Semantic Adversarial Autoencoder for Zero-Shot Learning

Shihui Li, Yu-Hsiang Lin, Kangyan Zhou
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{shihuil,yuhsianl,kangyanz}@andrew.cmu.edu

Abstract

In zero-shot learning tasks, how to classify the data in the unseen classes in one domain by leveraging the information from the semantic domain is a main challenge. We propose a novel model, the semantic adversarial autoencoder, to address this challenge. By injecting the global information of the distribution of the semantic representation, the model acquires better generalization ability. We perform experiments on the AwA and CUB datasets, and show that our model outperforms the semantic autoencoder in the generalized zero-shot learning tasks.

1 Introduction

Zero-shot learning (ZSL) has been an active research topic in the field of image classification. It simulates how people learn things: when people see an unfamiliar object, they use existing knowledge and try to evaluate the new object. Ideally, a large-scale image classification system should be able to recognize novel categories based on its previous training experience. One of the main challenges for object recognition is the lack of sufficient annotations for all possible concepts. This problem becomes even more severe when we target at the task of fine-grained classification because the annotation is more expensive and the number of fine-grained classes is huge. Realizing this limitation, researchers resort to additional information, for example, textual or attributive description, to solve this problem.

On the other hand, autoencoder has shown its great express power to present complicate distributions such human faces, natural sceneries, and natural language. It is able to convert complicated real world data distributions, e.g. text features, to very well-formed low dimension. Many studies have shown that after applying autoencoder techniques, the learned latent feature space often has clear semantic meanings. For example, in [13], all the digits can be well embedded into a low-dimensional manifold such that similar digits will have smaller distance within that manifold.

The fundamental problem of ZSL is to extract semantically meaningful feature embeddings that could bridge the gap between similar image/text features and fine-grained classes. Autoencoder becomes an ideal option for this task because it can automatically learn representative features. In [8], the authors present a novel solution to zero-shot learning, the Semantic Autoencoder (SAE). Taking the encoder-decoder paradigm, an encoder aims to project a visual feature vector into the semantic space as in the existing zero-learning shot models, and the decoder should be able to reconstruct the original visual feature. Their results show that under this framework, the learned projection function from the seen classes is able to generalize better to the new unseen classes. One shortcoming of this model is that the encoder tends to learn similar features across multiple classes, which is disappointing for the task of fine-grained classification.

This work is done to fulfill 10-707 course project.

Inspired by these models, we propose the model of semantic adversarial autoencoder (SAAE)[1]. In this model, we have an encoder that projects a visual feature vector into the semantic space and a decoder that decodes and reconstructs the original visual features. Besides, we incorporates a discriminator as in [13] to ensure that generating from any part of prior space results in meaningful samples. Two types of models that use different priors are investigated. In one type of the model, the prior is the Gaussian noise prior, and in the other type the semantic representation itself is used as the prior.

2 Related Works

Considering that fine-grained categories might share some common knowledge, a common approach is to seek for an intermediate semantic representation that could connect seen and unseen classes. Human specified attributes are first explored to represent the discriminative properties shared among both seen and unseen categories in zero-shot learning [15, 9]. One limitation of this method is that the creation of attributes still relies on human labors, making it difficult to scale up to meet large scale needs.

Apart from handcrafting attributes, another scheme is to directly use the online textual document as the additional information source. [3] is one of the first works to use Wiki documents as text attributes. [11] proposed a model that changes both the ways of extracting features from images and text domains. More specifically, the image features are extracted from the activation layer of a convolutional neural network (CNN), and then go through a linear projection layer to reduce dimensions. The input text, e.g. Wikipedia articles, is first converted to one-hot encodings of the words with their tf-idf scores, which can be viewed as attributes, and then fed into a multi-layer perceptron (MLP) to generate a deep representation, with the same dimension as the final image feature, and unique for each class. The final prediction is obtained by the dot product of the two generated features. Instead of learning an embedding space for each modalities, [4] learns joint image-word embeddings so as to embedding images and sentences into a common space.

