Quantum annealing: machine learning perspective

K. Tikhonov

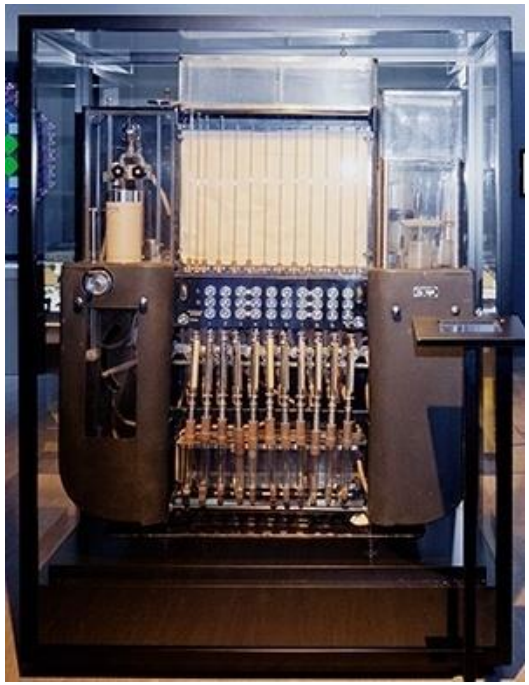
Landau Institute for Theoretical Physics, Karlsruhe Institute for Technology

Skoltech March 27

Outline

- Motivation
- Quantum annealing. Binary perceptron problem: classical and quantum approaches.
- Generalization in neural networks: biased SGD and its connection to quantum annealing.
- Quantum annealing and generative models in hardware: D-Wave examples.

Outline



The Water Integrator

Numbers are stored as water levels (mm precision)

Modelling irrigation channels

Quantum annealer

QBits are stored as current loop orientations

Modelling other quantum systems

(Q) Will quantum computer ever solve a scientific problem?

(A) Yes, in 2035

ML and quantum physics

S Das Sarma et al Physics Today 72, 3, 48 (2019).

- Uncovering phases of matter
- Neural-network representation
- Quantum enhanced machine learning

Quantum-enhanced ML

J Biamonte et al Nature 549 195 (2017)

- Linear algebra-based quantum ML
- Quantum machine learning for quantum data
- Quantum optimization
- Deep quantum learning

Binary perceptron

$$\xi \in \{-1, +1\}^N$$

inputs

$$\tau = \pm 1$$

outputs

m pairs:

$$(+1, -1, ..., -1) \to +1$$

.....

$$(-1, -1, ..., +1) \to -1$$

approximate the function above by $\tau(\xi) = \text{sign}\,(\sigma \cdot \xi)$

$\sigma \in \{-1, +1\}^N$ is the vector weights

Binary perceptron

More formally:

$$\text{minimize} \quad E\left(\{\sigma_j\}\right) = \sum_{\mu=1}^{\alpha N} \Theta\left(-\frac{\tau^\mu}{\sqrt{N}} \sum_{j=1}^{N} \xi_j^\mu \sigma_j\right), \qquad \text{over } \sigma$$

correct outputs τ inputs ξ weights σ

Storage capacity:

How many patterns can (typically) be perfectly memorized?

Binary perceptron

Storage capacity: define $\alpha = \lim_{N\to\infty} m/N$

theory

- W Krauth, M Mezard (1987): $\alpha = 0.833$
- H Horner (1992): $\alpha = 0$

What about evaluating α numerically?

Consider minimizing $E(\{\sigma_j\}) = \sum_{\mu=1}^{\alpha N} \Delta_\mu^n \Theta(-\Delta_\mu), \quad \Delta_\mu \doteq \frac{\tau^\mu}{\sqrt{N}} \sum_{j=1}^{N} \xi_j^\mu \sigma_j$

via classical simulated annealing

Binary perceptron

H Patel (1992)

Computational complexity, learning rules and storage capacities: a Monte Carlo study for the binary perceptron

$$E\left(\{\sigma_j\}\right)=\sum_{\mu=1}^{\alpha N}\Delta_\mu^n\Theta\left(-\Delta_\mu\right),\quad \Delta_\mu\doteq\frac{\tau^\mu}{\sqrt{N}}\sum_{j=1}^{N}\xi_j^\mu\sigma_j$$

Random walk in the space of weights σ according to the weight $\propto e^{-E(\sigma)/T}$

New configurations are generated by random-site single-flip dynamics

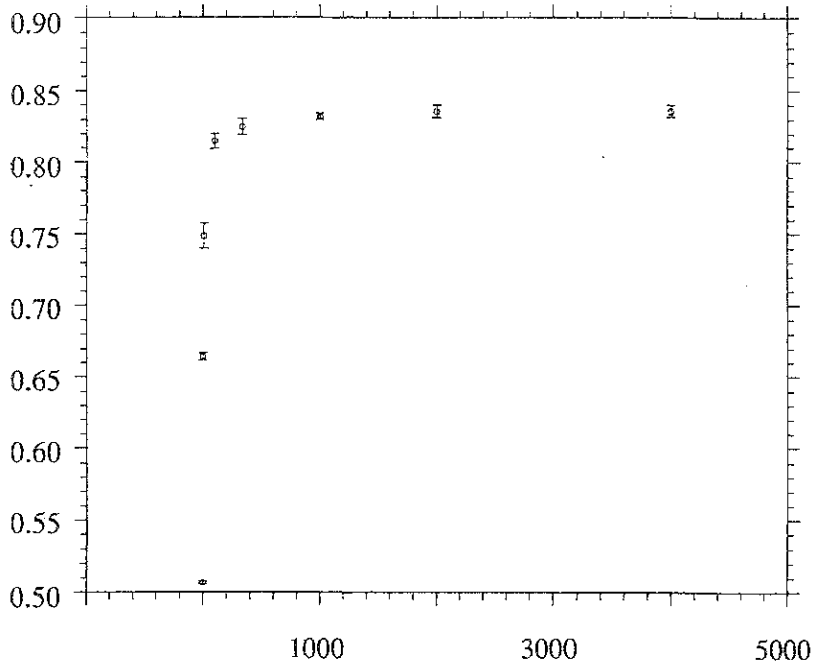
Gradual cooling: $\frac{1}{T}\leftarrow\frac{1}{T}+a$

Binary perceptron

H Patel (1992)

Gradual cooling: $\frac{1}{T} \leftarrow \frac{1}{T} + a$ $\qquad N = 65$

$\alpha(N)$



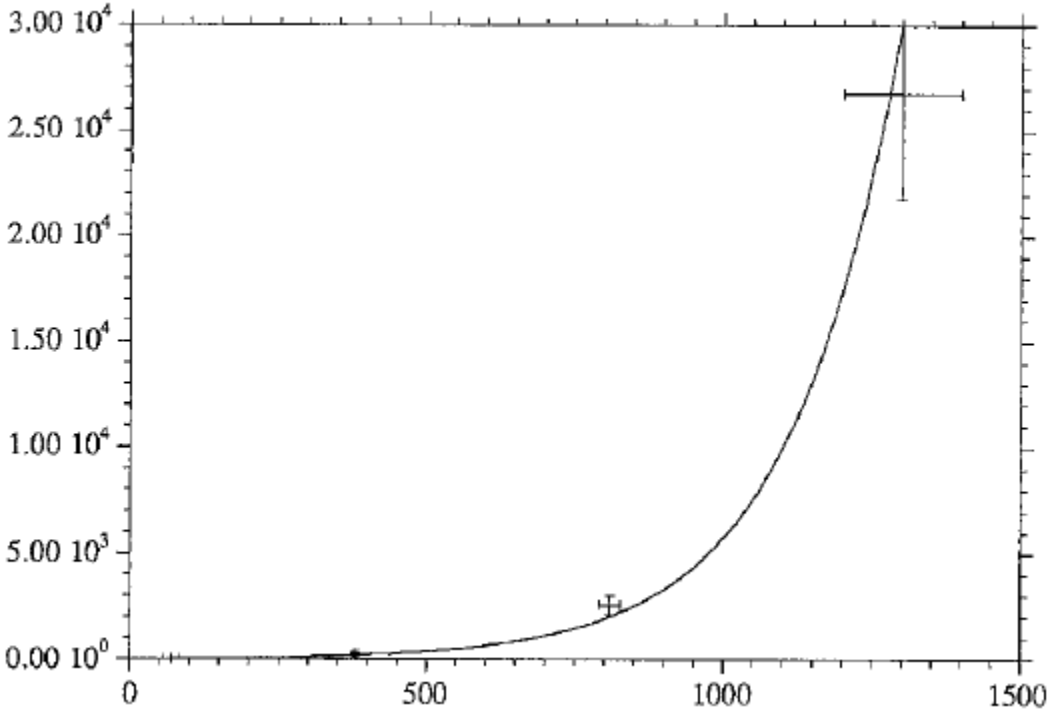
a^{-1}

Binary perceptron

H Patel (1992)

Time to reach $\alpha = 0.7 < \alpha_c = 0.833$

MC time



N

Binary perceptron

W Krauth, M Mezard (1987)

$$\alpha = 0.833$$

theory is "enumerative"

