Learning Feature-to-Feature Translator by Alternating Back-Propagation for Generative Zero-Shot Learning

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

Yizhe Zhu[1] Jianwen Xie[2] Bingchen Liu[1] Ahmed Elgammal[1]

1 Department of Science, Rutgers University 2 Hikvision Research Institute

海康威视 HIKVISION

Motivation

- Zero-shot learning makes it possible to recognize novel classes without seeing any samples.
- Most generative zero-shot learning algorithms are either GAN-based or VAE-based. Both GAN and VAE models require auxiliary networks to assist the training.

Contribution

- We propose a feature-to-feature translator that maps class-level semantic features as well as Gaussian noise to visual features.
- We propose to learn the translator via alternating back-propagation (ABP) algorithm for maximum likelihood.
- We show that the proposed framework can learn from incomplete training examples where visual features are partially visible.

Our Proposed Model

- Our model is a conditional latent variable model, which can be formulated as :

$$Z \sim \mathcal{N}(0, I_d),$$
$$X = g_\theta(C, Z) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D),$$

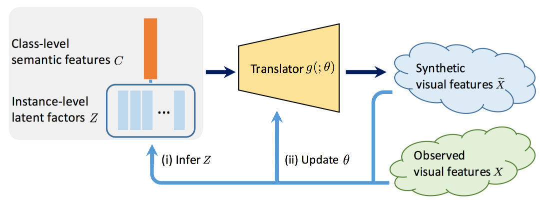
- We optimize our model by the maximum likelihood estimation (MLE). The gradient of weight θ is:

$$\frac{\partial}{\partial\theta}\log p_\theta(X|C) = \frac{1}{p_\theta(X|C)}\frac{\partial}{\partial\theta}p_\theta(X|C)$$
$$= \mathbb{E}_{Z\sim p_\theta(Z|X,C)}\left[\frac{\partial}{\partial\theta}\log p_\theta(X, Z|C)\right]$$

where the complete data model is:

$$\log p_\theta(X, Z|C) = \log[p_\theta(X|Z, C)p(Z)]$$
$$= -\frac{1}{2\sigma^2}\|X - g_\theta(C, Z)\|^2 - \frac{1}{2}\|Z\|^2 + \text{const},$$

Alternating Back-Propagation Algorithm



- We iterate two steps, where we back-propagate the gradient of either the latent variable Z or the weights θ.
- **(i) Inferential Back-Propagation:** We use Langevin dynamics sampling to compute the analytically intractable posterior distribution. We infer Z_i for each observed pair (X_i, C_i).

$$Z_{\tau+1} = Z_\tau + \frac{s^2}{2}\frac{\partial}{\partial Z}\log p_\theta(X, Z_\tau|C) + sU_\tau$$

- **(ii) Learning Back-propagation:** With inferred Z_i, we learn the model via stochastic gradient ascent .

$$\theta_{t+1} = \theta_t + \gamma_t\frac{\partial}{\partial\theta}L(\theta), \quad \text{where} \quad \frac{\partial}{\partial\theta}L(\theta) \approx \sum_{i=1}^{n}\frac{\partial}{\partial\theta}\log p_\theta(X_i, Z_i|C_i)$$

- Both gradients can be efficiently computed by back-propagation.

Learning from Incomplete Data

- Our model is able to learn from incomplete visual features via alternating back-propagation algorithm by letting latent vector only explain the visible part of the data.
- A slight change in the objective:

$$\|X - g_\theta(C, Z)\|^2 \Rightarrow \|M \circ (X - g_\theta(C, Z))\|^2$$

where M is the given binary indicator matrix with the same size of X, with 1 indicating "visible" and 0 indicating "missing".

Method	GTA	DET	DET* 30%	DET* 50%	DET* 70%	DET* 90%
GAZSL [67]	74.1	72.7	68.5	63.7	55.6	37.7
Ours	76.7	75.2	72.9	71.3	64.8	51.6

Table1: ZSL performance of the models trained on incomplete visual features with different missing ratios.

