Zero-shot learning using dictionary definitions

Hascoet Tristan
Graduate school of System Informatics
Kobe University
Kobe, Japan

Ariki Yasuo
Organization of Advanced Science and Technology
Kobe University
Kobe, Japan

Tetsuya Takiguchi
Organization of Advanced Science and Technology
Kobe University
Kobe, Japan

Zero-shot learning (ZSL) models use semantic representations of visual classes to transfer the visual knowledge learned from a set of source classes to a disjoint set of target classes. In this paper, we propose to learn these representations from dictionary definitions using an LSTM model. We explore several variants over our baseline, including the addition of an attention mechanism and pretraining on various tasks. We found that neither the attention mechanism nor pretraining of the LSTM model on pure NLP tasks had any effects on the model's accuracy. However, pretraining the model on the more related task of image retrieval yielded large gains in accuracy, suggesting a promising direction for future research.

Keywords—CNN, LSTM, Zero-shot learning

I. INTRODUCTION

Zero-shot learning models can be seen as the combination of three different modules as illustrated in *Figure 1*: The visual module extracts visual features from raw images; in this work, we use a Resnet-50 convolutional neural network [1]. The semantic module extracts class-wise semantic features from raw descriptions of the visual classes. In this work, we propose to use an LSTM to process the dictionary definitions of the visual classes. The core ZSL module assigns a similarity score between the visual and semantic features respectively extracted by the two lower modules. In this paper, we used a bilinear model. Together, the full model takes as input a pair *(x,y)* of raw image *x* and class definition *y* and outputs a similarity score between both inputs.

Training is performed by optimizing a contrastive loss function with stochastic gradient descent so as to maximizes the similarity score of matching inputs pairs *(x,y)*, i.e., for dictionary definition *y* matching the object being represented in input image *x*, while minimizing it for non-matching pairs.

Zero-shot learning models are classifiers able to generalize to classes unknown at training time. To assess the ability of our model to generalize to unseen classes, we train our model on a fixed set of training classes C_{test} and, at test time, we evaluate the generalization ability of our model on a disjoint set of test classes C_{test} so that $C_{test} \cap C_{train} = \emptyset$

We used images from the Imagenet dataset and dictionary definitions from Wordnet. In total, our dataset consists of more than 14 million images across 20,000 classes. One definition is given per class and each definition is made of between 5 and 50 words, with an average of 15 words per definition which is relatively little data with regard to learning in the LSTM. Our initial intuition was that by pretraining the LSTM on NLP tasks, it would learn some structure from sentences that would yield better representations of the visual class definitions after fine-tuning on our task at hand. Hence we first experimented on pretraining the LSTM module on the tasks of Neural Language Modeling and Document Classification.

In a second attempt, we tried pretraining our LSTM on the task of image retrieval. To do so, we used a popular image captioning dataset and casted the problem of image retrieval as an image classification task in which each pair of (image,caption) represents a class of its own.

Lastly, we experimented with adding an attention mechanism on top of the LSTM model. The intuition behind this addition was that, as illustrated by words marked in bold in Figure 1, some words of the definition contain more visually discriminative information than others.

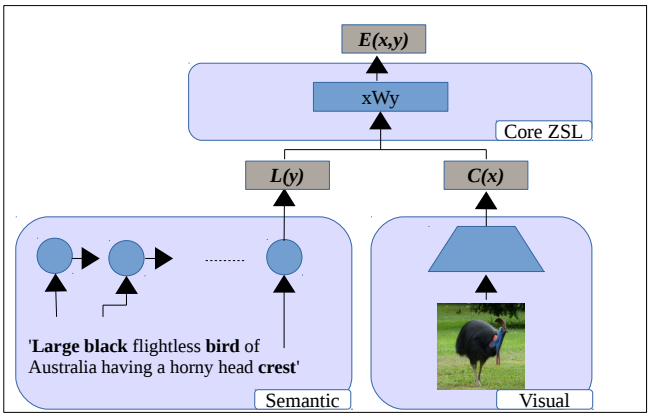

Figure 1: Illustration of our model architecture

The rest of this article is organized as follows. In section II., we present existing works related to ours. Section III. describes in more details our model, our optimization procedure as well as each of the improvements described above. Section IV. presents our results and we conclude in Section V.

