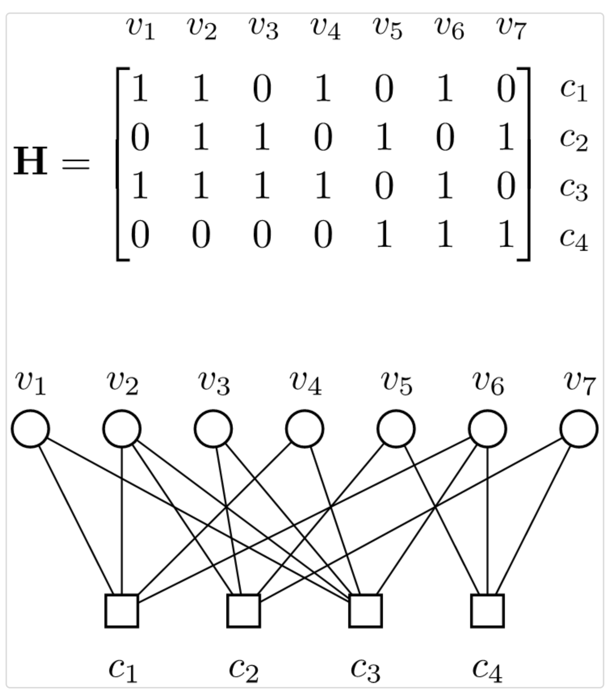
Deep Neural Network Probabilistic Decoder for Stabilizer Codes

Stefan Krastanov, Liang Jiang

Departments of Applied Physics and Physics, Yale Quantum Institute, Yale University, New Haven, Connecticut 06520, USA

Introduction



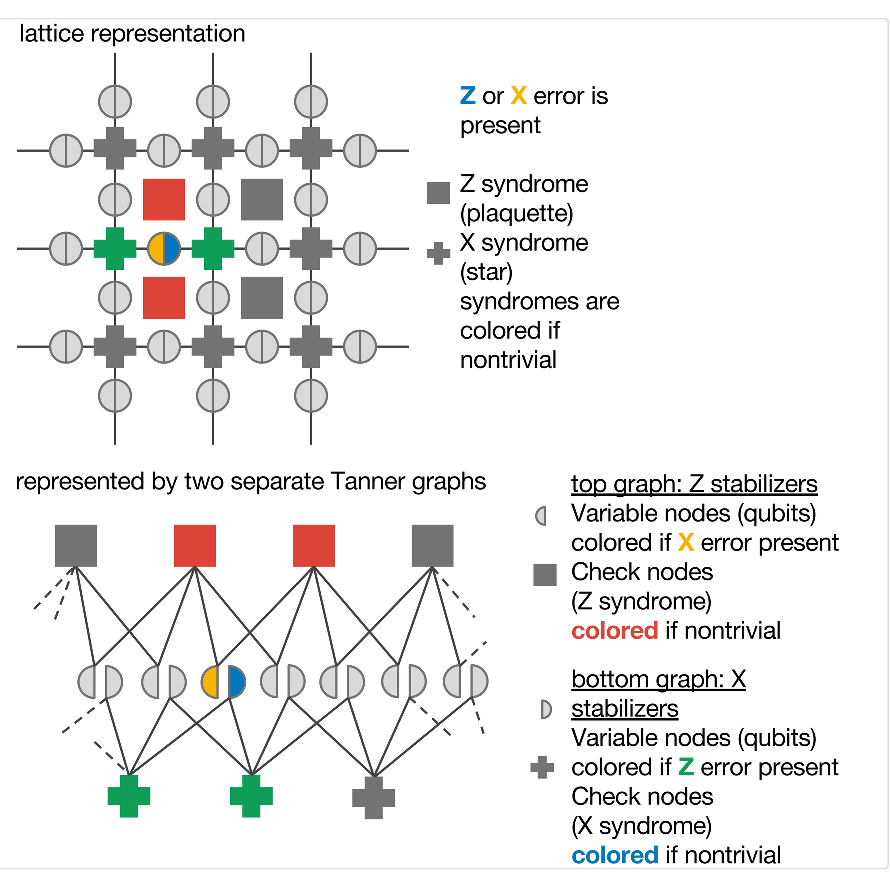
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

To redundantly encode information one can use a system of linear constrains (i.e. constraining the parity) on the physical qubits, as done in stabilizer codes (or their classical counterpart - linear codes). These constraints are represented as a matrix equation, or equivalently as the corresponding graph connecting each "check" c_i (constraint) to its "variables" v_i (qubits whose parity is constrained).

