Recurrent Attention Model for Image Classification

Vatsal Jatakia
Indiana University
Bloomington, IN, USA
vjatakia@iu.edu

Rohit Rane
Indiana University
Bloomington, IN, USA
rrrane@iu.edu

Ramya Rao
Indiana University
Bloomington, IN, USA
ramrao@iu.edu

Abstract

Convolutional Neural Networks have been successful in the field of Computer Vision for the task of object classification, but they become computationally more expensive as the resolution of training images increases. To alleviate this problem, we have tried to implement a variant of the Recurrent Attention Model(RAM) which incorporates retina-like model similar to that of human visual system, which learns to focus on the important regions of the image and make a prediction of the labels instead of seeing the entire image [1]. This method produces state-of-the-art results on a simple dataset like MNIST [2]. We are trying to use this concept for classification in a more complex set of images with different backgrounds - CIFAR-10 dataset [3]. In spite of incorporating various modifications along with the inclusion of InceptionNet[4] in the model for extracting features, we were not able to generalize the model on complex datasets like CIFAR-10. The report below describes the procedure and experiments we performed in order to come to this conclusion.

1. Introduction

Human visual system processes continuous stream of visual information from a real-world scene, and effectively performs task-dependent operations[5]. To reduce the complexity of scene analysis, the human visual system selects only certain parts of the scene to focus attention at a particular time[6]. These parts of the scene are then processed to accomplish tasks such as classification, detection, segmentation and so on. Work by [7] introduce a popular model to enumerate the idea mentioned previously which takes an image at high resolution as input and extract features from a subset of interesting locations and process them further to perform a specific task. This model attests to the fact that, machines could be trained to look more intelligently based on a specific task in the input space to extract features.

Human visual system, which is studied widely, also motivate the RAM model proposed by [1], which uses a Recurrent Neural Network (RNN) acting as an agent in the environment selecting a location to focus its attention, at a particular time step. RNN structure allows the model to propagate information obtained at previous locations and dynamically change the perception of the image/video frame fed as an input sequentially. This formulation has many advantages, such as independence from the resolution of the input images or videos, and reduction in the number of parameters as compared to other models performing the same task with the same resolution images/video frame. The agent is then trained through a policy gradient method in order to perform a classification task on MNIST dataset [8].

Performance of Deep Neural networks is remarkable and even better than accuracy of humans for some tasks [9]. But, the downside of such gains is the increased amount of computation in time and space complexity during training phase with the increase in the size of input images. The inception model GoogLeNet uses more than 6.7 million parameters [10] and the VGGNet has about 139 Million parameters [11], to give few examples. The number of parameters increase proportional to the resolution of the input image. This is an issue which has to be addressed as some applications need input at a high resolution. RAM model circumvents this problem effectively by just focusing on a small window of the image at a time and incrementally collects enough information to perform the main task. The RAM model proposition of devoting attention to a particular region of interest at a time, is general enough to attract a lot of applications. Self driving cars can leverage RAM to handpick only few most crucial events happening on the road from the input video.

In this work, we analyze the performance of the RAM [1] on colored image data which is more versatile than MNIST. We also show that replacing glimpse-sensor in the RAM with Inception-Net has potential to improve the performance of the model. We used CIFAR-10 dataset to test the hypothesis of generalization of RAM model. CIFAR-10 dataset has images pertaining to the ones in the real-world and has the same number of images as the MNIST dataset. It has 10 different objects with high variance per object class. The 10 classes are also very diverse spanning from automobile to birds.

2. Related Work

There are many methods for object classification in Computer Vision, our RAM is one of the methods. As we design our RAM-based classification system, we mention few very closely related works with respect to the RAM in this section, we introduce an attention model [12] which performs object tracking and classification on videos. Their model is also inspired by human visual system, and is based on the idea that humans process visual information with successive eye fixations and register changes in the scene dynamically. They define an identify pathway which is responsible for classification and encompasses a two-layered Restricted Boltzmann Machine and a control pathway. The control path pathway is subdivided into localization network and gaze selection network, it is responsible for tracking an object and selecting an attention strategy respectively. A Gaussian process along with online policy learning is used to train the network to learn intelligent fixations in a given video frame. [1] adapt this model by using an RNN to model the sequential nature of problem unlike [12].

