A Simple yet Effective Model for Zero-Shot Learning

Xi Hang Cao[1,2], Zoran Obradovic[2], Kyungnam Kim[1]
[1]HRL Laboratories, LLC, Malibu, CA
[2]Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA
{xi.hang.cao, zoran.obradovic}@temple.edu, kkim@hrl.com

Abstract

Zero-shot learning has tremendous application value in complex computer vision tasks, e.g. image classification, localization, image captioning, etc., for its capability of transferring knowledge from seen data to unseen data. Many recent proposed methods have shown that the formulation of a compatibility function and its generalization are crucial for the success of a zero-shot learning model. In this paper, we formulate a softmax-based compatibility function, and more importantly, propose a regularized empirical risk minimization objective to optimize the function parameter which leads to a better model generalization. In comparison to eight baseline models on four benchmark datasets, our model achieved the highest average ranking. Our model was effective even when the training set size was small and significantly outperforming an alternative state-of-the-art model in generalized zero-shot recognition tasks.

1. Introduction

Recent advances in deep neural network technologies have resulted in significant progress in large-scale systems for recognition tasks; for example, classification [21], localization [35], image captioning [19], speech recognition [9, 16], machine translation [3]. However, their successes strongly depend on the availability of labeled training data, which become costly or impossible to collect as the number of classes and complexity of the tasks increase. To address this problem, zero-shot learning [22, 24, 27] attempts to generalize what it has learned on a subset of all the possible classes (seen), to new classes it hasn't seen (unseen).

Zero-shot learning is considered as a special case of transfer learning [28]. The goal of transfer learning is to transfer knowledge (i.e. classification model) from a source domain to a target domain while the relation of the two domains is known *a priori* or implicitly. In zero-shot learning, the objective is to generalize a classification model that has been trained on the seen labeled data (source domain) to the unseen data (target domain). The generalization of a zero-shot learning model is realized by leveraging the class labels' semantic representations (descriptions) which may reveal the similarities/dissimilarities among seen class labels and unseen class labels. Widely used semantic representations include human annotated attributes [12, 23] and word vectors [5, 13].

Most zero-shot learning models share a similar structure (Figure 1). Typically, there is a feature extraction layer for extracting features from the images (e.g. SIFT [25], HoG [10] and pre-trained deep neural networks [32, 33]). Many of them also employ a feature embedding layer to exploit the latent structures of the features. Each class label is also associated with a semantic feature vector (e.g. human annotated attributes [12] and word vectors [5]).

Many efficient embedding algorithms have been proposed to exploit the structure of semantic space [1, 2, 18, 30, 39, 41, 42, 43]. In training, the models use the features and the labels' semantic vectors (or their embedded versions) to learn a compatibility function by minimizing a regularized empirical risk minimization objective. The classifier and the classification rule are then derived from the compatibility function, as seen in [1, 2, 6, 7, 13, 26, 27, 31, 34, 39, 43, 44]. Usually, in a classification task, an object is classified to a label whose semantic vector maximizes this compatibility function. Evidence [14, 41, 42, 44] showed that the classification accuracy could be further improved by considering the structures in the feature space of the unseen examples.

1.1. Summary of contributions

In this paper, we propose a simple yet effective model for zero-shot learning. Our model is simple in the sense of conception and implementation, and it is effective in the sense that it yields state-of-the-art performance without formulating sophisticated embeddings.

In our model, the comparability function is formulated by applying a softmax function upon a bilinear function which has been used in many existing models. The softmax-bilinear compatibility function is simple; however, it is capable of capturing the rich information in data. The novelty of our model lies in the formulation of the regularized

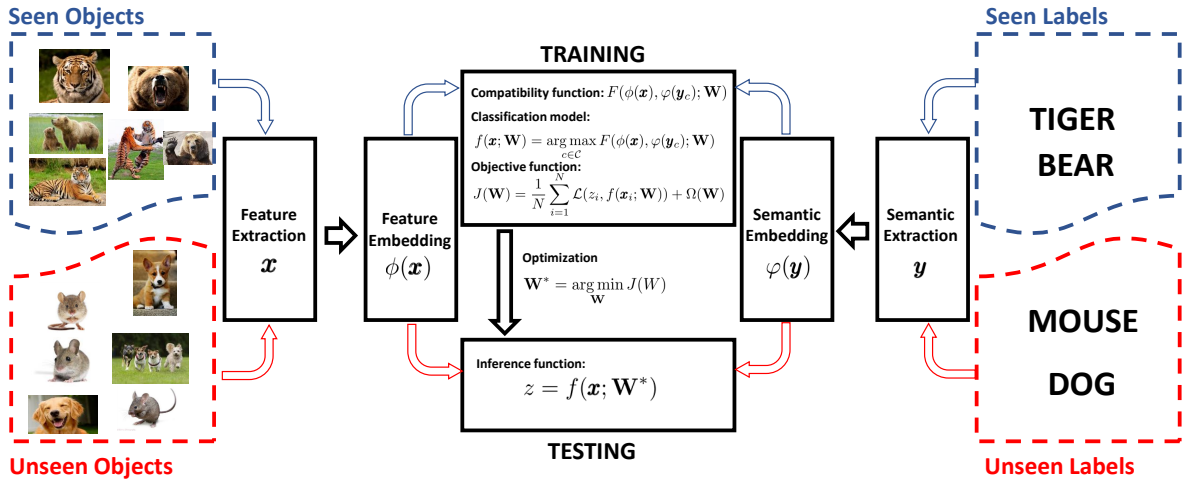

Figure 1: Canonical zero-shot learning framework. While the feature extraction and semantic extraction layers are relatively standard, a zero-shot learning model usually consists of two major components: embedding functions and a compatibility function. In training, the model parameters are learned by solving a regularized empirical risk minimization problem based on the seen examples. In testing, the learned parameter is then used for unseen example classification.

