Audio classifier development with the use of quantum machine learning

Andrey Yashkin
Science advisor: Evgeniy Pavlovskiy

Big Data AI
Novosibirsk State University

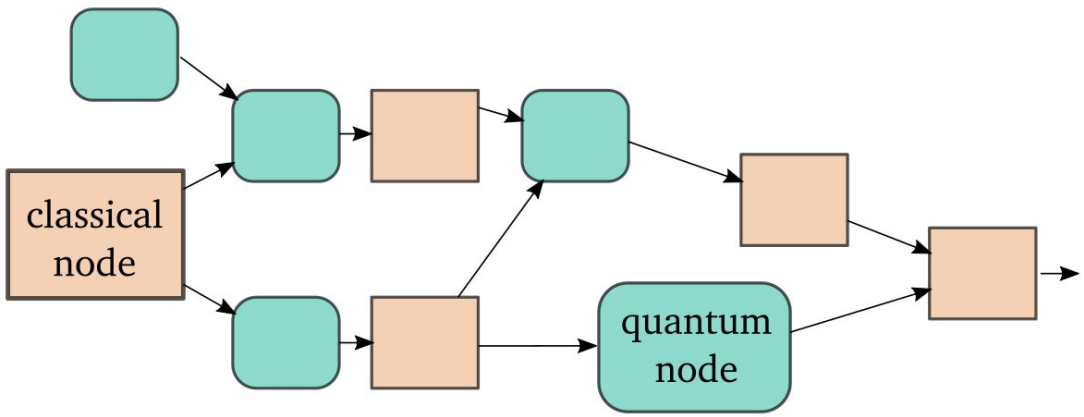
The 15th of December 2020

Quantum machine learning now

- Quantum computers are very hard to simulate
- There exist some very interesting quantum machine learning algorithms with exponential or polynomial speedups of classical alternatives
- Near term quantum computers are too noisy and don't have too much qubits
- It is not obvious if quantum machine learning is beneficial on near term quantum devices

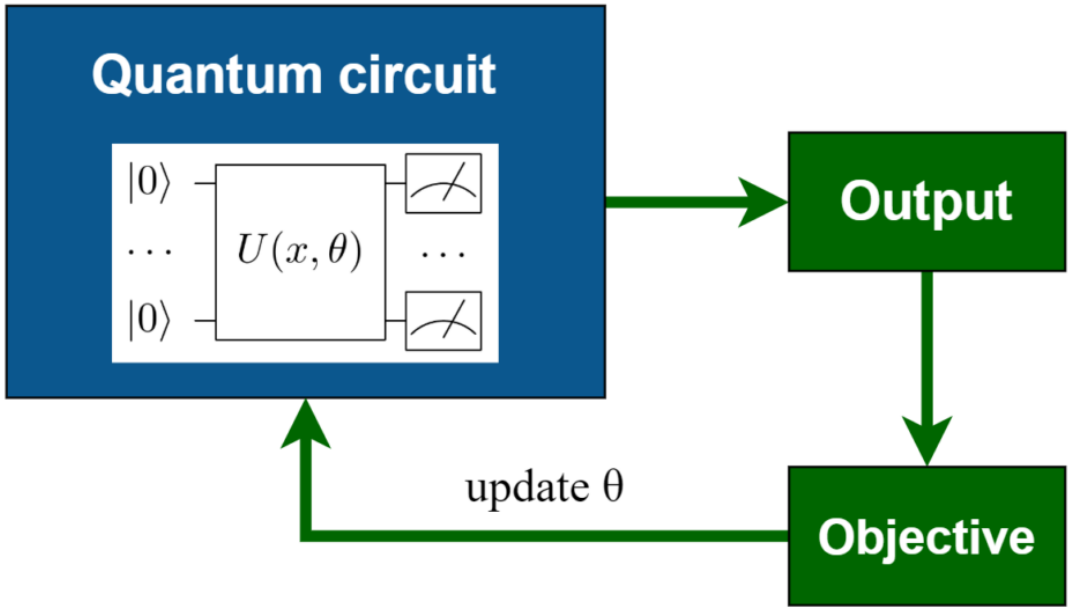
Hybrid quantum-classical computations

In hybrid algorithms a problem is splited on small parts which all can be solved easily by classical and a quantum computer.



