UPRIGHT PUBLICATIONS
https://upright.pub/index.php/ijrstp/

Original Contribution

Modeling Long Short-Term Memory in Quantum Optical Experiments

Ruhul Amin[1], Mani Manavalan[2]

Keywords: LSTM, Optical experiments, Quantum entanglement

International Journal of Reciprocal Symmetry and Theoretical Physics

Vol. 4, Issue 1, 2017 [Pages 6-13]

During the previous decade, artificial neural networks have excelled in a wide range of scientific disciplines, commercial applications, and everyday professions, including medical diagnostics, self-driving automobiles, and board games, to mention a few. In contrast to classic feed forward neural networks, long short-term memory (LSTM) designs use recurrent connections to process sequential data such as text and speech. We explain how machine learning can be used to describe quantum physics experiments. Quantum entanglement is a key component of quantum technologies such as quantum computation and quantum cryptography. The study of complex quantum states with more than two particles and a large number of entangled quantum levels is particularly interesting. Reconstructing an experimental setup that yields such a multi-particle high-dimensional quantum state is usually impossible. To come up with interesting experiments, one must randomly generate millions of setups on a computer and calculate the resulting states. In this study, we show that machine learning models beat random search by a significant margin. We show that without having to compute the states directly, an LSTM neural network can successfully train to simulate quantum experiments by correctly predicting output state characteristics for given settings. This strategy not only speeds up the search, but it's also a prerequisite for building multi-particle high-dimensional quantum experiments using generative machine learning models.

INTRODUCTION

Artificial neural networks have excelled in a variety of scientific disciplines, commercial applications, and everyday jobs during the last decade, including medical diagnosis, self-driving cars, and board games, to name a few (Esteva et al., 2017; Silver et al., 2017). Long short-term memory (LSTM) (Hochreiter, 1991; Hochreiter and Schmidhuber, 1997; Abedin et al., 2012; Azad et al., 2011; Begum et al., 2012; Neogy and Ahmed, 2015; Siddique & Ahmed, 2015; Bynagari, 2014; Donepudi, 2014a; Manavalan, 2014; Manavalan & Ganapathy, 2014) architectures, in contrast to traditional feedforward neural networks, feature recurrent connections, allowing them to process sequential data like text and speech (Sutskever et al., 2014; Achar, 2016).

One of the difficulties in using RNNs in our situation is that we need to be able to optimize tens of thousands of parameters. It is not possible to optimize at this scale with a fully linked RNN since it would require a large hidden state and a large number of parameters. To get around this problem, we'll utilize an optimizer called m, which works coordinate wise on the parameters of the objective function, similar to RMSprop and ADAM. This coordinate wise network architecture enables us to define the optimizer with a very small network that

[1]Senior Data Entry Control Operator (IT), ED-Maintenance Office, Bangladesh Bank (Head Office), Dhaka, BANGLADESH
[2]Technical Project Manager, Larsen & Toubro Infotech (LTI), Mumbai, INDIA

only looks at a single coordinate and share optimizer parameters across different optimize parameters.

Separate activations for each goal function parameter are used to achieve different behavior on each coordinate. This arrangement has the wonderful effect of making the optimizer invariant to the order of parameters in the network, because the same update rule is applied independently on each coordinate, in addition to allowing us to utilize a small network for this optimizer.

A two-layer Long Short Term Memory (LSTM) network is used (Hochreiter and Schmidhuber, 1997) with the now-standard forget gate design to implement the update rule for each coordinate. The network receives the optimize gradient for a single coordinate as well as the previous hidden state as inputs and returns the update for the corresponding optimize parameter as output. This architecture, seen in Figure 2, will be referred to as an LSTM optimizer.

The final state of quantum particles is determined by the sequence of elements, i.e. the experimental setup, through which they transit. Such sequence-processing capabilities can be particularly valuable for building complex quantum experiments. Photons may pass via a series of wave plates, beam splitters, and holographic plates in quantum optical studies, for instance. High-dimensional quantum states are critical for multiple particle and multiple setting violations of local realist models, as well as applications in upcoming quantum technologies like quantum communication and quantum error correction (Achar, 2015; Shor, 2000; Kaszlikowski et al., 2000; Manavalan & Bynagari, 2015; Manavalan & Donepudi, 2016; Donepudi, 2015; Donepudi, 2016).