empirical risk minimization objective. Traditionally, negative cross entropy is used as the empirical risk measure for softmax-based models in multi-class classification tasks; in our model, we use a cross-entropy-inspired term as the empirical risk measure, and more importantly, we propose a regularization term for the output of the softmax function. This regularization tremendously improves the generalization of our softmax-based model and makes it be effective for zero-shot learning tasks. The final form of the objective function is smooth, unconstrained, and conceptually simple; therefore, the minimization can be solved by any gradient-based algorithms. Experimental results have shown that our model achieved very competitive accuracies in zero-shot recognition tasks on four benchmark datasets, in comparisons to eight state-of-the-art zero-shot learning models. Experimental results also showed that our model was effective even when the training data size was small. In particular, in the generalized zero-shot recognition task, our model significantly outperformed an alternative state-of-the-art model.

The contribution of our work is three-fold:

- We propose a softmax-bilinear compatibility function which is simple yet effective to capture information in data.
- We introduce a regularization term which is specifically for the outputs of a softmax function; this regularization makes the model more generalized, leading to better performance in zero-shot recognition tasks.
- We have conducted extensive experiments, on four benchmark datasets, to analyze the performance of the proposed model in different settings.

The remainder of the paper is organized as follows. In *Related Work*, we give the formal formulation of zero-shot learning and brief comparisons of a few state-of-the-art zero-shot learning models. In *Proposed Model*, we describe our proposed model in great details. In *Experiments*, we describe the conducted experiments and analyze the results. In *Conclusion*, we summarize our contributions and findings.

2. Related Work

In this section, we introduce the state-of-the-art formulation of zero-shot learning. In order to have more clear and direct comparisons, we first formally define the problem of zero-shot learning.

2.1. Zero-shot learning problem definition

In zero-shot learning, each data point comes in the form of a triplet, $(\boldsymbol{x}_i, \boldsymbol{y}_i, z_i)$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the feature of an object, $z_i \in \mathcal{C} = \{c_1, c_2, \cdots, c_{|\mathcal{C}|}\}$ is the class label, and $\boldsymbol{y}_i \in \mathcal{Y} = \{\boldsymbol{y}^{c_1}, \boldsymbol{y}^{c_2}, \cdots, \boldsymbol{y}^{c_{|\mathcal{C}|}}\}$ is the corresponding semantic representation of the label. In training, only the triplets with class labels in $\mathcal{C}^s \subset \mathcal{C}$ are available (we call $\mathcal{C}^s$ the seen label set, $\mathcal{C}^u$ the unseen label set; $\mathcal{C}^s \bigcup \mathcal{C}^u = \mathcal{C}$ and $\mathcal{C}^s \bigcap \mathcal{C}^u = \emptyset$), and the objective of zero-shot learning is to learn a generalized mapping, $f : \mathcal{X} \rightarrow \mathcal{C}$, by solving the regularized empirical risk minimization problem over the

function parameter $\mathbf{W}$:

$$\underset{\mathbf{W}}{\text{minimize}}\ J(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(z_i, f(\boldsymbol{x}_i;\mathbf{W})) + \Omega(\mathbf{W}), \tag{1}$$

where $J(\cdot)$ is the objective function which includes two terms: $\mathcal{L}:\mathbb{R}\times\mathbb{R}\rightarrow\mathbb{R}$ the empirical loss term and $\Omega(\mathbf{W})$ the regularization term. The mapping f can be expressed by:

$$f(\boldsymbol{x};\mathbf{W}) = \arg\max_{c\in\mathcal{C}} F(\phi(\boldsymbol{x}), \varphi(\boldsymbol{y}^c);\mathbf{W}), \tag{2}$$

where $\phi(\cdot)$ is a feature embedding function, $\varphi(\cdot)$ is a semantic embedding function, and $F(\cdot,\cdot)$ is the compatibility function to measure the compatibility of an object's feature and a label's semantic representation.

Canonical zero-shot learning models usually consist of two major parts: compatibility function learning [40] and embedding function learning. Depending on the specific algorithm, these two components can be learned individually or jointly. In the following subsections, we introduce a few related work based on their formulations in these two components.

2.2. Compatibility functions and parameter learning objective functions

In zero-shot learning, the goal of a compatibility function, $F(\cdot,\cdot)$ is to measure the compatibility score of a feature vector and a semantic vector (i.e. $F(\boldsymbol{x},\boldsymbol{y}^c)$) or their embedded forms (i.e. $F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c))$). Ideally, knowing the label of $\boldsymbol{x}$ is z, the compatibility function should satisfy $F(\boldsymbol{x},\boldsymbol{y}^z) > F(\boldsymbol{x},\boldsymbol{y}^c)\ \forall c\neq z$.