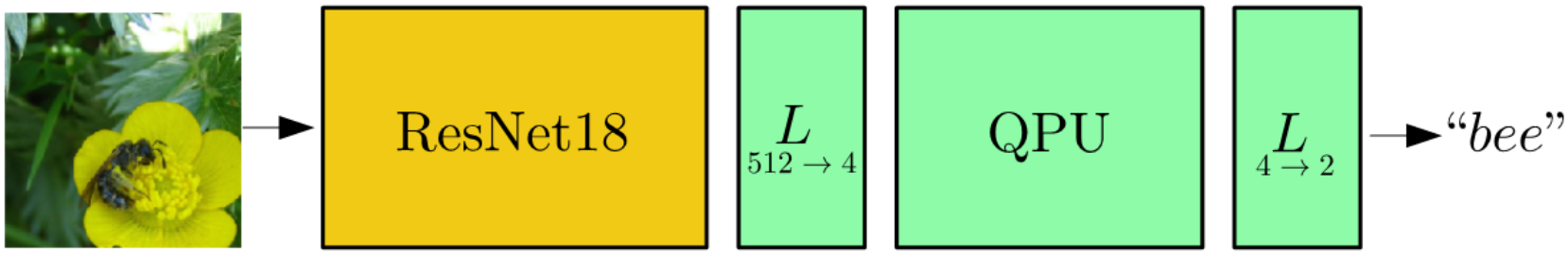
Variational circuits

Variational circuits are parameter-dependent quantum circuits that can be optimized by a classical computer with regards to a given objective and return the expectation values of some observables estimated by averaging the measurement results obtained.



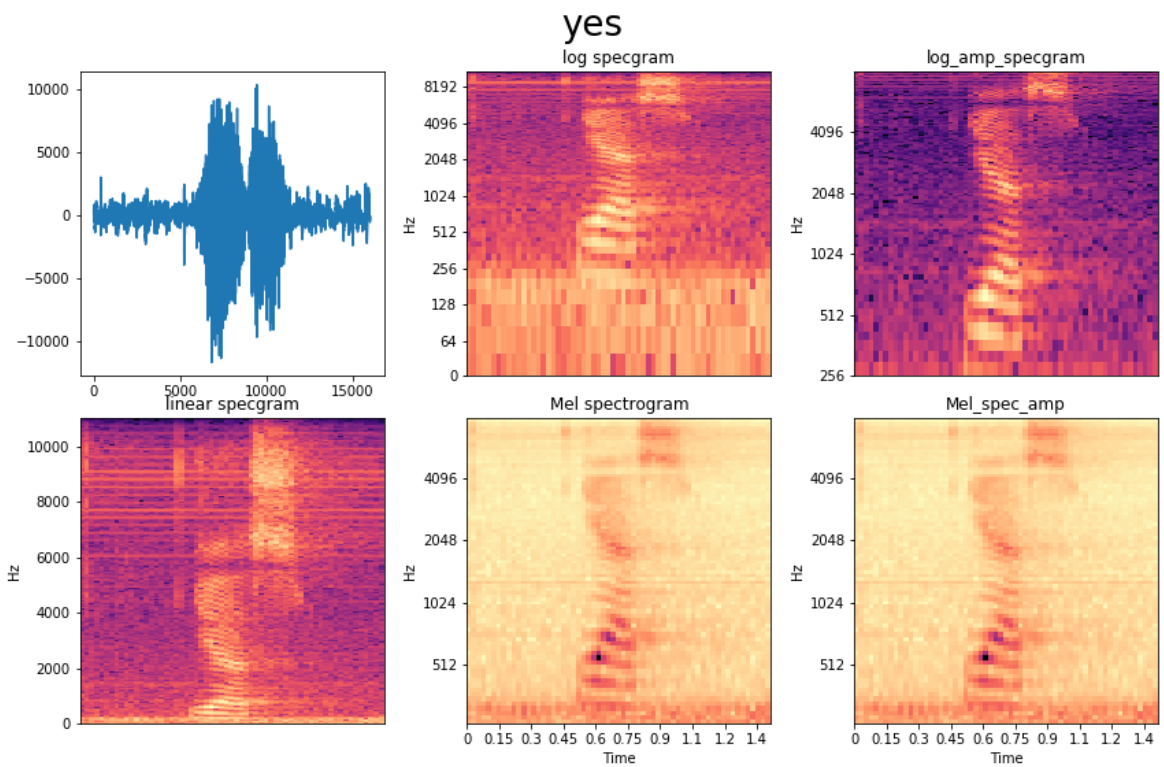
Quantum neural network architecture example