Experimental Results

	Method	CUB $A_\mathcal{U}$	CUB $A_\mathcal{S}$	CUB H	AwA1 $A_\mathcal{U}$	AwA1 $A_\mathcal{S}$	AwA1 H	AwA2 $A_\mathcal{U}$	AwA2 $A_\mathcal{S}$	AwA2 H	SUN $A_\mathcal{U}$	SUN $A_\mathcal{S}$	SUN H
§	DAP [22]	1.7	**67.9**	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.2	25.1	7.2
	DEVISE [12]	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	16.9	27.4	20.9
	CMT [38]	7.2	49.8	12.6	0.9	87.6	1.8	0.5	**90.0**	1.0	8.1	21.8	11.8
	SJE [2]	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	14.7	30.5	19.8
	LATEM [53]	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	14.7	28.8	19.5
	ESZSL [36]	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	11.0	27.9	15.8
	ALE [1]	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3
	SAE [19]	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	8.8	18.0	11.8
	DEM [62]	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1	20.5	34.3	25.6
†	VZSL [50]	44.9	54.1	49.1	53.4	68.3	59.9	51.7	67.2	58.4	43.5	34.9	38.7
	GAZSL [67]	26.5	57.4	36.2	32.8	84.7	47.3	**59.9**	68.3	53.4	21.7	34.5	26.7
	FGZSL [19]	45.9	54.6	49.9	53.1	68.0	59.6	50.2	67.5	57.5	40.2	36.4	38.2
	MCGZSL [11]	45.7	61.0	**52.3**	56.9	64.0	60.2	51.9	67.2	58.6	**49.4**	33.6	40.0
	Ours	**47.0**	54.8	50.6	**57.3**	67.1	**61.8**	55.3	72.6	**62.6**	45.3	**36.8**	**40.6**

Table2: Performance Comparison On Generalized ZSL

	Method	CUB	AwA1	AwA2	SUN
§	DAP [22]	40.0	44.1	46.1	39.9
	CMT [38]	34.6	39.5	37.9	39.9
	LATEM [53]	49.3	55.1	55.8	55.3
	ALE [1]	54.9	59.9	62.5	58.1
	DEVISE [12]	52.0	54.2	59.7	56.5
	SJE [2]	53.9	65.6	61.9	53.7
	ESZSL [36]	53.9	58.2	58.6	54.5
	SYNC [5]	55.6	54.0	46.6	56.3
	SAE [19]	33.3	53.0	54,1	40.3
	DEM [62]	51.7	65.7	66.5	60.8
	GFZSL [47]	49.3	68.3	63.8	60.6
†	VZSL [50]	56.3	67.1	66.8	59.0
	GAZSL [67]	55.8	63.7	64.2	60.1
	FGZSL [55]	57.7	65.6	66.9	58.6
	MCGZSL [11]	58.4	66.8	67.3	60.0
	Ours	**58.5**	**69.3**	**70.4**	**61.5**

Table3: Performance Comparison On ZSL

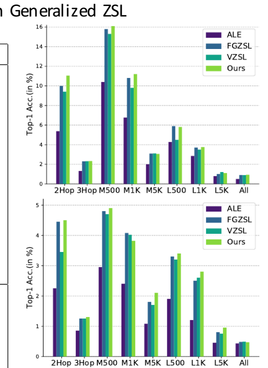


Figure2: ZSL/GZSL results on ImageNet. For GZSL, A_u is reported.

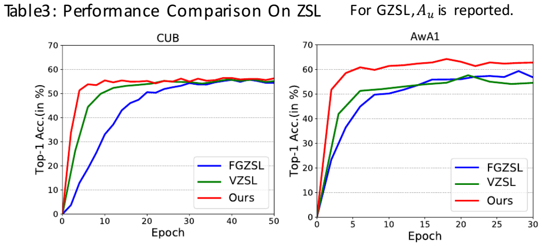


Figure4: Convergence comparison: top-1 accuracies in validation set over epochs

Dataset	# of Parameters	# of Mult-Adds
FGZSL [55]	20.62M	41.23M
VZSL [50]	21.90M	43.78M
Ours	9.71M	19.42M

Table4: Comparison on # of parameters and computational cost (CUB dataset).

Code available:
https://github.com/EthanZhu90/ZSL_ABP

Reference:

GAZSL: Zhu et al. A Generative Adversarial Approach for Zero-Shot Learning from Noisy Texts, CVPR18

FGZSL: Xian et al. Feature Generating Networks for Zero-Shot Learning, CVPR18

VZSL: Wang et al. Zero-Shot Learning via Class-Conditioned Deep Generative Models, AAAI18