It is generally impossible for humans to calculate the requisite setup for a desired end quantum state for three photons and only a few quantum levels, necessitating the use of automated design techniques to solve this inverse problem. The algorithm MELVIN (Krenn et al., 2016) is an example of such an automated procedure that takes a toolbox of optical elements, generates random 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. , i.e. fully entangled and involves a large number of quantum levels. In laboratory studies (Malik et al., 2016; Erhard et al., 2018b), the MELVIN setups have been accomplished. A reinforcement learning approach has recently been used to develop new experiments as well (Melnikov et al., 2018).

Objectives of the Study

This study is aimed at investigating how LSTM networks can learn quantum optical setups and predict the characteristics of the resulting quantum states, a task whose level of difficulty is significantly higher than that of other deep learning tasks such as object recognition or text generation, as a result of these advances. We use millions of setups created by MELVIN to train the neural networks. Deep learning algorithms are the top choice because to the vast volume of data. In order to evaluate the models, we apply cluster cross validation (Mayr et al., 2016).

LITERATURE REVIEW

Target values

Consider three photons with orbital angular momentum (OAM) (Yao and Padgett, 2011; Erhard et al., 2018a). The shape and handedness of the photon wave front are represented by the size and sign of the OAM of a photon. A three-particle quantum state, for example, might take the following form after a series of optical elements:

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

This condition depicts a physical situation in which there is a 1/4 probability (modulus square of the amplitude value 1/2) that all three photons have OAM value 0, 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 outstanding terms.

The two key properties of quantum states that we're interested in are:

(1) Do they have the greatest amount of intertwining?
(2) Do they have a lot of depth to them?

The Schmidt rank vector (SRV) depicts a state's dimensionality (Huber and de Vicente, 2013; Huber et al., 2013). All terms on the right hand side have the same amplitude value, indicating that state $\mathrm{I}\Psi\rangle$ is maximally entangled. It has an SRV of (4, 2, 2) because the first photon is four-dimensionally entangled with the other two, but photons two and

three are only two-dimensionally entangled with the rest.

If a setup's output state is maximally entangled and it obeys any additional constraints, such as behaving well under multi-pair emission, it is termed "positive" (yE = 1); otherwise, it is labeled "negative" (yE = 0). The SRV ySRV = (n, m, k)^{T} is the target label for capturing state dimensionality. Without actually forecasting the quantum state, we train LSTM networks to predict these state features (entanglement and SRV) from a particular experimental setup.

Loss Function

We employ binary cross entropy (BCE) in conjunction with logistic sigmoid output activation to learn classification (Ahmed & Dey, 2009a). It is always feasible to reorganize the photon labels for regression so that the SRV entries are in non-increasing order. The 3-tuple ySRV = (n, m, k)^{T} represents an SRV label, which satisfies n ≥m≥ k. We represent $n \sim P(\lambda)$ as a Poisson-distributed random variable and m B(n, p), k ∼B(m, q) as Binomials with ranges m ϵ {1,..., n} and k ϵ {1,..., m} with success probability p and q, respectively, with a small misuse of nomenclature. The resulting log-likelihood goal for a data point x with label (omitting all terms not dependent on, λ, p, q⟩ for a data point x with label (n, m, k)^{T} is

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

where $\hat{\lambda}, \hat{p}, \hat{q}$ are the network estimations functions that is functions of x for the distribution parameters of n, m, k respectively. The Schmidt rank value predictions are $\hat{n} = \hat{\lambda}\,\hat{m} = \hat{p}\,\hat{\lambda}\,\hat{k} = \hat{p}\,\hat{q}\,\hat{\lambda}$. To visualize this we need

$$f(n, m, k) = \frac{\lambda^{n} e^{-k}}{n!}\binom{n}{m} p^{m}(1-p)^{n-m}\binom{m}{n} q^{k}(1-p)^{m-k}$$

