Applying Active Learning to Recurrent Attentive Models

Ethan Brooks ETHANBRO@SEAS.UPENN.EDU
Luke DeLuccia LUKEDEL@SEAS.UPENN.EDU
Evan Inglett INGLETT@SEAS.UPENN.EDU

Abstract

Convolutional neural networks are currently the state-of-the-art in object recognition but they are slow to train. "Attentional" models are a promising new approach for improving the runtime of neural networks. These models take in small portions of the total data and process them in sequence, similar to how the brain processes sequential glimpses (saccades) of the eye. Our research proposes a modification to Google DeepMind's "Recurrent Attentive Model." Specifically, we propose a novel reward structure for selecting glimpses: rather than only rewarding glimpses that produce correct classifications, we explore providing an additional reward for any glimpse that modifies the models existing hypothesis. While ultimately this approach proved unsuccessful, this exploration yielded many interesting findings regarding deep learning.

1. Introduction

The current state of the art in image classification depends on techniques that are very computationally expensive. Convolutional neural networks, "sliding windows," and "image pyramids," improve accuracy but at the price of significant computational power and testing times, as the model must perform classification on each of the possible thousands of windows (D. Eigen, 2013; G. Hinton, 2012).

A common feature of images that these approaches fail to take advantage of is that information density varies widely between different regions. Salient details are often concentrated in a relatively small region of the image while other regions have little informational value (T. Darrell, 2013). An algorithm could take advantage of this fact by learning to focus on the former while disregarding the latter.

Research on human vision indicates that the brain uses an approach similar to this to process visual information. The retina captures only a one- to two-degrees cone of vision at full resolution and consequently, humans cannot view a scene with a fixed gaze. Instead, the eyes rove continuously around a scene, fixating on information-rich regions (Rinsing, 2000; Colombo, 2001; D. Ballard, 2005; S. Mathe, 2013). Extensive neuroscience research has been devoted to these short glimpse movements of the eye, known also as saccades. This research suggests that saccades are not random, and that effective execution of these sequences is critical for perception.

Recently Google DeepMind presented a classification algorithm that uses saccade-like mechanisms to achieve higher accuracy than traditional fully-connected neural networks. The DeepMind algorithm processes images in a series of "glimpses," retina-like representations with resolutions that decay with distance from the center. Each glimpse contains a fraction of the pixels of the entire image and therefore costs the algorithm much less time to process. Furthermore, the algorithm learns to guide these glimpses toward salient data while disregarding data that carries little informational value (A. Graves, 2014).

We began our research exploring modifications to the DeepMind's algorithm relating to the reward structure for learning glimpse patterns. Specifically, we introduced additional awards for glimpses that modified the model's outputs. Our aim was to steer the model towards information-rich regions of the image. Though our initial modifications achieved little success, our research helped us better understand the fundamental relationship between attention with which we were able to achieve some minor improvements to the original model's accuracy.

1.1. Related Work

Improvements to the computational runtime of object detection algorithms have received significant attention recently. Some strategies are modifications to the sliding window approach, such as Ranzato's method which first down-samples the image to detect candidate locations for examination at higher resolution (2014). Each successive glimpse is then chosen using these candidate locations and information about the current glimpse.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Among these approaches, attentional mechanisms modeling human saccades are especially gaining interest among machine learning researchers. Najemnik and Geisler were able to successfully capture some aspects of human saccades using their Ideal Searcher (2005). However, their policy for determining the next location to examine is greedy. Although this allowed them to maximize the instantaneous information gain of glimpses, it does not incorporate long term information as the human brain does.

Paletta, Fritz and Seifert also developed a model that uses information from locations with high saliency measures (2005). Their decision making agent utilized Q-learning to select glimpse patterns that maximized the cumulative reward for correct object recognition. Their algorithm rewards glimpses that encompass the actual object to be classified.

Many of the strategies to reduce computational complexity are highly task-dependent and are primarily focused on finding the bounding box around a target area in an image (Butko & Movellan, 2009). Given the set of training images with a label and ground-truth object locations, Alexe, Heess, Teh and Ferrari developed a strategy that uses characteristics about the object class and statistical relation between the position and appearance of sequential glimpses to localize objects (2012). For example, a model whose first glimpse is of the sky, could be trained to execute its next glimpse in a lower region of the image, given the knowledge that cars are located below the sky in most images.