To reduce a number of qubits required for neural network and computational computational complexity transfer learning can be used



My task

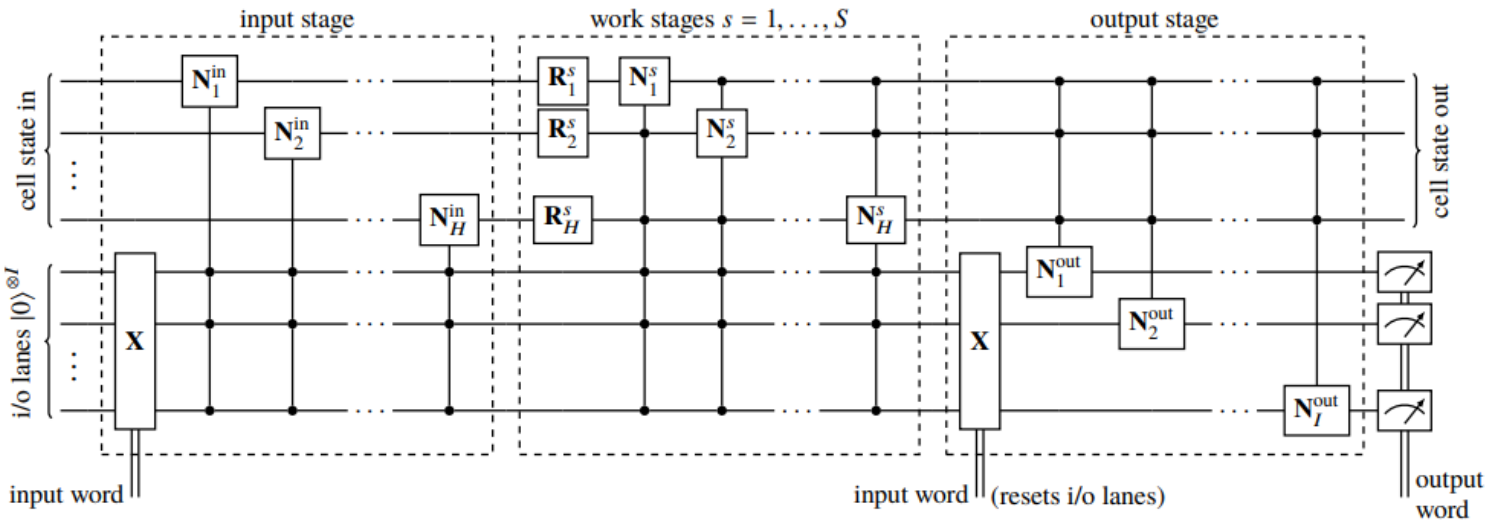
Kaggle Tensorflow Speech Recognition Challenge. Dataset includes 65000 one-second long utterances of 30 short words, by thousands of different people.



My work

- Apply variational circuits based neural networks for speech classification
- Demonstrate that variational circuits are suitable for solving non only toy tasks
- Investigate new variational circuits basic blocks
- Investigate applicably quantum like methods in neural networks

Quantum RNN


input stage
work stages $s = 1, \ldots, S$
output stage
cell state in
cell state out
i/o lanes $|0\rangle^{\otimes I}$
$\mathbf{N}_1^{\text{in}}$
$\mathbf{N}_2^{\text{in}}$
$\mathbf{N}_H^{\text{in}}$
$\mathbf{R}_1^s$
$\mathbf{R}_2^s$
$\mathbf{R}_H^s$
$\mathbf{N}_1^s$
$\mathbf{N}_2^s$
$\mathbf{N}_H^s$
$\mathbf{N}_1^{\text{out}}$
$\mathbf{N}_2^{\text{out}}$
$\mathbf{N}_I^{\text{out}}$
X
X
input word
input word
(resets i/o lanes)
output word

The end