H Horner (1992)

$$\alpha = 0$$

theory considers Glauber dynamics

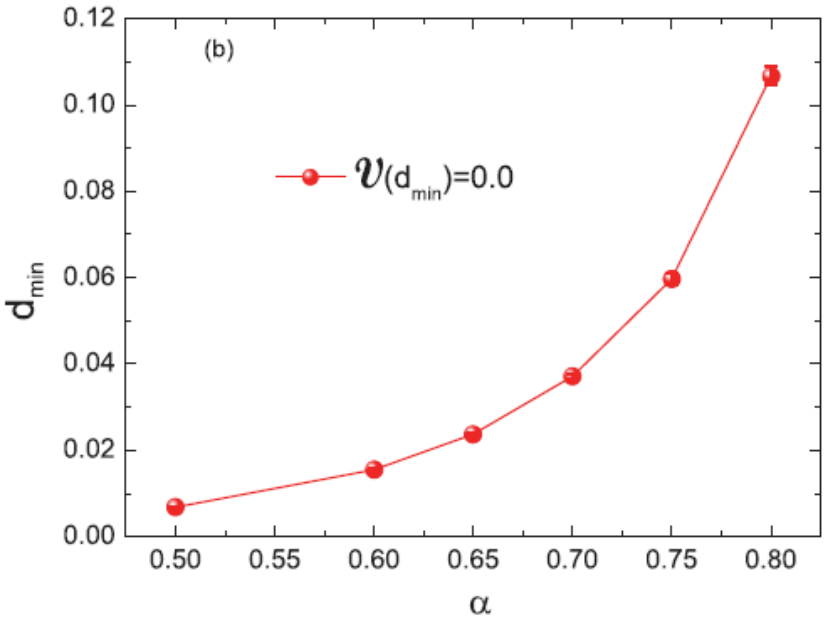
For polynomial MC time, storage capacity vanishes

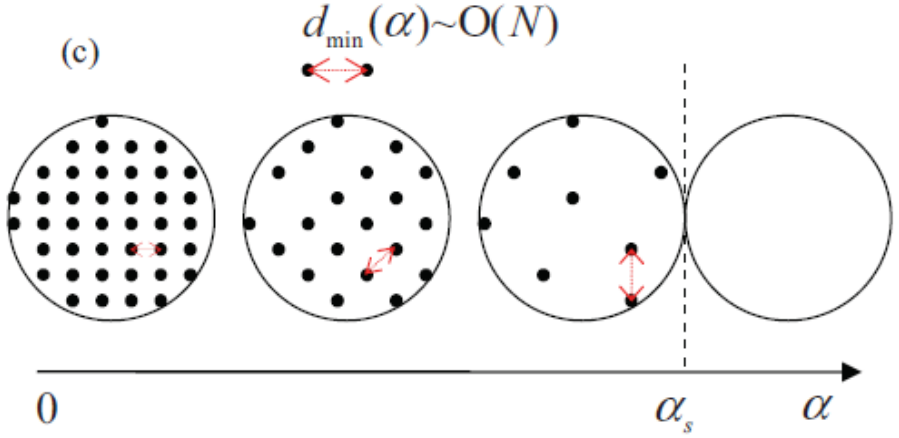
Binary perceptron

H Huang, Y Kabasima (2014)

Structure of the solution space (vanishing $E(\sigma)$)

Typical distance between two solutions d is bounded from below, $d > d_{\min}$





Now the structure of the problem and failure of SA are understood.

What about quantum?

Quantum annealing

$$H_{\mathrm{P}} = -\sum_{\mu} h_{\mu}\sigma_{\mu}^{z} - \sum_{\mu\nu} J_{\mu\nu}\sigma_{\mu}^{z}\sigma_{\nu}^{z} \qquad \sigma_{i=1..N}^{z} = \pm 1$$

2^N possible assignments

Min. H_P is equivalent to searching for lowest eigenvalue of $2^N \times 2^N$ matrix

Example: consider $N = 2$

$$H_P = -\begin{array}{c|cccc} & (+,+) & (+,-) & (-,+) & (-,-) \\ (+,+) & J_1 + J_2 & 0 & 0 & 0 \\ (+,-) & 0 & J_1 - J_2 & 0 & 0 \\ (-,+) & 0 & 0 & -J_1 + J_2 & 0 \\ (-,-) & 0 & 0 & 0 & -J_1 - J_2 \end{array}$$

Quantum annealing

$$H_{\mathrm{P}} = -\sum_{\mu} h_{\mu}\sigma_{\mu}^{z} - \sum_{\mu\nu} J_{\mu\nu}\sigma_{\mu}^{z}\sigma_{\nu}^{z} \qquad \sigma_{i=1..N}^{z} = \pm 1$$

2^N possible assignments

Min. H_P is equivalent to searching for lowest eigenvalue of $2^N \times 2^N$ matrix

Example: consider $N = 2$

$$H_P = -\begin{array}{c|cccc} & (+,+) & (+,-) & (-,+) & (-,-) \\ (+,+) & J_1 + J_2 & 0 & 0 & 0 \\ (+,-) & 0 & J_1 - J_2 & 0 & 0 \\ (-,+) & 0 & 0 & -J_1 + J_2 & 0 \\ (-,-) & 0 & 0 & 0 & -J_1 - J_2 \end{array}$$

Quantum annealing

Quantum evolution

Hermitian $n \times n$ matrix H $\qquad$ n-component state vector $\Psi(t)$

$$i\hbar\partial_t \Psi(t) = H\Psi$$

Adiabatic theorem for slow evolution: ground state remains ground state

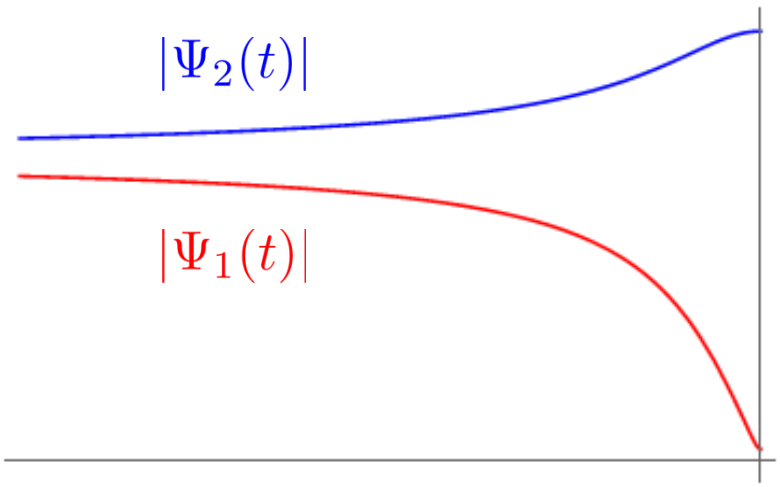
$$\Psi(-T) \longrightarrow \Psi(0)$$

If $\Psi(-T)$ is GS of $H(-T)$

and evolution is slow

then $\Psi(0)$ is (almost) GS of $H(0)$

$$H(t) = \begin{pmatrix} 1 & \gamma t \\ \gamma t & -1 \end{pmatrix}$$



Quantum annealing

More generally:

$$H = H_0 + H_I;\ H_0 = -\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z;\ H_I = -\Gamma\sum_{i=1}^{N}\sigma_i^x.$$

At $\Gamma \gg J$: GS of $H = H_I$ is easy to find: let $\Psi(0) \leftarrow \mathrm{GS}_I$

$\Gamma \to \Gamma(t)$ and find final $\Psi(T)$ according to $i\hbar\partial_t\Psi(t) = H(t)\Psi$ when $\Gamma(T) = 0$

$\Psi(T)$ is almost the GS of $H = H_0 + H_I$

Quantum annealing

How to quantum anneal (i.e., solve $i\hbar\partial_t\Psi(t) = H(t)\Psi$)?

$$i\hbar\partial_t\Psi(t) = H(t)\Psi$$

Solve an ODE | Perform an actual experiment | Quantum MC

In the rest of the talk:

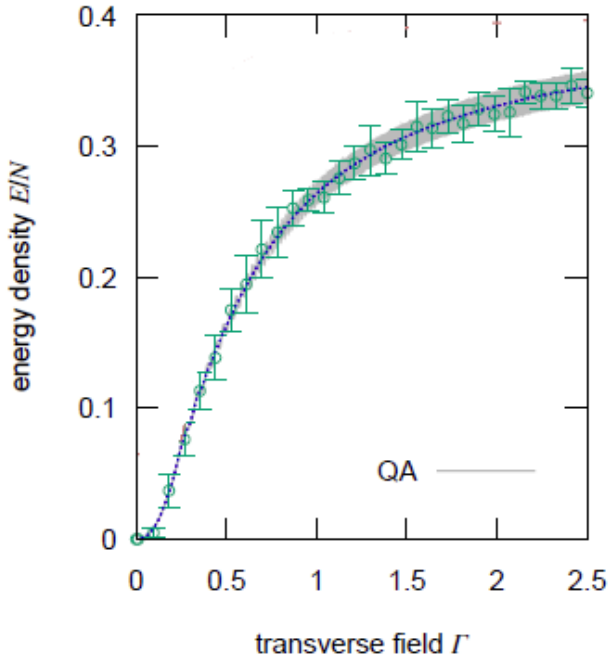
- QA approach to binary perceptron (both ODE and QMC approaches): the latter is extremely effective and instructive
- Experimental side: qA on D-Wave: optimization and sampling from quantum probabilities

QA of binary perceptron

C Baldassi, R Zecchina (2018)

$$E\left(\{\sigma_j\}\right)=\sum_{\mu=1}^{\alpha N}\Delta_\mu^n\Theta\left(-\Delta_\mu\right),\quad \Delta_\mu\doteq\frac{\tau^\mu}{\sqrt{N}}\sum_{j=1}^{N}\xi_j^\mu\sigma_j$$ classical function of N bits

$$\hat{H}=E\left(\{\hat{\sigma}_j^z\}\right)-\Gamma\sum_{j=1}^{N}\hat{\sigma}_j^x$$ quantum hamiltonian ($2^N\times 2^N$ matrix)



$N=21$: real-time QA works well

relatively very small problem size

large instances can not be modelled on a classical computer in this way [should be implemented physically]

Quantum Monte Carlo

How do we probe efficency of QA for large instances?

