Model-based meta-learning in neural networks

Aluno: Willian Wang   Orientadora: Nina S. T. Hirata
Departamento de Ciência da Computação
Instituto de Matemática e Estatística - Universidade de São Paulo

Introduction

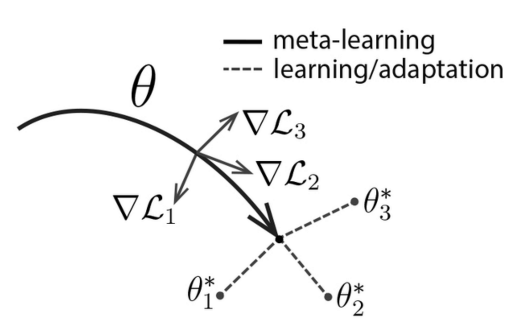
Traditional neural networks learn like a student memorizing an entire textbook: they need massive amounts of data and start from scratch each time. In contrast, humans can learn new skills from just a few examples - we don't need thousands of pictures to recognize a zebra or hundreds of attempts to learn a new game. Meta-learning aims to bridge this gap by teaching neural networks how to learn more efficiently, enabling them to master new tasks quickly with limited data, much like human learning.

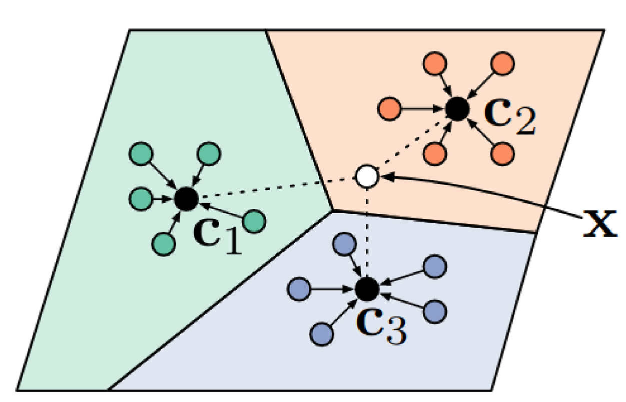
The meta-learning problem can be viewed as making a model to perform well over a task distribution $p(\mathcal{D})$, where each task has its own loss $\mathcal{L}$ and dataset $\mathcal{D}$:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}[\mathcal{L}_\theta(\mathcal{D})]$$

The field encompasses three main approaches:

