Utility programs for nitrogen-vacancy center identification and initialization analysis

Aaron Vontell, Michael Walsh, Eric Wadkins, Dirk Englund

Quantum Photonics Group, Massachusetts Institute of Technology, Cambridge MA, 02139, USA

vontell@mit.edu

Abstract: Here we present an automated tool utilizing TensorFlow to train and classify spectra scans to determine if the scan is characteristic of a nitrogen-vacancy center (NV). We also present a process using ARTIQ to determine characteristics of our NV initialization, determined by photon counts.

OCIS codes: 200.3050, 270.5585

1. Introduction

In the fall of 2016, we worked on setting up the infrastructure and an abstraction for signal-generating experiments, utilizing an FPGA interface called ARTIQ [1]. After testing this infrastructure and realizing both its limitations and advantages, we decided to begin looking into pulse sequences and experiments that we could run with the software. We have decided to work towards performing a Rabi oscillation experiment, which we started by performing an analysis on the initialization of NV centers, for use in conditional pulse sequences.

Additionally, as we further automated processes in the lab, there was a need for reliable classification of spectra scans to determine if a scan was characteristic of an NV center. This has now been accomplished with a neural network written in TensorFlow.

2. Neural Network for Spectra Scan Classification

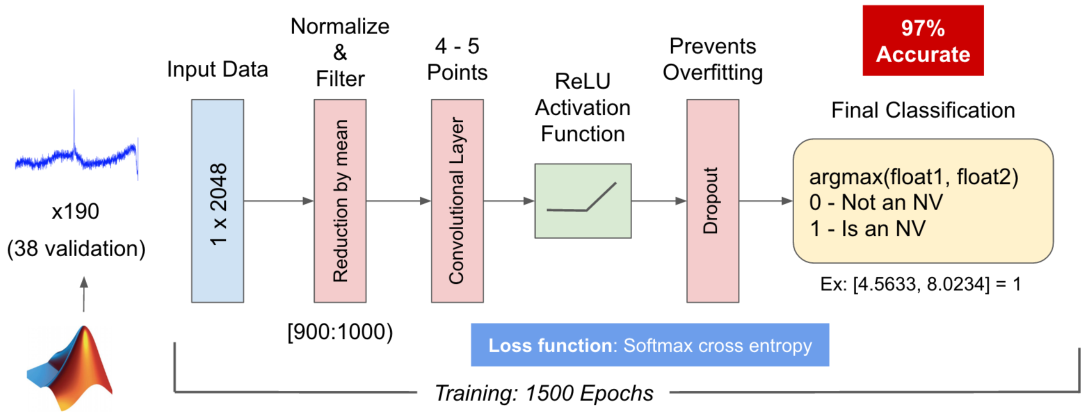

Fig. 1. Overview of the NV Neural Network

In an attempt to expedite the NV identification process in a diamond sample, we have developed a neural net (using TensorFlow) which would classify a spectra scan of a particular point on the diamond sample. As seen in Figure 1, the neural network takes as input the spectrum scan as a one dimensional array, formatted within a MATLAB file. In our particular testing, we had a total of 228 samples, of which 190 were used for training, and 38 were used for validation testing.

After receiving input, the automated software performs normalization and filtering on the sample, which reduces the values in the sample by the average value within the sample, as well as selects the particular region in which a peak appears for these particular scans.

The data then propagates through the network, which involves a convolutional layer, a rectified linear unit activation function, and a dropout (excluding about 10% of samples) which helps prevent over-fitting. Our final sum at the end of the network results in two values, with the max of the two values determining whether or not this particular scan is characteristic of a nitrogen-vacancy center.

For training, the network weights are configured over 1500 cycles (epochs), utilizing a softmax cross entropy loss function to minimize loss over the duration of training. In our final experiments, we have been able to achieve 97.37% accuracy. The parameters used to achieve this can be found on GitHub [2]. In terms of future development, we would like to train this network on more training data, as well as expand the neural network to be used in a variety of scanning operations.

3. Nitrogen-Vacancy Center Initialization Assessment

In order to perform a successful pulse sequence, such as that used to create a Rabi oscillation, one must ensure that the NV center has been initialized properly. High fidelity initialization is required in order to ensure that the results of a pulse sequence are susceptible to minimal amounts of error.

We are able to utilize ARTIQ and its real-time capabilities to measure the fidelity of an NV center initialization, by measuring photons emitted from the NV center after the initialization sequence. Below is a diagram from a Nature letter [3] which describes this conditional testing, which is adapted nicely with our previously developed ARTIQ abstraction and infrastructure.

