arXiv:1704.08345v1 [cs.CV] 26 Apr 2017

Semantic Autoencoder for Zero-Shot Learning

Elyor Kodirov Tao Xiang Shaogang Gong
Queen Mary University of London, UK
{e.kodirov, t.xiang, s.gong}@qmul.ac.uk

Abstract

Existing zero-shot learning (ZSL) models typically learn a projection function from a feature space to a semantic embedding space (e.g. attribute space). However, such a projection function is only concerned with predicting the training seen class semantic representation (e.g. attribute prediction) or classification. When applied to test data, which in the context of ZSL contains different (unseen) classes without training data, a ZSL model typically suffers from the project domain shift problem. In this work, we present a novel solution to ZSL based on learning a 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 ZSL models. However, the decoder exerts an additional constraint, that is, the projection/code must be able to reconstruct the original visual feature. We show that with this additional reconstruction constraint, the learned projection function from the seen classes is able to generalise better to the new unseen classes. Importantly, the encoder and decoder are linear and symmetric which enable us to develop an extremely efficient learning algorithm. Extensive experiments on six benchmark datasets demonstrate that the proposed SAE outperforms significantly the existing ZSL models with the additional benefit of lower computational cost. Furthermore, when the SAE is applied to supervised clustering problem, it also beats the state-of-the-art.

1. Introduction

A recent endeavour of computer vision research is to scale the visual recognition problem to large-scale. This is made possible by the emergence of large-scale datasets such as ImageNet [52] and the advances in deep learning techniques [31, 53, 57, 55]. However, scalability remains an issue because beyond daily objects, collecting image samples for rare and fine-grained object categories is difficult even with modern image search engines. Taking the ImageNet dataset for example, the popular large-scale visual recognition challenge (ILSVRC) [52] mainly focuses on the task of recognising 1K classes, a rather small subset of the full ImageNet dataset consisting of 21,814 classes with 14M images. This is because many of the 21K object classes are only composed of a handful of images including 296 classes with only one image.

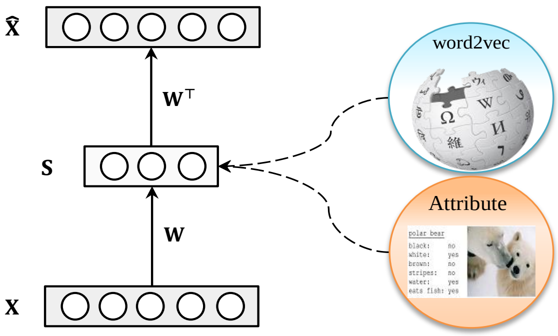


Figure 1. The proposed semantic autoencoder leverages the semantic side information such as attributes and word vector, while learning an encoder and a decoder.

Humans can identify approximately 30,000 basic object categories [9] and many more sub-classes, e.g. breeds of dogs and combination of attributes and objects. Importantly, humans are very good at recognising objects without seeing any visual samples. In machine learning, this is considered as the problem of *zero-shot learning* (ZSL). For example, a child would have no problem recognising a "zebra" if he/she has seen horses before and also learned that a "zebra" is like a horse with black-and-white stripes. Inspired by humans' ZSL ability, there is a recent surge of interest in machine learning based ZSL for scaling up visual recognition to unseen object classes without the need for additional data collection [66, 48, 19, 49, 2, 13, 58, 1, 54, 22, 23, 40, 11].

Zero-shot recognition relies on the existence of a labelled training set of seen classes and the knowledge about how each unseen class is semantically related to the seen classes. Seen and unseen classes are usually related in a high dimensional vector space, which is called semantic embedding space. Such a space can be a semantic attribute space [25] or a semantic word vector space [19, 56]. In the semantic embedding space, the names of both seen and unseen classes are embedded as vectors called class prototypes [20]. The semantic relationships between classes can then be measured by a distance, e.g. the prototypes of zebra and horse should be close to each other. Importantly, the same space can be used to project a feature representation of an object image, making visual recognition possible. Specifically, most existing ZSL methods learn a projection (mapping) function from a visual feature space to a semantic embedding space using the labelled training visual data consisting of seen classes only. At test time for recognising unseen objects, this mapping function is then used to project the visual representation of an unseen class image into the same semantic space where both seen and unseen classes reside. The task of unseen class recognition is then realised by a simple nearest neighbour (NN) search – the class label of the test image is assigned to the nearest unseen class prototype in the projected semantic space.

The training seen classes and testing unseen classes are different. Although they can be considered as two overlapping domains with some degrees of shared semantics, there exists significant domain differences, e.g. the visual appearance of the same attributes can be fairly different in unseen classes. Existing ZSL models mostly suffer from the projection domain shift problem [21]. This is, if the projection for visual feature embedding is learned *only* from the seen classes, the projections of unseen class images are likely to be misplaced (shifted) due to the bias of the training seen classes. Sometimes this shift could be far away from the correct corresponding unseen class prototypes, making the subsequent NN search inaccurate.

In this work, we present a novel approach to zero-shot learning based on the encoder-decoder paradigm [43]. Specifically, an encoder projects a visual feature representation of an image into a semantic representation space such as an attributes space, similar to a conventional ZSL model. However, we also consider the visual feature projection as an input to a decoder which aims to reconstruct the original visual feature representation. This additional reconstruction task imposes a new constraint in learning the visual → semantic projection function so that the projection must also preserve all the information contained in the original visual features, i.e. they can be recovered by the decoder [10]. We show that this additional constraint is very effective in mitigating the domain shift problem. This is because although the visual appearance of attributes may change from seen classes to unseen classes, the demand for more truthful reconstruction of the visual features is generalisable across seen and unseen domains, resulting in the learned project function less susceptible to domain shift.