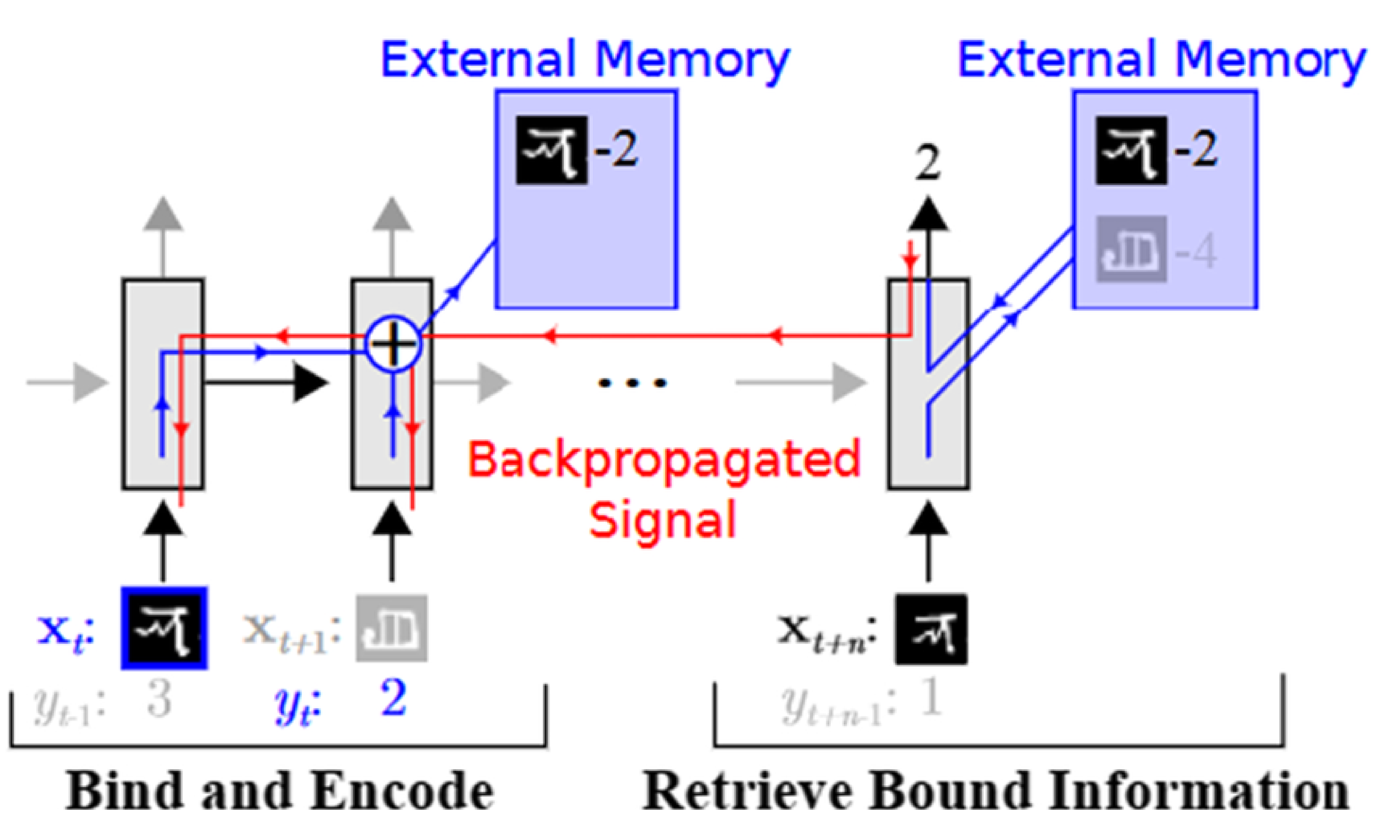
- **Metric-based**: learn a metric space where the distance between data points correlates with their label similarity, like Prototypical networks. They are computationally efficient but limited to classification.
- **Optimization-based**: try to optimize the backpropagation algorithm itself by treating it like a second-order optimization problem, like MAML. They are very expressive but are usually compute and memory intensive.
- **Model-based**: encodes the improved learning procedure within the model architecture. This work explores a few techniques in this subfield.