To obtain the marginal distribution of m, the first set of sum over all possible k, which is easy. To sum out n we first observe that $\binom{n}{m} = 0$ for n < m, that is the first m terms are zero and we may write

$$f(m) = \sum_{n=0}^{\infty} f(n, m) = \sum_{n=0}^{\infty} f(m+n, m)$$

Capturing only non-zero terms, it follows that

$$f(m) = \sum_{n=0}^{\infty} \frac{\lambda^{n+m} e^{-k}}{(n+m)!}\binom{n+m}{m} p^{m}(1-p)^{n}$$

$$e^{-\lambda} p^{m} \lambda^{m} \sum_{n=0}^{\infty} \frac{\lambda^{n}(1-p)^{n}}{(n+m)!}\binom{n+m}{m}$$

$$\frac{e^{-\lambda} p^{m} \lambda^{m}}{m!} \sum_{n=0}^{\infty} \frac{\lambda^{n}(1-p)^{n}}{n!} = \frac{e^{-p\lambda}(p\lambda)^{m}}{m!}$$

Which is P(pλ)-distributing the same argument for k we obtain that the marginal of k is $P(p\lambda)^{\lambda}$-distributed. The predictions $\hat{n}, \hat{m}, \hat{k}$ are obtained by taking the means of their respective marginal.

Network Architecture

Figure 1 depicts the sequence processing model that was used. Two networks are trained: one for entanglement classification (target yE) and the other for SRV regression (target ySRV). The reason we avoid multitask learning in this instance is that we don't want to include entanglement and SRV correlations in our models (Ahmed & Dey, 2009). The SRV (6, 6, 6), for example, has only been found in nonmaximally entangled samples so far, indicating a perfect connection. A multitask network would immediately categorize such a sample as negative based just on its SRV. We reduce the possibility of absorbing such correlations by training different networks.

The sequence of distinct optical components x = (x_1, x_2,..., x_N)^{T} feeds a setup of N elements into a network, where N in our data spans from 6 to 15. We utilize a 2048-hidden-unit LSTM and a 64-dimensional component embedding space. The technique of component embedding is similar to that of word embedding (Mikolov et al., 2013; Ahmed and Neogy, 2009).

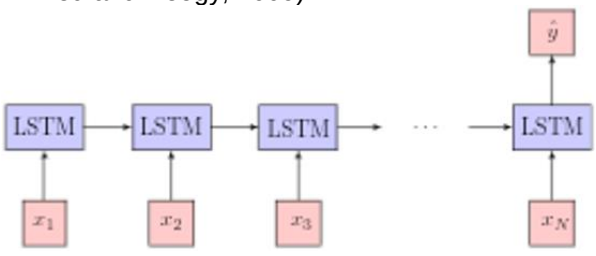

Figure 1. Structure handling model for a many-to-one mapping. The aim value y can be either predicted for $_{yE}$ (entanglement categorization or $_{ySRV}$ (SRV regression).

Experimental Methods

MELVIN generated a dataset with 7,853,853 different setups, 1,638,233 of which are positive samples. A sequence x of optical elements, as well as the two target values yE and ySRV, are included in each configuration. We're curious if the trained model can extrapolate to SRVs that haven't been seen before. As a result, we use leading Schmidt rank n to cluster the data. The number of positive and negative samples in the data set for each n is depicted in Figure 2.

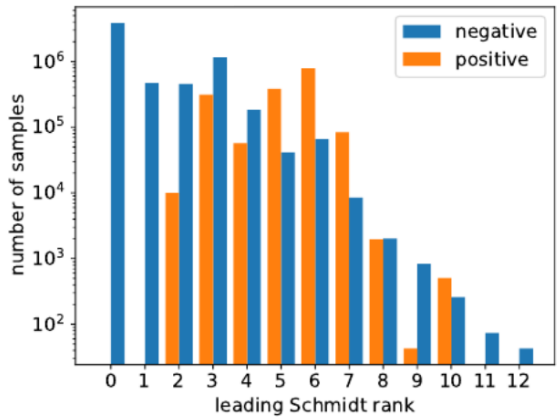

Figure 2: The data set's negative and positive samples as a function of the leading Schmidt rank n.