II. Related Work

Early work on ZSL focused on well established benchmarks of domain-specific image datasets such as animals [2] or birds [3]. This line of work uses handcrafted binary visual attributes as semantic features to represent the visual classes, thus bypassing the need of the semantic module in the architecture of *Figure 1*. More recently [4] proposed the use of word embeddings as semantic representation of visual classes. This is attractive as word embeddings can be computed in an unsupervised manner on large text corpora so that there is no need for expert annotations of visual attributes for visual classes. Using word embeddings as visual class semantic features has allowed to generalize the zero-shot learning setting from domain specific classification tasks to generic object recognition settings.

In [5], the authors train an LSTM on Wikipedia descriptions. They combine learning of the LSTM model with other semantic representations in an end-to-end differentiable deep architecture. In our work, we only use Wordnet definitions as textual descriptions and focus on enhancing the performance of the LSTM module by means of pretraining on various tasks.

III. Proposed method

A. Baseline model

Our model is illustrated in *Figure 1*. We initialized the visual module with a Resnet-50 pretrained on the ILSVRC2012 image classification dataset. We used pretrained GloVe word embeddings to encode the inputs of the semantic module. Encoded definitions *y* fed as input to the *LSTM* model, and we denote by $L(y)$ the final output of the LSTM model. Given an input image x, we denote by $C(x)$ the activation value of the top hidden layer of the visual module.

We train our model to minimize the following loss function:

$$L=\sum_{i=1}^{i=N}(\cos(C(x_i),L(y_i))-\sum_{k=1\ \ i\neq i}^{k=n}\cos(C(x_j),L(y_j))) \qquad (1)$$

where ***N*** denotes the number of sample in the training set, ***cos*** denotes the cosine distance and ***k*** is a negative sampling factor parameterizing our model.

B. Optimization

In its largest version, our dataset consists of over 20,000 visual classes that amounts to more than 14 million images. Randomly accessing such large data from a regular hard drive creates an important bottleneck in our computation pipeline so that optimizing a full epoch over the dataset takes more than a day, despite parallelizing disk accesses and running the computations on GPU. To speed up the learning and improve the stability of learning in the LSTM module, we decompose the training procedure as follow:

After pretraining the CNN, we cache in memory, for each class, the mean activation values of the CNN's top layer over their training sample: given a visual class c with N_c samples $\{x_{c,i}\}_{i\in N_c}$,we compute:

$$x_c=1/N_c\times\sum_{i\in N_c}x_{c,i} \qquad (2)$$

We then pretrain the LSTM module on the class-wise mean activation values x_c following equation (1). This way training of the LSTM module can be performed in-memory which considerably speeds up the learning.

Finally, the model can be fine-tuned sample-wise over the full dataset. During this phase, the error can be back-propagated through both modules in an end-to-end learning. In our experiments we explored various parameterization of our model, so that, for time constraints, the results presented in section *IV. w*ere computed without sample-wise fine-tuning.

We minimize equation (1) by stochastic gradient descent, randomly sampling mini-batches of 20 samples from the full training set. For each correct pair sample (x_i,y_i) , we randomly sample $k=19$ erroneous pairs (x_i,y_j) . We gradually decay the learning rate from 10^{-2} to 10^{-4} along training.

C. Improvements

1) Attention Mechanism

As illustrated by the words marked in bold characters in Figure 1., some words of the dictionary definitions are clearly more visually discriminative than others. To encode this prior in our model, we added a soft attention mechanism on top of the LSTM output. However, as presented in section *IV. (AT)*, this addition yielded little to no improvement.

2) LSTM Pretraining

The LSTM module is trained with 20,000 Wordnet definitions made of 5 to 50 words. The size of this dataset arguably does not allow for much learning in the LSTM model. Hence, we explored pretraining the LSTM module on several different tasks.

