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. To determine a general update rule, product states and entangled states need to be distinguished.

2.1.1 Product States

For product states $P(a_1, a_2) = P(a_1)P(a_2)$ holds.

3 Not Every single-Qubit Gate can be implemented via Bias Updates alone

In this discussion we will consider the entangled GHZ state for two qubits, also known as the Bell state:

$$|\Psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Suppose we want to describe it in the POVM formalism. Let our IC-POVM be the **tetrahedral POVM**:

$$M_{\text{tetra}} = \left\{ \frac{1}{4}(1 + \vec{s}^{(a)} \cdot \vec{\sigma})_{a \in \{0,1,2,3\}} \right\} \tag{4}$$

given by the vectors

$$\begin{aligned}
\vec{s}^{(0)} &= (0, 0, 1)^T \\
\vec{s}^{(1)} &= \left(\frac{2\sqrt{2}}{3}, 0, -\frac{1}{3} \right)^T \\
\vec{s}^{(2)} &= \left(-\frac{\sqrt{2}}{3}, \sqrt{\frac{2}{3}}, -\frac{1}{3} \right)^T \\
\vec{s}^{(3)} &= \left(-\frac{\sqrt{2}}{3}, -\sqrt{\frac{2}{3}}, -\frac{1}{3} \right)^T .
\end{aligned}$$

For practical reasons we will work with a slightly rotated ($\theta = 0.03, \phi = 0.02$) POVM basis such that all POVM probabilities are non-zero and can hence be implemented by finite biases and weights in our ANN. This is necessary because probabilities are encoded according to

$$P(a_1, a_2) = P(a_1)P(a_2|a_1) = \frac{\exp\{\beta_{a_1}^1\}}{\sum_i \exp\{\beta_i^1\}} \frac{\exp\{\beta_{a_2}^2 + \gamma_{a_2 a_1}\}}{\sum_j \exp\{\beta_j^2 + \gamma_{j a_1}\}} \tag{5}$$

where we have chosen softmax normalization of the probabilities. Vanishing probabilities would lead to infinite parameters.

Let us consider a standard 1-qubit gate, the hadamard gate H, that acts upon the second qubit. In the computational basis its action is given by

$$H = \frac{1}{\sqrt{2}} 111{-}1. \tag{6}$$

It is straightforward to formulate this gate in the POVM formalism according to the rule

$$H'^{(\vec{a},\vec{b})} = \mathrm{Tr}\Big\{ H M^{(\vec{a})} M^{(\vec{b'})} (T^{-1})^{(\vec{b'}\vec{b})} \Big\}$$

where $\vec{a}, \vec{b}, \vec{b'}$ are vectors denoting the individual POVM outcomes for each qubit for a specific POVM measurement and the summation over $\vec{b'}$ is implicitly assumed. Assume that our Bell state can be described by the POVM distribution $P(\vec{a})$ before the H-gate. Immediately after the gate, our 2-qubit state is described by the new distribution $P'(\vec{a})$ which is related to the old one according to

$$P'(\vec{a}) = H'^{(\vec{a},\vec{b})} P(\vec{b}). \tag{7}$$

These probability distributions are parametrized by our autoregressive neural network in terms of the network parameters $\{\beta^1, \beta^2, \gamma\}$ according to the rule 5. Here β^1, β^2 are 4-entry vectors and γ is a 4x4 matrix. Suppose P' can be described with the parameters $\{\beta^1, \beta^{'2}, \gamma\}$, i.e. only the biases corresponding to qubit 2 change. We will show that there does not exist such a $\beta^{'2} = f(\{\beta^1, \beta^2, \gamma\})$ that is related to the old parameters via some analytical update rule f for the specific case of the Hadamard gate acting upon the bell state.

Numerical values for the initial POVM distribution can easily calculated. For our Bell state and rotated POVM choice we find

$$P = \begin{pmatrix} 0.12488753 \\ 0.04177482 \\ 0.04472824 \\ 0.03860941 \\ 0.04177482 \\ 0.124896 \\ 0.03872253 \\ 0.04460665 \\ 0.04472824 \\ 0.03872253 \\ 0.04165755 \\ 0.12489168 \\ 0.03860941 \\ 0.04460665 \\ 0.12489168 \\ 0.04189225 \end{pmatrix}$$

