Data Selection for Supervised Dialogue Generation

Yahui Liu

Tencent AI Lab

yahui.cvrs@gmail.com

July 19, 2018

Self-paced learning

Self-Paced Curriculum Learning[1]
MentorNet: Regularizing Very Deep Neural Networks on Corrupted Labels[2]

$$\min_{\boldsymbol{\theta}, \mathbf{v} \in [0,1]^n} \mathbb{F}(\boldsymbol{\theta}, \mathbf{v}) = \frac{1}{n} \sum_{i=1}^{n} v_i \mathcal{L}(\mathbf{y}_i, G_{\boldsymbol{\theta}}(\mathbf{x}_i)) \tag{1}$$

[1] Jiang L. et al. Self-Paced Curriculum Learning, AAAI 2015

[2] Jiang L. et al. MentorNet: Regularizing Very Deep Neural Networks on Corrupted Labels, arXiv 2017

Curriculum Learning

Insights

learning principle underlying the cognitive process of humans and animals, which generally start with learning easier aspects of a task, and then gradually take more complex examples into consideration.

Curriculum

determines a sequence of training samples which essentially corresponds to a list of samples ranked in ascending order of learning difficulty.

Key

find a ranking function that assigns learning priorities to training samples.

Curriculum Learning

Curriculum Learning (CL)

The curriculum is assumed to be given by an oracle beforehand, and remains fixed thereafter.

- flexible to incorporate prior knowledge from various sources,
- the curriculum is predetermined a priori and cannot be adjusted accordingly, taking into account the feedback about the learner.

Self-Paced Learning (SPL)

- dynamically generated by the learner itself,
- a concise biconvex problem, ignoring prior knowledge.

SPL

$$\min_{\boldsymbol{\theta}, \mathbf{v} \in [0,1]^n} \mathbb{F}(\boldsymbol{\theta}, \mathbf{v}) = \frac{1}{n} \sum_{i=1}^{n} v_i \mathcal{L}(\mathbf{y}_i, G_{\boldsymbol{\theta}}(\mathbf{x}_i)) + \lambda \sum_{i=1}^{n} v_i \tag{2}$$

Alternative Convex Search

a block of variables are optimized while keeping the other block fixed.

(1) updating $\mathbf{v}$ with a fixed $\boldsymbol{\theta}$, a sample whose loss is smaller than a certain threshold λ is taken as an "easy" sample;

(2) when updating $\boldsymbol{\theta}$ with a fixed $\mathbf{v}$, the classifier is trained only on the selected "easy" samples.

Self-paced Curriculum Learning (SPCL)

instructor-student collaborative

$$\min_{\boldsymbol{\theta}, \mathbf{v} \in [0,1]^n} \mathbb{F}(\boldsymbol{\theta}, \mathbf{v}) = \frac{1}{n} \sum_{i=1}^{n} v_i \mathcal{L}(\mathbf{y}_i, G_{\boldsymbol{\theta}}(\mathbf{x}_i)) + f(\mathbf{v}; \lambda), \text{ s.t. } \mathbf{v} \in \Psi \qquad (3)$$

Given a predetermined curriculum $\gamma(\cdot)$ on training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ and their weights variable $\mathbf{v} = [v_1, \cdots, v_n]^T$.
A feasible region Ψ is called a curriculum region of γ if:

- *Soundness*: Ψ is a nonempty convex set;
- *Rule*: if $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$, it holds that $\int_\Psi v_i d\mathbf{v} > \int_\Psi v_j d\mathbf{v}$, where $\gamma(\mathbf{x}_i)$ calculates the expectation of v_i within Ψ.