With ARTIQ, code has been written to apply the charge repump, listen for photons using the APD, and record the time needed to detect n photons. This code can be used in the lab to assess the number of photons we expect to count after initialization. Once we have this data, we can then use the previously created handler functions to start a pulse sequence (i.e. the Rabi oscillation) once we determine that the correct threshold of photons has been reached.

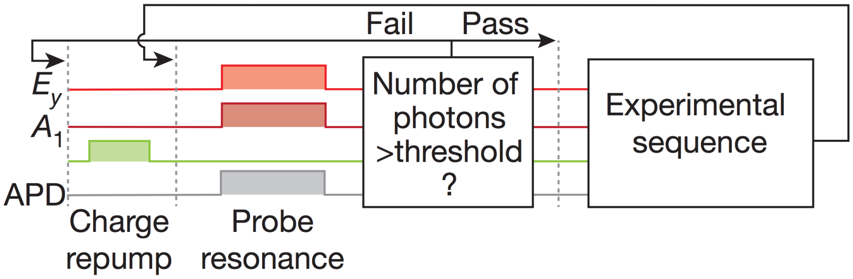

Fig. 2. Pulse sequence conditional on initialization

The main issue with determining the fidelity of our photons is that we cannot discern photons that are background noise from the photons that are emitted from the NV center. Simulations were therefore created which models three random ”events”, drawn randomly from exponential distributions: the detection of background photons ($\lambda = 10^{-2}$), the start delay of the charge repump ($\lambda = 10^{-6}$), and the detection of emitted photons ($\lambda = 10^{-6}$). Since we know whether a photon is a background or emitted photon in this simulation, we can measure the fidelity of each simulated initialization (i.e. the initialization is bad if the threshold is reached before the charge repump is activated). Figure 3 shows the measured fidelity for n 1 through 20, with 1000 trials each.

4. Conclusion and Next Steps

Now that the code has been written for both NV classification and NV initialization, Our next goal is to expand on their capabilities and to test them within the lab. The NV classification will be expanded on by another UROP, who will be applying machine learning and computer vision techniques to further assist in NV identification and management. The NV initialization software will now be tested in the lab with one of the cryogenic setups, bringing us closer to a successful pulse sequence experiment with high fidelity. This will involve coding the Rabi oscillation sequence, along with code to analyze the results.

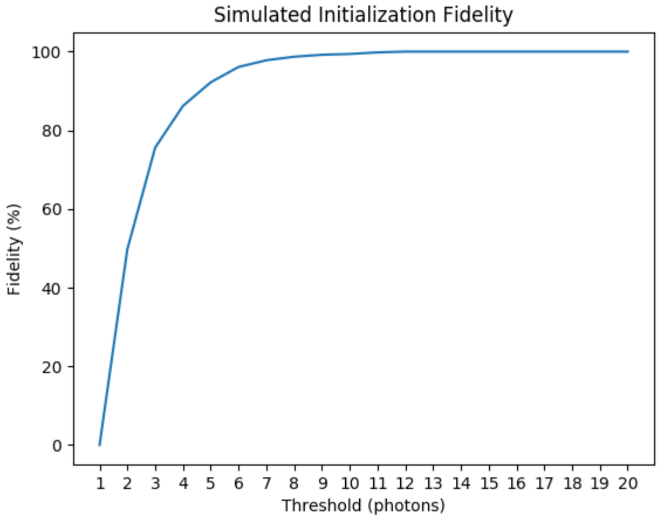

Fig. 3.

5. Other Interesting IAP Endeavors

Over this past month I have also familiarized myself with some interesting quantum computing and quantum information processing tools. In particular I have been using Rigetti's new quantum computing cloud infrastructure (Grove and pyQuil) to build and run quantum algorithms [4]. For example, I wrote a function to generate assembly code which performs the inverse Quantum Fourier Transform on their quantum computing infrastructure (which is now merged in their codebase) [5].

References

1. Bourdeauducq, Sbastien et al. (2016). ARTIQ 1.0. Zenodo. 10.5281/zenodo.51303.
2. ”NV-NeuralNet” GitHub Repository by vontell. https://github.com/vontell/NV-NeuralNet.
3. Bernien, H., Hensen, B., Pfaff, W., Koolstra, G., Blok, M. S., Robledo, L., . . . Hanson, R. (2013). ”Heralded entanglement between solid-state qubits separated by three metres.” Nature, 497(7447), 86-90. doi:10.1038/nature12016.
4. R. Smith, M. J. Curtis and W. J. Zeng, ”A Practical Quantum Instruction Set Architecture,” (2015), arXiv:1608.03355 [quant-ph], https://arxiv.org/abs/1608.03355
5. Vontell, A. ”Inverse QFT - Implementation and Test Pull Request #15 rigetticomputing/grove” (Rigetti Computing, Ed.). Retrieved February 03, 2017, from https://github.com/rigetticomputing/grove/pull/15