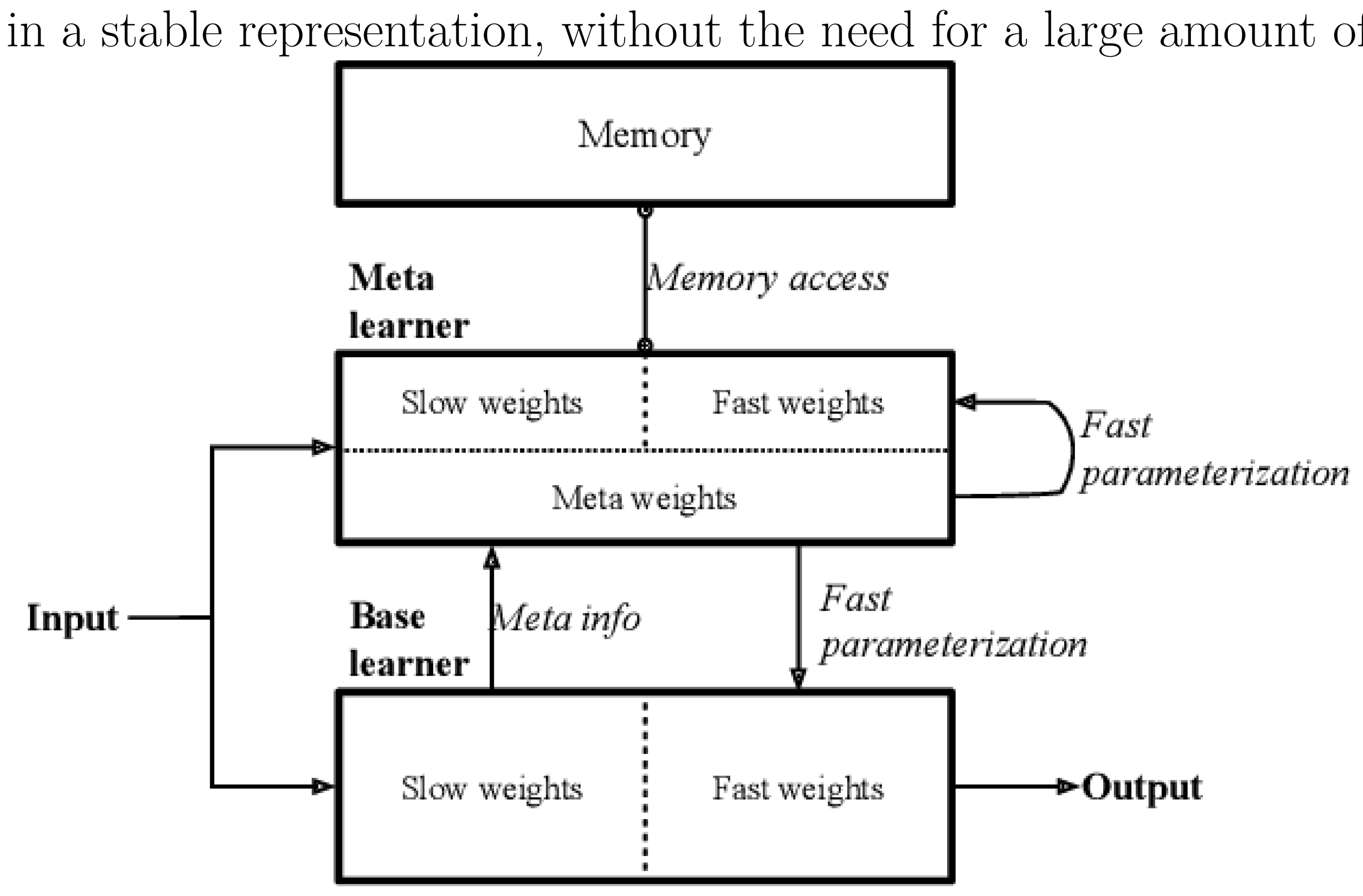


The left figure represents the model-agnostic meta-learning (MAML) algorithm, optimizing θ to quickly adapt to different tasks. The right figure represents a Prototypical network in a few-shot learning scenario.

