MACHINE LEARNING IN QUANTUM COMPUTERS VIA GENERAL BOLTZMANN MACHINES: GENERATIVE AND DISCRIMINATIVE TRAINING THROUGH ANNEALING

A PREPRINT

Siddhartha Srivastava
Department of Aerospace Engineering
University of Michigan Ann Arbor
sidsriva@umich.edu

Veera Sundararaghavan
Department of Aerospace Engineering
University of Michigan Ann Arbor
veeras@umich.edu

February 12, 2020

ABSTRACT

We present a hybrid Quantum-Classical method for learning Boltzmann machines (BM) for generative and discriminative tasks. Boltzmann machines are undirected graphs that form the building block of many learning architectures such as Restricted Boltzmann machines (RBM's) and Deep Boltzmann machines (DBM's). They have a network of visible and hidden nodes where the former are used as the reading sites while the latter are used to manipulate the probability of the visible states. BM's are versatile machines that can be used for both learning distributions as a generative task as well as for performing classification or function approximation as a discriminative task. We show that minimizing KL-divergence works best for training BM's for applications of function approximation. In our approach, we use Quantum annealers for sampling Boltzmann states. These states are used to approximate gradients in stochastic gradient descent scheme. The approach is used to demonstrate logic circuits in the discriminative sense and a specialized two-phase distribution using generative BM.

1 Introduction

Boltzmann machines (BM) are graph models that capture dependencies between variables by associating a probability to each combination of variables based on an Ising–type energy expression. Quantum annealers are physical realizations of Boltzmann machines, in which each sample (containing multiple qubit states with binary read-out values) corresponds to

an energy with the probability increasing with decreasing energy. An example of nomenclature of Boltzmann machines is shown in Fig. 1. The model observes the data by using "visible units" (v) and constructs abstract representation of the data by using "hidden units" (h). The challenge in this approach is to learn weights (eg. interconnection J) that are the best for hidden representation of data.

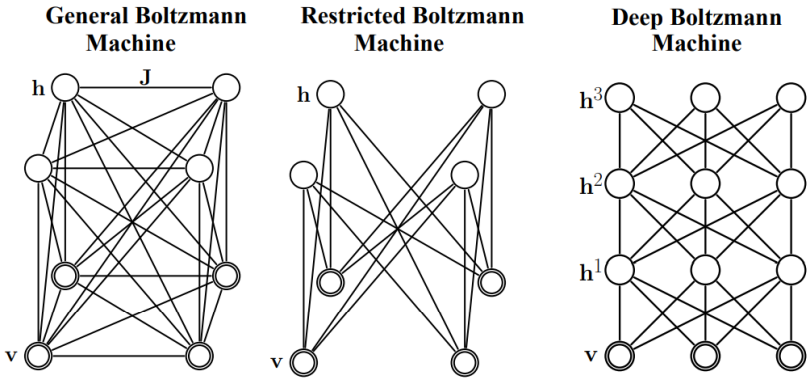


Figure 1: *Nomenclature of Boltzmann machines from [1]*

While all three models in Fig. 1 are Boltzmann machines, the particular nomenclature adopted is because of slowness of learning in general Boltzmann machines, which is NP–hard. Hinton (2002) [2] proposed Restricted Boltzmann machine (RBM), which has no connections between hidden units and can be trained efficiently on a classical computer. Boltzmann machines have received a lot of attention as building blocks of multi-layer learning architectures for speech and image recognition [3, 4]. The idea is that features from one RBM can serve as input to another RBM. By stacking RBMs in this way, one can learn features from features in the hope of arriving at a high level representation. It is known that approximate inference in deep Boltzmann machines can handle uncertainty in better way and deal with ambiguous data [1]. However, the learning process of deep Boltzmann machines on classical computers is quite slow due to the complexity involved in computing the likelihood of an undirected model or its gradient. Among classical methods, sampling techniques are typically employed but can be time-consuming due to the slow mixing of Gibbs sampling. This issue has been addressed in literature by methods that need fewer sampling steps, for instance, Contrapositive divergence (CD), Persistent Contrapositive divergence (PCD) and Fast Persistent Contrapositive divergence (FPCD) learning methods [2, 5].

Quantum computing has to potential to significantly speed up training of Boltzmann machines. The stochastic gradient based training of the objective (such has log–likelihood functions) can be accelerated by sampling from quantum states. In noisy intermediate scale quantum computers (NISQ) the approach of amplitude amplification and estimation have been developed that results in a quadratic speedup with respect to sample acceptance probabilities [6]. However, NISQ devices are limited small number of quantum bits, that restricts the problem to Boltzmann machines with small number of nodes. More recent work thus, has been along the lines of quantum/classical methods where classical training methods are enhanced with quantum state samples. The use of quantum annealers are promising for quantum/classical training since a large number of qubits are available and the training takes advantage of measurements on the physical

realization of the Boltzmann machine [7, 8]. Most importantly, the approach allows training of general Boltzmann machines as along as it can be embedded on the device. Our goal in this paper is to demonstrate the use of quantum annealers for discriminative and generative tasks involving Boltzmann machines.

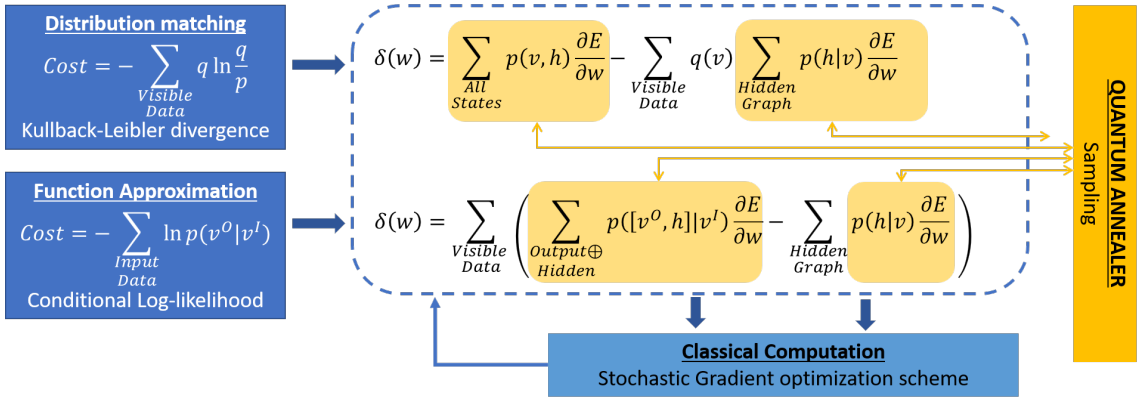


Figure 2: An illustration of the hybrid Quantum-Classical computation technique: Quantum Annealer is used as a Boltzmann sampler while the gradient optimization is carried out using classical computation

2 Mathematical description

A Boltzmann Machine is a probabilistic graphical model defined on a complete graph of binary variables. We can partition the graph into m "visible" nodes taking up values observed during training denoted by vector, $\boldsymbol{v}$, and n "hidden" nodes where values must be inferred taking up values denoted by vector, $\boldsymbol{h}$. The probability of observing a state in the Boltzmann Machine is governed by its energy function:

$$E(\boldsymbol{v},\boldsymbol{h}) = \sum_{i=1}^{m} \alpha_i^v v_i + \sum_{j=1}^{n} \alpha_j^h h_j + \sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{ij}^{vh} v_i h_j + \sum_{i=1}^{m}\sum_{j=i+1}^{m} \gamma_{ij}^{vv} v_i v_j + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \gamma_{ij}^{hh} h_i h_j \tag{1}$$

Where $\boldsymbol{\alpha^v}$ and $\boldsymbol{\alpha^h}$ denote self-interaction at nodes and $\boldsymbol{\gamma^{vh}}$, $\boldsymbol{\gamma^{vv}}$ and $\boldsymbol{\gamma^{hh}}$ denotes the node–node interaction terms. Together, these constitute the set of parameters ($\boldsymbol{\theta}$) that need to be learnt for achieving a desired probability of visible states $\boldsymbol{v}$. The Boltzmann Machine is equivalent to a model from statistical physics known as the Ising model. The Ising model is a mathematical description of a physical phenomenon, which suggests a procedure for sampling states from a quantum annealer.

