Meta-Learning with Memory-Augmented Neural Networks

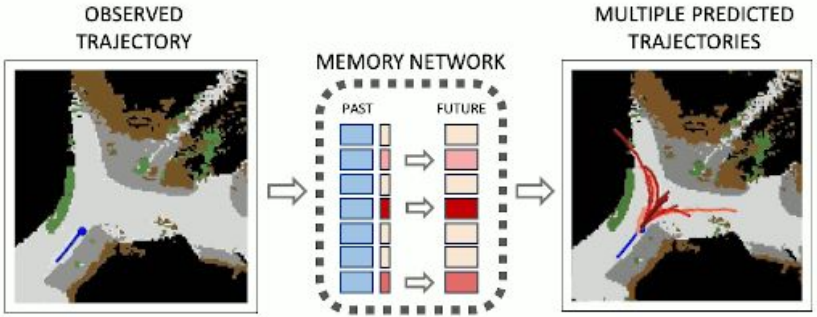
Santoro et. al.

Presenter: Amritha Haridasan

October 12 2023

Background

- Memory Augmented Neural Network(MANN) refers to the class of external memory-equipped network, in contrast to internal memory based architecture (such as LSTMs).
  -
- Memory-augmented neural networks, such as Neural Turing Machines (NTMs), offer the ability to quickly encode and retrieve new information, potentially overcoming the limitations of conventional models.



Motivation

- Rapid inference from small quantities of data and flexible adaptation are celebrated aspects of human learning.

Source: Fortune.com

- One-shot learning is a persistent challenge for traditional gradient-based networks, as they require extensive iterative training and relearning of parameters when encountering new data.

- Traditional gradient-based networks require a lot of data and inefficiently relearn their parameters when encountering new data, leading to catastrophic interference.

Motivation

- The proposed memory-augmented neural network demonstrates the ability to rapidly assimilate new data and make accurate predictions after only a few samples, potentially overcoming the challenges of one-shot learning.

Problem Setting

- **Objective: Develop memory-augmented neural networks for meta-learning, enabling rapid adaptation to new tasks.**

Task (or episode): A dataset of input-output pairs.
Memory: A data structure that can be used to store information. The memory is typically a matrix of real-valued numbers.
Read and write heads: Mechanisms that allow the network to access and modify memory. The read head reads a value from memory, and the write head writes a value to memory.
Controller: A neural network that interacts with memory to perform tasks. The controller is responsible for reading and writing to memory, and for generating outputs based on the contents of memory.

Formal Definition

Given a set of tasks T, each of which is a dataset of input-output pairs, train a model M that can generalize to new tasks after seeing only a few examples.

The model is evaluated on its ability to perform well on a set of test tasks that it has never seen before.

Prior Work

- **“Neural Turing Machines (NTMs)” by Graves et al. (2014)**
  - Introduced Neural Turing Machines, which combine neural networks with external memory, enabling them to perform algorithmic tasks such as copying and sorting.
- **"Memory Networks" by Weston et al. (2014)**
  - Presented Memory Networks, a class of neural networks with an external memory matrix, designed for question-answering tasks and language understanding.
- **"Meta-Learning with LSTM " by Hochreiter et al. (2001)**
  - Proposed a method for meta-learning using Long Short-Term Memory (LSTM) networks, enabling models to quickly adapt to new tasks with limited data.
- **"Learning to Learn by Gradient Descent by Gradient Descent" by Andrychowicz et al. (2016)**
  - Introduced an approach where a meta-learner learns to optimize the learning process of a model via gradient descent, improving the efficiency of training.

Limitations of Prior Work

- Overfitting: Many earlier works suffer from overfitting issues when dealing with limited data during meta-learning tasks.

- Complex Architectures: Some approaches use complex neural network architectures, making them computationally expensive and challenging to implement.

- Inefficiency: Some methods may not efficiently capture long-term dependencies or contextual information in memory.