Another line of research is to improve the quality of the classification procedure. [6] firstly proposed a SVM based classifier which takes the linear projection of both source and target domain data as the combined input. More recent works jointly project the class into an embedding space, and try to compute a compatability function $F(x, y)$ that tries to predict whether the image feature x is compatible with the embedded class feature y. Here each class is represented as a vector that contains the relevance scores of the class and a set of predefined attributes. In [1], $F(x, y)$ takes a linear form as $F(x, y) = xWy$. [17] takes this idea one step further, where a set of W_i is available for the the compatibility function $F(x, y)$, and the final prediction will choose the W that can produce the highest score. This W_i can be shared across different labels. The method is called as latent embedding, since it learns a latent embedding space explicitly based on clustering. [19] proposes a framework that generalizes deep learning embedding, label embedding, and latent embedding.

Recently deep encoder-decoder has become popular for a variety of multi-modal problems. In [7], they introduce an encoder-decoder pipeline that learns a multimodal joint embedding space with images and text and a novel language model for decoding distributed representations of the text semantic space. Their pipeline effectively unifies joint image-text embedding models with multimodal neural language models. [8] takes a step further in multimodal model under the autoencoder paradigm. They proposed a novel zero-shot learning model based on a semantic autoencoder that uses a fast linear projection function and introduce an additional reconstruction objective function for learning a more generalisable projection function.

3 Semantic Adversarial Autoencoder

3.1 Semantic Autoencoder

For transfer learning tasks, the semantic autoencoder (SAE) [8] is a tailor-made type of antoencoder-like structure that learns the transfer function between two domains. We consider the case in which one

[1]The code is available on `https://github.com/yuhsianglin/10707DL_proj`.

domain is the image feature $\boldsymbol{x}$, and the other domain is the attribute $\boldsymbol{t}$ (the "semantic representation") that describes the corresponding image. An autoencoder is trained to learn the transfer function W through the optimization problem,

$$\min_{W} ||\boldsymbol{x} - W^\top W\boldsymbol{x}||_2^2, \quad \text{s.t. } W\boldsymbol{x} = \boldsymbol{t}. \tag{1}$$

This model enforces the encoding of the image to be identical with the corresponding semantic representation. While this is a difficult constrained optimization problem, in the SAE architecture, the loss is relaxed to

$$L_{\text{SAE}} = ||\boldsymbol{x} - W^\top \boldsymbol{s}||_2^2 + \lambda||W\boldsymbol{x} - \boldsymbol{s}||_2^2, \tag{2}$$

where a coefficient λ is introduced to adjust the relative importance of the two terms. The SAE loss is a convex function, and in this form W can be solved efficiently via the Bartels-Stewart algorithm [12], which is adopted in [8].

Note that the SAE is more like a domain transfer machine, rather than a standard autoencoder: It is not minimizing the reconstruction loss; it is minimizing the two directions of transfers between the two domains. Later in our experiments we find that in practice the learning may be majorly driven by one of the two directions (see section 6.3).

3.2 Adversarial Autoencoder

In the adversarial autoencoder [13], a generative adversarial net (GAN) [5] is incorporated into the autoencoder, and the prior in the GAN is chosen to regularize the hidden representation generated by the encoder. The positive samples $\boldsymbol{z}$ are drawn from the prior $r(\boldsymbol{z})$, while the generator G, which is simply the encoder in this architecture, generates negative samples $G(\boldsymbol{x})$ by encoding the input $\boldsymbol{x}$, which is drawn from the underlying data distribution $p_d(\boldsymbol{x})$. During training, the discriminator D is trained to tell the positive samples from the negative samples, and the generator is trained to generate samples that has the aggregated distribution mimicking the prior distribution, so as to fool the discriminator. This procedure can be described as the optimization problem,

$$\min_{G}\max_{D} \; E_{\boldsymbol{z}\sim r(\boldsymbol{z})}[\log D(\boldsymbol{z})] + E_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log(1 - D(G(\boldsymbol{x}))) + L_{\text{recon}}], \tag{3}$$

where L_{recon} is the reconstruction loss of the autoencoder. Through this procedure, the model learns to generate the encodings whose distribution is close to that of the prior by jointly minimizing the reconstruction loss from the autoencoder, the loss of misclassifying the positive and negative samples from the discriminator, and the loss of failing to generate positive samples from the generator.