The distribution of equilibrated states can be modeled, at least approximately, as a Boltzmann distribution:

$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})/k_B T} \equiv \frac{1}{Z} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})} \tag{2}$$

Here, Z denotes the partition function and is estimated as $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}$. The probability of a particular visible state $\boldsymbol{v}$,

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})} \tag{3}$$

2.1 Distribution matching using BMs

In the last section, we saw that each visible state occurs with a probability determined by the Boltzmann distribution. This implies that in a sufficiently long simulation of a BM each visible state is sampled with a known probability distribution which is referred to as the model probability distribution, p (often written as p_θ to signify dependence on the parameters of energy). It is desirable that this distribution matches that of the data, denoted as the true probability, q. The process of learning distributions via BM essentially requires one to estimate the parameters θ such that p_θ and q are close to each other. The Kullback-Leibler divergence $D_{KL}(q||p)$ is defined as,

$$D_{KL}(q||p) = -\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v})}{q(\boldsymbol{v})}$$

where the set $\{\boldsymbol{v}^1, ..., \boldsymbol{v}^D\}$ represents the D data points. The KL divergence is always non-negative with $D_{KL}(q||p) = 0$ if and only if $q = p$ almost everywhere. For this property, this function is chosen to be the cost function. It is minimized using gradient based optimization techniques. The gradient is estimated as (See Appendix A for part of the calculation):

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} = \beta \sum_{d=1}^{D} q(\boldsymbol{v}^d) \left(\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^d) \frac{\partial E(\boldsymbol{v}^d,\boldsymbol{h})}{\partial \boldsymbol{\theta}} \right) - \beta \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') \frac{\partial E(\boldsymbol{v}',\boldsymbol{h}')}{\partial \boldsymbol{\theta}} \tag{4}$$

We first specialize this gradient for the Ising model (Eq (1)). We drop the β constant as it can be subsumed in the learning rate:

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \alpha_i^v} = \sum_{d=1}^{D} q(\boldsymbol{v}^d) v_i^d - \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') v_i' \tag{5a}$$

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \alpha_i^h} = \sum_{d=1}^{D} q(\boldsymbol{v}^d) \left(\sum_{\boldsymbol{h_i}} p(h_i|\boldsymbol{v}^d) h_i \right) - \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') h_i' \tag{5b}$$

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \gamma_{ij}^{vh}} = \sum_{d=1}^{D} q(\boldsymbol{v}^d) \left(\sum_{\boldsymbol{h_j}} p(h_j|\boldsymbol{v}^d) v_i^d h_j \right) - \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') v_i' h_j' \tag{5c}$$

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \gamma_{ij}^{vv}} = \sum_{d=1}^{D} q(\boldsymbol{v}^d) v_i^d v_j^d - \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') v_i' v_j' \tag{5d}$$

$$\frac{\partial D_{KL}(q||p(\boldsymbol{\theta}))}{\partial \gamma_{ij}^{hh}} = \sum_{d=1}^{D} q(\boldsymbol{v}^d) \left(\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^d) h_i h_j \right) - \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') h_i' h_j' \tag{5e}$$

2.1.1 Approximating the gradient

The second terms in Eq(5a-5e) are summed over all possible states for a given parameter and therefore estimating this term exactly entails the tedious task of calculating 2^{m+n} probabilities. However, we can approximate the probabilities by sampling the states using quantum annealers. The assumption here is that the annealer samples the states with the same Boltzmann distribution. The states which are not sampled are assigned zero probability. It is evident through experimentation that with increasing number of samples the sample probability approaches the model probability. The first terms in Eq(5a,5d) are independent of $\boldsymbol{\theta}$ and are only evaluated once. For the first terms in Eq (5b, 5c and 5e), required more work. If the visible data appears in the sampled states then the conditional probabilities can be directly calculated by considering only the samples with the specified visible data. In the event that there is no sample corresponding to some specific visible data, more samples can be generated. Since it is difficult to ensure sampling of a specific visible data, we just simulate the hidden subgraph of the Boltzmann machine i.e. we ignore the visible nodes and the connections between the visible node. The self-interaction term is augmented to be $b_j \rightarrow \sum_{i=1}^{m}(b_j + c_{ij}v_i^d)$. It is observed from Eq(6) that this procedure ensures that the energy gaps between the states of hidden subgraph remains unchanged. Since the Boltzmann distribution is unchanged by translation of energy, the samples of hidden subgraph are directly used to estimate the conditional probability in the second term of Eq (5b, 5c and 5e)

$$E(\boldsymbol{h}|\boldsymbol{v}^d) = \underbrace{\sum_{i=1}^{m} a_i v_i^d + \sum_{i=1}^{m}\sum_{j=i+1}^{m} d_{ij}v_i^d v_j^d}_{\text{Constant}} + \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{n}(b_j + c_{ij}v_i^d)h_j}_{\text{Self-interaction}} + \underbrace{\sum_{i=1}^{n}\sum_{j=i+1}^{n} e_{ij}h_i h_j}_{\text{Pairwise-interaction}} \tag{6}$$

2.1.2 Learning Algorithm

The following momentum-based update rule is used in this work:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \underbrace{\eta\frac{\partial}{\partial\boldsymbol{\theta}^{(t)}}D_{KL}(q||p) - \lambda\boldsymbol{\theta}^{(t)} + \nu\Delta\boldsymbol{\theta}^{(t-1)}}_{:=\Delta\boldsymbol{\theta}^{(t)}} \tag{7}$$

An empirical choice of learning parameters is given in table 1.

	η	λ	ν
Range	Empirically determined	$10^{-5} - 10^{-2}$	$0 - 0.9$

Table 1: Learning parameters range

For this problem to be well posed it is also required that the parameters are bounded. In all our example we will take $|H| \leq H_{max}$ and $|J| \leq J_{max}$ with $H_{max} = J_{max} = 1$ unless indicated otherwise. This choice of bound is motivated by the simplicity of the form and the acceptable range of D'Wave processor . The learning procedure is summarized in Algorithm 1

Algorithm 1 Learning algorithm (Distribution Matching)

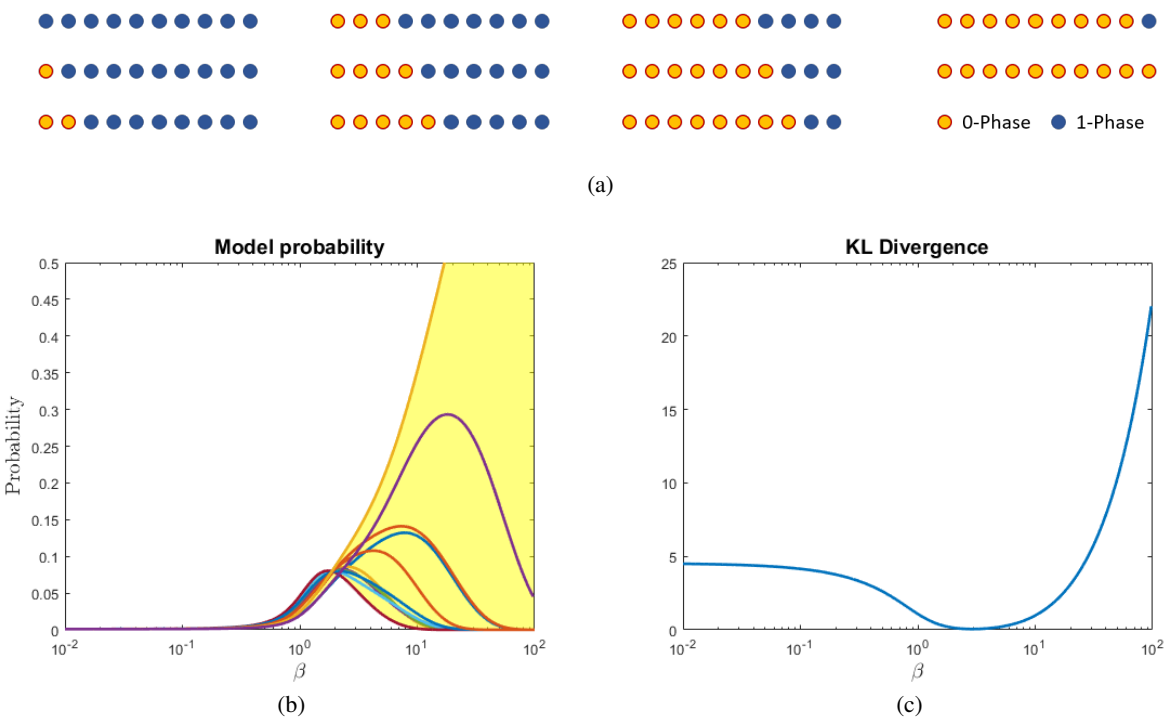
Find embedding of the BM graph and the hidden subgraph on Quantum annealer
Initialize $\boldsymbol{\theta}$
$\Delta\theta = M$ (Large number)
Count=1
while $|\Delta\theta| > \Delta\theta_{min}$ AND Count $<$ Maximum count **do**
    Embed graph with parameter $\boldsymbol{\theta}$ in quantum annealer and sample states
    for $\boldsymbol{v^d} \in \{\boldsymbol{v^1}, ..., \boldsymbol{v^D}\}$ **do**
        if $\boldsymbol{v^d} \notin$ Sampled data **then**
            Embed hidden subgraph with self and pairwise interaction parameters of energy form Eq (6)
            Anneal and sample states
    Evaluate gradient using Eqns (5a-5e)
    Calculate $\boldsymbol{\theta}_{new}$ using Eq(7)
    $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_{new} - \boldsymbol{\theta}$
    $\boldsymbol{\theta} = \boldsymbol{\theta}_{new}$
    Count=Count+1
end

