Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks

Trapit Bansal, Rishikesh Jha, Andrew McCallum

Motivation

- Fine-tuning on a new task still requires large amounts of task-specific labelled data
- Generalize to new tasks with few examples as a meta-learning problem
- Enables optimization-based meta-learning across tasks with **different number of classes**

Notation

Task: Episodic task (consist of support set and query set)

$$\{T_1, \ldots, T_M\}$$

Train: Support set $\mathcal{D}_i^{tr} \sim T_i$

Validation: Query set $\mathcal{D}_i^{val} \sim T_i$

Model-Agnostic Meta Learning (MAML)

Inner loop

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_i(\theta, \mathcal{D}_i^{tr})$$

Outer loop

$$\theta \leftarrow \theta - \beta \ \nabla_\theta \sum_{T_i \sim P(\mathcal{T})} \mathcal{L}_i(\theta_i', \mathcal{D}_i^{val})$$

Generating Softmax Parameters

Task-dependent softmax parameters

$$w_i^n, b_i^n = \frac{1}{|C_i^n|} \sum_{x_j \in C_i^n} g_\psi(f_\theta(\mathbf{x}_j)) \qquad C_i^n = \{x_j | y_j = n\}, \text{ where } n \in [N_i]$$

Prediction for a new instance x*

$$p(y|\mathbf{x}^*) = \text{softmax}\left\{\mathbf{W}_i h_\phi(f_\theta(\mathbf{x}^*)) + \mathbf{b}_i\right\}$$

Learn to Adapt

- Task-agnostic: BERT lower layers
- Task-specific: BERT higher layers, softmax-params
- Inner loop: Update task-specific params

$$\Theta = \theta_{\leq \nu} \cup \{\psi\}$$

$$\Phi_i = \theta_{>\nu} \cup \{\phi, \mathbf{W}_i, \mathbf{b}_i\}$$

$$\Phi_i^{(s+1)} = \Phi_i^{(s)} - \alpha_s \, \mathbb{E}_{\mathcal{D}_i^{tr} \sim T_i} \left[\nabla_\Phi \mathcal{L}_i(\{\Theta, \Phi_i\}, \mathcal{D}_i^{tr}) \right]$$

- Outer loop: update the task-agnostic params
  - Use first-order approximation for efficient computation

Algorithm 1 LEOPARD

Require: set of M training tasks and losses $\{(T_1, L_1), \dots, (T_M, L_M)\}$, model parameters $\Theta = \{\theta, \psi, \alpha\}$, hyper-parameters ν, G, β

Initialize θ with pre-trained BERT-base;

while not converged **do**

  # sample batch of tasks

  for all $T_i \in T$ **do**

    $\mathcal{D}_i^{tr} \sim T_i$ *# sample a batch of train data*

    $C_i^n \leftarrow \{x_j | y_j = n\}$ *# partition data according to class labels*

    $w_i^n, b_i^n \leftarrow \frac{1}{|C_i^n|} \sum_{x_j \in C_i^n} g_\psi(f_\theta(\mathcal{D}_i^{tr}))$ *# generate softmax parameters*

    $\mathbf{W}_i \leftarrow [w_i^1; \dots; w_i^{N_i}]; \quad \mathbf{b}_i \leftarrow [b_i^1; \dots; b_i^{N_i}]$

    $\Phi_i^{(0)} \leftarrow \theta_{>\nu} \cup \{\phi, \mathbf{W}_i, \mathbf{b}_i\}$ *# task-specific parameters*

    for $s := 0 \dots G-1$ **do**

      $\mathcal{D}_i^{tr} \sim T_i$ *# sample a batch of train data*

      $\Phi_i^{(s+1)} \leftarrow \Phi_i^{(s)} - \alpha_s \nabla_\Phi \mathcal{L}_i(\{\Theta, \Phi_i\}, \mathcal{D}_i^{tr})$ *# adapt task-specific parameters*

    end for

    $\mathcal{D}_i^{val} \sim T_i$ *# sample a batch of validation data*

    $g_i \leftarrow \nabla_\Theta \mathcal{L}_i(\{\Theta, \Phi_i^{(G)}\}, \mathcal{D}_i^{val})$ *# gradient of task-agnostic parameters on validation*

  end for

  $\Theta \leftarrow \Theta - \beta \cdot \sum_i g_i$ *# optimize task-agnostic parameters*

end while

Evaluation

- Training and Validation
  - GLUE benchmark
- Testing:
  - Entity typing: CoNLL-2003
  - Rating classification: Amazon Review
  - Text classification:

Results

	N	k	BERT$_{base}$	MT-BERT$_{softmax}$	MT-BERT	Proto-BERT	LEOPARD
Entity Typing							
CoNLL	4	4	50.44 $\pm$ 08.57	52.28 $\pm$ 4.06	**55.63** $\pm$ 4.99	32.23 $\pm$ 5.10	54.16 $\pm$ 6.32
		8	50.06 $\pm$ 11.30	65.34 $\pm$ 7.12	58.32 $\pm$ 3.77	34.49 $\pm$ 5.15	**67.38** $\pm$ 4.33
		16	74.47 $\pm$ 03.10	71.67 $\pm$ 3.03	71.29 $\pm$ 3.30	33.75 $\pm$ 6.05	**76.37** $\pm$ 3.08
MITR	8	4	49.37 $\pm$ 4.28	45.52 $\pm$ 5.90	**50.49** $\pm$ 4.40	17.36 $\pm$ 2.75	49.84 $\pm$ 3.31
		8	49.38 $\pm$ 7.76	58.19 $\pm$ 2.65	58.01 $\pm$ 3.54	18.70 $\pm$ 2.38	**62.99** $\pm$ 3.28
		16	69.24 $\pm$ 3.68	66.09 $\pm$ 2.24	66.16 $\pm$ 3.46	16.41 $\pm$ 1.87	**70.44** $\pm$ 2.89
Text Classification							
Airline	3	4	42.76 $\pm$ 13.50	43.73 $\pm$ 7.86	46.29 $\pm$ 12.26	40.27 $\pm$ 8.19	**54.95** $\pm$ 11.81
		8	38.00 $\pm$ 17.06	52.39 $\pm$ 3.97	49.81 $\pm$ 10.86	51.16 $\pm$ 7.60	**61.44** $\pm$ 03.90
		16	58.01 $\pm$ 08.23	58.79 $\pm$ 2.97	57.25 $\pm$ 09.90	48.73 $\pm$ 6.79	**62.15** $\pm$ 05.56

Results

k	Model	Entity Typing	Sentiment Classification	NLI
16	LEOPARD $_{10}$	37.62 $\pm$ 7.37	58.10 $\pm$ 5.40	78.53 $\pm$ 1.55
	LEOPARD $_{5}$	62.49 $\pm$ 4.23	71.50 $\pm$ 5.93	73.27 $\pm$ 2.63
	LEOPARD	69.00 $\pm$ 4.76	76.65 $\pm$ 2.47	76.10 $\pm$ 2.21
	LEOPARD-ZERO	44.79 $\pm$ 9.34	74.45 $\pm$ 3.34	74.36 $\pm$ 6.67

Table 3: Ablations: LEOPARD$_\nu$ does not adapt layers $0 - \nu$ (inclusive) in the inner loop, while LEOPARD adapts all parameters. Note that the outer loop still optimizes all parameters for all models. For new tasks (like entity typing) adapting all parameters is beneficial while for tasks seen at training time (like NLI) adapting fewer parameters is better. LEOPARD-ZERO is the model without the softmax-generator and a zero initialized softmax classifier which shows the importance of the softmax generator in LEOPARD.