HO HO HO HO HO!

INTRODUCTION TO QUANTUM MACHINE LEARNING

In the era of electronics and information, there is potential for, well, an enormous amount of data that is "only wainting" to be of some use. And the goal of machine learning is to leverage the data to create something useful. Depending on the way one do this, we speak either about

- **supervized learning**: This is the case when the *training data* are *labelled*. The machine is expected to learn to correctly assign the labels. One can then (based on observation) clasify a tumor either as benign or malign, one can ask: is there a dog on this picture? or should this email go into a spam folder or does it seem relevant?
- **unsupervized learning**: We might not know what or how many of labels there are. We would just like to find some structure, some patterns which the problem might have.
- others: **reinforcement learning** (optimises a strategy due to feedback by a reward function) or **recommender systems** (what movie might this person enjoy watching (based on what he liked/watched in the past)?)

And where the quantum computing can help? Well, machine learning is, in principle, lots of linear algebra (at least uses lots of thereof, I'm not an expert). So, voilà, they share the mathematical language. And as it happens, all the Fast Fourier Transform, finding eigenvalues and eigenvectors, solving linear sets of equations, matrix inversion can be done exponentially faster using quantum computers. [*... will try to find links for persuasive materials*]

At this point, I would like to paraphrase Maria Schuld [1]. Concretely, that quantum computation resembles:

- *Support vector machines* or *Kernel methods* – e.g. in how the data gets encoded into a representation that lives in a very high dimensional vectorspace,
- *Neural networks* – in a sence that we apply linear transformations (we lack in nonlinearity; we are speaking about a linear very deep kind of neural network), and we can train quantum computers with stochastic gradient descent very nicely.

But to give a conrete argument, let us take the most obvious point (valid for all types of quantum algorithms). Where, in general, one must consider 2^n possibilities (all n-bit strings composed of 0s and 1s) – 2^n numbers classically, one gets by with only n qubits:

$$\vec{v}_{\text{data}} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix}, \quad d = 2^n \qquad \xrightarrow[\text{encoding}]{\text{classical to quantum}} \qquad |v_{\text{data}}\rangle = \sum_{z\in\{0,1\}^n} v_{\text{int}(z)} |z\rangle, \quad |z| = n.$$

This gives us exponential compression. [*Caution*: This is true only given that we can successfully encode all the neccessary information into the amplitudes and do the right interference for finding the desired solution, which is, of course, the hard part.]

Quantum PCA

...will appear here around 29 December 2020 among with other stuff.

Sources and recommended materials

- To watch:
  https://www.youtube.com/watch?v=Lbndu5EIWvI
  [1] https://www.youtube.com/watch?v=Xh9pUu3-WxM
- To read:
  https://www.scottaaronson.com/papers/qml.pdf
  https://arxiv.org/pdf/1409.3097.pdf
- Here's an awesome course on (classical) machine learning.

TO BE CONTINUED...