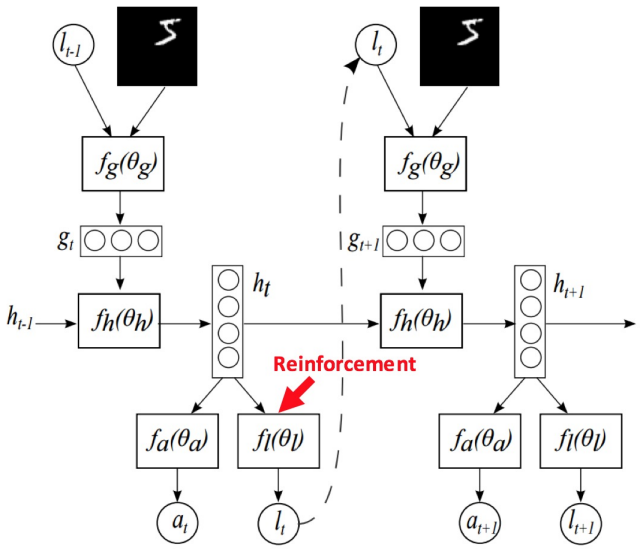

Figure 1. The glimpse network, $f_g(\theta)$ combines the output from the glimpse sensor with the location of the glimpse. The overall framework is a RNN with h_t representing the internal state of the model, a_t representing the classification of the model and l_t representing the location to extract a glimpse. Our implementation modified the location module of the framework

2. Methodology

Before discussing our research, we will describe the algorithm presented by Google DeepMind in some greater detail. The main classifier used by the DeepMind algorithm is a recurrent neural network. At each time step, the network receives the output of the previous time step, as well as the next "glimpse" or snapshot of the image. In addition to predicting the classification of the image, the network also learns a policy for selecting glimpses. The number of glimpses is set as a constant parameter. The model only receives a reward if it correctly classifies the image and receives no reward for incorrect classifications or for glimpses prior to prediction. Once all glimpses have been fed into the model, the algorithm performs Backpropogation Through Time. Most layers simply backpropogate error based on a loss function (our implementation uses negative log probability). However, the module that selects glimpse locations ignores the backpropogated error and instead learns using the REINFORCE algorithm (Williams, 1992). This algorithm maximize the log-probability of actions that have led to high cumulative reward, using the following equation for the gradient:

$$\bigtriangledown_\theta J = \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T}\bigtriangledown_\theta log\pi\left(u_t^i \mid s_{1:T}^i;\theta\right)R^i \qquad (1)$$

where s^i is the interaction sequence obtained by running the current agent π_θ for $i = 1...M$ episodes.

2.1. Problem/Motivation

Our goal was to improve the locator's ability to find the optimal glimpse policy. In particular, we hoped to improve on its current stochastic search policy.

Though the state space of a single glimpse is large, it is likely restricted enough to be explored stochastically. Because each image is 28×28 pixels, a glimpse can take on approximately 676 different values (excluding borders because the model does not permit glimpses that run off the edge of the image). However, the final prediction is not a function of the state of the final glimpse but of all n glimpses. The state space of n sequential glimpses is very large—676^n. Our hypothesis was that a stochastic search policy might find an effective glimpse pattern but was unlikely to find the optimal one.

We proposed a modification to the reward structure of the algorithm to guide the model more directly to an optimal glimpse policy on the assumption that optimal glimpses have certain properties that the model could search for.

Specifically, we assumed that a valuable glimpse contains information that contributes significantly to the model's "understanding" of the image. This contribution can be measured by changes in the classification outputs of the model because information-rich regions of the image modify the model's outputs more than a region with little information.

This strategy was based on several assumptions about the way the model processes glimpses. We assumed that the model did not forget old glimpses and that its knowledge of the image was cumulative with each new glimpse, similar to adding pieces to a puzzle without ever removing old ones. Furthermore, we assumed that a glimpse was at worst benign—it might contribute little to the model's prediction, but would never change a good prediction to a bad one. For example, a glimpse of MNIST data that entirely omitted the actual digit would contain little helpful information for the classifier and would simply leave the model's current classification unchanged. Mathematically, we assumed that the model's expected error was monotonic and decreasing with the number of glimpses, irrespective of the glimpse content.

This concept draws inspiration from the entropy-loss function of a decision tree, which steers the model toward parameters that significantly impact its understanding of the decision space, and is used as an active learning query strategy known as Expected Gradient Length (EGL) (M. Craven, 2008b). For active learning situations where labeling instances is expensive, this framework labels instances that impart the greatest change to the current model in order to properly classify all data. Change is approximated using the expectation of the Euclidean norm of the gradient vector over the N-best labelings (M. Craven, 2008a).

2.2. Approach

We tested our modifications on an open source implementation (N. Leonard, 2015) of the DeepMind paper. Referencing the REINFORCE algorithm (equation 1), we altered the total reward R by adding a bonus reward proportional to the change of the predicted classification. Specifically, we redefined r_t, the reward at timestep t, as follows:

$$r_t := r_t + ||O_t - O_{t-1}||_2 \tag{2}$$

where O_t is the output of the classifier module (the module responsible for classification of the image) on timestep t.

