Low-supervision visual learning through cooperative agents

Ashish Bora
ashish.bora@utexas.edu

Abhishek Sinha
as1992@cs.utexas.edu

Abstract

There are a lot of unlabeled images available on the internet. Although we do not have explicit visual labels for these images, we observe that we can create a supervisory signal by asking a system to find one image from a collection of several images based only on a restricted signal. The signal can be a natural language description, or a question and an answer.

To explot this supervision, we propose a system of two agents which play a partial information cooperative game to complete the disambiguation task. We hypothesize that we can usefully restrict the communication type and direction to force these agents to learn visual cues. We expect that for interpretability of the communication, we would need some supervision on human interpretations of the symbols communicated and thus, we propose to use supervised pretraning followed by semi-supervised finetuning. In this work, we present an attempt at proof of concept of this idea using image attribute prediction as our underlying task. The project code can be found at `https://github.com/AshishBora/vision-project`

1. Introduction

Most of the success of deep neural networks has been in supervised learning tasks ([10], [21]). The main bottleneck in this approach is that it needs a lot of labeled training data ([15]). This is cumbersome and expensive.

More recently, there has been a flurry of work in transfer learning in deep neural networks. The prominent approach here is to initialize the network from a pretrained model on an auxillary task and finetune only the last few layers. This approach has shown good results on various tasks, especially due to availibility of models pretrained on imagenet category prediction task ([4], [14]). Transfer learning is still limited by availibility and compatibility of labeled data on the auxillary task. For example, spatial invariance is fundamental to category prediction, while not so desirable for object detection or spatial reasoning based Visual Question Answering (VQA).

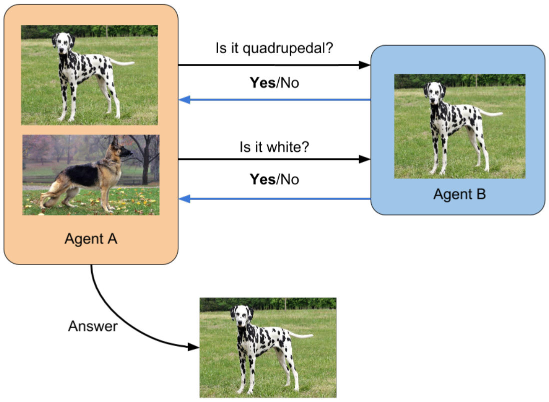

Figure 1. High Level Idea – A and B are 2 agents. A has images of 2 different dogs and C has image of one of the dogs (dalmation). A asks questions (attribute questions) and C answers them. Here we have simplifed the answers to Yes/No though in reality they would be confidences. Finally A must output its guess for the image possesed by B.

Unsupervised learning has been relatively less explored. The approach is essentially the same as in transfer learning, i.e. to pretrain on an auxillary task. The major difference is that the labels for the auxillary task are automatically generated. Stacked Denoising Autoencoders[2] try to reproduce the input from a corrupted version of it. Deep Belief Nets[6] pretraining tries to learn a generative model per layer. These methods are data domain agnostic.

Specifically for computer vision, the focus in many prior works has been to learn good image embeddings by exploiting structure in visual data. For example, [13], [19] use temporal consistency in videos, and [3] uses spatial consistency in single images to learn image embeddings. [8] learns image representations that are equivariant with respect to ego-motion transformations. These methods use the domain knowledge to extract labels, and design a specific loss function which is then used for unsupervised pretraining.

We propose a new unsupervised learning method which is data domain agnostic but task specific. In particular, our label extraction and loss are independent of data domain or task. On the other hand our framework uses a communication channel which can be tailored to suit the task at hand.

The rest of the paper is organized as follows. Section 2 presents our main idea. In section 3 we survey some related work. Section 4 briefly describes the attribute prediction task, related previous work, and the dataset we use. Section 5 describes a simplification of the general framework, which is used in this work to evaluate our idea. We show the network architecture we use in Section 6. Section 7 discusses the challenges and potential pitfalls in our system and our attempts at a solution to them. The next section outlines the training procedure we used. Section 9 presents the experimental results and analysis. Finally, we conclude with discussion and future work in sections 10 and 11.

