Optimizing Data Usage via Differentiable Rewards

Wang et al., In ICML 2020

Data Selection: What and Why

- Standard supervised learning:
  - sample training instances with equal weights
  - sensitivity to the structure and domain of data
  - need to optimize the data usage
- Data selection:
  - selecting a subset
  - instance weighting
  - curriculum learning
  - active learning
  - reinforcement learning (this paper)

Related Work

- data filtering critera & training curriculum
- domain-specific knowledge and hand-designed heuristics
- parameterized neural networks:
    - curriculum learning method that trains a mentor network to select clean data based on features from both the data and the main model. (MentorNet, Jiang et al., 2018)
    - teacher-student network (Fang et al., 2018) that directly optimizes development set accuracy over multiple training runs; single reward signal provided by dev set accuracy at the end of training
- we need:
    - no heuristics
    - generalizable to various tasks
    - adaptively optimize the data usage

Reinforcement Learning for Data Selection

- scorer network
  - minimizes the model loss on the development set
- reward
  - gradient alignment between the training examples and the dev set
- optimization
  - bi-level optimization
  - a direct differentiation of the scorer parameters to optimize the model loss on the dev set
  - Differentiable Data Selection (DDS)

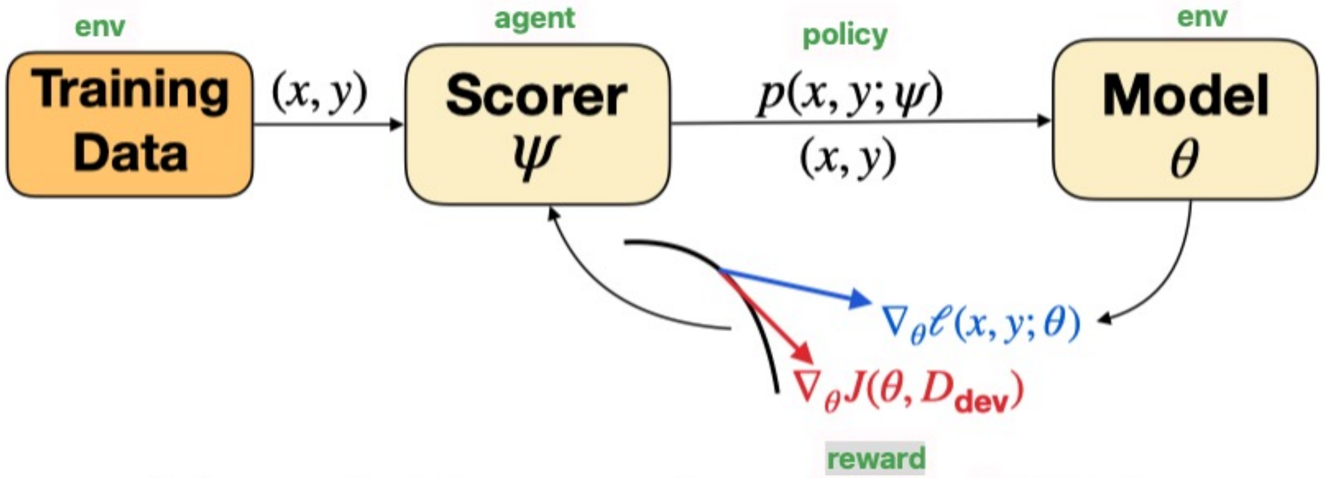

Figure 1: The general workflow of DDS.

Differentiable Data Selection

- Learning objective

$$\theta^* = \underset{\theta}{\operatorname{argmin}} J(\theta, P) \text{ where } J(\theta, P) = \mathbb{E}_{x,y \sim P(X,Y)}[\ell(x, y; \theta)]$$

- Scorer network adjusts the weights of examples in D_{train} to minimize $J(\theta, D_{dev})$

$$\psi^* = \underset{\psi}{\operatorname{argmin}} J\left(\theta^*(\psi), \mathcal{D}_{\text{dev}}\right) \text{ where } \theta^*(\psi) = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{x,y \sim P(X,Y;\psi)}[\ell(x, y; \theta)]$$

- Reward of RL agent
  - approximates the dev set performance of the resulting model after the model is updated on this example.