A powerful and simple formulation of the compatibility function is the bilinear form:

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c);\mathbf{W}) = \phi(\boldsymbol{x})\mathbf{W}\varphi(\boldsymbol{y}^c). \tag{3}$$

Some of the representative zero-shot learning models include Attribute Label Embedding (ALE) [1], Deep Visual Semantic Embedding (DEVISE) [13], Structured Joint Embedding (SJE) [2], and Embarrassingly Simple Zero-Shot Learning (ESZSL) [31]. Among them, DEVISE, ALE and SJE use margin-based objective function formulations to learn the parameters; specifically, DEVISE uses a formulation based on ranking SVM [17]:

$$\sum_{c\in\mathcal{C}^s}[\Delta(\boldsymbol{y}^c,\boldsymbol{y}_i) + F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}^c);\mathbf{W}) - F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}_i);\mathbf{W})]_+, \tag{4}$$

where $\Delta(\boldsymbol{y}^c,\boldsymbol{y}_i)$ equals 0 when $\boldsymbol{y}^c=\boldsymbol{y}_i$, and equals 1 otherwise. ALE uses a weighted approximate ranking objective [37]

$$\sum_{c\in\mathcal{C}^s}\gamma_{r_\Delta(\boldsymbol{x}_i,\boldsymbol{y}_i)}[\Delta(\boldsymbol{y}^c,\boldsymbol{y}_i) + F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}^c);\mathbf{W}) - F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}_i);\mathbf{W})]_+, \tag{5}$$

where γ_k is a decreasing function of k and $r_\Delta(\boldsymbol{x}_i,\boldsymbol{y}_i) = \sum_{c\in\mathcal{C}^s}\mathbb{I}(\Delta(\boldsymbol{y}^c,\boldsymbol{y}_i) + F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}^c);\mathbf{W}) - F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}_i);\mathbf{W}) > 0)$, and SJE uses a formulation based on structured SVM [36]

$$[\max_{c\in\mathcal{C}^s}(\Delta(\boldsymbol{y}^c,\boldsymbol{y}_i) + F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}^c);\mathbf{W})) - F(\phi(\boldsymbol{x}_i),\varphi(\boldsymbol{y}_i);\mathbf{W})]_+. \tag{6}$$

ESZSL adds the regularization terms, $\gamma||\mathbf{W}\varphi(\boldsymbol{y})||^2_{Fro} + \lambda||\phi(\boldsymbol{x})^T\mathbf{W}||^2_{Fro} + \beta||\mathbf{W}||^2_{Fro}$, to the empirical loss term, aiming at making the compatibility function more generalized.

Another popular formulation of the compatibility function is

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c);\mathbf{W}) = ||\varphi(\boldsymbol{y}^c) - h(\phi(\boldsymbol{x});\mathbf{W})||^2_2 \tag{7}$$

where $h(\cdot)$ is a function of $\phi(\boldsymbol{x})$ and parameterizing on $\mathbf{W}$. A typical example is the Direct Attribute Prediction (DAP), in which the function $h(\cdot)$ can be any regressors if elements in $\varphi(\boldsymbol{y}^c)$ are continuous, and can be any classifiers if elements in $\varphi(\boldsymbol{y}^c)$ are binary. The Cross Modal Transfer (CMT) model [34] uses a formulation of $h(\phi(\boldsymbol{x})) = \mathbf{W}_1 tanh(\mathbf{W}_2\phi(\boldsymbol{x}))$, where $(\mathbf{W}_1,\mathbf{W}_2)$ are the weights of the two layers in a neural network. The Metric Learning for Zero-Shot Classification (MLZSC) [6] uses a formulation of $h(\phi(\boldsymbol{x})) = max(0,\phi(\boldsymbol{x})^T\mathbf{W}_x + \boldsymbol{b}_x)$, and also incorporates a metric learning term and a regularization term in the objective function.

The Convex Semantic Embedding (CONSE) model uses a cosine similarity as the compatibility score, namely,

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c)) = cos(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c)). \tag{8}$$

The Semantic Similarity Embedding (SSE) [43] uses a comparability function

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c)) = \phi(\boldsymbol{x})^T\varphi(\boldsymbol{y}^c). \tag{9}$$

Although the above two models' compatibility functions are simple, their true contributions lie in the formulation of the embedding functions. The Latent Embeddings (LATEM) [39] uses a piece-wise linear compatibility function:

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c),\mathbf{W}_i) = \max_{1\leq i\leq K}\phi(\boldsymbol{x})^T\mathbf{W}_i\varphi(\boldsymbol{y}^c), \tag{10}$$

and uses the ranking SVM [17] for the loss function formulation. The Synthesized Classifier for Zero-Shot Learning (SYNC) model uses a compatibility function of the form:

$$F(\phi(\boldsymbol{x}),\varphi(\boldsymbol{y}^c)) = \boldsymbol{w}_z^T\phi(\boldsymbol{x}) \tag{11}$$

where $\boldsymbol{w}_z$ satisfies $\boldsymbol{w}_z = \sum_{r=1}^{R} s_{cr}\boldsymbol{v}_r$, in which $\boldsymbol{v}_r$ is the classifier obtained by solving the Crammer-Singer multi-class SVM loss function [8]. The Joint Latent Similarity Embedding (JLSE) model [44] uses the probabilistic formulation of the compatibility function:

$$F(\phi(\boldsymbol{x}), \varphi(\boldsymbol{y}^c)) = \log p(\phi(\boldsymbol{x})|\boldsymbol{x}) + \log p(\varphi(\boldsymbol{y}^c)|\boldsymbol{y}^c) + \log p(\Delta(z,c)|\phi(\boldsymbol{x}), \varphi(\boldsymbol{y}^c)) \quad (12)$$

The compatibility function of the proposed model belongs the bilinear category. However, a softmax function is applied upon the bilinear term, making the compatibility function nonlinear and thus enhancing the capacity the model. To make the softmax-based bilinear model be generalizable to zero-shot learning, we proposed to regularize the output of the softmax function; detailed discussions can be found in Section 3.

2.3. Embedding functions

The embedded functions, $\phi(\cdot)$ and $\varphi(\cdot)$, map the features and semantic representations to their embedded spaces (the embedded spaces can be the same) for the purpose of exploring the latent structures of the feature space and the semantic space such that the compatibility function is more generalized across data with seen class labels and data with unseen class labels.

The Attribute Label Embedding (ALE) [1] model learns the embedded representation of the semantic vector by adding the regularization term, $\frac{\mu}{2}||\varphi(\boldsymbol{y}^c) - \boldsymbol{y}^c||$, to the loss function (5). The Structured Joint Embedding (SJE) model [2] learns a convex combination of products, $\sum_{k=1}^{K} \alpha_k \mathbf{W}_k \varphi_k(\boldsymbol{y}^c)$ $s.t. \sum_k \alpha_k = 1$, to replace the product, $\mathbf{W}\varphi(\boldsymbol{y}^c)$, in (3), when multiple views of the semantic vector are available; the weight matrices, $\{\mathbf{W}_k\}_{k=1}^{K}$, are solved independently using (6) in each view. The Convex Semantic Embedding (ConSE) model [26] learns the embedding using $\phi(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t \varphi(\boldsymbol{y}^{c^t})$, where c^t is the t-th ranked likely label (e.g. c^1 is the most likely label) of $\boldsymbol{x}$, and $\alpha_t = p(c^t|\boldsymbol{x}) / \sum_{t=1}^{T} p(c^t|\boldsymbol{x})$. In the Semantic Similarity Embedding (SSE) model [43], embedding functions, $\phi(\cdot)$ and $\varphi(\cdot)$, map the feature vectors (i.e. $\boldsymbol{x}$'s) and the semantic vectors (i.e. $\boldsymbol{y}^c$'s) into a common space, such that $\phi(\boldsymbol{x})^T \varphi(\boldsymbol{y}^c)$ is valid; the mapping $\phi(\cdot)$ is learned by class dependent transformation, and the mapping $\varphi(\cdot)$ is learned by sparse coding.

Although embeddings can potentially improve the model performance in zero-shot learning, it increases the model complexity by introducing additional embedding function parameters which are learned jointly or separately with the compatibility function parameters. In the conducted experiments, without embedding, the proposed model showed competitive performance in comparison to existing models which utilize sophisticated embeddings.

Notation	Definition
N	Number of training examples
$\mathcal{C}^s, \mathcal{C}^u$	Seen label set and unseen label set; $\mathcal{C}^s \bigcup \mathcal{C}^u = \mathcal{C}, \mathcal{C}^s \bigcap \mathcal{C}^u = \emptyset$
d	Dimensionality of the feature space
$\{\boldsymbol{x}_i\}_{i=1}^{N}$	Feature vectors of training examples
$\{z_i\}_{i=1}^{N}$	Labels of training examples, i.e. $z_i \in \{c_1, c_2, \cdots, c_{\|\mathcal{C}^s\|}\}$
$\boldsymbol{y}^c$	Semantic vector of label c
$\mathbf{Y}$	A matrix whose c-th column, $\mathbf{Y}_{:c} = \boldsymbol{y}^c$
$\mathbf{W}, \boldsymbol{w}$	Model parameter and its vectorized form
$\boldsymbol{p}_i$	Softmax vector of the i-th example, i.e., $\boldsymbol{p}_i = \sigma(\mathbf{Y}^T\mathbf{W}^T\boldsymbol{x}_i)$
$\boldsymbol{q}_i$	One-hot encoding of the label, z_i

Table 1: Notations used in our method and derivations.

3. Proposed Model

Notations and symbols used in our model are summarized in Table 1.

