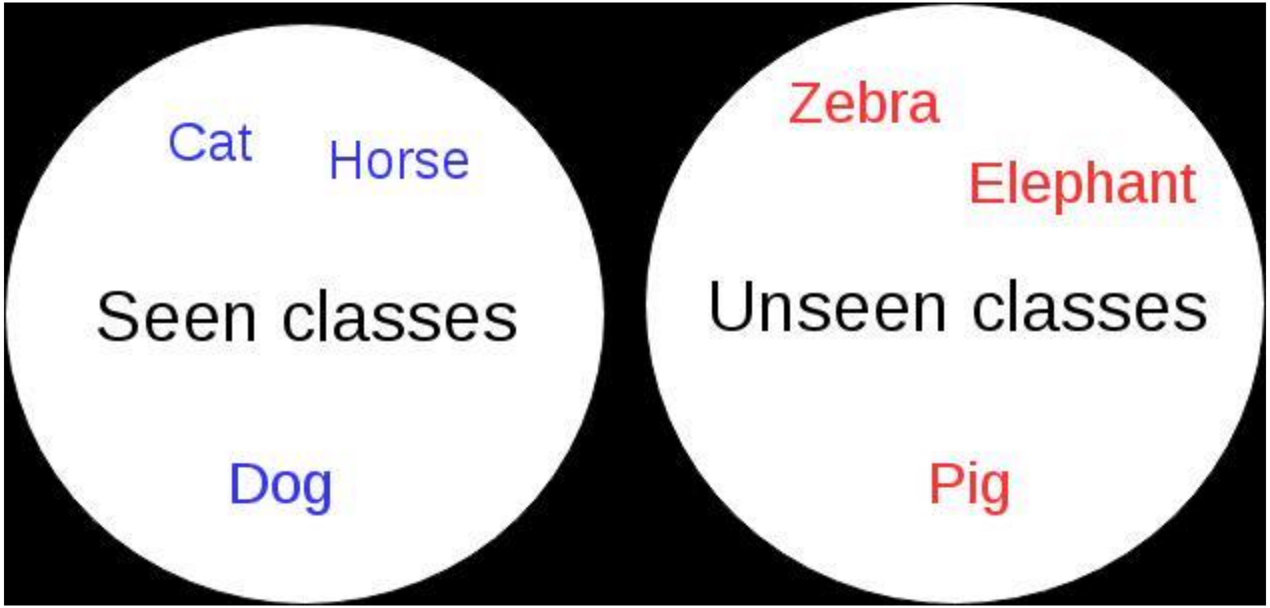
Latent Embeddings for Zero-shot Classification

Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, Bernt Schiele;
The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