2.2 Numerical Example

In this example, we consider the data presented in Fig3a. In this basic example, we consider 0 and 1 as two phases on a line discretized with 10 points. Each of these points is treated as a visible node which is assigned a phase $(0/1)$. We want to consider only the phases with at-most one boundary between the left phase (assigned value 0) and the right phase (assigned value 1). Thus, there are 11 possibilities and all are considered equally likely. We learned a fully-connected Boltzmann machine with 8 hidden nodes (Parameters presented in Appendix).The BM assigns high and similar probabilities to each phase as desired. The variation of probability of each state with respect to the β is presented in Fig 3b. It is observed that the result does not have the expected distribution for very high or very low value of β. This is due to the finite annealing temperature of D–Wave processor that decides the training β of the BM. It can be observed in Fig 3c that in this case the range of appropriate β is $[1.5, 3]$. In the next section, we will discuss the consequence of finite annealing temperature on the learning process.

2.3 Effect of Annealing temperature

To further understand the difference between a BM developed for $\beta \to \infty$ and for a finite β we look at the example presented in Fig 4. In this example two XOR gates are presented i.e the states $\{[0,0,0],[0,1,1],[1,0,1],[1,1,0]\}$ are chosen as visible data with each having equal probability. Here, the first two states are regarded as the input node (denoted by $\boldsymbol{v}^I$) and the third one as the output node (denoted by $\boldsymbol{v}^O$). Additionally, there is one more hidden node in



(a) (b) (c)

Figure 3: (a)Data points representing phases with at most one boundary. (b)The model probability of each data point with yellow region defining the spread between the minimum and the maximum probabilities (c) KL Divergence of the model at different β

the network. The ground states (energy minimising states of the Ising model) of the left BM (Fig 4a) are designed to represent the visible data. We will refer to this BM as ground state BM. The right BM (Fig 4b) is learned using the method outlined in the previous section. It is evident from the probability distribution of states for ground state BM (Fig 4c) that all states are equally probable and this probability increases with the increase in β. On the other hand, the probability distribution of some states for trained BM start to decrease for high BM. In fact, it can be shown that (see Appendix B for the proof of following statements):

Statement 1: As β increases the ground state probability increases. It strictly increases in the case when not all states are ground states.

Statement 2: For every model parameter $\boldsymbol{\theta}$, assuming the absolute value of energy of every state is bounded with E_{max} and not all states are ground states, there exists a β_c such that for all $\beta \geq \beta_c$ the probability of each excited state decreases with β. Here , we refer to excited state as any admissible state which is not a ground state

As a result, the probability of each of the visible data increases in this case and at $\beta \to \infty$, it exactly matches the data. And it can be seen in Fig4e that the KL Divergence decreases to zero in this limit. It should be noted that this behavior is not true in general. In fact, all ground states are equally probable in this limit, therefore arbitrary probability distribution of data cannot be exactly matched with a ground state BM. On the other hand, in the trained BM, the KL divergence (Fig 4f) is minimum for a finite value of β and it diverges as $\beta \to \infty$. This BM performs best near this minima as evident from the probability distribution. In fact, it can be shown that if $D_{KL} \to 0$ for a finite value of β i.e.

when $p \sim q$, we have

$$\frac{d^2 D_{KL}}{d\beta^2} \rightarrow \sum_{\boldsymbol{v} \in \{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^D\}} p(\boldsymbol{v}) \left(-\operatorname{Var}(E|\boldsymbol{v}) + \operatorname{Var}(E)\right) \qquad \text{(see Appendix A)}$$

$$= \operatorname{Var}(\mathbb{E}_{\boldsymbol{v}}(E|\boldsymbol{v})) \qquad \text{(Law of total variance)}$$

This implies that $\frac{d^2 D_{KL}}{d\beta^2} \geq 0$ in the above case with equality only in the extremely rare case of same expected value of energy for each visible data. Therefore, a well-trained BM (i.e. $q \sim p$ for the training β) often shows a local minima at the training β.

A quantum annealer samples Boltzmann states with an unknown β. In previous studies, it has been seen that the value of β is high but also dependent on the size and scale of the problem (eg. number of qubits used [9]). The next feature that we will investigate in this study is the the consequence of training the Boltzmann machine at an unknown β. We first look at the D_{KL} of the trained BM for the data corresponding to the AND-gate (Fig5a). Subsequent to training the Boltzmann machine, solutions are sampled and the D_{KL} for each sample is measured. It is observed that the distribution of D_{KL} of sample lies in the neighborhood of the minimum D_{KL} of the model with respect to β. This supports the intuitive argument that a trained BM performs best in the neighborhood of the sampling or training β. However, other than these experimental results, we are not aware of any theoretical guarantees for such a phenomenon. Next, we look at a similar example using OR-gate data. Three BM's with 2,5 and 10 hidden nodes are considered. It is observed that the D_{KL}(Fig5b) is minimum at different values of β which suggests that the training of these models is carried out at different β.

Motivated by this finding, we simulated complete graphs of different sizes with random parameters. We estimated the β values which corresponded to the closest match with analytical distribution. Specifically, we estimated

$$\beta^* = \operatorname{argmin}_\beta \frac{1}{2} \sqrt{\sum_i (\sqrt{p_i^s} - \sqrt{p_i^a(\beta)})^2}$$

where p^s is the sample probability, $p^a(\beta)$ is the analytical probability (dependent on β) and the cost function corresponds the Hellinger distance between two distributions. The training β calculated from this analysis is presented in Fig 6. It shows that the local solver ($C4$, an emulator) trains the BM at an almost constant value of $\beta = 3$ while the training β of remote solver (on the physical machine) decreases with graph size. It should be noted that such an analysis is not possible for very large graph as analytical calculation of probabilities take 2^N computations where N is the number of nodes.

2.4 Initial parameter dependence on convergence

The energy surface of Ising model is non-linear and hence gradient-based methods are prone to getting stuck in local minima. This is evident in the Fig 7 where the D_{KL} at each learning step is presented for different initial conditions. It

