1 Autoregressive Neural Networks

Consider the problem of encoding a probability distribution with an ANN. Are there any properties of the distribution at hand that can be exploited when trying to construct a suitable network structure?

It is a general result that any probability distribution $P(\vec{x})$ with $\vec{x} = (x_1, x_2, \ldots, x_N)^T$ can be rewritten as a product of conditional probabilities:

$$P(\vec{x}) = \prod_{i=1}^{N} p(x_i|x_{<i}), \tag{1}$$

where $p(x_i|x_{<i})$ is the conditional probability for outcome x_i given $x_{<i} = (x_1, \ldots, x_{i-1})$. For $i = 1$ we just have $p(x_1)$, independent of any other outcomes. It is possible to represent the joint $P(\vec{x})$ in terms of a collection of several networks that each encode a specific $P(x_i|x_{<i})$. If the networks encoding the conditional probabilities are all identical in terms of architecture and parameters, we refer to the total NN as a *recurrent neural network*. If we do not restrict each sub-network, the total NN is called an *autoregressive neural network*. Inputs to the network are POVM measurement outcomes (a_1, a_2), represented here by one-hot encoded vectors $v = (v_1, v_2, \ldots, v_8)^T$. Each network is a simple 1-layer feed-forward network with 4 inputs and softmax activation, which is defined according to

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2}$$

and automatically ensures normalization of the output.

2 2-qubit Systems

We will now concentrate on a specific architecture and describe how gates can – in princple – be implemented within that framework. The simplest system to describe that sttill allows one to evaluate the action of all fundamental gates is a 2-qubit system, described by the POVM distribution $P(\vec{a}) = P(a_1, a_2)$. This can be decomposed into $P(\vec{a}) = P(a_1)P(a_2|a_1)$. Considering that the effect of quantum gates in the POVM formalism is just to sum over probabilities allows the identification of analytical update rules for arbitrary gates.

2.1 1-qubit Gates

The effect of gates that only act on single qubits is that the POVM distribution after that gate consists of the weighted sum of all old probabilities that differ from the post-gate outcome only for the qubit on which the gate has acted. Without loss of generality it can be assumed that this is qubit 2. Then the gate acts according to

$$P'(a_1, a_2) = O_{b_2}^{a_1 a_2} P(a_1, b_2) \tag{3}$$

Here $O_{b_2}^{a_1 a_2}$ denotes the POVM-operator representation of the quantum gate.

3 Larger Systems

Consider, as the simplest non-trivial[1] example a three-qubit system acted upon by a single-qubit gate on the third qubit. New probabilities are give by

$$\begin{aligned} P'(a_1, a_2, a_3) &= \mathcal{O}^{a_3 a_2 a_1}_{b_3 b_2 b_1} \delta_{(a_2, b_2)} \delta_{(a_1, b_1)} P(b_1, b_2, b_3) \\ &= \mathcal{O}^{a_3}_{b_3} P(a_1, a_2, b_3) \end{aligned}$$

The autoregressive property ensures that the conditional probabilities, in which the joint probability can be decomposed, are implemented by seperate neural networks. In probabilities, the following conditions hold

$$\begin{aligned} P(a_1, a_2, a_3) &= P(a_1) P(a_2|a_1) P(a_3|a_2 a_1), \\ P'(a_1, a_2, a_3) &= P(a_1) P(a_2|a_1) P'(a_3|a_2 a_1). \end{aligned}$$

Hence we can write

$$P'(a_3|a_2 a_1) = \mathcal{O}^{a_3}_{b_3} P(b_3|a_2 a_1) \tag{4}$$

Assuming the conditional probabilites for the measurement outcome on qubit three are encooded by 1-layer feed-forward networks with softmax activation function allows us to rewrite 4 as

$$\frac{\exp\left(\beta'^3_{a_3} + \gamma'^{(23)}_{a_3 a_2} + \gamma'^{(13)}_{a_3 a_1}\right)}{\sum_{i,j,k} \exp\left(\beta'^3_i \gamma'^{(23)}_{ij} + \gamma'^{(13)}_{i,}\right)} = \sum_{b_3} \mathcal{O}^{a_3}_{b_3} \frac{\exp\left(\beta'^3_{b_3} + \gamma'^{(23)}_{b_3 a_2} + \gamma'^{(13)}_{b_3 a_1}\right)}{\sum_{i,j,k} \exp\left(\beta'^3_i \gamma'^{(23)}_{ij} + \gamma'^{(13)}_{i,}\right)}$$

Ignoring the LHS normlization, we can solve for $\{\beta'^3, \gamma'^{(23)}, \gamma'^{(13)}\}$. This gives

$$\beta'^3 + \gamma'^{(23)} + \gamma'^{(13)} = \ln\left(\sum_{b_3} \mathcal{O}^{a_3}_{b_3} \frac{\exp\left(\beta'^3_{b_3} + \gamma'^{(23)}_{b_3 a_2} + \gamma'^{(13)}_{b_3 a_1}\right)}{\sum_{i,j,k} \exp\left(\beta'^3_i \gamma'^{(23)}_{ij} + \gamma'^{(13)}_{i,}\right)}\right) \tag{5}$$

We now see that a fundamental formal problem arises: The new biases and weight matrices $\{\beta'^3, \gamma'^{(23)}, \gamma'^{(13)}\}$ all can be though of as functions mapping an input (a_1, a_2, a_3) to some number. The fact that the parameters are summed however indicates a function of the structure

$$\phi = f(a_3) + g(a_3, a_2) + h(a_3, a_1). \tag{6}$$

Because of this structure the function can only depend on 32 distinct values (no $a_1 a_2$-coupling). But the right hand side in general is of the form

$$\phi' = f'(a_1, a_2, a_3) \tag{7}$$

This discrete function depends on *3-tuples*, whereas ϕ only depends on pairs of *2-tuples*. In general, ϕ' can take more distinct values than ϕ, hence justifying that there cannot exist some $\{\beta'^3, \gamma'^{(23)}, \gamma'^{(13)}\}$ such that $\phi = \phi'$.

[1] not overparametrized

How does this agree with the fact that for 2-qubit systems correct updates can be found? This is because for 2-qubit systems we just have 16 different (a_1, a_2)-pairs which can be explicitly considered since they all correspond to entries in the weight matrix $\gamma'^{(12)}$. For bigger systems, we can use the same approach to encode probabilities if we replace weight matrices by *weight tensors*. However, since these scale exponentially with system size (all possible outcomes need to be considered), this approach is not useful for the efficient simulation of quantum algorithms with neural networks.