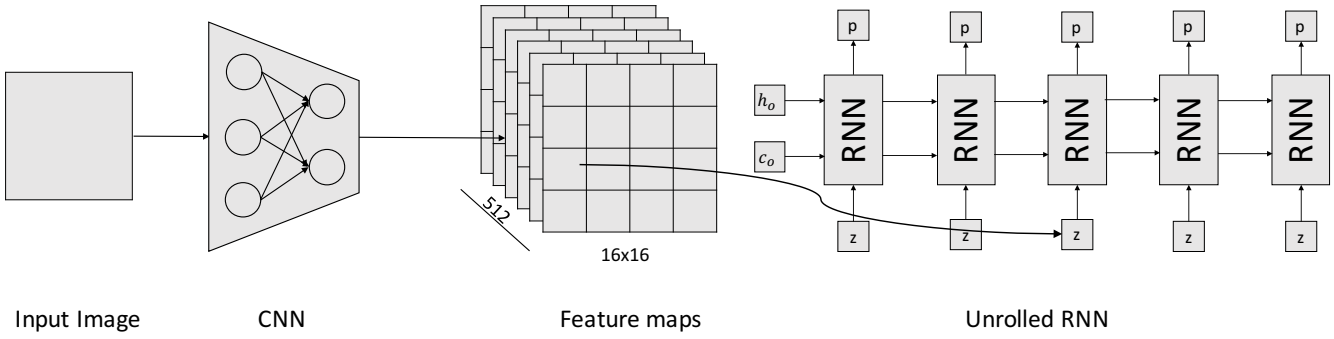

Fig. 1. The overall of our model. Given an image, feature maps from CNN are extracted to feed into a RNN.

2. APPROACH

The overall approach is shown in Figure 1. Given an image, we first extract features from a pretrained CNN. These features are then used as inputs into a RNN to make predictions about the fixation locations. We now will discuss in detail each component of our model.

2.1. CNN Feature extraction

Many recent works exploited transfer learning or domain adaptation using pretrained CNNs. The idea is that knowledge gained from training millions of images for classification could be used for different tasks with limited labeled data. The pretrained features from convolutional neural network have been shown to be effective for many different tasks ranging from fine-grained image classification [13], image segmentation [14] to image caption generation [15]. These features contain not only the semantic information of the images as a whole, but also the locations of such information [16], which is a desirable property for our model. We modify the original 16-layer VGG network [17] to retain only convolutional and pooling layers. This helps maintain the spatial information in the extracted feature maps and allows to run the network with different image sizes. In our experiments, we use input size of 512x512 and the resulting output feature maps are of size 16x16.

2.2. Spatial Quantization

In line with the CNN feature extraction, we spatially quantize the input image into 256 regions by a 16x16 grid. Each region could be represented by a 512-dimensional feature vector from the CNN feature maps. At the same time eye tracking fixation data are binned into those regions. Each fixation is now represented by the center of the region it is in, and a sequence of fixations is a sequence of jumps from one region to another. This greatly simplifies the modeling process compared to other models where fixation locations are at superpixel level [10, 12].

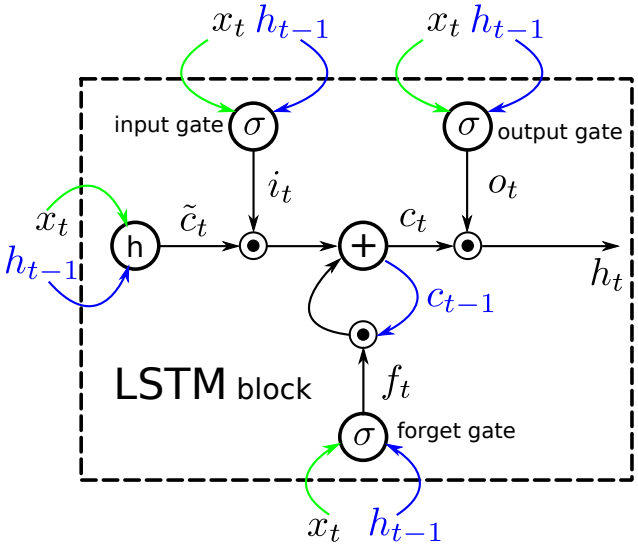

Fig. 2. Illustration of a LSTM block. The green connections are feedforward with associated weight matrices. The blue are feedback weight connections. Best viewed in color.

