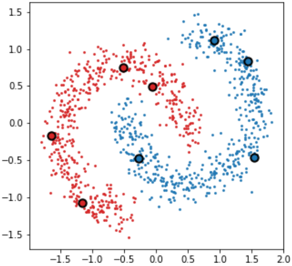
Semi-supervised learning with MixUp method - ICT

YC, Choi

2019.09.09

Semi-supervised learning

- Semi-supervised learning : Leveraging large amounts of unlabeled data to improve the performance of supervised learning.
- Cluster assumption : if two samples belong to the same cluster in the input distribution, then tehy are likely to belong to the same class.
- low-density separation assumption : the decision boundary should lie in the low-density regions.



Consistency-Regularization approach

- encouraging invariant prediction $f(u) = f(u + \delta)$ for perturbations $u + \delta$ of unlabeled data u.
- There are many consistency-regularizion techniques depending on how to choose δ.
- Random perturbation, data augmentation are kind of Consistency-Regularization methods.

Virtual adversarial training(VAT)(2018)

- VAT(Miyato et al., 2018) searches for small perturbation δ that maximize the change in the prediction of the model.
- $r_{\text{advr}}(\mathbf{u}, c) = \underset{r;||r||\leq c}{\text{argmax}} D_{\text{KL}} \left(p(\cdot|\mathbf{u}; \hat{\theta}) || p(\cdot|\mathbf{u} + r; \hat{\theta}) \right)$

Bad-GAN(2017)

- Bad-GAN uses a complement generator which generates complements samples in the feature space.
- For K classification problem, we give K+1 label to complements samples.
- Under mild assumptions, optimal discriminator learns correct decision boundary.
- The discriminator obtains class boundaries in low-density area. (cluster assumption)

Fast adversarial training(FAT)

- Idea : Generating complements samples without GAN would be computationally efficient.
- The perturbation(r_{advr}) of VAT is toward decision boundary.
- The region of decision boundary would be expected to low-density.
- We give larger value Cr_{advr} and $x + Cr_{\text{advr}}$ is considered complement sample. $C > 0$

Interpolation Consisntency Training(ICT)(2019)

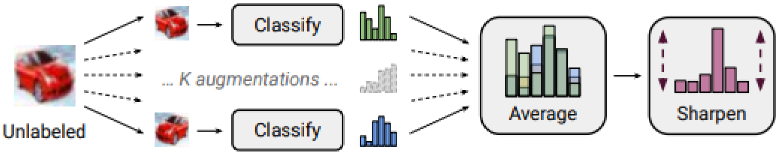
- Consistency-Regularization method.
- Encouraging consistent predictions $f(\alpha u_1 + (1-\alpha)u_2) = \alpha f(u_1) + (1-\alpha)f(u_2)$
- Let $\text{Mix}_\lambda(u_j, u_k) = \lambda u_j + (1-\lambda)u_k$
- Most of $\text{Mix}_\lambda(u_j, u_k)$ lie on regions of low density.
- The entropy of $\text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k))$ may
- So, ICT uses unlabeled loss : $L(\theta) = \|f_\theta(\text{Mix}_\lambda(u_j, u_k)) - \text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k))\|_2$

MixMatch(2019)

- Consistency-Regularization method
- The differences between ICT and MixMatch are
  1. Label Guessing
  2. Sharpening
  3. Using Labeled data to make mixup loss.

MixMatch(2019)

- Label Geussing : $\bar{q}_b = \frac{1}{K}\sum_{k=1}^{K} \mathrm{p}_{\text{model}}(y|\hat{u}_{b,k};\theta)$
  where $\hat{u}_{b,k}$ is a k-th augmented data from u_b
- $\text{Sharpen}(p, T)_i := p_i^{1/T} / \sum_{j=1}^{L} p_j^{1/T}$
  Using sharpening technique, $q_b = \text{Sharpen}(\bar{q}_b, T)$ is considered as target for the model's prediction.



MixMatch(2019)

- Let $\hat{\mathcal{X}} = ((\hat{x}^b, p_b) : b \in (1, ..., B))$ and $\hat{\mathcal{U}} = ((\hat{u}_{b,k}, q_b) : b \in (1, ..., B), k \in (1, ..., K))$
- The new generated datasets are ....
  $\mathcal{X}'_i = Mix_\lambda(\hat{\mathcal{X}}_i, \text{shuffle}(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}}))_i)$
  $\mathcal{U}'_i = Mix_\lambda(\hat{\mathcal{U}}_i, \text{shuffle}(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}}))_{i+|\hat{\mathcal{X}}|})$
- The final loss are...
  $$\mathcal{L}_\mathcal{X} = \frac{1}{\mathcal{X}'} \sum_{x,p \in \mathcal{X}'} H(p, p_{\text{model}}(y|x;\theta))$$
  $$\mathcal{U}_\mathcal{X} = \frac{1}{\mathcal{U}'} \sum_{u,q \in \mathcal{U}'} \|q - p_{\text{model}}(y|u;\theta)\|_2^2$$

Experiments

- Dataset : CIFAR-10(4000 labels)
- Preprocessing : zero-pad each image with 2 pixels, random crop, horizontal flip w/ prob 0.5 followed by per-channel standardization and ZCA.
- Architecture : CNN-13, Wide-Resnet-28-2.

Experiments-result

Method	Test acc.(%)	
Model	CNN-13	WRN28-2
CrossEnt(SL)	50.30	-
VAT	84.19	-
FAT	85.13	-
ICT	92.08	92.11
VAT + ICT	91.39	92.02
FAT + ICT	91.78	92.30
MixMatch	-	$95.05^{\dagger}$

Table 1: Comparison of prediction accuracies. † refers to the result reported in the paper. The red text refers to under training

Implementation details

- I run the experiments for 450 epochs.
- The initial learning rate was set to 0.1
- The momentem was set to 0.9, L2 regularization coefficient 0.0001 and a batch-size of 100.
- The Consistency coefficient is ramped up form its initial value 0.0 to its maximum value at one-fourth of the the total number of epochs using the same sigmoid schedule of (Tarvainen and Valpola, 2017)
- The maximum value of consistency coefficient is set to 100.