3.3 Semantic Adversarial Autoencoder

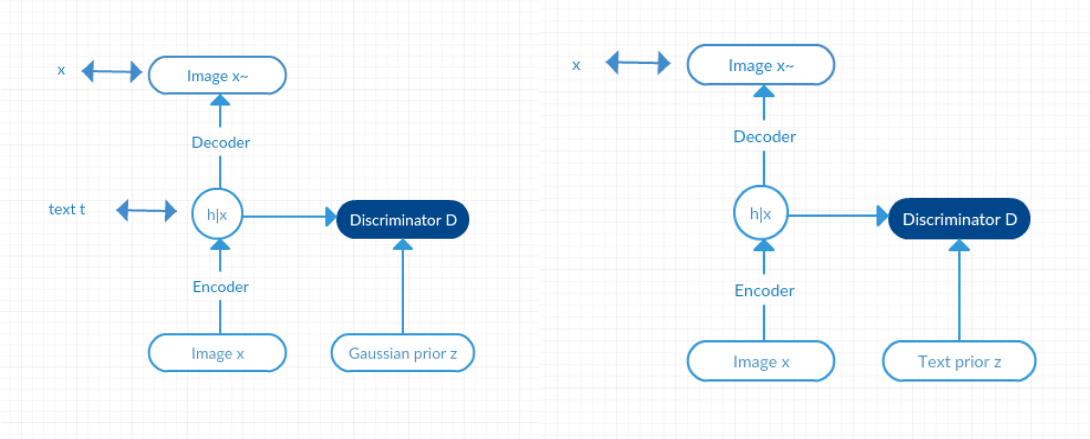
We propose the semantic adversarial autoencoder (SAAE) of which the generator learns to transfer the representation in one domain into that in another domain. The SAAE differs from the plain adversarial autoencoder in that it needs to encourage the input in one domain to be encoded into the representation in another domain (it learns a specific semantic). It also differs from SAE in that it uses the adversarial net to inject the global information of the distribution of the training data into the process of locally learning the encoding of each mini-batch of instances.

We perform experiments on two architectures: the SAAE with explicit matching (SAAE-exp) and the SAAE with implicit matching (SAAE-imp). Their architectures are shown in Figure 1. In SAAE-exp, we explicitly require the encoding to approximate the semantic representation, and the positive samples is drawn from the Gaussian prior. In this case, the prior serves as a regularizer. In SAAE-imp, we directly use the semantic representation as the positive samples drawn from some underlying distribution that describes the semantic representation, and the encoder is trained by the adversarial net to match the semantic representation. In this case, the prior guides the generator (the encoder) to learn the encoding that is close to the semantic representation.

3.3.1 SAAE-exp

SAAE-exp has an autoencoder that learns the encoding $\boldsymbol{h}^c$ of the input images $\boldsymbol{x}^c$ belonging to a class c. The encoding $\boldsymbol{h}^c$ is explicitly required to match the text (or, semantic/attribute)[2] representation $\boldsymbol{t}^c$

[2]We interchangeably use "text", "semantic", and "attribute" in this report.



(a) (b)

Figure 1: Semantic adversarial autoencoders with (a) Gaussian prior and explicit matching to the text (or semantic/attribute) representation, and (b) text (semantic/attribute) representations as positive samples drawn from some underlying prior of the text (semantic/attribute) distribution.

of that class, as well as regularized by the Gaussian prior in the adversarial net. The learning task can be expressed as the optimization problem,

$$\min_G \max_D E_{\boldsymbol{z}\sim r(\boldsymbol{z})}\left[\log D(\boldsymbol{z})\right] + E_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}\left[\log(1-D(G(\boldsymbol{x}))) + ||G(\boldsymbol{x})-\boldsymbol{t}||_2^2\right]. \tag{4}$$