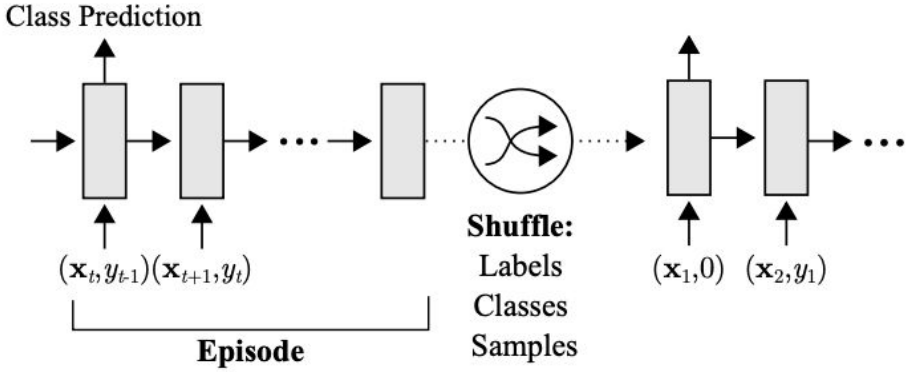
Proposed Approach

- **Learn to do classification on unseen class**
- **Learn the sample-class binding on memory instead of weights**
- **Let the weights learn higher level knowledge**

Model

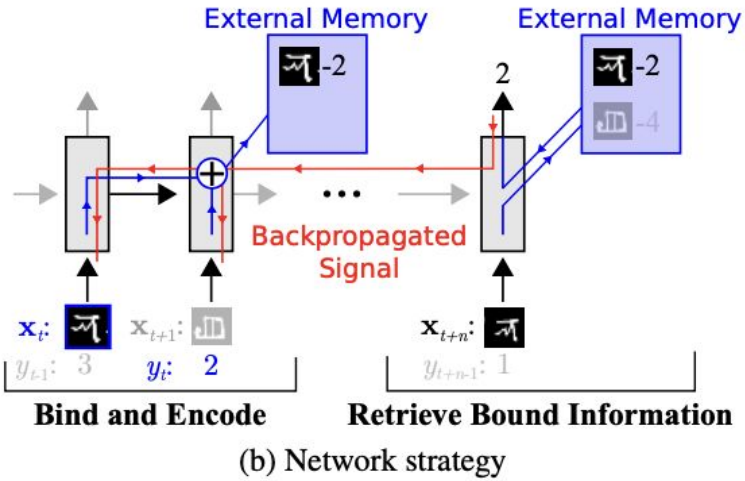
- y_t (label) is present in a temporally offset manner.

- Labels are shuffled from dataset-to-dataset. This prevents the network from slowly learning sample-class binding.



(a) Task setup

Model

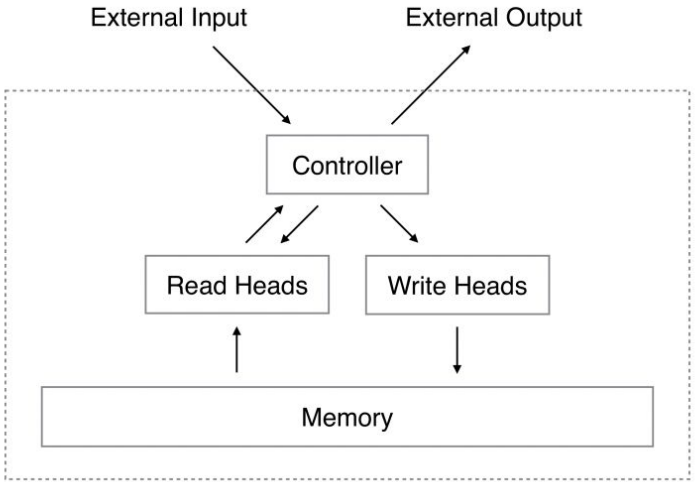


(b) Network strategy

- It must learn to hold data samples in memory until the appropriate labels are presented at the next time-step, after which sample-class information can be bound and stored for later use.