The network parameters that parametrize this distribution can be found analytically. To do so, we first consider the probabilities $P(a_1)$ for the POVM

outcome measured on the first qubit. This is given by

$$P(a_1) = \sum_{a_2} P(a_1, a_2)$$

which inherits normalization from the POVM distribution so that $\sum_{a_1} P(a_1) = 1$ is indeed fulfilled. We can now simply associate

$$\beta^1_{a_1} = \log\left(\sum_{a_2} P(a_1, a_2)\right),$$

where the RHS can be obtained by summing over the entries in the probability vector for the *combined* outcomes (a_1, a_2) in groups of four probabilities. The normalization constant from 5 was excluded in this consideration here as it is identical for all biases, and contributes only to a constant offset which gets cancelled out by the softmax normalization anyways (can be factored out in denominator). With $\log 0.25 = -1.3863$ we obtain biases

$$\beta^1 = (-1.3863,\ -1.3863,\ -1.3863,\ -1.3863)^T.$$

As we have enough free parameters to completely characterise the measurement distribution, we can set the biases $\beta^2 = 0$, but in general allow them to attain a non-zero value after updating them to account for the action of the gate. By recognizing $P(a_1, a_2) = P(a_2|a_1)P(a_1)$ and remembering that conditional probabilities are encoded according to 5 we can write

$$\frac{\exp\{\gamma_{a_2 a_1}\}}{\sum_j \exp\{\gamma_{j a_1}\}} = P(a_2|a_1) = \frac{P(a_1, a_2)}{P(a_1)}$$

we can determine

$$\gamma_{a_2 a_1} = \log\left(\frac{P(a_1, a_2)}{P(a_1)}\right)$$

This yields

$$\gamma = \begin{pmatrix} -0.69404732 & -1.78916718 & -1.72085591 & -1.86796488 \\ -1.78916718 & -0.69397956 & -1.86503928 & -1.72357789 \\ -1.72085591 & -1.86503928 & -1.79197831 & -0.6940141 \\ -1.86796488 & -1.72357789 & -0.6940141 & -1.78635998 \end{pmatrix}$$

Now let's assume $P'(a_1, a_2)$ can be represented by $\{\beta^{(1)}, \beta'^{(2)}, \gamma\}$. This means

$$\frac{\exp\left\{\beta^{(2)'}_{a_2} + \gamma_{a_2 a_1}\right\}}{\sum_j \exp\left\{\beta'^{(2)}_j + \gamma_{j a_1}\right\}} = \hat{O}^{a_2 b_2} \frac{\exp\left\{\beta^{(2)}_{b_2} + \gamma_{b_2 a_1}\right\}}{\sum_j \exp\left\{\beta^{(2)}_j + \gamma_{j a_1}\right\}}, \tag{8}$$

where we implicitly summed over b_2 and excluded $P(a_1)$ because it isn't of interest for our purposes. This can be simplified to

$$\beta_{a_2}^{(2)'} = \log\left(\hat{O}^{a_2 b_2} \frac{\exp\left\{\beta_{b_2}^{(2)} + \gamma_{b_2 a_1}\right\}}{\sum_j \exp\left\{\beta_j^{(2)} + \gamma_{j a_1}\right\}}\right) - \gamma_{a_2 a_1} \tag{9}$$

$$= \log\left(\hat{O}^{a_2 b_2} \frac{\exp\{\gamma_{b_2 a_1}\}}{\sum_j \exp\{\gamma_{j a_1}\}}\right) - \gamma_{a_2 a_1}. \tag{10}$$

Here we used that our initial biases for the second qubit were all zero. Going from 8 to 10, we excluded the normalization of the new probabilities. This however is not an issue, since the normalization just corresponds to an identical offset for all biases $\beta^{'(2)}$ after applying the log. This offset however cancels out when we later apply the softmax operation, as it can be factored out in both numerator and denominator. Since we made the restriction that $\beta^{(1)}$ and γ should not be affected by the gate, the only unknown in this equation is $\beta^{'(2)}$. For a fixed $a_2 \in \{0, 1, 2, 3\}$, the above relation has to hold for all $a_1 \in \{0, 1, 2, 3\}$. This means we have 4 equations that each contain only contain $\beta^{'(2)}$ as a free parameter and that need to be fulfilled simultaneously. Restricting ourselves to the case $a_2 = 0$, we have four equations that $\beta_0^{'(2)}$ needs to fulfill. For $a_1 = 0$ we obtain:

$$\beta_0^{(2)'} = -0.69135$$

For $a_1 = 1$, however, 10 leads to

$$\beta_0^{(2)'} = 1.06625$$

which clearly contradicts the result obtained for $a_1 = 0$. But by assumption the bias $\beta_{a_2}^{(2)'}$ should attain a value independent of the outcome a_1. This condition however leads to contradicting results. Hence, in general, to encode the action of single-qubit gates on entangled states, it is not possible to encode the resulting probability distribution merely in terms of bias updates for the qubit on which the gate acts. Updated weight matrix γ' derived to encode probabilities after (general) gate:

$$\gamma'_{a_2 a_1} = \log\left(\mathcal{O}_G^{a_2 a_1 b_2 b_1} \frac{\exp\{\beta_{b_2}^2 + \gamma_{b_2 b_1} + \beta_{b_1}^1\}}{\sum_{ij \in \{1,2,3,4\}} \exp\{\beta_j^2 + \gamma_{ji} + \beta_i^1\}}\right) - \beta_{a_2}^2 - \beta_{a_1}^1$$

this choice guarantees:

$$\frac{\exp\{\beta_{a_2}^2 + \gamma'_{a_2 a_1} + \beta_{a_1}^1\}}{\sum_{ij \in \{1,2,3,4\}} \exp\left\{\beta_{a_j}^2 + \gamma'_{a_j a_i} + \beta_{a_i}^1\right\}} = \mathcal{O}_G^{a_2 a_1 b_2 b_1} P_{b_2 b_1} = P'(a_2, a_1)$$

