Meta Learning Framework.

A distribution over tasks $p(\mathcal{T})$

$$\mathcal{T} \begin{cases} \text{training data} : D_{\mathcal{T}}^{tr} = \{ X_{\mathcal{T}}^{tr} \; y_{\mathcal{T}}^{tr} \; \mathcal{L}_{\mathcal{T}} \} \\ \text{validation data} : D_{\mathcal{T}}^{val} = \{ X_{\mathcal{T}}^{val} \; y_{\mathcal{T}}^{tr} \; \mathcal{L}_{\mathcal{T}} \} \end{cases}$$

w belongs to φ

not task adaptive

$\Uparrow$

Predictor

$P_{\theta,w}$

$$\text{meta-learning} \begin{cases} \text{inner loop} : \text{operates within a task } \mathcal{T} \\ \text{outer loop} : \text{operates across a task } \mathcal{T} \end{cases}$$

meta-parameters φ

Aim: design a meta-learner that can generalize well on a new task by appropriately choosing the predictor's task adaptive parameters θ after observing $D_{\mathcal{T}_i}^{tr}$

For each training task $\mathcal{T}_i$

$$P_{\theta,w} : X_{\mathcal{T}_i}^{val} \to \hat{y}_{\mathcal{T}_i}^{val} \quad \text{conditioned on} \quad D_{\mathcal{T}_i}^{tr}$$

The meta-parameter φ are updated in the outer loop so as to obtain good generalization in the inner loop.

minimize $\mathbb{E}_{\mathcal{T}} \, \mathcal{L}_{\mathcal{T}} (\hat{y}_{\mathcal{T}}^{val}, y_{\mathcal{T}}^{val})$

Training on multiple tasks enables the meta-learner to produce $P_{\theta,w}$ that generalize well on a set of unseen tasks $\{\mathcal{T}_i\}$ sampled from $p(\mathcal{T})$

Meta-Learning Ingredients.

(1) $\varphi = (t_0, w, u)$ meta parameters

(2) I_{t_0} initialization function

(3) $\mathcal{U}_u$ update function.

task meta-data.

Initialization function $I_{t_0}(D^{tr}_{\tau_i}, C_{\tau_i})$

defines the initial values of θ for a given τ_i

* task meta-data C_{τ_i} may have a form of task ID or a texture description.

Update function $\mathcal{U}_u(\theta_{l-1}, D^{tr}_{\tau_i})$

defines an iterated update to predictor parameters θ at iteration l.

The initialization and update functions produces a sequence of predictor parameter. $\theta_{0:l} \equiv \{\theta_0 \cdots \theta_{l-1}, \theta_l\}$

We let the final predictor be a function of the whole sequence of parameters. $P_{\theta_{0:l}, w}$

eg. ① $P_{\theta_{0:l}, w}(\cdot) = \sum_{j=0}^{l} w_j P_{\theta_j, w}(\cdot)$

② $P_{\theta_{0:l}, w}(\cdot) = P_{\theta_l, w}(\cdot)$

Parameter θ ; Meta-parameters $\varphi = (t_0, w, u)$

Inner Loop : $\theta_0 \leftarrow I_{t_0}(D^{tr}_{\mathcal{T}_i}, c_{\mathcal{T}_i})$

$\theta_l \leftarrow \mathcal{U}_u(\theta_{l-1}, D^{tr}_{\mathcal{T}_i}) \quad \forall l > 0$

Prediction at x: $P_{\theta:l, w}(x)$

Outer Loop: $\varphi \leftarrow \varphi - \eta \cdot \nabla_\varphi \mathcal{L}_{\mathcal{T}_i}[P_{\theta_0:l,w}(X^{val}_{\mathcal{T}_i}), Y^{val}_{\mathcal{T}_i}]$

N-Beats as a Meta-learning Algorithm.

$$
\begin{tikzcd}
 & x_l \arrow[d, "f"'] & \\
Q & h_{l,4} \arrow[dl, "q"'] \arrow[d, "g"] & G \\
\hat{x}_l & \hat{y}_l &
\end{tikzcd}
$$

$$\hat{y} = \sum_{l=1}^{L} \hat{y}_l$$

$$\hat{y}_1 = g \circ f(x)$$

$$\hat{y}_l = g \circ f(x_{l-1} - \hat{x}_{l-1}) \qquad l > 1$$

$$\hat{x}_{l-1} = q \circ f(x_{l-1})$$

$$\hat{y} = g \circ f(x) + \sum_{l>1} g \circ f(x_{l-1} - q \circ f(x_{l-1})) \qquad (7)$$

(i) each application of $g \circ f$ in (7) is a predictor and

(ii) each block of N-Beats is the iteration of the inner meta-learning loop.

$$P_{\theta,w}(\cdot) = g_{w_g} \circ f_{w_f,\theta}(\cdot)$$

$w = (w_g, w_f)$ learned across tasks in the outer loop.

Task-specific parameters θ consist of the sequence of input shift vectors $\theta \equiv \{\mu_l\}_{l=0}^{L}$

defined such that the l-th block takes input $x_l = x - \mu_{l-1}$

$\mu_{l-1} = x - x_l$ for N-Beats. $x_{l+1} = x_l - \hat{x}_l$

$\mu_l = x - x_{l+1}$ $x_l - x_{l+1} = \hat{x}_l$ ②

$\mu_l = \mu_{l-1} + x_l - x_{l+1}$ ①

From ① ②: $\mu_l = \mu_{l-1} + \hat{x}_l$

This yields a recursive expression.

$$\mu_l \leftarrow \mathcal{U}_u(\mu_{l-1}, D^{tr}_{\tau_i})$$

$$\equiv \mu_{l-1} + g_{w_g} \circ f_{w_f}(x - \mu_{l-1}) \qquad D^{tr}_{\tau_i} \equiv \{x\}$$

$\theta = \{\mu_l\}_{l=0}^{L}$ are combined in the final output.

$$P_{\mu_{0:L}, w}(\cdot) = \sum_{j=0}^{L} w_j \cdot P_{\mu_j, w}(\cdot) \qquad w_j = 1 \ \forall j.$$

Conclusion: Even if predictor parameters w_g w_f are shared across blocks and fixed.

The behavior of $P_{\mu_{0:L}, w}(\cdot) = g_{w_g} \circ f_{w_f, \mu_{0:L}}(\cdot)$

is governed by $(w, \mu_1, \mu_2 \cdots)$

Therefore, the expressive power of the architecture can be expected to grow with the growing number of blocks in proportion to the growth of the space spanned by $\mu_{0:L}$

I think this claim is wrong for the size of θ is nearly negligible compared with $w = (w_g, w_f)$