Investigating Attention Mechanisms for Variant MNIST Datasets

Chien-Sheng Wu, Peng Xu, Nayeon Lee
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[cwuak,pxuab,nyleeaa]@connect.ust.hk

Abstract

Attention mechanism is one of the key technologies for deep learning models to improve performance, especially when encoding large information simultaneously. In this project, we mainly compare two different attention mechanisms on image classification tasks, including Recurrent Attention Models (RAM) with reinforcement learning, and self-attention mechanism in Transformer models. In addition, two extensions of MNIST datasets, cluttered-MNIST and Fashion-MNIST, are used to evaluate the performance. Our experimental results and visualization show that although RAM can reduce the computational cost, it cannot easily achieve good performance with its local attention and learned policy. On the other hand, self-attention models are able to achieve similar results compared to Convolutional Neural Networks (CNN) using only one-fourth parameters on those two MNIST datasets.

1 Introduction

Recurrent neural networks, long short-term memory [1] and gated recurrent neural networks, in particular, [2], have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [3, 4]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [5]. Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states, as a function of the previous hidden state and the input for the current position. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples.

Attention is simply a vector, often the outputs of a dense layer using softmax function. Before the proposal of attention mechanism, machine translation models rely on reading a complete sentence and compress all information into a fixed-length vector [6]. As one can image, a sentence with hundreds of words represented by several words will surely lead to information loss or inadequate translation. However, attention tends to partially solve this problem. It allows machine translation models to look over all the information the original sentence holds and then generates the proper word based the context words. It can even allow a translator to zoom in or out, that is, focusing on local or global features.

Adding attention mechanism has been empirically proven to improve many tasks, such as machine translation [3, 4], image captioning [7] and image recognition [8]. One proposed a model with attention mechanism which tackles the problem of speech recognition, image classification, image captioning, and machine translation all together [9]. Although attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing

Preprint. Work in progress.

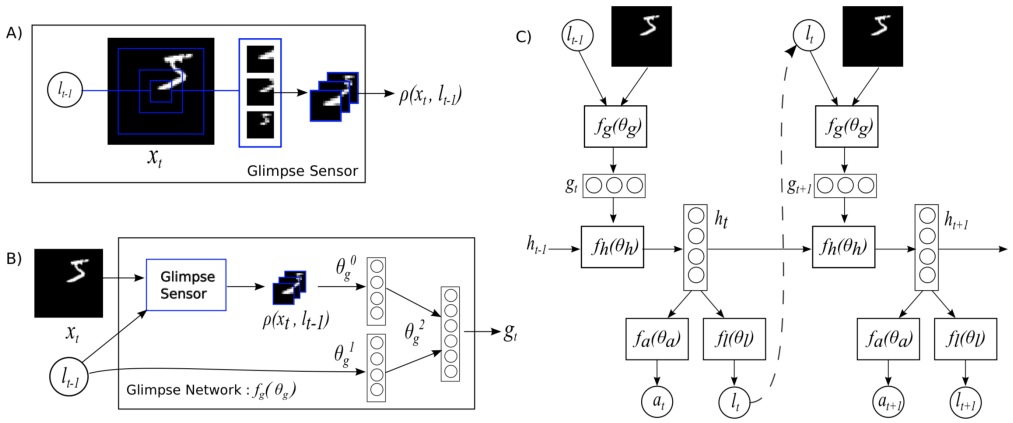

Figure 1: RAM model architecture. A) is Glimpse Sensor, B) is Glimpse Network. C) is the reinforcement learning diagram using RNN model.

modeling of dependencies regard to their distance in the input or output sequences, how these attention models perform differently from each other is still an open question.

In this paper, we empirically investigate different attention mechanisms on variant MNIST datasets. As they are simple and standard datasets, researchers usually use them to evaluate their algorithms. Our main contributions are listed below:

- Utilize the Recurrent Attention Models (RAM) [10] to perform dynamic recurrent attention mechanism with reinforcement learning.
- Implement self-attention of image convolutional features to improve the performance and reduce the parameters.
- Analyze and visualize the attention to understand deeper to the models.

2 Methodology

2.1 Recurrent Attention

Recurrent Attention Model (RAM) [10], as shown in Figure 1, is a model that learns the attention on images so that the performance of downstream tasks, such as image classification is improved. It consists of three parts: 1) Glimpse Sensor 2) Glimpse Network and 3) RNN network (agent). The environment is observed by the sensor and the agent will decide the action (which is the glimpse location) and get a predefined reward signal.