2. Our Idea

Finding an object from a collection of objects, based on a restricted description is a basis of many popular games. For example in the game of *charade*, an entity or word is conveyed through miming or other physical communication. In *pictionary*, a word is to be communicated through drawings, but you are not allowed to write the word directly. In *guess who?* or *twenty questions*, the objective is to guess the card the other person has based on yes/no questions.

We observe that the common thread in all these tasks is disambiguation from a collection of several different objects based on limited but disambiguating information. Thus, we propose the following system:

The basic setup is as follows: There are two agents. The first agent (A) has a lot of image(s), while the other agent (B) has only one of them. Either agent does not know which images the other agent has. The task of A is to identify the image that B has. For this purpose, they can communicate with each other through some restricted communication channel.

It is important to recognize that there must be some restriction on the communication modality. Otherwise, the agents can communicate information which is not tied to high level visual concepts to distinguish the images. For example, they can exchange pixel values with locations .These will be very discriminative, but are not tied to high level visual understanding. Thus the idea is that by usefully restricting the communication modality, we can force these agents to learn interesting visual cues about the images.

Based on the restrictions on the communication modality and direction, we get different algorithms which will train the agents to do different tasks. The particular setup we explore here is question-answer type communication about attributes (see section 10 for other types). More concretely, the setup is as follows: A can send a question to B. Based on the image it has, B produces an answer to A's question. Based on the images it has, the question it asked and the answer it got, A decides either to ask another question or outputs its guess for the image it thinks B has. This is illustrated in Fig 1.

For our setting, the questions are about attributes in the images. Thus B learns attribute prediction and A learns to ask attribute based questions that can help disambiguate the two images it has.

3. Related Work

Our work is greatly inspired from the seminal work of [18]. In this paper, the authors describe a cooperative 2 player game called *Peekaboom*. In this game, there are 2 players called *Peek* and *Boom*. Boom initially has a word, image pair with him(or her) while Peek has a blank image. In each step Boom can only reveal a small portion of the image to Peek and in the end if Peek can guess the word, the 2 players win else they loose. Boom has an incentive to reaveal only those portions of the image which correspond to the word. Hence, this game provides an unsupervised way for creating word to bounding box mappings. We try to extend this idea of using cooperative games to create labelled data for computer vision algorithms to improving their performance.

Another work which is close to ours is [20]. In this paper, the authors describe a technique for generating question-answer pairs for a single image which they refer to as *Self Talk*. However, they train the Question Generation LSTM and Answering LSTM independently. The Question LSTM is trained using image and human question pairs whereas the Answer LSTM is trained using image, question and answer triplets. After this, self talk for a new image can be generated as follows – use the question generating LSTM to sample multiple questions and use the Answering LSTM to generate the corresponding answers. Our system is decidedly different from their's in generating the question-answer pairs in an unsupervised way where learning is carried out by disambiguating among images. Also, in their system question-answer generation seems to be the end goal whereas in our case it is a means to an end.

Another work that is closely related to ours is *Image Description with a Goal* by Sadovnik, Chiu et al [16]. In the paper, the authors try to generate a short description for a target image that discriminates it from a collection of images. They also describe a new quantitative metric to measure the effectiveness of the description. However, there is no notion of multi-round communication in their approach. Also they use handcrafted feature engineering to generate the scores for different items with respect to the target image.

Another work related to ours is *Image Specificity* by Jas and Parikh et al [7]. In this paper the authors introduce the notion of specific images which they define to be images that elicit consistent descriptions from different people. Intuitively pairs of images with high specificity scores are easier to disambiguate than pairs of ambgious images which have low specificity scores. However pairs of low specificity images seem to be more suited to our multi-round communication framework since we expect that multiple rounds of question-answering to search over a larger space of possibilities.

4. Attribute Prediction Task

Attribute prediction task is the following: We fix a finite set of visual attributes apriori. These usually describe a large part of the image such as dark, snowy, vegetation, etc. Given an image, we want to predict how strongly each attribute exists in that image.

