Presented by Akshay Iyangar, 1208604211
Vijetha Gattupalli, 1208652064

Claas project on

DeViSE: A Deep Visual-Semantic Embedding Model

Neural Information Processing Systems, 2013

Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio
Jeffrey Dean, Marc'Aurelio Ranzato, Tomas Mikolov

ABSTRACT

In this paper, the authors propose a way of imposing knowledge obtained from textual data to general object classification task. The authors thus show that this way of training a classifier would result in a better model incorporating the semantic information obtained from the text corpus. This is different from the normal image classification where they just predict the class label. Here, the authors try to predict the word vector of the class label which has more degrees of freedom and therefore could accomodate more information. Regarding testing their belief, they perform the experiments on the imagenet dataset which contains 1000 classes. They further extend this approach to classifying new objects using knowledge obtained from text i.e. zero-shot learning. We, as a part of this class project implemented the same algorithm and experimented on two datasets, i. cifar-10 and ii. smaller version of the imagenet-2012 dataset. Our results are on the lower end of the results presents in this paper.

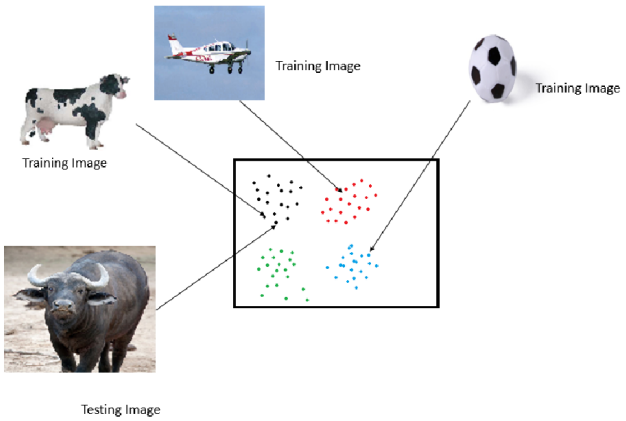

Figure 1: Classes which are semantically closer will have word vectors closer to each other than otherwise. Figure describing clasification of a previously unseen class's test image.

1 Introduction

Current object classification techniques try to build a model by discretizing the objects of the world and using these labels for training. To accomodate this approach, reseachers have built datasets which have images and corresponding labels given. On similar lines, imagenet dataset(ILSVRC-2012)[3] is built which have 1000 labels and 1.2 million images. However, the authors claim that such discretization of world may not be appropriate to build better object detection models as the real world itself is continuous. Also, labels learned by such models are all disconnected which may be of less use as it doesn't learn any intuitive meaning. The authors therefore try to embed the visual information and the much available text information to build a classifier which learns the meanings of labels instead of just trying to predict the image patterns.

Their idea is to embed the image labels on to a space such that semantically well-related labels are closer and the ones which are not are away. This approach is especially useful when the amount of training data is limited. In addition this paper addresses the challenge of predicting the label of an object never seen before, called zero-shot learning. The arguement for this is in similar lines to the one above where if a classifier can learn the meanings of the words, it should be able to predict a synonymically closer class. They got an accuracy of 18% over 20,000 image labels for zero-shot learning, which they claim to be 65% better than the then state of the art.

This report deals with the approach presented in this paper along with the experiments performed as a part of this project. It is organized as follows: Section 2 describes the higher level details about the approach along with some back ground about Convolutional Neural Networks and the text and language processing models. Section 3 talks learning involved in building DeViSE. Section 4 talks about experiments and results from our implementation. Section 5 mentions about the contribution of each member in the project and we finally conclude