Model

- Basically the same as neural turing machine (NTM)
- Read from memory using the same content-based approach in NTM
- Write to memory using **Least Recent Used Access(LRUA)**
  - **LRUA: Content-based writer that writes memories to either the least used or most recently used memory location.**
-
- **Least**: Do you use this knowledge often?
- **Recent**: Do you just learn it?



source: Graves A, Wayne G, Danihelka I. Neural turing machines. arXiv preprint arXiv:1410.5401. 2014 Oct 20.

Content-based approach (MANN Reading)

- Controller produces the key

$$K\big(\mathbf{k}_t, \mathbf{M}_t(i)\big) = \frac{\mathbf{k}_t \cdot \mathbf{M}_t(i)}{\| \mathbf{k}_t \| \| \mathbf{M}_t(i) \|}$$

$$w_t^r(i) \leftarrow \frac{\exp\big(K\big(\mathbf{k}_t, \mathbf{M}_t(i)\big)\big)}{\sum_j \exp\big(K\big(\mathbf{k}_t, \mathbf{M}_t(j)\big)\big)}.$$

A memory, $\mathbf{r}_t$, is retrieved using this weight vector:

$$\mathbf{r}_t \leftarrow \sum_i w_t^r(i)\mathbf{M}_t(i).$$

Least Recent Used Access(LRUA) (MANN Writing)

- Usage weights w^u_t keep track of the locations most recently read or written to
- gamma is the decay parameter

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

- **least-used weights w^{lu}_t**
- $m(w^u_t\, m)$ denotes the n smallest element of the vector w^u_t
- here we set n equals to the number of read

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

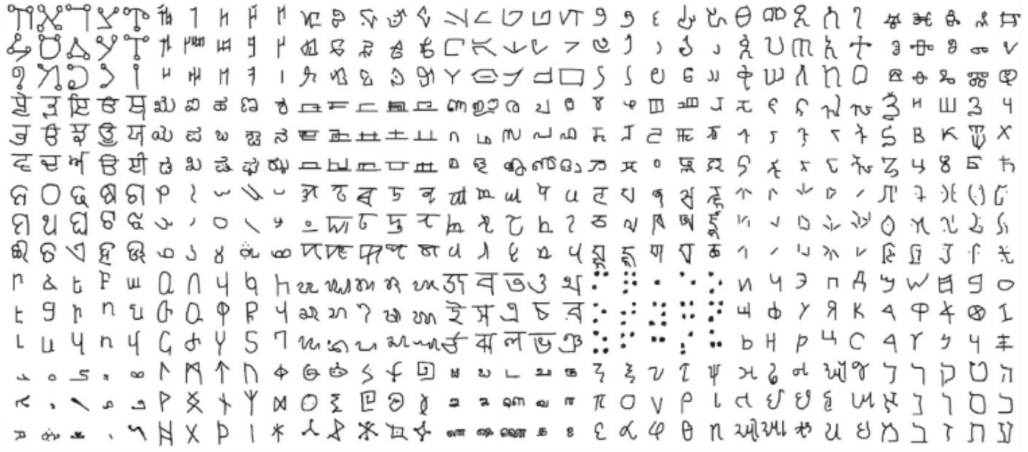
- **write weights w^w_t**
- alpha is a learnable parameter
- prior to writing to memory, the least used memory location is set to zero

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha))\mathbf{w}_{t-1}^{lu}.$$

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

Experimental Setup

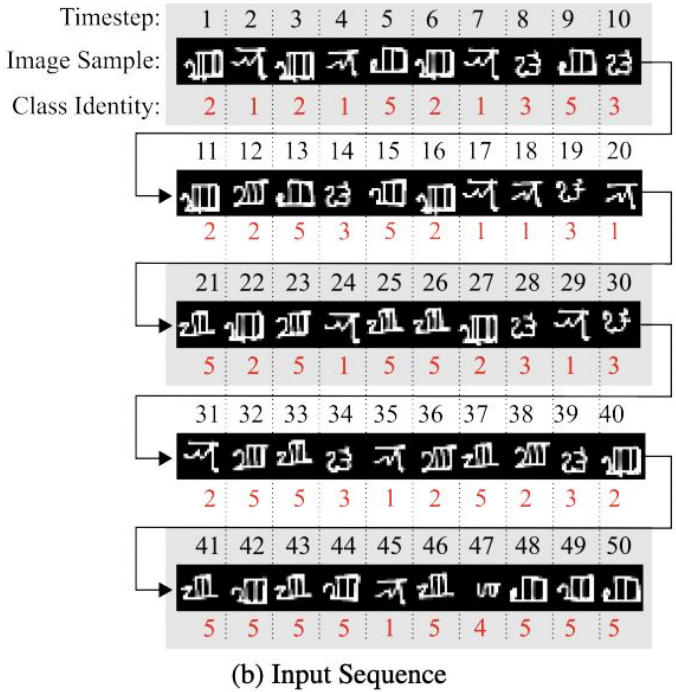
- **Dataset: Omniglot**
  - 1643 classes with only a few example per class(the transpose of MNIST)
  - 1200 training classes
  - 443 test classes