Instead of solving $i\hbar\partial_t \Psi(t) = H(t)\Psi$, use Quantum MC

Quantum system with Hamiltonian H is described by $\rho = e^{-\hat{H}/T}$

Sampling from quantum distribution $\hat{\rho}$: QMC

$$\hat{H} = E\left(\{\hat{\sigma}_j^z\}\right) - \Gamma \sum_{j=1}^{N} \hat{\sigma}_j^x$$

$$H_{\text{eff}}\left(\{\sigma_j^a\}_{j,a}\right) = \frac{1}{y}\sum_{a=1}^{y} E\left(\{\sigma_j^a\}_j\right) - \frac{\gamma}{\beta}\sum_{a=1}^{y}\sum_{j=1}^{N}\sigma_j^a\sigma_j^{a+1} - \frac{NK}{\beta}$$

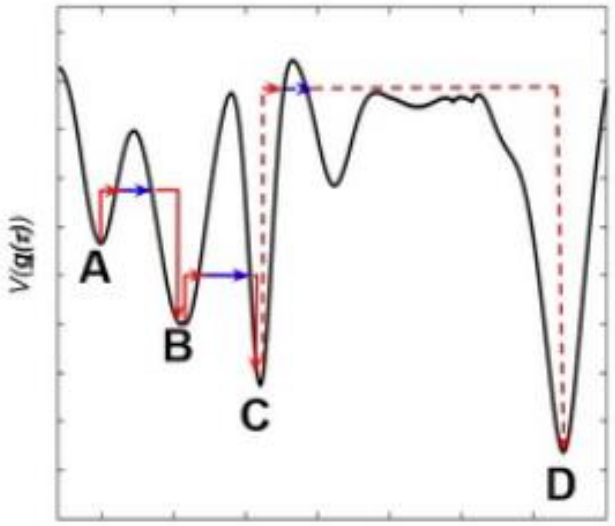
$$\sigma_j^a = \pm 1,\ a = 1, 2, ..., y \text{ and } j = 1, 2, ..., N,\ y \to \infty$$

$$\gamma = \frac{1}{2}\log\coth\left(\frac{\beta\Gamma}{y}\right) \qquad K = \frac{1}{2}y\log\left(\frac{1}{2}\sinh\left(2\frac{\beta\Gamma}{y}\right)\right)$$

Quantum Monte Carlo

Number of replicas $y \sim$ inverse temperature

At finite y the QMC is essentially a mix of QA and SA



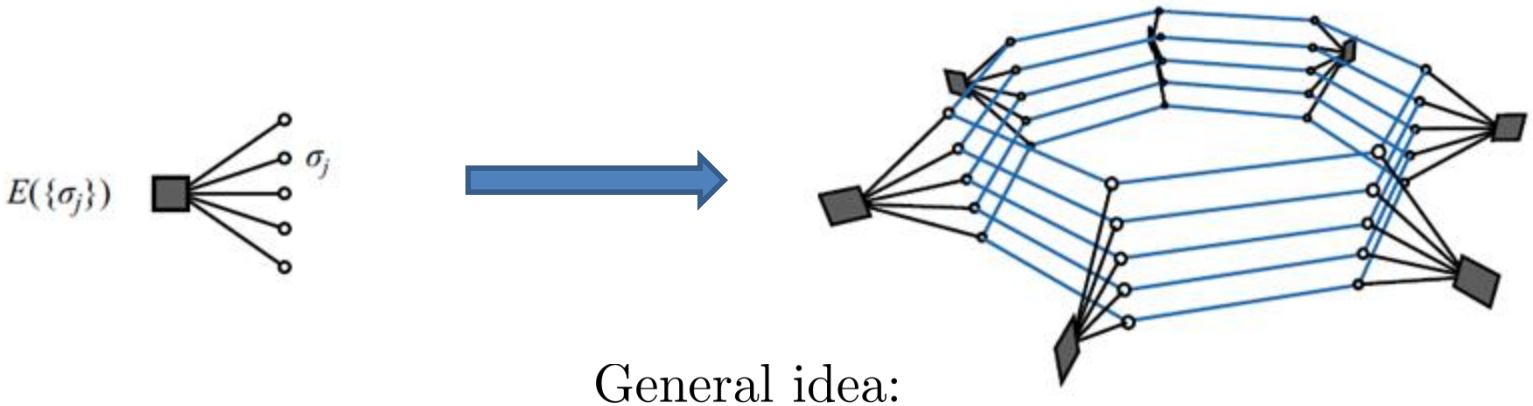
At $y \to \infty$, the pure QA is realized

(for some problems, finite y is optimal)

Quantum Monte Carlo

$$H_{\text{eff}}\left(\left\{\sigma_j^a\right\}_{j,a}\right)=\frac{1}{y}\sum_{a=1}^{y}E\left(\left\{\sigma_j^a\right\}_j\right)-\frac{\gamma}{\beta}\sum_{a=1}^{y}\sum_{j=1}^{N}\sigma_j^a\sigma_j^{a+1}-\frac{NK}{\beta} \qquad (*)$$

$$\sigma_j^a=\pm 1,\ a=1,2,...,y \text{ and } j=1,2,...,N,\ y\to\infty$$



General idea:

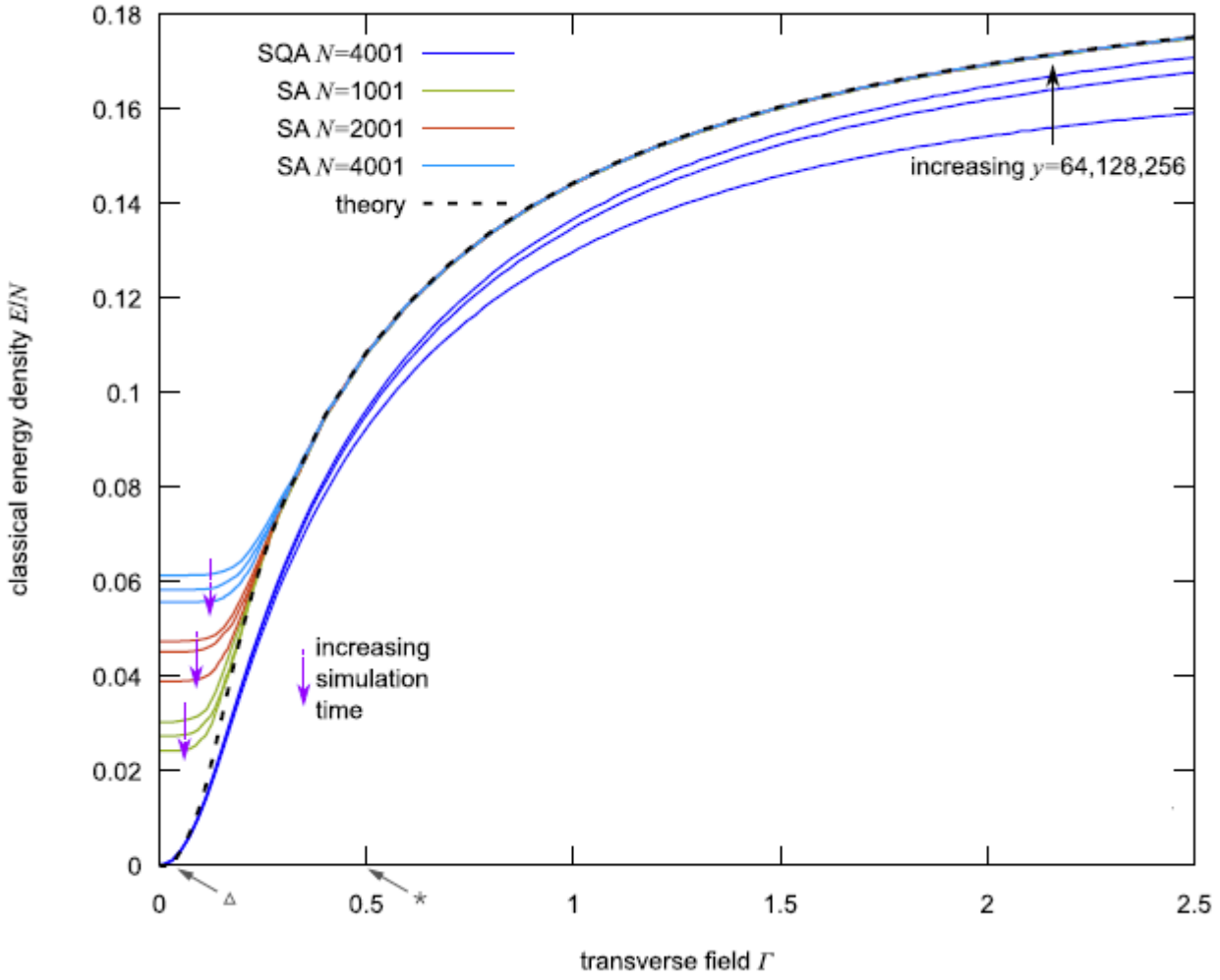
We want to try QA (annealing according to Schroedinger equation)

