

# 1 Model Structure

We implement the hidden Markov Model (HMM) directly using tensorflow using the tensor operation, i.e. `Einsum`. It is more efficient to train parameters of the testing HMM using tensor operations than solely using circuit operations, i.e. the TAC.

In this section, we will explain the procedure we use to evaluate the query  $\Pr(h_L \mid \Lambda_L)$  in a HMM. The inference procedure follows the forward pass of Viterbi algorithm. During each time step  $i$ , it keeps track a vector  $Q_i$ , and  $Q_i(n)$  represents the  $\Pr_i(h_i = n \mid \Lambda_i)$ . Therefore, the result of our query is represented by the vector  $Q_L$ . We can evaluate each  $Q_i$  inductively.

$$Q_0(n) = \frac{1}{\eta_0} \Pr(h_0 = n) \times M_{e_0 \rightarrow h_0}(n),$$

where  $M_{e_0 \rightarrow h_0}$  represents the message sent from evidence node  $e_0$  to hidden node  $h_0$ , which can be calculated using the emission probability,  $\Pr(e_0 \mid h_0)$ . For example, under the observation  $e_0 = 1$ , the message  $M_{e_0 \rightarrow h_0}(n) = \Pr(e_0 = 1 \mid h_0 = n)$ .  $\eta_0$  is simply the normalization constant to ensure that the vector  $Q_0$  represents a distribution.

For  $Q_i, i > 0$ , they can be evaluated inductively using

$$Q_i = \frac{1}{\eta_i} \text{Einsum}("l, lm, m \rightarrow m", Q_{i-1}, H, M_{e_i \rightarrow h_i})$$

If we want to construct an AC, we simply implement the tensor operation to calculate  $Q_L$ , while keeping the prior probability  $\Pr(h_0)$ , the transition probability  $\Pr(h_i \mid h_{i-1})$ , and the emission probability  $\Pr(e_i \mid h_i)$  be the network parameter.

If we want to construct a TAC, we will keep the prior probability and the emission probability as the model parameter. However, the transition probability  $\Pr(h_i \mid h_{i-1})$  will be selected at each time step  $i > 0$ , based on the  $Q_{i-1} = \Pr_{i-1}(h_{i-1} \mid \Lambda_{i-1})$ . To achieve this, we first models the logits of the transition probability using the introduced network parameters  $H_{\text{coeff}}$  and  $H_{\text{bias}}$ ,

$$L_{i+1|i} = \text{Einsum}("i, ij \rightarrow j", P_i, H_{\text{coeff}}) + H_{\text{bias}}.$$

Hence, the selected transition probability at each time step  $i > 0$  evaluates to

$$\Pr(h_i = m \mid h_{i-1} = l) = \frac{\exp(L_{i|i-1}(l, m))}{\sum_n \exp(L_{i|i-1}(l, n))},$$

## 2 AC vs TAC for HMM

In this section, we compare performance of two models, AC and TAC, under the query  $\Pr(h_L \mid \Lambda_L)$ . To train both models, we generate synthetic data from a second order HMM. We use (0.5, 0.5) as prior probability for  $h_0$ , and the emission

00	01	10	11	AC	TAC	00	01	10	11	AC	TAC
0	0	0	1	100%	100%	0	0	0	1	100%	100%
0	0	1	0	100%	100%	0	0	1	0	73%	100%
0	1	0	0	100%	100%	0	1	0	0	100%	100%
0	1	0	1	100%	100%	0	1	0	1	100%	100%
0	1	1	0	50%	100%	0	1	1	0	100%	100%
0	1	1	1	100%	100%	0	1	1	1	100%	100%
1	0	0	1	50%	100%	1	0	0	1	57%	100%
1	0	1	0	100%	100%	1	0	1	0	100%	100%
1	0	1	1	100%	100%	1	0	1	1	71%	100%
1	1	0	1	100%	100%	1	1	0	1	100%	100%
1	1	1	0	73%	100%	1	1	1	0	71%	100%

(a) Chain length 8
(b) Chain length 11

Table 1: Experiment Result

is perfect for every time step except the last one, i.e. the observation of each time step directly corresponds to the hidden state, except the observation of the last time step is always noisy 0.5, 0.5. For the transition probability, we try all possible deterministic transition and reported them as each row in the Table 1. For example, under the transition indicated by row 1 in Figure 1a, a possible sequence of the generated hidden states would be 01000000. The corresponding observation will be

$$o_0 = 0, o_1 = 1, o_2 = 0, o_3 = 0, o_4 = 0, o_5 = 0, o_6 = 0, o_7 = 0.5,$$

and the task to predict the last hidden state, which is 0 in this case.

To summarize, TAC can always perfectly fit the second order HMM under our experimental setups. At the same time, surprisingly, AC can also perfectly fit under many different transition cases. The appendix provides a case study which explains how an AC perfectly fit a second order HMM for the our prediction task, under a particular deterministic transition probability.

## A Model second order HMM using first order HMM

We consider a second-ordered HMM with length 4, and its transition is,

$$\begin{aligned}\Pr(h_i = 1 \mid h_{i-1} = 0, h_{i-2} = 0) &= 1 \\ \Pr(h_i = 1 \mid h_{i-1} = 0, h_{i-2} = 1) &= 0 \\ \Pr(h_i = 1 \mid h_{i-1} = 1, h_{i-2} = 0) &= 1 \\ \Pr(h_i = 1 \mid h_{i-1} = 1, h_{i-2} = 1) &= 0,\end{aligned}$$

i.e. the hidden state flip the bit every two steps. Further, we have a uniform prior probability over  $pr(H_0, H_1)$ , and exact observation,  $\Pr(o_i = 1 \mid h_i = 1) = 1, \Pr(o_i = 1 \mid h_i = 0) = 0$ .

We are interested in query  $\Pr(h_k = 1 \mid o_1 \cdots o_{k-1})$ . For this query, a first order HMM can fit it well with the following parameters.

$$\begin{aligned}\Pr(h_0 = 0) &= 0.5 \\ \Pr(h_i = 1 \mid h_{i-1} = 1) &= 10^{-12} \\ \Pr(h_i = 1 \mid h_{i-1} = 0) &= 1 - 10^{-12} \\ \Pr(o_i = 1 \mid h_i = 1) &= 10^{-6} \\ \Pr(o_i = 1 \mid h_i = 0) &= 1 - 10^{-6}\end{aligned}$$

Please note that the chain length matters.

input	label	$\Pr(h_0 = 1 \mid o_0)$	$\Pr(h_1 = 1 \mid o_0 o_1)$	$\Pr(h_2 = 1 \mid o_0 o_1 o_2)$	$\Pr(h_3 = 1 \mid o_0, o_1, o_2)$
0 0 1	1	$1 - 10^{-6}$	0.5	$10^{-6}$	$1 - 10^{-6}$
0 1 1	0	$1 - 10^{-6}$	$10^{-12}$	$1 - 10^{-6}$	$10^{-6}$
1 1 0	0	$10^{-6}$	0.5	$1 - 10^{-6}$	$10^{-6}$
1 0 0	1	$10^{-6}$	$1 - 10^{-12}$	$10^{-6}$	$1 - 10^{-6}$