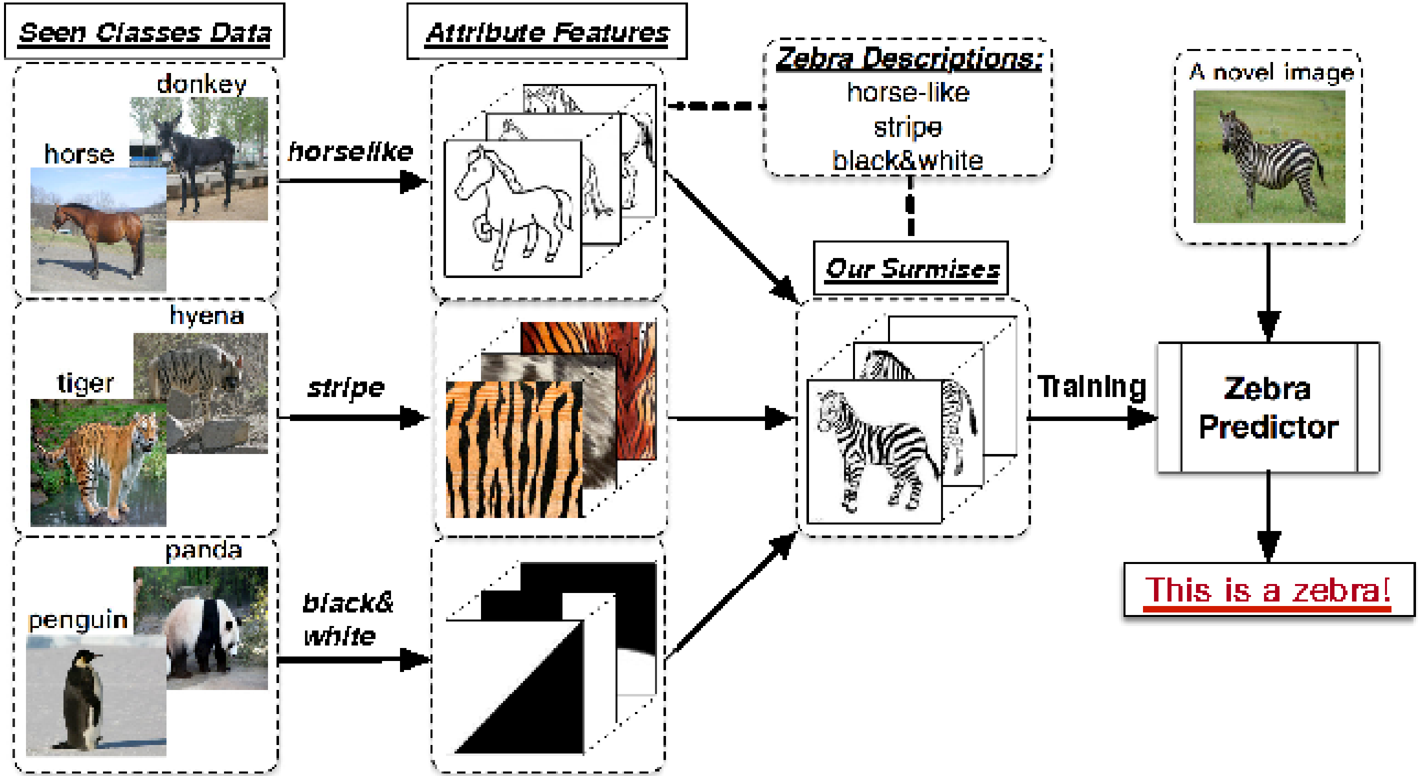
Latent Embeddings for Zero-shot Classification

- Zero-shot Learning?
  - 목표



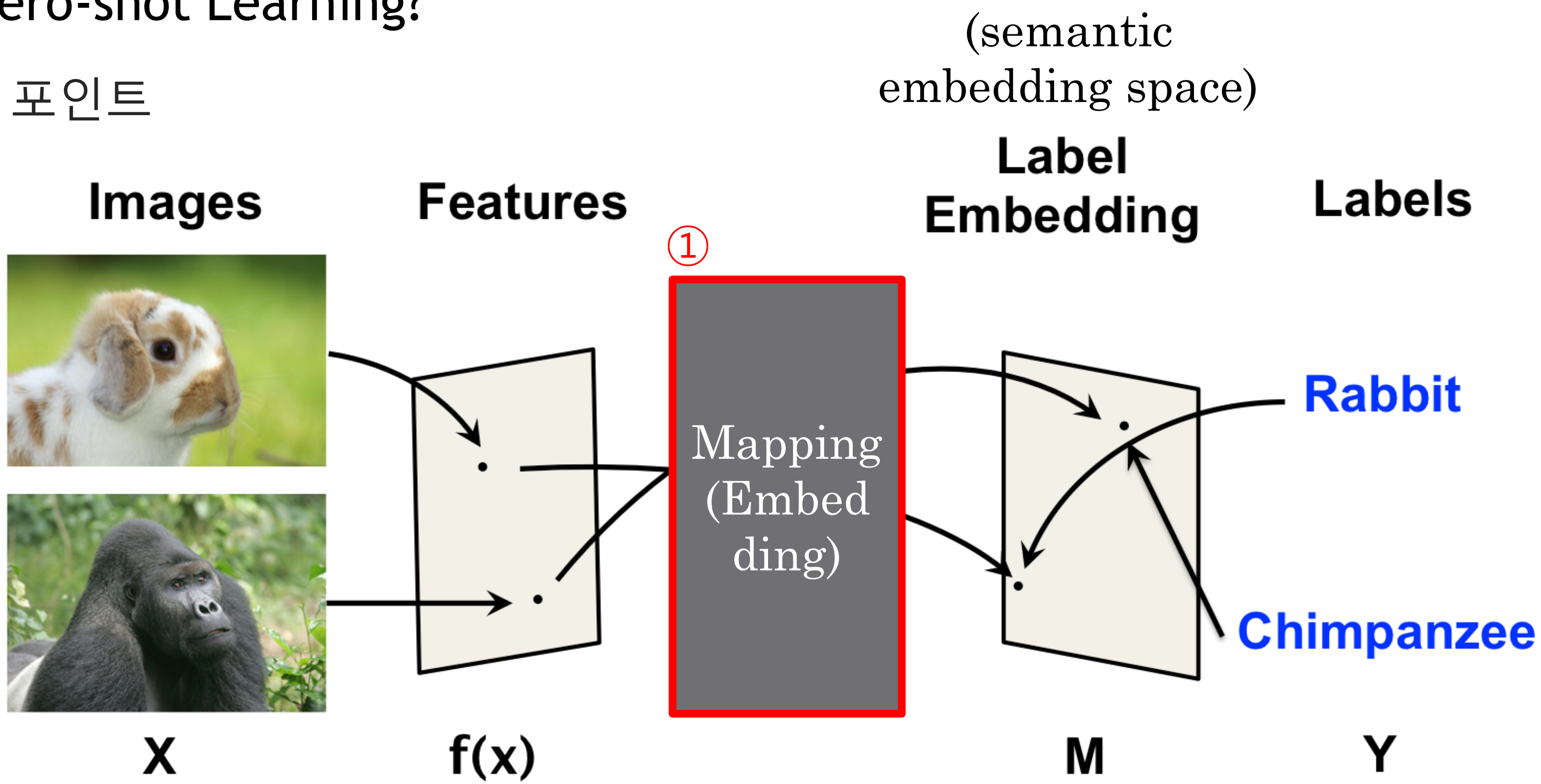
labeled -> 학습!　　　unlabeled -> 예측!

