A Method to Extend the Capacity of the Quadratic Unconstrained Binary Optimization Problem on the Quantum Annealer.

Once again, consider our binary quadratic cost function $L_N(\boldsymbol{w}^N) = \sum_{s=1}^{S} \left(\frac{1}{N} \sum_{i=1}^{N} w_i h_i(x_s) - y_s \right)^2$. Let us denote it in its standard form $L_N(\boldsymbol{w}^N) = \sum_{1 \leq i < j \leq N} Q_{ij} w_i w_j + \sum_{1 \leq i \leq N} q_i w_i$, equivalent to its unexpanded form, for the sake of simplicity. Suppose that our annealer can handle a problem size of at most $N = 60$. We run our binary quadratic cost function $L_N(\boldsymbol{w}^N)$ (utilizing 60 of our weak classifiers) onto the quantum annealer, and obtain an solution $\boldsymbol{w}_{\boldsymbol{0}}^{60}$ from the quantum annealer, that is supposedly close to the optimal solution (however, it is not necessarily the optimal solution due to the imperfection of the quantum annealer).

If our solution obtained from the quantum annealer is close to optimal, then it must be theoretically true that for most sets $\{i, j\}$ where $w_i w_j = 0$, then Q_{ij} is a relatively large positive real number. This line of reasoning makes sense, since a close-to-optimal solution would most likely avoid appending large positive real numbers to the loss function. In our problem we are trying to handle a problem size of hundreds of features and weights, so more than one anneal will be necessary to determine the optimal weights to be included in the final classifier. Given the information from our previous solution $\boldsymbol{w}_{\boldsymbol{0}}^{60}$, we can alter our binary quadratic cost function in certain ways to reduce our problem size, so that our annealer obtains the capability to handle more than 60 binary variables in a single optimization.

Given our previous solution $\boldsymbol{w}_{\boldsymbol{0}}^{60}$, consider all $\{w_i, w_j\}$, $1 \leq i < j \leq 60$ that are both elements of the vector $\boldsymbol{w}_{\boldsymbol{0}}^{60}$ such that $w_i = w_j = 0$. From our cost function $L_N(\boldsymbol{w}^N)$, we remove the term Q_{ij} and instead add the real number $k_{ij} Q_{ij}$ to q_i and add the real number $(1 - k_{ij}) Q_{ij}$ to q_j, for real parameters $0 < k_{ij} < 1$. This method reduces the problem size by 1 for each relevant pair $\{i, j\}$ due to the fact that it eliminates quadratic terms, while the optimal solution of $L_N(\boldsymbol{w}^N)$ remains invariant. We know that the optimal solution of $L_N(\boldsymbol{w}^N)$ must not change because no new methods of reducing the value of our cost function are introduced by making such changes (it is important to keep in mind that Q_{ij} is positive).

The problem size is henceforth decreased by $\binom{L}{2}$, where L is the number of binary variables in $\boldsymbol{w}_{\boldsymbol{0}}^{60}$ that are 0. Therefore, we can add at most M weak classifiers to our loss function, such that $\binom{60+M}{2} - \binom{L}{2} < \binom{60}{2}$, in other words, the number of quadratic terms in our new loss function should not exceed the number of quadratic terms in our old loss function (before we performed the procedure of eliminating quadratic terms), otherwise the problem would have too large a capacity for our quantum computer. If an annealing process is performed again utilizing $L_{N+M}(\boldsymbol{w}^{N+M})$ to obtain $\boldsymbol{w}_{\boldsymbol{0}}^{60+M}$ and the previously existing N binary variables in $\boldsymbol{w}_{\boldsymbol{0}}^{60+M}$ (the vector formed by only taking the first 60 binary variables of $\boldsymbol{w}_{\boldsymbol{0}}^{60+M}$) are sufficiently similar to the vector $\boldsymbol{w}_{\boldsymbol{0}}^{60}$, then this process will be said to have successfully preserved the previously existing solution and have successfully extended the capacity of the binary classifier. This process can be repeated recursively any number of times before it does not preserve the previously existing solution, or not enough pairs of new binary variables are pairs of zeroes, before we would have to resort to the method of concatenating features by running the quantum computer onto disjoint sets of weak classifiers. $\square$