Workflow

All samples with n 9 are shifted to a special extrapolation set (gray cell in Table 1) with only 1,754 configurations. The remaining data, which includes all samples with n 9, is divided into a training set and a traditional test set, with 20% of the data picked at random (iid). Figure 3 depicts this process. The test set is used to calculate the traditional generalization error, while the extrapolation set is utilized to assess the learnt model's ability to perform on higher Schmidt rank values. We can hope to find experimental setups that lead to new intriguing quantum states if the model extrapolates well (Ahmed and Day, 2009).
CCV is a type of cross validation that is similar to normal cross validation. Instead of grouping the folds by iid, CCV uses a clustering algorithm to group them. As a result, CCV eliminates commonalities between the training and validation sets and simulates scenarios in which the withheld folds have yet to be attained, allowing us to test the network's ability to uncover these withheld settings. With nine folds, we use CCV (white cells in Table 1). Table 1 shows the cluster cross validation folds (0-8) and extrapolation set (9-12) as a function of the leading Schmidt rank n. The samples with n = 0 and n = 1 are merged and then randomly divided into two folds (0, 1).

Table 1. Cluster Cross validation

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

The leading Schmidt ranks 2,...,8 correspond to seven of these folds. Negative by definition are the samples with n = 1 (not entangled) and n = 0 (not even a proper three-photon state). These examples are special circumstances of yE = 0 settings, thus there is no need to generalize to them without first training on them. As a result, the 4,300,268 examples with n < 2 are randomly divided into two folds, ensuring that the model sees some of these unusual samples throughout training.

Figure 3 depicts the workflow. We divided the data into groups based on their leading Schmidt rank n. The extrapolation set is made up of all samples with n 9, which we utilize to test our model's out-of-distribution capabilities. We make a random test split at a ratio of 1/4 for the remaining samples (i.e. n 9) The test set is utilized to calculate our model's conventional generalization error. Cluster cross validation is performed using the training set.

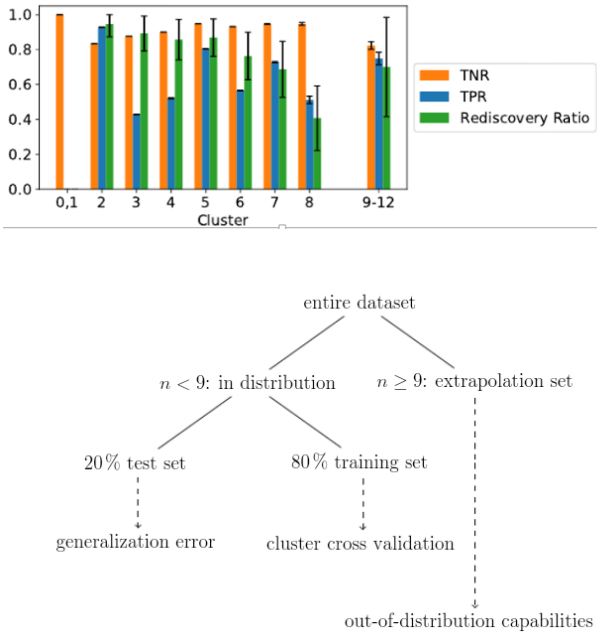

Figure 3. Workflow

RESULTS AND DISCUSSION

Let's see if the LSTM network has picked up any quantum physics knowledge. Positive configurations will be correctly identified by a competent model, while as many negative setups as possible will be discarded. True positive rate TPR = TP/(TP + FN) and true negative rate TNR = TN/(TN + FP) illustrate this tendency, with TP, TN, FP, FN representing true positives, true negatives, false positives, and false negatives, respectively. The precision (or positive predictive value), defined as PPV = TP/(TP + FP), is a statistic that evaluates the success rate within positive predictions.