Impossible to implement exactly on classical computer

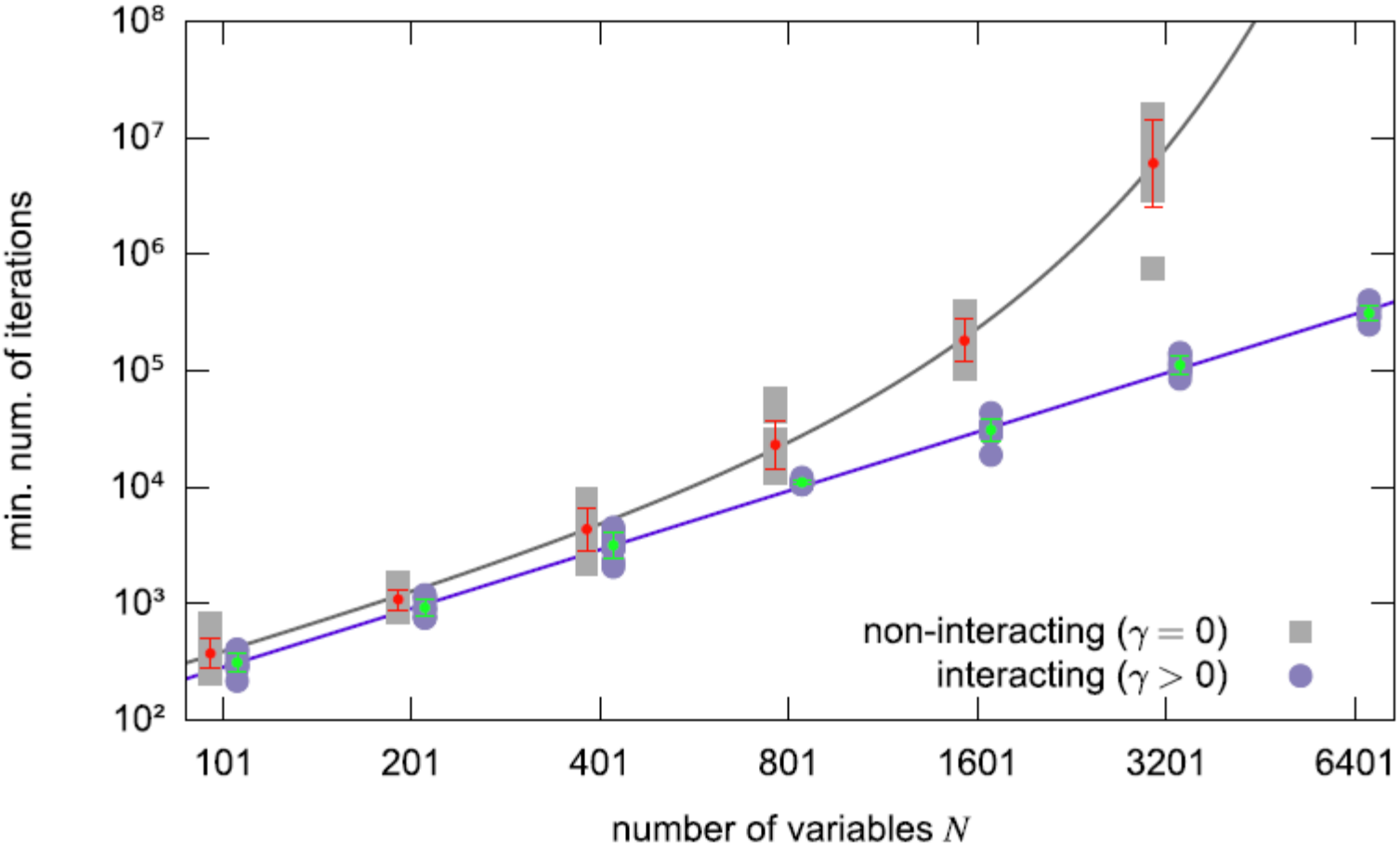
Possible to implement approximately via QMC

QMC is equivalent to classical sampling from (*)

Quantum Monte Carlo on Binary Perceptron

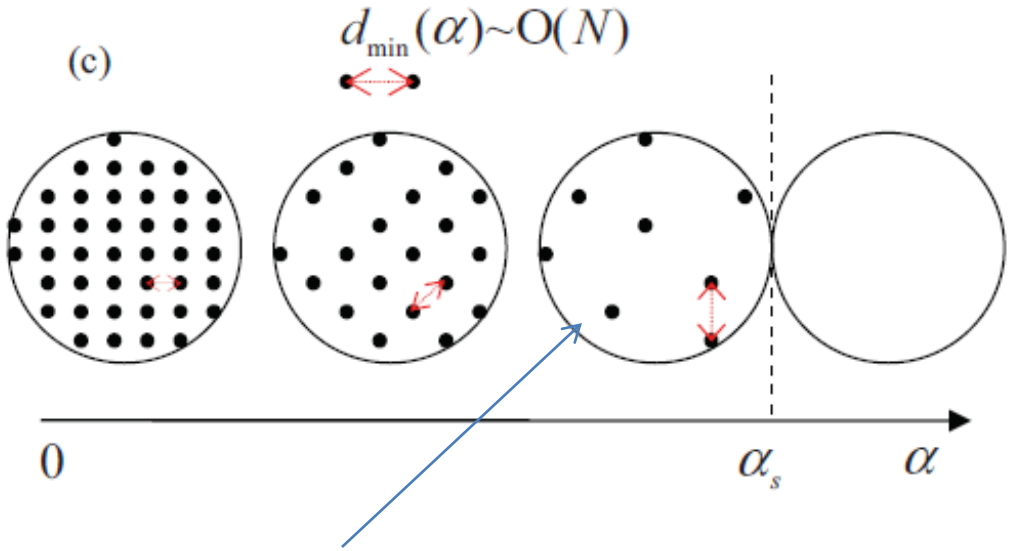


Quantum Monte Carlo on Binary Perceptron



time to solution at $\alpha = 0.3$ level: classical vs quantum (replicated) MC

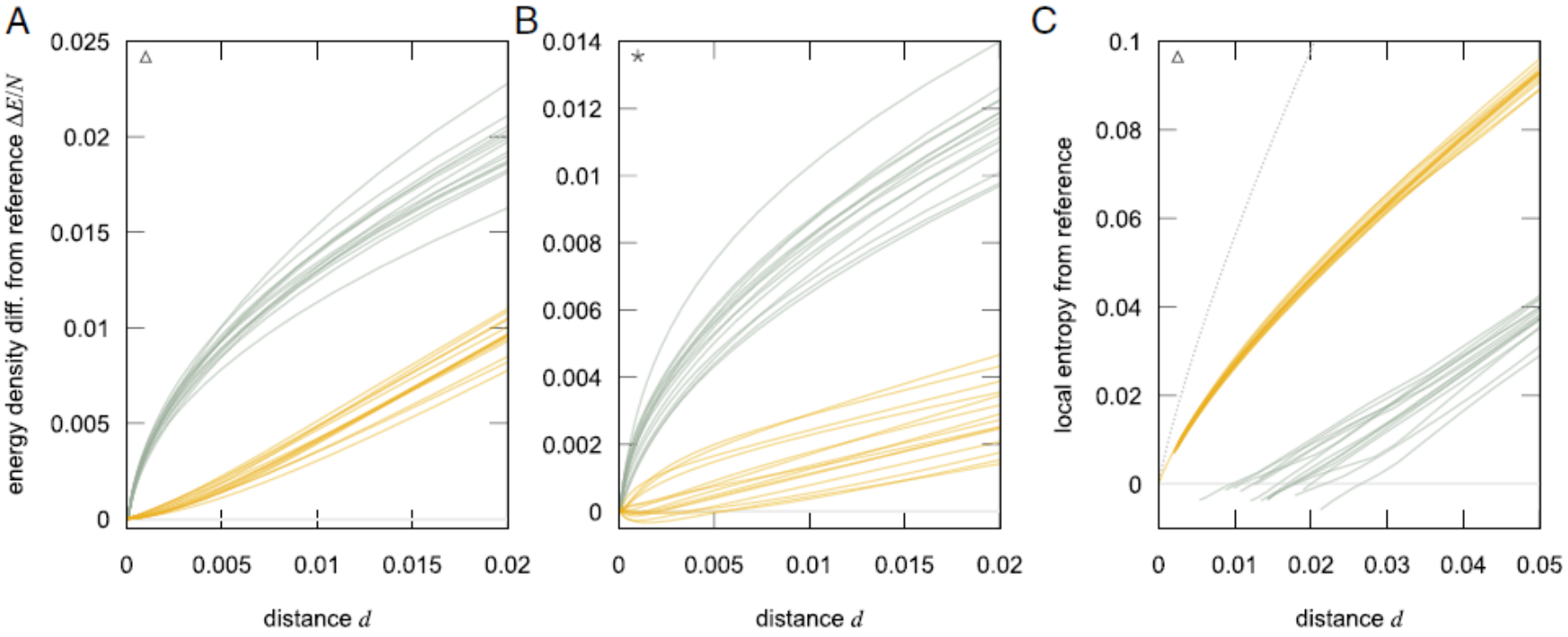
Why is QA more effective?



Some (small fraction) of dots are actually the clusters of O(1) spin flip connected solutions. Quantum part of the energy reduces effective enery of such delocalized clusters and help to find them effectively.

$$\hat{H} = E\left(\{\hat{\sigma}_j^z\}\right) - \Gamma \sum_{j=1}^{N} \hat{\sigma}_j^x$$

Why is QA more effective?



Simulated quantum annealing (QMC) vs Simulated annealing (MC)

For SA: the reference configuration is state of the system; for SQA the mode of replicas: sign $\langle \sigma_i \rangle$

Note a marked qualitative difference in the type of landscape that is typically explored by the two algorithms!

Optimization in DL problems

The binary perceptron is by itself very different from typical ML problem

What can we learn from it?

