ATTRIBUTE VS CLASSIFICATION ACCURACY IN ZERO-SHOT LEARNING.

IE 590: DEEP LEARNING IN MACHINE VISION

Akash Ramesh Reddy
School of Industrial Engineering
Purdue University
reddy72@purdue.edu

Gokulakrishnan Swaminathan
School of Industrial Engineering
Purdue University
gswamina@purdue.edu

Mayur Kishor Deo
School of Industrial Engineering
Purdue University
deo1@purdue.edu

December 13, 2019

ABSTRACT

Zero-shot learning involves the recognition of new, unseen classes by utilising the features/attributes learned during training with the additional information provided about the unseen classes which relates them to the seen classes. This work proposes a Zero-shot learning(ZSL) approach that achieves a new accuracy in detection on unseen images using various deep learning architectures. This approach utilizes images, attributes and information provided on seen classes. The distinction between attribute accuracy and ZSL accuracy is explored for three deep-learning (specifically CNN) architectures: Inception v3, ResNet-50 and Inception-ResNet-v2. This work also provides an estimate of the number of images various deep-learning architectures require to achieve the maximum ZSL accuracy.

***K*eywords** attribute · zero-shot · accuracy · number of images

1 Introduction and Related Work

Many deep learning methods focus on classifying instances whose classes have already been seen during training. However, many applications in real-world situations require classifying instances whose classes have not been seen before. One such scenario is when we deal with an ever growing set of classes, such as detecting new species of plants/animals. Also, collecting large amounts of data for each class to train a model is very tedious and sometimes impossible due to the lack of data for certain classes. To solve these problems, there is an increasing interest in the study of Zero-shot learning (ZSL). Zero-shot learning is a promising learning method, wherein the classes seen by the model during training and the classes we aim to classify are disjoint. ZSL consists in recognising new categories of instances without training examples, by providing a high-level description of the new categories that relate them to categories previously learned by the machine. This can be done by means of leveraging an intermediate level: the attributes that provide semantic information about the categories to classify.ZSL refers to specific case of deep learning where the model classifies data based on non-labelled data. ZSL is about leveraging deep learning networks already trained by superficial learning in other ways, without additional supervised learning.[1]

This approach to image classification is inspired from the way humans are able to identify new objects (never seen before) by just having previous knowledge about the features, description of it, by drawing similarities between the previously learned concepts and description of the new object. In the exact method, ZSL approaches are developed to understand this intermediate layer: attributes/features and apply these learned attributes at the time of testing to predict new classes. ZSL relies on the availability of labelled training set of classes(seen) and knowledge about how each unseen class is semantically related to seen classes. Seen and unseen classes are related in a high-dimensional vector space, called the semantic embedding space (semantic attribute space or word-vector space). In this space, names of both seen and unseen classes are embedded as vectors called class prototypes. Semantic relationships between classes are then measured by a distance. ZSL methods learn this projection function from a visual feature space to a semantic

Figure 1: Zero-shot Learning

embedding space based only on seen training classes. At test time for recognizing unseen classes, this projection function is used to project the visual representation of unseen class image into the same semantic space where both seen and unseen classes reside. The role of unseen class classification is reduced to nearest neighbour search, class label of test image is assigned to nearest unseen class prototype in the projected semantic space.Although the training and testing classes are different they can be considered as two overlapping domains with some degrees of shared semantics but there exists significant domain differences. The existing ZSL algorithms mostly suffer from domain shift problem, which is if the projection for visual feature embedding is learned only from seen classes, projections of unseen class images are likely to be misplaced due to bias of training seen classes, which makes the nearest neighbour search imprecise. [2].