Given an input image representation $\boldsymbol{x} \in R^{d_x}$, the encoder encodes it into $\boldsymbol{h} \in R^{d_h}$ by

$$\boldsymbol{h} = \tanh(W^e\boldsymbol{x}), \tag{5}$$

where $W^e \in R^{d_h \times d_x}$, and the hyperbolic tangent function is applied element-wisely. Note that the hidden representation space R^{d_h} is also the space of the semantic representation; that is, $\boldsymbol{t} \in R^{d_h}$. The decoder computes

$$\tilde{\boldsymbol{x}} = \tanh((W^e)^\top \boldsymbol{h}), \tag{6}$$

where $W^d \in R^{d_x \times d_h}$.

The generator of the adversarial net is the encoder of the autoencoder; that is,

$$G(\boldsymbol{x}) = \tanh(W^e\boldsymbol{x}) = \boldsymbol{h}. \tag{7}$$

The positive samples are drawn from the Gaussian prior,

$$\boldsymbol{z} \sim \mathcal{N}(\mu, \sigma^2), \tag{8}$$

where $\boldsymbol{z} \in R^{d_h}$, and μ and σ are chosen to be the mean and standard deviation of the semantic representation.

We use a fully connected neural network with a single hidden layer as the discriminator. Given an input $\boldsymbol{z} \in R^{d_h}$, it computes

$$\boldsymbol{z}^1 = \text{sigmoid}(W^1\boldsymbol{z} + \boldsymbol{b}^1), \tag{9}$$

$$z^2 = \text{sigmoid}((\boldsymbol{w}^2)^\top \boldsymbol{z}^1 + b^2), \tag{10}$$

where $\boldsymbol{z}^1 \in R^{d_1}$, $W^1 \in R^{d_1 \times d_h}$, $\boldsymbol{b}^1 \in R^{d_1}$, $\boldsymbol{w}^2 \in R^{d_1}$, and b^2 and z^2 are scalars. The discriminator returns

$$D(\boldsymbol{z}) = z^2, \tag{11}$$

which is the estimation of the probability that $\boldsymbol{z}$ is drawn from the prior.

For the autoencoder and the generator of the adversarial net, the loss function for a mini-batch of instances, S, is the empirical risk,

$$f_g = \frac{1}{|S|} \sum_{i \in S} \left\{ \log \left[1 - D(\boldsymbol{h}^{(i)}) \right] + ||\boldsymbol{x}^{(i)} - \tilde{\boldsymbol{x}}^{(i)}||_2^2 + ||\boldsymbol{h}^{(i)} - \boldsymbol{t}^{c_i}||_2^2 \right\}, \tag{12}$$

where the log function is applied element-wisely, and c_i is the class of the instance $\boldsymbol{x}^{(i)}$. Here we assume that there is only one text $\boldsymbol{t}^c$ for each class c.

For the discriminator of the adversarial net, the loss function is

$$f_d = -\frac{1}{|S|} \sum_{j=1}^{|S|} \log D(\boldsymbol{z}^{(j)}) - \frac{1}{|S|} \sum_{i \in S} \log \left[1 - D(\boldsymbol{h}^{(i)}) \right], \tag{13}$$

where $\boldsymbol{z}^{(j)}$ are $|S|$ samples drawn from the prior distribution $\mathcal{N}(\mu, \sigma^2)$.

The optimization problem for the autoencoder and the generator is

$$\min_{W^e} \ f_g, \tag{14}$$

while the parameters of the discriminator $(W^1, \boldsymbol{b}^1, \boldsymbol{w}^2, b^2)$ are held constant. The optimization problem for the discriminator is

$$\min_{W^1, \boldsymbol{b}^1, \boldsymbol{w}^2, b^2} \ f_d, \tag{15}$$

while the parameters of the generator (W^e) are held constant.