B Neyshabur et al (2017) Exploring generalization in Deep Learning

- DL are over-parametrized ($N_{\text{param}} \gg N_{\text{data}}$): multiple global minima, all minimize the training error (it may even vanish) but many of them do not generalize well. Finding global minima is not that relevant (*early stopping*)

- Usually trained by SGD: large batch size - trained networks generalize worse than small batch size (implicit regularization by optimization algorithm)

- "Wide"minima generalize better than "sharp"ones

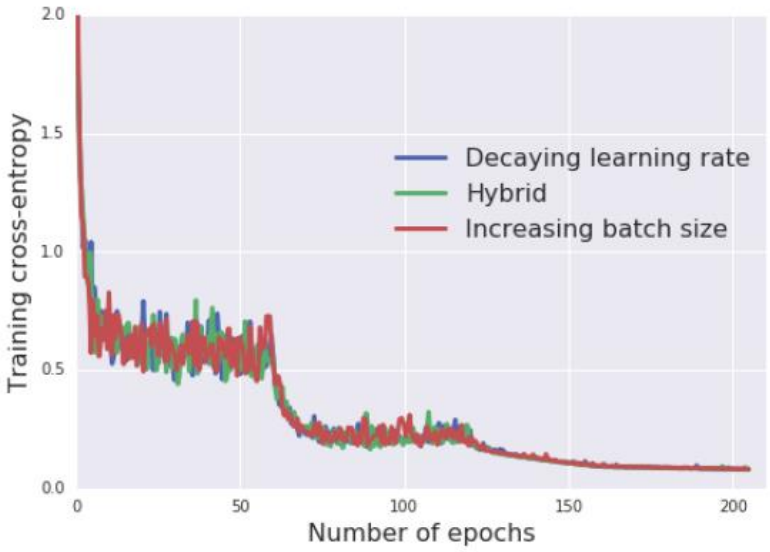
SGD ~ simulated annealing

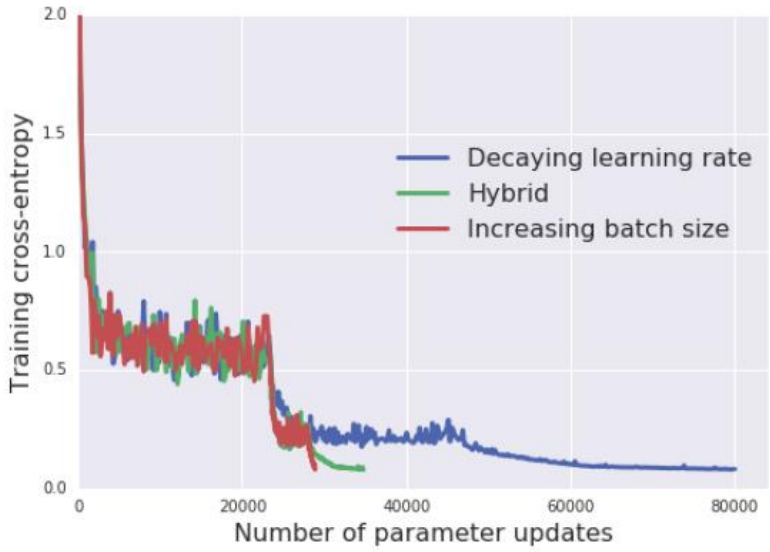
SGD: just a trick to overcome computational bottlenecks?

BSL Smith et al (2018) Don't decay the learning rate, increase the batch size

- When one decays the learning rate, one simultaneously decays the scale of random fluctuations in the SGD dynamics. Decaying the learning rate is simulated annealing

- Increasing the batch size and decaying the learning rate are quantitatively equivalent





Sharp vs Flat minima

H Li et al (2018) Visualizing the loss landscape of neural nets

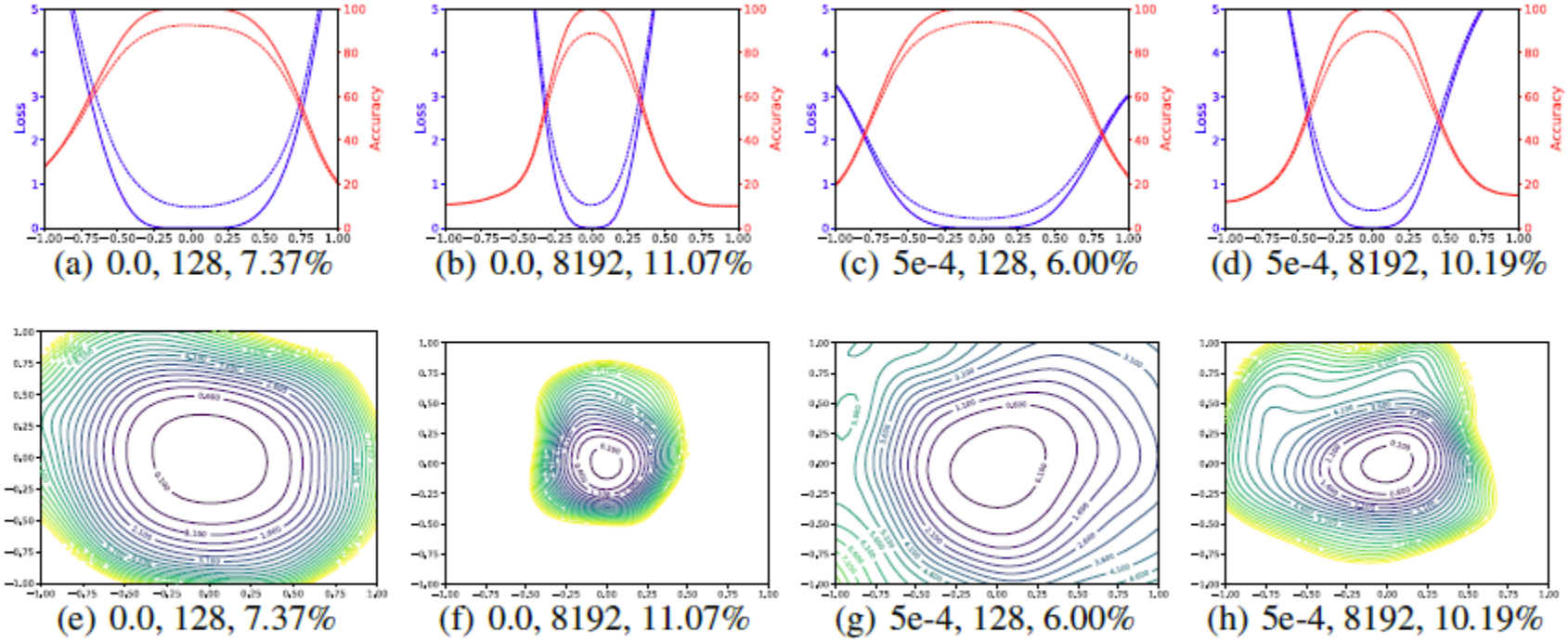


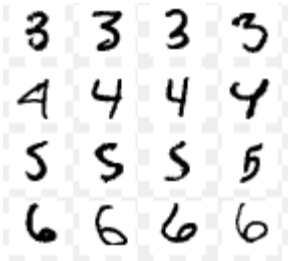
Figure 3: The 1D and 2D visualization of solutions obtained using SGD with different weight decay and batch size. The title of each subfigure contains the weight decay, batch size, and test error.

Entropy-SGD: biasing gradient descent

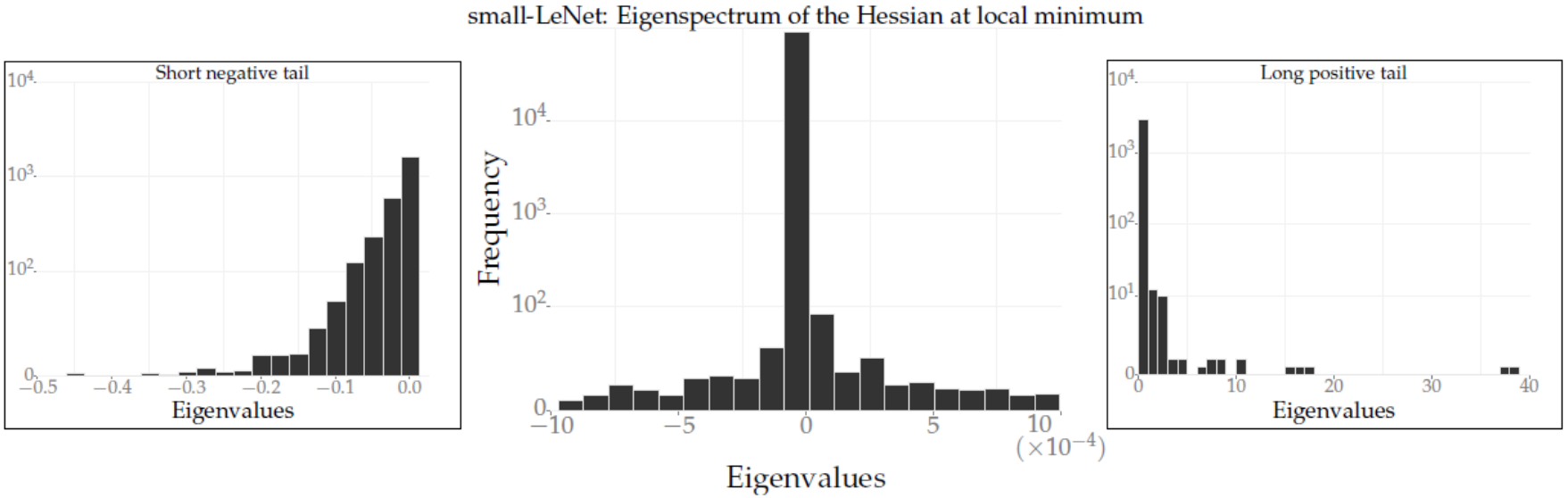
P Chauhari, A Choromanskya, S Soatto, Y LeCunn et al (2017)