More precisely, we formulate a semantic autoencoder with the simplest possible encoder and decoder model architecture (Fig. 1): Both have one linear projection to or from a shared latent embedding/code layer, and the encoder and decoder are symmetric so that they can be represented by the same set of parameters. Such a design choice is motivated by computational efficiency – the true potential of a ZSL model is when applied to large-scale visual recognition tasks where computational speed is essential. Even with this simple formulation, solving the resultant optimisation problem efficiently is not trivial. In this work, one such solver is developed whose complexity is independent of the training data size therefore suitable for large-scale problems.

Our semantic autoencoder differs from conventional autoencoder [50] in that the latent layer has clear semantic meaning: It corresponds to the semantic space and is subject to strong supervision. Therefore our model is not unsupervised. Beyond ZSL learning, it can also be readily used for solving other problems where a discriminative low-dimensional representation is required to cluster visually similar data points. To demonstrate its general applicability, our SAE model is formulated for the supervised clustering problem [41, 33].

Our contributions are: (1) A novel semantic encoder-decoder model is proposed for zero-shot learning. (2) We formulate a semantic autoencoder which learns a low-dimensional semantic representation of input data that can be used for data reconstruction. An efficient learning algorithm is also introduced. (3) We show that the proposed semantic autoencoder can be applied to other problems such as supervised clustering. Extensive experiments are carried out on six benchmarks for ZSL which show that the proposed SAE model achieves state-of-the-art performance on all the benchmarks.

2. Related Work

Semantic space A variety of zero-shot learning models have been proposed recently [66, 48, 19, 49, 2, 13, 58, 1, 54, 22, 23, 40, 11]. They use various semantic spaces. Attribute space is the most widely used. However, for large-scale problems, annotating attributes for each class becomes difficult. Recently, semantic word vector space has started to gain popularity especially in large-scale zero-shot learning [19, 56]. Better scalability is typically the motivation as no manually defined ontology is required and any class name can be represented as a word vector for free. Beyond semantic attribute or word vector, direct learning from textual descriptions of categories has also been attempted, e.g. Wikipedia articles [18, 35], sentence descriptions [44].

Visual → Semantic projection Existing ZSL models differ in how the visual space → semantic space projection function is established. They can be divided into three groups: (1) Methods in the first group learn a projection function from a visual feature space to a semantic space either using conventional regression or ranking models [25, 2] or via deep neural network regression or ranking [56, 19, 44, 35]. (2) The second group chooses the reverse projection direction, i.e. semantic → visual [54, 28]. The motivation is to alleviate the hubness problem that commonly suffered by nearest neighbour search in a high-dimensional space [42]. (3) The third group of methods learn an intermediate space where both the feature space and the semantic space are projected to [36, 67, 13]. The encoder in our model is similar to the first group of models, whilst the decoder does the same job as the second group. The proposed semantic autoencoder can thus be considered as a combination of the two groups of ZSL models but with the added visual feature reconstruction constraint.

Projection domain shift The projection domain shift problem in ZSL was first identified by Fu et al. [21]. In order to overcome this problem, a transductive multi-view embedding framework was proposed together with label propagation on graph which requires the access of all test data at once. Similar transdutive approaches are proposed in [47, 28]. This assumption is often invalid in the context of ZSL because new classes typically appear dynamically and unavailable before model learning. Instead of assuming the access to all test unseen class data for transductive learning, our model is based on inductive learning and it relies only enforcing the reconstruction constraint to the training data to counter domain shift.

Autoencoder There are many variants of autoencoders in the literature [5, 27, 34, 59, 46, 51]. They can be roughly divided into two groups which are (1) undercomplete autoencoders and (2) overcomplete autoencoders. In general, undercomplete autoencoders are used to learn the underlying structure of data and used for visualisation/clustering [62] like PCA. In contrast, overcomplete autoencoders are used for classification based on the assumption that higher dimensionnal features are better for classification [15, 8, 7]. Our model is an undercomplete autoencoder since a semantic space typically has lower dimensionality than that of a visual feature space. All the autoencoders above focus on learning features in a unsupervised manner. On the contrary, our approach is supervised while keeping the main characteristic of the unsupervised autoencoders, i.e. the ability to reconstruct the input signal.

Semantic encoder-decoder An autoencoder is only one realisation of the encoder-decoder paradigm. Recently deep encoder-decoder has become popular for a variety of vision problems ranging from image segmentation [4] to image synthesis [64, 45]. Among them, a few recent works also exploited the idea of applying semantic regularisation to the latent embedding space shared between the encoder and decoder [64, 45]. Our semantic autoencoder can be easily extended for end-to-end deep learning by formulating the encoder as a convolutional neural network and the decoder as a deconvolutional neural network with a reconstruction loss.

Supervised clustering Supervised clustering methods exploit labelled clustering training dataset to learn a projection matrix that is shared by a test dataset unlike conventional clustering such as [60, 29]. There are different approaches of learning the projection matrix: 1) metric learning-based methods that use similarity and dissimilarity constraints [32, 30, 63, 16], and 2) regression-based methods that use 'labels' [41, 33]. Our method is more closely related to the regression-based methods, because the training class labels are used to constrain the latent embedding space in our semantic autoencoder. We demonstrate in Sec 5.2 that, similar to the ZSL problem, by adding the reconstruction constraint, significant improvements can be achieved by our model on supervised clustering.