- Glimpse Sensor: Glimpse sensor takes a full-sized image and a location, outputs the retina-like representation $\rho(x_t, l_{t-1})$ of the image x_t around the given location l_{t-1}, which contains multiple resolution patches.
- Glimpse Network: it takes the retina representation$\rho(x_t, l_t)$ and glimpse location l_{t-1} as the inputs, and maps them into a hidden space using independent linear layers parameterized by θ_g^0 and θ_g^1 respectively using rectified units followed by another linear layer θ_g^2 to combine the information from both components. Finally, it outputs a glimpse representation g_t.
- Recurrent Neural Network (RNN): Overall, the agent is an RNN. The core network takes as the input the glimpse representation g_t at each step and history internal state h_{t-1}, then outputs a transition to a new state h_t, which is then mapped to action a_t by an action network $f_a(\theta_a)$ and a new location l_t by a location network $f_l(\theta_t)$. The location is to give an attention at next step, while the action, for image classification, gives a prediction based on current information. The prediction result, then, is used to generate the reward point, which is used to train these networks using Reinforcement Learning.

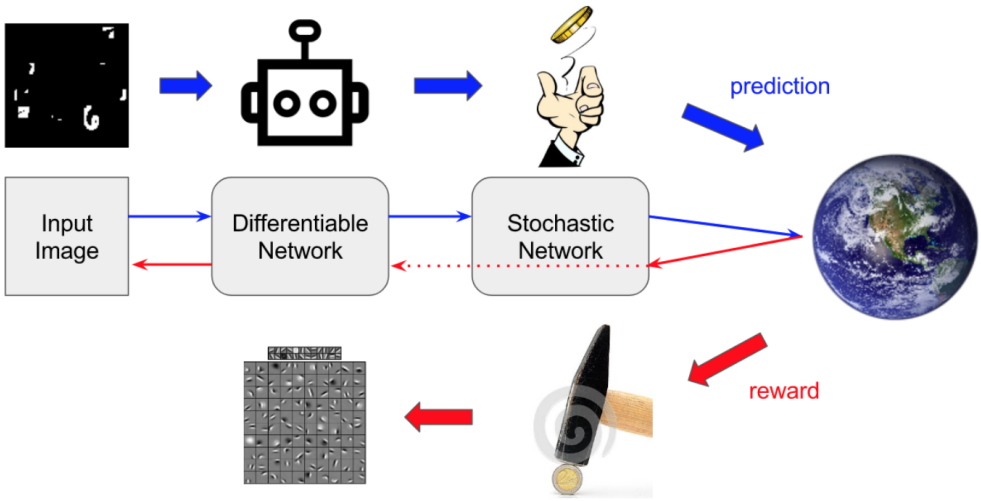

Figure 2: REINFORCE learning framework

High-level speaking, the advantage of this approach is:

- The model can pay more attention to the relevant region of the image.
- The image processing power required is reduced.
- "Off-focus" blurry area still captures a general concept of what's happening.

However, one possible disadvantage is that location network and classification model share the same representation, which may introduce undesired correlations.

Since Location Network is non-differentiable, reinforcement learning, specifically, policy gradient method, is used to train this special part of the network. In the problem, the "policy" determines how do we choose our next glimpse location. The answer (i.e. policy) is, by sampling from a Gaussian distribution with parameter $mean$ and $sigma$, where $mean$ is the mean value and $sigma$ is the standard deviation. This Gaussian distribution is our stochastic process. In this implementation, we fix $sigma$ as a network hyper-parameter and only try to learn $mean$.

To illustrate the training process of the stochastic network, the diagram in Figure 2 shows a high-level concept. The blue arrow represents forward pass, red arrow represents backward pass. Our process starts with passing an input image into our network. The differentiable network will work as normal (e.g. $XW + b$), represented as a robot head. Then the stochastic network (in our case the Location Network), will apply a random stochastic process (in our case Gaussian distribution) to generate a result (in our case the prediction of next glimpse location). It is represented as flipping a coin, to denote a random process. Then the location prediction (together with classification result using the glimpse) is evaluated by the environment, and now we have a reward to start backward propagation. Conceptually, the backward propagation through the stochastic network can think of as if we make our coin more biased towards the direction that will get us more rewards in the future. In other words, we learn the parameters of the Gaussian distribution. This step is represented as a hammer on a coin, to denote that, we want to "forge" the coin in our favor.