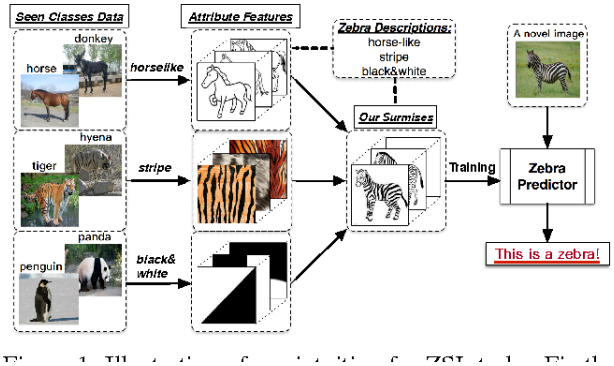


Figure 2: Zero-shot Learning

In this work, we present an approach to Zero-shot Learning from encoder-decoder model architecture. The encoder projects a visual feature representation of an image into a semantic representation space (attributes space) . Here, we also consider the input to the decoder to be the visual feature projection which aims to reconstruct the original visual feature representation. [3]This additional constraint is very effective in improving the accuracy of Zero-shot learning. In this work, we are also estimating the attribute prediction by measuring its accuracy, contrasting it with the ZSL classification accuracy for the different deep learning models/architectures.[4] Existing ZSL models differ in the way visual space to semantic space projection function is established. Our architecture is a combination of two groups of ZSL models with the additional reconstruction condition.

2 Methodology

For the purpose of Zero Shot Learning we used the Semantic Auto-encoder approach. This approach mainly involves training on the seen classes to obtain the involved weights. Sylvester Equation was used for updating the weights. The obtained weights are used at test time for detecting the unseen classes. The input to the the Semantic Auto-encoder is a feature vector that is extracted using deep convolutional neural networks. The attribute space was used as semantic space for the Semantic Auto-encoder.[5] A major premise to the project is testing the sensitivity of the evaluation metrics to the use of different pre-trained convolutional neural networks. The evaluation metric used is the Zero Shot Learning accuracy and the Attribute accuracy.[6]

Sylvester Equation:

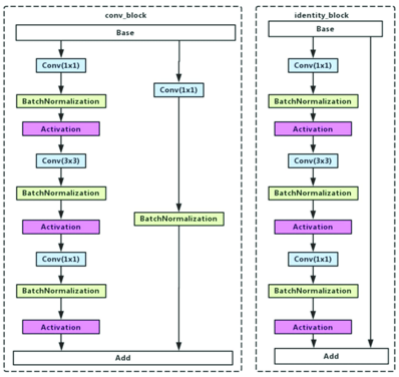
$$AW + BW = C \tag{1}$$

2.1 ABC Network Architecture

The deep convolutional neural networks used are Resnet-50, Inception-v3, and InceptionResnet-v2. The pre-trained version of the mentioned networks were used.

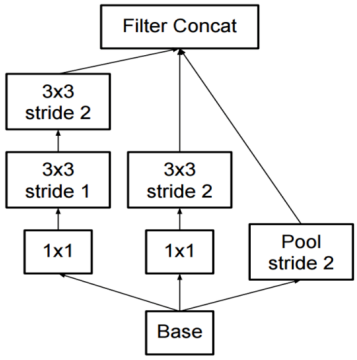
ResNet-50 is a convolutional neural network that is trained on more than a million images from the ImageNet database. It uses residual mapping instead of direct mapping for better optimization. The network is 50 layers deep and can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

Figure 3: ResNet-50 architecture



Inception-v3 is a convolutional neural network that is trained on same database as ResNet-50. This uses parallel filter operation on the input from the previous layer, then concatenates the output. The network is 48 layers deep and has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299.

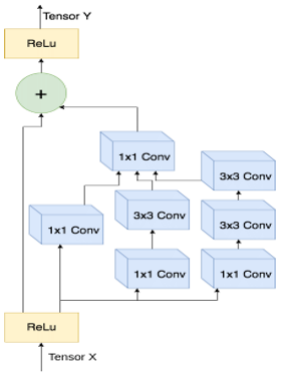
Figure 4: Inception-v3 architecture



InceptionResnet-v2 uses the inception network with residual connections. The network is 164 layers deep and can classify images into 1000 object categories. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299.

