Quantum Annealing

- Quantum annealing as implemented in the D-Wave machines is a metaheuristic to solve quadratic unconstrained binary optimization problems:
  \[
  \arg\min_{x\in\{0,1\}^n} x^T A x
  \]
- Advantageous to simulated annealing for energy landscapes with high, narrow spikes [1]:
  - Major progress in experimental realization: D-Wave 2000Q has 2048 solid state qubits and D-Wave Advantage has over 5000 qubits.
  - But connectivity restricted to hardware graph: m ≤ 65, 150

**Motivation**

- Previously the A matrix for the quantum annealer was crafted by hand.
- It is necessary to develop one algorithm per problem setting.
- Deep learning covers multiple problem instances with one setting.

**Loss Function**

**Total loss function:**
\[
L(A) = L_{acc}(A) + \lambda_1 L_{uniqueness}(A) + \lambda_2 L_{mlp}
\]

- **Accuracy loss:**
  \[
  L_{acc}(A) = x^{gt}_i A x^{gt}_i - x^{min}_i A x^{min}_i,
  \]
  where \(x^{min} = \arg\min x^{gt}_i A x^{gt}_i\) is the vector with the lowest energy.

- **Uniqueness loss:**
  \[
  L_{uniqueness}(A) = |x^{gt}_i A x^{gt}_i - x^{min}_i A x^{min}_i|,
  \]
  where \(x^{min}\) is the vector with the second lowest energy.

- **MLP loss:**
  \[
  L_{mlp} = \sum_{f\in\mathcal{F}} \frac{1}{|f|} \|f\|_F^2,
  \]
  where \(\mathcal{F}\) is the set of all network layer outputs.

**Derivative of \(L_{acc}(A)\) with regards to the entries of \(A\):**
\[
\frac{\partial L_{acc}(A)}{\partial A_{ij}} = 2(x^{gt}_i) (x^{gt}_j) - 2(x^{min}_i) (x^{min}_j) - \frac{\partial x^{min}_i}{\partial A_{ij}} A_{ij} - \frac{\partial x^{min}_j}{\partial A_{ij}} A_{ij}.
\]

Backpropagation through Quantum annealer is hardly feasible, but the blue term is mostly 0. Hence we approximate the derivative of \(L_{acc}(A)\) with:
\[
\frac{\partial L_{acc}(A)}{\partial A_{ij}} = 2(x^{gt}_i) (x^{gt}_j) - 2(x^{min}_i) (x^{min}_j).
\]

**Website (Code is available):**
https://4dqv.mpib-mpg.de/QuAnt/

**Architecture & Problem Encoding**

- Networks are either 3 or 5 layer MLPs with hidden dimensions of 32 or 78.
  - Hidden layer activation is ReLU and the final activation is a sine function.
- 5 layer setup contains a skip connection from the first to the third layer and uses concatenation for the combination of feature streams.

**Encoding of solutions:**

- Wave matching: Binary representation of permutation table
- Point set registration: Binning the angle intervals into equal sized bins
- 3D rotation estimation: Concatenating 3 binned angles into one vector

**Baselines:**

- Diag: Our method but we set all off-diagonal elements to zero
- Pure: Same network but direct prediction without any QUBO solver
- Direct: Brute Force solution of the input quadratic assignment problem

**Evaluation on D-Wave**

- **Graph matching on the Willow Dataset** [2] (% of correct permutations): Ours \(Diag\) \(Pure\)
  - L=3, H=32: 69 53 90 97
  - L=3, H=78: 65 47 90 97

- **Point set registration on 2D shape dataset** [3] (angle difference to ground truth):
  \[
  \begin{array}{cccc}
    L=3, H=32 & 84 & 0.8 & 11.1 & 1.3 \\
    L=3, H=78 & 72 & 1.1 & 6.3 & 0.7 \\
    L=5, H=32 & 86 & 0.5 & 10.9 & 1.2 \\
    L=5, H=78 & 68 & 0.7 & 7.7 & 0.5 
  \end{array}
  \]

- **3D rotation estimation on ModelNet10** [4] (angle difference to ground truth):
  \[
  \begin{array}{cccc}
    L=3, H=32 & 5.9 & 0.10 & 5.4 & 1.0 \\
    L=3, H=78 & 4.1 & 0.5 & 5.0 & 0.3 \\
    L=5, H=32 & 3.7 & 0.8 & 5.0 & 0.4 \\
    L=5, H=78 & 3.4 & 0.4 & 4.7 & 0.2 
  \end{array}
  \]

**Qualitative Results**

- Evaluation with exhaustive search (ES), quantum annealing (QA) and simulated annealing (SA) returns similar results for test data in rotation estimation.
- Right hand side is the evolution during training on a quantum annealer.
- Red bars in right most histogram indicate projection to valid permutation.

**References**

- [1] A. Das et al., Quantum annealing in a kinetically constrained system, 2005