3.1. Compatibility function

In our zero-shot learning model, we propose a softmax-based bilinear compatibility function:

$$F(\phi(\boldsymbol{x}), \varphi(\boldsymbol{y}^c); W) = \sigma(\varphi(\mathbf{Y})^T \mathbf{W}^T \phi(\boldsymbol{x}))_c, \quad (13)$$

where $\mathbf{Y} = [\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots \boldsymbol{y}^{|\mathcal{C}|}]$ is a matrix whose columns are the semantic vectors of the label set $\mathcal{C}$, $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{C}|}$ is the softmax function, and we let $\varphi(\mathbf{Y})$ denote $[\varphi(\boldsymbol{y}^1), \varphi(\boldsymbol{y}^2), \cdots, \varphi(\boldsymbol{y}^{|\mathcal{C}|})]$. Note that $\sigma(\cdot)_c$ is the c-th component of the vector $\sigma(\cdot)$. Comparing to the bilinear compatibility function [1, 13, 2, 31], the additional softmax layer makes the compatibility function nonlinear and thus improves the capacity of the model.

3.2. Objective function

3.2.1 Empirical risk

The softmax function has been widely used in multi-class logistic regression and the output layer of artificial neural networks. Usually the negative cross entropy is used to model the empirical risk. If we denote

$$\boldsymbol{p} = \sigma(\varphi(\mathbf{Y})^T \mathbf{W}^T \phi(\boldsymbol{x})), \quad (14)$$

and use $\boldsymbol{q}$ to denote the one-hot encoding of the corresponding label of $\boldsymbol{x}$, then the negative cross entropy is

$$-H(\boldsymbol{q}, \boldsymbol{p}) = -\boldsymbol{q}^T log(\boldsymbol{p}) \quad (15)$$

.

In our model, we use a cross-entropy-inspired term to model the empirical risk. Specifically, we define the empirical risk as

$$e(\boldsymbol{q}, \boldsymbol{p}) = -\boldsymbol{q}^T \boldsymbol{p}. \tag{16}$$

This formulation gives us a better interpretation when we introduce our regularizations.

3.2.2 Regularization

Regularization is particularly important for parameter optimization in a zero-shot learning model because model generalization is the key to achieve strong performance in unseen data.

In our model, we use two regularization terms. We use the squared Frobenius norm, $||\mathbf{W}||_{Fro}^2$, as the first regularization term to regularize the parameter to prevent overfitting; we use $||\boldsymbol{p}||_2^2$ to encourage the elements in $\boldsymbol{p}$ to be equal [1] to enhance the generalization of the model. The balance of the empirical risk and the regularizations is controlled by trade-off coefficients.

Please note that, if the objective function consists only the empirical risk term and the first regularization term, $||\mathbf{W}||_{Fro}^2$, the parameter is optimized for multi-class classification, and the model is for recognizing examples in the seen classes, so a "peak" in $\boldsymbol{p}$ is desirable for promoting a large margin between the matched class and unmatched classes. However, in zero-shot learning, if the test data contain examples in both the seen class and unseen classes, adding the second regularization term, $||\boldsymbol{p}||_2^2$, which is smoothening the "peak" in $\boldsymbol{p}$, helps prevent the inference dominations of seen class (i.e., all the examples are classified to the seen classes); if the test data contain examples of only the unseen classes, the second regularization term helps prevent the situation that all the example are classified to the unseen classes that closest to the seen classes.

Given N triplets, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, z_i)\}_{i=1}^N$ with $z_i \in \mathcal{C}^s$, we arrive at the following regularized empirical risk minimization problem:

$$\underset{\mathbf{W}}{\text{minimize}}\ J(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N}(-\boldsymbol{q}_i^T \boldsymbol{p}_i + \frac{\alpha}{2}||\boldsymbol{p}_i||_2^2) + \frac{\beta}{2}||\mathbf{W}||_{Fro}^2, \tag{17}$$

where $\boldsymbol{p}_i = \sigma(\varphi(\mathbf{Y})^T \mathbf{W}^T \phi(\boldsymbol{x}_i))$, $\boldsymbol{q}_i$ is the one-hot encoding of the label z_i, and the parameters $\alpha \geq 0$ and $\beta \geq 0$ are the coefficients of the regularization terms. Solving (17) is equivalent to solving the following minimization problem which is more interpretable:

$$\underset{\mathbf{W}}{\text{minimize}}\ J(\mathbf{W}) = \frac{1}{2N}\sum_{i=1}^{N}||\boldsymbol{p}_i - \frac{1}{\alpha}\boldsymbol{q}_i||_2^2 + \frac{\lambda}{2}||\mathbf{W}||_{Fro}^2, \tag{18}$$

where $\lambda = \beta/\alpha$. The first term of (18) is the cumulative squared 2-norm of the difference between $\boldsymbol{p}_i$ and $\frac{1}{\alpha}\boldsymbol{q}_i$, for $i = 1, 2, \cdots, N$. Please note that, when $\alpha \to 0$, the first term encourages the compatibility score of the matched feature-semantic pair as high as possible, and thus, the resultant model is a multi-class classifier of the seen classes; when $\alpha \to \infty$, the first term encourages the components in $\boldsymbol{p}_i$ are equal.

When $\alpha = 1$, we arrive at the following more interpretable objective function, (19), in which, the first term models the mean squared difference between $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$

$$\underset{\mathbf{W}}{\text{minimize}}\ J(\mathbf{W}) = \frac{1}{2N}\sum_{i=1}^{N}||\boldsymbol{p}_i - \boldsymbol{q}_i||_2^2 + \frac{\lambda}{2}||\mathbf{W}||_{Fro}^2. \tag{19}$$

For simplicity, we kept $\alpha = 1$ in our experiments, and tuned λ via cross-validation.