3. Semantic Autoencoder

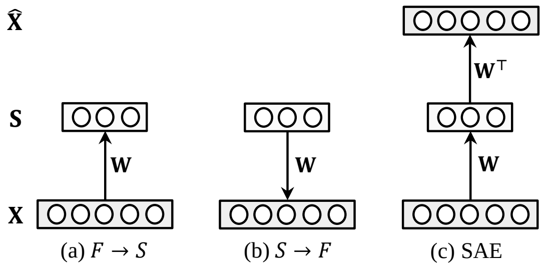
3.1. Linear autoencoder

We first introduce the formulation of a linear autoencoder and then proceed to extend it into a semantic one. In its simplest form, an autoencoder is linear and only has one hidden layer shared by the encoder and decoder. The encoder projects the input data into the hidden layer with a lower dimension and the decoder projects it back to the original feature space and aims to faithfully reconstruct the input data. Formally, given an input data matrix $\mathbf{X} \in \mathbb{R}^{d \times N}$ composed of N feature vectors of d dimensions as its columns, it is projected into a k-dimensional latent space with a projection matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$, resulting in a latent representation $\mathbf{S} \in \mathbb{R}^{k \times N}$. The obtained latent representation is then projected back to the feature space with a projection matrix $\mathbf{W}^* \in \mathbb{R}^{d \times k}$ and becomes $\hat{\mathbf{X}} \in \mathbb{R}^{d \times N}$. We have $k < d$, i.e. the latent representation/code reduces the dimensionality of the original data input. We wish that the reconstruction error is minimised, i.e. $\hat{\mathbf{X}}$ is as similar as possible to $\mathbf{X}$. This is achieved by optimising against the following objective:

$$\min_{\mathbf{W},\,\mathbf{W}^*} \|\mathbf{X} - \mathbf{W}^*\mathbf{W}\mathbf{X}\|_F^2 \tag{1}$$

3.2. Model Formulation

A conventional autoencoder is unsupervised and the learned latent space has no explicit semantic meaning. With the proposed Semantic AutoEncoder (SAE), we assume that each data point also has a semantic representation, e.g., class label or attributes. To make the latent space in the autoencoder semantically meaningful, we take the simplest approach, that is, we force the latent space $\mathbf{S}$ to be the semantic representation space, e.g., each column of $\mathbf{S}$ is now an attribute vector given during training for the corresponding data point. In other words, the latent space is not



(a) $F \to S$ (b) $S \to F$ (c) SAE

Figure 2. Different ways of learning embedding space: (a) F $\longrightarrow$ S, (b) S $\longrightarrow$ F, and (c) Both (our SAE). 'F' – Feature space, and 'S' – Semantic space.

latent any more during training. The learning objective thus becomes:

$$\min_{\mathbf{W},\,\mathbf{W}^*} \|\mathbf{X} - \mathbf{W}^*\mathbf{W}\mathbf{X}\|_F^2 \quad s.t. \quad \mathbf{W}\mathbf{X} = \mathbf{S} \tag{2}$$

To further simplify the model, we consider tied weights [10], that is:

$$\mathbf{W}^* = \mathbf{W}^\top$$

The learning objective is then rewritten as follows:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}^\top\mathbf{W}\mathbf{X}\|_F^2 \quad s.t. \quad \mathbf{W}\mathbf{X} = \mathbf{S} \tag{3}$$

Now we have only one projection matrix to estimate, instead of two (see Fig. 2(c)).

3.3. Optimisation

To optimise the objective in Eq. (3), first we change Eq. (3) to the following form:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}^\top\mathbf{S}\|_F^2 \quad s.t. \quad \mathbf{W}\mathbf{X} = \mathbf{S} \tag{4}$$

by substituting $\mathbf{W}\mathbf{X}$ with $\mathbf{S}$. Solving an objective with a hard constraint such as $\mathbf{W}\mathbf{X} = \mathbf{S}$ is difficult. Therefore, we consider to relax the constraint into a soft one and rewrite the objective as:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}^\top\mathbf{S}\|_F^2 + \lambda\|\mathbf{W}\mathbf{X} - \mathbf{S}\|_F^2 \tag{5}$$

where λ is a weighting coefficient that controls the importance of first and second terms, which correspond to the losses of the decoder and encoder respectively. Now Eq. (5) has a standard quadratic formulation, and it is convex function which has global optimal solution.

To optimise it, we simply take a derivative of Eq. (5) and set it zero. First, we re-organise Eq. (5) using trace properties $\mathrm{Tr}(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^\top)$ and $\mathrm{Tr}(\mathbf{W}^\top\mathbf{S}) = \mathrm{Tr}(\mathbf{S}^\top\mathbf{W})$:

$$\min_{\mathbf{W}} \|\mathbf{X}^\top - \mathbf{S}^\top\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\mathbf{X} - \mathbf{S}\|_F^2 \tag{6}$$

Algorithm 1 SAE in MATLAB

```
function W = SAE(X,S,lambda)
  % SAE - Semantic AutoEncoder
  % Input:
  %    X: dxN data matrix.
  %    S: kxN semantic matrix.
  %    lambda: regularisation parameter.
  %
  % Return:
  %    W: kxd projection matrix.

  A = S*S';
  B = lambda*X*X';
  C = (1+lambda)*S*X';
  W = sylvester(A,B,C);
end
```

