Note that the key vector is clipped between $-1$ and 1. These sets of outputs along with the memory matrix and read and write weighting vectors at previous time step are used to update the read and write weighting vectors by making use of the addressing mechanism defined in section 4.2.3

$$P_t \leftarrow \sigma(\mathbf{W_{PC}} C_t + \mathbf{b_{PC}}) \tag{4.34}$$

$$\mathbf{k_{u,t}} \leftarrow \mathbf{W_{k_u C}} C_t + \mathbf{b_{k_u C}} \tag{4.35}$$

$$\beta_{u,t} \leftarrow relu(\mathbf{W}_{\beta_\mathbf{u}\mathbf{C}} C_t + \mathbf{b}_{\beta_\mathbf{u}\mathbf{C}}) \tag{4.36}$$

$$g_{u,t} \leftarrow \sigma(\mathbf{W_{g_u C}} C_t + \mathbf{b_{g_u C}}) \tag{4.37}$$

$$\mathbf{s_{u,t}} \leftarrow \sigma(\mathbf{W_{s_u C}} C_t + \mathbf{b_{s_u C}}) \tag{4.38}$$

$$\gamma_{u,t} \leftarrow relu(\mathbf{W}_{\gamma_\mathbf{u}\mathbf{C}} C_t + \mathbf{b}_{\gamma_\mathbf{u}\mathbf{C}}) \tag{4.39}$$

The updated hidden layer of neurons, then, also produces an erase and an add vector using equations 4.31 and 4.32 which, with the updated write weighting vector, are used to update the memory matrix. This process goes on until the input sequence gets exhausted. Once an external output $P_t$ has been produced for each element $X_t$ in the input sequence, an error or loss $Loss_t$ is calculated between the external outputs $P_t$ and the targets $Y_t$. Here, the loss function is chosen based on the task. Since we are assuming that the values in vectors of target sequence lie between 0 and 1, therefore, the binary cross entropy loss will be an appropriate choice.

$$Loss_t = binary\_crossentropy(P_t, Y_t) \tag{4.40}$$

Let $P$ be the matrix of predicted external output vectors $[P_1, P_2, \ldots P_T]$ and $Y$ be the matrix of target external output vectors $[Y_1, Y_2, \ldots Y_T]$, then one can compute the loss corresponding to the complete output sequence by computing the binary cross entropy error (refer section 3.2.3) between $P$ and $Y$.

$$overall\_loss, L = binary\_crossentropy(P, Y) \tag{4.41}$$

Now, the derivative of the *overall_loss* $L$ is computed with respect to each parameter of the model by making use of the chain rule of partial derivatives (refer section 3.2.4) and the derivative equations in this chapter. After computing the derivative of $L$ with respect to each parameter, the parameters of the model are updated using the RMSprop version of gradient descent as described in section 3.3.

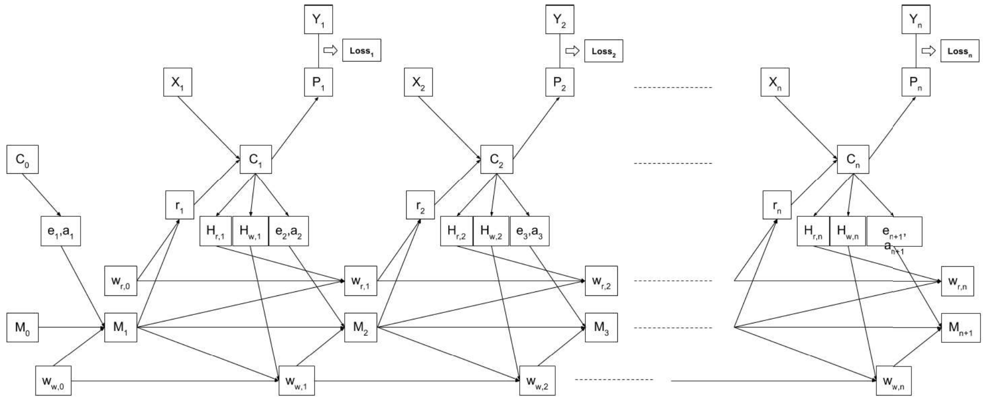

Figure 4.3: **Neural Turing Machine learning network.**

## 4.5 Experiments

We developed two versions of NTM, one with a single head which is used to read from as well as write to the memory and the second with two separate heads, one for reading from the memory and other for writing into the memory. We tested these two versions of NTM on tasks described below.