A substantial challenge presents itself when one attempts to decode a code after an error has occurred. In general, this is an infeasible NP-complete problem, but in many cases the particular code would have some additional structure that permits the creation of a smart efficient decoding algorithm.

We present a method of creating a decoder for any stabilizer code, by training a neural network to invert the syndrome-to-error map for a given error model[1]. We evaluate the performance of our decoder when applied to the Toric Code.

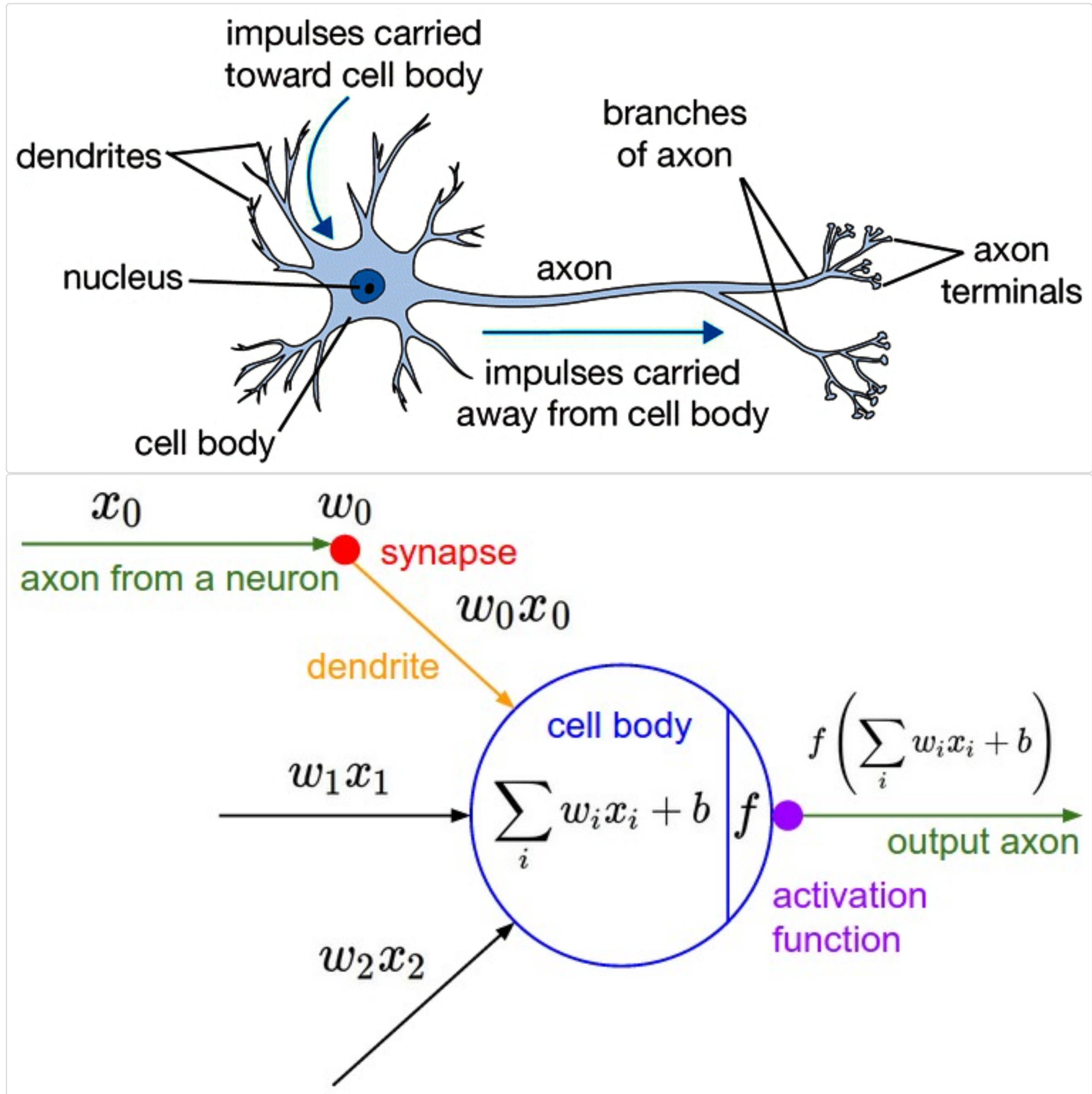
Example: The Toric Code and its MWPM Decoder