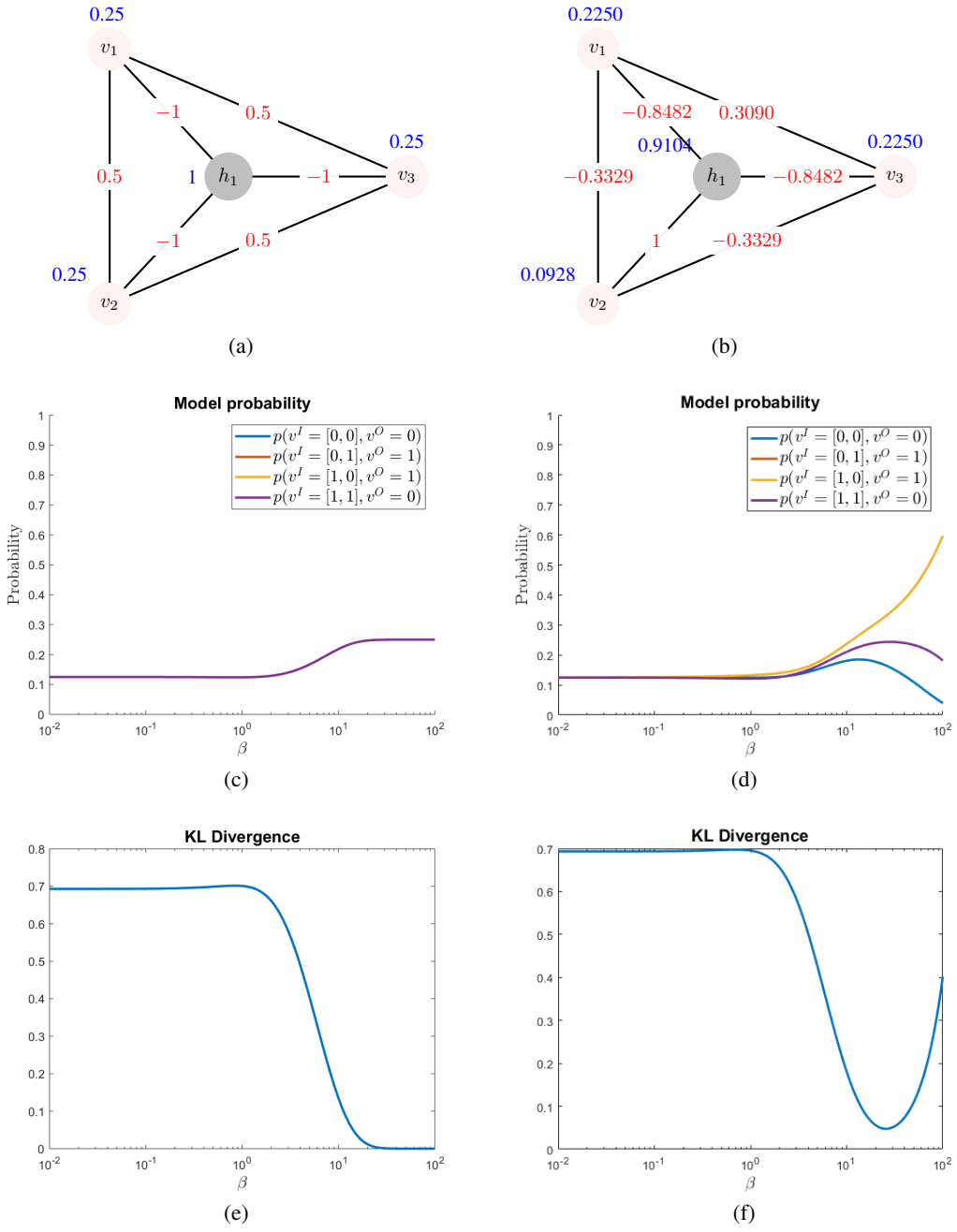
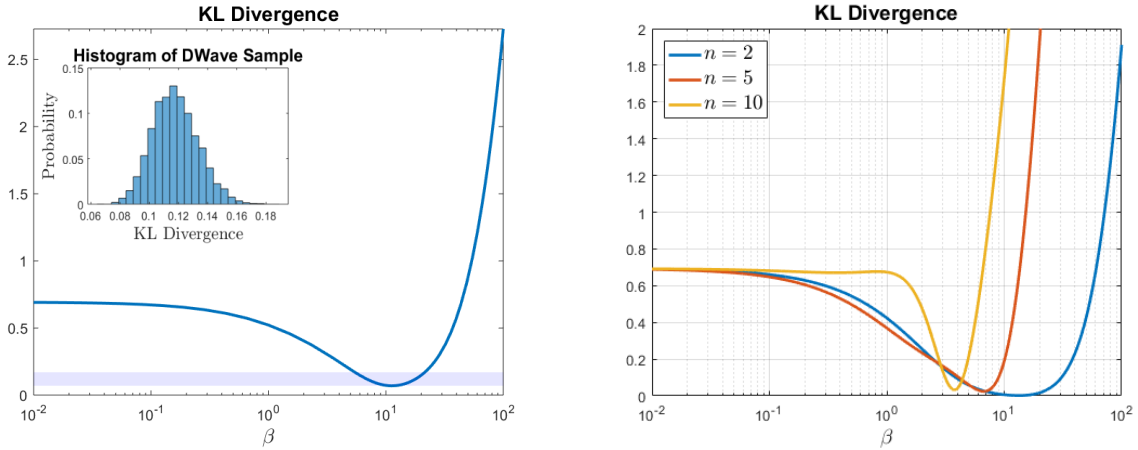


Figure 4: (a) A ground state BM for XOR-gate data, (b) A trained BM for XOR-gate data (c) Model probability distribution variation with β for the ground state BM, (d) Model probability distribution variation with β for the trained BM, (e) KL-Divergence variation with β for the ground state BM and (f) KL-Divergence variation with β for the trained BM



(a) (b)

Figure 5: (a)The KL-divergence of trained BM for data corresponding to OR-gate with 1 hidden node, (b) The KL-divergence of trained BM's for data corresponding to OR-gate with 2, 5 and 10 hidden nodes

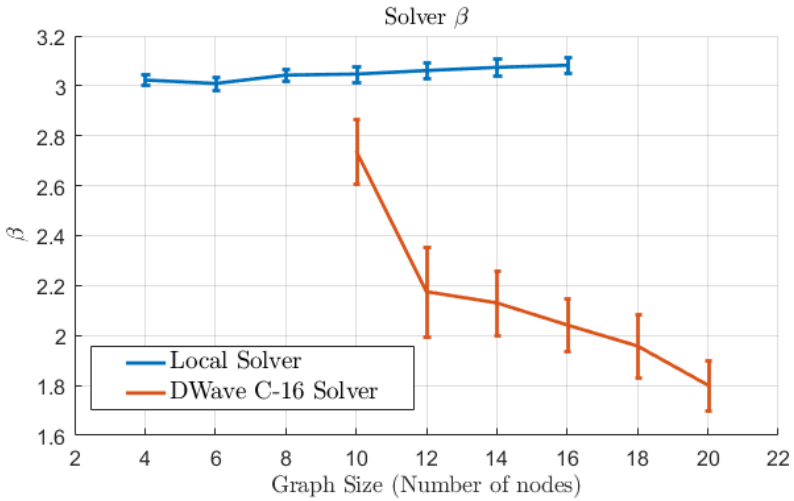


Figure 6: DWave training β for different graph sizes (Complete graph)

can observed that some solution go as low as 10^{-5} while others do not. Following learning parameters are used in the simulation: Learning rate = 10^{-1}, Learning Steps (η) = 10^3, Weight decay rate (λ) = 10^{-5}, Momentum rate (ν) = 0.6.

3 Function Approximation

So far, we have talked about using BM's for imitating the distribution of a given data. It can also be used as a function. In simple terms, we want to encode the tuple $(x, f(x))$ as the state of the visible nodes. Here, $f(x)$ denotes the function that is to be approximated. In the context of the BM, the visible data can be thought of as the binary expansion of x and $f(x)$. It is desired that if the visible nodes corresponding to the input variable take a particular value x', then the visible nodes corresponding to the output variable take a value of $f(x')$ with high probability. Thus a BM for approximating function $f(x)$ has high conditional probability, $p(f(x)|x)$. This conditional probability can be estimated using Eq(8)

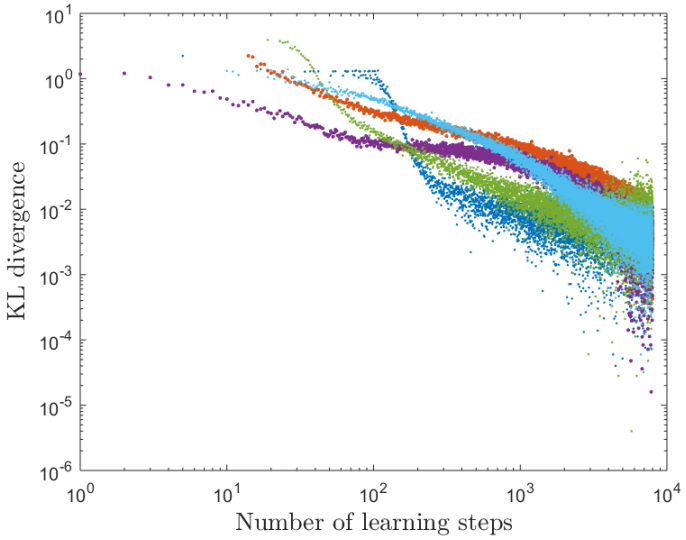
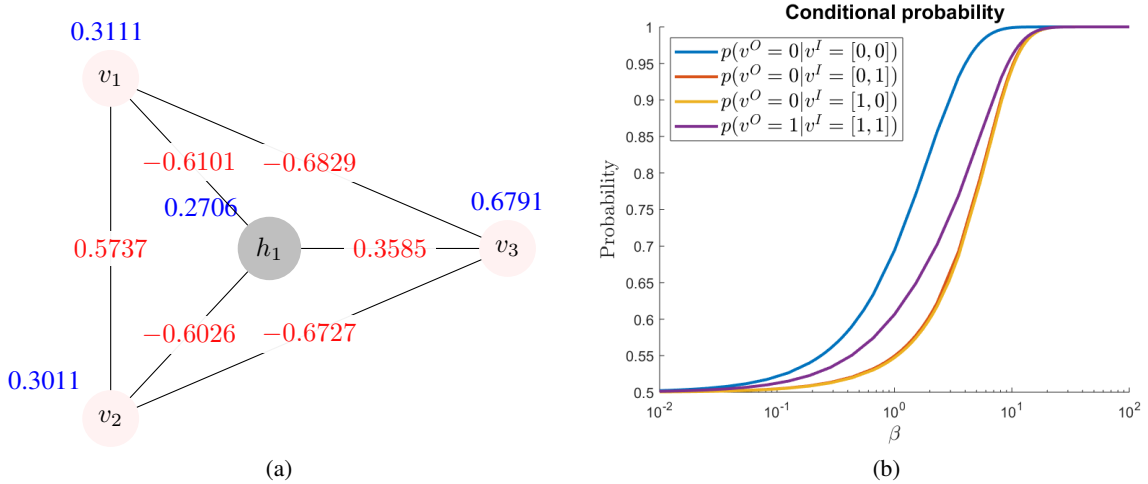