- Zero-shot Learning?
  - 개념적으로는..



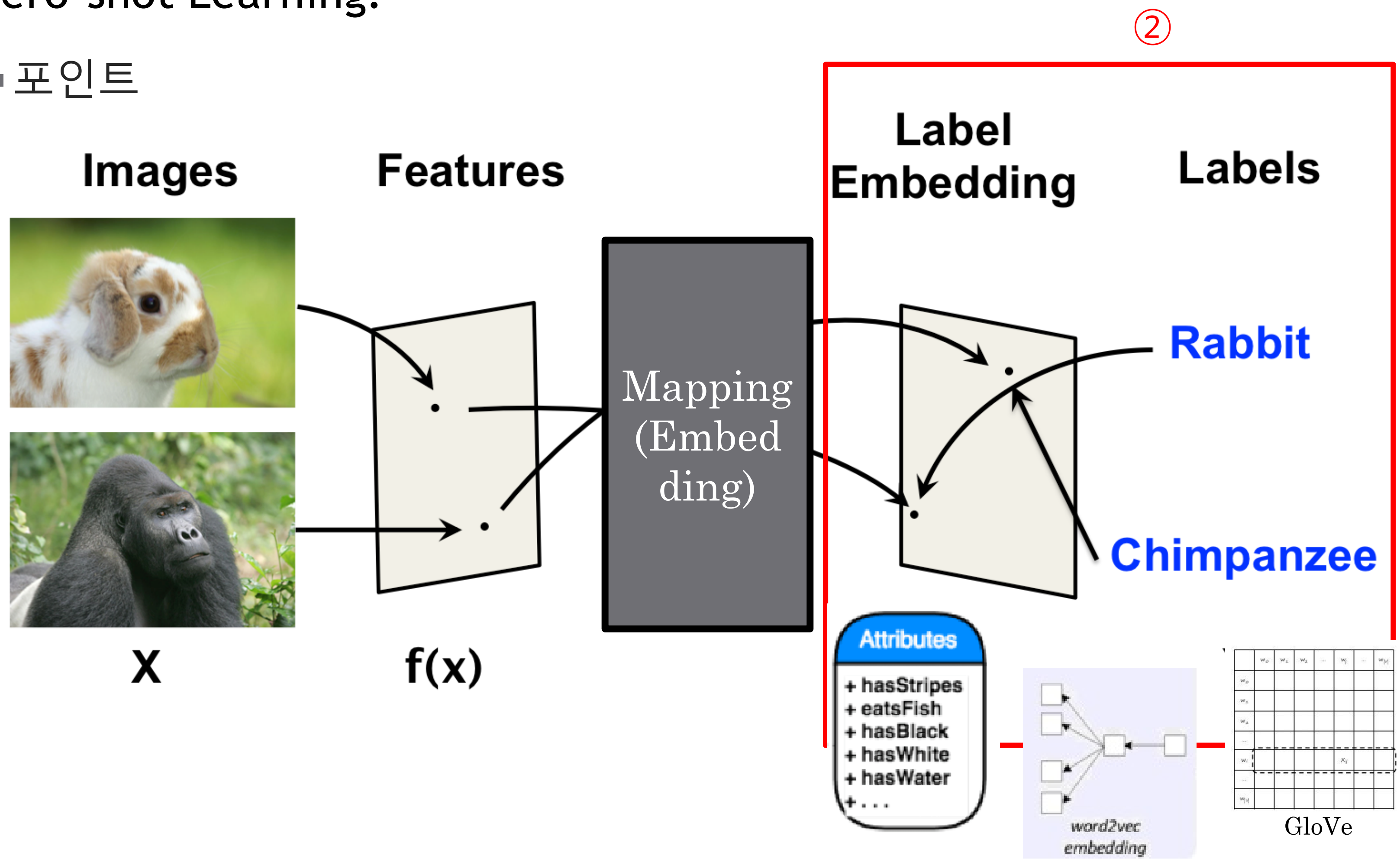
Latent Embeddings for Zero-shot Classification