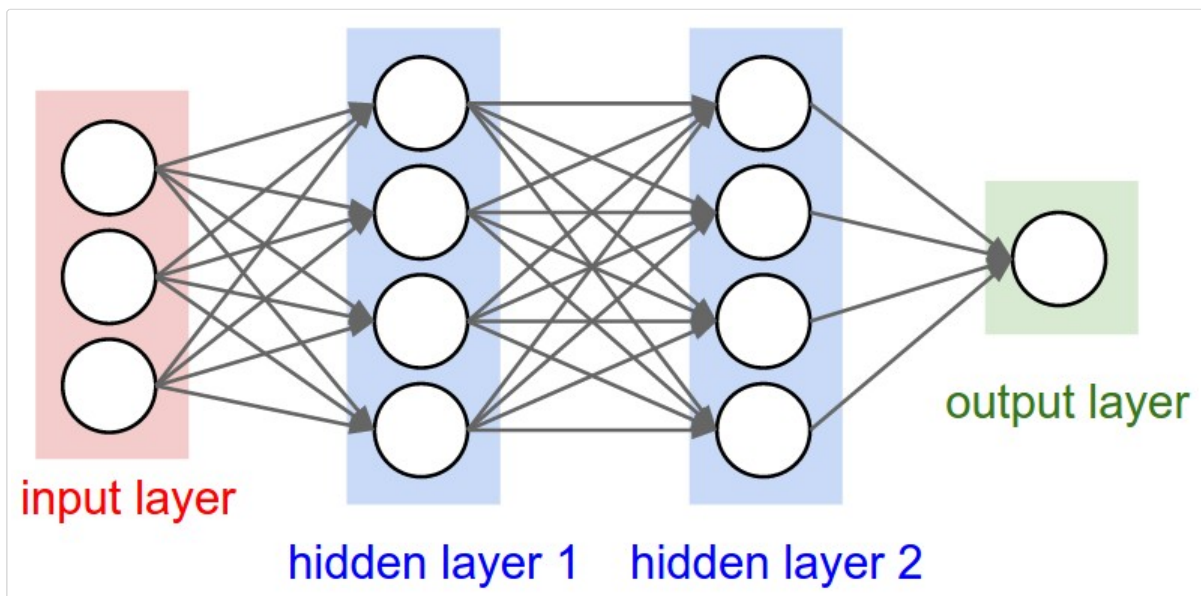
We will use the Toric code as a benchmark for our otherwise general protocol. In it the physical qubits are laid out in a lattice, and the parity checks act on cells of the lattice. After interpreting non-trivial syndromes as pairs of particles, decoding becomes a problem of matching and annihilating those pairs, for instance by minimizing the total path they have to travel as done by Minimal Weight Perfect Matching (MWPM).

Neural Networks 101

Neural Networks are exceedingly good at fitting high-dimensional nonlinear functions to given training data The function to be "learned" is expressed as a network of "neurons" parametrized by the strengths of the connections between neurons.



A real neuron and a model of a neuron used in artificial neural networks. The incomming signals are summed, passed through a nonlinearity and the output "activation" is sent to the next layer. The bias b and numerous connection weights w_i are the optimization parameters during fitting/training.