- **Pretraining as language modeling**: Given the relatively small size of our training text corpora, we question if our model could benefit from large unsupervised pretraining. In a first experiment, we pretrained the LSTM model as a Neural Language Model (NLM) on the English Wikipedia corpus. NLM are trained in an unsupervised manner so they can be trained on very large text corpora without requiring any labeling.

- **Pretraining as document classification**: In a second experiment, we pretrained the LSTM model on a Document Classification (DC) task. Imagenet classes represent a subset of 20,000 out of the full 117,000 concepts defined in Wordnet. In

this experiment, we pretrain the LSTM model on the task of classifying each input definition into its associated Wordnet concept. This contrasts to the NLM pretraining setting in which the LSTM model was trained for a predictive task on data from a different domain (the Wikipedia corpus). In this experiment, we pretrain the LSTM in a discriminative setting on similar domain data (the Wordnet definition). This corpus, however, is much smaller than that of the unsupervised NLM setting.

- Pretraining as image retrieval:

Lastly, we pretrain our model on an image retrieval task using the COCO captioning dataset (IC).

Figure 2: Examples of (image, caption) pairs from the COCO captioning dataset on the top two rows and (image,definition) pairs from the Imagenet/Wordnet dataset on the lower row

In this setting, we treat each (image,caption) pair of the dataset as a class of its own. Similar to ZSL training, we extract visual feature representations from COCO images as the CNN's top layer activation values and update the LSTM weights so as to minimize equation (1) by stochastic gradient descent.

Figure 2 gives some examples of image/caption pairs used for image retrieval pretraining and illustrates their difference to image/definition pairs of the target ZSL task. We found interesting differences among captions. Samples in the top row are illustrative of more *narrative* captions. In these examples, the caption mentions several different objects scattered through the image in a coherent narrative. In contrast, samples of the second row are illustrative of more *descriptive* captions. Those captions mainly focus on a single object of the image, describing it in a manner reminiscent to dictionary definitions (illustrated in the last row). Every caption can not be categorized as fully descriptive or narrative as most captions lie somewhere in the middle of these two categories. This figure illustrates the gap between caption domain and definition domain. This gap seems to be wider for some captions that tend to be more narrative and narrower for more other, more descriptive captions.

In this experiment, we first pretrain our LSTM model on the COCO captioning dataset and then fine tune it on Wordnet definitions. We show that, fine-tuning the model on a large enough number of training classes improves the accuracy over the pretrained image-retrieval model, suggesting that the gap between caption and definition domain is an import factor.

IV. Experiments and Results

A. *Experiments*

In a first experiment, we used a fixed test set of 200 classes as was proposed in [4] and train our model with a training set of increasing number of classes. Training classes were randomly sampled from the whole Imagenet dataset. Results presented in the following figures were averaged over 5 runs of different training classes to reduce the noise due to the random selection of the training classes.

In a second experiment, we used a fixed training set of 5,000 classes and randomly sampled classes for test sets of different sizes. We average our results over 5 runs of different randomly drawn test classes. In both experiments, we report our results in terms of top-k accuracy as traditionally reported in Imagenet challenges.

B. *Results*

Figure 3. shows the top-k classification scores obtained by the different variations of our model for different k and different sizes of training set.

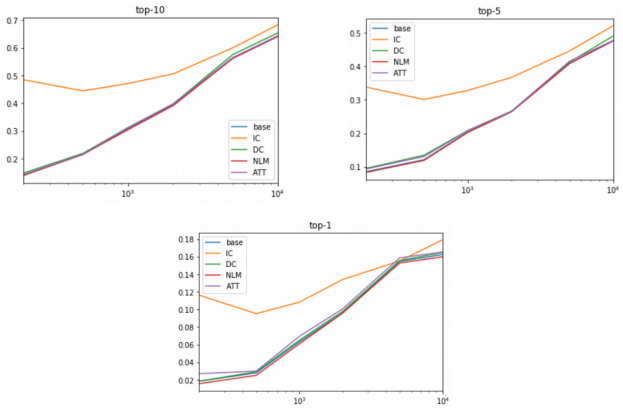

Figure 3: Top-k classification accuracy for different number of training classes on a 200 test-split. The x-axis shows the number of classes in logarithmic scale