As a baseline we use the Deep Carving model introduced in [17]. The average precision across all test images, as used in the same paper, is the metric we shall use for evaluation. Precision for single image is fraction of its top-k attributes that are also in the ground truth attributes, where k is the number of ground truth attributes. As per the paper, the Deep Carving model achieves average precision of 52.53, but we learn from the authors (and can reproduce) that a simple change in thresholding yields 61.01 average precision.

We use the SUN Weakly Supervised dataset introduced in [17] for training of C because it provides the attribute most strongly present for each training image. While training A and finetuning the whole system however, we can potentially use any (even unlabeled) images. To keep the input distribution similar to what was used to train C, we use the images from SUN Attribute dataset [12]

5. Simplification

To get a proof of concept and to debug the idea, in this work we consider the following simplified version of the system:

1. Agent A has only two images
2. There is only question-answer round

Since there is no back and forth question answering, we can split agent A into two agents (A and C) and create a feed-forward network. This is shown in Fig 2. Here, A is the questioning network: it takes two images as input and asks a question. B is the answering network: it has access to one of the images with A (randomly selected) and it answers A's question. C is the judgement network: it has the same two images that A has, and based on A's question and B's answer, it is supposed to ouput its guess for which image it thinks B has.

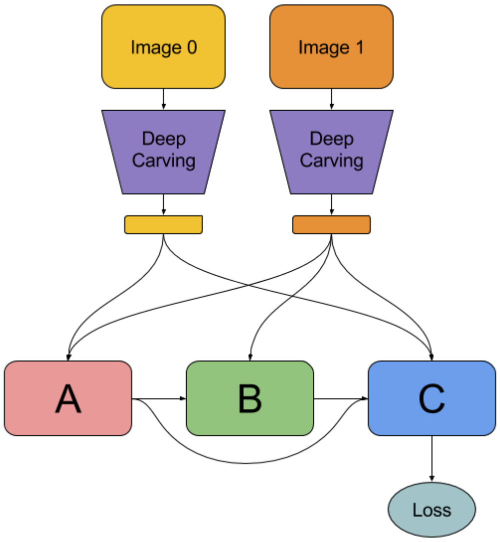

Figure 2. Simplified Architecture – Two Images, Single Question, Feed-forward

6. Network Architecture

We now describe in detail the architectures for A, B and C. Firstly, we note that while A, B and C all take images as input, in the diagrams we show them taking 4096 feature vectors as input for clarity.

A thus takes the two 4096 dimensional vectors and computes 42 dimensional attribute vectors from them using a fully connected layer. There is weight sharing between both the fully connected layers. This fc layer is followed by a sigmoid layer. The next layer is an 84 to 42 fully connected layer with a 42-way softmax layer in the end. The output of A which is a soft combination of attributes can be interpreted as a question vector.

B also has a 4096 to 42 fully connected layer followed by sigmoid layer. It then takes a dot product with the incoming question vector to produce a single value (confidence) as the output.

C initally has two units with the same architecture as B.

These two units share weights with each other and compute two values correponding to the two images. These two values are merged with the input confidence to get a 3-dimensional vector. After this there is a 3 to 2 fully connected layer followed by an Absolute Value non-linearity. Finally there is a 2 to 1 fully connected layer followed by sigmoid non-linearity. The value output is C's estimate of the probability of the Image1 being present with B.

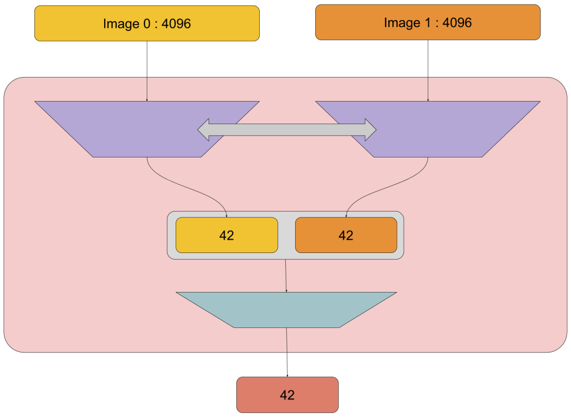

Figure 3. Questioning Network (A). Double Arrow indicates weight sharing and Trapezoids indicate fully connected layers. Non-Linearities are not shown to avoid clutter

