Meta-Learning with Memory-Augmented Neural Networks

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, Timothy Lillicrap

ICML 2016

Reviewed by : Jack Lanchantin

[1]Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Meta-learning

- Scenario in which an agent learns at two levels
  - Rapid learning occurs *within* a task, for example, when learning to accurately classify within a particular dataset.
  - This learning is guided by knowledge accrued more gradually *across* tasks, which captures the way in which task structure varies across target domains.
- Given its two-tiered organization, meta-learning is often described as "learning to learn."

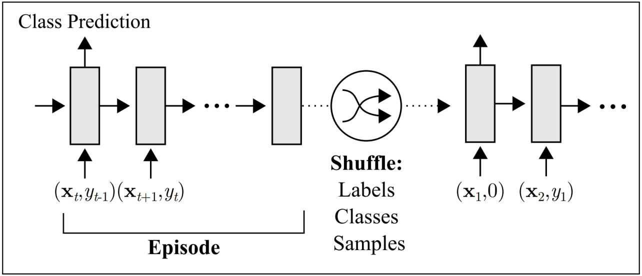
Meta-learning Task Methodology

- Usually we try to choose parameters θ to minimize loss $\mathcal{L}$ across dataset D.
- In meta-learning, we choose parameters to reduce the expected loss across a distribution of datasets $p(D)$:

$$\theta^* = argmin_\theta E_{D \sim p(D)}[\mathcal{L}(D; \theta)]$$

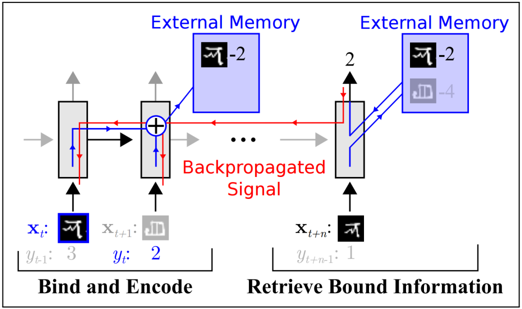
Setup for This Paper

- Dataset $D = \{d_t\}_{t=1}^{T} = \{(x_t, y_t)\}_{t=1}^{T}$
- At each timestep t, the network receives input x_t as well as the label of the previous example, y_{t-1}:
  - $(x_1, \text{null}),(x_2, y_1),\ldots,(x_T, y_{T-1})$
- Labels, classes, and samples are shuffled in each training "episode".



Memory-Augmented Neural Nets (MANNs)

- Learns to hold samples in memory until the correct labels are shown, after which they can be bound and stored for later use.



MANN Reading

Given input x_t, controller (LSTM) produces key k_t.

$\boldsymbol{M}_t$ is addressed using cosine similarity:

$$K(k_t, \boldsymbol{M}_t(i)) = \frac{k_t \cdot \boldsymbol{M}_t(i)}{||k_t||\,||c_t(i)||},$$

which is used to produce read-weight vector w_r^t:

$$w_r^t(i) \leftarrow \frac{exp(K(\boldsymbol{k}_t, \boldsymbol{M}_t(i)))}{\sum_j K(\boldsymbol{k}_t, \boldsymbol{M}_t(j))}.$$

A certain memory r_t is read using this read-weight vector:

$$\boldsymbol{r}_t \leftarrow \sum_i w_r^t(i)\boldsymbol{M}_t(i)$$

MANN Writing

Least Recently Used Access (LRUA)

LRUA: Content-based writer that writes memories to either the least used or most recently used memory location.

Usage weights:

$$\boldsymbol{w}_t^u \leftarrow \gamma \boldsymbol{w}_{t-1}^u + \boldsymbol{w}_t^r + \boldsymbol{w}_w^w$$

Least used weight:

$$w_t^{lu}(i) = \begin{cases} 0, & \text{if } w_t^u(i) > m(\boldsymbol{w}_t^u, n) \\ 1, & \text{if } w_t^u(i) \leq m(\boldsymbol{w}_t^u, n) \end{cases}$$

def: $m(\boldsymbol{v}, n) = n^{th}$ smallest element of vector $\boldsymbol{v}$

MANN Writing

Least Recently Used Access (LRUA)

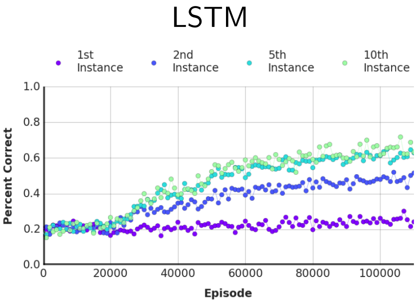
Write weight $\boldsymbol{w}_t^w$:

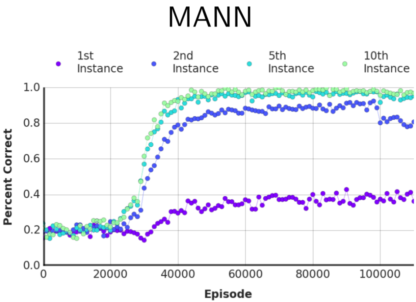
$$\boldsymbol{w}_t^w \leftarrow \sigma(\alpha) w_{t-1}^r + (1 - \sigma(\alpha)) w_{t-1}^{lu}$$

Writing to $\boldsymbol{M}$

$$\boldsymbol{M}_t(i) \leftarrow \boldsymbol{M}_{t-1}(i) + w_t^w(i) \boldsymbol{k}_t, \forall i$$

Omniglot Experiment Results





Omniglot Experiment Results

Table 1. Test-set classification accuracies for humans compared to machine algorithms trained on the Omniglot dataset, using one-hot encodings of labels and five classes presented per episode.

MODEL	INSTANCE (% CORRECT)					
	1ST	2ND	3RD	4TH	5TH	10TH
HUMAN	34.5	57.3	70.1	71.8	81.4	92.4
FEEDFORWARD	24.4	19.6	21.1	19.9	22.8	19.5
LSTM	24.4	49.5	55.3	61.0	63.6	62.5
MANN	**36.4**	**82.8**	**91.0**	**92.6**	**94.9**	**98.1**