Learning to Optimize Data Usage

Scorer network update:

$$\psi_{t+1} \leftarrow \psi_t + \underbrace{\nabla_\theta \ell\left(x, y; \theta_{t-1}\right) \cdot \nabla_\theta J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)}_{R(x,y)} \nabla_\psi \log(P(X, Y; \psi))$$

REINFORCE (Wiliams, 1992)

Model update:

$$\theta_t \leftarrow \theta_{t-1} - \nabla_\theta J\left(\theta_{t-1}, P(X, Y; \psi)\right)$$

Deriving Rewards through Direct Differentiation

Approximate derivation of the gradient:

$$\nabla_\psi J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)$$

$$= \nabla_{\theta_t} J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)^\top \cdot \nabla_\psi \theta_t(\psi)$$ > apply chain rule

$$\theta_t \leftarrow \theta_{t-1} - \nabla_\theta J\left(\theta_{t-1}, P(X, Y; \psi)\right)$$

$$= \nabla_{\theta_t} J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)^\top \cdot \nabla_\psi \left(\theta_{t-1} - \nabla_\theta J\left(\theta_{t-1}, \psi\right)\right)$$ > subsitute θ_t

$$\approx -\nabla_{\theta_t} J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)^\top \cdot \nabla_\psi \left(\nabla_\theta J\left(\theta_{t-1}, \psi\right)\right)$$ > Markov assumption: $\nabla_\psi \theta_{t-1} \approx 0$

$$= -\nabla_\psi \mathbb{E}_{x,y \sim P(X,Y;\psi)} \left[\nabla_\theta J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)^\top \cdot \nabla_\theta \ell\left(x, y; \theta_{t-1}\right)\right]$$

$$J\left(\theta_{t-1}, \psi\right) = \mathbb{E}_{x,y \sim P(X,Y;\psi)}\left[\ell(x, y; \theta_{t-1})\right]$$

$$= -\mathbb{E}_{x,y \sim P(X,Y;\psi)} \left[\left(\nabla_\theta J\left(\theta_t, \mathcal{D}_{\text{dev}}\right)^\top \cdot \nabla_\theta \ell\left(x, y; \theta_{t-1}\right)\right) \cdot \nabla_\psi \log P(x, y; \psi)\right]$$

Instantiations of DDS

Classification

- identical model acrchitecture with independent weights
- uniform mini-batch data sampling
- scaled gradient update
- approximation of per-example gradient via first order Taylor expansion

$$v^\top \cdot \nabla_\theta \ell\left(x_i, y_i; \theta_{t-1}\right) \approx \frac{1}{\epsilon}\left(\ell\left(x_i, y_i; \theta_{t-1} + \epsilon v\right) - \ell\left(x_i, y_i; \theta_{t-1}\right)\right)$$

Algorithm 1 Training a classification model with DDS.

Input : $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}$

Output : Optimal parameters θ^*

Initializer θ_0 and ψ_0

for $t = 1$ **to** *num_train_steps* **do**

  Sample B training data points $x_i, y_i \sim \text{Uniform}(\mathcal{D}_{\text{train}})$

  Sample B validation data points $x'_i, y'_i \sim \text{Uniform}(\mathcal{D}_{\text{dev}})$

  ▷ *Optimize* θ

  $g_\theta \leftarrow \sum_{i=1}^{B} p(x_i, y_i; \psi_{t-1}) \nabla_\theta \ell(x_i, y_i; \theta_{t-1})$

  Update $\theta_t \leftarrow \text{GradientUpdate}\left(\theta_{t-1}, g_\theta\right)$

  ▷ *Evaluate* θ_t *on* $\mathcal{D}_{dev}$

  Let $d_\theta \leftarrow \frac{1}{B} \sum_{j=1}^{B} \nabla_\theta \ell(x'_j, y'_j; \theta_t)$

  ▷ *Optimize* ψ

  $r_i \leftarrow d_\theta^\top \cdot \nabla_\theta \ell(x_i, y_i; \theta_{t-1})$

  Let $d_\psi \leftarrow \frac{1}{B} \sum_{i=1}^{B} \left[r_i \cdot \nabla_\psi \log p(x_i, y_i; \psi)\right]$

  Update $\psi_t \leftarrow \text{GradientUpdate}(\psi_{t-1}, d_\psi)$

end

Machine Translation

- Settings
  - S: low-resource language
  - (S1, S2, … , Sn): multilingual parallel corpus
  - T: target language
  - dev set consists parallel data between S and T
- Aim
  - pick parallel data from any of the source languages to the target language to improve translation of a particular LRL S

Algorithm 2 Training multilingual NMT with DDS.

