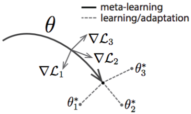
Optimisation-based Meta-learning

Reference - *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*

Choose model parameters w such that taking one or few gradient-steps on an unknown task (i.e., dataset) is maximally optimal: For each task, *adapt* $g_{\theta_0=\mathrm{w}}$ via gradient-step(s) using *test* loss on tasks, i.e., $\theta - \alpha\nabla_\theta\mathcal{L}_{\mathcal{T}_i}(g_\theta)$, over a number of tasks $\mathcal{T}_i$ sampled from a distribution. Note: f_{w} requires similar *adaptation* at (meta)-test time also. Also, as formulated MAML applies both to supervised as well as other ML tasks, e.g. reinforcement learning. Thus:



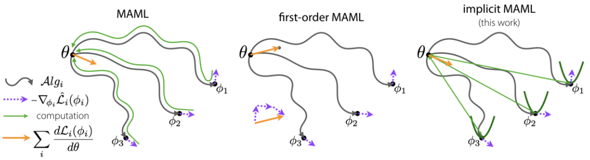
$$f_{\mathrm{w}}^{MAML}(x, D_{Train}, g) = g_{\mathrm{w}-\alpha\nabla\mathcal{L}(g_{\mathrm{w}}(D_{Train}))}(x) \equiv g_\phi(x)$$

In practice, more than one gradient steps are taken: *fast adaptation*.
Note that training f^{MAML}, i.e., optimizing for w across meta-training tasks, requires second-order derivatives, i.e., we need $\nabla_w\mathcal{L}(g_{\hat{\phi}}, D_{Test})$
$= (I - \alpha\nabla_w^2\mathcal{L}(g_w(D_{Train}))\nabla_\phi\mathcal{L}(g_\phi, D_{Test})|_{\phi=\hat{\phi}}$, where $\hat{\phi} = \mathrm{w} - \alpha\nabla\mathcal{L}(g_{\mathrm{w}}(D_{Train}))$

MAML, FO-MAML, and iMAML (Implicit layers)

References - *On First-Order Meta-Learning Algorithms* and *Meta-Learning with Implicit Gradients*

FO-MAML: $\mathrm{w} \leftarrow \mathrm{w} - \eta \sum_i \hat{\delta \mathrm{w}_i}$



iMAML: fully we minimize $G(\phi, \mathrm{w}) = \hat{\mathcal{L}}(\phi) + \frac{1}{2}\|\phi - \mathrm{w}\|^2$ fully, where $\hat{\mathcal{L}}$ denotes loss on D_{Train}. Let $\phi^*(\mathrm{w}) = \mathrm{argmin}_\phi\, G(\phi, \mathrm{w})$. For updating w we need $\nabla_{\mathrm{w}}\mathcal{L}(\phi^*)$ $= \mathrm{d}_{\mathrm{w}}\phi^* \nabla_\phi \mathcal{L}(g_\phi)|_{\phi=\phi^*}$. To compute $\mathrm{d}_{\mathrm{w}}\phi^*$:
$\frac{dG}{d\phi} = \nabla_\phi \hat{\mathcal{L}}(\phi) + (\phi - \mathrm{w}) = 0$ at ϕ^* so $\phi^* = \mathrm{w} - \nabla_\phi \hat{\mathcal{L}}(\phi)|_{\phi=\phi*}$ Thus,

$$\frac{d\phi^*}{d\mathrm{w}} = \left(I + \nabla^2_\phi \hat{\mathcal{L}}(\phi)|_{\phi=\phi^*}\right)^{-1}$$

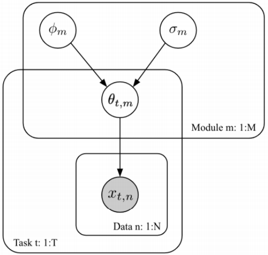
$$\mathrm{w} \leftarrow \mathrm{w} - \eta \sum_i \left(I + \nabla^2_\phi \hat{\mathcal{L}}_i(\phi)|_{\phi=\phi_i^*}\right)^{-1} \nabla_\phi \mathcal{L}_i(\phi)|_{\phi=\phi_i^*}$$

Modular Meta-learning: Variants of MAML

Reference - *Modular Meta-Learning with Shrinkage*

In general $\mathrm{w} = \{\theta_1 \dots \theta_M\}$ e.g., different layers of a network. Variants of MAML learn a prior for w that is adapted for each task; but do all layers need to adapt? E.g. if only one layer is adapted, perhaps it could be trained for many more steps per task without risk of over-fitting. This paper learns to *differently* adapt each layer: assuming each θ_m is normally distributed as $\mathcal{N}(\phi_m, \sigma_m^2)$. **Layers with small or zero σ_m^2 will not adapt.** To learn ϕ, σ^2 we take Bayesian view:

$$p(\mathrm{w}^{1:T}, \mathcal{D}|\phi, \sigma^2) = \prod_{t=1}^{T} \prod_{m=1}^{M} \mathcal{N}(\theta_m^t|\phi_m, \sigma_m^2) \prod_{t=1}^{T} p(\mathcal{D}_t|_t)$$



using the MAML approach to update ϕ, σ^2:
the inner loop computes:

$$\hat{\theta}^t(\phi, \sigma^2) \equiv \underset{\mathrm{w}^t}{\mathrm{argmin}} \left[-\log p(\mathcal{D}_t^{Train}|\mathrm{w}^t) - \log p(\mathrm{w}^t|\phi, \sigma^2) \right]$$

& the outer loop minimizes $\dfrac{1}{T} \sum_{t=1}^{T} -\log p(\mathcal{D}_t^{Test}|\hat{\mathrm{w}}^t)$.