3.3.2 SAAE-imp

SAAE-imp takes the text representation $\boldsymbol{t}^c$ as the positive sample drawn from some underlying prior dictating the distribution of the text representation of a class. In this architecture, the encoding is driven to match the text representation through the adversarial net itself. The encoder, decoder, generator, and discriminator are the same as described in section 3.3.1. The difference is that we use the text representation, instead of Gaussian noise, as the prior in the adversarial net. For an input image $\boldsymbol{x}^{(i)}$ of class c_i and the text representation $\boldsymbol{t}^{c_i}$ of this class, we use $\boldsymbol{t}^{c_i}$ as the sample drawn from some underlying prior for the representation distribution of this class,

$$\boldsymbol{z}^{c_i} = \boldsymbol{t}^{c_i}. \tag{16}$$

In addition, we remove the term measuring the L2 distance between the text representation and the hidden representation of the image from the loss function.

The loss function for the autoencoder and the generator is now the standard one,

$$f_g = \frac{1}{|S|} \sum_{i \in S} \left\{ \log \left[1 - D(\boldsymbol{h}^{(i)}) \right] + ||\boldsymbol{x}^{(i)} - \tilde{\boldsymbol{x}}^{(i)}||_2^2 \right\}. \tag{17}$$

For the discriminator of the adversarial net, the loss function is

$$f_d = -\frac{1}{|S|} \sum_{i \in S} \log D(\boldsymbol{t}^{c_i}) - \frac{1}{|S|} \sum_{i \in S} \log \left[1 - D(\boldsymbol{h}^{(i)}) \right]. \tag{18}$$

The optimization and test procedures are the same as those described in section 3.3.1.

3.4 SAE-GAN

We also check whether adding GAN to the original architecture of SAE will change the performance of SAE. We solve the following optimization problem,

$$\min_G \max_D E_{\boldsymbol{z} \sim r(\boldsymbol{z})} \left[\lambda_{\text{GAN}} \log D(\boldsymbol{z}) \right] + E_{\boldsymbol{x} \sim p_d(\boldsymbol{x})} \left[\lambda_{\text{GAN}} \log(1 - D(G_{\text{SAE}}(\boldsymbol{x}))) + \lambda_{\text{SAE}} L_{\text{SAE}} \right], \tag{19}$$

where

$$G_{\text{SAE}}(\boldsymbol{x}) = W\boldsymbol{x}, \tag{20}$$

and λ_{GAN} and λ_{SAE} are introduced to specify the relative importance of the GAN and SAE parts in the optimization objective.

3.5 Classification task

There are two scenarios for ZSL, namely standard ZSL and generalized ZSL. For the standard ZSL, During training time, given training images $\boldsymbol{x}^c$ and their corresponding semantic representations $\boldsymbol{t}^c$ for N_s seen classes c, the model learns to bridge the gap between x^c and t^c. During testing time, we are given images $\boldsymbol{x}^{uc}$ from N_{us} unseen classes and semantic representations $\boldsymbol{t}^{uc}$ for those unseen classes without knowing the correspondence between $\boldsymbol{x}^{uc}$ and $\boldsymbol{t}^{uc}$. The task is to determine which unseen class each image belongs to. For the generalized ZSL, the training procedure is the same, but the test images are not constrained to unseen classes: images from both seen and unseen classes can be used as test images and semantic representations for both classes are provided.

During the test time, we follow the same classification method as in [8]. The input test image $\boldsymbol{x}$ is first projected to the semantic representation $\boldsymbol{h}$, and the class is predicted by

$$\hat{c} = \arg\min_c \text{dist}(\boldsymbol{t}^c, \boldsymbol{h}), \tag{21}$$

where dist is the distance function. The distance function can be the negative cosine similarity, $\frac{(\boldsymbol{t}^c)^\top \boldsymbol{h}}{||\boldsymbol{t}^c||_2||\boldsymbol{h}||_2}$, or the Euclidean distance, $||\boldsymbol{t}^c - \boldsymbol{h}||_2$. We find that cosine similarity in general gives higher classification accuracy than the Euclidean distance does. We use the former in our experiments.

4 Datasets

We conduct our experiments on two datasets. The first one is the Caltech-UCSD Birds 200-2011 dataset [16], with 200 categories of bird images. The total number of images is 11,788, and each class consists of about 40 to 80 images. The second dataset is AwA [10], which consists of 30,475 images of 50 classes of animals.