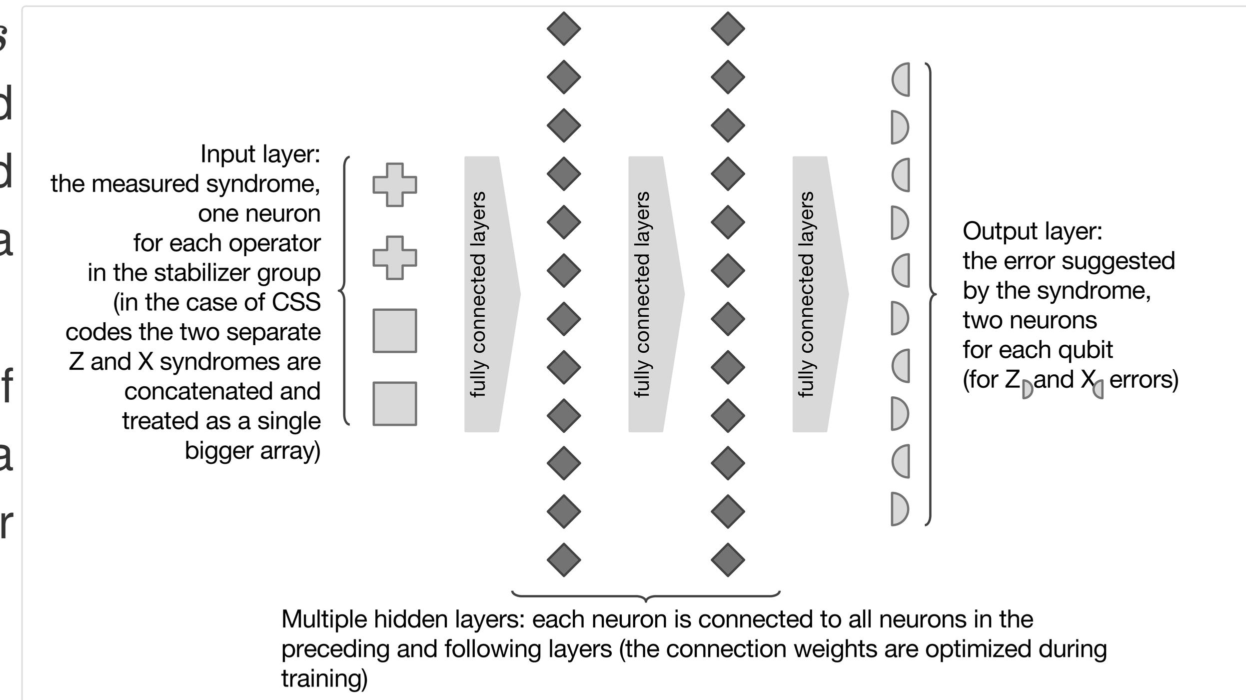


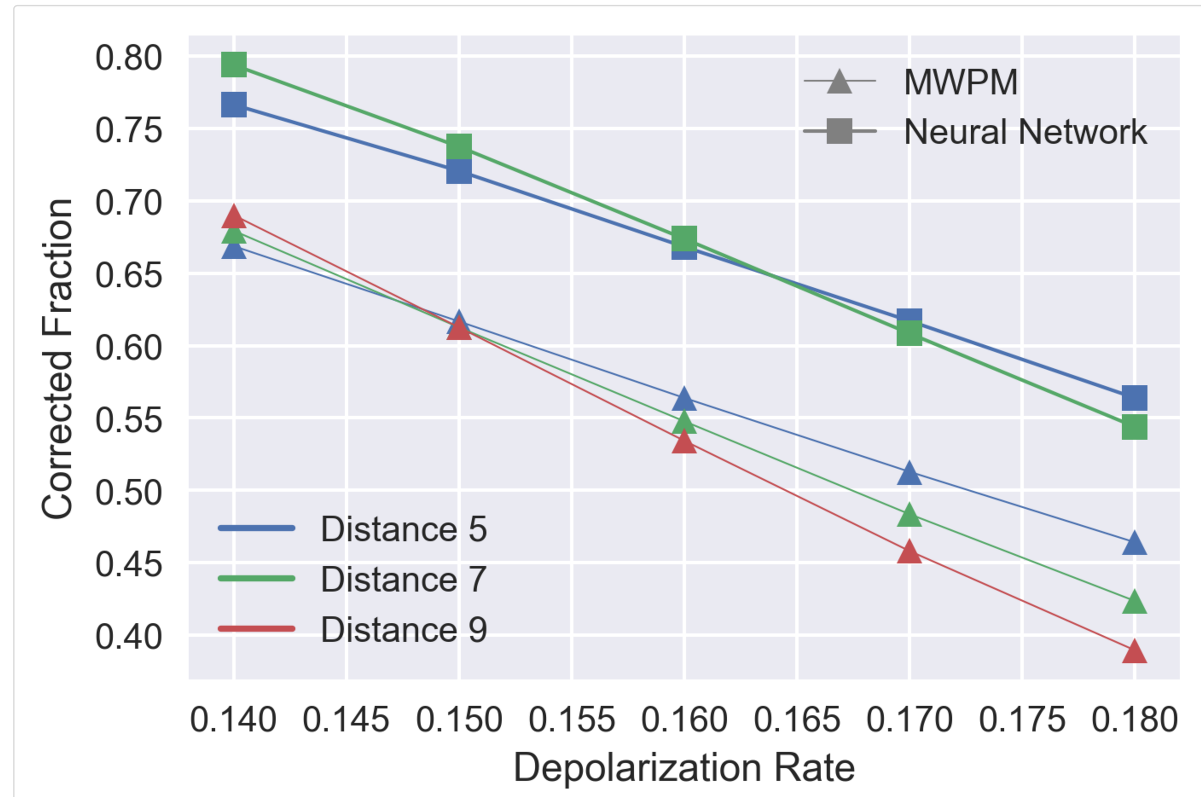
Neural Networks are created from multiple layers of connected neurons. The first layer is the input. The "activation values" are propagated through the network and the last layer is the output. The strength of the connections between neurons is what is trained/optimized. Image credit: [2].

