Christian B. Mendl, Richard M. Milbradt

Tutorial 13 (Tensor networks for generative modeling)
Tensor networks are capable of representing certain stochastic probability models. We first consider a *Markov chain*, which describes a stochastic sequence of discrete events $(s_0, s_1, \dots s_T)$, such that the probability distribution of the next event s_{t+1} only depends on the current event s_t. Formally, this setup is expressed by a transition probability

$$\mathbb{P}(s_{t+1} = j | s_t = i) = p_{ij}.$$

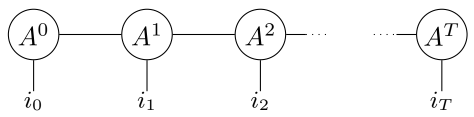
The left side is the conditional probability that the next event is j, given that the current event is i. The matrix $P = (p_{ij}) \in \mathbb{R}^{n \times n}$ is a stochastic matrix, i.e., it has non-negative entries and obeys $\sum_j p_{ij} = 1$ for all i. The initial event s_0 follows some prescribed probability distribution $p^{\text{init}} \in \mathbb{R}^n$.

In view of quantum computing, we now try to construct a so-called "Born machine": the squared entries $|\psi_j|^2$ of a quantum statevector ψ are interpreted as probabilities. For the Markov chain example, each entry corresponds to one possible sequence of events, and the squared amplitude to the probability that this sequence occurs:

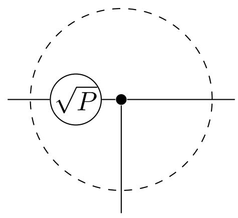
$$|\psi_{i_0,\dots,i_T}|^2 \stackrel{!}{=} \mathbb{P}(s_T = i_T, \dots, s_0 = i_0) = p^{\text{init}}_{i_0} \cdot p_{i_0,i_1} \cdot p_{i_1,i_2} \cdots p_{i_{T-1},i_T}.$$

Note that a quantum measurement of ψ is then equivalent to sampling a realization of the Markov chain.

(a) Consider a MPS Ansatz for ψ:

ψ =


Show that the MPS tensors A^t for $1 \le t < T$ drawn on the right with $\sqrt{P}$ defined entrywise, together with suitably chosen tensors A^0 and A^T, give rise to the desired ψ.

A^t =


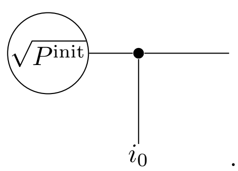
For the second part we discuss decision tree-like models.[1]

(b) Consider now a scenario where $T = 2$ and event s_0 results in two independent events s_1, each of which results in two independent s_2. Draw the tree diagram which represents this scenario.

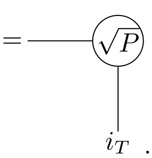
(c) It is possible to interpret the tree diagram as a tensor network where the leaves have a free index. Contracting the network would then result in the probability of the state $\{s_2^{(1)}, s_2^{(2)}, s_2^{(3)}, s_2^{(4)}\}$ occurring, without any knowledge of the states s_0 and s_1. Assume, for simplicity, that the probabilities for the events at each branch are the same and draw the tree tensor network, clearly defining the tensor at each node.

Solution

(a) First we need to define A^0 and A^T to be tensors in an appropriate form. Note that $p^{\text{init}}_{i_0}$ can be considered a vector. Therefore, we can just leave out the left leg:

$A^0_{i_0}$ =


Similarly to make A^T a 2-tensor, we can just leave out the δ-tensor:

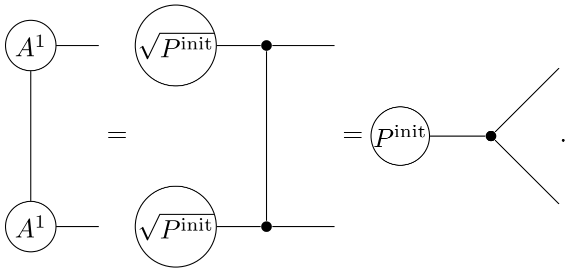
$A^T_{i_T}$ =


[1] S. Cheng, L. Wang, T. Xiang, P. Zhang: *Tree tensor networks for generative modeling*, PRB 99, 155131 (2019), arXiv:1901.02217

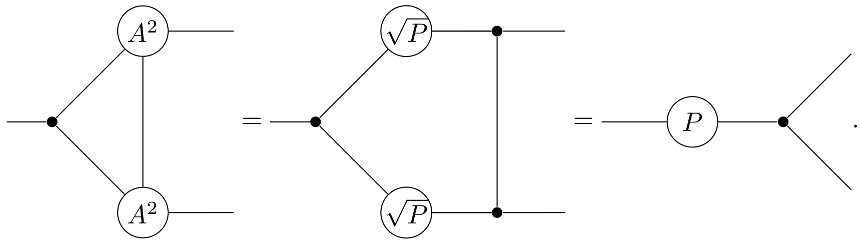
Since all elements of P are real $A^{s*} = A^s$. Now we compute the contraction.

$$|\psi_{i_0,\cdots,i_r}|^2 = \sum_{\vec{i}} \psi_{i_0,\cdots,i_r} \psi^*_{i_0,\cdots,i_r} = \langle\psi|\psi\rangle$$

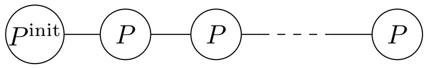
As a first step consider the contraction



Moving one site to the right we have the contraction

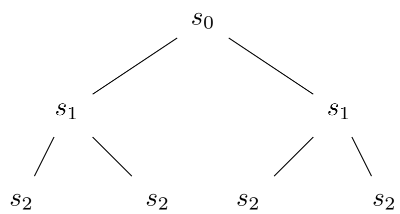


This will be true for all sites until the last, which can be contracted analogously. Eventually we get

$\langle\psi|\psi\rangle =$ 

$= p^{\text{init}}_{i_0} \cdot p_{i_0,i_1} \cdot p_{i_1,i_2} \cdots p_{i_{T-1},i_T}.$

(b) The resulting tree is



(c) Again there is a probability matrix P that we can use to construct the required tensors. The final tensor network is