LSTM-Designed Quantum Experiments

Thomas Adler
Institute for Machine Learning &
LIT AI Lab
Johannes Kepler University Linz
Linz, Austria
adler@ml.jku.at

Manuel Erhard
Institute for Quantum Optics and Quantum Information
Austrian Academy of Sciences &
Vienna Center for Quantum Science & Technology
University of Vienna
Vienna, Austria
manuel.erhard@univie.ac.at

Mario Krenn
Dept. of Chemistry & Dept. of Computer Science
University of Toronto &
Vector Institute for Artificial Intelligence
Toronto, Canada
mario.krenn@univie.ac.at

Johannes Brandstetter
Institute for Machine Learning &
LIT AI Lab
Johannes Kepler University Linz
Linz, Austria
brandstetter@ml.jku.at

Johannes Kofler
Institute for Machine Learning &
LIT AI Lab
Johannes Kepler University Linz
Linz, Austria
kofler@ml.jku.at

Sepp Hochreiter
Institute for Machine Learning &
LIT AI Lab
Johannes Kepler University Linz
Linz, Austria
hochreit@ml.jku.at

Abstract

We demonstrate how machine learning helps to design experiments in quantum physics. Quantum entanglement is a cornerstone for upcoming quantum technologies such as quantum computation and quantum cryptography. Of particular interest are complex quantum states with more than two particles and a large number of entangled quantum levels. Given such a multiparticle high-dimensional quantum state, it is usually impossible to reconstruct an experimental setup that produces it. To search for interesting experiments, one thus has to randomly create millions of setups on a computer and calculate the respective output states. In this work, we show that machine learning models can provide significant improvement over random search. We demonstrate that a long short-term memory (LSTM) neural network can successfully learn to model quantum experiments by correctly predicting output state characteristics for given setups without the necessity of computing the states themselves. This approach not only allows for faster search but is also an essential step towards automated design of multiparticle high-dimensional quantum experiments using generative machine learning models.

1 Introduction

In the past decade, artificial neural networks have been applied to a plethora of scientific disciplines, commercial applications, and every-day tasks with outstanding performance in, e.g., medical diagnosis, self-driving, and board games [1, 2]. In contrast to standard feedforward neural networks, long short-term memory (LSTM) [3, 4] architectures have recurrent connections, which allow them to process sequential data such as text and speech [5].

Such sequence-processing capabilities can be particularly useful for designing complex quantum experiments, since the final state of quantum particles depends on the sequence of elements, i.e. the experimental setup, these particles pass through. For instance, in quantum optical experiments, photons may traverse a sequence of wave plates, beam splitters, and holographic plates. High-dimensional quantum states are important for multiparticle and multisetting violations of local realist models as well as for applications in emerging quantum technologies such as quantum communication and error correction in quantum computers [6, 7].

Already for three photons and only a few quantum levels, it becomes in general infeasible for humans to determine the required setup for a desired final quantum state, which makes automated design procedures for this inverse problem necessary. One example of such an automated procedure is the algorithm MELVIN [8], which uses a toolbox of optical elements, randomly generates sequences of these elements, calculates the resulting quantum state, and then checks whether the state is interesting, i.e. maximally entangled and involving many quantum levels. The setups proposed by MELVIN have been realized in laboratory experiments [9, 10]. Recently, also a reinforcement learning approach has been applied to design new experiments [11].

Inspired by these advances, we investigate how LSTM networks can learn quantum optical setups and predict the characteristics of the resulting quantum states. We train the neural networks using millions of setups generated by MELVIN. The huge amount of data makes deep learning approaches the first choice. We use cluster cross validation [12] to evaluate the models.