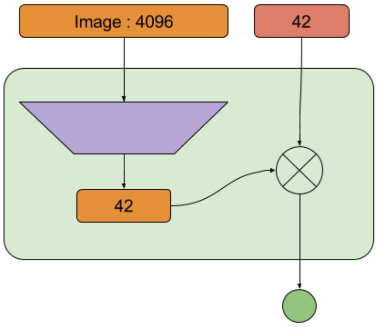

Figure 4. Answering Network (B). ⊗ indicates dot product

7. Challenges and Solutions

We describe the challenges we faced while getting this system to work followed by solution approached we tried.

1. **The interpretability problem:** The two agents can conspire to communicate in a strange code language that humans do not understand. This can make each learner useless in absence of the others. This problem is fundamental. If we do not provide our human interpretations to these agents, the unsupervised method has no way to become interpretable since any permutation mapping between human interpretations and code language interpretations works just fine for the purpose of disambiguation.

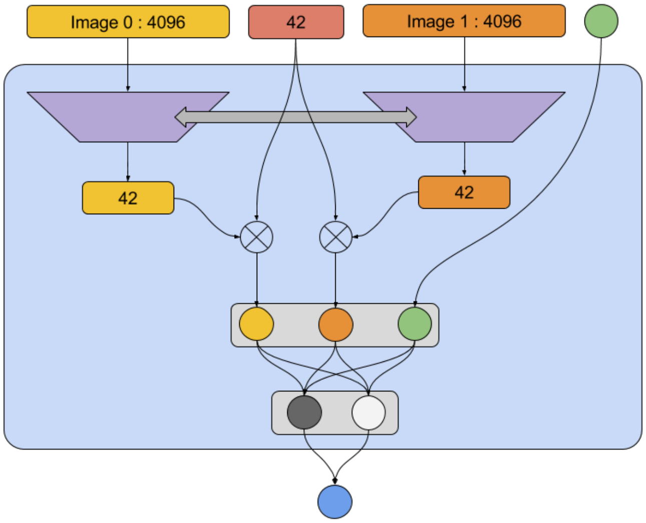

Figure 5. Judgement Network (C)

2. **Mimicking :** The two agents can simply learn to mimick each other exactly and not learn anything visual. i.e. they can both compute the same non-visual function from image to attributes and just compare the results.

3. **Weak Supervisory Signal :** The signal of whether the image was correctly identified or not is at most a single bit of information. It is challenging to backpropagate this signal through a deep network.

Based on the observations made above we deploy the following solutions which might help in training:

- **The interpretability problem:** First solution is supervised pretraining. i.e. We first pretrain B to do theattribute prediction task, but in a supervised way with human interpretable language. We then freeze B and train others to converse with B which forces them to learn human interpretations as well. Then, we finetune from there.

  We also tune B on original task intermittently while the whole system is being finetuned to avoid any potential drift from interpretability.

- **Mimicking :** To avoid this, we use dropout in all models. Thus due to randomness in dropout the models cannot rely on each other. We also do independent random cropping and horizontal flipping on images that A, B and C get. Thus, they see slightly different versions of the same input which will make them more robust.

Another thing to try would be to have very different architectuers for the two networks so that it is non-trivial to mimick each other. We have not used this idea in the current work.

- **Weak Supervisory Signal :** We expect that using a large batch-size and low learning rate would help in training.

8. Training Details

8.1. Encoding

Question encoding : We consider a set of 42 attributes as in the SUN Weakly Supervised datset. The question is then encoded as a probability distribution over these attributes.

Answer encoding : Answer is a single number which is the dot product of the question vector with the image attributes predicted by B. Thus, if A asks a question that peaks at a particular attribute, the output is close to the probability that the attribute specified in the question exists in the image with B. We could also have used one-hot encoded questions which are sampled from the softmax output of A, but we hypothesize that A will have to ask peaky questions since non-peaky questions are likely to be non-discriminative.