[13] propose a multi-fixation Restricted Boltzmann Machine model which forms a feature vector based on few locations from an image and perform classification on those features instead of the whole image. They promote a novel method, to extract features from each location based on human retina and name it as Glimpse Sensor. The Glimpse Sensor extracts high resolution features from the location of interest and also incorporates the surrounding region by decreasing the resolution. This is similar to the feature extraction by the retina when focusing on a given region in the scene. The retinal representation of a region in the scene contains not only the region with high resolution, but also additional information about the scene from peripheral vision with lower resolution. [1] absorb the Glimpse Sensor and use them to extract features from each location and feed them into a Long Short Term Memory (LSTM) unit.

3. Dataset

We experiment with the well-known MNIST dataset [2] to validate the performance of the RAM model. MNIST dataset contains 55,000 training images and 10,000 testing images of size 28x28 pixels containing a set of handwritten digits from 0-9 and each class represents a digit.

We used the CIFAR-10 dataset [14] which consists of 60,000 images of size 32x32 and three color channels. There are 10 classes with about 6000 images per class. The dataset has 50000 images for training and 10000 images for testing phase. The ten classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Figure 1 demonstrates the variability in CIFAR-10 dataset.

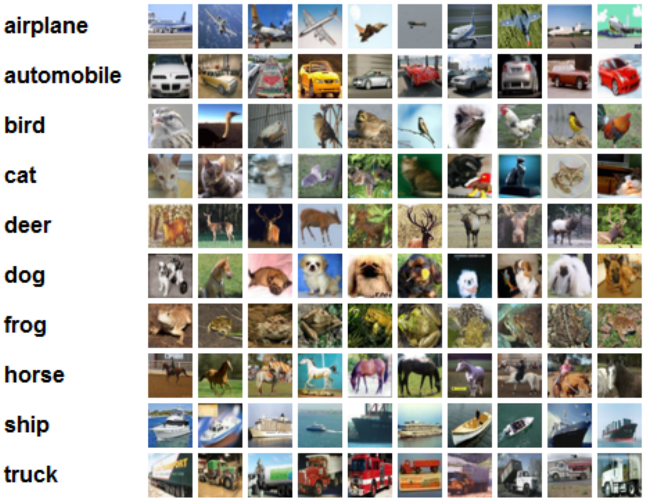

Figure 1: Variability in CIFAR-10 dataset [14]

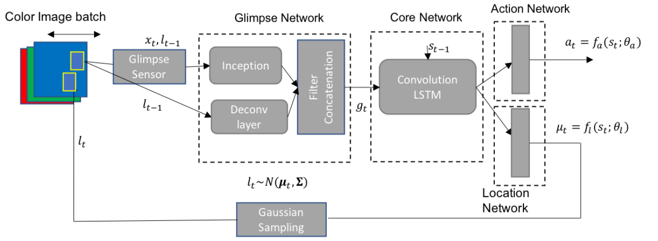

Figure 2: Our Architecture

4. Architectural Details

Our RAM model architecture consists of Glimpse Sensor, Glimpse Network, Core Network, Action Network and Location Network, each described below in detail. Figure 2 illustrates the model for a single glimpse.

4.1. Glimpse Sensor

It is the ”retina-like” component used by [1] that extracts information from the image by taking square window region $g_w \times g_w$ around location l_t which is called as a glimpse. In addition to this region, the sensor also extracts k square patches centered at location l_t, where successive patches have twice the resolution of the previous patch. All the patches are resized to a predefined window size. In RAM model by [1], these patches are flattened and fed into Glimpse Network, to retain spatial information, we do not flatten the output of Glimpse sensor.

4.2. Glimpse Network

Glimpse Network in RAM model passed the content information obtained at location l_t from the Glimpse sensor and context/location information through two fully connected layers and combined the output which was fed to the Core net-

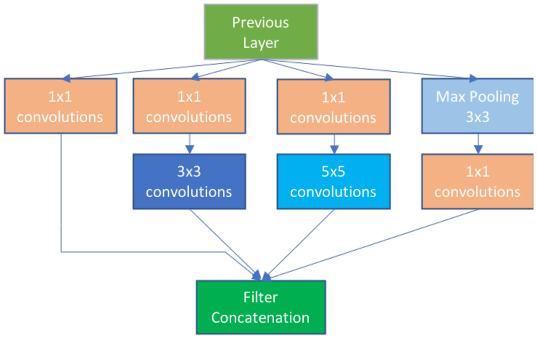

Figure 3: Inception Module

work. We modified this structure by introducing the inception module, which produced 24 feature maps of size 12x12 which was the window size chosen for our model across various scales. We added de-convolution layers to the location information to match the dimensions of output of inception module. Finally, the content and context information were combined by addition and this output is fed to the Core network.