2 Methods

2.1 Target values

Let us consider a quantum optical experiment using three photons with orbital angular momentum (OAM) [13]. The OAM of a photon is characterized by an integer whose size and sign represent the shape and handedness of the photon wavefront, respectively. For instance, after a series of optical elements, a three particle quantum state may have the following form:

$$|\Psi\rangle = \tfrac{1}{2}\left(|0,0,0\rangle + |1,0,1\rangle + |2,1,0\rangle + |3,1,1\rangle\right). \tag{1}$$

This state represents a physical situation, in which there is 1/4 chance (modulus square of the amplitude value 1/2) that all three photons have OAM value 0 (first term), and a 1/4 chance that photons 1 and 3 have OAM value 1, while photon 2 has OAM value 0 (second term), and so on for the two remaining terms.

We are generally interested in two main characteristics of the quantum states: (1) Are they maximally entangled? (2) Are they high-dimensional? The dimensionality of a state is represented by its Schmidt rank vector (SRV) [14]. State $|\Psi\rangle$ is indeed maximally entangled because all terms on the right hand side have the same amplitude value. Its SRV is (4,2,2), as the first photon is four-dimensionally entangled with the other two photons, whereas photons two and three are both only two-dimensionally entangled with the rest.

A setup is labeled "positive" ($y_\mathrm{E} = 1$) if its output state is maximally entangled and if the setup obeys some further restrictions, e.g., behaves well under multi-pair emission, and otherwise labeled "negative" ($y_\mathrm{E} = 0$). The target label capturing the state dimensionality is the SRV $y_\mathrm{SRV} = (n, m, k)^\top$. We train LSTM networks to directly predict these state characteristics (entanglement and SRV) from a given experimental setup without actually predicting the quantum state itself.

For classification, we use binary cross entropy (BCE) in combination with logistic sigmoid output activation for learning. For regression, it is always possible to reorder the photon labels such that the SRV has entries in non-increasing order. An SRV label is thus represented by 3-tuple $y_\mathrm{SRV} = (n, m, k)^\top$ which satisfies $n \geq m \geq k$. We model $n \sim \mathcal{P}(\lambda)$ as a Poisson-distributed random variable and $m \sim \mathcal{B}(n,p), k \sim \mathcal{B}(m,q)$ as Binomials with ranges $m \in \{1, \ldots n\}$ and $k \in \{1, \ldots, m\}$ and success probabilities p and q, respectively. The resulting log-likelihood objective (omitting all terms not depending on λ, p, q) for a data point x with label $(n, m, k)^\top$ is

$$\ell(\hat\lambda, \hat p, \hat q \mid x) = n \log \hat\lambda - \hat\lambda + m \log \hat p + (n - m) \log(1 - \hat p) + k \log \hat q + (m - k) \log(1 - \hat q) \tag{2}$$

where $\hat\lambda, \hat p, \hat q$ are the network predictions (i.e. functions of x) for the distribution parameters of n, m, k respectively. The Schmidt rank value predictions are $\hat n = \hat\lambda$, $\hat m = \hat p \hat\lambda$, $\hat k = \hat p \hat q \hat\lambda$.
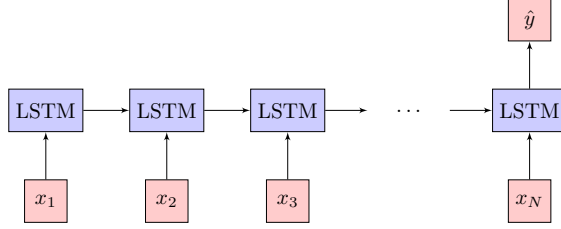
Figure 1: Sequence processing model for a many-to-one mapping. The target value $\hat{y}$ can be either an estimate for y_{E} (entanglement classification) or y_{SRV} (SRV regression).

0,1 / 0,1	2	3	4	5	6	7	8	9-12

Table 1: Cluster cross validation folds (0-8) and extrapolation set (9-12) by leading Schmidt rank n.