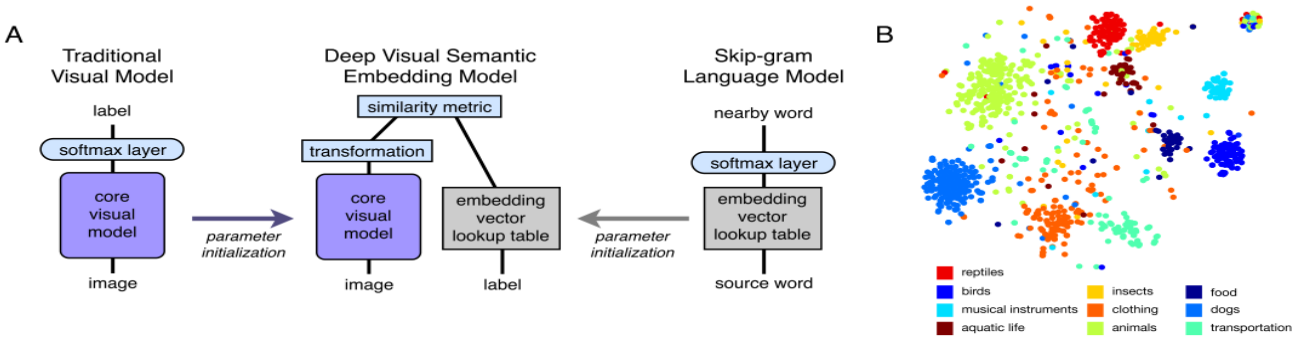

Figure 2: Figure describing the architecture of the devise model as given by [1]

with section 6.

2 Approach and Background

The authors begin by pretraining two models, one is a state of the art model to classify imagenet-2012 challenge and the other one to embed different words into their corresponding meaningful vector representations. The authors then combined these two models to get the deep visual semantic model which learns the labels in a semantically intuitive way. A brief back ground of CNNs and the text model are given below

2.1 Convolutional Neural Network

A convolutional Neural Network is a neural network where the weights are not fully connected. They hold two main properties which make them distinct from a fully connected network.

i. Local connectivity

ii. Weight sharing

It is generally the case that the pattern of a particular pixel in an image is more related to the pixels surrounding it rather than all of the pixels. So it is reasonable to connect each weight to only a local sub space of pixels rather than all of them. This reduces the number of weights to be learned by the network making it more practical to work on.

Furthermore, CNNs use the concept of shared weights which makes the network invariant to spatial location of a particular pattern. The kernel goes around the entire image to find patterns which look similar and updates the weights accoringly. There could be many such distint patterns and therefore, there are multiple featuremaps to learn each of these patterns.

The first pretrained model of this paper uses Alexnet [2] which uses five convolutional layers followed by three fully connected ones. Our experimental set up for this part is explained in experiments and results section.

2.2 Word2Vec Model

The word2vec model is a two layer neural network unlike the classical Neural Net Language Model (NNLM) which also consist of hidden layers. The word2vec model is used for efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. Both these models are log-linear classifiers.

In both these layers the hidden layers are removed as most of the complexity in Neural Net language Model was due to the fact that there were hidden layers. Thus in both the architecture the hidden layers are removed and the projection layer is shared for all the words in the corpus.

i. Continuous bag of words tries to determine the word given the context. All the words can be in random order in this architecture as the order of words in the context does not influence the projection. Thus in Continuous Bag of word we try to predict a word given the future and the history words at the input.

ii. Skip gram text model on the other hand tries to determine the context given the word. This is particularly useful when needed to classify a word into a corresponding category where it tries to maximise the classification based on the context in which the word occurred.

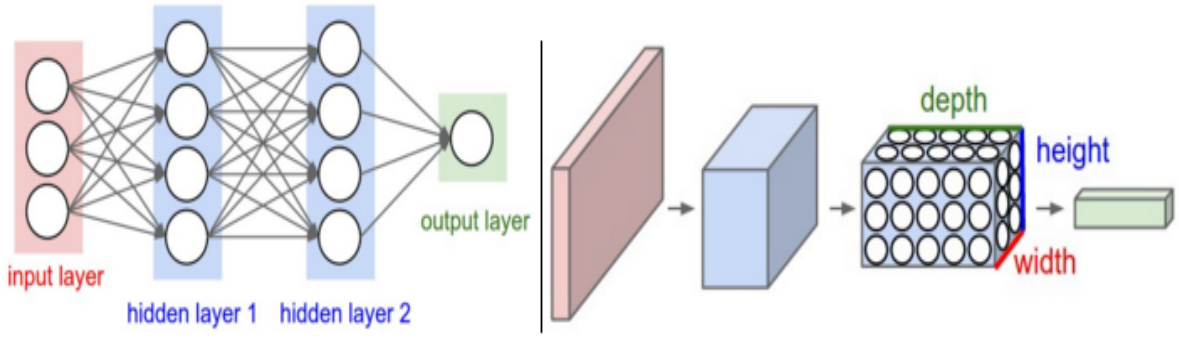

Figure 3: Diagram explaining the Convolutional Neural Network

The current paper uses Skip gram text model outputting a vector of 500 and 1000 words. The details of the same are given in the next two sections.