2.2 Self Attention

Self-attention, sometimes called intra-attention is an attention mechanism relating to different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [11, 12, 13].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and

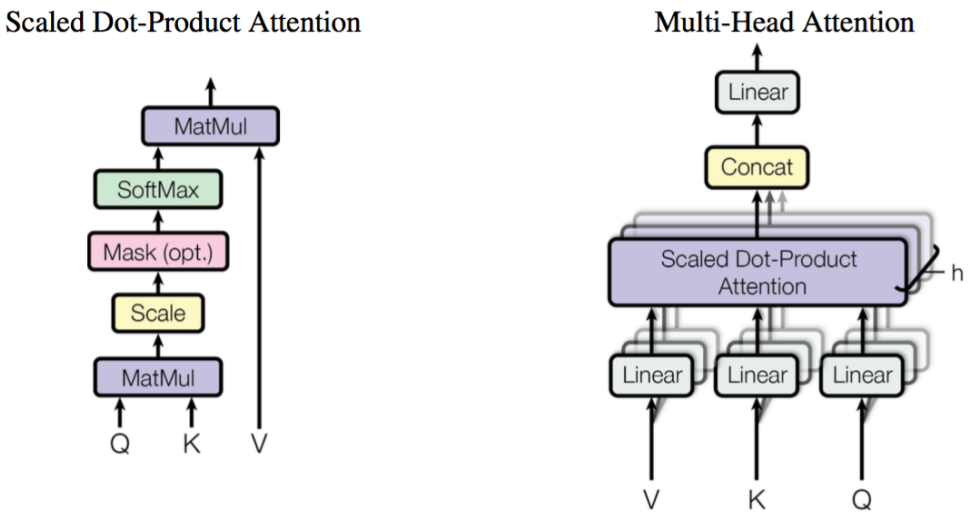

Figure 3: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

language modeling tasks [14]. The Transformer [15] is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. The multi-head attention mechanism is shown in Figure 3

In order to utilize self-attention mechanism into image classification problem, first, we create a simple convolutional neural network to encode the images. It is two layers of convolution and max-pooling, with a ReLU activation layer in between. An easy way is just flattening the last convolutional layer, then added 2 fully connected layers to predict the image classes. We refer this as a simple CNN baseline.

Similar as the Transformer Model, the multi-head attention mechanism is set for the following parameters: The numbers of blocks in the feature map that convolution network made, the dimension of the block, the dimension of the linear space the input to be projected, the output of the block after applying this attention, and the number of projections for each block (in the original paper is called heads). query, key and value vectors are from the input, which is the features extraction from the simple CNN. These tensor will be a block from the convolutional feature map. In addition, another important technique that makes the model to learn faster is a normalization layer. This layer makes the tensor to have a standard normal distribution and to delete some dimensions of the vector that are not important.

3 Experimental Setup

3.1 Dataset

Two datasets are used in the paper: Fashion-MNIST and Clutter-MNIST, where both of them are an extension of original MNIST [1] dataset.

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes. We intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

One possible advantage of an attention mechanism is that it may make it easier to learn in the presence of clutter by focusing on the relevant part of the image and ignoring the irrelevant part. We test

[1] http://yann.lecun.com/exdb/mnist/

Table 1: Evaluation on RAM

Models (glimpse no. - retina size - scale no. in retina - optimizer - lr)	Accuracy	Epoch
Cluttered, Simple CNN	0.944	NA
Cluttered, RAM, 4 glimpse, 12 x 12, 3 scale, Adam 3e-4	0.927	92
Cluttered, RAM, 6 glimpse, 12 x 12, 3 scale, Adam 3e-4	0.919	76
Cluttered, RAM, 8 glimpse, 12 x 12, 3 scale, Adam 3e-4	0.935	87
Cluttered, RAM, 6 glimpse, 12 x 12, 3 scale, SGD 1e-2	0.910	179
Cluttered, RAM, 6 glimpse, 12 x 12, 3 scale, SGD 1e-3	0.919	156
Fashion, Simple CNN	0.926	NA
Fashion, RAM, 6 glimpse, 8 x 8, 1 scale, Adam 3e-4	0.856	196
Fashion, RAM, 6 glimpse, 7 x 7, 3 scale, Adam 3e-4	0.867	56

this hypothesis with several experiments on a new task of Cluttered Translated MNIST. The dataset was created by placing a random MNIST digit on a canvas of size 40×40 pixels. It was placed by randomly sampling an y position on the canvas. The x position is randomly sampled subject to the entire sequence must fit the canvas. The procedure is placed by following a slope sampled from $\pm$ 45°. Finally, the images are cluttered by randomly placing 5 patches of size 5×5 pixels sampled from the original MNIST digits. For the test, validation and training sets we sample from the corresponding set in the original MNIST dataset. We create 10000 examples for training, 1000 for validation, and 1000 for testing.