2.3. Long Short Term Memory (LSTM)

LSTM [18] is designed to mitigate the vanishing and exploding gradients during training of recurrent neural networks, and has been widely used for sequential modeling [19, 15]. Training a traditional RNN could be difficult because the gradient signal is multiplied many times by the recurrent weight matrix during back propagation. If the weights are small, the resulting gradient signal could be so small that learning will become either too slow or even stop working (vanishing gradients). On the other hand, if the weights are large, the gradient signal could end up being too big and cause the learning to diverge (exploding gradients). LSTM aims to fix these issues by introducing a modular structure, referred to as a memory cell. A memory cell includes an input gate, an output gate, a forget gate and a self-recurrent connection. The gates control how the cell modulates its dynamics and its interactions with

the input and the output. The forget gate modulates whether the cell should remember or forget its previous state. The input gate controls how much of the input would have an effect in the cell, and the output gate controls the effect of the cell at the output (on other neurons).

An example of a LSTM block is shown in Figure 2 and its main computations with input x_t, output h_t and its recurrent cell c_t are as follows:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{1}$$
$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2}$$
$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3}$$
$$\tilde{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{5}$$
$$h_t = o_t \odot c_t \tag{6}$$

The gates $\{i_t, f_t, o_t\}$ are perceptron units with sigmoid activation σ. $\{W_i, W_f, W_o, W_c\}$ and $\{U_i, U_f, U_o, U_c\}$ corresponds to the weight matrices for the feedforward and feedback connection; $\{b_i, b_f, b_o, b_c\}$ are biases. The gates' modulation operator $\odot$ is element-wise multiplication.

2.4. Scanpath Model with LSTM

The key observation is that saccade planning is not memoryless, i.e. it is influenced by the gaze history [20]. Such visual working memory could be naturally modeled using an LSTM. In our case, the LSTM models the conditional transition probability of the next fixation given the current fixation and the history of information it has seen so far. The history is expected to be remembered in the state vectors c_t and h_t. The transition probability p_t is computed as follows:

$$z_t = \text{CNN}(S_t) \tag{7}$$
$$x_t = \text{FF}_\text{x}(z_t) \tag{8}$$
$$h_t = \text{LSTM}(x_t, h_{t-1}, c_{t-1}) \tag{9}$$
$$y_t = \text{FF}_\text{y}(h_t) \tag{10}$$
$$p_t = \text{softmax}(y_t) \tag{11}$$

Here we use CNN to extract the feature z_t at the current location S_t. FFs are one-layer perceptrons: the first one is used to map the dimension of z_t to the number of LSTM hidden units and the second is used to map LSTM output to the distribution over possible locations:

$$P(S_{t+1}|S_t, c_{t-1}, h_{t-1}) = p_t \tag{12}$$

In order to predict the first fixation, the initial states of the LSTM is also computed from the CNN features using one-layer perceptrons:

$$h_o = \text{FF}_\text{h}(\text{CNN}), c_o = \text{FF}_\text{c}(\text{CNN}) \tag{13}$$

2.5. Learning

We aim to estimate parameters θ of the model, including all weight matrices and biases, from eye tracking data. The log-likelihood of one fixation sequence $S = \{S_2, S_3, \ldots, S_{n+1}\}$ given an image I:

$$L(S|I;\theta) = \sum_{t=1}^{n} \log P(S_{t+1}|S_t, c_{t-1}, h_{t-1}) \tag{14}$$
$$= \sum_{t=1}^{n} \log p_t(S_{t+1}) \tag{15}$$

The model is optimized so that the log-likelihood over all training samples is maximized:

$$\theta^\star = \arg\max_\theta \sum_{(I,S)} L(S|I;\theta) \tag{16}$$

The equivalent form of this is the minimization problem on traditional log loss (negative log-likelihood). From Figure 1 we can see that the computation flow of equation 15 for each sequence is actually a DAG and thus the learning can be performed using standard back propagation.

