Adversarial Semi-Supervised Active Learning

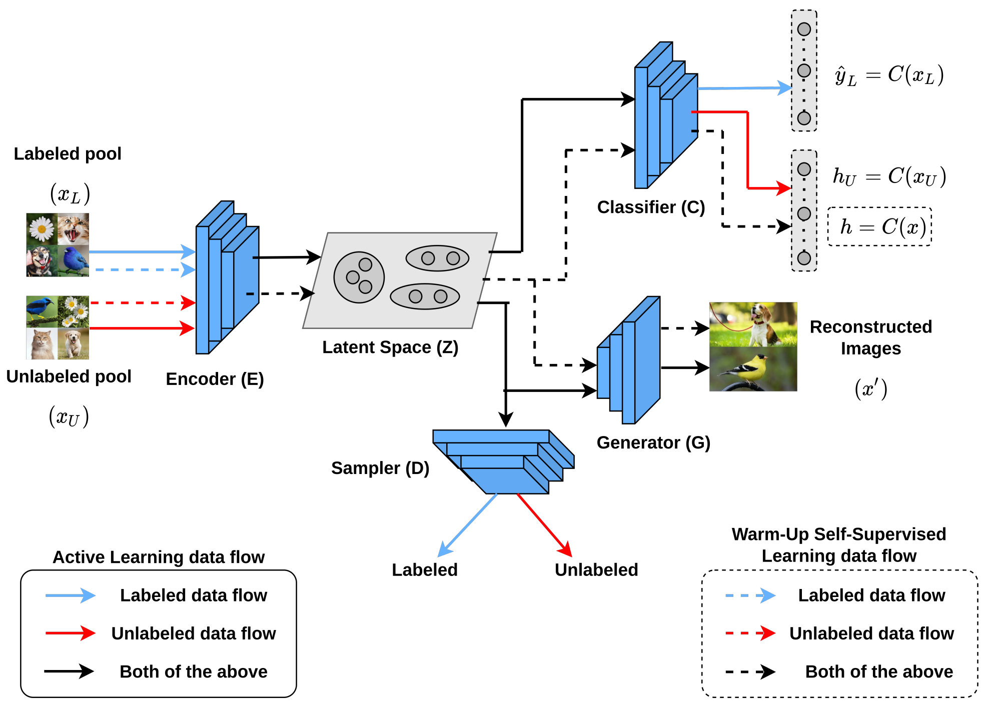

Figure 1: Proposed *Adversarial Semi-Supervised Active Learning* Framework. It consists of a unified representation learner (VAE), which consists of an encoder and a generator network, a classifier network, and a label/unlabel state sampler. The VAE network learns a unified latent representation using both labeled/unlabeled data. The classifier network embeds the annotation information into the latent representation. It further optimizes a self-supervised loss with the unlabeled data, which further embeds the uncertainty about the unlabeled data in the latent code. Finally, the sampler learns to discriminate between labeled/unlabeled samples and helps in selecting the most information samples. The warm-up step includes VAE learning and self-supervised learning using both labeled/unlabeled data, it helps in better initialization of the network parameters.

1 Objective

The existing work, such as VAAL [5], while achieves state-of-the-art performances in active learning, it has following issues:

i since it is a task-agnostic method, therefore, it does not incorporate the task-model uncertainty in the sampler.

ii it does not use the large number of available unlabeled data while optimizing the task-model.

In this work, we are interested in an adversarial semi-supervised active learning strategy with the following properties:

Preprint. Under review.

i it implicitly learns the sampling mechanism in an adversarial manner, however, unlike VAAL [5] it actually incorporates the task-model uncertainty while learning the sampler (refer Figure 1).

ii it uses the data from the both labeled and unlabeled pool in an semi-supervised manner while optimizing the task-model.

Furhtermore, we use a warm-up self-supervised learning which uses both the labeled and the unlabeled data as an initialization step for the network parameters. Finally, we explore various strategies for selecting the initial labeled samples from the unlabeled pool to avoid the cold-start.