Figure 7: $D_{KL}(q||p)$ at each learning step for 5 randomly initiated parameters for training data set corresponding to AND-gate

where the visible data $\boldsymbol{v}$ is written as concatenation of input data, $\boldsymbol{v^I}$ and output data, $\boldsymbol{v^O}$. An example of a BM for approximating AND gate is presented in Fig.8. It is observed that $p(\boldsymbol{v^O}|\boldsymbol{v^I}) \to 1$ as $\beta \to \infty$ (Low temp solution) and $p(\boldsymbol{v^O}|\boldsymbol{v^I}) \to 0.5$ as $\beta \to 0$ (High temp solution).

$$p(\boldsymbol{v^O}|\boldsymbol{v^I}) = \frac{p([\boldsymbol{v^I}, \boldsymbol{v^O}])}{\sum_{\boldsymbol{v^O}} p([\boldsymbol{v^I}, \boldsymbol{v^O}])} \tag{8}$$



(a) (b)

Figure 8: (a) Boltzmnann machine to approximate AND Gate with 3 visible nodes and 1 hidden node. The numerical values in blue represent the self-interaction parameters and in red represent the node-node interaction parameters. (b) The conditional probability of the BM evaluated at different values of β

The energy defined in Eq (1) can be slightly modified to distinguish the interaction terms of the input and output visible nodes as follows:

$$E(\boldsymbol{v}^I, \boldsymbol{v}^O, \boldsymbol{h}) = \sum_{i=1}^{m_I} \alpha_i^{v^I} v_i^I + \sum_{i=1}^{m_O} \alpha_i^{v^O} v_i^O + \sum_{j=1}^{n} \alpha_j^h h_j + \sum_{i=1}^{m_I}\sum_{j=1}^{n} \gamma_{ij}^{v^I h} v_i h_j + \sum_{i=1}^{m_O}\sum_{j=1}^{n} \gamma_{ij}^{v^O h} v_i^O h_j$$
$$+ \sum_{i=1}^{m_I}\sum_{j=i+1}^{m_O} \gamma_{ij}^{v^I v^O} v_i^I v_j^O + \sum_{i=1}^{m_I}\sum_{j=i+1}^{m_I} \gamma_{ij}^{v^I v^I} v_i^I v_j^I + \sum_{i=1}^{m_O}\sum_{j=i+1}^{m_O} \gamma_{ij}^{v^O v^O} v_i^O v_j^O + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \gamma_{ij}^{hh} h_i h_j \tag{9}$$

Here the superscripts v^I and v^O are introduced to signify that the respective parameter affects input and the output node, respectively.

3.1 Learning Method for function approximators

The cost function in this case will be taken as the Negative Conditional Log-likelihood $\mathcal{L}$ defined as,

$$\mathcal{L}(\theta) = -\sum_{[\boldsymbol{v}^I, \boldsymbol{v}^O] \in \{\boldsymbol{v}^1, ..., \boldsymbol{v}^D\}} \ln p(\boldsymbol{v^O}|\boldsymbol{v^I}, \boldsymbol{\theta})$$

where, as before, the set $\{\boldsymbol{v}^1, ..., \boldsymbol{v}^D\}$ represents the D data points. The gradient of the cost function is estimated as:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \boldsymbol{\theta}} = \beta \sum_{[\widetilde{\boldsymbol{v}}^I, \widetilde{\boldsymbol{v}}^O] \in \{\boldsymbol{v}^1, ..., \boldsymbol{v}^D\}} \left(\sum_{\boldsymbol{v^O}, \boldsymbol{h}} p(\boldsymbol{v^O}, \boldsymbol{h}|\widetilde{\boldsymbol{v}}^I) \frac{\partial E(\widetilde{\boldsymbol{v}}^I, \boldsymbol{v^O}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} - \sum_{\boldsymbol{h'}} p(\boldsymbol{h'}|\widetilde{\boldsymbol{v}}^I, \widetilde{\boldsymbol{v}}^O) \frac{\partial E(\widetilde{\boldsymbol{v}}^I, \widetilde{\boldsymbol{v}}^O, \boldsymbol{h'})}{\partial \boldsymbol{\theta}} \right) \tag{10}$$

As before we use the momentum-based gradient scheme defined in Eq(7) to learn the BM. In this case the sampling will be done on (1) the subgraph of hidden nodes (2) the subgraph of hidden and output nodes. The parameters of these graphs can be estimated in the same way as in the case of distribution matching.

Algorithm 2 Learning algorithm (Function approximator)

Find embedding of the BM hidden subgraph and the hidden+output subgraph on Quantum annealer
Initialize $\boldsymbol{\theta}$
$\Delta\theta = M$ (Large number)
Count=1
while $|\Delta\theta| > \Delta\theta_{min}$ AND Count $<$ Maximum count **do**
    Embed graph with parameter $\boldsymbol{\theta}$ in quantum annealer and sample states
    for $\boldsymbol{v^d} \in \{\boldsymbol{v^1}, ..., \boldsymbol{v^D}\}$ **do**
        Embed hidden subgraph with modified parameters
        Anneal, sample states and approximate $p(\boldsymbol{h'}|\widetilde{\boldsymbol{v}}^I, \widetilde{\boldsymbol{v}}^O)$.
        Embed hidden+output subgraph with modified parameters
        Anneal, sample states and approximate $p(\boldsymbol{v^O}, \boldsymbol{h}|\widetilde{\boldsymbol{v}}^I)$.
    Evaluate gradient using Eq (10)
    Calculate $\boldsymbol{\theta}_{new}$ using Eq(7)
    $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_{new} - \boldsymbol{\theta}$
    $\boldsymbol{\theta} = \boldsymbol{\theta}_{new}$
    Count=Count+1
end

3.2 Example of an Adder Gate

As an example we present a BM for 2-bit adder shown in Table 2. A comparison of BM learned as a Function approximator and a Distribution is presented in Fig.9 with values of the trained parameter presented in Appendix D. It is observed that the KL Divergence of distribution BM (Fig 9d) has a minimum at around $\beta \sim 5$ which asserts that the training β for this BM is close to 5. As expected, the function approximator does not show any minimizing effect in the KL divergence. However, the conditional probabilities in the case of Function approximator (Fig 9a) are higher in comparison to distribution BM (Fig 9b). At β 5, the minimum conditional probability in the case of Function approximator is close to 0.3 while that of distribution BM is 0.15 This is the desired outcome from this procedure.

4 Conclusion

We have developed and analysed two training methodologies for training Boltzmann machine. A summary of these method is as follows:

1. Generative training for distribution matching is carried out by minimizing KL divergence. We present the example of learning a distribution of 2-phase 1D material with at-most one boundary.

Input Nodes				Output Nodes		
A_1/v_1^I	A_0/v_2^I	B_1/v_3^I	B_0/v_4^I	C_2/v_1^O	C_1/v_2^O	C_0/v_3^O
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Table 2: Truth table for a 2-bit adder circuit

2. Discriminative training for function learning is carried out by minimizing conditional log-likelihood. We present the example of a 2-bit adder circuit

Gradient based methods are used to minimize these costs. Quantum annealer is employed for direct sampling for the approximation of the gradients. In both methods the number of sampling steps depend linearly on the number of data points. We also analysed the effect of training BM at a finite annealing temperature. The resulting trained BM is not a ground state BM. Moreover, the annealing temperature is shown to depend on the graph-size (with possibly other factors). As a next step, we will analyse the result of combined Generative-Discriminative training of classifiers, focusing on parameter estimation in differential equations [10].

References

[1] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[2] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[3] Navdeep Jaitly and Geoffrey Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5887. IEEE, 2011.