Then, we can obtain the derivative of Eq. (6) as follows:

$$\begin{aligned} -\mathbf{S}(\mathbf{X}^\top - \mathbf{S}^\top\mathbf{W}) + \lambda(\mathbf{W}\mathbf{X} - \mathbf{S})\mathbf{X}^\top &= 0 \\ \mathbf{S}\mathbf{S}^\top\mathbf{W} + \lambda\mathbf{W}\mathbf{X}\mathbf{X}^\top &= \mathbf{S}\mathbf{X}^\top + \lambda\mathbf{S}\mathbf{X}^\top \end{aligned} \tag{7}$$

If we denote $\mathbf{A} = \mathbf{S}\mathbf{S}^\top$, $\mathbf{B} = \lambda\mathbf{X}\mathbf{X}^\top$, and $\mathbf{C} = (1 + \lambda)\mathbf{S}\mathbf{X}^\top$, we have the following formulation:

$$\mathbf{A}\mathbf{W} + \mathbf{W}\mathbf{B} = \mathbf{C}, \tag{8}$$

which is a well-known Sylvester equation which can be solved efficiently by the Bartels-Stewart algorithm [6]. In MATLAB, it can be implemented with *a single line* of code: `sylvester`[1]. Importantly, the complexity of Eq. (8) depends on the size of feature dimension ($\mathcal{O}(d^3)$), and not on the number of samples; it thus can scale to large-scale datasets. Algorithm 1 shows a 6-line MATLAB implementation of our solver.

4. Generalisation

4.1. Zero-Shot Learning

Problem definition Let $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ and $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_u\}$ denote a set of s seen and u unseen class labels, and they are disjoint $\mathbf{Y} \cap \mathbf{Z} = \varnothing$. Similarly $\mathbf{S}_Y = \{\mathbf{s}_1, \dots, \mathbf{s}_s\} \in \mathbb{R}^{s \times k}$ and $\mathbf{S}_Z = \{\mathbf{s}_1, \dots, \mathbf{s}_u\} \in \mathbb{R}^{u \times k}$ denote the corresponding seen and unseen class semantic representations (e.g. k-dimensional attribute vector). Given training data with N number of samples $\mathbf{X}_Y = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{s}_i)\} \in \mathbb{R}^{d \times N}$, where $\mathbf{x}_i$ is a d-dimensional visual feature vector extracted from the i-th training image from one of the seen classes, zero-shot learning aims to learn a classifier $f : \mathbf{X}_Z \to \mathbf{Z}$ to predict the label of the image coming from unseen classes, where $\mathbf{X}_Z = \{(\mathbf{x}_i, \mathbf{z}_i, \mathbf{s}_i)\}$ is the test data and $\mathbf{z}_i$ and $\mathbf{s}_i$ are unknown.

SAE for zero-shot learning Given semantic representation $\mathbf{S}$ such as attributes, and the training data $\mathbf{X}_\mathbf{Y}$, using

[1] https://uk.mathworks.com/help/matlab/ref/sylvester.html

our SAE, we first learn the encoder $\mathbf{W}$ and decoder $\mathbf{W}^\top$ by Algorithm 1. Subsequently, zero-shot classification can be performed in two spaces:

1) With the encoder projection matrix $\mathbf{W}$: We can embed a new test sample $\mathbf{x}_i \in \mathbf{X}_Z$ to the semantic space by $\hat{\mathbf{s}}_i = \mathbf{W}\mathbf{x}_i$. After that, the classification of the test data in the semantic space can be achieved by simply calculating the distance between the estimated semantic representation $\mathbf{s}_i$ and the projected prototypes $\mathbf{S}_Z$:

$$\Phi(\mathbf{x}_i) = \arg\min_j D(\hat{\mathbf{s}}_i, \mathbf{S}_{Z_j}) \tag{9}$$

where $\mathbf{S}_{Z_j}$ is j-th prototype attribute vector of the j-th unseen class, D is a distance function, and $\Phi(\cdot)$ returns the class label of the sample.

2) With the decoder projection matrix $\mathbf{W}^\top$: Similarly, we can embed the prototype representations to the visual feature space by $\hat{\mathbf{x}}_i = \mathbf{W}^T\mathbf{s}_i$ where $\mathbf{s}_i \in \mathbf{S}_Z$ and $\hat{\mathbf{x}}_i \in \hat{\mathbf{X}}_Z$ is the projected prototype. Then, the classification of the test data in the feature space can be achieved by calculating the distance between the feature representation $\mathbf{x}_i$ and the prototype projections in the feature space $\hat{\mathbf{X}}_Z$:

$$\Phi(\mathbf{x}_i) = \arg\min_j D(\mathbf{x}_i, \hat{\mathbf{X}}_{Z_j}) \tag{10}$$

where $\hat{\mathbf{X}}_{Z_j}$ is j-th unseen class prototype projected in the feature space.

In our experiments we found that the two testing strategies yield very similar results (see Sec. 5.1). We report results with both strategies unless otherwise specified.

4.2. Supervised Clustering

For supervised clustering we are given a set of training data with class labels only, and a test set that share the same feature representation as the training data and need to be grouped into clusters. Let $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ be a set of s training class labels. Denote $\mathbf{S}_Y = \{\mathbf{s}_1, \dots, \mathbf{s}_s\} \in \mathbb{R}^{s\times k}$ as the corresponding semantic representations. Given a training data set with N number of samples $\mathbf{X}_Y = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{s}_i)\} \in \mathbb{R}^{d\times N}$, where $\mathbf{x}_i$ is a d-dimensional visual feature vector extracted from the i-th training image we aim to learn a projection function $f : \mathbf{X}_Y \to \mathbf{S}_Y$ from the training data and then apply the same projection function to a set of test data $\mathbf{X}_Z$ before clustering can be carried out.