2.2 Network architecture

The used sequence processing model is depicted in Figure 1. We train two networks, one for entanglement classification (target y_{E}), and one for SRV regression (target y_{SRV}). The reason why we avoid multitask learning in this context is that we do not want to incorporate correlations between entanglement and SRV into our models. For instance, the SRV (6,6,6) was only observed in non-maximally entangled samples so far, which is a perfect correlation. This would cause a multitask network to automatically label such a sample as negative only because of its SRV. By training separate networks we lower the risk of incorporating such correlations.

A setup of N elements is being fed into a network by its sequence of individual optical components $x = (x_1, x_2, ..., x_N)^\top$, where in our data N ranges from 6 to 15. We use an LSTM with 2048 hidden units and a component embedding space with 64 dimensions. The component embedding technique is similar to word embeddings [15].

3 Experiments

3.1 Data

The dataset produced by MELVIN consists of 7,853,853 different setups of which 1,638,233 samples are labeled positive. Each setup consists of a sequence x of optical elements, and the two target values y_{E} and y_{SRV}. We are interested in whether the trained model is able to extrapolate to unseen SRVs. Therefore, we cluster the data by leading Schmidt rank n. All samples with $n \geq 9$ are moved to a special *extrapolation set* consisting of only 1,754 setups (gray cell in Table 1). The remainder of the data, i.e. all samples with $n < 9$, is then divided into a training set and a conventional *test set* with 20 % of the data drawn at random (iid). The test set is used to estimate the conventional generalization error, while the extrapolation set is used to shed light on the ability of the learned model to perform on higher Schmidt rank numbers. If the model extrapolates successfully, we can hope to find experimental setups that lead to new interesting quantum states.

Cluster cross validation (CCV) is an evaluation method similar to standard cross validation. Instead of grouping the folds iid, CCV groups them according to a clustering method. Thus, CCV removes similarities between training and validation set and simulates situations in which the withheld folds have not been obtained yet, thereby allowing us to investigate the ability of the network to discover these withheld setups. We use CCV with nine folds (white cells in Table 1). Seven of these folds correspond to the leading Schmidt ranks $2, \ldots, 8$. The samples with $n = 1$ (not entangled) and $n = 0$ (not even a valid three-photon state) are negative by definition. These samples represent special cases of $y_{\mathrm{E}} = 0$ setups and it is not necessary to generalize to these cases without training on them. Therefore, the 4,300,268 samples with $n < 2$ are divided into two folds at random such that the model will always see some of these special samples while training.
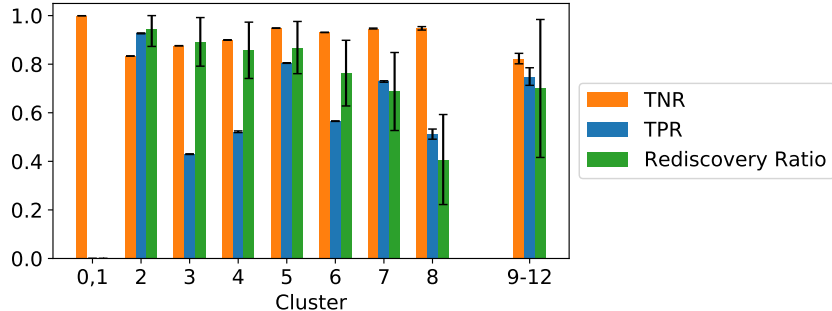
Figure 2: True negative rate (TNR), true positive rate (TPR), rediscovery ratio of the LSTM network using cluster cross validation for different folds 0-8. True negative rates are high for all validation folds. All metrics are good for the extrapolation set 9-12, demonstrating that the models perform well on data beyond the training set distribution, covering only Schmidt rank numbers 0-8. Error bars represent 95 % binomial proportion confidence intervals.

3.2 Results

Let us examine if the LSTM network has learned something about quantum physics. A good model will identify positive setups correctly while discarding as many negative setups as possible. This behavior is reflected in the metrics *true positive rate* $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$ and *true negative rate* $\mathrm{TNR} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP})$, with TP, TN, FP, FN the true positives, true negatives, false positives, false negatives, respectively. A metric that quantifies the success rate within the positive predictions is the *precision* (or positive predictive value), defined as $\mathrm{PPV} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$.