SPCL

Self-Paced Function

(1) $f(\mathbf{v};\lambda)$ is convex with respect to $\mathbf{v} \in [0,1]^n$;

(2) When all variables are fixed except for v_i, ℓ_i, v_i^* decreases with l_i, and it holds that $\lim\limits_{\ell_i \to 0} v_i^* = 1$, $\lim\limits_{\ell_i \to \infty} v_i^* = 0$;

(3) $\|\mathbf{v}\|_1 = \sum_{i=1}^n v_i$ increases with respect to λ, and it holds that $\forall i \in [1,n]$, $\lim\limits_{\lambda \to 0} v_i^* = 0$, $\lim\limits_{\lambda \to \infty} v_i^* = 1$;

where $\mathbf{v}^* = \arg\min_{\mathbf{v}\in[0,1]^n} \sum v_i \ell_i + f(\mathbf{v};\lambda)$.

Algorithm & Implementation

Algorithm

Algorithm 1: Self-paced Curriculum Learning.

input : Input dataset $\mathcal{D}$, predetermined curriculum γ, self-paced function f and a stepsize μ
output: Model parameter $\mathbf{w}$

```
Derive the curriculum region Ψ from γ;
Initialize v*, λ in the curriculum region;
while not converged do
    Update w* = arg min_w E(w, v*; λ, Ψ);
    Update v* = arg min_v E(w*, v; λ, Ψ);
    if λ is small then increase λ by the stepsize μ;
    ;
end
return w*
```

Implementation

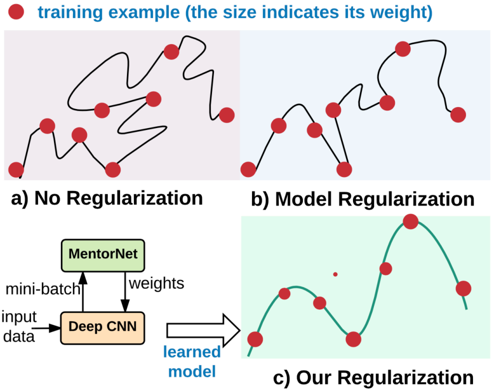
- Binary Scheme:
  $f(\mathbf{v};\lambda) = -\lambda\|v\|_1 = -\lambda\sum_{i=1}^{n} v_i$
- Linear Scheme:
  $f(\mathbf{v};\lambda) = \frac{1}{2}\lambda\sum_{i=1}^{n}(v_i^2 - 2v_i)$;
- Logarithmic Scheme:
  $f(\mathbf{v};\lambda) = \sum_{i=1}^{n}\zeta v_i - \frac{\zeta^{v_i}}{\log\zeta}$;
- Mixture Scheme:
  $f(\mathbf{v};\lambda) = -\zeta\sum_{i=1}^{n}\log(v_i + \frac{1}{\lambda_1}\zeta)$.

Comparison

	CL	**SPL**	**Proposed SPCL**
Comparable to human learning	Instructor-driven	Student-driven	Instructor-student collaborative
Curriculum design	Prior knowledge	Learning objective	Learning objective + prior knowledge
Learning schemes	Multiple	Single	Multiple
Iterative training	Heuristic approach	Gradient-based	Gradient-based

Motivation

Deep models are trained on big data where labels are often noisy, the ability to overfitting noise can lead to poor performance.



MentorNet

Formulation

$$\min_{\mathbf{w}\in\mathbb{R}^d,\mathbf{v}\in[0,1]^{n\times m}} \mathbb{F}(\mathbf{w},\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^T \mathcal{L}(\mathbf{y}_i, g_s(\mathbf{x}_i,\mathbf{w})) + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2 \quad (4)$$

Bottleneck

- minimizing **w** when fitting **v**, stochastic gradient descent often takes many steps before converging;
- minimizing **v** when fitting **w**, fixed vector **v** may not even fit into memory.

MentorNet

Algorithm 1. SPADE Alg. for optimizing Eq. (1)

Input : Input dataset $\mathcal{D}$, an explicit regularizer G or a function g_m.

Output: Model parameters $\mathbf{w}$ of the StudentNet.