2.6. Sequence Prediction

There are multiple approaches that can be used to generate a sequence from the trained model. The simplest prediction scheme is to sample the most probable location given the current location S_t and the states of the LSTM:

$$S_{t+1} = \arg\max_{L_{t+1}} P(L_{t+1}|S_t, c_{t-1}, h_{t-1}) \tag{17}$$

We sample the first location according to p_1. Given the feature at that location we can sample from p_2, and continue until we reach some pre-defined sequence length. This, however, is far from optimal since the error from one time step could easily be propagated to the next.

The optimal strategy would be searching for the sequence with maximum likelihood based on equation 15. However, this is too expensive because of the exponential growth of the search space with respect to sequence length. Instead we use beam search to generate m best sequences with largest likelihoods. We do this by always maintaining m best candidate sequences at each step. At the end of the step, each candidate will have 256 children for the following locations. Among all resulting sequences we choose m of them with the maximum likelihood. Finally, we return the best among m candidates as the predicted sequence. We use the beam search in the our experiments with a beam of size 20.

3. EXPERIMENTS

3.1. Evaluation metric

We use the metric proposed by [3] to evaluate consistency of eye tracking sequences. For each image, fixations from all human subjects are clustered using meanshift clustering with an optimal bandwith that maximizes the interaction rate I among clusters. Similar to [12], the interaction rate is defined as:

$$I = \frac{N_b - N_w}{C} \tag{18}$$

N_b is number of fixation transition between clusters, N_w is number of transitions within clusters and C is the number of clusters. Each cluster is then associated with a character. A sequence of fixations is now represented as a string of characters and Needleman-Wunsch string matching algorithm [21] can be used to to compute the distance between any two sequences.

A predicted sequence is compared to data from all of the human subjects and scores are averaged to obtain the final score. We also compute scores between any two subjects and average them to get the upper bound for the performance.

3.2. Dataset

We evaluate our model using the MIT [22] dataset. The dataset contains 1003 images with various types of object categories. Eye tracking data was collected during a free viewing task with 15 subjects per image, with a total of about 15,000 sequences. This is currently the largest free-viewing eye-tracking dataset available for natural images.

3.3. Training Details

Due to limited training data, the main challenge is to deal with overfitting. To reduce overfitting, we only train the RNN part of the model, leaving the CNN part untouched (no fine-tuning). We do a simple data augmentation with horizontal flipping to create more training examples. Even though eye-tracking data are not absolutely invariant to this transformation (humans might not look at flipped texts for example), we observe slight improvement in the results. We also try to keep the number of model parameters small, limiting the number of hidden units of the LSTM to only 64. The model is trained with dropouts [23] using rmsprop [24], a variant of adaptive stochastic gradient descent, without using momentum term. The LSTM weights are initialized with random orthogonal matrices, and the remaining weights are randomly initialized with a small variance.

3.4. Results

It was first shown in [3] that when using winner-take-all method (WTA) to compute fixation sequences, GBVS [25] and Judd [22] saliencies perform significantly better than other saliency models. Recently [12] has shown that SVM saliency models like Judd's with WTA fixations can perform comparably with state of the art [12] itself. Since we were not able to get access to the evaluation codes of the referenced models [9, 10, 12], we only compare our models with WTA using Judd and GBVS saliencies. We randomly split the data, using 90% for training and 10% for testing. We retrain the SVM saliency model using the same training data. The comparisons are limited to 6 fixations for each sequence (similar to [12]). Figure 3 shows the performance of three models on the test set. The inter-subject performance is calculated over the whole dataset. Our method performs better than WTA on SVM (Judd's saliency).