4.3. Core Network

The core network in the original RAM model consisted of an LSTM [15] unit which unrolled to the number of glimpses, which is a hyper-parameter. In order to accommodate processing 3 dimensional inputs, we replaced the normal LSTM unit with a convolutional LSTM unit [16]. The state and output variables for the convolutional LSTM units are all 3 D tensors where the last two dimensions represent the height and width of the window being evaluated at the current time step. As the spatial information is retained in this formulation, we expected the RAM to improve performance on color images classification.

4.4. Action Network

The action network is responsible for the task of classification. The output of the convolutional LSTM unit at the last glimpse is flattened and fed into a softmax layer. The rationale behind selecting the output of the last glimpse, is that, once the network has seen all the glimpses, it equipped with all the information necessary to perform the task of classification. Action network is also responsible for generating the rewards. If the network predicts correctly after the final timestep, then is is awarded 1 point and 0 otherwise. The goal here is to maximize the rewards and minimize cross entropy loss.

4.5. Location Network

The location network consists of a fully connected layer which requires the output of the convolutional LSTM unit at each time step as the input. The output is treated as the mean value in both x and y direction of the input image. Then we sample a Gaussian distribution with the mean obtained and fixed standard deviation and use this as the location input for the next time step. We treat both x and y direction independently and assume both has fixed variance. This method of obtaining the location is non-differentiable and RAM uses reinforcement learning methods to train location network. The REINFORCE algorithm is used to tune the parameters of the location network.

4.5.1 REINFORCE

Location Network f_l is a non-differentiable component of the RAM and therefore is trained by REINFORCE. In REINFORCE, we define objective function J as the expected cumulative reward earned by an agent.

$$J(\theta) = E_{p(s_{1:T};\theta)}[R] \tag{1}$$

where R is the total rewards earned in an episode of T timesteps. The exact solution to this equation is intractable and hence we use approximate gradient defined as below:

$$\nabla_\theta J = \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T}\nabla_\theta log\pi(u_t^i|s_{1:t}^i;\theta)(R_t^i - b_t) \tag{2}$$

where b_t is the baseline used to reduce the variance.

5. Experiments

For the baselines we used 2-layer fully connected network wit 256 units and 1-layer convolutional network with 8 filters of size 10×10 followed by fully connected layer of 256 units. We trained both the baselines using Gradient Descent with learning rate 0.001 on standard cross-entropy loss.

We first trained RAM on MNIST (28x28), translated MNIST (60x60), cluttered MNIST (60x60), CIFAR-10 (32x32) and translated CIFAR-10 (60x60) dataset using hybrid loss and Adam optimizer with a learning rate of 0.0001 using Tensorflow on Futuresystems. We used mini-batch size of 20 images along with exponential decay after every 10,000 epochs. We trained the model for 100000 epochs and recorded the test accuracy for the model obtained.

6. Results and Discussion

6.1. Original MNIST (28 x 28)

We were able to reproduce the results of [1] on original MNIST with the Recurrent Attention Model achieving accuracy of 94.21. Our FC-2 baseline achieved accuracy of 97.27% while CNN baseline performed slightly better with accuracy 97.78%. Table 1 summarizes these results.

Model	Accuracy(%)
FC 2 layers	97.27
Convolutional, 2 layers	97.78
RAM, 4-glimpses, 1 scale	92.45
RAM, 6-glimpses, 1 scale	93.79
RAM, 8-glimpses, 1 scale	94.21

Table 1: Results on 28 x 28 MNIST

6.2. Translated MNIST

The Recurrent Attention Model could achieve maximum accuracy of 88.63% on translated MNIST. Our CNN baseline achieved 85.17% accuracy while FC 2-layer baseline could merely reach 70.62%. Table 2 shows the results.

Model	Accuracy(%)
FC 2 layers	70.62
Convolutional, 2 layers	81.2
RAM, 6-glimpses, 2 scales	85.17
RAM, 8-glimpses, 3 scales	88.63

Table 2: Results on Translated MNIST

6.3. Cluttered MNIST

On cluttered MNIST, the FC 2-layer baseline performed worst with accuracy 42.5%. CNN baseline performed slighty better with accuracy 54.9%. The Recurrent Attention Model could attain maximum accuracy of 82.01%. Table 3 shows the results.