We see that the specific choice for β^1, β^2 do not influence the encoded probability distribution, since by construction of $\gamma'_{a_2 a_1}$ they do not appear in the

total expression for the probability distribution (added and subtracted in the exponent). Hence one could – for simplicity – set $\beta^2, \beta^1 = 0$. However, this would mean

$$P(a_1) = \frac{\exp\{\beta^1_{a_1}\}}{\sum_i \exp\{\beta^1_{a_i}\}} = 0.25 \quad \forall a_1 \in \{1,2,3,4\}$$

which clearly is not correct in general. Indeed $P(a_1)$ depends on the specific POVM probability distribution of the state under investigation:

$$P(a_1) = \sum_{a_2} P(a_1, a_2).$$

Suppose we have a two-qubit gate. For the first qubit, we can formulate the following operator:

$$\hat{O}^{a_1 b_1} = \sum_{a_2, b_2} \hat{O}^{a_1 a_2 b_1 b_2} \tag{11}$$

which acts according to

$$P'(a_1) = \hat{O}^{a_1 b_1} P(b_1) \tag{12}$$

where summation over indices appearing twice is implicitly assumed. This allows us to simply update the biases β^1 according to the single-qubit rule:

$$\beta'^1_{a_1} = \log\left(\hat{O}^{a_1 b_1} P(b_1)\right) = \log\left(\hat{O}^{a_1 b_1} \frac{\exp\{\beta^1_{b_1}\}}{\sum_i \exp\{\beta^1_i\}}\right)$$

While the effect of the update $\beta' 1$ will be compensated by the weight matrix, it is still useful because $P(a_1)$ is now encoded correctly, independent from a_2 (that is, we do not require $\gamma_{a_2 a_1}$ to obtain an expression for the $P(a_1)$ encoded by our network.

Hence two qubit gates acting on qubit 2 would lead to $\{\beta^1, \beta^2, \gamma\} \Rightarrow \{\beta'^1, \beta^2, \gamma'\}$. While technically the new distribution could be uniquely specified by the choice of γ', an update of β^1 is also included such that $P(a_1)$ is correctly encoded. The biases β^2 can be absorbed in the weight matrix γ:

$$\gamma_{a_2 a_1} = \begin{pmatrix} \gamma_{11} + \beta^2_1 & \gamma_{12} + \beta^2_1 & \gamma_{13} + \beta^2_1 & \gamma_{14} + \beta^2_1 \\ \gamma_{21} + \beta^2_2 & \gamma_{22} + \beta^2_2 & \dots & \\ \vdots & & \ddots & \vdots \\ \gamma_{41} + \beta^2_4 & \dots & & \gamma_{44} + \beta^2_4 \end{pmatrix}$$

This does not – unlike for the case of the biases β^1 – lead to false probabilities (β^2 can never appear independently from γ). For non-entangled qubits, we can simply include the bias in the weight matrix:

$$\gamma_{a_2 a_1} = \begin{pmatrix} \beta^2_1 & \beta^2_1 & \beta^2_1 & \beta^2_1 \\ \beta^2_2 & \beta^2_2 & \dots & \\ \vdots & & \ddots & \vdots \\ \beta^2_4 & \dots & & \beta^2_4 \end{pmatrix}$$

Hence our 2-qubit network can just as well be described in terms of $\{\beta^1, \gamma\}$ and does not require explicit biases for qubit 2, as they can be easily absorbed into the weight matrix.

4 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

$$P'(a_3|a_2 a_1) = \mathcal{O}^{a_3}_{b_3} P(b_3|a_2 a_1) \tag{13}$$

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 13 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

$$\beta'^3 + \gamma'^{(23)} + \gamma'^{(13)} = \ln\left(\sum_{b_3} \mathcal{O}^{a_3}_{b_3} \frac{\exp\left(\beta'^3_{b_3} + \gamma'^{(23)}_{b_3 a_2} + \gamma'^{(13)}_{b_3 a_1}\right)}{\sum_{i,j,k} \exp\left(\beta'^3_i \gamma'^{(23)}_{ij} + \gamma'^{(13)}_{i,}\right)}\right) \tag{14}$$

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

$$\phi = f(a_3) + g(a_3, a_2) + h(a_3, a_1). \tag{15}$$

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

$$\phi' = f'(a_1, a_2, a_3) \tag{16}$$

[1] not overparametrized

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 $\phi = \phi'$.

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.