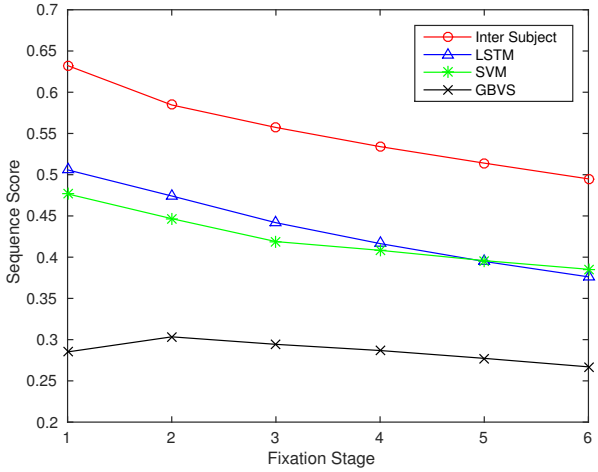

Fig. 3. Evaluation of the RNN model (blue triangle) and baseline models GBVS [25] (black cross) and Judd [22] (green star) and inter-subject performance (red circle) on the MIT dataset. Best viewed in color.

4. CONCLUSION

We have presented a model to predict a sequence of fixations that humans are likely to look at in a given image (in free-viewing task). We present a simple framework to model sequences with recurrent neural networks using localized features extracted from a pre-trained convolutional neural network. The model is trained to maximize the likelihood of fixation sequence given an image using free-viewing human eye-tracking data. Despite its simplicity, we achieve favorable performance compared to more complicated methods. With recent advances in eye-tracking technology, we would expect better performance of our model when large eye-tracking datasets become available in the future. Exploring a similar framework to predict dynamic gaze in videos will have interesting applications in video object tracking and human assisted annotations of large data sets.

5. REFERENCES

[1] Sabine Kastner Ungerleider and Leslie G, "Mechanisms of visual attention in the human cortex," *Annual review of neuroscience*, vol. 23, no. 1, pp. 315–341, 2000.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] A. Borji, H.R. Tavakoli, D.N. Sihite, and L. Itti, "Analysis of scores, datasets, and models in visual saliency prediction," in *ICCV*, Dec 2013, pp. 921–928.

[4] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, "Measuring the objectness of image windows," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[5] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014, pp. 3286–3293.

[6] Laurent Itti, Christof Koch, and Ernst Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 11, pp. 1254–1259, 1998.

[7] Dirk Brockmann and Theo Geisel, "The ecology of gaze shifts," *Neurocomputing*, vol. 32, pp. 643–650, 2000.

[8] Giuseppe Boccignone and Mario Ferraro, "Modelling gaze shift as a constrained random walk," *Physica A: Statistical Mechanics and its Applications*, vol. 331, no. 1, pp. 207–218, 2004.

[9] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao, "Simulating human saccadic scanpaths on natural images," in *CVPR*, 2011, pp. 441–448.

[10] Huiying Liu, Dong Xu, Qingming Huang, Wen Li, Min Xu, and S. Lin, "Semantically-based human scanpath estimation with hmms," in *ICCV*, Dec 2013, pp. 3232–3239.

[11] Stefan Mathe and Cristian Sminchisescu, "Action from still image dataset and inverse optimal control to learn task specific visual scanpaths," in *Advances in neural information processing systems*, 2013, pp. 1923–1931.

[12] Ming Jiang, Xavier Boix, Juan Xu Gemma Roig, Luc Van Gool, and Qi Zhao, "Learning to predict sequences of human visual fixations," *IEEE Transactions on Neural Networks*, 2015.

[13] Niko Sunderhauf, Christopher McCool, Ben Upcroft, and P Tristan, "Fine-grained plant classification using convolutional neural networks for feature extraction," in *Working notes of CLEF 2014 conference*, 2014.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, June 2015, pp. 3431–3440.

[15] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, "Show and tell: A neural image caption generator," in *CVPR*, June 2015, pp. 3156–3164.

[16] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler, "Efficient object localization using convolutional networks," in *CVPR*, June 2015, pp. 648–656.

[17] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015.

[18] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3104–3112.

[20] Paul M Bays and Masud Husain, "Active inhibition and memory promote exploration and search of natural scenes," *Journal of Vision*, vol. 12, no. 8, pp. 8, 2012.

[21] Saul B Needleman and Christian D Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[22] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba, "Learning to predict where humans look," in *CVPR*, 2009, pp. 2106–2113.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] Tijmen Tieleman and Geoffrey Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," in *Coursera: Neural Networks for Machine Learning*, 2012.

[25] Jonathan Harel, Christof Koch, and Pietro Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2006, pp. 545–552.