Optimization exploiting local geometric properties of the objective function

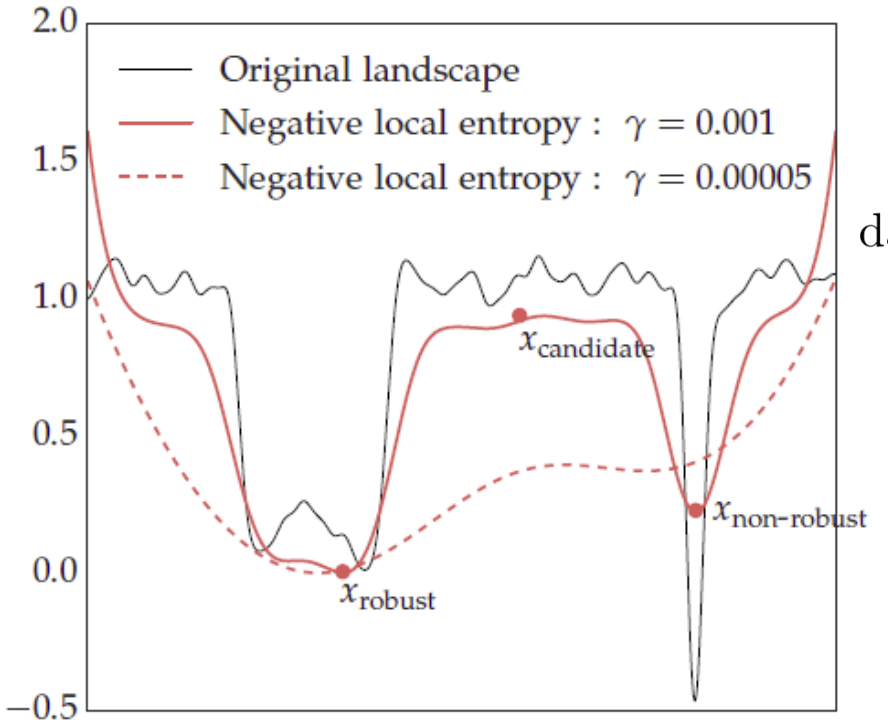
CNN on MNIST



Almost-flat regions of the energy landscape are robust to data perturbations, noise in the activations, as well as perturbations of the parameters, all of which are widely-used techniques to achieve good generalization. This suggests that wide valleys should result in better generalization and, indeed, standard optimization algorithms in deep learning seem to discover exactly that — without being explicitly tailored to do so.



Entropy-SGD: biasing gradient descent



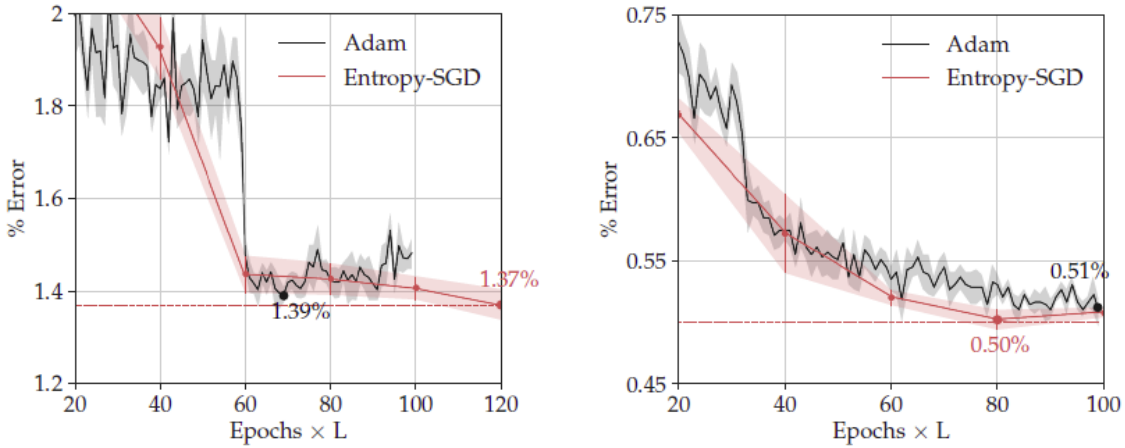
$$F(x, \gamma) = \log \int_{x'} \exp\left(-f(x') - \frac{\gamma}{2} \|x - x'\|_2^2 \right) dx'.$$

Let x be weights of NN, ξ_k - samples from dataset and f - loss function

$$x^* = \operatorname*{argmin}_x \frac{1}{N} \sum_{k=1}^{N} f(x; \xi_k)$$

$$x^*_{\text{Entropy-SGD}} = \operatorname*{argmin}_x -F(x, \gamma; \Xi)$$

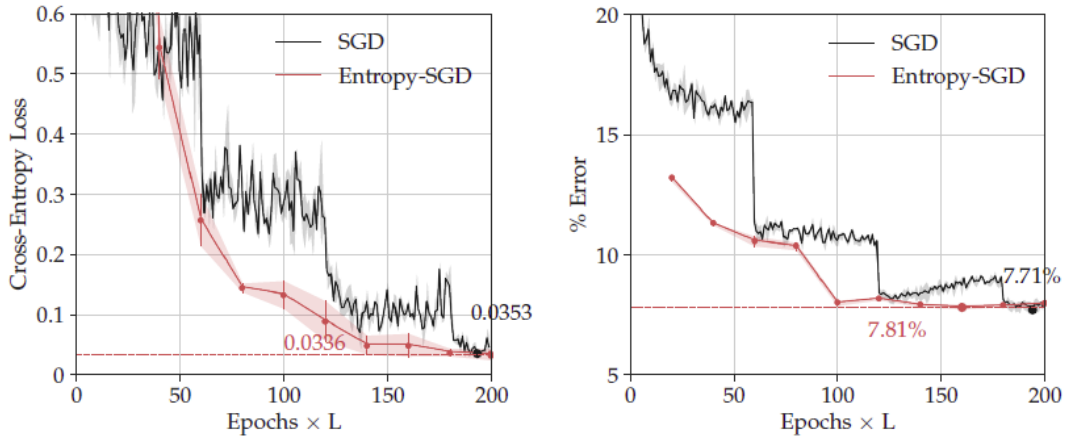
Entropy-SGD: biasing gradient descent



(a) mnistfc: Validation error

(b) LeNet: Validation error

Figure 4: Comparison of Entropy-SGD vs. Adam on MNIST



(a) All-CNN-BN: Training loss

(b) All-CNN-BN: Validation error

Figure 5: Comparison of Entropy-SGD vs. SGD on CIFAR-10

Intermediate summary

- Quantum annealing is a very efficient optimizer for discrete single-layer network: binary perceptron

- Efficiency of QA is due to its ability to target rare but dense clusters of solutions, unaccessible for SA

- Observations above are for discrete-weight networks (single- or multi-layered). From the other side, for usual continuous-weight NN the minimizer biased towards wide minima improves generalization. It would be interesting to find quantum analog of this algorithm.

Optimization of neural networks via finite-value quantum fluctuations

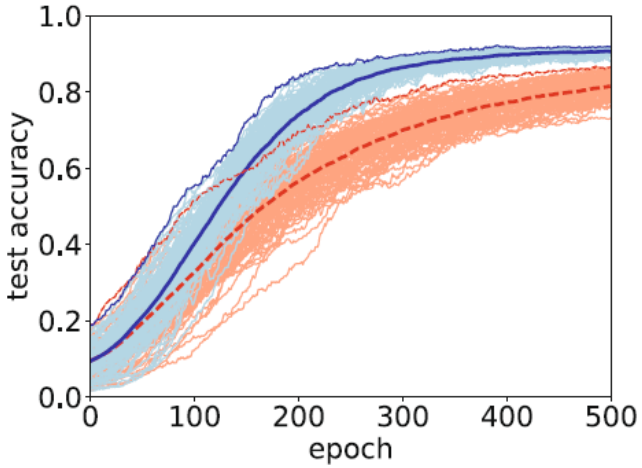
Ohzeki M et al (2018)

$$\hat{H}(t) = V(\hat{\mathbf{w}}) + \frac{1}{2\rho(t)}\hat{\mathbf{p}}^2$$

$$\hat{\rho} = \frac{1}{Z}\exp(-\beta\hat{H}(t))$$

$\longrightarrow$ Replicate the weights: $w \to w_{k=1,2,\ldots,M}$

$$P(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_M) = \prod_{k=1}^{M} \exp\left(-\frac{\beta}{M}V(\mathbf{w}_k) - \frac{M\rho(t)}{2\beta}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|_2^2\right), \quad \beta \to \infty \text{ with } \beta/M = \text{finite}$$



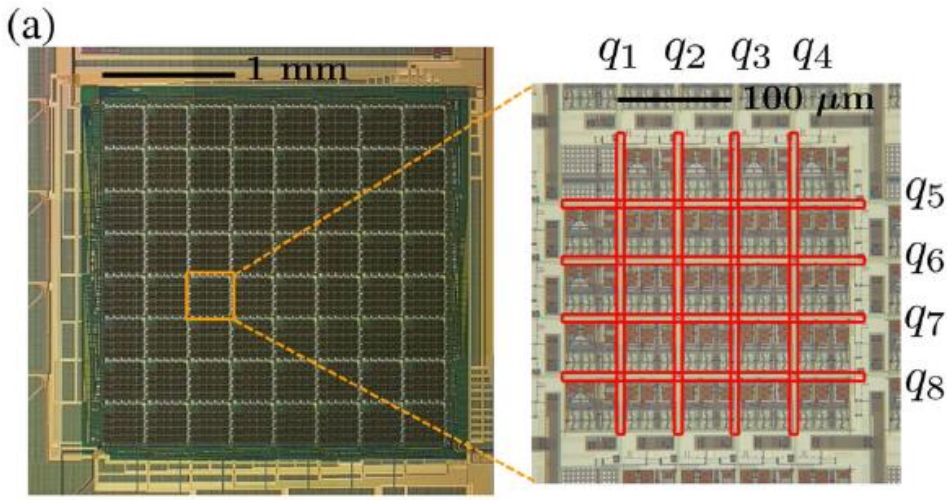
Single-layer NN; MNIST dataset

classical (usual) Adam

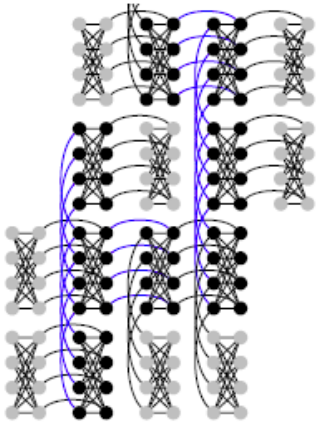
quantum (replicas + interaction) Adam

[no anealing (ρ) is finite in a final state]