Input : $\mathcal{D}_{\text{train}}$; K: number of data to train the NMT model before updating ψ; E: number of updates for ψ; α_1,α_2: discount factors for the gradient

Output : The converged NMT model θ^*

Initialize ψ_0, θ_0

▷ *Initialize the gradient of each source language*

$grad[S_i] \leftarrow 0$ **for** *i in n*

while θ *not converged* **do**

  $X, Y \leftarrow \text{load\_data}(\psi, \mathcal{D}_{\text{train}}, K)$

  ▷ *Train the NMT model*

  for x_i, y *in* X, Y **do**

    $\theta_t \leftarrow \text{GradientUpdate}\left(\theta_{t-1}, \nabla_{\theta_{t-1}} \ell(x_i, y; \theta_{t-1})\right)$

    $\mathrm{g}[S_i] \leftarrow \alpha_1 \times \mathrm{g}[S_i] + \alpha_2 \times \nabla_{\theta_{t-1}} \ell(x_i, y; \theta_{t-1})$

  end

  ▷ *Optimize* ψ

  for *iter in E* **do**

    sample B data pairs from $\mathcal{D}_{\text{train}}$

    $r_i \leftarrow \mathrm{g}[S_i]^\top \mathrm{g}[S]$

    $d_\psi \leftarrow$

    $\frac{1}{B} \sum_{j=1}^{B} \sum_{i=1}^{n} \left[r_i \nabla_{\psi_{t-1}} \log \left(p \left(S_i | y_j; \psi_{t-1} \right) \right) \right]$

    $\psi_t \leftarrow \text{GradientUpdate}(\psi_{t-1}, d_{\psi_{t-1}})$

  end

end

Machine Translation

- Target conditioned sampling
  - assume a uniform distribution over the target sentence Y
  - Given the target sentence, parameterize the conditional distribution of which source sentence to pick p(X|y; ψ)
- Only update ψ after updating the NMT model for a fixed number of steps
- Sample the data according to p(X|y; ψ) to get a Monte Carlo estimate of the objective of scorer network

Algorithm 2 Training multilingual NMT with DDS.

Input : $\mathcal{D}_{\text{train}}$; K: number of data to train the NMT model before updating ψ; E: number of updates for ψ; α_1,α_2: discount factors for the gradient
Output : The converged NMT model θ^*
Initialize ψ_0, θ_0
▷ *Initialize the gradient of each source language*
$grad[S_i] \leftarrow 0$ **for** *i in n*
while θ *not converged* **do**
  $X, Y \leftarrow \text{load\_data}(\psi, \mathcal{D}_{\text{train}}, K)$
  ▷ *Train the NMT model*
  for x_i, y *in* X, Y **do**
    $\theta_t \leftarrow \text{GradientUpdate}\left(\theta_{t-1}, \nabla_{\theta_{t-1}}\ell(x_i, y; \theta_{t-1})\right)$
    $\text{g}[S_i] \leftarrow \alpha_1 \times \text{g}[S_i] + \alpha_2 \times \nabla_{\theta_{t-1}}\ell(x_i, y; \theta_{t-1})$
  end
  ▷ *Optimize* ψ
  for *iter in E* **do**
    sample B data pairs from $\mathcal{D}_{\text{train}}$
    $r_i \leftarrow \text{g}[S_i]^\top \text{g}[S]$
    $d_\psi \leftarrow$
    $\frac{1}{B}\sum_{j=1}^{B}\sum_{i=1}^{n}\left[r_i \nabla_{\psi_{t-1}} \log\left(p\left(S_i|y_j;\psi_{t-1}\right)\right)\right]$
    $\psi_t \leftarrow \text{GradientUpdate}(\psi_{t-1}, d_{\psi_{t-1}})$
  end
end

Experiments

Image Classification

- CIFAR-10:
  - reduced setting of roughly 10% training labels, first 4k examples in the training set
  - pre-activation WideResNet-28
- ImageNet:
  - first 102 TFRecord shards
  - post-activation ResNet-50

DDS with Prior Knowledge

- retrained DDS: initalize with trained scorer network
- TCS+DDS: initialize the parameters of DDS with the TCS heuristics