2 Adversarial Semi-Supervised Active Learning

Let consider (x_L, y_L) be a sample pair belonging to labeled pool samples (X_L, Y_L). x_U denotes a sample belonging to a considerably larger unlabeled pool of samples X_U. The objective of the active learning is to train the task-model in a most label-efficient, such that, using minimum number of labeled samples, by iteratively querring a fixed number of samples, i.e., sampling budget b. These b samples represent the most informative samples from the unlabeled pool X_U, which are selected using an acquisition function to be annotated by an oracle such that the task-model loss is minimized, i.e., the task-model performance improves.

2.1 Unified Representation Learning

In this work, we consider learning a unified representation from both labeled/unlabeled samples. We use a variational autoencoder (VAE) [3] for representation learning in which the encoder network (E) learns a latent representation for the underlying data distribution using a Gaussian prior and the generator network (G) reconstructs the input data from the latent code. We pass both the labeled and the unlabeled through the VAE and learns the latent representation by minimizing the variational lower bound. The objective function for the VAE can be formulated as follows:

$$\mathcal{L}_{\text{VAE}}^{\text{URL}} = \mathbb{E}\left[\log p_\theta(x_L|z_L)\right] - \mathrm{D}_{\text{KL}}\left(q_\phi(z_L|x_L)\,||\,p(z)\right) + \mathbb{E}\left[\log p_\theta(x_U|z_U)\right] - \mathrm{D}_{\text{KL}}\left(q_\phi(z_U|x_U)\,||\,p(z)\right) \tag{1}$$

where q_ϕ and p_θ represents the encoder (E) and the generator network (G) parameterized by ϕ and θ resprectively. $p(z)$ is the prior chosen as the unit Gaussian.

2.2 Classifier Learning

$$\mathcal{L}_C^L = \mathbb{E}\left[\log p_\psi(y_L|z_L)\right] - \mathrm{D}_{\text{KL}}\left(q_\phi(z_L|x_L)\,||\,p(z)\right) \tag{2}$$

$$\mathcal{L}_C^U = \mathbb{E}\left[\log p_\psi(h_U|z_U)\right] - \mathrm{D}_{\text{KL}}\left(q_\phi(z_U|x_U)\,||\,p(z)\right) \tag{3}$$

where q_ϕ and p_ψ represents the encoder (E) and the classifier network (C) parameterized by ϕ and θ resprectively. $p(z)$ is the prior chosen as the unit Gaussian.

2.3 Adversarial Representation Learning

$$\mathcal{L}_{\text{VAE}}^{adv} = -\mathbb{E}\left[\log(D(q_\phi(z_L|x_L)))\right] - \mathbb{E}\left[\log(D(q_\phi(z_U|x_U)))\right] \tag{4}$$

$$\mathcal{L}_D = -\mathbb{E}\left[\log(D(q_\phi(z_L|x_L)))\right] - \mathbb{E}\left[\log(1 - D(q_\phi(z_U|x_U)))\right] \tag{5}$$

where D represents the discriminator (sampler).

2.4 Training

$$\mathcal{L}_{\text{VAE}} = \lambda_1 \mathcal{L}_{\text{VAE}}^{\text{URL}} + \lambda_2 \mathcal{L}_C^L + \lambda_3 \mathcal{L}_C^U + \lambda_4 \mathcal{L}_{\text{VAE}}^{adv} \tag{6}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyper-parameters which determines the effect of various components to learn an effective variational adversarial representation.

2.5 Warm-up Initialization

$$\mathcal{L}'_C = \mathbb{E}\left[\log p_\psi(h|z)\right] - \mathrm{D}_{\mathrm{KL}}\left(q_\phi(z|x)\,||\,p(z)\right) \tag{7}$$

where $x \sim (X_L \cup X_U)$

$$\mathcal{L}'_{\mathrm{VAE}} \leftarrow \lambda'_1 \mathcal{L}^{\mathrm{URL}}_{\mathrm{VAE}} + \lambda'_2 \mathcal{L}'_C \tag{8}$$

where λ'_1, and λ'_2 are hyper-parameters which determines the effect of various components to learn an effective initial variational representation.

