DAVID ISHAK KOSASIH
2019503 3

2a) i) $\theta_i' = \theta - \alpha \nabla_\theta L_{T_i}(f_\theta)$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum L_{T_i}(f_{\theta_i'})$$

$$= \theta - \beta \sum_{T_i \sim p(T)}' \nabla_\theta L_{T_i}(f_{\theta_i'})$$

$$= \theta - \beta \sum_{T_i \sim p(T)} (\nabla_\theta \theta_i') \nabla_{\theta_i'} L_{T_i}(f_{\theta_i'})$$

$$= \theta - \beta \sum_{T_i \sim p(T)} \underbrace{(I - \alpha \nabla_\theta^2 L_{T_i}(f_\theta))} \nabla_{\theta_i'} L_{T_i}(f_{\theta_i'})$$

Hessian matrix calculation is required

ii) This is because the adaptation of MAML requires hessian matrix calculation, as shown in 2a(i). As we know, hessian matrix is a technique for calculating the second derivative of an n-dimensional function.
$H(x) \in R^{n \times n}$ (the second order derivative and a symmetric matrix).
Therefore, this makes MAML computationally expensive.

iii) In 1st order approximation, we regard
$I - \alpha \nabla_\theta^2 L_{T_i}(f_\theta)$
as identity matrix I.
Update rule of MAML with 1st order approximation:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}(f_\theta - \alpha \delta)$$

where $\delta \leftarrow \nabla L_{T_i}(f_\theta)$

c) I selected Meta-SGD: learning to learn quickly for Few-Shot Learning

i) previous work, such as meta-LSTM, are difficult to train. Moreover, each parameter of the learner is updated independently in each step and thus limits its potential.

ii) Meta-SGD has a much higher capacity by learning to learn not just the learner initialization, but also the learner update direction and learning rate in a single meta-learing process.

iii) The objective function of meta-SGD is as follows:

$$\min_{\theta, \alpha} E_{T \sim p(T)}[L_{Test(T)}(\theta')] = E_{T \sim p(T)}[L_{Test(T)}(\theta - \alpha \circ \nabla L_{Train(T)}(\theta))]$$

where θ and α are (meta-) parameters of the meta-learner to be learned. To be more precise, θ represents the state of a learner that can be used to initialize the learner for any new task, and α is a vector of the same size as θ that decides both the update direction and learning rate. The adaptation term $\alpha \circ \nabla L_T(\theta)$ is a vector whose direction represents the update direction and whose length represents the learning rate.

Therefore, given a few examples $T = \{(x_i, y_i)\}$ for a few shot learning problem, meta-SGD first initializes the learner with θ and then adapts it to θ' in just one step, in a new direction $\alpha \circ \nabla L_T(\theta)$ different from the gradient $\nabla L_T(\theta)$ and using a learning rate implicitly implemented in $\alpha \circ \nabla L_T(\theta)$.

(IV) Because in meta-SGD, the initialization, update direction and learning rate are all learned via meta-learing.