Both the attention mechanism (ATT) and the NLP pretraining (NLM and DC) did not significantly affect the model accuracy. In contrast, pretraining on the image retrieval task (IC) yielded very positive results. We observed that fine-tuning the image retrieval model on small (200-1000 classes) training sets degraded the accuracy as the model overfits to the small set of training classes. However, for larger number of classes, fine-tuning did improve on the accuracy of the image-retrieval model. An interesting result is that the gain in accuracy from image-caption pretraining persisted, albeit diminished, even for large number of training classes. In this case, and contrarily to the other two pretraining methods we experimented with, this confirmed our hypothesis that the LSTM model would benefit from pretraining on different domain data and fine-tuning on similar domain data.

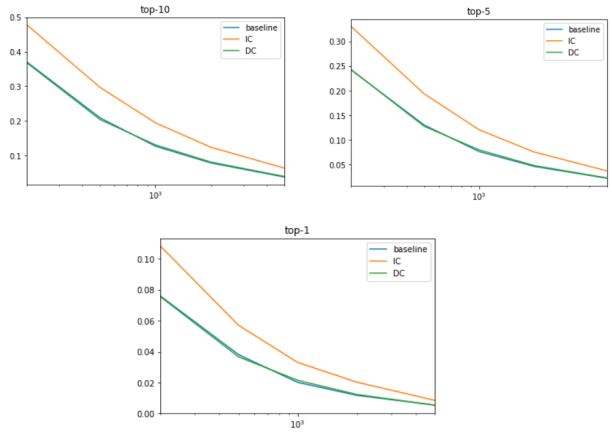

Figure 4: Top-k classification results for different number of training classes on a 5000 train-split. The x-axis shows the number of classes in logarithmic scale

Figure 4. shows the result of our second experiments. This experiment highlighted similar trends: Pretraining on the image retrieval task did improve on the model's accuracy while NLP pretraining (DC) did not significantly affect the model's accuracy.

C. Future work

As dictionary definitions are not specifically designed to be visually discriminative, they contain a lot of noisy information, seemingly useless for visual recognition tasks, which led us to believe that we could gain in accuracy by adding an attention mechanism on top of the LSTM to filter out non visually discriminative words. Although the attention mechanism proposed in our model did not yield significant improvement, we will keep exploring this direction in future work. One possible direction would be to implement a co-attention mechanism that simultaneously attends image and text contents.

IV Conclusion

In this paper, we proposed to use an LSTM model to extract visual class representations from dictionary definition. We were hoping that using sophisticated text processing models such as an LSTM either pretrained as a Neural Language Model on a very large text corpus or as a document classifier on a smaller corpus of similar domain would increase the accuracy of our ZSL model. Instead, we observed the opposite effect as the accuracy either decreased (DC) or stagnated (NLM) with pretraining. However, pretraining the LSTM module on an image retrieval task yielded promising results. Without fine-tuning, the raw image retrieval model gave result comparable to medium-sized (1000 classes) training sets. For larger training sets, fine-tuning further improved the accuracy of the raw image retrieval module. Most importantly, the benefits of pretraining as an image retrieval task persisted even for larger sizes of training set. This confirmed our hypothesis that the LSTM module would gain from both pretraining on different domain data (narrative/descriptive captions) and fine-tuning on similar domain data (descriptive dictionary definitions.)

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[2] C. H. Lampert, H. Nickisch, and S. Harmeling. "Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer". In CVPR, 2009

[3] P., Branson S., Mita T., Wah C., Schroff F., Belongie S., Perona, P. "Caltech-UCSD Birds 200". California Institute of Technology. CNS-TR-2010-001. 2010

[4] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., & Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems* (pp. 2121-2129).

[5] *Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, Ruslan salakhutdinov*; Predicting Deep Zero-Shot Convolutional Neural Networks Using Textual Descriptions. The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4247-4255