2.6 Sampling Strategy

Algorithm 1 Adversarial Semi-Supervised Active Learning

Require: Labeled pool (X_L, Y_L), Unlabeled pool (X_U)
Require: Initialized model parameters: θ_{VAE}, and θ_D ▷ θ_{VAE} represents the parameters of the whole VAE network including Encoder (E), Generator (G), & Classifier (C), and θ_D represents the parameters of the sampler (D)
Require: Hyper-parameters: epochs, $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \alpha_1$, and α_2
for $e = 1$ to epochs **do**
  sample $(x_L, y_L) \sim (X_L, Y_l)$
  sample $x_U \sim X_U$
  Compute $\mathcal{L}^{\mathrm{URL}}_{\mathrm{VAE}}$ by using Eq.(1)
  Compute $\mathcal{L}^L_C$ by using Eq.(2)
  Compute $\mathcal{L}^U_C$ by using Eq.(3)
  Compute $\mathcal{L}^{adv}_{\mathrm{VAE}}$ by using Eq.(4)
  $\mathcal{L}_{\mathrm{VAE}} \leftarrow \lambda_1 \mathcal{L}^{\mathrm{URL}}_{\mathrm{VAE}} + \lambda_2 \mathcal{L}^L_C + \lambda_3 \mathcal{L}^U_C + \lambda_4 \mathcal{L}^{adv}_{\mathrm{VAE}}$
  Update VAE by descending stochastic gradients:
  $\theta'_{\mathrm{VAE}} \leftarrow \theta_{\mathrm{VAE}} - \alpha_1 \nabla \mathcal{L}_{\mathrm{VAE}}$
  Compute $\mathcal{L}_D$ by using Eq.(5)
  Update D by descending its stochastic gradients:
  $\theta'_D \leftarrow \theta_D - \alpha_2 \nabla \mathcal{L}_D$
end for
return Trained θ_{VAE}, and θ_D

Algorithm 2 Warm-Up Self-Supervised Learning

Require: X_L, and X_U
Require: Randomly initialized VAE parameters: θ_{VAE} ▷ θ_{VAE} represents the parameters of the whole VAE network including Encoder (E), Generator (G), & Classifier (C)
Require: Hyper-parameters: T, λ'_1, λ'_2, and α'_1
for $e = 1, \ldots, T$ **do**
  sample $x \sim (X_L \cup X_U)$
  Compute $\mathcal{L}^{\mathrm{URL}}_{\mathrm{VAE}}$ by using Eq.(1)
  Compute $\mathcal{L}'_C$ by using Eq.(7)
  $\mathcal{L}'_{\mathrm{VAE}} \leftarrow \lambda'_1 \mathcal{L}^{\mathrm{URL}}_{\mathrm{VAE}} + \lambda'_2 \mathcal{L}'_C$
  Update VAE by descending stochastic gradients:
  $\theta'_{\mathrm{VAE}} \leftarrow \theta_{\mathrm{VAE}} - \alpha'_1 \nabla \mathcal{L}_{\mathrm{VAE}}$
end for
return Trained θ_{VAE}

Algorithm 3 Sampling Strategy

Require: X_L, and X_U
Require: Sampling budget: b
1: Select samples (X_s) with $\min_b\{\theta_D(z_U)\}$
2: $Y_o \leftarrow \mathcal{ORACLE}(X_s)$
3: $(X_L, Y_L) \leftarrow (X_L, Y_L) \cup (X_s, Y_o)$
4: $X_U \leftarrow X_U - X_s$
5: **return** Trained X_L, and X_U

3 Relevant Existing Works

- Variational Adversarial Active Learning [5]
- Dual Generative Adversarial Active Learning [1]
- Task-Aware Adversarial Active Learning [2]
- Adversarial Representition Active Learning [4]
- State-Relabeling Adversarial Active Learning [6]

References

[1] Guo, J., Z. Pang, M. Bai, P. Xie, and Y. Chen (2021). Dual generative adversarial active learning. *Applied Intelligence*, 1–12.

[2] Kim, K., D. Park, K. I. Kim, and S. Y. Chun (2020). Task-aware variational adversarial active learning. *arXiv preprint arXiv:2002.04709*.

[3] Kingma, D. P. and M. Welling (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[4] Mottaghi, A. and S. Yeung (2019). Adversarial representation active learning. *arXiv preprint arXiv:1912.09720*.

[5] Sinha, S., S. Ebrahimi, and T. Darrell (2019). Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5972–5981.

[6] Zhang, B., L. Li, S. Yang, S. Wang, Z.-J. Zha, and Q. Huang (2020). State-relabeling adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8756–8765.