Model	Accuracy(%)
FC 2 layers	42.5
Convolutional 2 layers	54.9
RAM, 6-glimpses, 2 scales	77.5
RAM, 8-glimpses, 3 scales	82.01

Table 3: Results on Cluttered MNIST

6.4. CIFAR-10 (Grayscale)

On CIFAR-10 Grayscale dataset, FC 2-layer baseline achieved 41.4% while CNN baseline achieved 55.35% accuracy. Accuracy of Recurrent Attention Model could reach to 20.67%.

6.5. CIFAR-10 (Colored)

On the colored version of CIFAR-10, both the baselines performed same as they did on its grayscaled version. The Recurrent Attention Model with pixel averaging glimpse sensor did not learn anything. The Recurrent Attention Model with inception-net sensor could achieve maximum accuracy of 25.37%

Model	Accuracy(%)
FC 2 layers	41.4
Convolutional 2 layers	55.35
RAM, 4-glimpses, 3 scales	15.45
RAM, 8-glimpses, 3 scales	20.67

Table 4: Results on CIFAR-10(Grayscale)

Model	Accuracy(%)
FC 2 layers	41.4
Convolutional 2 layers	55.35
RAM with inception-net, 4-glimpses	22.16
RAM with inception-net, 8-glimpses	25.37

Table 5: Results on CIFAR-10(Colored)

RAM model, works on simple datasets such as MNIST where the classes have only digits. Assumptions such as sampling the next location from a Gaussian distribution and treating the rows and columns independently holds true only on MNIST dataset but not on CIFAR-10 dataset, based on our experiments. The finding from our project after experimenting with variants of RAM model is that, it does not generalize to complex real-world datasets such as CIFAR-10. As the RAM model does not see the classes completely and only sees glimpses at every step, it is very difficult for the model to distinguish between and across classes with high variability.

We experimented with a variant of RAM model by introducing the inception module in the glimpse network and integrated a convolutional LSTM cell [16] in place of normal LSTM cell, to retain the spatial construction of CIFAR-10 images. However, the model is still not able extract innate features from each class which is very essential to distinguish them. We can speculate that confusion can arise between the automobile and truck class as the primary difference between the category is the length of the vehicles and as the RAM model does not see the entire image, the task of classifying them successfully is extremely challenging. Similarly, based on glimpses of an image of dog or cat, if the glimpses only captured the body, then the model can find it very hard to classify it as either a cat or a dog. Since classifying CIFAR-10 was a laborious task for RAM model, classification on translated CIFAR-10 did not improve above chance, the model found it extremely difficult to find structure in translated version. And same case with cluttered dataset as well. Hence, we conclude that, attention model by Google DeepMind does not have the capacity to generalize to real-world images and tasks contrary to information presented in the paper [1].

7. Conclusion

Our results suggest that, RAM is a simple attention model and is not equipped to handle complicated real-world images for the etask for classifixation. The glimpse sensor being used is too naive in this case to capture complex curvatures and color-variations present in the scene. Even though CIFAR-10 images have low resolution, they offer high level of complexity to be captured by the glimpse sensor.

When Inception module is used as the sensor, Recurrent Attention Model shows small improvement in accuracy on colored images. This is possibly due to the ability of Inception module to capture prominent features of the objects in the colored images. We could not experiment more on this modified version due to time constraint but these initial observations show that this variant of Recurrent Attention Model has the potential to be useful in the task of classification of colored images. If successful, this can further be tested on high-resolution images.

References

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

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

[3] https://www.cs.toronto.edu/ kriz/cifar.html.

[4] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016.

[5] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[6] E. Niebur and C. Koch. Computational architectures for attention. In R. Parasuraman, editor, *The Attentive Brain*, chapter 9, pages 163–186. MIT Press, Cambridge, MA, 1998.

[7] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Patrice Simard, Yann LeCun, and John S Denker. Efficient pattern recognition using a new transformation distance. In *Advances in neural information processing systems*, pages 50–58, 1993.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

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

[12] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012.

[13] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Advances in neural information processing systems*, pages 1243–1251, 2010.

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

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

[16] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.