We define a configuration as "interesting" when it meets the following two conditions for each withholding CCV fold n: I It is classified as positive $(yE > \tau)$ using τ the sigmoid output activation classification threshold. (ii) The SRV prediction $\hat{y}SRV = (\hat{n}\hat{m}\hat{k})^T$ is such that there exists a $\|\hat{y}SRV\ \hat{y}SRV\|_2 < \tau$ with $ySRV = (n, m, k)^T$. We call τ the SRV radius is denoted by the letter r. True positives are samples that are evaluated as intriguing (uninteresting) and indeed positive (negative) (negatives). False positives are samples that are labelled as intriguing (uninteresting) and indeed negative (positive) (false negatives).

With a momentum of 0.5 and a batch size of 128, we use stochastic gradient descent to train the LSTM network. We sample mini-batches in such a way that both positive and negative samples emerge in training at the same rate. The leading Schmidt rank vector number n is utilized as a class label in balanced SRV regression. Early halting after 40000 weight update steps for the entanglement classification network and 14000 update steps for the SRV regression network was used to train the models. On a data set similar to the training set, a hyper-parameter search was done in advance.

TNR, TPR, and rediscovery ratio for sigmoid threshold τ = 0.5 and SRV radius r = 3 are shown in Figure 4. The rediscovery ratio is calculated by dividing the number of distinct SRVs for which our algorithm identifies at least 20% of the samples as interesting by the number of distinct SRVs in the particular cluster. The TNR for fold 0,1 is 0.9996, and the extrapolation set 9-12 has a precision of 0.659. In Figure 4 and elsewhere in the text, the error bars represent 95 percent binomial proportion confidence intervals. The performance of the model is highly influenced by parameters and r. For a variety of sigmoid thresholds and SRV radii, Figure 5 displays the “beyond distribution” findings.

Finally, we use the test set (20 percent of the data) to evaluate the traditional in-distribution generalization mistake. Classification of entanglements: The BCE loss value for entanglement training is 10.2. TNR and TPR are respectively 0.9271 ± 0.00024 and 0.9469 ± 0.00041. The test error for this is 10.4. TNR and TPR are respectively 0.9261 ± 0.00038 and 0.9427 ± 0.00065. SRV regression:

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

According to above equation, the SRV training loss value is 2.247, the accuracy is 93.82 percent with r = 3, and the mean distance between label and prediction is 1.3943. The mean distance between label and prediction is 1.40, the SRV test error is 2.24, the accuracy with r = 3 is 0.938, and the SRV test error is 2.24. These figures are in line with a proper training program.

Figure 4 depicts the LSTM network's true negative rate (TNR), true positive rate (TPR), and rediscovery ratio for different folds 0-8, cluster cross validation was used. For all validations, true negative rates are high folds. For the extrapolation set 9-12, all metrics are good, indicating that the models work based on data outside of the training set distribution, only spanning Schmidt rank numbers 0-8 Error bars are a type of error message provide 95% confidence intervals for binomial proportions.

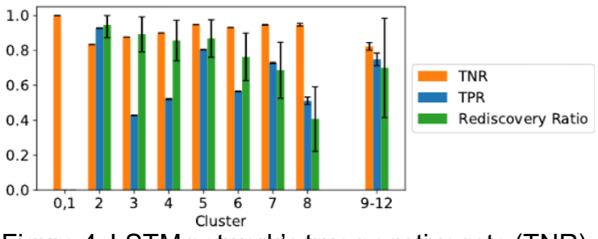

Figure 4. LSTM network’s true negative rate (TNR), true positive rate (TPR) and rediscovery ratio.

