Formalizing Meta-Learning


3

$$\min_{\omega} \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})} \mathcal{L}\text{meta}(D_{\mathcal{T}}, \omega)$$

Meta loss on task $\mathcal{T}$ when trained with learning parameters ω

$\mathcal{L}_{\text{meta}}$: score for generalization, learning speed, adaptability ... (**improve how?**)

ω: hyper params, initial weights, neural architecture... (**learn what?**)

We want the ω that yields the lowest $\mathscr{L}_{\text{meta}}$ on average in our set of tasks $p(\mathscr{T})$

It's a bi-level optimization!

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{M} \mathscr{L}_{\text{meta}} \left(\mathscr{D}_{\text{source}}^{\text{val}\,(i)}; \theta^{*(i)}(\omega), \omega \right)$$

$$\theta^{*(i)}(\omega) = \arg\min_{\theta} \mathscr{L}_{\text{task}} \left(\mathscr{D}_{\text{source}}^{\text{train}\,(i)}; \theta, \omega \right)$$

Formalizing Meta-Learning

We want the ω that yields the lowest $\mathscr{L}_{\text{meta}}$ on average in our set of tasks $p(\mathscr{T})$

$$\min_{\omega} \mathbb{E}_{\mathscr{T}\sim p(\mathscr{T})} \mathscr{L}_{\text{meta}}(D_{\mathscr{T}}, \omega)$$

Meta loss on task $\mathscr{T}$ when trained with learning parameters ω

$\mathscr{L}_{\text{meta}}$: score for generalization, learning speed, adaptability ... (**improve how?**)

ω: hyper params, initial weights, neural architecture... (**learn what?**)

It's a bi-level optimization!

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{M} \mathscr{L}_{\text{meta}} \left(\mathscr{D}_{\text{source}}^{\text{val}\,(i)}; \theta^{*(i)}(\omega), \omega \right)$$

$$\theta^{*(i)}(\omega) = \arg\min_{\theta} \mathscr{L}_{\text{task}} \left(\mathscr{D}_{\text{source}}^{\text{train}\,(i)}; \theta, \omega \right)$$

Metric Learning

Learn an embedding network that transforms raw inputs into a representation suitable for recognition by simple similarity comparison between query(“test”) and support(“training”) instances, (e.g., with cosine similarity or euclidean distance).

Bird

Dog

Monkey

Embedding Module