MATH6380P Project 3

Imitating Human Vision: A Study of the Recurrent Attention Model

HUANG Yifei, SUN Jiaze, DENG Yizhe, TAN Haiyi Department of Mathematics, HKUST

Introduction

When we examine static pictures or scenes, our visual perception consists of two kinds of eye movements that allow us to digest visual information effectively :

- **Fixation**: This represents moments when both eyes rest and focus on a single small portion of the scene;
- **Saccade:** This is a quick, simultaneous movement of both eyes between two points of fixation, or in other words, a 'jump' between two foci.

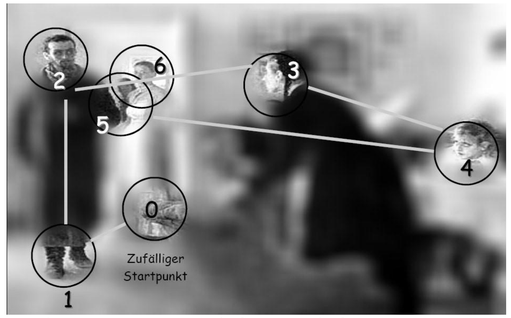

Figure 1. Eye movements in the first 2 seconds of examining a scene.

By combining series of fixations and saccades, our visual perception allows us to:

- Filter out **redundant information** in the scene;
- Quickly locate relevant information which **speeds up our perception and understanding**.

In computer vision, the **Recurrent Attention Model (RAM)** uses reinforcement learning to mimic these advantages. In this project, we aim to evaluate its effectiveness.

Data

In addition to the original MNIST data set, we tested our model with two modified versions: ***translated*** and ***cluttered***. Both data sets consist of 60x60 images, which are illustrated below. Since the images are larger, the relevance of RAM becomes significant in this situation.

Figure 2. Examples of the three versions of the number "6": original (left), translated (middle), and cluttered (right).

Recurrent Attention Model (RAM) Architecture

There are three components in RAM, a **Glimpse Sensor**, a **Glimpse Network**, and a **Recurrent Neural Network (RNN)**. The three components are integrated together using reinforcement learning. In particular, at glimpse t:

- **Glimpse Sensor**: The image patch at location l_{t-1} is extracted, and is represented by $\rho(x_t, l_{t-1})$;
- **Glimpse Network**: A feature vector $g_t = f_g(l_{t-1}, \rho(x_t, l_{t-1}))$ is computed from the representation $\rho(x_t, l_{t-1})$;
- **RNN**: To utilize the information obtained in previous glimpses, successive hidden states $h_t = f_h(g_t, h_{t-1})$ are calculated from the previous hidden state h_{t-1} and the feature vector g_t. The RNN then computes two things from h_t :
  1. **Prediction probabilities:** In classification, class probabilities are computed via a softmax layer f_a.
  2. **Generate next location:** The model computes a new location l_t for the patch to be used in the next glimpse $t+1$.

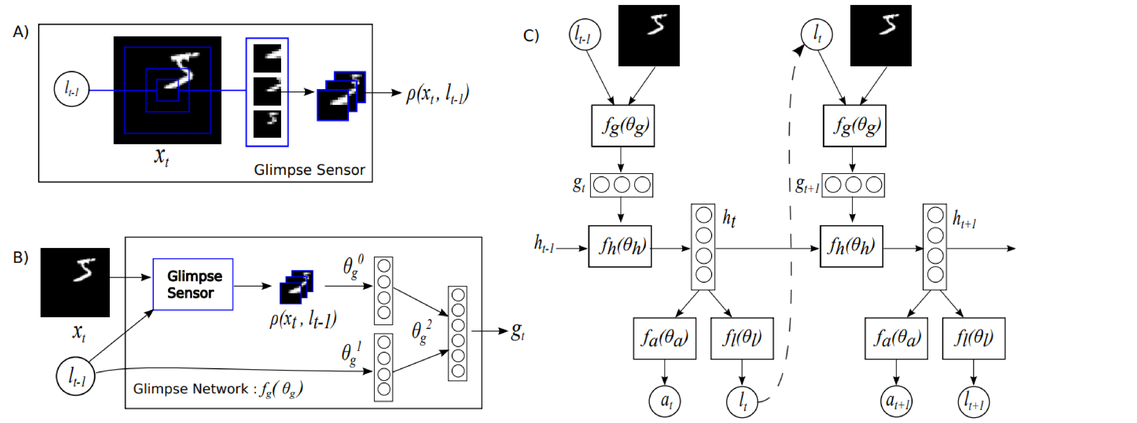

Figure 3. The architecture of RAM. The components are: **Glimpse Sensor** (A), **Glimpse Network** (B), and **RNN** (C).