3.3. Optimization

If there are no embedding functions being applied to the feature vectors and semantic vectors, we can simplify $\boldsymbol{p}_i$ as:

$$\boldsymbol{p}_i = \sigma(\mathbf{Y}^T \mathbf{W}^T \boldsymbol{x}_i), \tag{20}$$

and the c-th component of $\boldsymbol{p}_i$ is

$$p_{ic} = \frac{exp(\boldsymbol{w}^T(\boldsymbol{x}_i \otimes \boldsymbol{y}^c))}{\sum_{c' \in \mathcal{C}^s} exp(\boldsymbol{w}^T(\boldsymbol{x}_i \otimes \boldsymbol{y}^{c'}))}, \tag{21}$$

where $\otimes$ is the Kronecker product operator, and we denote $\text{vec}(\mathbf{W})$ as $\boldsymbol{w}$. Instead of solving the minimization problem over the matrix, $\mathbf{W}$, we solve the problem over its vectorized version, $\boldsymbol{w}$, and thus, we can rewrite the objective function (19) as the following:

$$J(\boldsymbol{w}) = \frac{1}{2N}\sum_{i=1}^{N}||\boldsymbol{p}_i - \boldsymbol{q}_i||_2^2 + \frac{\lambda}{2}||\boldsymbol{w}||_2^2, \tag{22}$$

noting that $||\mathbf{W}||_{Fro} = ||\text{vec}(\mathbf{W})||_2$. This objective function is smooth and unconstrained, and its gradient with respect to $\boldsymbol{w}$ is

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{n}(\nabla_{\boldsymbol{w}}\boldsymbol{p}_i)(\boldsymbol{p}_i - \boldsymbol{q}_i) + \lambda\boldsymbol{w}, \tag{23}$$

where $\nabla_{\boldsymbol{w}}\boldsymbol{p}_i$ is a matrix, whose c-th column is

$$\begin{aligned}[\nabla_{\boldsymbol{w}}\boldsymbol{p}_i]_{:c} &= \nabla_{\boldsymbol{w}} p_{ic} \\ &= p_{ic}(\boldsymbol{x}_i \otimes \boldsymbol{y}^t - \sum\nolimits_{c' \in \mathcal{C}^s} p_{ic'}\boldsymbol{x}_i \otimes \boldsymbol{y}^c).\end{aligned} \tag{24}$$

[1] Because $\sum_c p_c = 1$ and $||\boldsymbol{p}||_2^2 = \sum_c p_c^2$, when $||\boldsymbol{p}||_2^2$ is minimized, $p_1 = p_2 = \cdots p_c = \cdots$.

With the analytic expression of the gradient of the objective function, we can use any first order optimization algorithm to solve the minimization problem.

3.4. Classification decision rule

When the optimized parameter, $\mathbf{W}^*$ is obtained, the label, z of an example, $\boldsymbol{x}$, is determined by:

$$z = \arg\max_{c \in \mathcal{C}} \boldsymbol{x}^T \mathbf{W}^* \boldsymbol{y}^c. \quad (25)$$

4. Experiments

4.1. Implementation

4.1.1 Parameter tuning by cross validation

We used cross-validation to tune the regularization coefficient, λ in (19). Precisely, we randomly picked two seen classes from the training data for validation and trained the model with different parameter values using the data of the rest of the seen classes. We repeated the above training-validation process 10 times and chose the parameter value with the highest average validation accuracy. Particularly, the values of λ were chosen from $\{1, 10, 100, 1000\}$.

4.1.2 Optimization

We used the function, *fminunc* in MATLAB, for optimization. We found that the *quasi-newton* was the best (in terms of speed) for the optimization algorithm option, and we found that 30 iterations could return a good solution. All other options were set to be the default values.

4.2. Datasets

In order to evaluate the performance of our proposed zero-shot learning model, we have conducted experiments on four standard benchmark datasets: *Animal with Attributes* (AwA) [20], *aPascal and aYahoo* (aPY) [12], *Caltech-UCSD Birds-200-2011* (CUB) [38] and *SUN* [29]. We downloaded the datasets from the supplementary website of [43]. The features of the examples were extracted from the VGG (Visual Geometry Group) very-deep convolution neural network [33]. The baseline models were trained by the same VGG features unless otherwise specified; the splits of training and testing (seen/unseen classes) were the same across all the methods to ensure fair comparisons. The summary of the four datasets are shown in Table 2.

4.3. Experiment 1: Comparisons to baseline models

In this experiment, we compared the classification accuracies of the proposed and a few state-of-the-art baseline models on the standard seen/unseen splits of the four datasets. The classification accuracies of the baseline methods were obtained from their original published papers or from published papers that used them as baselines. The accuracies (ranks) of all models are shown in Table 3.