Source : https://paperswithcode.com/dataset/omniglot-1

Experimental Setup

- Train for 100,000 episodes, each episode with five randomly chosen classes with five randomly chosen labels, and 10 instances each.
- Test on never-seen classes



(b) Input Sequence

Figure 8. Example string label and input sequence.

Source: Meta-Learning with Memory-Augmented Neural Networks (Supplementary Material)

Experimental Results

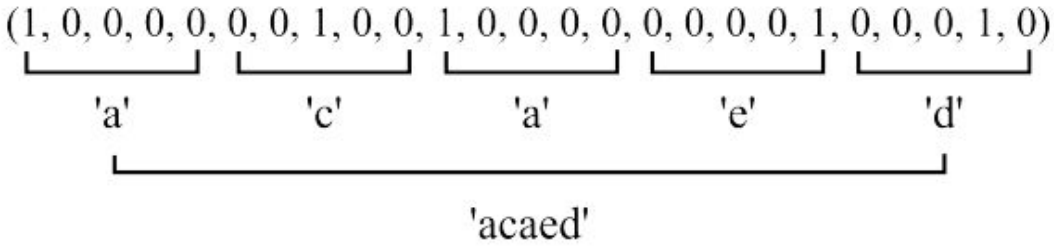
- **Machine v.s. Human**

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**

Class Representation

- A different approach for labeling classes was employed so that the number of classes presented in a given episode could be arbitrarily increased.
- Characters for each label were uniformly sampled from the set {'a', 'b', 'c', 'd', 'e'}, producing random strings such as 'ecdba'
- This combinatorial approach allows for 3125 possible labels, which is nearly twice the number of classes in the dataset.