For each withheld CCV fold n, we characterize a setup to be "interesting" when it fulfills the following two criteria: (i) It is classified positive ($\hat{y}_{\mathrm{E}} > \tau$) with τ the classification threshold of the sigmoid output activation. (ii) The SRV prediction $\hat{y}_{\mathrm{SRV}} = (\hat{n}, \hat{m}, \hat{k})^{\top}$ is such that there exists a $y_{\mathrm{SRV}} = (n, m, k)^{\top}$ with $\|y_{\mathrm{SRV}} - \hat{y}_{\mathrm{SRV}}\|_2 < r$. We call r the SRV radius. We denote samples which are classified as interesting (uninteresting) and indeed positive (negative) as true positives (negatives).

We employ stochastic gradient descent for training the LSTM network with momentum 0.5 and batch size 128. We sample mini-batches in such a way that positive and negative samples appear equally often in training. For balanced SRV regression, the leading Schmidt rank vector number n is misused as class label. The models were trained using early stopping after 40000 weight update steps for the entanglement classification network and 14000 update steps for the SRV regression network. Hyperparameter search was performed in advance on a data set similar to the training set.

Figure 2 shows the TNR, TPR, and rediscovery ratio for sigmoid threshold $\tau = 0.5$ and SRV radius $r = 3$. The rediscovery ratio is defined as the number of distinct SRVs, for which at least 20% of the samples are rediscovered by our method, i.e. identified as interesting, divided by the number of distinct SRVs in the respective cluster. The TNR for fold 0,1 is 0.9996, and the precision on the extrapolation set 9-12 is 0.659. Error bars in Figure 2 and later in the text are 95 % binomial proportion confidence intervals. Model performance depends heavily on parameters τ and r. Figure 3 shows the "beyond distribution" results for a variety of sigmoid thresholds and SRV radii.

Finally, we investigate the conventional in-distribution generalization error using the test set (20 % of the data). Entanglement classification: The entanglement training BCE loss value is 10.2. TNR and TPR are 0.9271 ± 0.00024 and 0.9469 ± 0.00041, respectively. The corresponding test error is 10.4. TNR and TPR are 0.9261 ± 0.00038 and 0.9427 ± 0.00065, respectively. SRV regression: The SRV training loss value according to Equation (2) is 2.247, the accuracy with $r = 3$ is 93.82 % and the mean distance between label and prediction is 1.3943. The SRV test error is 2.24, the accuracy with $r = 3$ is 0.938 and the mean distance between label and prediction is 1.40. These figures are consistent with a clean training procedure.
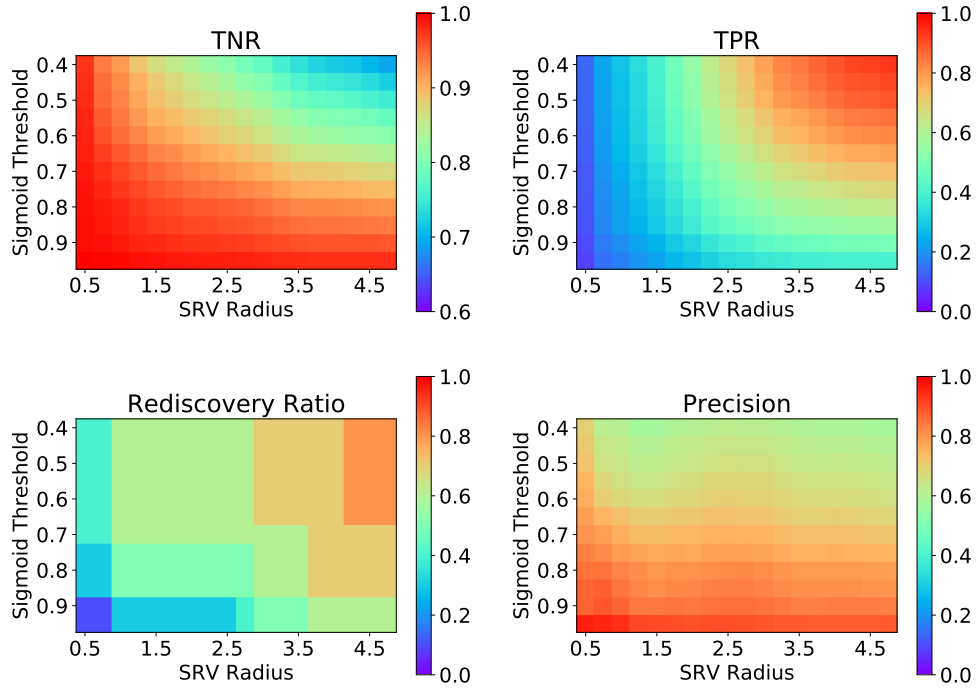
Figure 3: True negative rate (scale starts at 0.6), true positive rate, rediscovery ratio, and precision for the extrapolation set 9-12 for varying sigmoid threshold τ and SRV radius r. For too restrictive parameter choices ($\tau \to 1$ and $r \to 0.5$) the TNR and precision approach the value 1, while TPR and rediscovery ratio approach 0, such that no interesting new setups would be identified. For too loose choices (small τ, large r), too few negative samples would be rejected, such that the advantage over random search becomes negligible, reflected in smaller precision values. Hence, there is a trade-off between rediscovery ratio (diversity of discoveries) and precision (speed of discoveries). For a large variety of τ and r the models perform satisfyingly well, allowing a decent compromise between TNR and TPR. This is also reflected by a value of 0.64 for the mean average precision, where the mean is taken over $r = 0.5$ to $r = 7$ with a step size of 0.1 and the average precision is over $\tau = 1$ to $\tau = 0$ with a step size of 0.01 for each value of r.