From the table, we observe that none of the models achieved the highest rank in all the datasets. This may suggest that different models have different advantages, and these advantages are more applicable in one dataset than another. Our model achieved rank 1 out of 9 models in the AwA dataset and rank 1 out of 7 models in the SUN dataset; it ranked 2 out of 8 models in the CUB dataset and ranked 4 out of 6 in the aPY dataset. Overall, our proposed model achieved the highest average rank of all the datasets.

4.4. Experiment 2: Model performance under different training set sizes

In *Experiment 1*, the zero-shot learning models were trained by utilizing all the examples in the training data. In this experiment, we evaluated the accuracies of our proposed model when it was trained by subsets of the training data with different sizes. The procedure was as following:

Step 1: a certain percentage of examples were selected completely at random from each seen class to form a sub-sampled training set.

Step 2: trained the model using the sub-sampled training set created in Step 1.

Step 3: classified the examples in the unseen classes using the trained model in Step 2, and recorded the accuracy.

The sub-sampling percentages were chosen from $\{20\%, 40\%, 60\%, 80\%\}$. We repeated the above procedure 5 times for each choice of percentage and computed the mean and standard deviation of the classification accuracies.

For comparison, the ESZSL model [31] was also trained and evaluated by the above procedure. Please note that, in this experiment, we only used the ESZSL model as a baseline because 1) the ESZSL model was the closest to the proposed model and it was also claimed that it's simple and effective, 2) except the ESZSL, the baseline models required substantial training time, while the above procedure needed to repeat multiple times under different settings, only the ESZSL model allowed that the experiment could be finished in a reasonable time. The plots of the (accuracy $\pm$ standard

Dataset	# examples	# seen/unseen classes
AwA	30,475	40/10
aPY	15,339	20/12
CUB	11,788	150/50
SUN	14,340	707/10

Table 2: Summaries of the four benchmark datasets

Methods	AwA	aPY	CUB	SUN	Average Rank
DAP [23]	57.23 (9)	38.16 (6)	-	72.00 (6)	7.00
ESZSL♯ [31]	61.99 (7)	40.58 (5)	44.97 (3)	84.00 (2)	4.25
SJE♮ [2]	61.90 (8)	-	40.30 (6)	-	7.00
SSE-INT [43]	71.52 (6)	44.15 (3)	30.19 (8)	82.17 (5)	5.50
SSE-ReLU [43]	76.33 (3)	46.23 (2)	30.41 (7)	82.50 (4)	4.00
SYNC [7]	72.90 (5)	-	54.50 (1)	62.80 (7)	4.33
KDICA [15]	73.80 (4)	-	43.70 (4)	-	4.00
JLSE [44]	80.46 (2)	50.35 (1)	42.11 (5)	83.83 (3)	3.67
Ours	84.50 (1)	42.40 (4)	48.10 (2)	85.50 (1)	**2.00**

Table 3: Accuracies (ranks) of our approach and eight state-of-the-art methods on AwA, aPY, CUB, and SUN datasets. '-': accuracy was not reported in the original paper or in any published papers; ♮: features were extracted by AlexNet [21]; ♯: accuracies were obtained by our own implementation of the method.

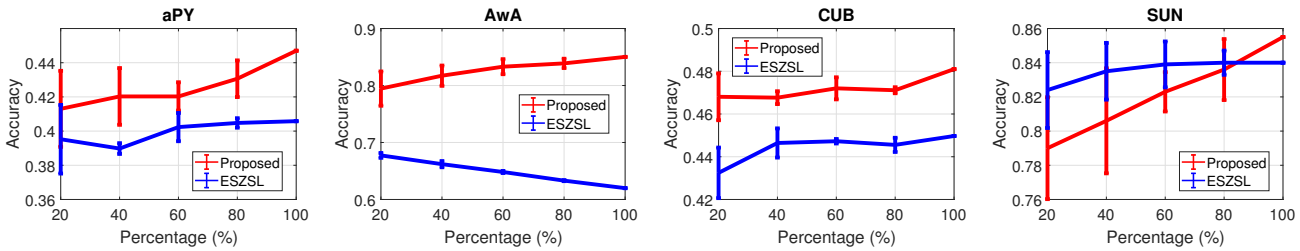

Figure 2: Zero-shot recognition accuracies($\pm$ standard deviation) of the proposed model and ESZSL model as they were trained by different training set sizes (in terms of percents of the full training data size).

deviation) V.S. percentage of both the models are shown in Figure 2.

In all the datasets, as the size of training set increased, the accuracy of our model increased. We found that, in the aPY, AwA and CUB datasets, by using 20% of the training data, our model achieved over 95% of the accuracy when the full size of the training data was used. In the SUN dataset, by using 20% of the training data, our model achieved over 90% of the accuracy when full size of the training data was used. Our model was more sensitive to the training set size in the SUN dataset; the reason might be because the number of classes in the SUN dataset was large (No. of seen class = 701 and No. of unseen class = 10), and the number of examples per class was small (20 examples per class). When 20% of the training data was used for training, there were only 4 examples in each seen class. Therefore, as the size of the training data increased, the accuracy of our model increased more significantly. In contrast, ESZSL was relatively insensitive to the size of the training set, but its accuracies were always worse than the accuracies of our model. This might suggest that the capacity of the ESZSL model is low, such that it can be fully trained by a very small amount of data. However, this simple model may not be capable to fully make use of the information provided in the training set, and thus had worse performance. We would like to point out that the variances of the accuracies in the aPY and SUN were large which may be due to the noise in the features, and this is consistent to the findings in [11].