- Zero-shot Learning?
  - 포인트



Latent Embeddings for Zero-shot Classification

- Zero-shot Learning?
  - 포인트



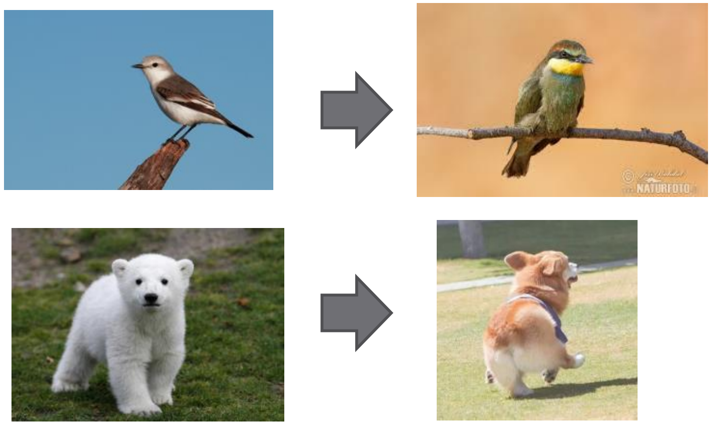
Latent Embeddings for Zero-shot Classification

- Zero-shot Learning?
  - ZSL 기본 모델의 가정(한계점?)

1. 관련성이 높은 seen 과 unseen class는 이미지에 비슷한 feature를 포함

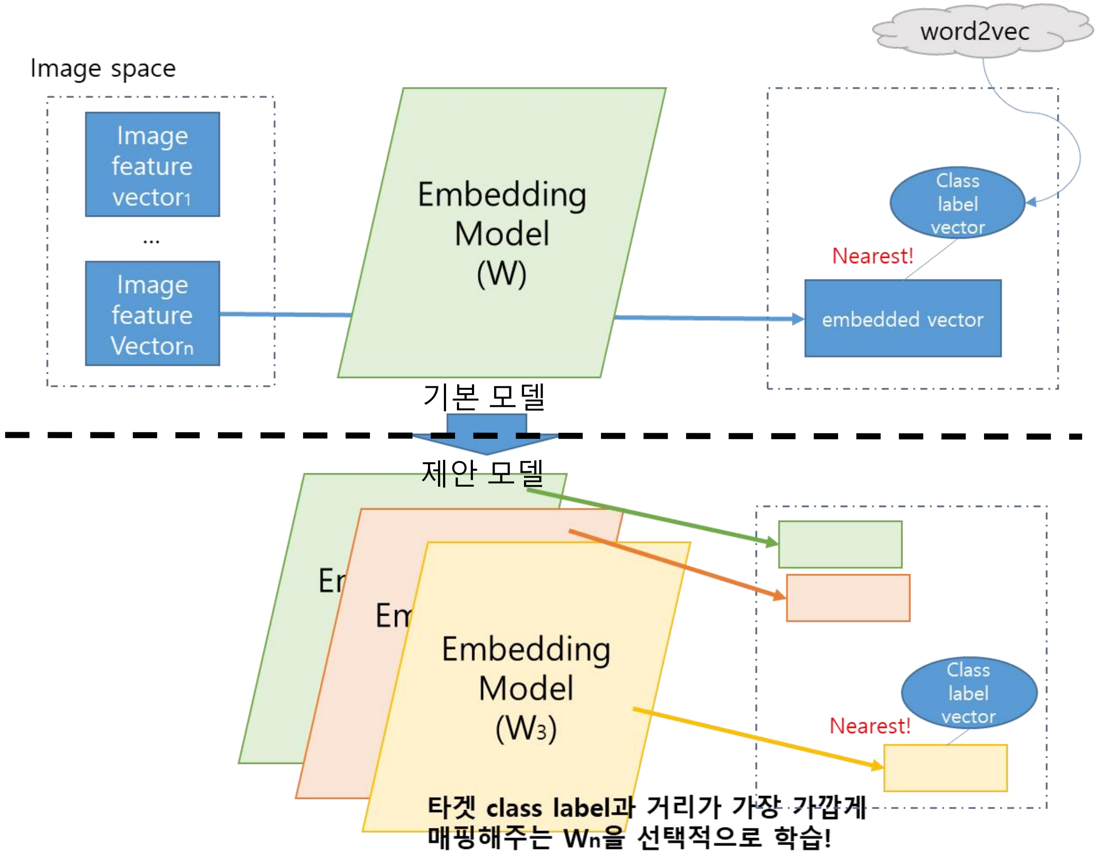
2. 그런 class들의 label은 semantic embedding space 내 비슷한 위치에 있을 것

