1703.00837 - Meta Networks

- **Yunqiu Xu**

- **Unfinished, to be continued**

---

1. Introduction and Related Work

- Challenge of traditional DNN:
    - Needs large dataset
    - Can not learn continuously without forgetting previously learned patterns
- Previous meta-learning:
    - Formulate the problem as two-level learning:
    - Slow learning for meta-level across the tasks: get general knowledge
    - Fast learning for base-level within one task (即之前MAML的内循环) : general knowledge can be transferred fastly
- Our work : MetaNet
    - support meta-level continual learning
    - learn / generalize a new task from single example : **又是个"one-shot"?**

2. Meta Networks

- Total goal : fast learning and generalization by processing higher-order meta information
- Architecture : meta learner + base learner + external memory
    - **Why use external memory: for fast learning and generalization**

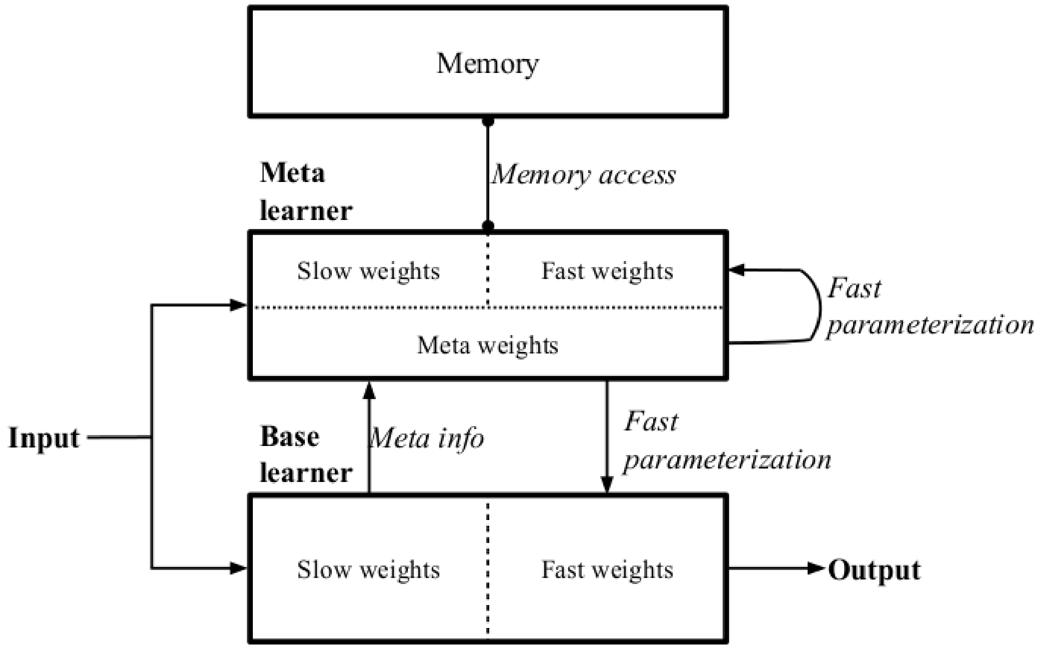

Figure 1. Overall architecture of Meta Networks.

- Meta learner :
  - Goal: fast weight generation
  - Task agnostic, performed in abstract meta space, supports continual learning
  - Performs meta-knowledge acquisition across tasks
  - Take input task and meta info as input
  - Then perform fast parameterization for itself and base learner
  - In this way, meta learner can get new concepts from input task
- Base learner:
  - Performs within each task by capturing the task objective
  - Provide the feedback (meta infomation) $\rightarrow$ explain its own status in this task space
- The weights:
  - Standard slow weights: updated via RL
  - Task-level fast weights: updated within the scope of each task
  - Example-level fast weights: updated for a specific input example
  - **Note that fast weights are generated and integrated into both base and meta learner to shift their inductive bias**

- Meta information : two types of loss functions
  - representation (embedding) loss: for building good representation learner criteria
  - main (task) loss: for input task objective

Algorithm 1 MetaNet for one-shot supervised learning

Require: Support set $\{x'_i, y'_i\}_{i=1}^{N}$ and Training set $\{x_i, y_i\}_{i=1}^{L}$
Require: Base learner b, Dynamic representation learning function u, Fast weight generation functions m and d, and Slow weights $\theta = \{W, Q, Z, G\}$
Require: Layer augmentation scheme

Sample T examples from support set
for $i = 1, T$ **do**
  $\mathcal{L}_i \leftarrow loss_{emb}(u(Q, x'_i), y'_i)$
  $\nabla_i \leftarrow \nabla_Q \mathcal{L}_i$
end for
$Q^* = d(G, \{\nabla\}_{i=1}^{T})$
for $i = 1, N$ **do**
  $\mathcal{L}_i \leftarrow loss_{task}(b(W, x'_i), y'_i)$
  $\nabla_i \leftarrow \nabla_W \mathcal{L}_i$
  $W_i^* \leftarrow m(Z, \nabla_i)$
  Store W_i^* in i^{th} position of memory M
  $r'_i = u(Q, Q^*, x'_i)$
  Store r'_i in i^{th} position of index memory R
end for
$\mathcal{L}_{train} = 0$
for $i = 1, L$ **do**
  $r_i = u(Q, Q^*, x_i)$
  $a_i = attention(R, r_i)$
  $W_i^* = softmax(a_i)^\top M$
  $\mathcal{L}_{train} \leftarrow \mathcal{L}_{train} + loss_{task}(b(W, W_i^*, x_i), y_i)$ {Alternatively the base learner can take as input r_i instead of x_i}
end for
Update θ using $\nabla_\theta \mathcal{L}_{train}$