3 Learning DeViSE

To achieve the objective of learning sementially meaningful classification model, the authors first use two pretrained models each of visual and text. They then combine these two pretrained models to get the DeViSE model.

3.1 Visual Model Pre-training

The visual model is trained using the ILSVRC-2012 which contains 1,000 labels.. They used the well known Alexnet[2] for this purpose which has five convolutional layers optionally followed by max pooling layers and three fully connected layers. The network uses the local contrast normalization and dropout regularization techniques. The final layer is covered by a softmax layer to predict one of the 1,000 categories.

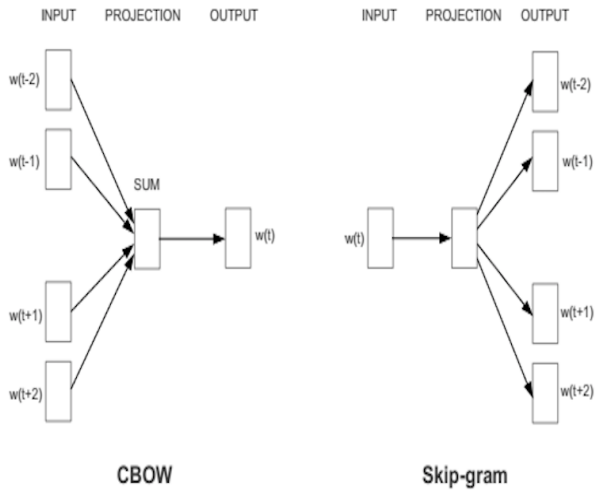

Figure 4: Diagram explaining the text models

3.2 Text Model Pre-training

A skip-gram text model described in the paper was trained on a corpus of 5.4 billion wikipedia.org words. The window length used was 20 and the length of the output vectors was 500 and 1000. The skip-garm model used a hierarchial softmax layer.The main idea behind using text to extract information is the semantic meaning the text data holds. For example, there could be different words (Synonymns) used in different sentences but having the same contexts. The word corpus would have several such sentences and thus a model trained on this would map synonymns closer. Thus a model using textual information for image classification could be of much use than just classification.

3.3 Deep Visual-Semantic Embedding Model

The authors combine the above two models to build the DeViSE algorithm which uses the below loss function. They extract the image feature vector from the Visual Model just before the softmax layer. They thus obtained a vector of length 4096 per-image. They used the text model to obtain the vector representations of the labels of length 500 and 1000. A transformation matrix is then learnt which would project the image vector into a word vector feature space. The authors use the above hinge loss function to train the transformation matrix.

4 Experiments and Results

The authors performed two sets of experiments, one is to get the state of the art accuracy on imagenet dataset using DeViSE and the second is to do zero-shot learning for 20,000 classes which have not been seen during training.Regarding our implementation and experiments, we performed the first experiment

$$loss(image, label) = \sum_{j \neq label} max[0, margin - \vec{t}_{label} M \vec{v}_{image} + \vec{t}_j M \vec{v}_{image}] \quad (1)$$

of the paper, i.e. trying to achieve the state of the art accuracy using DeViSE using two datasets. The experimental details are given below.

4.1 Datasets and preprocessing

We used two datasets namely, CIFAR 10 and a subset of Imagenet 2012 for our experiments. The first dataset is obtained from a library called Skdata [6] where there are classes written to download the dataset onto your machine and make them into batches. There are totally 50,000 samples in the datasets which contain both color and grey scale images. It contains images from 10 different classes like cow, aeroplane etc. The size of each image is 32*32*3 which we used as it is. We split the dataset into two batches of 40,000 and 10,000 for training and testing respectively.

The second dataset is a subset of Imagenet-2012 where we used 100 class data from imagenet where we picked 46,000 samples intotal. 35,000 of this is used for training and the remaining is used for testing. The images were of varying sizes and each of these is resized to 256*256*3. We downloaded the dataset from the imagenet challenge website [3] and did all the preprocessing manually. We handpicked 100 classes from the 1000 which had one-word labels explaining the classes decently. i.e we didn't pick a class called Egyptian cat because neither of the two words would explain the class properly. We did this because our language model could convert only words into vectors and not phrases. We therefore felt it would be best to consider classes which can be explained in a single word(Eg. peacock).