Using our SAE, the projection function is our encoder $\mathbf{W}$. With only the training class label, the semantic space is the label space, that is, $\mathbf{s}_i$ is an one-hot class label vector with only the element corresponding to the image class assuming the value 1, and all other elements set to 0. After the test data is projected into the training label space, we use k-means clustering as in existing work [41, 33] for fair comparison. The demo code of our model is available at https://elyorcv.github.io/projects/sae.

4.3. Relations to Existing Models

Relation to ZSL models Many existing ZSL models learn a projection function from a visual feature space to a semantic space (see Fig. 2(a)). If the projection function is formulated as linear ridge regression as follows:

$$\min_{\mathbf{W}} \|\mathbf{W}\mathbf{X} - \mathbf{S}\|_F^2 + \lambda\|\mathbf{W}\|_F^2, \tag{11}$$

we can see that comparing Eq. (11) with Eq. (5), this is our encoder with an additional regularisation term on the project matrix $\mathbf{W}$.

Recently, [54] proposed to reverse the projection direction: They project the semantic prototypes into the features space:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}^\top\mathbf{S}\|_F^2 + \lambda\|\mathbf{W}\|_F^2 \tag{12}$$

so this is the decoder of our SAE but again with the regularisation term to avoid overfitting (see Fig. 2(b)).

Our approach can thus be viewed as the combination of both models when ridge regression is chosen as the project function and without considering the $\|\mathbf{W}\|_F^2$ regularisation. This regularisation is unnecessary in our model due to the symmetric encoder-decoder design – since $\mathbf{W}^* = \mathbf{W}^\top$, the norm of the encoder projection matrix $\|\mathbf{W}\|_F^2$ cannot be big because it will then produce large-valued projections in the semantic space, and after being multiplied with the large-norm decoder project matrix, will result in bad reconstruction. In other words, the regularisation on the norm of the projection matrices have been automatically taken care of by the reconstruction constraint [10].

Relation to supervised clustering models Recently, [41, 33] show that regression can be used to learn a mahalanobis distance metric for supervised clustering. Specifically, given data $\mathbf{X}$ with corresponding labels, the so-called 'encoded labels' $\mathbf{S}$ are generated and normalised as $\mathbf{S} = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1/2} \in \mathbb{R}^{s\times N}$, where s is the number of training labels [33]. Then linear regression is employed to obtain a projection matrix $\mathbf{W}$ for projecting the data from the feature space to the label space. At test time, $\mathbf{W}$ is applied to test data. Then, k-means clustering is applied to the projected data. Again, these models can be considered as the encoder of our SAE. We shall show that with the decoder and the additional reconstruction constraint, the learned code and distance metric become more meaningful, yielding superior clustering performance on the test data.

Dataset	#instances	SS	SS-D	# seen/unseen
AwA [25]	30,475	A	85	40 / 10
CUB [12]	11,788	A	312	150 / 50
aP&Y [3]	15,339	A	64	20 / 12
SUN [24]	14,340	A	102	645 / 72 $^{(*)}$
ImNet-1 [52]	$1{,}2 \times 10^6$	W	1,000	800 / 200
ImNet-2 [52]	218,000	W	1,000	1,000 / 360

Table 1. Benchmark datasets for evaluation. Notation: 'SS' – semantic space, 'SS-D' – the dimension of semantic space, 'A' – attribute, and 'W' – word vector. $^{(*)}$ – another split of 707/10 is also used for SUN [26, 67].

5. Experiments

5.1. Zero-Shot Learning

Datasets Six benchmark datasets are used. Four of them are small-scale datasets: Animals with Attributes (AwA) [25], CUB-200-2011 Birds (CUB) [12], aPascal&Yahoo (aP&Y) [3], and SUN Attribute (SUN) [24]. The two large-scale ones are ILSVRC2010 [17] (ImNet-1), and ILSVRC2012/ILSVRC2010 [52] (ImNet-2). In ImNet-2, as in [22], the 1,000 classes of ILSVRC2012 are used as seen classes, while 360 classes of ILSVRC2010, which are not included in ILSVRC2012, for unseen classes. The summary of these datasets is given in Table 1.

Semantic spaces We use attributes as the semantic space for the small-scale datasets, all of which provide the attribute annotations. Semantic word vector representation is used for large-scale datasets. We train a skip-gram text model on a corpus of 4.6M Wikipedia documents to obtain the word2vec[2] [38, 37] word vectors.

Features All recent ZSL methods use visual features extracted by deep convolutional neural networks (CNNs). In our experiments, we use GoogleNet features [57] which is the 1024D activation of the final pooling layer as in [2]. The only exception is for ImNet-1: For fair comparison with published results, we use Alexnet [31] architecture, and train it from scratch using the 800 seen classes, resulting in 4096D visual feature vectors computed using the FC7 layer.

Parameter settings Our SAE model has only one free parameter: λ (see Eq. (5)). As in [67], its values is set by class-wise cross-validation using the training data. The dimension of the embedding (middle) layer always equals to that of the semantic space. Only SUN dataset has multiple splits. We use the same 10 splits used in [13], and report the average performance.

Evaluation metric For the small-scale datasets, we use multi-way classification accuracy as in previous works, while for the large-scale datasets flat hit@K classification accuracy is used as in [19]. hit@K means that the test image is classified to a 'correct label' if it is among the top K labels. We report hit@5 accuracy as in other works, unless otherwise stated.

