CS231A Course Project Milestone Report
Unsupervised Multi-Modal Feature Learning: Images and Text

Maurizio Calo Caligaris
Stanford University
maurizio@cs.stanford.edu

Abstract

Hand-engineering task-specific features for single modalities (e.g. vision) is a difficult and time-consuming task. Furthermore, the challenge gets significantly more pronounced when the data comes from multiple sources (e.g. images and text). In this work, we seek to leverage freely available images on the web along with nearby text to create meaningful feature representations that capture both visual and sematic information. Our hypothesis is that these learnt features can then be used to improve on many different computer vision features on a wide variety of computer vision tasks.

1. Introduction

Multimodal learning involves relating information from disparate sources. For example, Wikipedia contains text, audio and images; YouTube contains audio, video and text; and Flickr contains images and text. To maximize performance on specific tasks, we would like to use all of the information available to us. It is not clear how to hand-engineer features that capture information from different modalities; thus, we adopt an unsupervised feature learning approach that learns joint correlations between different modalities-in particular, between images and text- using deep neural networks [2].

Specifically, we would like to leverage vast amounts of freely available data on the web - images along corresponding text captions that appears nearby- to create meaningful feature representations of multimodal data and ultimately improve on existing computer vision features.

In this particular project, we learn meaningful feature representations from images and tags crawled from Flickr. We've downloaded hundreds of thousands of images from Flickr along with their corresponding tags, and we wish to learn meaningful representations from that data that can be useful for other tasks not necessarily related to Flickr. The analogy here is, in loose terms, that we let out a robot "in the wild" and let it learn a meaningful representation of the world, and then we evaluate it on specific tasks of interest.

One approach consists in taking an off-the-shelf descriptor such as HOG and trying to predict the tags associated with an image using neural networks. This way, we learn a representation of images that captures both visual and semantic information. Another approach consists in using unsupervised feature learning to learn our own descriptors from image patches, and combine nearby descriptors to predict tags from the descriptors.

2. Dataset and Evaluation of Performance

We focus on evaluating the performance of our method in the SUN scene classification task[1]. Torralba *et al*. have put up together a large-database of images containing 397 scene categories to evaluate numerous state-of-the-art algorithms for scene recognition. Our hope is to improve on the best performing algorithm for the task (In this case, HOG 2x2 features).

They provide code to compute a number of different image descriptors, such as color histograms, GIST, SIFT and HOG. We have already been able to reproduce the results they report - that is, scene classification accuracies for different . In particular, we expect to improve on the descriptor that performs best in this task by incorporating freely available data to create more meaningful representations.

3. First Approach

Our framework can be summarized as follows:

1. We are given a computer vision feature (e.g. HOG) along with a computer vision task (e.g. scene classification).

2. We train a (one-layer) neural network that learns to predict the concatenation of the input with the Flickr tags from the given computer vision features applied to the corresponding Flickr images.

3. The hidden layer constitutes the learned joint features that captures image/text correlations.

4. We then forward propagate the network using as input the images from the dataset to obtain a new set of features for the specified task.

Our hypothesis is that the learnt features (possibly concatenated with the original features) will improve the performance of the original features . Since this approach does not take into account any task-specific knowledge, and the text data is available at no extra cost, we envision this approach to work with a number of computer vision features for a wide variety computer vision tasks.

3.1. Results

We evaluated the performance of our method on the SUN scene classification task, using $n = 5$ training examples per class. HOG is the feature that performs best in this task, so we've devoted most of our efforts into trying to improve the performance of HOG.

We've observed that most of the predictive power of the features such as HOG comes from the similarity histogram kernel. Unfortunately, since our learned features are sigmoid units in the 0-1 range, the similarity histogram kernel applied to our features does not yield particularly good results. We've experimented with many different kernels such as chi-squared and Gaussian kernels, but the best results that we've obtained are 5.32 % accuracy in the task, which are not very good compared to HOG's performance of 11.15 %, using $n = 5$ training examples per class in both cases. Concatenating the learnt features with the original features yields no improvement whatsoever (compared to the performance of the original features by themselves). This is mostly because our learn features don't perform very well by themselves.