2.3. Results

Our reward was successful at increasing change in outputs between glimpses as shown in figure 2, but did not produce good results in predictions. Accuracy initially climbed to over 50% but abruptly fell to 10 or 11% by the tenth epoch (each epoch comprised 50,000 classification attempts).

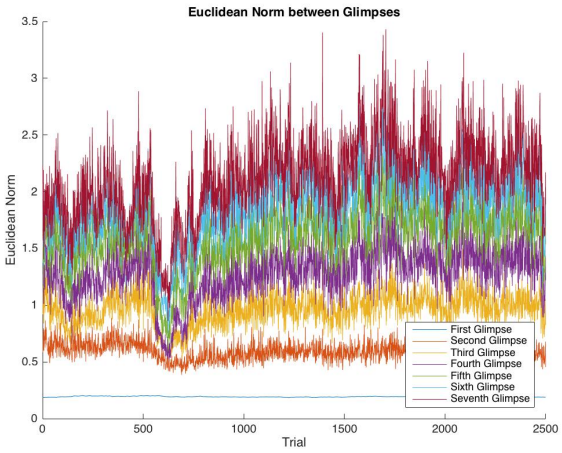

Figure 2. The norm increases with each glimpse

Instead of rewarding change in outputs, we experimented with rewarding low standard deviation of outputs. Similar to change in outputs, this reward structure encouraged skepticism in the model - that is, it discouraged it from settling quickly on a false hypothesis, and rewarded high confidence in a single classification. However, this method proved equally ineffective.

These results led us to question some of our original assumptions–specifically that accuracy was monotonic with the number of glimpses. In order to test this theory, we checked the loss incurred by our model's predictions after each glimpse using random glimpses and then using trained glimpses. As expected, trained glimpses reliably increased the model's accuracy. Random glimpses did diminish the cost, but had much less effect (figure 3). While this result was somewhat ambiguous, it encouraged us to revisit our initial hypothesis, that new information from glimpses was inherently benign.

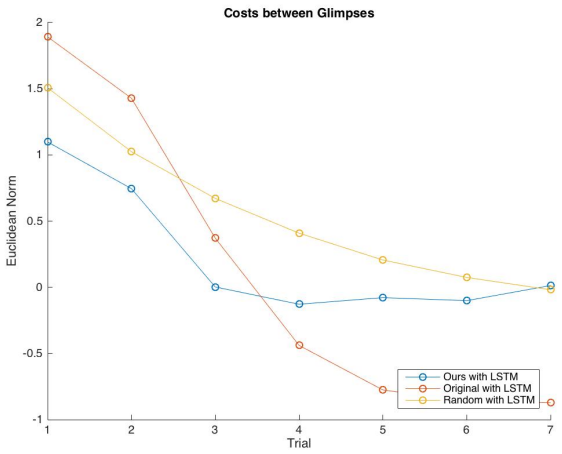

Figure 3. The cost of each glimpse decreases with time,

Instead we formulated a new theory: that each glimpse was contributing positive information, but the model forgot old information with each time-step. This issue related to the problem of the "vanishing gradient" which motivated Hochreiter et. al. to invent the Long Short Term Memory unit (LSTM) (1997). Conventional feed-forward recurrent neural networks are unable to backpropogate through a large number of time-steps because gradients (and the error information that they contain) tend to fade. We concluded that while the information in each glimpse always contributed positively to the model's understanding, each came at the cost of an old glimpse that the model forgot.

In order to combat this tendency, and give the model more control over what information it retained or overwrote with new information, we replaced the linear feed-forward layer responsible for selecting glimpse locations with an LSTM (the Google DeepMind paper had actually implemented an LSTM in another module in the network but this was not included in the publicly available implementation).

The results of this replacement were immediate–both our model and the original Element-Research implementation significantly improved their accuracy, as shown in figures 4 and 5. We tested the original model on both the MNIST dataset as well as the Cifar10 dataset, a dataset of everyday images with 10 different classifications. In both cases the LSTM yielded significant improvements.

2.4. Conclusion and Future Work

Our initial failures yielded interesting lessons on the nature of deep nets and seemed to reinforce some of the core principles of the deep learning approach. A common theme of deep learning research is a reluctance to interfere with the inner workings of a model beyond establishing it's architecture. That is, researchers will determine which parameters the model trains, but leave the actual task of training entirely up to backpropogation. Error or cost is entirely a function of the discrepancy between outputs and targets, and the researcher avoids making assumptions about the process by which the model minimizes this discrepancy.

Instead, they assume that the backpropogated error signal carries all the information that each parameter needs to optimize its contribution, no matter how indirect the contribution to the ultimate prediction may be. Consequently, the model learns best when the error signal flows back from the final layer in the network without any interference. Essentially, the role of the deep learning researcher it to create the pathways through which that error signal flows, but not to modify the content of that signal in any way as it travels through them.