[4] SM Ali Eslami, Nicolas Heess, Christopher KI Williams, and John Winn. The shape boltzmann machine: a strong model of object shape. *International Journal of Computer Vision*, 107(2):155–176, 2014.

[5] Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In *Iberoamerican Congress on Pattern Recognition*, pages 14–36. Springer, 2012.

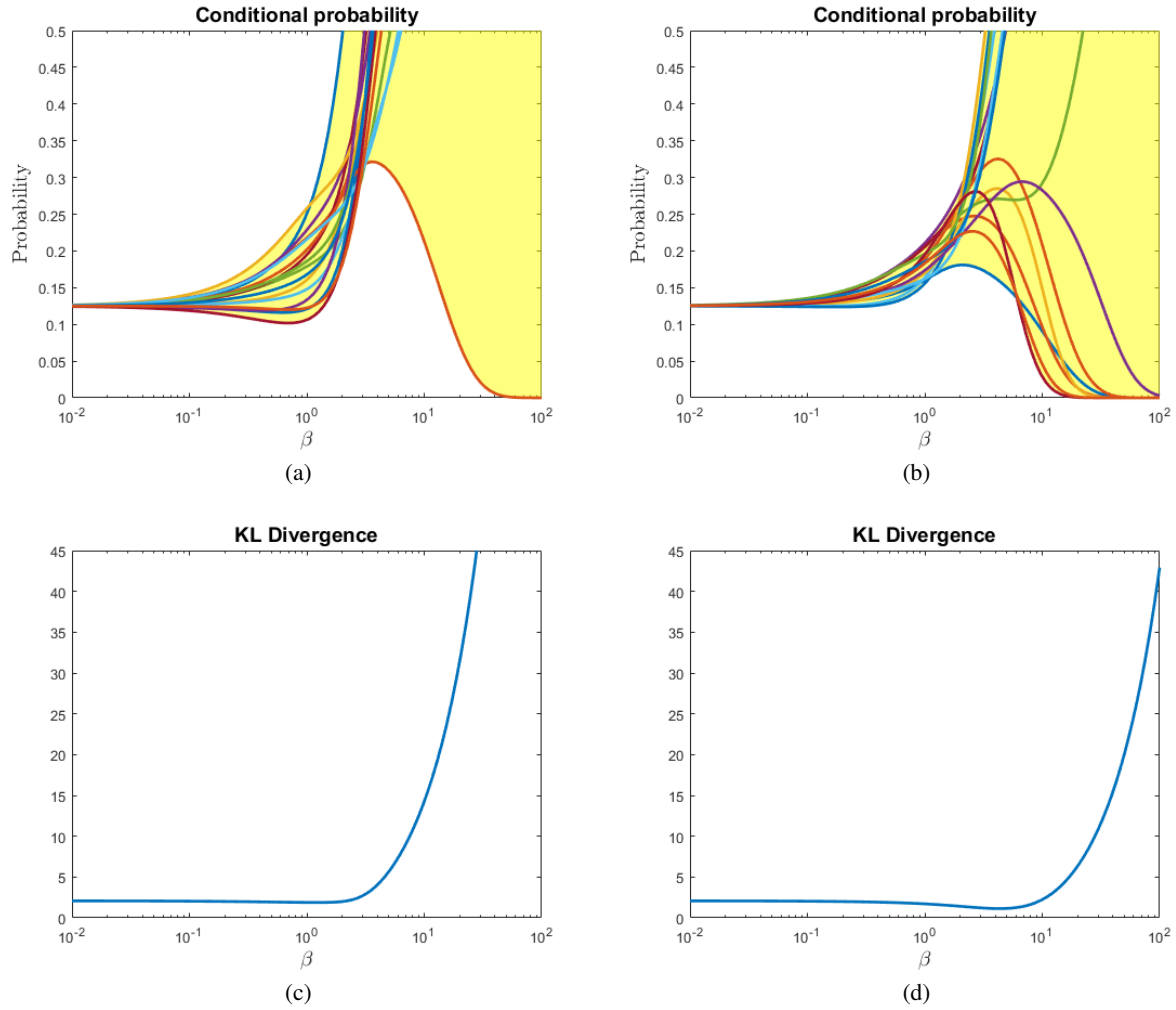


Figure 9: Boltzmann machine representing a 2-bit adder circuit: (a) Conditional Probability of a BM trained as a function approximator (b) Conditional Probability of a BM trained as a distribution (c) KL Divergence of a BM trained as a function approximator (d) KL Divergence of a BM trained as a distribution

[6] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum deep learning. *arXiv preprint arXiv:1412.3489*, 2014.

[7] Amir Khoshaman, Walter Vinci, Brandon Denis, Evgeny Andriyash, and Mohammad H Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1):014001, 2018.

[8] Walter Vinci, Lorenzo Buffoni, Hossein Sadeghi, Amir Khoshaman, Evgeny Andriyash, and Mohammad H Amin. A path towards quantum advantage in training deep generative models with quantum annealers. *arXiv preprint arXiv:1912.02119*, 2019.

[9] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

[10] Siddhartha Srivastava and Veera Sundararaghavan. Box algorithm for the solution of differential equations on a

quantum annealer. *Physical Review A*, 99(5):052355, 2019.

A Estimation of gradients

A.1 Gradient of Log-likelihood for a single data

$$
\begin{aligned}
\frac{\partial \ln p_{\boldsymbol{\theta}}(\boldsymbol{v})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})} \right) - \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{v}',\boldsymbol{h}'} e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')} \right) \\
&= -\beta \sum_{\boldsymbol{h}} \frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}'} e^{-\beta E(\boldsymbol{v},\boldsymbol{h}')}} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \beta \sum_{\boldsymbol{v}',\boldsymbol{h}'} \frac{e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}}{\sum_{\boldsymbol{v}'',\boldsymbol{h}''} e^{-\beta E(\boldsymbol{v}'',\boldsymbol{h}'')}} \frac{\partial E(\boldsymbol{v}',\boldsymbol{h}')}{\partial \boldsymbol{\theta}} \\
&= -\beta \sum_{\boldsymbol{h}} \frac{\frac{1}{Z} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\frac{1}{Z} \sum_{\boldsymbol{h}'} e^{-\beta E(\boldsymbol{v},\boldsymbol{h}')}} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \beta \sum_{\boldsymbol{v}',\boldsymbol{h}'} \frac{e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}}{Z} \frac{\partial E(\boldsymbol{v}',\boldsymbol{h}')}{\partial \boldsymbol{\theta}} \\
&= -\beta \sum_{\boldsymbol{h}} \frac{p(\boldsymbol{v},\boldsymbol{h})}{p(\boldsymbol{v})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \beta \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') \frac{\partial E(\boldsymbol{v}',\boldsymbol{h}')}{\partial \boldsymbol{\theta}} \\
&= -\beta \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \beta \sum_{\boldsymbol{v}',\boldsymbol{h}'} p(\boldsymbol{v}',\boldsymbol{h}') \frac{\partial E(\boldsymbol{v}',\boldsymbol{h}')}{\partial \boldsymbol{\theta}}
\end{aligned}
\tag{11}
$$

A.2 Derivative of D_{KL} w.r.t. β

$$
\begin{aligned}
\frac{dD_{KL}}{d\beta} &= -\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \frac{d}{d\beta} \ln \frac{p(\boldsymbol{v})}{q(\boldsymbol{v})} = -\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} \frac{q(\boldsymbol{v})}{p(\boldsymbol{v})} \frac{d}{d\beta} \frac{\sum_{\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}',\boldsymbol{h}'} e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}} \\
&= -\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} \frac{q(\boldsymbol{v})}{p(\boldsymbol{v})} \left(\frac{-\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}',\boldsymbol{h}'} e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}} + \frac{\left(\sum_{\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}\right) \left(\sum_{\boldsymbol{v}',\boldsymbol{h}'} E(\boldsymbol{v}',\boldsymbol{h}') e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}\right)}{\left(\sum_{\boldsymbol{v}'',\boldsymbol{h}''} e^{-\beta E(\boldsymbol{v}'',\boldsymbol{h}'')}\right)^2} \right) \\
&= \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} \frac{q(\boldsymbol{v})}{p(\boldsymbol{v})} \left(\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) p(\boldsymbol{v},\boldsymbol{h}) - p(\boldsymbol{v}) \sum_{\boldsymbol{v}',\boldsymbol{h}'} E(\boldsymbol{v}',\boldsymbol{h}') p(\boldsymbol{v}',\boldsymbol{h}') \right) \\
&= \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \left(\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) p(\boldsymbol{h}|\boldsymbol{v}) - \sum_{\boldsymbol{v}',\boldsymbol{h}'} E(\boldsymbol{v}',\boldsymbol{h}') p(\boldsymbol{v}',\boldsymbol{h}') \right) \\
&= -\mathbb{E}_{\boldsymbol{v},\boldsymbol{h}}(E) + \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) p(\boldsymbol{h}|\boldsymbol{v})
\end{aligned}
$$