Following the preprocessing steps taken in [8], for CUB dataset, we extract the 1024D activation from last pooling layer of Inception-v1 [14] as our image features. For AwA dataset, we use the extracted 1024D features provided by [8]. We use the attribute vector of each class as the semantic representation of the class. The dimensions of the attribute vectors are 312 and 85 for the CUB and AwA datasets, respectively.

For the standard ZSL tasks, following the description of [8], we withhold 10 and 50 classes as the unseen classes for AwA and CUB datasets, respectively. For the generalized ZSL tasks on the CUB dataset, we follow the description of [18][3], using 7,125 training images from the 150 seen classes, and 4,663 test images from both 150 seen and 50 unseen classes. On the AwA dataset, we use 19,094 training images from the 40 seen classes, and 11,381 test images from both 40 seen and 10 unseen classes.

When we run our experiments, the image features and the attribute vectors are both normalized to $[0, 1]$ over the entire dataset.

5 Experiments

We compare the results of our models with the Semantic Autoencoder (SAE) [8], the Joint Latent Semantic Embedding (JLSE) [19], and the Synthesized classifiers (Sync) [2]. Due to the lack of an identical dataset, we decide to re-implement SAE to produce a reasonable baseline. It is worth noting that our implementation of SAE differs from that of [8] in that we solve the optimization problem by gradient descent[4]in its original form, while in [8] the problem is solved by first transforming it into a simpler linear equation. We find that the classification accuracy of the standard ZSL tasks using our implementation of SAE on both CUB and AwA datasets is lower than that reported in [8] (see Table 1). Although we cannot reproduce the results in [8], we implement our models in the same framework which we use for implementing SAE, in the following discussion, we will therefore focus on comparing our results with our own implementation of SAE for self consistency. Table 1 and 2 show the experimental results as well as the accuracy reported in [8].

[3]Since we do not have the information about the details of the splitting but only the number of instances in each split, we only follow roughly there number of instances used in each split, but not the exactly same split.

[4]We use the AdaGrad optimizer of TensorFlow.

	CUB	AwA
SAAE-exp	10.3	64.6
SAAE-imp	2.0	14.4
SAE-dir	**60.2**	77.0
SAE-GAN	5.7	**78.5**
SAE [8]	61.4	84.7
JLSE [19]	41.8	80.5
Sync [2]	54.4	72.9

Table 1: Top-1 per-class accuracy (%) of ZSL. SAAE-exp and SAAE-imp are the semantic adversarial autoencoder using Gaussian and attribute as priors, respectively. SAE-dir is our implementation of SAE which directly solve the optimization problem in its original form.

	CUB	AwA
SAAE-exp	**17.8**	**60.0**
SAAE-imp	0.5	2.4
SAE-dir	7.0	53.8
SAE-GAN	6.9	54.0

Table 2: Top-1 per-class accuracy (%) of generalized ZSL. SAAE-exp and SAAE-imp are the semantic adversarial autoencoder using Gaussian and attribute as priors, respectively. SAE-dir is our implementation of SAE which directly solve the optimization problem in its original form.

In the experiments of SAAE with explicit matching, the coefficients of the matching, reconstruction, and GAN terms are all set to 1. We run for 100 epochs, and report the result top-1 per-class accuracy using negative cosine similarity as the distance at test time. The mean and standard deviation of the Gaussian priors are set to be the mean and standard deviation of the attributes of the dataset.

In the experiments of SAAE with implicit matching, the coefficient of the reconstruction term is 10 and that of the GAN is 1. We run for 100 epochs, and report the result top-1 per-class accuracy using negative cosine similarity as the distance at test time.

In the experiments of our implementation of SAE, we find that better performance is given when the coefficient of the matching loss is 100 and that of the reconstruction loss is 1. This indicates that the performance of SAE is mostly driven by the direct matching between the encoded representation from the image and the attribute vector, and is only marginally driven by the autoencoder.

We find that on the CUB dataset, SAAE with explicit matching achieves the best accuracy in both standard and generalized ZSL tasks. On the AwA dataset, our SAE implementation gives the best accuracy on both standard and generalized ZSL tasks.