8.2. Training order

We follow the following training order:

1. Supervised pretraining of B to answer questions. (We just reuse the pretrained Deep Carving model)
2. Supervised pretraining of C to understand B and correctly identify the image B has. For this task, we need questions that will be asked to B. For the SUN Weakly supervised training dataset, for each image, we know the attribute which is most strongly present. We randomly pick strongest attribute of one of the images and ask a one-hot encoded question about it.
3. Semi-supervised training of A by fixing B and C : We just use the SUN Weakly supervised training set. We do not need any labels at this point.
4. Semi-supervised finetuning of A, B and C together : We use SUN Weakly Supervised as well as SUN Attribute DB images

For all tasks where two images are required, they are sampled independently from the training dataset. We do not explicitly ensure that they are from different classes.

8.3. Training objective

For training C, training A and then joint finetuning, we must define a loss based on whether the correct image is identified or not. The final Loss neuron (Fig 2) will be a sigmoid output between 0 and 1 representing a probability of the image being the 1st image. Let $(x_i, y_i)_{i=1}^{m}$ be the training set where each x_i is the ordered pair of two images and y_i is the index (0 for image-0 and 1 for image-1). of the image out of those two to be given to network B. Let θ be the parameters of the model. For the i-th training example, Let $p(x_i;\theta)$ be the output of the loss neuron and y_i be the actual label. We will use the following cross-entropy loss:

$$L(\theta) = \sum_{i}^{m} \left(y_i \log(p(x_i;\theta)) + (1-y_i)\log(1-p(x_i;\theta))\right)$$

9. Experiments

9.1. Experimental Procedure

We use the pretrained attribute prediction model of [17] for B. We use the following metrics to evaluate our models:

1. Average precision for attribute prediction: initially and afte finetuning B with our procedure
2. Accuracy for image disambiguation across all the examples
3. Qualitative inspection of the kind of attribute questions asked by A

9.2. Results and analysis

9.3. Experiment 1 : Fix B, train C with mimicking

We finetuned C with B and C mimicking the weights of the Deep Carving model. This should ideally very quickly converge to a very low error. This experiment serves two purposes: first it helps ensure that the our implementation is correct. Secondly, it gives a sanity check that the model architecture has enough capacity to perform its task. We also shut down random cropping, horizontal flips and dropout for this sanity check experiment. Fig 6 shows the training curves. As expected we achieve very high accuracy (about 96%).

9.4. Experiment 2 : Fix B, train C without mimicking

We trained C from scratch with B initialized to be the Deep Carving model. Fig 7 shows the training curves. We achieve

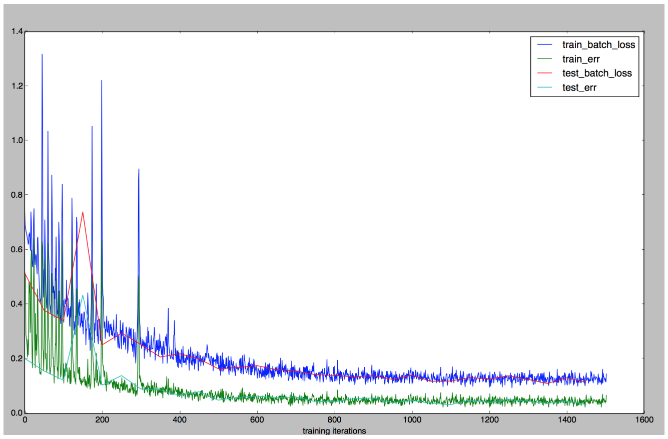

Figure 6. Learning Curve for C with mimicking

about 86% accuracy. At this point we stop the training since otherwise C will start copying weights from B. Thus, to maintain some difference, we use early stopping.