4.5. Experiment 3: Generalized zero-shot recognition

In this experiment, we used the same procedure described in *Experiment 2* to train the models; however, the models were evaluated by their classification accuracies over the union of the remaining seen class examples (not used in training) and the unseen class examples. This was a *generalized* zero-shot recognition task [4], which was considered to be more realistic and much more challenging. Due to the same reasons mentioned in Experiment 2, only the ESZSL model was used for comparison. The generalized zero-shot recognition accuracies of both the models are shown in Figure 3.

We found that our model consistently outperformed ESZSL in a large margin. Specifically, in the aPY dataset, our model had an average 25% improvement over ESZSL; in the AwA dataset, our model had an average 50% improvement over ESZSL; in the CUB dataset, our model had an average 60% improvement over ESZSL; in the SUN dataset,

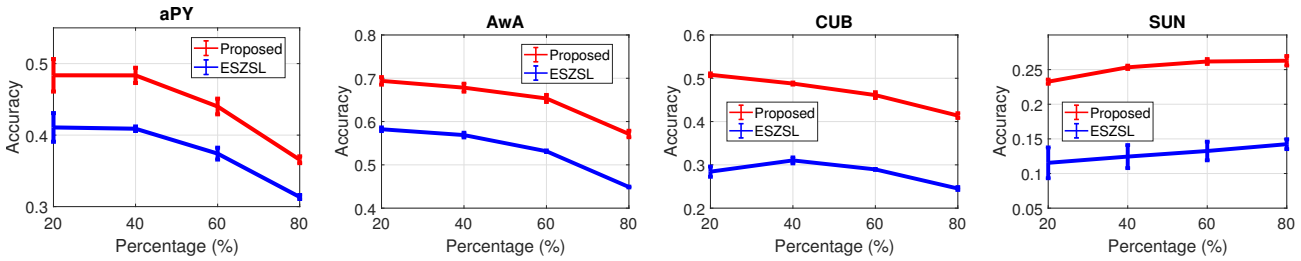

Figure 3: Generalized zero-shot recognition accuracies($\pm$ standard deviation) of the proposed model and ESZSL model as they were trained by different training set sizes (in terms of percents of the full training data size).

our model had an average 100% improvement over ESZSL. The accuracies of both models decreased as the size of train data increased. This was because the seen class examples were easier to be recognized by the models, as the size of training data increased, the number of seen class examples in the testing set decreased, and thus the generalized recognition accuracies decreased. The generalized recognition accuracies of both the models in the SUN dataset increase because of the significant imbalance of seen classes and unseen classes, and thus, the seen class examples were still dominant in the testing set.

5. Conclusion

In this paper, we propose a simple yet effective model for zero-shot learning. Similar to many state-of-the-art models, we adopt the compatibility learning based approach in our model. We find that a simple softmax-based bilinear compatibility function formulation can effectively capture the information in data. More importantly, we introduce a regularization term which is specifically for the outputs of a softmax function, and such a regularization term tremendously improve the generalization of the model. By solving a regularized empirical risk minimization problem, we obtain a generalized model for zero-shot recognition tasks. The performance of our model was compared to 8 other baseline models in four benchmark datasets. Particularly, our model achieved the highest average accuracy ranking. Additional experiments also showed that our model remained effective when the training data size was small. In the generalized zero-shot learning tasks, the performance of our model was significantly better than another competitive baseline model.

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.

[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.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[5] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, pages 663–676. Springer, 2010.

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

[7] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[8] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[11] S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto. Zero shot learning via multi-scale manifold regularization. 2017.

[12] A. Farhadi, I. Endres, D. Hoiem, 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.

[13] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embed-

ding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015.

[15] C. Gan, T. Yang, and B. Gong. Learning attributes equals multi-source domain generalization. *arXiv preprint arXiv:1605.00743*, 2016.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[18] N. Karessli, Z. Akata, A. Bulling, and B. Schiele. Gaze embeddings for zero-shot image classification. *arXiv preprint arXiv:1611.09309*, 2016.

[19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[20] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *University of Toronto, Technical Report*, 2009.

[21] 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.

[22] C. H. Lampert, H. Nickisch, and S. 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.

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

[24] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008.

[25] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[26] 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. *arXiv preprint arXiv:1312.5650 (ICLR)*, 2013.

[27] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009.

[28] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[29] G. Patterson, C. Xu, H. Su, and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014.

[30] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-shot action recognition with error-correcting output codes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[31] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015.

[32] 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.

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

[34] 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.

[35] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[36] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484, 2005.

[37] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th annual international conference on machine learning*, pages 1057–1064. ACM, 2009.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[39] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[40] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. *arXiv preprint arXiv:1703.04394*, 2017.

[41] X. Xu, F. Shen, Y. Yang, J. Shao, and Z. Huang. Transductive visual-semantic embedding for zero-shot learning. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 41–49. ACM, 2017.

[42] M. Ye and Y. Guo. Zero-shot classification with discriminative semantic representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[43] 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.

[44] 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.