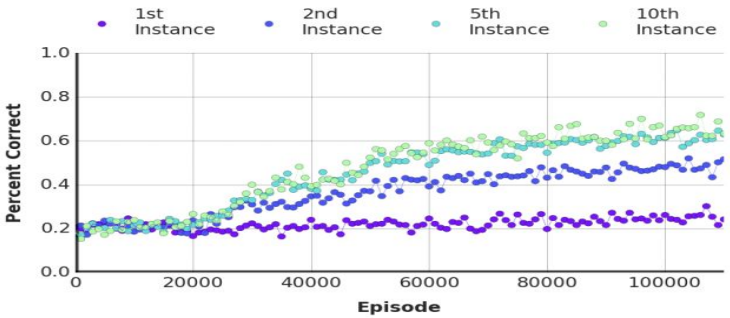


(a) String label encoded as five-hot vector

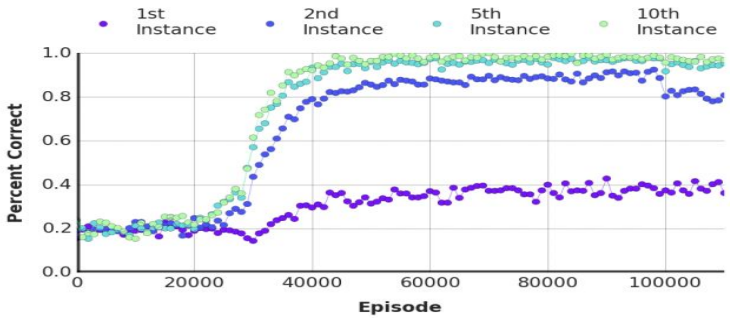
Experimental Results

5 classes/episode

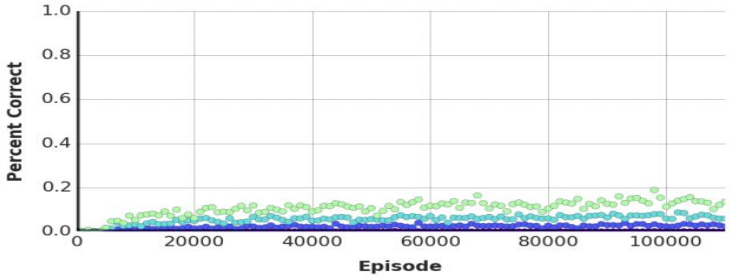
15 classes/episode



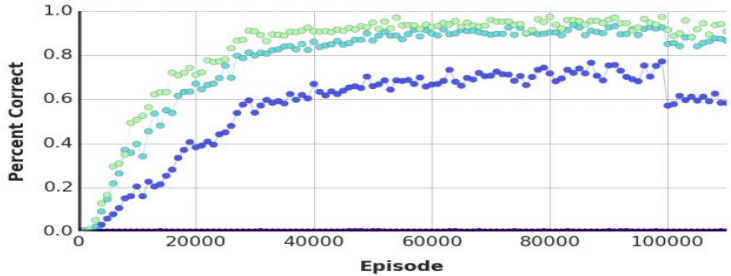
(a) LSTM, five random classes/episode, one-hot vector labels



(b) MANN, five random classes/episode, one-hot vector labels



(c) LSTM, fifteen classes/episode, five-character string labels



(d) MANN, fifteen classes/episode, five-character string labels

Figure 2. Omniglot classification. The network was given either five (a-b) or up to fifteen (c-d) random classes per episode, which were of length 50 or 100 respectively. Labels were one-hot vectors in (a-b), and five-character strings in (c-d). In (b), first instance accuracy is above chance, indicating that the MANN is performing "educated guesses" for new classes based on the classes it has already seen and stored in memory. In (c-d), first instance accuracy is poor, as is expected, since it must make a guess from 3125 random strings. Second instance accuracy, however, approaches 80% during training for the MANN (d). At the 100,000 episode mark the network was tested, without further learning, on distinct classes withheld from the training set, and exhibited comparable performance.

Experimental Results

- **Experiment with Different Algorithms**

Table 2. Test-set classification accuracies for various architectures on the Omniglot dataset after 100000 episodes of training, using five-character-long strings as labels. See the supplemental information for an explanation of 1st instance accuracies for the kNN classifier.

Model	Controller	# of Classes	Instance (% Correct)					
			1st	2nd	3rd	4th	5th	10th
kNN (raw pixels)	–	5	4.0	36.7	41.9	45.7	48.1	57.0
kNN (deep features)	–	5	4.0	51.9	61.0	66.3	69.3	77.5
Feedforward	–	5	0.0	0.2	0.0	0.2	0.0	0.0
LSTM	–	5	0.0	9.0	14.2	16.9	21.8	25.5
MANN	Feedforward	5	0.0	8.0	16.2	25.2	30.9	46.8
MANN	LSTM	5	0.0	**69.5**	**80.4**	**87.9**	**88.4**	**93.1**
kNN (raw pixels)	–	15	0.5	18.7	23.3	26.5	29.1	37.0
kNN (deep features)	–	15	0.4	32.7	41.2	47.1	50.6	60.0
Feedforward	–	15	0.0	0.1	0.0	0.0	0.0	0.0
LSTM	–	15	0.0	2.2	2.9	4.3	5.6	12.7
MANN (LRUA)	Feedforward	15	0.1	12.8	22.3	28.8	32.2	43.4
MANN (LRUA)	LSTM	15	0.1	**62.6**	**79.3**	**86.6**	**88.7**	**95.3**
MANN (NTM)	LSTM	15	0.0	35.4	61.2	71.7	77.7	88.4