Figure 5 shows extrapolation set 9-12, true negative rate (scale starts at 0.6), true positive rate, rediscovery ratio, and precision for varying sigmoid threshold and SRV radius r**.** true negative rate (TNR), and accuracy approach 1 for excessively restrictive parameter selections ($\tau \to 1$ and $\tau \to 0.5$), while true positive rate (TPR) and rediscovery

ratio approach 0, resulting in no interesting new setups being found. Too few negative samples would be discarded for too loose selections (small τ, large r), therefore the gain over random search would be minimal, as indicated in lower precision values. As a result, there is a trade-off between precision (speed of discoveries) and rediscovery ratio (diversity of findings). The models function admirably for a wide range of τ and r, offering a reasonable compromise between TNR and TPR. This is represented in a mean average precision of 0.64, where the mean is calculated over r = 0.5 to r = 7 with a step size of 0.1 and the average precision is calculated over r = 1 to r = 0 with a step size of 0.01 for each value of r.

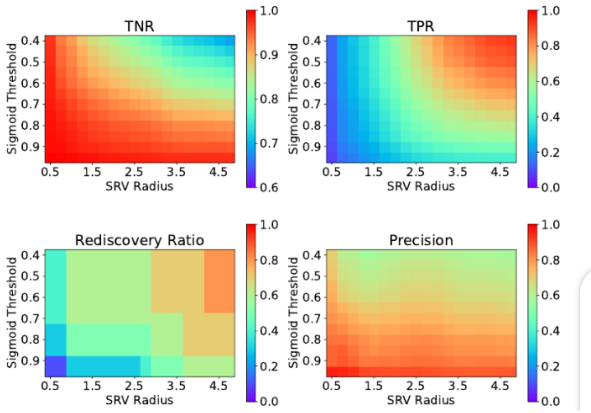

Figure 5. Sigmoid threshold extrapolation of SRV for true negative rate (TNR), true positive rate (TPR), rediscovery ratio, and precision

DISCUSSION

We show that an LSTM-based neural network can be taught to model some features of complicated quantum systems in our experiments. Our method isn't restricted to entanglement and Schmidt rank; it can also be used to additional objective functions like multiparticle transformations, interference, and fidelity quality, among others (Ahmed, 2009).

The use of generative models is another possible next step in expanding our approach toward the objective of automated design of multiparticle high-dimensional quantum experiments. As possible methodologies, we explore Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and beam search (Lowerre, 1976).

1D CNNs and LSTMs have been used to generate sequences such as text in adversarial scenarios (Gulrajani et al., 2017; Yu et al., 2016; Fedus et al., 2018; Ahmed, 2016; Ahmed et al., 2011). The LSTM-based techniques use reinforcement learning ideas to solve the problem of gradient propagation across the network's softmax outputs. These methodologies are directly applicable to our setting because our data has a structure similar to text.

There are two approaches to beam search: a discriminative method and a generative one (Bynagari, 2015; Bynagari, 2016; Donepudi, 2014b; Manavalan, 2016). The discriminative technique takes into account all of the data (positive and negative samples). The models developed for this project can be employed in a discriminative method, in which new sequences are created by maximizing the network's belief that the outcome would be a favorable setup. The objective behind the generative technique is to train a model solely on positive samples in order to learn their distribution through next element prediction.

Beam search can be used for inference to approximate the most likely sequence given some initial conditions (Bengio et al., 2015). Another way to produce new sequences is to take a sample from the softmax distribution of the network output at each sequence point, like in text generation models (Graves, 2013; Karpathy and Fei-Fei, 2015).

Automated experiment design processes, in general, have far larger applications than quantum optical setups, and can be useful in a variety of scientific disciplines other than physics.

CONCLUSION

Without any explicit knowledge of quantum mechanics, an LSTM-based neural network can be trained to successfully predict certain characteristics of high-dimensional multiparticle quantum states from the experimental setup. Quantum optical experiments are far more difficult to analyze for humans than other deep learning problems, such as image classification. Even on unseen data outside of the training distribution, the network performs well, demonstrating its extrapolation ability. This paves the path for the use of generative machine learning models to automate the design of complex quantum experiments.

REFERENCES

Abedin, M. M. M., Ahmed, A. A. A., and Neogy, T. K. (2012). Mechanism of Accountability and Auditing: Public Sector Scenarios of Bangladesh. Journal of Business Studies, 4, 131-148.

Achar, S. (2015). Requirement of Cloud Analytics and Distributed Cloud Computing: An Initial Overview.