- Baselines
  - Uniform: standard supervised training
  - SPCL: a curriculum learning method that dynamically updates the curriculum to focus more on the “easy” training examples based on model loss.
- Filtering noisy data
  - BatchWeight: scales example training loss in a batch with a locally optimized weight vector using a small set of clean data.
  - MentorNet: select clean data based on features from both the data and the main model

Image Classification

Methods	CIFAR-10 (WRN-28-k)		ImageNet (ResNet-50)	
	4K, $k=2$	Full, $k=10$	10%	Full
Uniform	82.60±0.17	95.55±0.15	56.36/79.45	76.51/93.20
SPCL	81.09±0.22	93.66±0.12	-	-
BatchWeight	79.61±0.50	94.11±0.18	-	-
MentorNet	83.11±0.62	94.92±0.34	-	-
DDS	83.63± 0.29	96.31± 0.13	**56.81/79.51**	**77.23/93.57**
retrained DDS	**85.56±0.20**	**97.91±0.12**	-	-

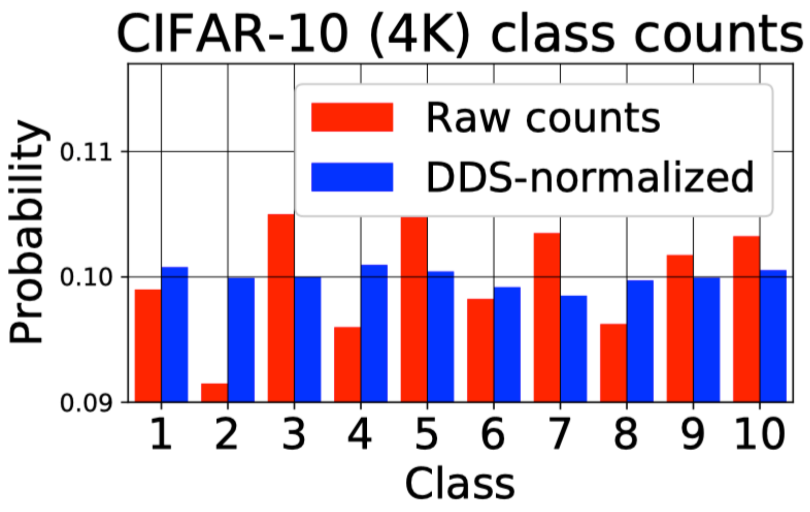

Figure 3: A trained DDS scorer learns to balance the class distributions of CIFAR-10 4K.

Image Classification

Figure 2: Example images from the ImageNet and their weights assigned by DDS. A trained DDS scorer assigns higher probabilities to images from ImageNet, in which the class content is more clear. Each image's label and weight in the minibatch is shown.

Multilingual NMT

- Model
  - standard LSTM-based attention model
- Dataset
  - TED: 58-language-to-English
- Baselines
  - Uniform: standard supervised training
  - SPCL: a curriculum learning method that dynamically updates the curriculum to focus more on the "easy" training examples based on model loss.
  - Related: data is selected uniformly from the target LRL and a linguistically related HRL
  - TCS: uniformly chooses target sentences, then picks which source sentence to use based on heuristics such as word overlap

Methods	**aze**	**bel**	**glg**	**slk**
Uniform	10.31	17.21	26.05	27.44
SPCL	9.07	16.99	23.64	21.44
Related	10.34	15.31	27.41	25.92
TCS	11.18	16.97	27.28	27.72
DDS	10.74	17.24	27.32	**28.20***
TCS+DDS	**11.84***	**17.74**†	**27.78**	27.74