Other approaches to machine learning tend to be much less laissez-faire about the researcher's role in training. The researcher might make strong assumptions about which features are most helpful to the model (in the case of hand-coded feature-extraction), or the shape of the data (which might influence a researcher's choice of classifier). Our efforts to "tell the model what to look for" violated a fundamental assumption of deep learning—that the model will find the optimal interpretation of the data without the help of the researcher. Associating each glimpse with the properties that our modified reward structures were based on (change in output, or decrease in standard deviation) amounted to a kind of feature-extraction. Modifying the reward structure by pushing it to focus on information with certain properties clearly contaminated the backpropogated error signal and inhibited the model from learning effectively.

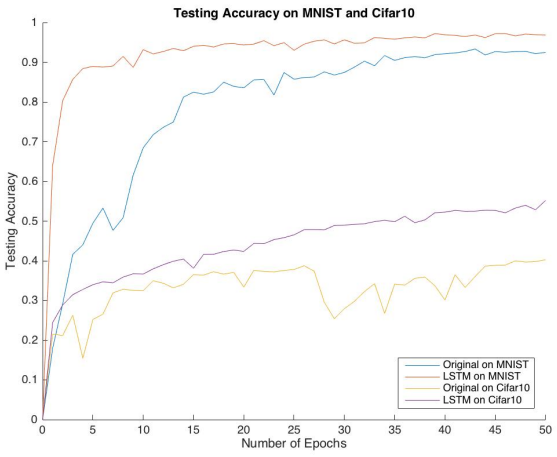

Figure 4. On datasets more complex than MNIST, such as the Cifar10 data set, our replacement of the locator network with an LSTM provided a greater impact on testing accuracy.

2.5. Future research

On the other hand, the improvement to both our model and the original after introducing the LSTM module clearly demonstrates the interdependence of attention and memory in deep networks. Even over a very short trial-period (six glimpses in the case of the Recurrent Attention Model), memory is critical to the performance of the model. Contextualizing each glimpse within the frame of the image and stitching together the information contained in each requires memory. This is intuitively evident, and the performance increase after introducing LSTMs seems to confirm this theory. As the tasks that attentive models undertake get more complex and span longer periods of time, these models come to increasingly rely on effective memory systems for storing and retrieving information over time.

Meanwhile, as memory structures become more complex, they become increasingly reliant on attention. In DeepMind's Neural Turing Machine paper, they introduced a Recurrent Neural Network that has access to external read/write memory and learns to use this memory to perform basic algorithms (I. Danihelka, 2014). The attention mechanism is a vector with a probability distribution over the memory addresses. Eventually, the model learns to focus its attention on a single memory address at a time, but nevertheless, the model performs matrix operations over every address in memory at every timestep. The size of memory in Deepminds' implementation is 128 x 20. However, external memory equivalent in size to that of a modern computer (or certainly a human brain) is far too large for this approach. Instead a learning model would need some kind attention mechanism to navigate around a larger and more complex memory structure. We believe that combining memory with effective attention mechanisms will dramatically improve the capabilities of deep nets.

References

A. Graves, N. Heess, K. Kavukcuoglu V. Mnih. Recurrent models of visual attention. *NIPS*, 2014.

B. Alexe, N. Heess, V. Ferrari Y.W. Teh. Searching for objects driven by context. *NIPS*, 2012.

Butko, N. and Movellan, J. Optimal scanning for faster object detection. *CVPR*, 2009.

Colombo, J. The development of visual attention in infancy. *Annual Review of Psychology*, 2001.

D. Ballard, M. Hayhoe. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 2005.

D. Eigen, R. Fergus, Y. LeCun M. Mathieu P. Sermanet X. Zhang. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, 2013.

G. Fritz, L. Paletta, C. Seifert. Q-learning of sequential attention for visual object recognition from informative local descriptors. *CVPR*, 2005.

G. Hinton, A. Krizhevsky, I. Sutskever. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

I. Danihelka, A. Graves, G. Wayne. Neural turing machines. *Archive*, 2014.

M. Craven, B. Settles. An analysis of active learning strategies for sequence labeling tasks. *EMNLP*, 2008a.

M. Craven, B. Settles, S. Ray. Multiple-instance active learning. *NIPS*, 2008b.

N. Leonard, S. Waghmare, Y. Wang. rnn : Recurrent library for torch7. *Archive*, 2015.

Ranzato, M. On learning where to look. *Archive*, 2014.

Rinsing, R. The dynamic representation of scenes. *Visual Cognition*, 2000.

S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

S. Mathe, C. Sminchisescu. Action from still image dataset and inverse optimal control to learn task specific visual scanpaths. *NIPS*, 2013.

T. Darrell, J. Donahue, R. Girshick J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, 2013.

W. Geisler, J. Najemnik. Optimal eye movement strategies in visual search. *Nature*, 2005.

Williams, R. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.