Initialize $\mathbf{w}^{(0)}, \mathbf{v}^{(0)}, \theta^{(0)}, \ell_{pt}^{(0)}, t = 0$

while *Not Converged* **do**

  Fetch a mini-batch Ξ_t uniformly at random;

  Compute the loss ℓ for Ξ_t and its percentile $\ell_{pt}^{(t)}$;

  $\ell_{pt}^{(t)} = (1-\gamma)\ell_{pt}^{(t-1)} + \gamma\ell_{pt}^{(t)}$;

  if *G is used* **then**

    $\mathbf{v}_{\Xi}^{(t)} = \mathbf{v}_{\Xi}^{(t-1)} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{(t-1)}, \mathbf{v}^{(t-1)})|_{\Xi_t}$;

  else Update $\mathbf{v}_{\Xi}^{(t)} = g_m(\Xi_t, \mathbf{w}^{(t-1)})$;

  $\theta^{(t)} = \theta^{(0)} \frac{1}{k} \sum_{i=1}^{k} \mathbf{v}_{\Xi_i}^{(t)}$;

  $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{(t-1)}, \mathbf{v}^{(t)})|_{\Xi_t}$;

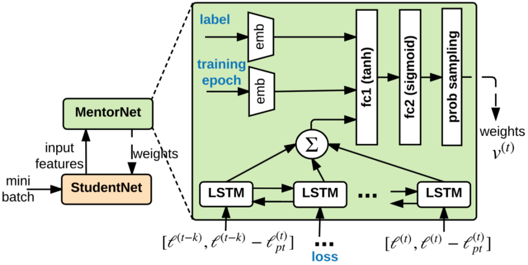
  $t \leftarrow t + 1$

end

return $\mathbf{w}^{(t)}$

MentorNet

Architecture



MentorNet

The parameters of MentorNet and StudentNet are not learned jointly to avoid a trivial solution of producing zero weights for all examples.

Pretraining

a pretraining dataset $\mathcal{D}_{pre} = \{(\mathbf{z}_i, v_i^*)\}_i$, where $\mathbf{z}_i$ the i-th input feature about loss, label and traning epoch, and $v_i^* \in [0, 1]$ is a desirable weight. If explicit regularizer G is known:

$$\arg\min_{\Theta} \sum_{\mathbf{z}_i \in \mathcal{D}_{pre}} g_m(\mathbf{z}_i; \Theta)\ell_i + G(g_m(\mathbf{z}_i; \Theta); \lambda) \tag{5}$$

Otherwise:

$$\arg\min_{\Theta} \sum_{\mathbf{z}_i \in \mathcal{D}_{pre}} \|v_i^* - g_m(\mathbf{z}_i; \Theta)\|_2^2 \tag{6}$$

MentorNet

a third dataset $\mathcal{D}_{ft} = \{(\mathbf{x}_i, \mathbf{y}_i, v_i^*)\}$, v_i is a binary label indicating whether this example should be learned.

Fine-tuning

Mixture of Experts:
For each $(\mathbf{x}_i, \mathbf{y}_i)$ in $\mathcal{D}_{ft}$ we first compute its input features $\mathbf{z}_i$. Denote $\mathbf{g_k}(\mathbf{z}_i) = [g_1(\mathbf{z}_i), \cdots, g_k(\mathbf{z}_i)]$ the weights obtained by k pretrained MentorNet $g_1, \cdots, g_k$.

$$\begin{aligned} \arg\min_{\Theta, \mathbf{w_g}} \sum_{v_i \in \mathcal{D}_{ft}} & v_i^* \log(G_\sigma(\mathbf{w_g}^T \mathbf{g_k}(\mathbf{z}_i) + \epsilon)) \\ & + (1 - v_i^*) \log(1 - G_\sigma(\mathbf{w_g}^T \mathbf{g_k}(\mathbf{z}_i) + \epsilon)) \end{aligned} \tag{7}$$

Summerization

- Data selection/regularization is an useful tool for supervised learning models.
- Our reweighting methods only depends on prior knowledge, which can be improved in a SPCL way.

Thanks!