3.2 Training

For simple CNN model, we use two layers of convolution and max-pooling. The first convolutional layer is with kernel size 2 and 32 filters followed by a pooling layer. The second one is 2 kernel size and 64 filters. We use Adam [16] optimizer with learning rate 0.001. The learning rate is decayed by the factor 0.5 and early stopping with patience equals to 2 epochs.

For RAM, we have 4 networks that are: glimpse network, location network, action network, and core RNN network. For glimpse network, scale of successive patches is 2, number of downscaled patches per glimpse is 3, the hidden sizes of the location and glimpse fully connected layers (h_l and h_g) are both 128, and the extracted patch size at highest res is 12 and 7 respectively for Cluttered-MNIST and Fashion-MNIST. For training the action network, which uses reinforcement learning, the gaussian policy standard deviation is set to 0.17, and Monte Carlo sampling for valid and test sets is set to 10. For core RNN network, the number of glimpses (i.e. BPTT iterations) is 6, and the hidden size of the RNN is 256. Lastly, we use Adam [16] optimizer with learning rate 0.0003 and early stopping with patience equal to 50 epochs.

4 Results

4.1 Recurrent Attention

As shown in 3.2, our RAM achieves the classification accuracy of 0.935 and 0.867 for Cluttered MNIST and Fashion MNIST respectively, which are lower than the simple CNN baseline. This is different from the original paper which demonstrated a RAM model that outperformed simple CNN model. We believe possible reasons for such discrepancy are 1) The policy network is not trained properly. That is, the location prediction of location network is not good enough. 2) Not enough glimpse RNN steps, which implies that the local attention of recurrent attention may also increase the information loss at the same time to reduce computational cost.3

In the original paper, SGD is used as an optimization method. However, we have also tried using Adam to explore if it can train RAM better. As it can be seen in 3.2, Adam with learning rate $3e-4$ is the best, achieving 0.935 test accuracy. The performance of Adam is even more notable given the fact that it has faster training time compared to SGD method. It only took 87 epoch to achieve accuracy of 0.935, when with SGD method only achieved 0.919 even with 156 epoch.

Figure 4: Visualization of attention on Fashion MNIST

In addition, for Fashion MNIST, we explored different choices of retina sizes, which is controlled by a number of scale in the retina and the size of the retina. Despite the small size of the image (28x28), large retina size is required for the RAM model to learn better. By increasing the scale from 1 to 3, both the testing accuracy and training speed is improved, from 0.856 to 0.867 and from 196 epoch to 56 epoch respectively. 3.2. This could be due to more challenging nature of the Fashion MNIST compared to Cluttered MNIST, requiring more information about the input image.

We have also visualized in Figure. 4 how our attention model put attention to different parts of the image to make the classification. The red boxes represent the location of the attention, and images are arranged top to down in chronological order. We can easily see that, in most of the times, the attention is correctly being applied to the target object, not to the background.

4.2 Self-Attention

As shown in Table 2, with and without self-attention layer have similar results in both datasets. In Fashion-MNIST, simple CNN can achieve 92.61% accuracy, which is 0.5% higher than the self-attention model. On the other hand, in the cluttered-MNIST dataset, self-attention can achieve 95.1% accuracy, of which 0.7% higher that simple CNN. However, the point we want to highlight is the number of parameters used for two model. Self-attention models can achieve the results with only one-fourth of the parameters simple CNN used. The main reason is, for simple CNN, the final classification results are decided by a fully-connected layer which takes all the convolution features as input and thus needs 1.77M parameters. Instead, self-attention increases the "operation" but not parameters. It learns the self-attention weights between each patch of convolutional outputs and "weighted-sum" to classify. In this way, the parameters can reduce dramatically.

Table 2: Evaluation on Attention Models

Dataset	Simple CNN	+Self-Attention
Number of Parameters	1.83M	0.47M
Fashion-MNIST	0.9261	0.9205
Cluttered-MNIST	0.9440	0.9510

5 Conclusion

In this paper, we investigate different attention mechanism models on variants of MNIST datasets, that are cluttered MNIST and fashion-MNIST. We choose Recurrent Attention Models with reinforcement

learning and self-attention mechanism in Transformer models for comparison. Our results and visualization show that although RAM cannot easily achieve good performances due to its local attention even though it greatly reduces the computational cost. On the other hand, using only 25% parameters of CNN, self-attention models achieves competitive results.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[5] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[7] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[8] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2231–2239. IEEE, 2016.

[9] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

[10] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[11] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[12] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[13] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[14] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc., 2015.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.