The Neural Network Quantum Error Correcting Code Decoder

Decoding is deducing from a syndrome measurement s most probable error e that caused it. For a given code and error model, we generate a large sample of errors and compute the corresponding syndromes. We use that data to train a neural network to do the mapping from s to e.

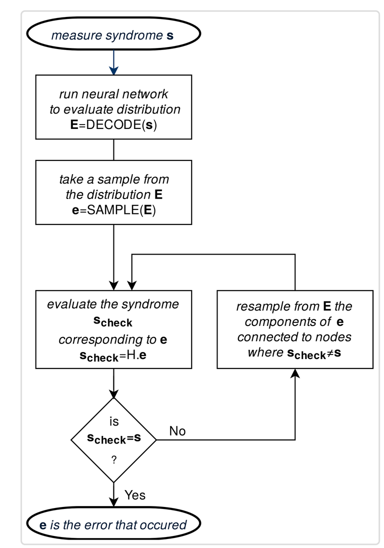
An important caveat is the need to interpret the output of the neural network as a probability distribution - it is not a discrete yes/no answer, rather a probability that an error occurred given the syndrome.



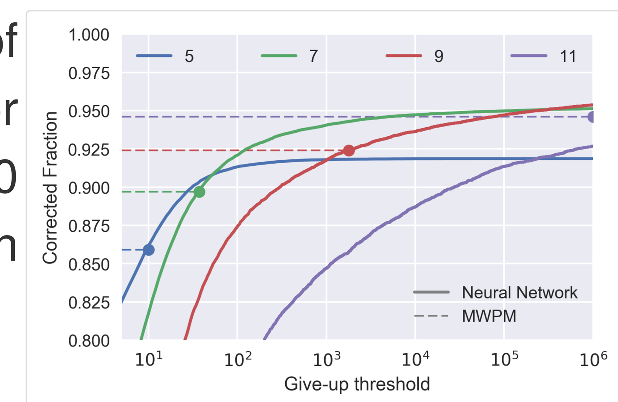


Our decoder significantly outperforms MWPM without having hard-coded knowledge of its lattice structure. The threshold is much better, and the properly corrected fraction of codewords is higher.

Note: Efficient Sampling from the Neural Network



For a given syndrome s, the network's output E is evaluated and interpreted as a list of error probabilities from which an array e (whether an error occurred) is sampled. If the guess e does not produce the syndrome s we resample, but **only the qubits taking part in the stabilizer measurement corresponding to the incorrect elements of the syndrome**.



After a set number of iterations, we give up. For codes of more than 200 qubits, this protocol can become impractical.

Benchmark: Comparing to Other Decoders

Our neural decoder's threshold (16.4%) compares favorably to renormalization group decoders[3] (15.2%) and outperforms MWPM. Only a renormalization sparse code decoder[3] reaches a similar threshold. These decoders have been hand-crafted for the Toric Code, while our design can be applied to other codes. There are a number of other attempts to employ neural networks as decoders, notably[4], but they do not outperform MWPM.

Conclusions and Outlook

Our architecture provides a practical high-fidelity decoder for Toric codes of less than 200 qubits, outperforming most alternatives. Exciting developments in the near future would be the use of more advanced sampling algorithms employing recurrent neural nets, and on the other hand, applying this architecture to promising LDPC codes that do not yet have any known decoders.

References

Our neural net architecture [1]: *Krastanov, Jiang, "Deep Neural Network Probabilistic Decoder for Stabilizer Codes", Scientific Reports, 2017, (arXiv:1705.09334)*

[2]: *Karpathy, "CS231n: Convolutional Neural Networks for Visual Recognition", 2015*

[3]: *Duclos-Cianci, Poulin, "Fast decoders for topological quantum codes", PRL, 2010*

[4]: *Torlai, Melko, "Neural Decoder for Topological Codes", PRL, 2017*

Yale Quantum Institute