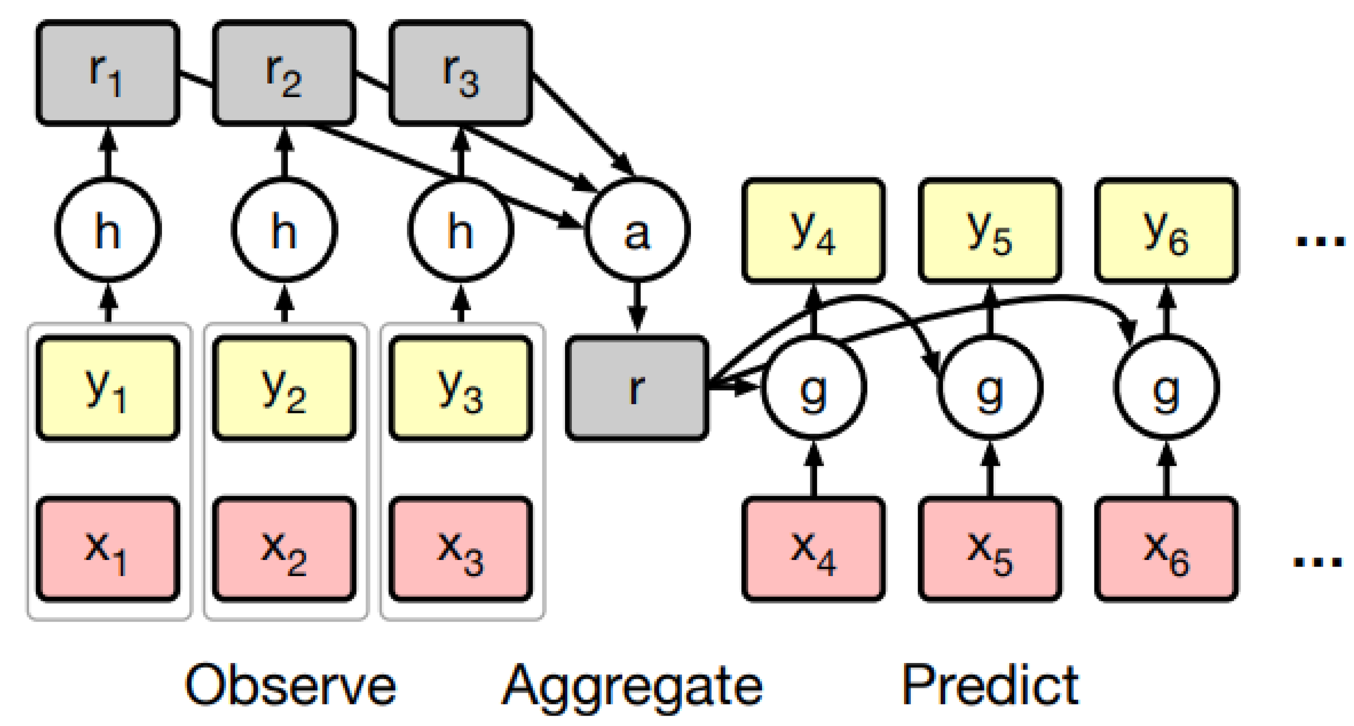
1 Architectures based on memory



Santoro et al., 2016 propose the use of Neural Turing Machines with explicit memory modules so information can be quickly encoded and decoded in a stable representation, without the need for a large amount of data.

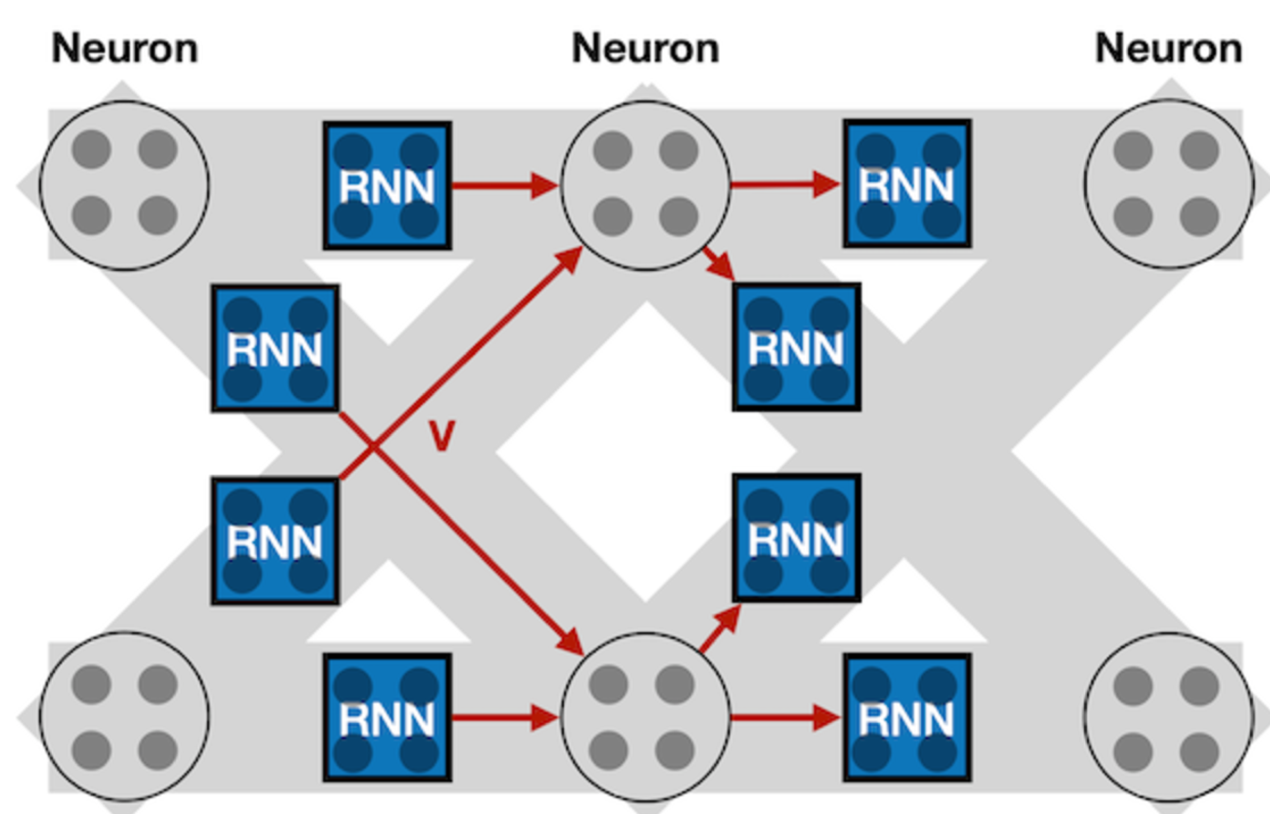


Munkhdalai and Yu, 2017 propose the use of fast weights, which have their parameters generated by another network called a meta learner, and slow weights, which are updated through ordinary gradient descent and have their losses used as feedback to improve the meta learner.