   (이건 현재로선 어쩔수 없음)



Latent Embeddings for Zero-shot Classification

Latent Embeddings for Zero-shot Classification



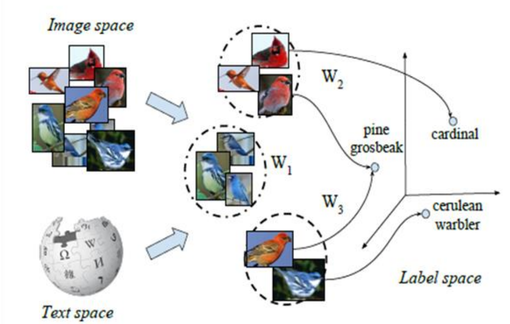


Figure 1: LatEm learns multiple W_i's that maximize the compatibility between the input embedding (image, text space) and the output embedding (label space) of all training examples. The different W_i's may capture different visual characteristics of objects, i.e. color, beak shape etc. and allow distribution of the complexity among them, enabling the model to do better classification.

Latent Embeddings for Zero-shot Classification

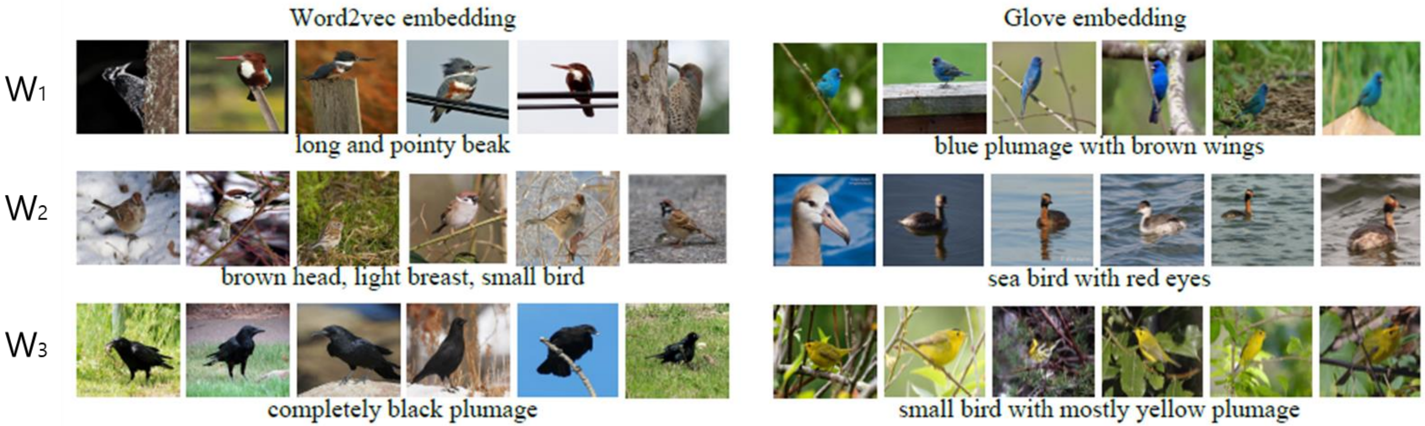


Figure 2: Top images ranked by the matrices using word2vec and glove on CUB dataset, each row corresponds to different matrix in the model. Qualitative examples support our intuition – each latent variable captures certain visual aspects of the bird. Note that, while the images may not belong to the same fine-grained class, they share common visual properties.