### 4.5.1 Copy Task

The memory matrix size was set to 128 × 20 i.e. 128 memory slots each of length 20. For training the first version of NTM, we took random binary sequences of length less than or equal to 20 and for the second version of NTM, we took binary sequences of length less than or equal to 5. One such input output sequence is shown in Fig. 4.4.

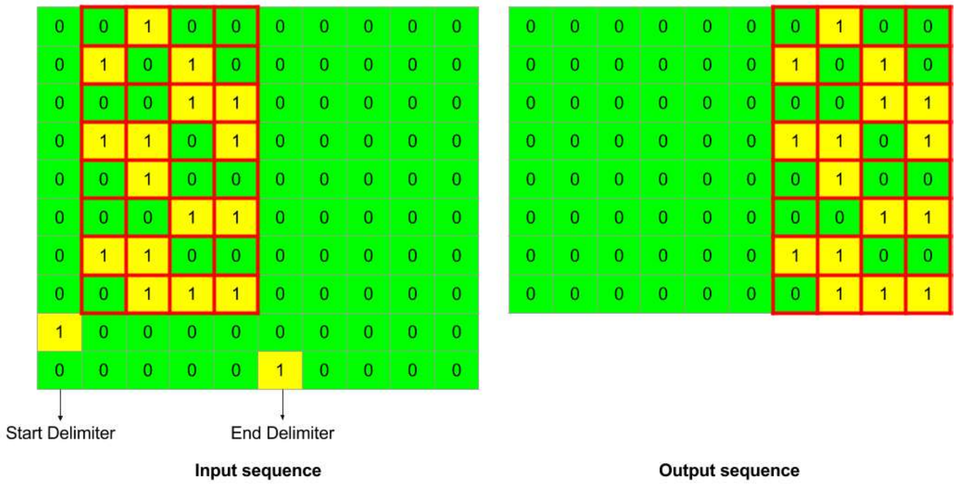

Figure 4.4: **Input and output sequence in copy task.**

Our first version of NTM was able to correctly predict the external output sequence with input sequences of length 116 and our second version of NTM was able to correctly predict the external output sequence with input sequences of length 34. Fig. 4.5 and Fig. 4.6 shows the learning curves in the two cases. Fig. 4.7 and Fig. 4.8 shows the corresponding plots which includes the visualization of the interaction between the controller and the memory matrix.

