1 Task setup

1. There are 2 mains general ideas behind their setup: temporally extended presentation of labels and shuffling of labels.
2. Temporally extended presentation of labels: In their setup, the input into the model is x_t and y_{t-1}, the label of the previous input. Thus, the network sees a sequence of $(\mathbf{x}_1,\ \text{null}\ ), (\mathbf{x}_2, y_1), \ldots, (\mathbf{x}_T, y_T - 1)$.
3. The network is then tasked to output the appropriate label at the current time step, i.e. output y_t at timestep t.
4. Shuffling of labels: in between episode, the labels for the different datapoints are shuffled. This is to prevent the network from learning sample-class bindings in its weights.
5. With these two task setups, the intended behavior of the network is to:
6. Make a random guess the first time a sample from a new class is present
7. Learn a representation of this sample so that it can be bound to the class label, which is presented at the next step.
8. The next time another sample from the same class is presented, received the previously bound label, and returns it.

2 Architecture

2.1 Neural Turing Machine

1. A NTM is used as the model architecture.
2. A NTM has 2 main components, a controller (LSTM or FFN), and an external memory.
3. It can quickly write to and read from the external memory.
4. Thus, if the NTM is trained to store the sample-class binding in the external memory, and retrieve these bindings for predictions, it might be able to make accurate predictions for data it has only seen once.
5. Given an input x_t, the controller produces a key k_t, which is used either for read or write operation.
6. The external memory at time t is referred to as M_t. A row of the external memory is referred to as $M_t(i)$

2.2 Read operations to the external memory

1. When retrieving memory, the external memory is addressed using the cosine similarity measure:

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

2. which is used to compute a read-weight vector:

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

3. which is then used to retrieve a memory r_t:

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

4. which is then used by the controller as the input to a classifier (softmax output layer, or as an additional input for the next controller state).

2.3 Write operations

1. During a write operation, the key k_t produced by the controller is stored in a row of the memory matrix M_t.
2. The paper devises a new module, called Least Recently Used Access module to decide which row in the memory matrix to store k_t to.
3. The module has 2 functions, either:
4. write to rarely used locations to preserve recently encoded information, or
5. write to last used location, to update the memory with more relevant information.
6. The module produces a weighting between the previous row and the new key:

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

3 Experiments

1. They test their approach on classification and regression task.
2. They make an interesting argument about regression being harder than classification because in regression, the model needs to interpolate the function values between seen input-output pairs whereas in classification, it is more of a nearest neighbor search.

4 Questions

1. To what extend is the Least Recently Used Access module important? Is there an ablation studies? They demonstrated in the experiments section that some in some experiments, the LRU module leads to performance improvement.
2. The LRUA is argued to have two roles, writing to rarely used locations or to last used location. Is there an empirical study to demonstrate that these different behaviors were displayed at the opportune time? That is, when there are new information to encode, it writes to a rarely used location. And when there are relevant information to update an existing row of the memory, it updates the row?