4 Conclusion

We have shown that an LSTM-based neural network can be trained to successfully predict certain characteristics of high-dimensional multiparticle quantum states from the experimental setup without any explicit knowledge of quantum mechanics. The network performs well even on unseen data beyond the training distribution, proving its extrapolation capabilities. This paves the way to automated design of complex quantum experiments using generative machine learning models.

References

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(115), 2017.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(354), 2017.

[3] S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma Thesis, TU München*, 1991.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(1735), 1997.

[5] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[6] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52(R2493), 2000.

[7] D. Kaszlikowski, P. Gnacínski, M. Zukowski, W. Miklaszewski, and A. Zeilinger. Violations of local realism by two entangled N-dimensional systems are stronger than for two qubits. *Phys. Rev. Lett.*, 86(4418), 2000.

[8] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger. Automated Search for new Quantum Experiments. *Phys. Rev. Lett.*, 116(090405), 2016.

[9] M. Malik, M. Erhard, M. Huber, M. Krenn, R. Fickler, and A. Zeilinger. Multi-photon entanglement in high dimensions. *Nature Photonics*, 10(248), 2016.

[10] M. Erhard, M. Malik, M. Krenn, and A. Zeilinger. Experimental GHZ entanglement beyond qubits. *Nature Photonics*, 12(759), 2018.

[11] A. A. Melnikova, H. Poulsen Nautrupa, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel. Active learning machine learns to create new quantum experiments. *PNAS*, 115(1221), 2017.

[12] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter. DeepTox: Toxicity Prediction using Deep Learning. *Frontiers in Environmental Science*, 3(80), 2016.

[13] A. M. Yao and M. J. Padgett. Orbital angular momentum: origins, behavior and applications. *Adv. Opt. Photon.*, 3(161), 2011.

[14] M. Huber and J. I. de Vicente. Structure of multidimensional entanglement in multipartite systems. *Physical Review Letters*, 110(030501), 2013.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR Workshop, arXiv:1301.3781*, 2013.