Competitors 14 existing ZSL models are selected for the small-scale datasets and 7 for the large-scales ones (much fewer existing works reported results on the large-scale datasets). The selection criteria are: (1) recent work: most of them are published in the past two years; (2) competitiveness: they clearly represent the state-of-the-art; and (3) representativeness: they cover a wide range of models (see Sec. 2).

Comparative evaluation From the results in Table 2 we can make the following observations: (1) Our SAE model achieves the best results on all 6 datasets. (2) On the small-scale datasets, the gap between our model's results to the strongest competitor ranges from 3.5% to 6.5%. This is despite the fact that most of the compared models use far complicated nonlinear models and some of them use more than one semantic space. (3) On the large-scale datasets, the gaps are even bigger: On the largest ImNet-2, our model improves over the state-of-the-art SS-Voc [22] by 8.8%. (4) Both the encoder and decoder projection functions in our SAE model (SAE ($\mathbf{W}$) and SAE ($\mathbf{W}^\top$) respectively) can be used for effective ZSL. The encoder projection function seems to be slightly better overall.

Ablation study The key strength of our model comes from the additional reconstruction constraint in the autoencoder formulation. Since most existing ZSL models use more sophisticated projection functions than our linear mapping, in order to evaluate how important this additional constraint is, we consider ZSL baselines that use the same simple projection functions as our model. As discussed in Sec. 4.3, without the constraint both the encoder and decoder can be considered as conventional ZSL models with linear ridge regression as projection function, and they differ only in the project directions. Table 3 shows than, when the projection function is the same, adding the additional reconstruction constraint makes a huge difference. Note that comparing to the state-of-the-art results in Table 2, simple ridge regression is competitive but clearly inferior to the best models due to its simple linear projection function. However, when the two models are combined in our SAE, we obtain a much more powerful model that beats all existing models.

Generalised Zero-Shot Learning Another ZSL setting that emerges recently is the generalised setting under which the test set contains data samples from both the seen and unseen classes. We follow the same setting of [14]. Specifically, we hold out 20% of the data samples from the seen classes and mix them with the data samples from the unseen classes. The evaluation metric is now Area Under Seen-Unseen accuracy Curve (AUSUC), which measures how well a zero-shot learning method can trade-off between recognising data from seen classes and that of un-

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

Small-scale datasets						Large-scale datasets			
Method	SS	AwA	CUB	aP&Y	SUN	Method	SS	ImNet-1	ImNet-2
DAP [25]	A	60.1	-	38.2	72.0 | 44.5	Rohrbach *et al.* [48]	W	34.8	–
ESZSL [49]	A	75.3	48.7	24.3	82.1 | 18.7	Mensink *et al.* [58]	W	35.7	–
SSE [66]	A	76.3	30.4	46.2	82.5 | –	DeViSE [19]	W	31.8	12.8
SJE [2]	A+W	73.9	50.1	-	– | 56.1	ConSE [40]	W	28.5	15.5
JLSE [67]	A	80.5	41.8	50.4	83.8 | –	AMP [23]	W	41.0	13.1
SynC^{struct} [13]	A	72.9	54.4	-	– | 62.7	SS-Voc [22]	W	–	16.8
MLZSC [11]	A	77.3	43.3	53.2	84.4 | –	PST [47]	W	34.0	–
DS-SJE [44]	A/D	-	50.4/56.8	-	– | –				
AMP [11]	A+W	66.0	-	-	– | –				
DeViSE [19]	A/W	56.7/50.4	33.5	-	– | –				
RRZSL [54]	A	80.4	52.4	48.8	84.5 | –				
Ba *et al.* [35]	A/W	69.3/58.7	34.0	-	– | –				
MTMDL [65]	A/W	63.7/55.3	32.3	-	– | –				
SS-voc [22]	A/W	78.3/68.9	-	-	– | –				
SAE ($\mathbf{W}$)	A	**84.7**	**61.4**	**55.4**	**91.0 | 65.2**	SAE ($\mathbf{W}$)	W	**46.1**	**26.3**
SAE ($\mathbf{W}^{\top}$)	A	**84.0**	**60.9**	**54.8**	**91.5 | 65.2**	SAE ($\mathbf{W}^{\top}$)	W	**45.4**	**27.2**

Table 2. Comparative ZSL classification accuracy (%, hit@5 for large-scale datasets). For SS (Semantic Space), '/' means 'or' and '+' means 'and'. For CUB, 10 sentence description per image are also used in [44] as input to a language model (word-CNN-RNN) to compute semantic space ('D'). For the SUN dataset, the results are for the 707/10 and 645/72 splits respectively, separated by '|'. '-' means that no reported results are available. $\mathbf{W}$ parametrises the projection function of the encoder and $\mathbf{W}^{\top}$ the decoder.

Projection	AwA	CUB	aP&Y	SUN
F → S	60.6	41.1	30.5	71.5
F ← S	80.4	52.4	48.8	84.5
SAE	**84.7**	**61.4**	**55.4**	**91.0**

Table 3. The importance of adding the reconstruction constraint. Both compared methods are based on ridge regression and differ in the projection direction between the visual and semantic spaces. Attributes are used. The encoder is used.