Discussion of Results

- kNN(single nearest neighbour) has an unlimited amount of memory, and could automatically store and retrieve all previously seen examples.
- MANN outperforms kNN
- Using LSTM as controller is better than using feedforward NN

Experimental Results

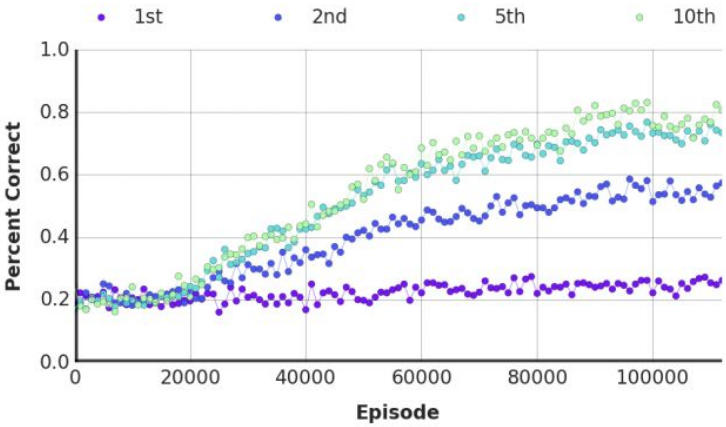
- **Experiment on Memory Wiping**

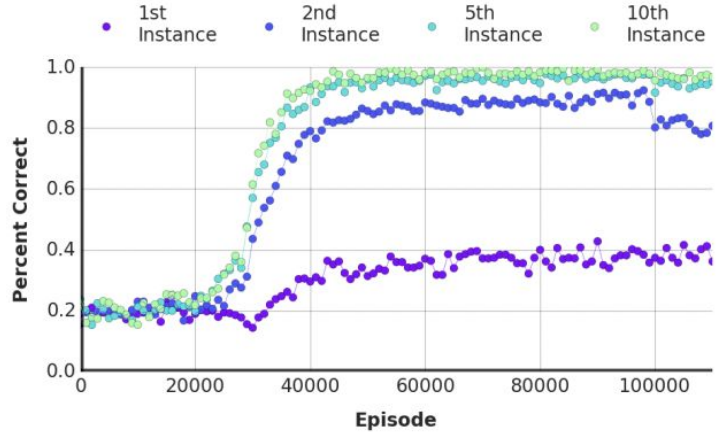
- A good strategy is to wipe the external memory from episode to episode, since each episode contains unique classes with unique labels.