$$
\begin{aligned}
\frac{d^2 D_{KL}}{d\beta^2} &= \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \left(\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) \frac{d}{d\beta} p(\boldsymbol{h}|\boldsymbol{v}) - \sum_{\boldsymbol{v}',\boldsymbol{h}'} E(\boldsymbol{v}',\boldsymbol{h}') \frac{d}{d\beta} p(\boldsymbol{v}',\boldsymbol{h}') \right) \\
&= \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}} q(\boldsymbol{v}) \left(\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h}) \underbrace{\frac{d}{d\beta} \frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}''} e^{-\beta E(\boldsymbol{v},\boldsymbol{h}'')}}}_{\text{Term } I} - \sum_{\boldsymbol{v}',\boldsymbol{h}'} E(\boldsymbol{v}',\boldsymbol{h}') \underbrace{\frac{d}{d\beta} \frac{e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}}{\sum_{\boldsymbol{v}'',\boldsymbol{h}''} e^{-\beta E(\boldsymbol{v}'',\boldsymbol{h}'')}}}_{\text{Term } II} \right)
\end{aligned}
$$

Here, we evaluate term I as:

$$\frac{d}{d\beta}\frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}''}e^{-\beta E(\boldsymbol{v},\boldsymbol{h}'')}} = -\frac{E(\boldsymbol{v},\boldsymbol{h})e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}''}e^{-\beta E(\boldsymbol{v},\boldsymbol{h}'')}} + \frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}\sum_{\boldsymbol{h}'}E(\boldsymbol{v},\boldsymbol{h}')e^{-\beta E(\boldsymbol{v},\boldsymbol{h}')}}{\left(\sum_{\boldsymbol{h}''}e^{-\beta E(\boldsymbol{v},\boldsymbol{h}'')}\right)^2}$$
$$= -E(\boldsymbol{v},\boldsymbol{h})p(\boldsymbol{h}|\boldsymbol{v}) + p(\boldsymbol{h}|\boldsymbol{v})\sum_{\boldsymbol{h}'}E(\boldsymbol{v},\boldsymbol{h}')p(\boldsymbol{h}'|\boldsymbol{v})$$

And term II as:

$$\frac{d}{d\beta}\frac{e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}}{\sum_{\boldsymbol{v}'',\boldsymbol{h}''}e^{-\beta E(\boldsymbol{v}'',\boldsymbol{h}'')}} = -\frac{E(\boldsymbol{v}',\boldsymbol{h}')e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}}{Z} + \frac{e^{-\beta E(\boldsymbol{v}',\boldsymbol{h}')}\sum_{\boldsymbol{v}'',\boldsymbol{h}''}E(\boldsymbol{v}'',\boldsymbol{h}'')e^{-\beta E(\boldsymbol{v}'',\boldsymbol{h}'')}}{Z^2}$$
$$= -E(\boldsymbol{v}',\boldsymbol{h}')p(\boldsymbol{v}',\boldsymbol{h}') + p(\boldsymbol{v}',\boldsymbol{h}')\sum_{\boldsymbol{v}'',\boldsymbol{h}''}E(\boldsymbol{v}'',\boldsymbol{h}'')p(\boldsymbol{v}'',\boldsymbol{h}'')$$

Combining the two terms:

$$\frac{d^2D_{KL}}{d\beta^2} = \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}}q(\boldsymbol{v})\left(\sum_{\boldsymbol{h}}-E^2(\boldsymbol{v},\boldsymbol{h})p(\boldsymbol{h}|\boldsymbol{v}) + \left(\sum_{\boldsymbol{h}'}E(\boldsymbol{v},\boldsymbol{h}')p(\boldsymbol{h}'|\boldsymbol{v})\right)^2\right.$$
$$\left. + \sum_{\boldsymbol{v}',\boldsymbol{h}'}E^2(\boldsymbol{v}',\boldsymbol{h}')p(\boldsymbol{v}',\boldsymbol{h}') - \left(\sum_{\boldsymbol{v}'',\boldsymbol{h}''}E(\boldsymbol{v}'',\boldsymbol{h}'')p(\boldsymbol{v}'',\boldsymbol{h}'')\right)^2\right)$$
$$= \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^D\}}q(\boldsymbol{v})\left(-\operatorname{Var}(E|\boldsymbol{v}) + \operatorname{Var}(E)\right)$$

B Proof of Propositions

Proposition 1: As β increases the ground state probability increases. It strictly increases in the case when not all states are ground states.

Proof: Let $\boldsymbol{S_g}$ be a ground state and $E_{min} = E(S_G)$. It suffices to show that:

$$\frac{d\ln p(\boldsymbol{S_g})}{d\beta} \geq 0$$

We estimate the derivative as follows:

$$\frac{d\ln p(\boldsymbol{S_g})}{d\beta} = \frac{d\ln\frac{e^{-\beta E_{min}}}{Z}}{d\beta} = \frac{d\left(-\beta E_{min} - \ln Z\right)}{d\beta}$$
$$= -E_{min} + \sum_{\boldsymbol{S}}E(\boldsymbol{S})\frac{e^{-\beta E(\boldsymbol{S})}}{Z}$$
$$= -E_{min} + \sum_{\boldsymbol{S}}E(\boldsymbol{S})p(\boldsymbol{S})$$
$$= -E_{min} + \mathbb{E}(E) \geq 0$$

Here $\mathbb{E}(E)$ denotes the expected energy of the system. It can be easily seen that the inequality is strict if and only if there not all states have the same energy. ■

Proposition 2: For every model parameter $\boldsymbol{\theta}$, assuming the absolute value of energy of every state is bounded with E_{max}, then given any two states $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ with $E(\boldsymbol{S}_1) < E(\boldsymbol{S}_2)$ there exists $\beta_{\boldsymbol{S}_1} > \beta_{\boldsymbol{S}_2}$ such that for all $\beta > \beta_{\boldsymbol{S}_i}$, the probability of state $\boldsymbol{S}_i$, denoted by $p(\boldsymbol{S}_i)$ decreases.

Proof: Let $\boldsymbol{S}$ be any state. Following the calculation done in proof of proposition 1, we get:

$$\frac{d\ln p(\boldsymbol{S})}{d\beta} = -E(\boldsymbol{S}) + \mathbb{E}(E)$$

Let $\boldsymbol{S_g}$ be a ground state with $E_{min} = E(S_G)$ and $\boldsymbol{S}_{e_1}$ be a first excited state (State with energy strictly greater than the ground state but less than every excited state) with $E_1 = E(\boldsymbol{S}_{e_1})$. This means for any excited state, $\boldsymbol{S}_{exc}$, we have:

$$\frac{d\ln p(\boldsymbol{S}_{exc})}{d\beta} = -E(\boldsymbol{S}_{exc}) + \mathbb{E}(E) \leq -E(\boldsymbol{S}_1) + \mathbb{E}(E) = \frac{d\ln p(\boldsymbol{S}_1)}{d\beta}$$

We will first show that $\frac{d\ln p(\boldsymbol{S}_1)}{d\beta} \leq 0$ for all $\beta \geq \beta_c$ for some $\beta_c > 0$.

$$\begin{aligned}
\frac{d\ln p(\boldsymbol{S}_{e_1})}{d\beta} &= -E_1 + \mathbb{E}(E) = -E_1 + \sum_{\boldsymbol{S}\in\mathcal{S}} E(\boldsymbol{S})p(\boldsymbol{S}) \\
&= -E_1 + \sum_{\boldsymbol{S}_g\in\mathcal{G}} E(\boldsymbol{S}_g)p(\boldsymbol{S}_g) + \sum_{\boldsymbol{S}\in\mathcal{S}-\mathcal{G}} E(\boldsymbol{S})p(\boldsymbol{S}) \\
&\leq -E_1 + E_{min}\sum_{\boldsymbol{S}_g\in\mathcal{G}} p(\boldsymbol{S}_g) + E_{max}\sum_{\boldsymbol{S}_g\in\mathcal{G}} p(\boldsymbol{S}_g)
\end{aligned}$$