The mentioned architectures were used to parse the input image and output the feature vector. The encoder then projects this vector into the semantic space and the decoder then projects it back to the original feature space and aims to faithfully recreate the input vector. The weights are then altered on each training step to minimize the losses generated during encoding and decoding. The obtained weights are further used for zero shot learning. The method used for zero shot class detection was nearest neighbor with cosine similarity. This was used as the base for the performed experiments.

Figure 5: Inception-ResNet-v2 architecture



2.2 Dataset

The dataset we are using is Animals with Attributes (AwA). This dataset provides a platform to benchmark transfer-learning algorithms. The dataset consists of 30,475 images with 50 animal classes and six feature representations for every image. The animals' classes are aligned with Osherson's classical class/attribute matrix, thereby providing 85 numeric attribute values for each class called the predicate matrix (matrix described as a black-white diagram where the attributes present in the class is represented by white blocks and attribute absent in the class is represented by black blocks). Of the 50 classes, 10 classes are randomly chosen to be ZSL (testing) classes and the rest 40 classes are used for training the models.

Figure 6: Predicate Matrix

Table 1: Dataset

Dataset	instances	instances/class	SS_D	seen/unseen
AwA	30,475	650	85	40/10

3 Experiments and Results

Semantic spaces
We use attributes as the semantic space for the AWA datasets, the involved attribute annotations are provided.
Features

All recent ZSL methods use visual features extracted by deep convolutional neural networks (CNNs). In our experiments, we use Resnet-50, Inception-v3, and InceptionResnet-v2 for obtaining feature vector which is the 85D activation in the final pooling layer. The used architectures have been pretrained on the images from the ImageNet database.

Parameter settings

Our SAE model has only one free parameter: (weighting coefficient). It's values are set by class-wise cross-validation using the training data. The dimension of the embedding (middle) layer always equals that of the semantic space.

Evaluation metric:

We use the Zero Shot Learning accuracy as one of the metric, which pertains to direct identification of the correct class. Attribute accuracy is also used as a metric, which is basically the correctness of the predicted attribute vector. The aim of the experiment is to test the sensitivity of the base model in terms of the evaluation metrics to the changes in the combinations of the used architecture and number of images.

4 Discussion and Conclusions

Figure 7: ZSL accuracy Vs Number of Images

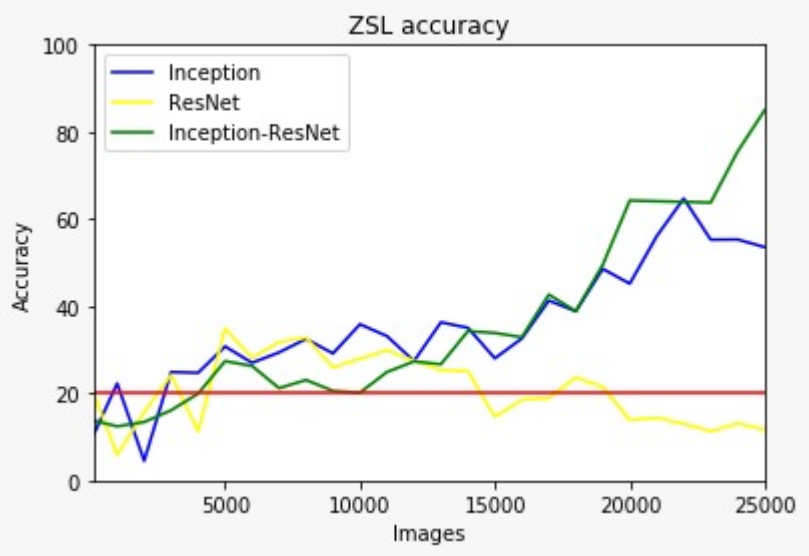
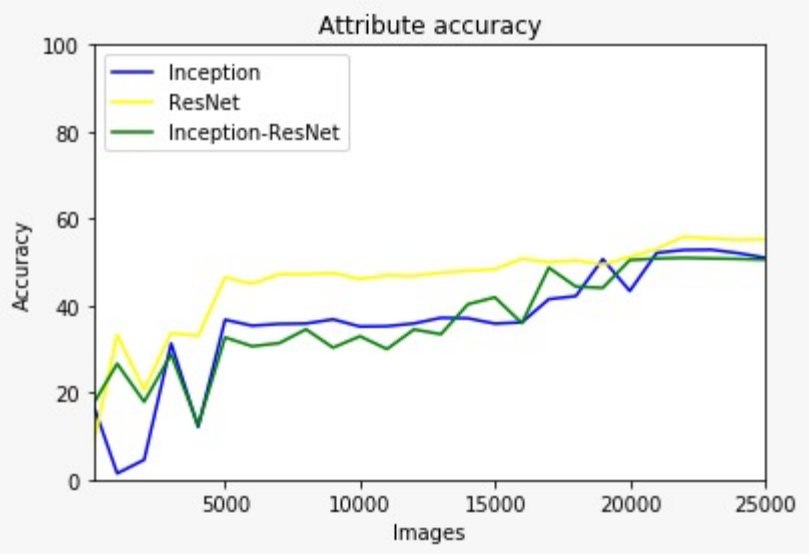


