Large Scale Visual Semantic Extraction

Samy Bengio[1]
Google Research

Abstract

Image annotation is the task of providing textual semantic to new images, by ranking a large set of possible annotations according to how they correspond to a given image. In the large scale setting, there could be millions of images to process and hundreds of thousands of potential distinct annotations. In order to achieve such a task we propose to build a so-called "embedding space", into which both images and annotations can be automatically projected. In such a space, one can then find the nearest annotations to a given image, or annotations similar to a given annotation. One can even build a visio-semantic tree from these annotations, that corresponds to how concepts (annotations) are similar to each other with respect to their visual characteristics. Such a tree will be different from semantic-only trees, such as WordNet, which do not take into account the visual appearance of concepts.

1. Introduction

The emergence of the web as a tool for sharing information has caused a massive increase in the size of potential datasets available for machines to learn from. Millions of images on web pages have tens of thousands of possible annotations in the form of HTML tags (which can be conveniently collected by querying search engines (Torralba et al, 2008)), tags such as in www.flickr.com, or human-curated labels such as in www.image-net.org (Deng et al, 2009). We therefore need machine learning algorithms for image annotation that can scale to learn from and annotate such data. This includes: (i) scalable training and testing times, and (ii) scalable memory usage. In the ideal case we would like a fast algorithm that fits on a laptop, at least at annotation time. For many recently proposed models tested on small datasets, it is unclear if they satisfy these constraints.

In the first part of this work, we study feasible methods for just such a goal. We consider models that learn to represent images and annotations jointly in a low dimension embedding space. Such embeddings are fast at testing time because the low dimension implies fast computations for ranking annotations. Simultaneously, the low dimension also implies small memory usage. To obtain good performance for such a model, we propose to train its parameters by learning to rank, optimizing for the top annotations in the list, e.g. optimizing precision at k (p@k).

In the second part of this work, we propose a novel algorithm to improve testing time in multi-class classification tasks where the number of classes (or labels) is very large and where even a linear algorithm in the number of classes can become computationally infeasible. We propose an algorithm for learning a tree-structure of the labels in the previously proposed joint

[1]This work summarizes the following papers: (Weston et al, 2010) with Jason Weston and Nicolas Usunier, and (Bengio et at, 2010), with Jason Weston and David Grangier.

embedding space which, by optimizing the overall tree loss, provides a superior accuracy to existing tree labeling methods.

2. Joint Embedding of Images and Labels

We propose to learn a mapping into a feature space where images and annotations are both represented, as illustrated in Figure 1. The mapping functions are therefore different, but are learnt jointly to optimize the supervised loss of interest for our final task, that of annotating images. We start with a representation of images $x \in \Re^d$ and a representation of annotations $i \in \Upsilon = \{1,..Y\}$, indices into a dictionary of possible annotations. We then learn a mapping from the image feature space to the joint space $\Re^D$:

$$\Phi_I(x): \Re^d \rightarrow \Re^D$$

while jointly learning a mapping for annotations:

$$\Phi_W(i): \{1,...,Y\} \rightarrow \Re^D.$$

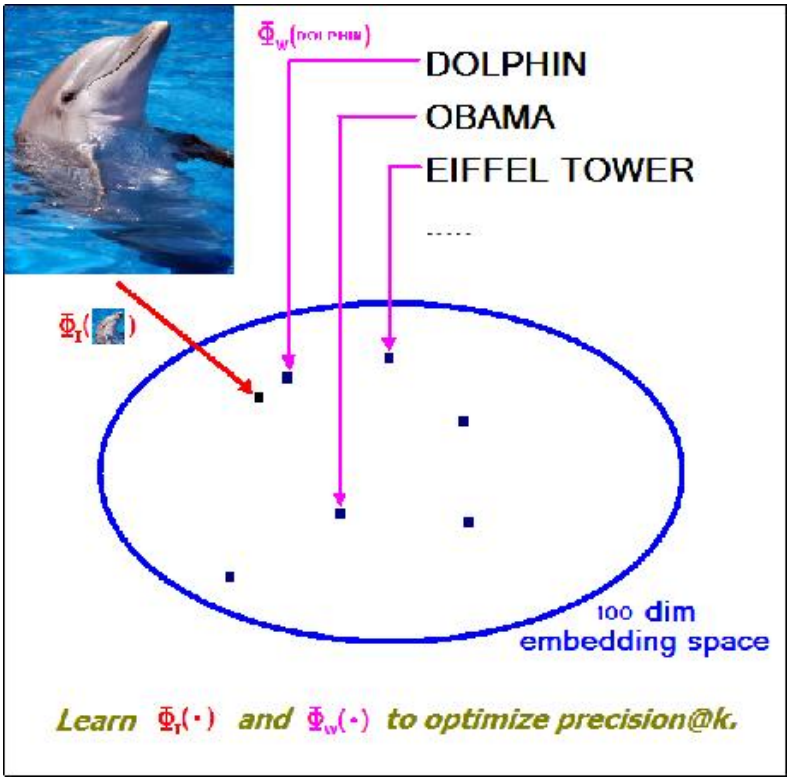

Figure 1: Joint Embedding Space

These are chosen to be linear maps, i.e. $\Phi_I(x) = Vx$ and $\Phi_W(i) = W_i$, where W_i indexes the i^{th} column of a $D \times Y$ matrix, but potentially any mapping could be used. In our work, we use sparse high dimensional feature vectors of bags-of-visual terms for image vectors x and each annotation has its own learnt representation (even if, for example, multi-word annotations share words). Our goal is, for a given image, to rank the possible annotations such that the highest ranked annotations best describe the semantic content of the image. We consider the following model:

$$f_i(x) = \Phi_W(i)^T \Phi_I(x) = W_i^T V x$$

where the possible annotations i are ranked according to the magnitude of $f_i(x)$, largest first.

2.1. Image Labelling as a Learning-To-Rank Task

Labelling an image can be viewed as a ranking task where, given an image, one needs to order labels such that the top ones correspond to the image, while the bottom ones are unrelated to it. Various learning-to-rank methods have been proposed in the machine learning literature over the years, some of which can scale to large datasets (while others can't). The simplest scalable approach is the following: one can decompose the ranking task as a large sum of several smaller tasks:

$$loss = \sum_x \sum_{y^+} \sum_{y^-} \left| 1 - f_{y^+}(x) + f_{y^-}(x) \right|_+$$

where for each training image x, we want the score of each good label y^+ to be higher than the score of any bad label y^- by a margin of at least one, otherwise we pay the corresponding price. This loss can be trained very efficiently on very large datasets using stochastic gradient descent. However, a better alternative would be a loss that concentrates on the top of the ranking, instead of considering every triplets (x, y^+, y^-) uniformly. In (Weston et al, 2010), we proposed the WARP loss, which can weigh each of the triplets according to the estimated rank of the good labels, and still yield an efficient implementation. The resulting model is much faster to train and obtains a much better performance at the top of the ranking.

2.2. Large Scale Learning

We trained an embedding model with the WARP loss on a very large dataset, containing more than 10 million training images and more than 100 thousand labels, where labels correspond to queries uttered on Google Image Search and images attributed to these labels were images often clicked for these queries. That meant a very noisy data, where queries are in several languages, with many spelling mistakes an many apparently similar queries.

An interesting side effect of training such a model is that it provides a natural way of organizing labels among themselves, by looking at the nearest labels of a given label in the embedding space. Table 1 shows some examples of the nearest labels of some labels, where we see several misspellings, translations, and semantically similar labels.

Target Label	**Nearest Labels**
barack obama	barak obama, obama, barack, barrack obama, bow wow, george bush
david beckham	beckham, david beckam, alessandro del piero, del piero, david becham
santa	santa claus, papa noel, pere noel, santa clause, joyeux noel, tomte
dolphin	delphin, dauphin, whale, delfin, delfini, baleine, blue whale, walvis

cows	cattle, shire, dairy cows, kuh, horse, cow, shire horse, kone, holstein
rose	rosen, hibiscus, rose flower, rosa, roze, pink rose, red rose, a rose
pine tree	abies alba, abies, araucaria, pine, neem tree, oak tree, pinus sylvestris
mount fuji	mt fuji, fujisan, fujiyama, mountain, zugspitze, fuji mountain
eiffel tower	eiffel, tour eiffel, la tour eiffel, big ben, paris, blue mosque, eifel tower
ipod	i pod, ipod nano, apple ipod, ipod apple, new ipod, ipod shuffle
f18	f 18, eurofighter, f14, fighter jet, tomcat, mig21, f 16

Table 1: Nearest labels in the embedding space learnt on the Web-data.

Finally, we show in Table 2 examples of images from our test set (there was more than 3 millions of them, different from the 10 millions of training images), as well as the nearest 10 labels in the embedding space.

Target Image	**Nearest Labels**	**Target Image**	**Nearest Labels**
	delfini, orca, dolphin, mar, delfin, dauphin, whale, cancun, killer whale, sea world		barrack obama, barack obama, barack hussein obama, barack obama, james marsden, jay z, obama, nelly, falco, barack
	eiffel tower, statue, eiffel, mole antoneliana, la tour eiffel, londra, cctv tower, big ben, calatrava, tokyo tower		ipod, ipod nano, nokia, i pod, nintendo ds, nintendo, lg, pc, nokia 7610, vino

Table 2: Examples of the top 10 labels obtained for some test images.