QA in hardware (D-Wave)



black box



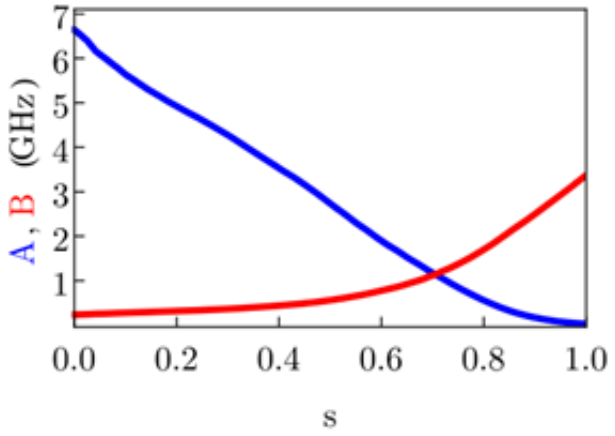
$$H_{\rm P} = -J \sum_{\langle j,j' \rangle \in \text{intra}} \sigma^z_{k,j}\sigma^z_{k,j'} - \sum_{j=1}^{8} h_k\, \sigma^z_{k,j} \; -J \sum_{j \in \text{inter}} \sigma^z_{1,j}\sigma^z_{2,j} \; .$$

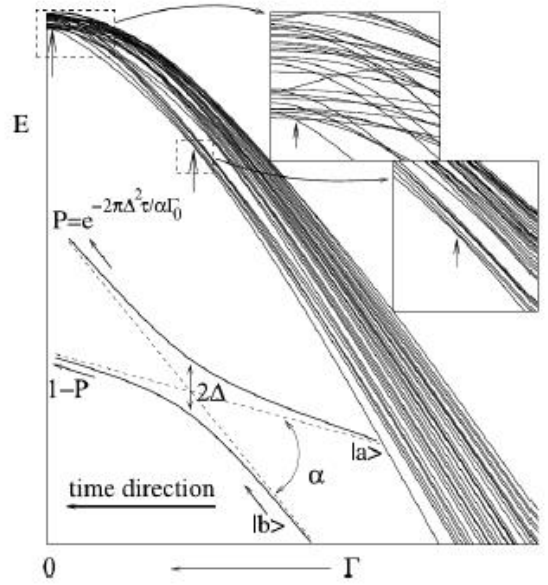
$$H(t) = -A(t) \sum_{j=1}^{N} \sigma^x_j + B(t) H_{\rm P}$$

QA: good and bad

Good: For quantum tunneling to be effective in the hardware, there is no need to maintain (almost) perfect coherence, as required for universal quantum computer.

$$H(t) = -A(t)\sum_{j=1}^{N}\sigma_j^x + B(t)H_{\rm P}$$





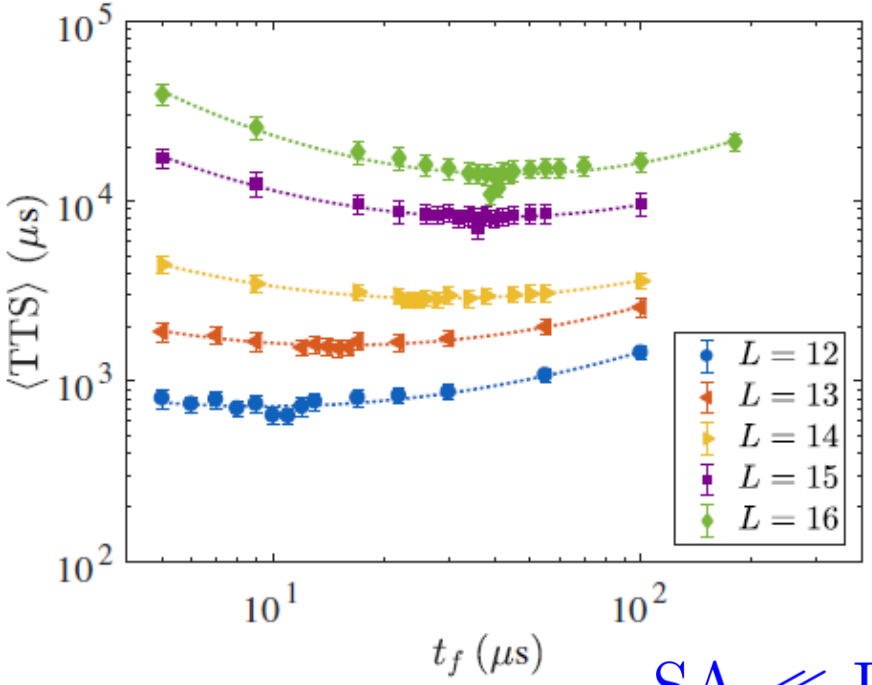
Bad: Adiabaticity: we want the system to always remain in the lowest eigenvector of $H(t)$. If $\dot{H}(t)$ is too large, the system leaves this manifold. QA for some problems is spoiled by presence of tiny energy gaps.

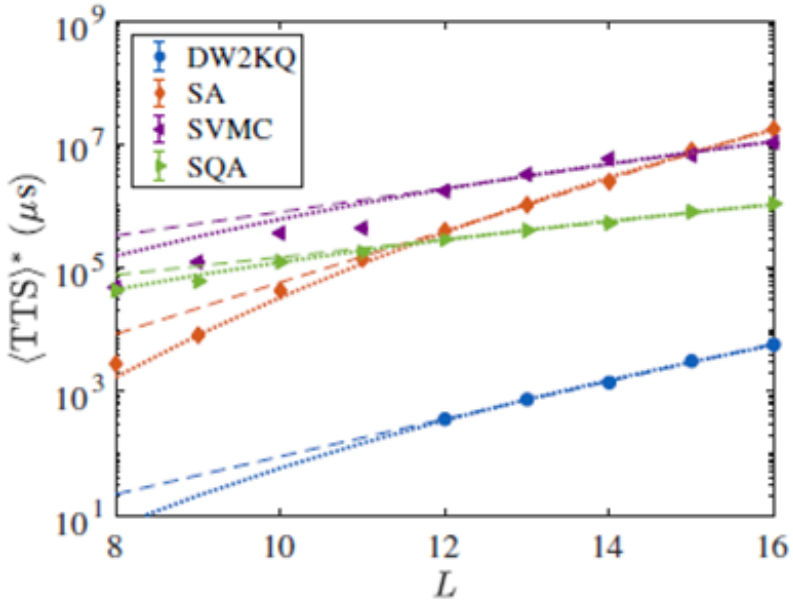
Quantum Annealing on D-Wave

T Albash, D Lidar (2018)

In practice: after each run, measure the system. The outcome is random, with certain probability to end up in the GS, $p_s(t_f)$. $R(t_f)$ depends on the time t_f - the slower the annealing, the higher is probability to end up in the lowest energy state.

$$\mathrm{TTS}(t_f) = t_f R(t_f), \quad R(t_f) = \frac{\ln(1-p_d)}{\ln[1-p_S(t_f)]}.$$

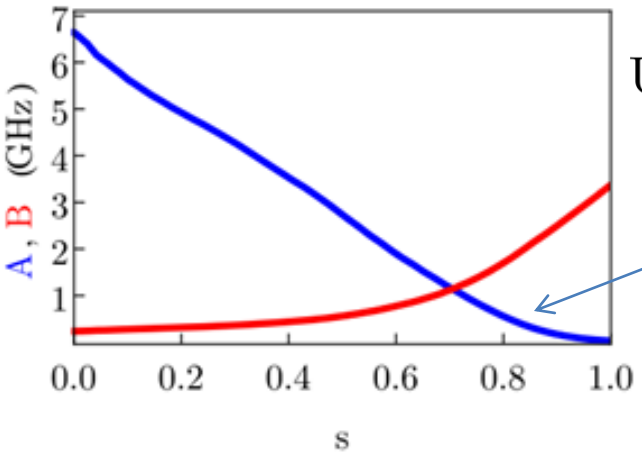




SA $\ll$ DW $\ll$ SQA ($\equiv$ QMC)

Quantum sampling

$$H(t) = -A(t)\sum_{j=1}^{N}\sigma_j^x + B(t)H_{\rm P}$$



MH Amin (2015)

Stop at $s_* < 1$

Under certain conditions for s_* and for $A(s_*) \ll B(s_*)$,

final state is close to equilibrium classical one:

$$P(\mathbf{z}) = \frac{e^{-\beta E(\mathbf{z})}}{\mathcal{Z}} \qquad E(\mathbf{z}) = -\sum_{(i,j)\in\mathcal{E}} J_{ij}z_iz_j - \sum_{i\in\mathcal{V}} h_iz_i$$

In this formulation, quantum dynamics during the times $s < s_*$

serves to thermalize the classical degrees of freedom z_i

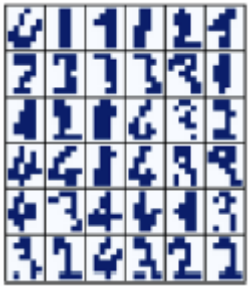
(the temperature β^{-1} is in general unknown and should be inferred)

Quantum sampling on D-Wave

M Benedetti, J Realpe-Gómez, R Biswas,A Perdomo-Ortiz (2017).