Training and Prediction

Training

RAM is trained by reinforcement learning. At each glimpse t, the agent is given a reward r_t. In a classification task, $r_t = 1$ if the prediction correct, and 0 otherwise. The purpose of the agent is to maximize the total reward $\sum_{t=1}^{T} r_t$.

Prediction

Sequential movements improve the prediction result in each subsequent glimpse, because at each step more information is received to produce better predictions. Figure 4 illustrates the predictions made by a very well-trained RAM. As expected, later predictions are much more accurate than earlier ones.

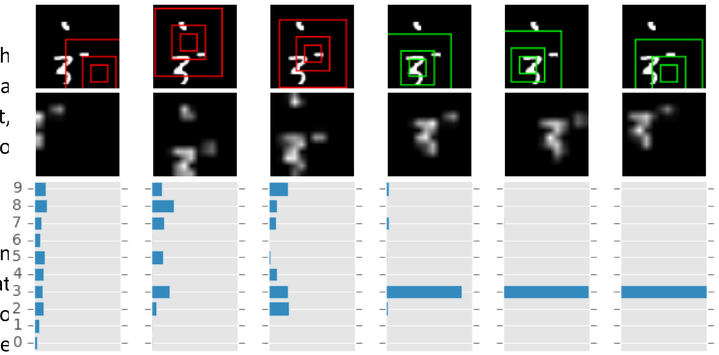

Figure 4. The prediction probabilities for "3" during each glimpse. Columns represent glimpses, which are sequentially ordered from left to right. The first row shows the original images with the selected patches (of three different sizes). The second row shows the largest image patch. This image was retrieved from https://github.com/amasky/ram.

Experiments

We trained the RAM model for 50 epochs. The prediction probabilities for the number "6" in each glimpse are shown below. As expected, the prediction indeed gets progressively better as more parts of the images are looked at.

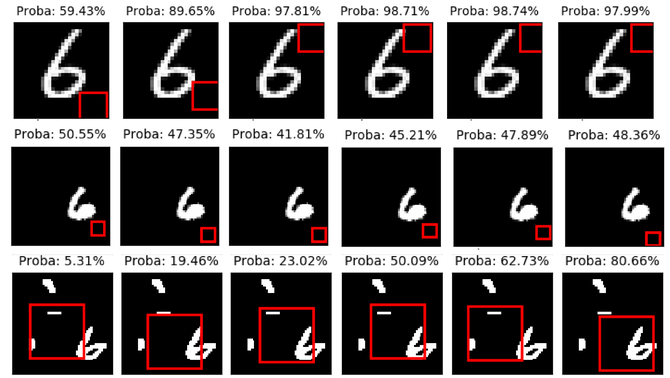

Figure 5. The image patches RAM chose at each glimpse for the "6" in the original (1st row), translated (2nd row), and cluttered (3rd row) versions of MNIST. Each column is a glimpse, ordered temporally from left to right. The prediction probability for the true class is shown above each image.

Figure 5 shows that our RAM model was able to make more accurate predictions in each subsequent glimpse for the original and cluttered MNIST, in which the image patches used were fairly large. On the other hand, the performance was significantly poorer if the image patches were small. For the translated MNIST, the prediction probability even deteriorated in later glimpses. In addition, the convergence rate was slow, resulting in long training time and made hyper-parameter tuning a challenging task.

Data	Accuracy
Original	97.45%
Translated	82.01%
Cluttered	70.62%

Table 1. The classification accuracies for different data sets.

Acknowledgement

- **HUANG Yifei**: Code modification & model traning
- **SUN Jiaze**: Result summarization and interpretation, poster writing
- **DENG Yizhe**: Reading papers & discussion
- TAN Haiyi: Reading papers & discussion

References

- Mnih V, Heess N, Graves A, and Kavukcuoglu K. *Recurrent Models of Visual Attention*. 24 Jun, 2014.
- Amasky. RAM. Retrieved from https://github.com/amasky/ram.
- Yarbus A L. *Eye movements and vision*. 1967. Retrieved from https://en.wikipedia.org/wiki/Eye_movement.