dom to treat different types of images differently. Let us consider a fixed class $\hat{\mathbf{y}}$ and two substantially visually different types of images $\mathbf{x}_1, \mathbf{x}_2$, e.g. the same bird flying and swimming. In SJE [2] these images will be mapped to the class embedding space with a single mapping $W^\top \mathbf{x}_1, W^\top \mathbf{x}_2$. On the other hand, LatEm will have learned two different matrices for the mapping i.e. $W_1^\top \mathbf{x}_1, W_2^\top \mathbf{x}_2$. While in the former case a single W has to map two visually, and hence numerically, very different vectors (close) to the same point, in the latent case such two different map-

Latent Embeddings for Zero-shot Classification

- Objective
  - Compatibility function F (수치값)
    - K : the number of W, W: embedding matrix
    - x : image feature vector, y : label vector

$$F(\mathbf{x}, \mathbf{y}) = \max_{1 \le i \le K} \mathbf{x}^\top W_i \mathbf{y}.$$

  - Loss function

$$L(\mathbf{x}_n, \mathbf{y}_n) = \sum_{\mathbf{y} \in \mathcal{Y}} \max\{0, \Delta(\mathbf{y}_n, \mathbf{y}) + F(\mathbf{x}_n, \mathbf{y}) - F(\mathbf{x}_n, \mathbf{y}_n)\} \quad (6)$$

where $\Delta(\mathbf{y}, \mathbf{y}_n) = 1$ if $\mathbf{y} \neq \mathbf{y}_n$ and 0 otherwise. This ranking-based loss function has been previously used in [12, 38] such that the model is trained to produce a higher compatibility between the image embedding and the class embedding of the correct label than between the image embedding and class embedding of other labels.

Latent Embeddings for Zero-shot Classification

- Optimization

Algorithm 1 SGD optimization for LatEm

$\mathcal{T} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbb{R}^{d_x}, \mathbf{y} \in \mathbb{R}^{d_y}\}$

for all $t = 1$ to T **do** epochs

  for all $n = 1$ to $|\mathcal{T}|$ **do** training set

    Draw $(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{T}$ and $\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_n\}$ 비교 대상(정답이 아닌) label 선택

    if $F(\mathbf{x}_n, \mathbf{y}) + 1 > F(\mathbf{x}_n, \mathbf{y}_n)$ **then** L(xn,yn) > 0

      $i^* \leftarrow \underset{1 \leq k \leq K}{\arg\max}\, \mathbf{x}_n^\top W_k \mathbf{y}$

      $j^* \leftarrow \underset{1 \leq k \leq K}{\arg\max}\, \mathbf{x}_n^\top W_k \mathbf{y}_n$

현재 image feature가 각 class label에
가장 큰 영향을 주는 W를 선택

      if $i^* = j^*$ **then**

        $W_{i^*}^{t+1} \leftarrow W_{i^*}^t - \eta_t \mathbf{x}_n (\mathbf{y} - \mathbf{y}_n)^\top$ 그 W가 같으면 정답은 가까워지게 정답
아닌건 멀어지게 학습

      end if

      if $i^* \neq j^*$ **then**

        $W_{i^*}^{t+1} \leftarrow W_{i^*}^t - \eta_t \mathbf{x}_n \mathbf{y}^\top$

        $W_{j^*}^{t+1} \leftarrow W_{j^*}^t + \eta_t \mathbf{x}_n \mathbf{y}_n^\top$ 그 W가 다르면 각각 학습

      end if

    end if

  end for

end for

Latent Embeddings for Zero-shot Classification

	Total		train+val		test	
	imgs	cls	imgs	cls	imgs	cls
CUB	11786	200	8855	150	2931	50
AWA	30473	50	24293	40	6180	10
Dogs	19499	113	14681	85	4818	28

Table 1: The statistics of the three datasets used. CUB and Dog are fine-grained datasets whereas AWA is a more general concept dataset.