Method	AwA	CUB
DAP [25]	0.366	0.194
IAP [25]	0.394	0.199
ConSE [40]	0.428	0.212
ESZSL [49]	0.449	0.243
SynC^{struct} [13]	**0.583**	0.356
SAE	0.579	**0.448**

Table 4. Comparative evaluation on generalised zero-shot learning on AwA and CUB. Encoder is used.

seen classes [14]. The upper bound of this metric is 1. The results on AwA and CUB are presented in Table 4 comparing our model with 5 other alternatives. We can see that on AwA, our model is slightly worse than the state-of-the-art method SynC^{struct} [13]. However, on the more challenging CUB dataset, our method significantly outperforms the competitors.

Computational cost We evaluate the computational cost of our method in comparison with three linear ZSL models SSE [66], ESZSL [49] and AMP [23] which are among the more efficient existing ZSL models. Table 5 shows that for model training, our SAE is at least 10 times faster. For testing, our model is still the fastest, although ESZSL is close.

Method	Training	Test
SSE [66]	1312	9.20
ESZSL [49]	16	0.08
AMP [23]	844	0.23
SAE	1.3	0.07

Table 5. Evaluating the computational cost (in second) on AwA. Encoder is used.

5.2. Supervised Clustering

Datasets Two datasets are used. A *synthetic dataset* is generated following [33]. Specifically, the training set is composed of 3-dimensional samples divided into 3 clusters, and each cluster has 1,000 samples. Each of these clusters is composed of two subsclusters as shown in Fig. 3(a). What makes the dataset difficult is that the subclusters of the same cluster are closer to the subsclusters from different categories than to each other when the distance is measured with Euclidean distance. Furthermore, some samples are corrupted by noise which put them in the subclusters of other categories in the feature space. We generate our test dataset with the similar properties (and the same number of examples N=3000) as the training set. To make clustering more challenging, the number of samples for each cluster is made different: 1000, 2000, and 4000 for three clusters respectively. This dataset is designed to evaluate how robust the method is against the size of clusters and its ability to avoid being biased by the largest category. More details on the dataset can be found in [33, 32, 63]. We also test our algorithm with a *real dataset* – Oxford Flowers-17

Method	SAE	L_2	Xiang *et al.* [63]	Lajugie *et al.* [32]	KISSME [30]	ITML [16]	LMNN [61]	MLCA [33]
Test loss	**0.01**	3.0	0.7	0.11	0.07	0.08	3.0	0.07
Training Time (in second)	0.020	NT	4378	336	0.5	370	4	**0.004**

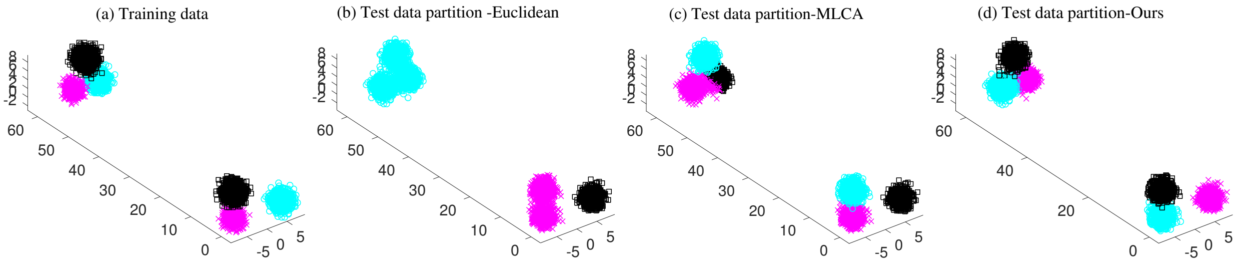
Table 6. Supervised clustering results on synthetic data with clusters of same size. 'NT'– No Training, L_2–Euclidean distance. Encoder is used.

Method	SAE	L_2	Xiang *et al.* [63]	Lajugie *et al.* [32]	KISSME [30]	ITML [16]	LMNN [61]	MLCA [33]
Test loss	**0.01**	3.0	3.0	0.09	2.02	3.0	3.0	0.09
Training Time (in second)	0.026	NT	21552	2462	2	1260	11	**0.005**

Table 7. Supervised clustering results on synthetic data with clusters of different sizes and with noise (lower is better). Encoder is used.

Method	SAE	Lajugie *et al.* [32]	KISSME [30]	ITML [16]	LMNN [61]	MLCA [33]
Test loss	**1.19±0.01**	1.38±0.02	1.59±0.02	1.50±0.02	1.79±0.02	1.29±0.01
Training Time	93 seconds	5 days	11 minutes	2 hours	1 day	**39 seconds**

Table 8. Supervised clustering (segmentation) results on Oxford Flowers. Encoder is used.


(a) Training data

(b) Test data partition -Euclidean

(c) Test data partition-MLCA

(d) Test data partition-Ours

Figure 3. (a) Original training dataset, (b) Clustering obtained by k-means with Euclidean distance, (c) MLCA [33], (d) SAE

(848 images) [39]. We follow exactly the same settings of [33]. Specifically, a ground truth foreground/background segmentation is provided for every image. To extract features, first, images are resized with a height of 100 pixels, and SIFT and color features (Lab, RGB, and intensity) are extracted from each 8×8 patch centred at every pixel, resulting a 135D feature vector for each pixel. Each image has about 10^4 patches, and the data matrix for the whole dataset has about 2.2×10^6 rows – this is thus a large-scale problem. The dataset has 5 random split with 200 images for training, 30 for validation, and the rest for testing.

Evaluation metric We calculate the clustering quality with a loss defined as $\Delta = \|\hat{C} - C\|^2$ [32, 33], where C and $\hat{C}$ are ground truth and predicted clustering matrix (obtained using k-means) respectively.