International Journal of Reciprocal Symmetry and Physical Sciences, 2, 12–18. https://upright.pub/index.php/ijrsps/article/view/70

Achar, S. (2016). Software as a Service (SaaS) as Cloud Computing: Security and Risk vs. Technological Complexity. Engineering International, 4(2), 79-88. https://doi.org/10.18034/ei.v4i2.633

Ahmed, A. A. A. & Dey, M. M. (2009). Timeliness attributes and the extent of accounting disclosure: a study of banking companies in Bangladesh. *Osmania Journal of International Business Studies, 4*(1).

Ahmed, A. A. A. (2009). The Effect of Timeliness Regulation of Corporate Financial Reporting: Evidence from Banking Sector of Bangladesh. *Accounting and Management Information Systems, 8*(2), 216 - 235. http://online-cig.ase.ro/jcig/art/8_2_4.pdf

Ahmed, A. A. A. (2016). Relationship between Foreign Direct Investment and Company Taxation: Case of Bangladesh. *American Journal of Trade and Policy, 3*(1), 11-14. https://doi.org/10.18034/ajtp.v3i1.394

Ahmed, A. A. A. and Day, M. M. (2009). Bank loan officers' perceptions of corporate financial disclosure in the banking sector of Bangladesh: An empirical analysis, Proceedings 2nd CBRC, Lahore, Pakistan, 1-12.

Ahmed, A. A. A. and Neogy, T. K. (2009). Merger & Acquisitions (M&A) Goodwill Accounting: Principles and Practice. The Bangladesh Accountant, 65, 75-91.

Ahmed, A. A. A., & Dey, M. M. (2009a). Corporate Attribute and the Extent of Disclosure: A Study of Banking Companies in Bangladesh. Proceedings of the 5th International Management Accounting Conference (IMAC), OCT 19-21, 2009, UKM, Kuala Lumpur, MALAYSIA, Pages: 531-553. https://publons.com/publon/11427801/

Ahmed, A. A. A., Khan, W., & Hossain, M. S. (2011). Reporting Practice of Accounting Disclosure on Changes in Listed Companies of Bangladesh. ASA University Review, 5(1), 83-96. https://www.researchgate.net/publication/336664901

Azad, M. R., Khan, W., & Ahmed, A. A. A. (2011). HR Practices in Banking Sector on Perceived Employee Performance: A Case of Bangladesh. Eastern University Journal, 3(3), 30–39. https://doi.org/10.5281/zenodo.4043334

Begum, R., Ahmed, A. A. A., & Neogy. T. K. (2012). Management Decisions and Univariate Analysis: Effects on Corporate Governance in Bangladesh. Journal of Business Studies, 3, 87-115.

Bengio, S., O. Vinyals, N. Jaitly, and N. Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In Advances in Neural Information Processing Systems 28, pp. 1171–1179. Curran Associates, Inc., 2015.

Bynagari, N. B. (2014). Integrated Reasoning Engine for Code Clone Detection. *ABC Journal of Advanced Research*, *3*(2), 143-152. https://doi.org/10.18034/abcjar.v3i2.575

Bynagari, N. B. (2015). Machine Learning and Artificial Intelligence in Online Fake Transaction Alerting. *Engineering International*, *3*(2), 115-126. https://doi.org/10.18034/ei.v3i2.566

Bynagari, N. B. (2016). Industrial Application of Internet of Things. *Asia Pacific Journal of Energy and Environment*, *3*(2), 75-82. https://doi.org/10.18034/apjee.v3i2.576

Donepudi, P. K. (2014a). Technology Growth in Shipping Industry: An Overview. *American Journal of Trade and Policy*, *1*(3), 137-142. https://doi.org/10.18034/ajtp.v1i3.503

Donepudi, P. K. (2014b). Voice Search Technology: An Overview. *Engineering International*, *2*(2), 91-102. https://doi.org/10.18034/ei.v2i2.502

Donepudi, P. K. (2015). Crossing Point of Artificial Intelligence in Cybersecurity. American Journal of Trade and Policy, 2(3), 121-128. https://doi.org/10.18034/ajtp.v2i3.493

Donepudi, P. K. (2016). Influence of Cloud Computing in Business: Are They Robust?. Asian Journal of Applied Science and Engineering, 5(3), 193-196. Retrieved from https://journals.abc.us.org/index.php/ajase/article/view/1181

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

Erhard, M., R. Fickler, M. Krenn, and A. Zeilinger. 2018a. Twisted photons: new quantum perspectives in high dimensions. Light: Science & Applications, 7(3):17146.