It is worth noting that the our learned features are not bad by themselves. In fact, our learned features beat HOG if we use a linear kernel in both cases. The performance of HOG features using a linear kernel is of 4.25 %, whereas our learned features yield a performance of 4.98 % (again, using $n = 5$ training examples per class). Furthermore, using text information does better than using image information alone (4.98 vs. 4.45 % classification accuracy). The problem is that we don't know of a kernel that we can apply to our learned features to improve on HOG+similarity histogram. That being said, we're now considering a different approach to learn meaningful representations from vast amounts of unlabeled data.

4. Another Approach

We are currently experimenting with a different approach: build our own local descriptors. More specifically, our approach is as follows:

1. From an rgb image, extract a 16 by 16 patch.

2. Since pixels are highly redundant, we use PCA to reduce the dimensionality of the data.

3. We normalize the data so as to have zero mean and unit variance.

4. We learn an auto-encoder that learns to predict the preprocessed patch from itself (i.e. a neural network with one hidden layer in which the output is the same as the output. The hidden layer constitutes our learned features). The challenge is how to incorporate the text information into the descriptors. Given that the descriptors use only local information, we can't expect to predict specific tags such as "cat" from 16 by 16 patches, as most of the patches in an image containing a cat probably don't contain a cat and thus the statistics corresponding to those patches will be different from those patches associated with the particular tag. Our approach to incorporate text information is thus:

   (a) Use dense-sampling to extract features from many locations of an image.

   (b) Take a 2 by 2 window of descriptors (corresponding to 4 16x16 adjacent patches), and pool them together (using average or max pooling) to create a new descriptor representative of the whole window.

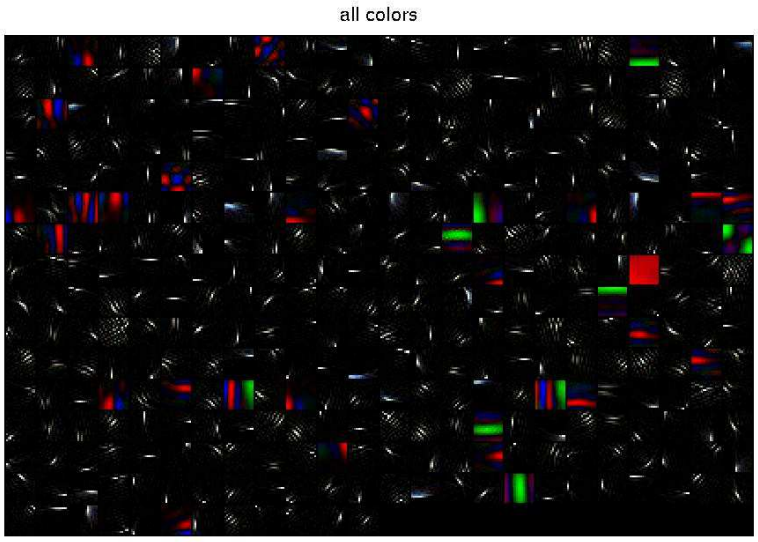
   (c) From the new descriptor corresponding to the 2 by 2 window, learn an autoencoder that learns to predict the concatenation of itself with the tags corresponding to the whole image.

   We believe that using a sufficiently large window, we will be able to extract descriptors that are related to the tags corresponding to the image, and that using such information will have a positive impact on. We then use the standard bag-of-words model: use dense-sampling to extract many descriptors from an image, and create a histogram of visual words corresponding to the image which is suitable for scene classification. We obtain the codewords by running k-means on patches extracted randomly from Flickr.

   It is worth noting that we are experimenting with different possibilities for the patch size, number

4.1. Preliminary Results

A visualization of the features that we've learned is as follows:



Each square is a 2D depiction of an image that would cause each hidden unit of the autoencoder to be maximally activated. We can see that different hidden units have learned to detect edges at different positions,colors and orientations of the image, while others are Gabor-like.

References

[1] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba SUN Database: Large-scale Scene Recognition from Abbey to Zoo *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[2] G. Hinton and R. Salakhutdinov *Science*, 313(5876):504–507, 2006.