One-shot Learning with Memory-Augmented Neural Networks

Idea

> Conventional gradient based NN has to learn the new parameter thus makes it inefficient to fit the new data. This article proposed to use MANN (Memory-Augmented Neural Network) to solve this problem due to that this model could encode fast and trace back previous information.
>
> The article shows that MANN could adapt to new data quickly and reach a very high accuracy under small data set.
>
> Not every NN with memory capacity is able to do meta-learning, it should meet below requirement
>
> 1. Comparatively reliable storing information
> 2. The storage size should not be limited by the size of model's parameter

Model

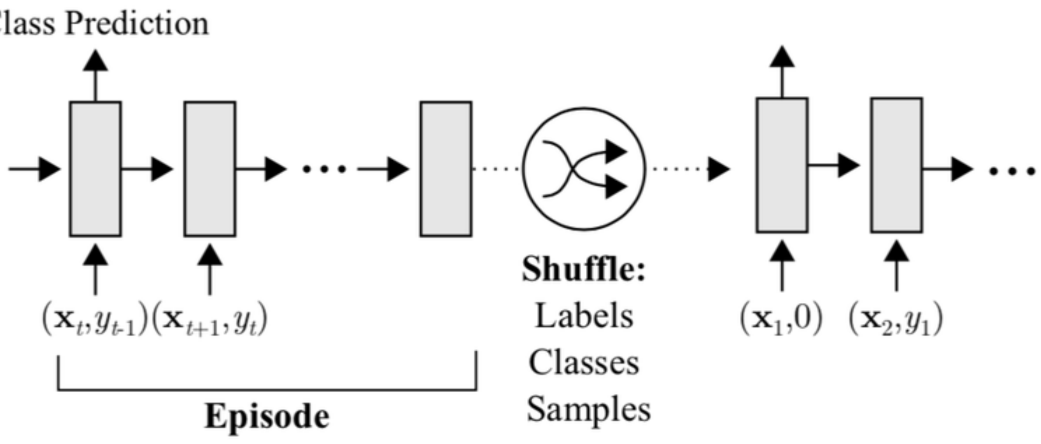
Method

Different from other studying tasks that optimized θ in minimizing $\mathcal{L}$, meta-learning is aiming at decreasing the expectation loss on specific set distribution $p(D)$:

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

The article uses the two following strategies:

1. Sequential Input, and each input accompanies with previous label (in case the model only learns the mapping relationship):
$$(x_1, null), (x_2, y_1), \ldots, (x_T, y_{T-1})$$
2. Between different sets, the labels are shuffled (have the model to preserve some sample information for next tracing)



Memory-Augment Model

Model this paper used is similar to NTM (Neural Turing Machine), applying forward feed network or LSTM as the controller. Each time it gets an input X_t, that value would be mapped into a new key K_t, which is used for soft reading (the cos similarity between key value and memory unit) or writing back to memory.

For writing to memory, it gave out an idea of LRUA (Least Recently Used Access), which is similar to LRU. Also, it used soft writing.

1. Get current weight by linear transforming reading weight W_t^r, writing weight W_t^w and the previous usage weight W_{t-1}^u

$$W_t^u \leftarrow \gamma W_{t-1}^u + W_t^r + W_t^w$$

2. Generating the 'Least Used Weight' (n is the time of reading memory, $m(\mathbf{v}, n)$ is the least element of $\mathbf{v}$)

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

3. Generating writing weight:

$$\mathbf{w}_t^w = \sigma(a)\mathbf{w}_{t-1}^w + (1 - \sigma(a))\mathbf{w}_{t-1}^w$$

4. Writing to the memory:

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