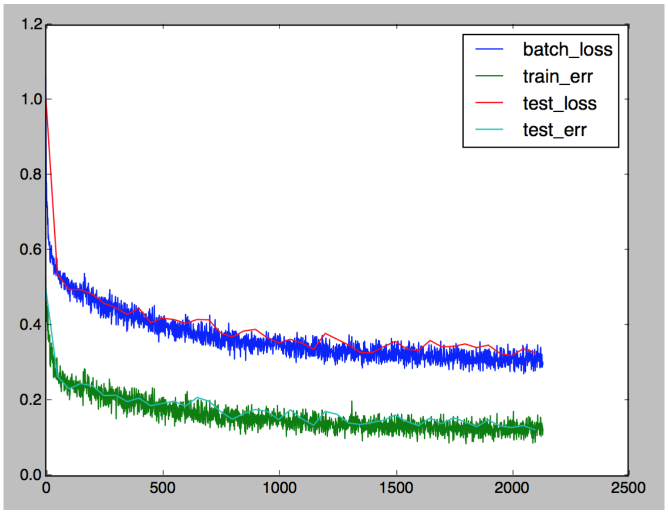

Figure 7. Learning Curve for C trained from scratch

9.5. Experiment 3 : Fix B and C. Train A

We trained A from scratch with B initialized and fixed to the Deep Carving model, and C fixed to be the model from the previous experiment. Fig 8 shows the training curves. We see that no learning whatsoever is taking place. We confirmed that A is indeed recieving gradients at each iteration. Although it is not clear why this happens, one possible reason is that C is mimicking B. Thus, there is no need for A to ask a good question: alsmost every question is disambiguating enough.

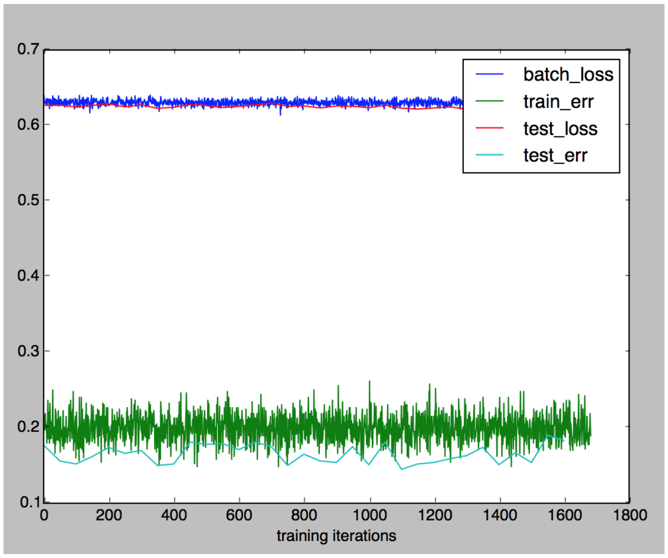

Figure 8. Learning Curve for A

9.6. Experiment 4 : Finetune the whole system

We use A and C models as obtained from the previous experiments. We finetune the whole system on images from SUN Weakly supervised dataset as well as the full SUN dataset. At this point the training procedure does not need any labels and can be tuned on unlabeled images as well. We used SUN images because we wanted the unlabeled images to have roughly the same distribution as lableled ones. The training curve is shown in Fig 9. We see that there is very little improvement initially, followed by a steep decrease and then the error stabilizes again. On the other hand, we do not see any improvement on the original attribute prediction task as we finetune B.

The finetuning procedure is minimizing a different loss function that the original attribute prediction task. Thus, it is expected that the new loss surface will have a minima at a different location than the local minimum achieved by the Deep Carving model. Nonetheless, it is quite interesting that the finetuning procedure did not break the original model. Our hypothesis is that the minimum of the unsupervised finetuning method and that of the deep carving method are very close and that is why the performance is not hurt by a large margin. Since the minimas would most likely be different, we still see a small drop from 0.61 to 0.605 in the average precision on the attribute model prediction (See Fig 10)