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

Fedus, W., I. Goodfellow, and A. M. Dai. 2018. MaskGAN: Better text generation via filling in the ….. In International Conference on Learning Representations, 2018.

Goodfellow, I. J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc..

Graves, A. 2013. Generating sequences with recurrent neural networks. arXiv:1308.0850.

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. 2017. Improved training of wasserstein gans. In Advances in Neural

Information Processing Systems 30, pp. 5767–5777. Curran Associates, Inc., 2017.

Hochreiter, S and J. Schmidhuber. 1997. Long short-term memory. Neural Computation, (1735), 1997.

Hochreiter, S. 1991. Untersuchungen zu dynamischen neuronalen Netzen. Diploma Thesis, TU München, 1991.

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

Huber, M., M. Perarnau-Llobet, and J. I. de Vicente. 2013. Entropy vector formalism and the structure of multidimensional entanglement in multipartite systems. Physical Review A, 88(4):042328.

Karpathy A and L. Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3128–3137.

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

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

Lowerre, B. T. 1976. The Harpy speech recognition system. PhD Thesis, Carnegie Mellon University, Pittsburgh, 1976.

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

Manavalan, M. (2014). Fast Model-based Protein Homology Discovery without Alignment. *Asia Pacific Journal of Energy and Environment*, *1*(2), 169-184. https://doi.org/10.18034/apjee.v1i2.580

Manavalan, M. (2016). Biclustering of Omics Data using Rectified Factor Networks. *International Journal of Reciprocal Symmetry and Physical Sciences*, *3*, 1–10. Retrieved from https://upright.pub/index.php/ijrsps/article/view/40

Manavalan, M., & Bynagari, N. B. (2015). A Single Long Short-Term Memory Network can Predict Rainfall-Runoff at Multiple Timescales. *International Journal of Reciprocal Symmetry and Physical Sciences*, *2*, 1–7. Retrieved from https://upright.pub/index.php/ijrsps/article/view/39

Manavalan, M., & Donepudi, P. K. (2016). A Sample-based Criterion for Unsupervised Learning of Complex Models beyond Maximum Likelihood and Density Estimation. *ABC Journal of Advanced Research*, *5*(2), 123-130. https://doi.org/10.18034/abcjar.v5i2.581

Manavalan, M., & Ganapathy, A. (2014). Reinforcement Learning in Robotics. *Engineering International*, *2*(2), 113-124. https://doi.org/10.18034/ei.v2i2.572

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

Melnikov, A. A., H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel. 2018. Active learning machine learns to create new quantum experiments. PNAS, 115(1221), 2018.

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

Neogy, T. K. and Ahmed, A. A. A. (2015). The Extent of Disclosure of Different Components of Disclosure Index: A Study on Commercial Banks in Bangladesh. *Global Disclosure of Economics and Business, 4*(2), 100-110. https://doi.org/10.18034/gdeb.v4i2.139

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

Siddique, M. N. & Ahmed, A. A. A. (2015). Congruence of Competitive Advantage and Transfer Pricing: A Study on Selected MNCs Operating in Bangladesh. *Asian Accounting & Auditing Advancement, 5*(2), 119-126. https://www.researchgate.net/publication/354712086

Silver, D., 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. 2017. Mastering the game of Go without human knowledge. Nature, 550(354), 2017.

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

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

Yu, L., W. Zhang, J. Wang, and Y. Yu. 2016. Seqgan: Sequence generative adversarial nets with policy gradient. arxiv:1609.05473, 2016.

--0--