Figure 8: Attribute accuracy Vs Number of Images



The primary objective of the project is to achieve the same equilibrium that is present in the supervised learning models i.e., attribute accuracy is directly proportional to classification accuracy.

By performing the experiments using Adam optimizer with a learning rate of 1e-3 and loss for both attribute and classification as binary cross-entropy the below results are obtained.

We could achieve the zsl accuracy 82 percent as shown in the paper using Inception-V1 because the model used here are very deep convoluted networks and the difference between the layers are different from Inception-V3 to chosen models. The input layer for SAE zsl from the paper was [batch-size,1024] whereas in the models used is [batch-size,2048] for inception based models and resnet [batch-size,1536]. All these layers are obtained from the pre-trained model using tensorflow backend function.

From the graphs of attribute and classifications we can conclude the following results:

1. If we have lower images in the range of 5000-10000 images we can use the resnet because the accuracy in both attribute and the classification are higher.
2. If we have more images (10000-15000) we can use either inception or inception-resnet model
3. If we have more than 15000 images inception-resnet model should be used for zero shot learning

We can infer these following observations regarding model performance in various training conditions.

1. From the attribute accuracy graph, we can observe that irrespective of the models the attribute accuracy monotonically increases with increase in the images.
2. From the classification accuracy graph, we can observe that the resnet models after 10000 starts losing its accuracy. This might be due to the fact that in resnet model the update is so frequent the weights generated are similar to random weights and thus performs poorly. Whereas Inception based models are deeply convoluted and thus retains the changes in weights and produces a much better accuracy with increase in images.
3. The reason for obatined performance is because the model is trained on same learning rate, different batch sizes, etc. If we could tune the hyperparameters of the models, then maybe the models can perform in the optimum rate

5 Contribution of Team Members

1. Team Member 1 - Mayur Kishor Deo
   Contributed equally in the project. Dataset cleaning and ran Inception model, observed and reported the results
2. Team Member 2 - Akash Ramesh Reddy
   Contributed equally in the project. Built all the reports needed throughout the project, ran ResNet model, observed and reported the results
3. Team Member 3 - Gokulakrishnan Swaminathan
   Contributed equally in the project. Built all 3 models,Ran Inception-ResnNet model, observed the performance and reported the results

References

[1] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015.

[2] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3174–3183, 2017.

[3] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

[4] Daniel N Osherson, Joshua Stern, Ormond Wilkie, Michael Stob, and Edward E Smith. Default probability. *Cognitive Science*, 15(2):251–269, 1991.

[5] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[6] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.