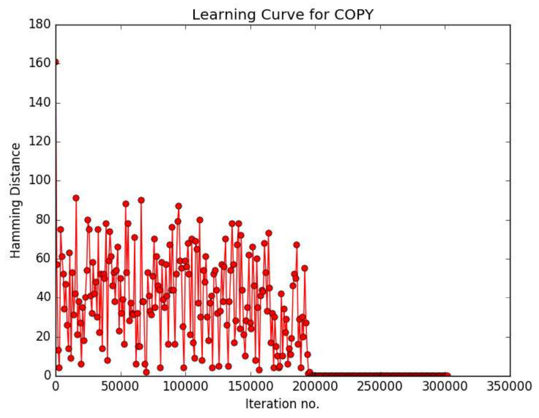

Figure 4.5: **Learning curve in copy task with our first version of NTM.**

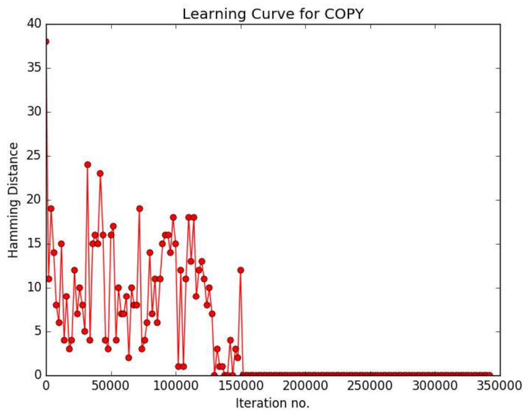

Figure 4.6: **Learning curve in copy task with our second version of NTM.**

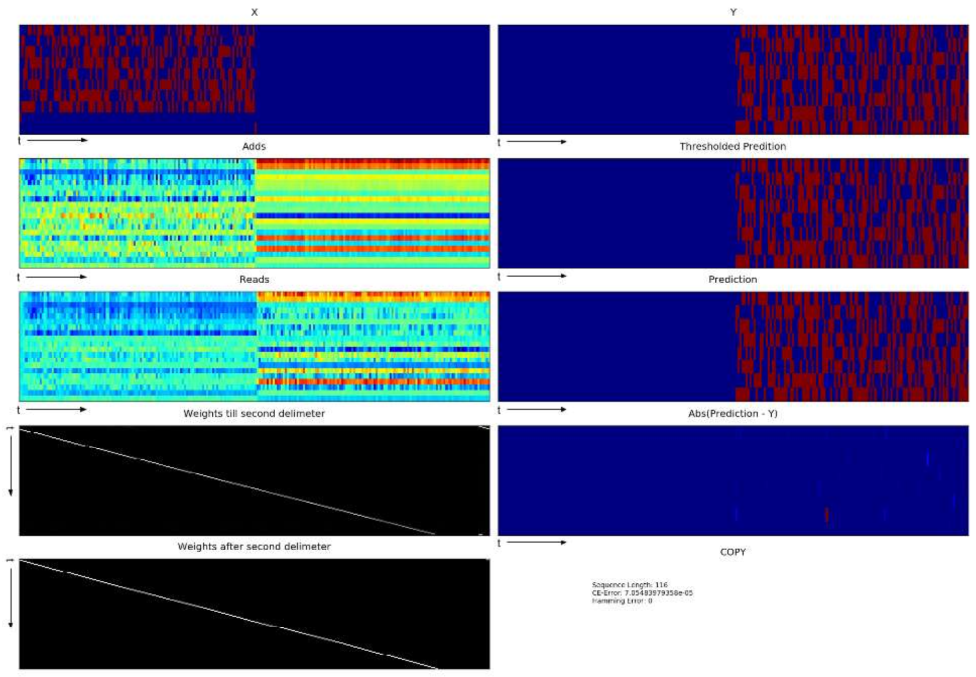

Figure 4.7: **Graphical visualization of copy task with first version of NTM.** External Input Sequence (X), Target Sequence (Y), Prediction Sequence (Prediction), Thresholded Prediction Sequence (Thresholded Prediction), Error (Abs(Prediction-Y)), Read vectors (Reads), Add vectors (Adds) and Weightings before and after ending delimiter

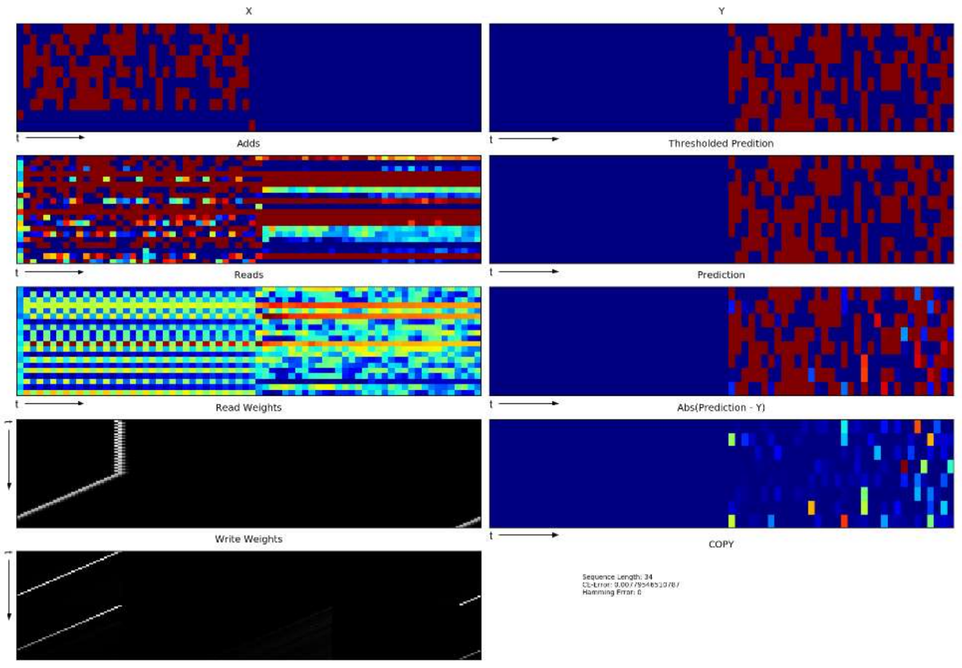

Figure 4.8: **Graphical visualization of copy task with second version of NTM.** External Input Sequence (X), Target Sequence (Y), Prediction Sequence (Prediction), Thresholded Prediction Sequence (Thresholded Prediction), Error (Abs(Prediction-Y)), Read vectors (Reads), Add vectors (Adds), Read Weighting vectors (Read Weights) and Write Weighting vectors (Write Weights)