I. Learning algorithm

Data: OptDigits dataset reduced to 7x6 binary images (current HW limit)



goal: learn parameters of generative model for the data, ρ_D

$$\rho = e^{-E(z)} \text{ with } \qquad E(\mathbf{z}) = -\sum_{(i,j)\in\mathcal{E}} J_{ij} z_i z_j - \sum_{i\in\mathcal{V}} h_i z_i$$

minimize $S(\rho_D\|\rho) = \mathrm{Tr}\rho_D \ln \rho_D - \mathrm{Tr}\rho_D \ln \rho$ over h, J

$$J_{ij}^{(kl)}(t+1) = J_{ij}^{(kl)}(t) + \eta \frac{\partial S}{\partial J_{ij}^{(kl)}}, \qquad \frac{1}{\beta}\frac{\partial S}{\partial J_{ij}^{(kl)}} = \langle \hat{Z}_i^{(k)} \hat{Z}_j^{(l)} \rangle_{\rho_D} - \langle \hat{Z}_i^{(k)} \hat{Z}_j^{(l)} \rangle_{\rho},$$

$$h_i^{(k)}(t+1) = h_i^{(k)}(t) + \eta \frac{\partial S}{\partial h_i^{(k)}}, \qquad \frac{1}{\beta}\frac{\partial S}{\partial h_i^{(k)}} = \langle \hat{Z}_i^{(k)} \rangle_{\rho_D} - \langle \hat{Z}_i^{(k)} \rangle_{\rho}.$$

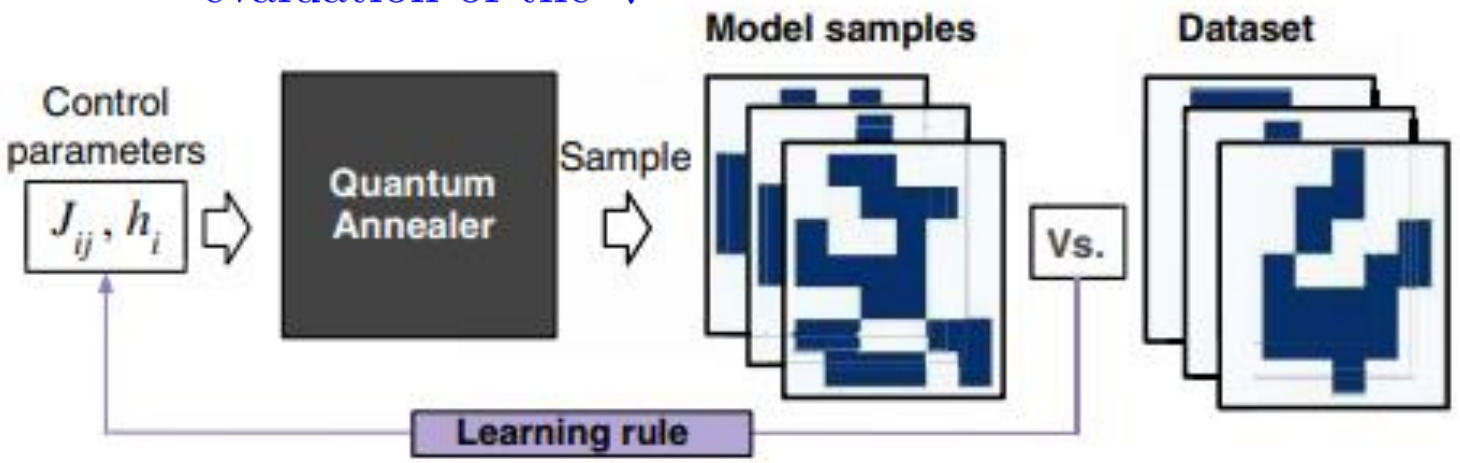
most computation-intensive part: evaluation of the gradients

Quantum sampling on D-Wave

M Benedetti, J Realpe-Gómez, R Biswas,A Perdomo-Ortiz (2017).

I. Learning algorithm

evaluation of the ∇



updating parameters

the role of QA: evaluation of thermal averages $\langle \hat{Z}_i^{(k)} \rangle_\rho$

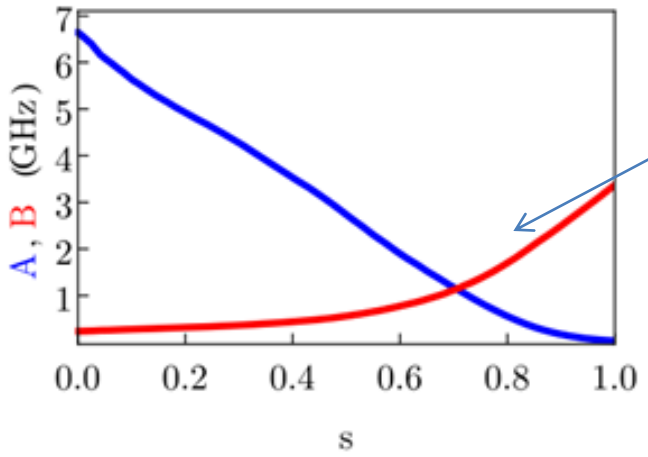
averaging over thermal distribution, parametrized by J, h, QA replaces MCMC routine: hopefully thermalizes better than MCMC

Quantum sampling on D-Wave

$$\rho = \frac{e^{-\beta_{\mathrm{QA}}\mathcal{H}(\tau^*)}}{\mathcal{Z}}$$ parametrized by device parameters

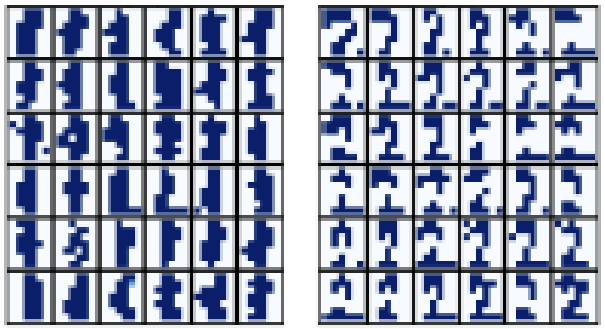
II. Sampling from ρ

$$H(t) = -A(t)\sum_{j=1}^{N}\sigma_j^x + B(t)H_{\mathrm{P}}$$



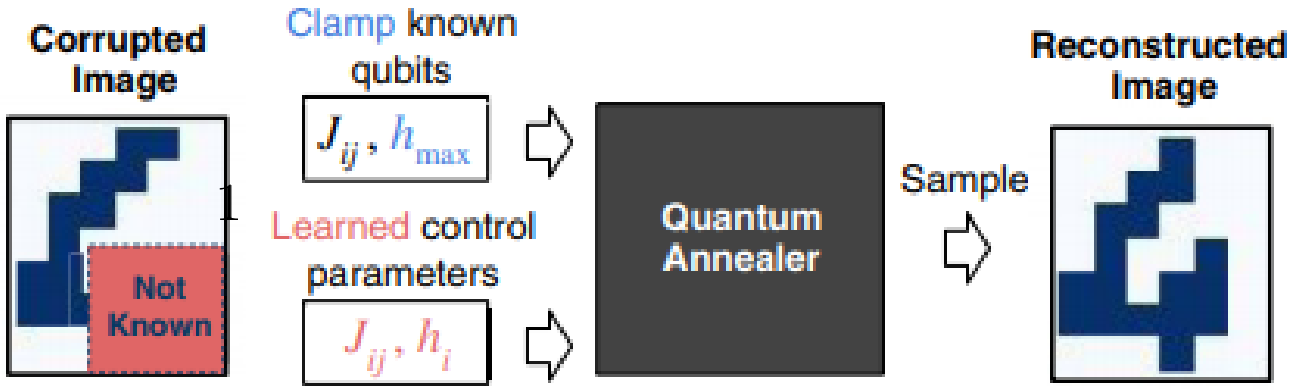
Stop at $s_* < 1$

Meazure z_i variables

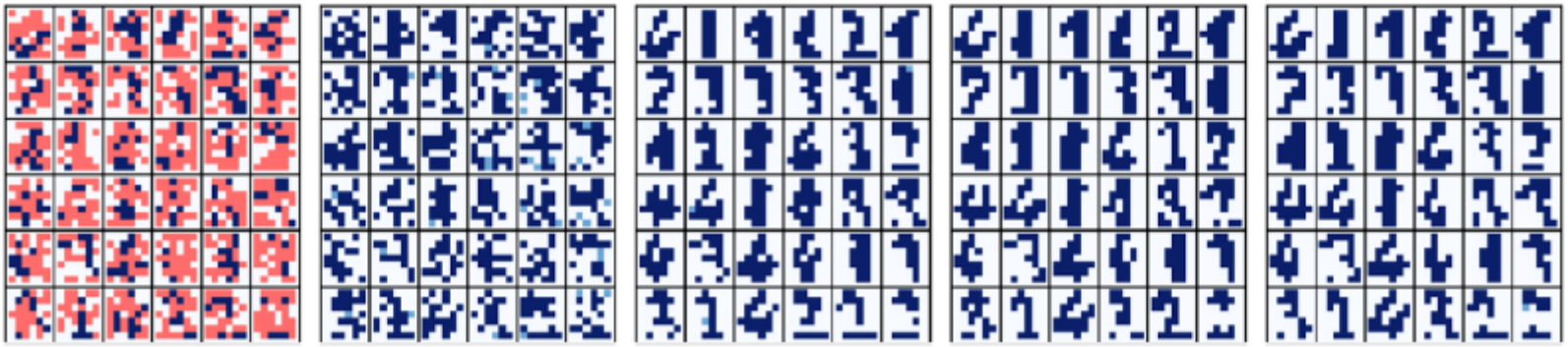


Quantum sampling on D-Wave

III. Image restoration



(strongly bias variables z_i for which the values are known, sample the rest according to ρ)



upd. (phase I) → 1 100 1000 6000 does not rotate

Summary

- Binary Perceptron problem: playground for showing suppremacy of QA over SA
- Success of QA in Binary Perceptron relies on its ability to target wide minima (relevant for generalization capabilities)
- The approach generalizes from Binary Perceptron (single layer, discrete weights) to multi-layer discrete and continous weights. On standard datasets leads to excellent generalization without explicit regularization
- Efficient hardware evaluation of Gibbs averages for training generative models
- Fair sampling from multivariate $\mathcal{N} \sim 2^{50}$ probability distributions is expected to be the first proof of *quantum supremacy*