Multilingual NMT

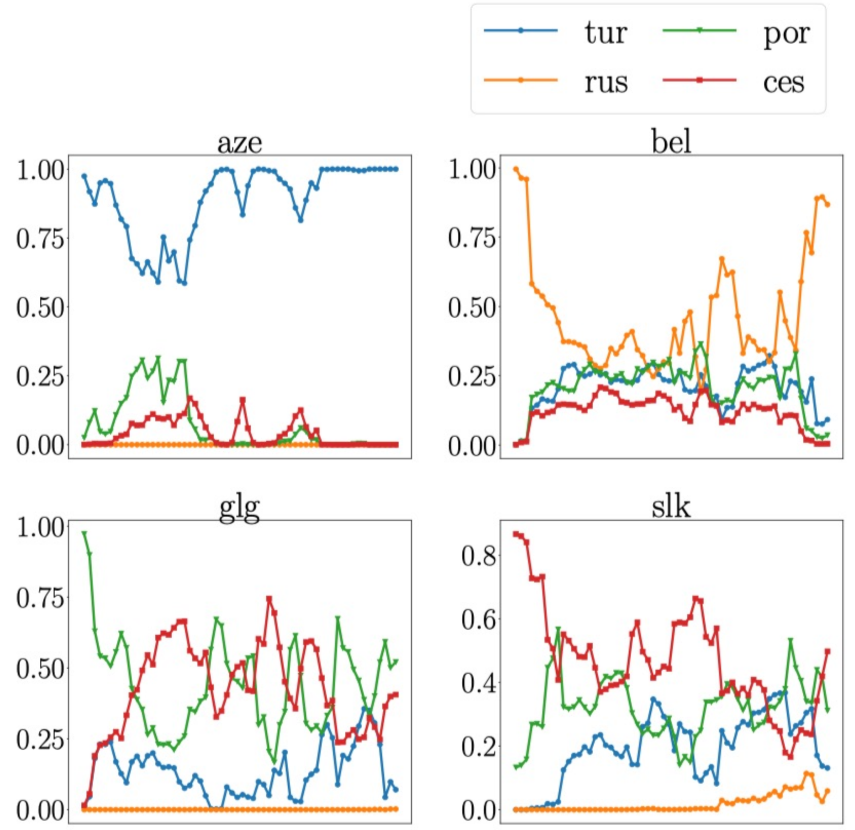

Figure 4: Language usage for TCS+DDS by training step. The distribution is initialized to focus on the most related HRL, and DDS learns to have a more balanced usage of all languages.

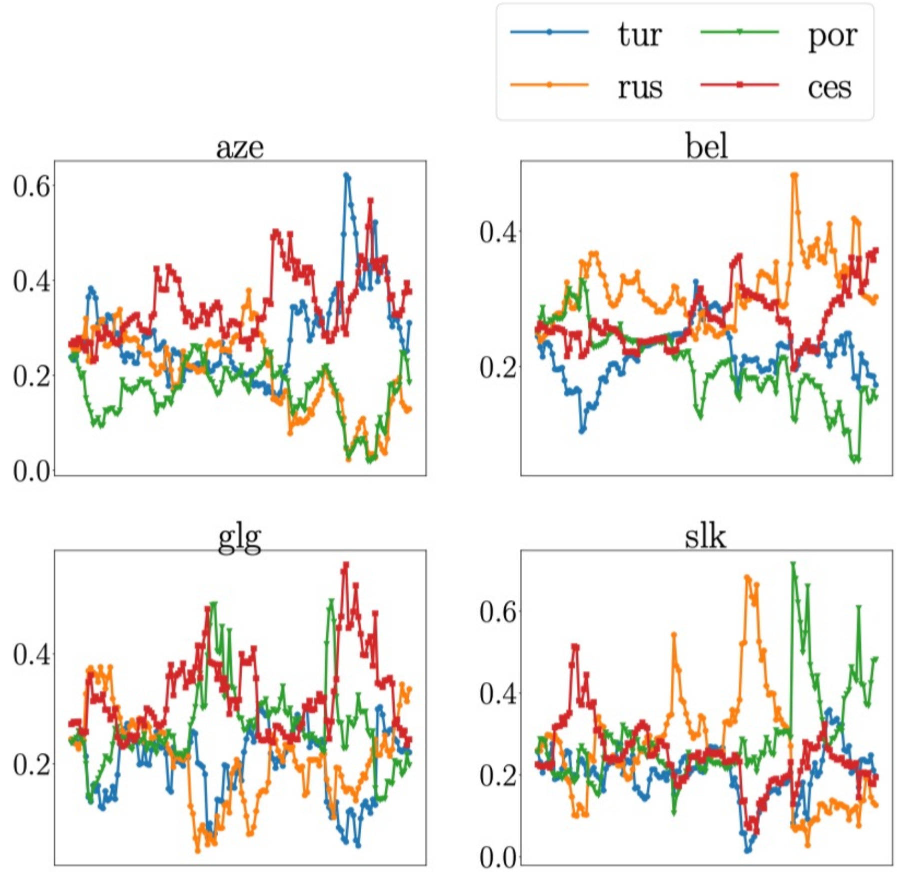

Figure 5: Language usage for DDS by training step. DDS learns to upweight the most related HRL after certain training steps.

shaoxiong.ji@aalto.fi