Latent Embeddings for Zero-shot Classification

- Epochs : 150
- Learning rate : 0.1, 0.001, 0.01 (CUB, AWA, DOG, respectively)

	CUB		AWA		Dogs	
	SJE	LatEm	SJE	LatEm	SJE	LatEm
att	**50.1**	45.5	66.7	**71.9**	N/A	N/A
w2v	28.4	**31.8**	51.2	**61.1**	19.6	**22.6**
glo	24.2	**32.5**	58.8	**62.9**	17.8	**20.9**
hie	20.6	**24.2**	51.2	**57.5**	24.3	**25.2**

Table 2: Comparison of Latent Embeddings (LatEm) method with the state-of-the-art SJE [2] method. We report average per-class Top-1 accuracy on unseen classes. We use the same data partitioning, same image features and same class embeddings as SJE [2]. We cross-validate the K for LatEm.

Latent Embeddings for Zero-shot Classification

- Model Selection
  - Cross-validation
    - data split을 다르게 해서 validation 단계에서 가장 좋은 성능을 나타내는 W의 개수(K) 선택
    - K = {2, 4, 6, 8, 10} 으로 각각 돌려봄
  - Pruning
    - 성능을 유지하며 더 빠르게 학습시키게 하기 위한 선택 방법
    - 학습 5번 돌 때 5프로 미만으로 선택된 W를 제거
    - 16개로 시작해서 pruning 수행

Latent Embeddings for Zero-shot Classification

	CUB		AWA		Dogs	
	PR	CV	PR	CV	PR	CV
att	3	4	7	2	N/A	N/A
w2v	8	10	8	4	6	8
glo	6	10	7	6	9	4
hie	8	2	7	2	11	10

	CUB		AWA		Dogs	
	PR	CV	PR	CV	PR	CV
att	43.8	**45.6**	63.2	**72.5**	N/A	N/A
w2v	**33.9**	33.1	48.9	**52.3**	**25.0**	24.5
glo	**31.5**	30.7	**51.6**	50.7	18.8	**20.2**
hie	**23.8**	**23.7**	45.5	**46.2**	25.2	**25.6**

Table 5: (Left) Number of matrices selected (on the original split) and (right) average per-class top-1 accuracy on unseen classes (averaged over five splits). PR: proposed model learnt with pruning, CV: with cross validation.

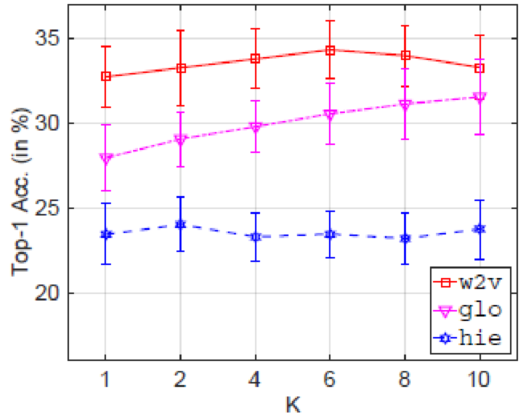


Figure 3: Effect of latent variable K, with unsupervised class embeddings (on CUB dataset with five splits).

Latent Embeddings for Zero-shot Classification

Dataset : Caltech-UCSD Birds**(CUB)**
training set (seen)
label – 100 classes
images – 5894 images

test set (unseen)
label – 50 classes
images – 2931 images

epoch : 150
learning rate : 0.1

class embedding	average per-class Top-1 accuracy(%)
attributes	**38.2**
word2vec (wikipedia)	**27.8**
glove (wikipedia)	**25.7**
hierarchies (WordNet)	**17.0**

	CUB		AWA		Dogs	
	SJE	LatEm	SJE	LatEm	SJE	LatEm
att	**50.1**	45.5	66.7	**71.9**	N/A	N/A
w2v	28.4	**31.8**	51.2	**61.1**	19.6	**22.6**
glo	24.2	**32.5**	58.8	**62.9**	17.8	**20.9**
hie	20.6	**24.2**	51.2	**57.5**	24.3	**25.2**

Table 2: Comparison of Latent Embeddings (LatEm) method with the state-of-the-art SJE [2] method. We report average per-class Top-1 accuracy on unseen classes. We use the same data partitioning, same image features and same class embeddings as SJE [2]. We cross-validate the K for LatEm.

Latent Embeddings for Zero-shot Classification