Regarding the language model, we used a wikipedia dump which contained 3 billion english words. Preprocessing is done by the word2vec script [4] where they convert the unorganized text into a trainable format. No manual preprocessing is performed for this part of the project.

4.2 Experiment-1

We tried to get the baseline accuracy(the procedure is described below) on CIFAR10 by using DeViSE. We first trained a Convolutional Neural Network using batch gradient descent. We used our own architecture of [5X5, 4X4, 3X3, 2X2, 2X2] filter sizes with features maps [32, 64, 64, 128, 128] at each layer. This network is followed by a fully connected layer. We got a baseline accuracy of 54.23%. This is lesser than the state-of-the art accuracy on this dataset which is expected as we could not implement many techniques like dropout, batch-normalization etc.We then extracted the feature vector from the last convolutional layer where we got a feature vector of 128 dimensions.

On the other hand we trained the word2vec text model obtained from [4] using the 3 billion words mentioned above. We trained it to get a final word vector of length 100 with a skip of 5. We then extracted the 100 dimensional word vector for each of our class labels.

We then built a MATLAB program to obtain the tranformation matrix of size 128X100. We used the exact same equation as mentioned in the above section to train our model. We ran the program through a total of 50 iterations of which the program reached it convergence point at around 10th iteration. We used stocastic gradient descent for training M. While testing, we first compute the predicted word vector of the test sample feature vector and compared how far it is from the each of the label vectors using cosine similarily. We then assign the class label which is most similar to the predicted vector. Following this approach, we got a test accuracy of 39.17%.

4.3 Experiment-2

We performed a similar experiment on a subset of Imagenet-2012 containing 100 classes. To establish the baseline accuracy, we used the pretrained overfeat *accurate* model [5] which has 24 layers. We used the python version of the model as it is, without doing any further training or finetuning. We got an accuracy of 57.88% for top-1 class prediction and and accuracy of 79.98% for the top-5 classes. We then extracted features from the 21st layer of the network to obtain an image feature vector of 4096 dimensions. We obtained the word vectors

Table 1: Results from our Implementation

DATASET	BASELINE	DeViSE
CIFAR10	54.23%	39.17%
ImageNet-subset		
Top-1	57.88%	35.2%
Top-5	79.98%	48.43%

for this experiment in a similar way like we did for the above one, except that we trained it for a word vector of length 500 instead of 100.

We used the same MATLAB program as above and the same stocastic gradient descent as above except that the size of the matrix now is 4096X500 instead of 128X100. The program ran for 50 iterations and used cosine similarity to get the predicted label. We obtained an accuracy of 35.2% for the top-1 class prediction and 48.43% on the top-5 class prediction.

5 Individual Contribution

5.1 Akshay

Akshay worked on getting the word vectors for the class labels for both the experiments, the overfeat code to get it running and performed the pretraining of visual model for imagenet subset. He used the standard word2vec C code from [4] and used the python library gensim to generate the word vectors. A wrapper python script is written on top of the overfeat model to obtain features from the 21st layer.

5.2 Vijetha

Vijetha worked on pretraining the CIFAR-10 model, handpicking the class labels based on a strategy mentioned above and training the DeVISE model for both the experiments. Code for CIFAR-10 pretraining is written using the python library theano and the code for training the DeViSE mode was written in matlab.

6 Conclusion

In conclusion, working on this project introduced us to the area of text and language models while we had some idea about the Convolutional Neural Networks previously. We now belive that text data contains a lot of semantic information which can be used in various vision tasks like image classification, image description etc. Also, as mentioned in the paper, it could be used for novel object detection which again could be used in various applications. The text information used in this paper thus adds a lot of value to many computer vision tasks compared to just using the patterns learnt from images.

References

[1] A. Frome, GS Corrado, and J. Shlens, DeViSE: A deep visual-semantic embedding model, NIPS, 2013.

[2] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012.

[3] http://www.image-net.org/

[4] https://code.google.com/p/word2vec/

[5] http://cilvr.nyu.edu/doku.php?id=software:overfeat:start

[6] https://github.com/jaberg/skdata