2.1 Meta Learner

2.2 Base Learner

2.3 Layer Augmentation

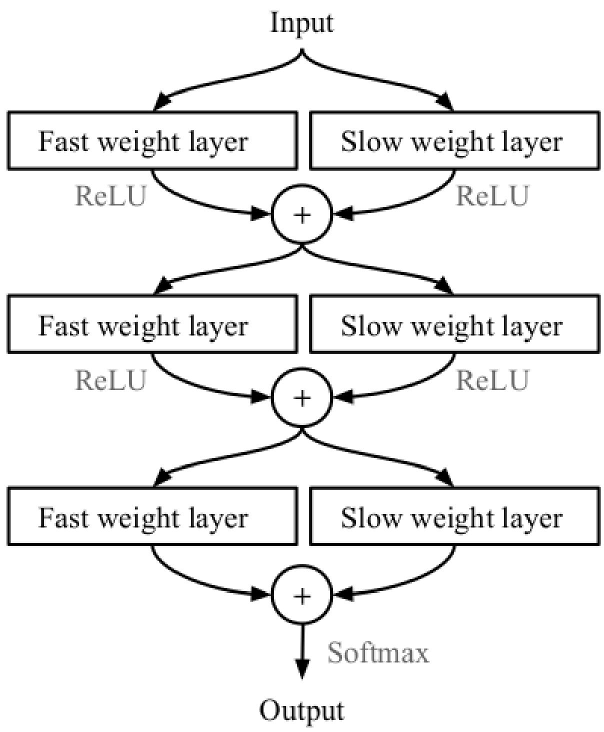

Figure 2. A layer augmented MLP

3. Experiment

Table 1. One-shot accuracy on Omniglot previous split

Model	5-way	10-way	15-way	20-way
Pixel kNN (Kaiser et al., 2017)	41.7	-	-	26.7
Siamese Net (Koch, 2015)	97.3	-	-	88.1
MANN (Santoro et al., 2016)	82.8	-	-	-
Matching Nets (Vinyals et al., 2016)	98.1	-	-	93.8
Neural Statistician (Edwards & Storkey, 2017)	98.1	-	-	93.2
Siamese Net with Memory (Kaiser et al., 2017)	98.4	-	-	95.0
MetaNet-	98.4	98.32	96.68	96.13
MetaNet	**98.95**	**98.67**	**97.11**	**97.0**
MetaNet+	98.45	97.05	96.48	95.08

Table 2. One-shot accuracy on Mini-ImageNet test set

Model	5-way
Fine-tuning (Ravi & Larochell, 2017)	28.86 ± 0.54
kNN (Ravi & Larochell, 2017)	41.08 ± 0.70
Matching Nets (Vinyals et al., 2016)	43.56 ± 0.84
MetaLearner LSTM (Ravi & Larochell, 2017)	43.44 ± 0.77
MetaNet	$\mathbf{49.21 \pm 0.96}$

Table 3. One-shot accuracy on Omniglot standard split

Model	5-way	10-way	15-way	20-way
Human performance (Lake et al., 2015)	-	-	-	95.5
Pixel kNN (Lake et al., 2013)	-	-	-	21.7
Affine model (Lake et al., 2013)	-	-	-	81.8
Deep Boltzmann Machines (Lake et al., 2013)	-	-	-	62.0
Hierarchial Bayesian Program Learning (Lake et al., 2015)	-	-	-	**96.7**
Siamese Net (Koch, 2015)	-	-	-	92.0
MetaNet	98.45	97.32	96.4	95.92

4. Conclusion and Future Work

- MetaNet:
    - Goal: rapid generalization
    - Performs generic knowledge acquisition in a meta space
    - Shifts parameters / inductive biases via fast parameterization

    - Use gradients as meta-information: generic and problem independent
- Future work:
    - More robust and expressive meta information
    - Better method to integrate slow and fast weights
- My understanding:
    - 这个工作和MAML有类似之处, 目的为快速学习泛化新任务, 且达到 "one-shot" 效果
    - 训练的目的在于学习slow weights, 然后通过快速学习 fast weights 来适应新任务
    - 和MAML的不同在于MetaNet所需的 "外部记忆" 是啥, MAML的卖点在于不需要额外的参数以及 "model agnostic" , 这篇文章呢?
    - 实验主要还是基于 Omniglot 和 Mini-ImageNet, 即MAML监督学习任务部分
        - Omniglot 这篇文章和 MAML 差不多
        - Mini-ImageNet 这篇文章效果稍强于 MAML