Experimental Results

w/o wiping

with wiping

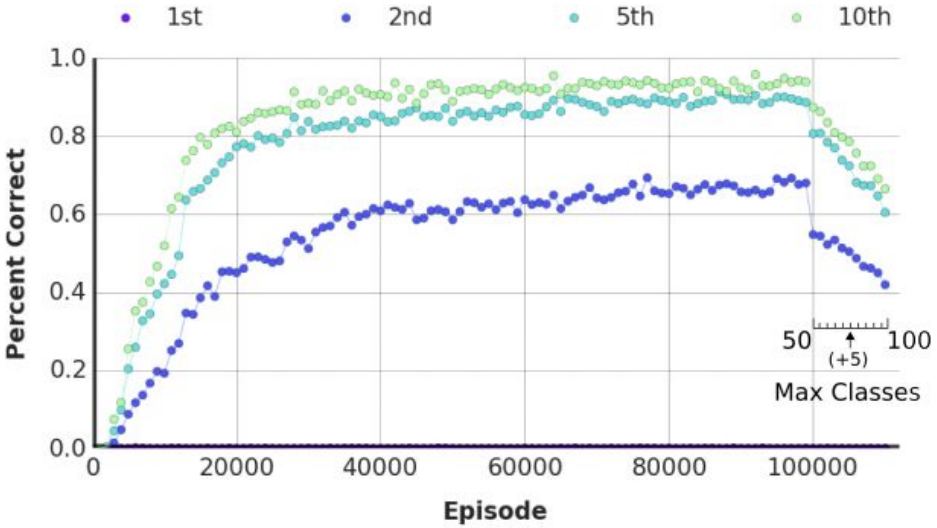




(b) MANN, five random classes/episode, one-hot vector labels

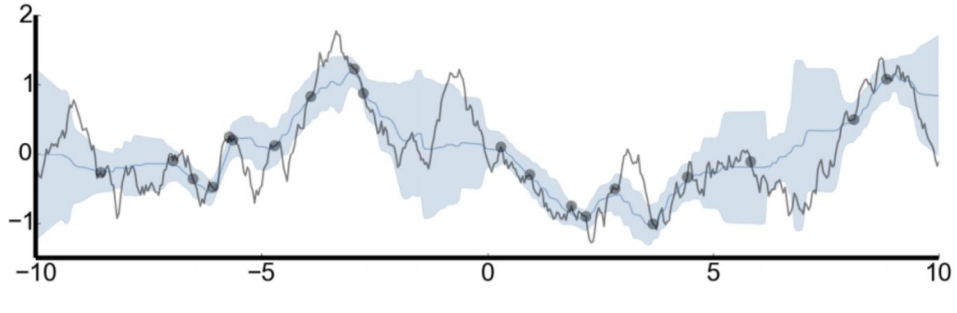
Experimental Results

- **Experiment on Curriculum Training**
  - gradually increase the classes per episode

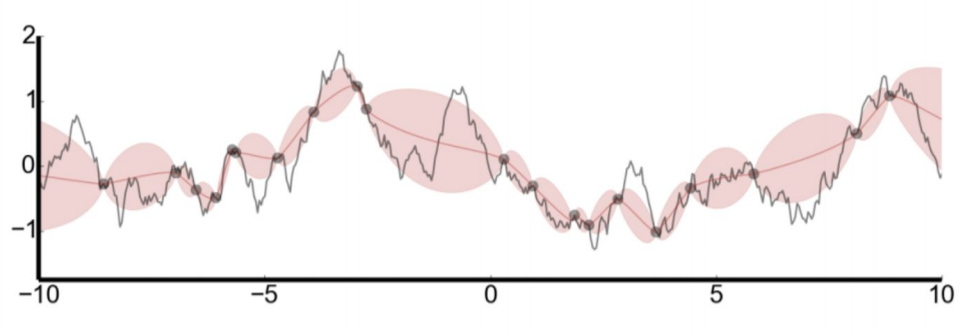


(a) One additional class per 10,000 episodes

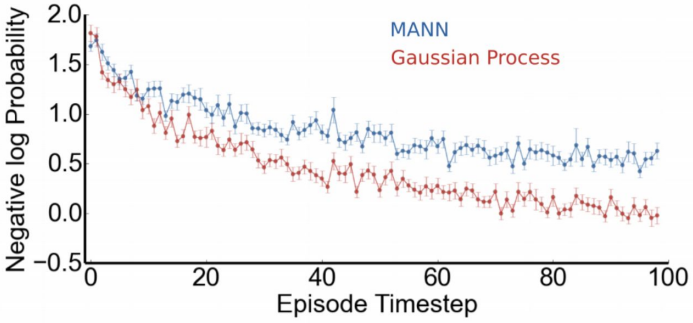
Experimental Results - Gaussian Process



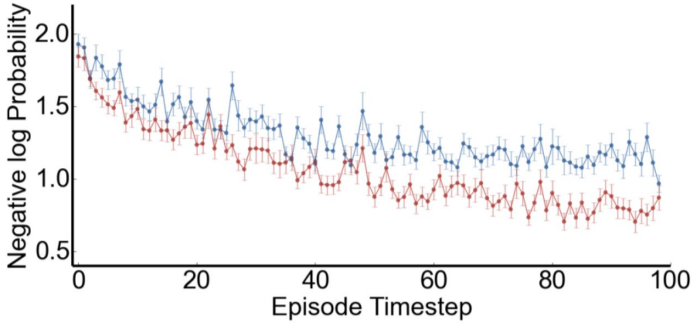
(a) MANN predictions along all x-inputs after 20 samples