3. Learning Label Trees

Labelling images when the number of labels is large (in Section 2, we had on the order of 100,000 labels) can be prohibitive for real-time applications. We thus proposed in (Bengio, 2010) a novel approach to learn a Label Tree, where each node makes a prediction of the subset of labels to be considered by its children, thus decreasing the number of labels at a logarithmic rate until a prediction is reached. Existing approaches (Beygelzimer et al, 2009a, Beygelzimer et al, 2009b, Hsu et al, 2009) typically lose accuracy compared to naive linear time

approaches. Instead, we apply the following two steps: (a) learning a label tree, and (b) learning predictors for each of the nodes of the tree.

3.1. Learning the Label Tree Structure

In order to learn a label tree such as the one in Figure 2, we proceed as follows: given a set of labels in a node, we look for a partition of that set into subsets such that inside a subset, labels are difficult to distinguish with classifiers trained on their corresponding images, while it is easier to distinguish images belonging to labels of a subset from images belonging to labels of another subset. We do so by computing the confusion matrix between all labels, where we count the number of times our classifiers confuse class i with class j, and use this matrix to apply spectral clustering (Ng et al, 2002). This procedure can then be applied recursively to obtain a complete label tree. Table 3 gives an examples of labels that were clustered together thanks to that technique.

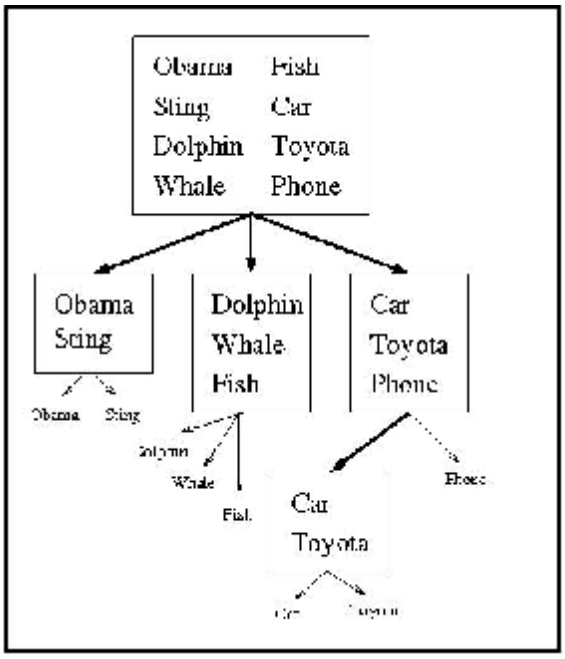

Figure 2: Example of a label tree.

great white sharks, imagenes de delfines, liopleurodon meduse, mermaid tail, monstre du loch ness, monstruo del lago ness, oarfish, oceans, sea otter, shark attacks, sperm whale, tauchen, whales	apple iphone 3gs, apple ipod, apple tablet, bumper, iphone 4, htc diamond, htc hd, htc magic, htc touch pro 2, iphone 2g, iphone 3, iphone 5g, iphone app, iphone apple, iphone apps, iphone nano	chevy colorado, custom trucks, dodge ram, f 250, ford excursion, ford f 150, mini truck, nissan frontier, offroad, pickup, toyota tundra

Table 3: Examples of obtained clusters of labels.

3.2. Learning a Label Embedding Tree

Once labels are organized into a tree, one can retrain jointly an embedding model (using the algorithm described in Section 2) where each image can now be labeled either with its original labels, or any of the nodes of the tree that contains them. Moreover, whenever an internal node is selected as a positive label for a given image during training, we select a competing negative label as a sibling node in the label tree, as this corresponds to how the tree would then be used at test time.

The result provides a structure of labels based on both semantic and visual similarities. Furthermore, the performance of a label embedding tree is not only faster at test time, it is also better on average, as can be seen in (Bengio et al, 2010).

References

[Bengio et al, 2010] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In Advances in Neural Information Processing Systems, NIPS, 2010.
[Beygelzimer et al, 2009a] A. Beygelzimer and J. Langford and P. Ravikumar. Error-correcting tournaments. In International Conference on Algorithmic Learning Theory, ALT. pp. 247-262, 2009.
[Beygelzimer et al, 2009b] A. Beygelzimer and J. Langford and Y. Lifshits and G. Sorkin and A. Strehl. Conditional Probability Tree Estimation Analysis and Algorithm. In Conference in Uncertainty in Artificial Intelligence, UAI, 2009.
[Deng et al, 2009] Deng, J. and Dong, W. and Socher, R. and Li, L.-J. and Li, K. and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009.
[Hsu et al, 2009] D. Hsu and S. Kakade and J. Langford and T. Zhang. Multi-Label Prediction via Compressed Sensing. In Advances in Neural Information Processing Systems, NIPS, 2009.
[Ng et al, 2002] Ng, A.Y. and Jordan, M.I. and Weiss, Y. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems, NIPS, 2002.
[Torralba et al, 2008] A. Torralba and R. Fergus and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. In IEEE Transactions on Pattern Analysis and Machine Intelligence. vol 30, issue 11, pp. 1958-1970, 2008.
[Weston et al, 2010] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML-PKDD, 2010.