Competitors We compare our method with the state-of-the-art methods which all formulate the supervised clustering problem as a metric learning problem. These include Xiang *et al.* [63], Lajugie *et al.* [32], KISSME [30], ITML [16], LMNN [61], and MLCA [33].

Comparative evaluation Table 6 and Table 7 show the synthetic data results with and without noise respectively. It can be seen that in terms of clustering accuracy, our method is much better than all compared methods. On computational cost, our model is more expensive than MLCA but much better than all others. Figure 3 visualises the clustering results. On the real image segmentation data, Table 8 compares our SAE with other methods. Again, we can see that SAE achieves the best clustering accuracy. The training time for SAE is 93 seconds, while MLCA is 39 seconds. Note that the data size is 2.2×10^6, so both are very efficient.

6. Conclusion

We proposed a novel zero-shot learning model based on a semantic autoencoder (SAE). The SAE model uses very simple and computationally fast linear projection function and introduce an additional reconstruction objective function for learning a more generalisable projection function. We demonstrate through extensive experiments that this new SAE model outperforms existing ZSL models on six benchmarks. Moreover, the model is further extended to address the supervised clustering problem and again produces state-of-the-art performance.

Acknowledgement

The authors were funded in part by the European Research Council under the FP7 Project SUNNY (grant agreement no. 313243).

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013. 1, 2

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, pages 2927–2936, 2015. 1, 2, 6, 7

[3] F. Ali, E. Ian, H. Derek, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009. 6

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 3

[5] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989. 3

[6] R. BARTELS. Solution of the matrix equation ax+ xb= c [f4]. *Commun. ACM*, 15:820–826, 1972. 4

[7] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. 3

[8] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. 2007. 3

[9] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94(2):115, 1987. 1

[10] Y.-l. Boureau, Y. L. Cun, et al. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192, 2008. 2, 4, 5

[11] M. Bucher, S. Herbin, and F. Jurie. Improving semantic embedding consistency by metric learning for zero-shot classification. In *European Conference on Computer Vision*, pages 730–746. Springer, 2016. 1, 2, 7

[12] W. Catherine, B. Steve, W. Peter, P. Pietro, and B. Serge. The caltech-ucsd birds-200-2011 dataset. In *California Institute of Technology*, 2011. 6

[13] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 3, 6, 7

[14] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*. Springer, 2016. 6, 7

[15] M. Chen, W. EDU, and Z. E. Xu. Marginalized denoising autoencoders for domain adaptation. In *ICML*, 2014. 3

[16] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007. 3, 8

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 6

[18] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013. 2

[19] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 1, 2, 3, 6, 7

[20] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, pages 584–599, 2014. 2

[21] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *IEEE Tran. PAMI*, 37(11):2332–2345, 2015. 2, 3

[22] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 6, 7

[23] Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot object recognition by semantic manifold distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2644, 2015. 1, 2, 7

[24] P. Genevieve, X. Chen, S. Hang, and H. James. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014. 6

[25] L. C. H, N. Hannes, and H. Stefan. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 1, 2, 6, 7

[26] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems*, pages 3464–3472, 2014. 6

[27] K. Kavukcuoglu, R. Fergus, Y. LeCun, et al. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1605–1612. IEEE, 2009. 3

[28] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015. 3

[29] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Learning robust graph regularisation for subspace clustering. In *British Machine Vision Conference*, 2016. 3

[30] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012. 3, 8

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 6

[32] R. Lajugie. Large-margin metric learning for constrained partitioning problems. In *ICML*, 2014. 3, 7, 8

[33] M. T. Law, Y. Yu, M. Cord, and E. P. Xing. Closed-form training of mahalanobis distance for supervised clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3, 5, 7, 8

[34] H. Lee, P. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009. 3

[35] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, pages 4247–4255, 2015. 2, 3, 7

[36] Y. Lu. Unsupervised learning on neural network outputs: with application in zero-shot learning. *arXiv preprint arXiv:1506.00990*, 2015. 3

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 6

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 6

[39] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1447–1454. IEEE, 2006. 8

[40] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, pages 488–501, 2014. 1, 2, 7

[41] M. Perrot and A. Habrard. Regressive virtual metric learning. In *Advances in Neural Information Processing Systems*, pages 1810–1818, 2015. 2, 3, 5

[42] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *JMLR*, 11(9):2487–2531, 2010. 3

[43] M. Ranzato, Y. Boureau, S. Chopra, and Y. Lecun. *A unified energy-based framework for unsupervised learning*. 2007. 2

[44] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 2, 3, 7

[45] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text-to-image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016. 3

[46] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 833–840, 2011. 3

[47] M. Rohrbach, S. Ebert, and B. Schiele. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54, 2013. 3, 7

[48] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1641–1648. IEEE, 2011. 1, 2, 7

[49] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, 2015. 1, 2, 7

[50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. 2

[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 2012. 3

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 6

[53] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1

[54] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 135–151. Springer, 2015. 1, 2, 3, 5, 7

[55] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[56] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 2, 3

[57] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1, 6

[58] M. Thomas, V. Jakob, P. Florent, and C. Gabriela. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Computer Vision–ECCV 2012*, pages 488–501. Springer, 2012. 1, 2, 7

[59] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 3

[60] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 3

[61] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 8

[62] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. *ICML*, 2015. 3

[63] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2003. 3, 7, 8

[64] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016. 3

[65] Y. Yang and T. Hospedales. A unified perspective on multi-domain and multi-task learning. 2015. 7

[66] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4166–4174, 2015. 1, 2, 7

[67] Z. Zhang and V. 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. 3, 6, 7