6 Discussion

6.1 SAAE-exp

Our extension with adding adversarial part for the autoencoder does not yield higher accuracy than that achieved by the SAE in the standard ZSL scenario. We have experimented with a lot of other settings, such as adding more layers for the autoencoder, changing the hidden dimensions of the generator and the discriminator in the adversarial part, and different sampling strategies (such as sampling from the hidden state to produce negative samples, as described in [13], section 2), but none of them gives any performance boost.

We suspect the reason is that the semantic features plays the essential role in the standard ZSL tasks. We also try ablation study that does not include the semantic space as constraints. In this case the model is just slightly better than random guess. This is expected since ZSL task usually deals with fine-grained image classification, and without the additional semantic space input, the models trained

GAN coefficient	10^{-1}	10^{-2}	10^{-3}	10^{-4}	0
Accuracy (%)	35.4	76.7	78.5	77.7	77.0

Table 3: The top-1 per-class accuray of the standard ZSL on the AwA dataset, using different coefficients for the GAN in SAE-GAN.

only based on image features are likely to have a poor result. However, the only way we figure out to incorporate the semantic features into the adversarial autoencoder is to minimize the L2 loss between the hidden state of the encoder and the semantic representation, and this major constraint limits the accuracy that can be achieved by the model.

The SAAE-exp model performs the best in generalized classification task. We think this is as expected due to the nature of the task. Adversarial autoencoder is proposed to inject the global information of the distribution of the semantic representation of the training data. Through testing the model on both the seen and unseen classes, the model performs well because it has learned the distribution over all the seen classes.

6.2 SAAE-imp

We observe that the classification accuracy using SAAE with the attribute as the prior is very low in all experiments. The reason is that using attribute vectors as the prior only encourages the encoding to have the same distribution as that of the attributes, but not educating the encoder to learn how to generate the encoding such that each component of the encoding can best match its corresponding component of the attribute vector. Since each component of the attribute vector has very specific meaning, and at test time we predict the class label by matching each component of the given attribute vector with the corresponding component of the encoding generated from the test image, lacking the ability to correctly predict each component of the attribute vector leads to the poor performance of this approach.

6.3 SAE

We note that SAE gives the best performance when the coefficient λ in (2) is much smaller than 1. This indicates that the learning is actually driven by learning how to generate (from attribute to reconstructed image), rather than how to infer (from image to attribute). We think this is an observation that has not be pointed out in the original SAE paper [8].

6.4 SAE-GAN

By adding GAN on top of the SAE, we find that on the AwA dataset it improves the accuracy, but on the CUB dataset it does not. For the standard ZSL on the AwA dataset, in which GAN makes the most significant improvement, we conduct more detailed experiments to check how accuracy changes by using different coefficients λ_{GAN} for the GAN while keeping λ_{SAE} fixed. The results are shown in Table 3. We observe that with suitable coefficient, GAN slightly improves the accuracy, but if the coefficient of GAN is too large, it harms the performance.

7 Conclusion

In this paper, we proposed to use the adversarial antoencoder framework incorporating semantic features as a solution to ZSL problem. Our method, the semantic adversarial autoencoder, projects the image representation into the semantic feature space with a discriminator matching the projection to a given prior. It outperforms the semantic autoencoder in the generalized zero-shot learning tasks on the AwA and CUB datasets.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[2] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. pages 5327–5336, 2016.

[3] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. *International Conference on Computer Vision*, 2013.

[4] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[6] Bharath Hariharan, SVN Vishwanathan, and Manik Varma. Efficient max-margin multi-label classification with applications to zero-shot learning. *Machine learning*, 88(1-2):127–155, 2012.

[7] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[8] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.

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

[10] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[11] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] CS Lu. Solution of the matrix equation ax+ xb= c. *Electronics Letters*, 7(8):185–186, 1971.

[13] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[15] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. *Computer Vision–ECCV 2010*, pages 776–789, 2010.

[16] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[17] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[18] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *arXiv:1707.00600*.

[19] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016.