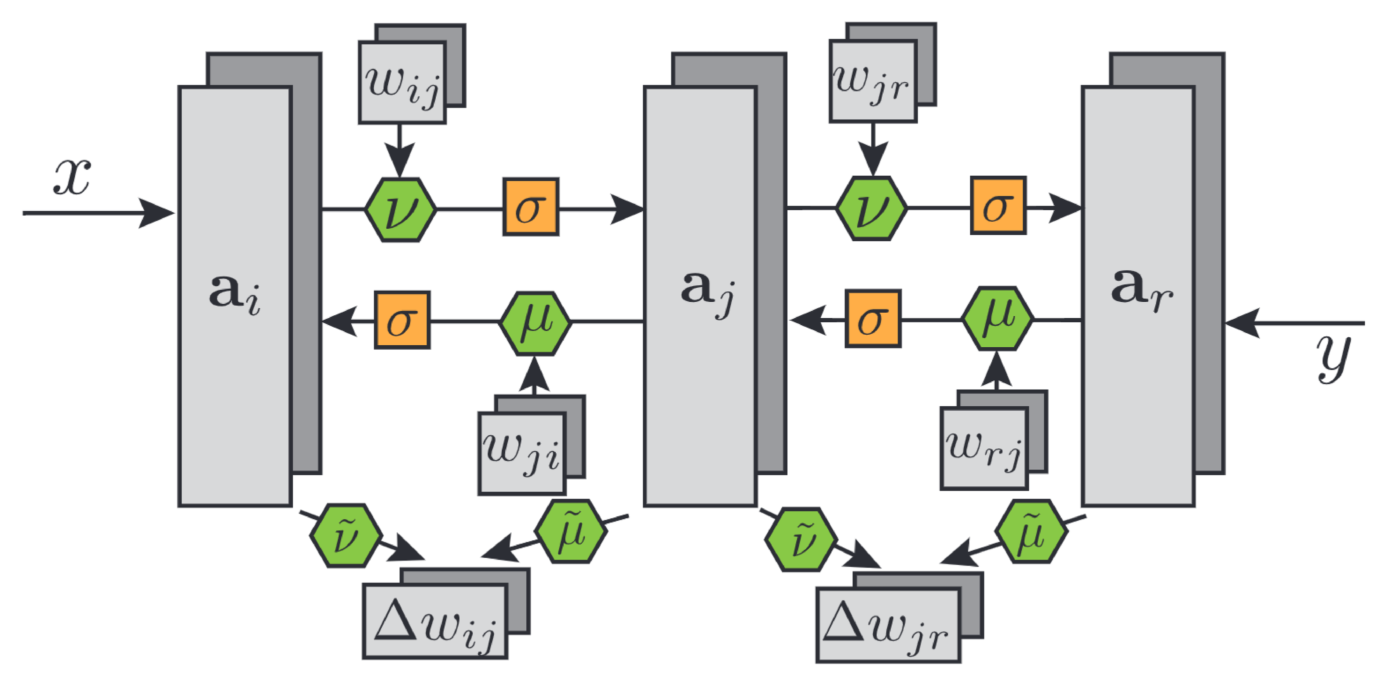


Garnelo et al., 2018 use the embeddings of the few examples and aggregate them to use as a context vector for the network that is going to solve the target task.

2 Learning alternatives to backpropagation



Kirsch and Schmidhuber, 2021 add more dimensions and reuse parameters of RNNs and change their connections in such a way that it can be viewed as a network where every synapse is an identical RNN. This new network creates properties for bidirectional flow of information and allows the network to emulate the backpropagation algorithm or even learn a more efficient one by just running it in forward mode.



Sandler et al., 2021, similar to the idea described above, propose a type of generalized neural network where both neurons and synapses maintain multiple states, making traditional neural networks a special case of it. The update rule is parametrized as a low-dimensional genome vector shared across the system.