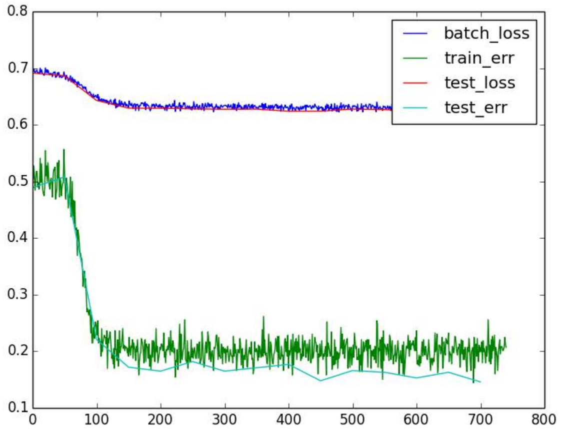

Figure 9. Learning Curve for B

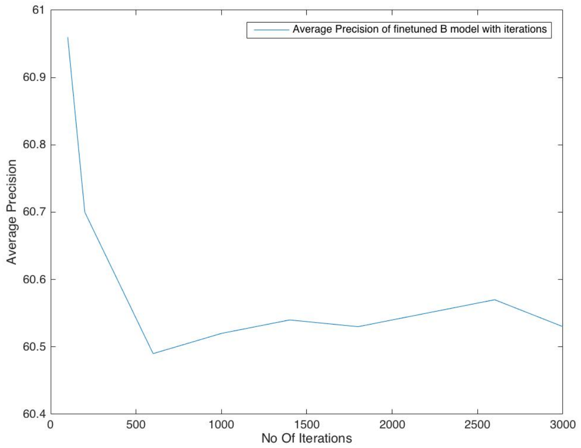

Figure 10. Variation of Average Precision of finetuned B model with number of iterations

10. Future Work

We should first investigate why A is not learning anything at all.

The approach is quite general, but we have restricted it to a simplified model as a first step. Thus, the idea can be applied to more complicated tasks such as VQA [1][11] and Dense Captioning [9] [5].

For VQA, the question can be encoded as a natural language question followed by the ? token. Answer can be taken to be single word or even natural language answer. Both question and answer can be represented as sequence of one-hot vectors over a fixed vocabulary or as word2vec vectors. Due to variable length of the questions, and answers we can use RNNs for all the agents.

Another extension is to use multiround communication, i.e. several messages are exchanged between the two agents back and forth. This would obviously demand that each agent be recurrent.

For Dense Captioning task, the communication restriction is a bit different. In this case, B gives natural language descriptions of the image it has. Based on this A has to tell whether it can guess the image that B has. If not, it requests another description from B (at some cost). Finally A has to output a guess for B's image and gets rewarded if its correct. It is easy to see that multiround communication is vital for this procedure to work.

11. Conclusion

We proposed the novel idea of cooperating agents for low-supervision visual learning. Though we haven't got any positive results till now, we are investigating the pitfalls and plan to make changes in our system to address them.

Acknowledgements: Both authors would like to acknowledge the help from Sukrit Shankar, the lead author of the Deep Carving paper for sharing all the paper related code and patiently responding to multiple emails when we were having a hard time reproducing the original results.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 153–160. MIT Press, 2007.

[3] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[5] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. *arXiv preprint arXiv:1511.05284*, 2015.

[6] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[7] M. Jas and D. Parikh. Image specificity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2736, 2015.

[8] D. Jayaraman and K. Grauman. Learning image representations equivariant to ego-motion. *arXiv preprint arXiv:1505.02206*, 2015.

[9] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–9, 2015.

[12] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012.

[13] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4479, 2015.

[14] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 199–207. Curran Associates, Inc., 2015.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[16] A. Sadovnik, Y.-I. Chiu, N. Snavely, S. Edelman, and T. Chen. Image description with a goal: Building efficient discriminating expressions for images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2791–2798. IEEE, 2012.

[17] S. Shankar, V. K. Garg, and R. Cipolla. Deep-carving: Discovering visual attributes by carving deep neural nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3403–3412, 2015.

[18] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006.

[19] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

[20] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos. Neural self talk: Image understanding via continuous questioning and answering. *arXiv preprint arXiv:1512.03460*, 2015.

[21] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.