(b) GP predictions along all x-inputs after 20 samples



(a) 2D regression log-likelihoods within an episode



(b) 3D regression log-likelihoods within an episode

Discussion of Results

- **Rapid Learning:** Demonstrates rapid assimilation of new data for accurate predictions with minimal samples.
- **Episodic Memory:** Utilizes episodic memory to efficiently store and retrieve past task information.
- **Outperformance:** Outperforms gradient-based networks in one-shot learning tasks with limited data.
- **Key-Value Memory:** Efficiently retrieves relevant information using key-value memory, enhancing task adaptation.
- **Enhanced Memory Access:** Introduces a novel memory access method focusing on content, further boosting performance.
- **Human-Like Learning:** Indicates potential to bridge the gap between machine and human learning with flexible adaptation and inductive transfer abilities.

Critique

- **Conceptual Misalignment:** The paper introduces Memory-Augmented Neural Networks (MANN) as meta-learners, but there's a misalignment in how "learning" is perceived. Learning is traditionally seen as model parameter updates with new data, while MANN focuses on data storage in external memory. This conceptual gap creates cognitive dissonance.

- **Short-Term Learning Ambiguity:** Storing data in external memory is framed as short-term learning in MANN. This raises questions about whether data storage equates to learning, and if so, how it's distinct from parameter updates.

- **Memory Wiping Impact:** The comparison between LRUA MANN and humans may not be entirely fair. The LRUA MANN's memory wipe is necessary for surpassing human performance, but the paper lacks clear statistics on the LRUA MANN's success without memory wiping in various setups. This omission affects the comprehensiveness of the evaluation.

Future Work

- **Optimal Memory-Addressing Procedures:** Developing efficient and effective memory-addressing mechanisms is crucial for enhancing the performance of Memory-Augmented Neural Networks (MANNs). This area of research holds the potential for significant advancements in MANN capabilities.

- **Learning to Learn:** Focusing on the concept of "learning to learn" in MANNs is a fundamental direction. This approach aims to improve the adaptability and generalization of MANNs in one-shot learning scenarios, which is a core goal of meta-learning.

- **Rapid Inference from Small Data:** Exploring the potential of MANNs in tasks that require quick and accurate inference from limited data is highly relevant. This direction highlights the practical applications of MANNs in real-world scenarios.

Extended Readings

- **Gemici, Mevlana, et al. "Generative temporal models with memory." arXiv preprint arXiv:1702.04649 (2017)**
- **Kaiser, Łukasz, et al. "Learning to remember rare events." arXiv preprint arXiv:1703.03129 (2017).**
- **Weston, Jason, Sumit Chopra, and Antoine Bordes. "Memory networks." arXiv preprint arXiv:1410.3916 (2014).**
- **Hospedales, Timothy, et al. "Meta-learning in neural networks: A survey." IEEE transactions on pattern analysis and machine intelligence 44.9 (2021): 5149-5169.**
- **Finn, Chelsea, and Sergey Levine. "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm." arXiv preprint arXiv:1710.11622 (2017).**

Summary

- Addressing the challenge of one-shot learning, where models must rapidly adapt to new tasks with minimal data.

- One-shot learning is vital in scenarios with limited data, mirroring human adaptability. However, traditional models struggle with this, necessitating memory-augmented approaches.

- Prior methods often lacked efficient memory retrieval and adaptation mechanisms, limiting their applicability.

- Memory-Augmented Neural Networks (MANN) employ an external memory module for rapid data storage and retrieval. This enables short-term adaptation akin to human learning.

- MANN showcases superior performance in one-shot learning tasks, bridging the gap between machine and human learning capabilities.

Discussion Questions

- How might the concept of meta-learning with memory-augmented neural networks impact the development of more advanced AI systems or applications?
- What are the advantages of using key-value memory in memory-augmented neural networks?
- Can memory-augmented neural networks be used as a model for investigating the computational basis of human meta-learning?
- How does the MANN with LRU Access compare to a MANN with a standard NTM access module in terms of performance?
- How does memory augmentation in neural networks improve task adaptation?
- What is the role of the episodic memory module in memory-augmented neural networks?