Denoting with $P \equiv P(\boldsymbol{\theta}, \beta)$ the total probability of ground states, estimated as $P = \sum_{\boldsymbol{S}_g\in\mathcal{G}} p(\boldsymbol{S}_g) = |\mathcal{G}|\frac{e^{-\beta E_{min}}}{Z}$, where $|\mathcal{G}|$ denotes the multiplicity of ground states, we get that

$$\begin{aligned}
\frac{d\ln p(\boldsymbol{S}_1)}{d\beta} &\leq -E_1 + E_{min}P + E_{max}(1-P) \\
&= (E_{max} - E_1) - P(E_{max} - E_{min})
\end{aligned}$$

We want to choose β_c such that $P(\boldsymbol{\theta}, \beta_c) \geq \frac{E_{max}-E_1}{E_{max}-E_{min}}$. Noting that not all states have energy E_{min}, we get, $(E_{max} - E_1) < P(E_{max} - E_{min})$. From Proposition 1, we know that P is strictly increasing with $\sup_\beta P = 1$. We get that there is some β_c such that $P(\boldsymbol{\theta}, \beta_c)$ has the desired value. Next, to prove the claim we note that:

$$\frac{d\ln p(\boldsymbol{S}_1)}{d\beta} = -E(\boldsymbol{S}_1) + \mathbb{E}(E) > -E(\boldsymbol{S}_2) + \mathbb{E}(E) = \frac{d\ln p(\boldsymbol{S}_2)}{d\beta}$$

This means $p(\boldsymbol{S}_2)$ is decreasing whenever $p(\boldsymbol{S}_1)$ is decreasing. ■

Corollary 3: For every model parameter $\boldsymbol{\theta}$, assuming the absolute value of energy of every state is bounded with E_{max} and not all states are ground states, there exists a β_c such that for all $\beta \geq \beta_c$ the probability of each excited state decreases with β. ■

C Change of basis

It is a common practice to define the Ising states as either $\{0, 1\}$ or $\{-1, +1\}$ states. We choose the former format while D-Wave works with the latter one. In this section we will present the details about conversion between these formats. For the purpose of discussion, we will denote the $\{0, 1\}$ Ising model with lower case variables, $\{\boldsymbol{s}, \boldsymbol{\theta}^I, \boldsymbol{\theta}^{II}\}$ and the $\{-1, 1\}$ Ising with upper case variables, $\{\boldsymbol{S}, \boldsymbol{\Theta}^I, \boldsymbol{\Theta}^{II}\}$. Here, $\boldsymbol{s}/\boldsymbol{S}$ denotes the state of a graph, $\boldsymbol{\theta}^I/\boldsymbol{\Theta}^I$ denotes the self-interaction parameter and $\boldsymbol{\theta}^{II}/\boldsymbol{\Theta}^{II}$ denote the Pairwise interaction parameter.The states of the system can be interchanged using the following equation:

$$S = 2s - 1 \tag{12}$$

In this transformation of variables, the equivalent transformation in energy parameters is given by:

$$\begin{aligned} \Theta^{II}_{kl} &= \frac{1}{4}\theta^{II}_{kl} \\ \Theta^{I}_{k} &= \frac{1}{2}\theta^{I}_{k} + \frac{1}{4}\sum_{l \in \mathcal{N}(k)} \theta^{II}_{kl} \end{aligned} \tag{13}$$

Here we denote the neighbouring vertices of node k as $\mathcal{N}(k)$. This transformation shifts the energy of each state with a constant value and hence leaves the Boltzmann probability unchanged as required.

D Parameters for Boltzmann machine

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
v_1	7009	-9446	-4488	-0392	-0918	1762	1967	2256	2911	1935	-3519	2152	5159	-0329	1238	-0146	2742	4321
v_2	0	6927	-9555	-3119	-1885	0984	1944	2216	2024	0946	-4469	1052	7119	2408	1518	0369	3999	2524
v_3	0	0	6546	-7917	-2526	-0039	-1300	0823	0540	1529	-4151	3168	5857	-0242	2070	2206	5279	3935
v_4	0	0	0	2176	-6351	-2532	-1320	-0058	0887	-0831	-2895	3605	7262	0516	1932	0069	6981	5151
v_5	0	0	0	0	4953	-8388	-4282	-2942	0295	-1597	-1135	3144	4633	1975	0508	0668	4561	7515
v_6	0	0	0	0	0	3723	-9747	-4126	-0873	-1271	0162	5440	3709	1110	-2177	-0613	3087	6950
v_7	0	0	0	0	0	0	7447	-9664	-3826	-1666	1104	5864	1671	1570	-4787	-0780	2767	3082
v_8	0	0	0	0	0	0	0	8708	-9827	-6112	-0021	5564	1466	1168	-5490	0846	0739	2666
v_9	0	0	0	0	0	0	0	0	-0008	-1.0000	0555	5624	-0300	2914	-5560	-1283	0778	4185
v_{10}	0	0	0	0	0	0	0	0	0	1141	-0101	3150	-3195	1092	-4924	0112	-1851	-0366
h_1	0	0	0	0	0	0	0	0	0	0	7293	2351	1420	0785	-0848	1482	0751	1185
h_2	0	0	0	0	0	0	0	0	0	0	0	-6076	-7248	-1570	1920	1674	-3135	-5219
h_3	0	0	0	0	0	0	0	0	0	0	0	0	-8765	-2469	-0265	0868	-6644	-8203
h_4	0	0	0	0	0	0	0	0	0	0	0	0	0	1248	2644	-1203	1251	-2923
h_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6754	2002	1893	1262
h_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5498	1390	1628
h_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-7724	-8370
h_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-5756

Table 3: Parameters for Boltzmann machine (10^{-4} units) representing single jump from 0 to 1 trained at $\beta \sim 2$. The diagonal terms represent the field strength. Due to symmetric relation of interaction the lower diagonal is taken to be zero.

	v_1^I	v_2^I	v_3^I	v_4^I	v_5^O	v_6^O	v_7^O	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_10
v_1^I	7388	1673	0403	1804	-10000	-1099	3941	-4622	-4148	-4683	-2312	1966	-5299	2105	-5598	2222	2365
v_2^I	0	5546	-0027	0909	-3553	0027	-0622	-3041	-4952	-0808	-2131	-0575	-0112	-2668	-0757	-5554	5716
v_3^I	0	0	-1313	-0765	-9831	-2594	-1880	8027	3455	2541	-2504	6611	-1098	-0637	2159	-3021	-0940
v_4^I	0	0	0	8349	-5823	-0585	-3950	-5099	6058	-1130	2169	-7715	4310	-6061	-0156	2999	6243
v_5^O	0	0	0	0	4358	9143	4360	3205	-4261	-0752	-0595	-1433	0219	4846	-1109	0457	6712
v_6^O	0	0	0	0	0	-5516	1903	-3466	0996	2507	0742	4412	3241	-0128	-4531	-0171	-0898
v_7^O	0	0	0	0	0	0	1380	-0413	-4725	-5755	2427	4392	-4545	5817	-4866	-2200	-7568
h_1	0	0	0	0	0	0	0	-1569	4807	-0577	-0395	-0796	-2633	3799	4702	-1537	3292
h_2	0	0	0	0	0	0	0	0	1274	3303	4542	2661	2408	-0423	3445	2003	-1049
h_3	0	0	0	0	0	0	0	0	0	9421	4099	1761	-4561	-0824	-3707	2405	-1580
h_4	0	0	0	0	0	0	0	0	0	0	-9482	-4292	3708	-5064	5116	-3102	-6825
h_5	0	0	0	0	0	0	0	0	0	0	0	-10693	-4500	-5952	4610	-1515	-4579
h_6	0	0	0	0	0	0	0	0	0	0	0	0	10075	-2927	-5589	2537	-8234
h_7	0	0	0	0	0	0	0	0	0	0	0	0	0	-4459	-0662	3866	-4704
h_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0137	-0330	-0786
h_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4713	4098
h_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6301

Table 4: Parameters for Boltzmann machine (10^{-4} units) representing Adder circuit trained at $\beta \sim 5$ as a function approximator. The diagonal terms represent the field strength. The bounds for of training were taken as $H_{max} = 2$ and $J_{max} = 1$. Due to symmetric relation of interaction the lower diagonal is taken to be zero.

	v_1^I	v_2^I	v_3^I	v_4^I	v_5^O	v_6^O	v_7^O	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}
v_1^I	7388	1673	0403	1804	-10000	-1099	3941	-4622	-4148	-4683	-2312	1966	-5299	2105	-5598	2222	2365
v_2^I	0	5546	-0027	0909	-3553	0027	-0622	-3041	-4952	-0808	-2131	-0575	-0112	-2668	-0757	-5554	5716
v_3^I	0	0	-1313	-0765	-9831	-2594	-1880	8027	3455	2541	-2504	6611	-1098	-0637	2159	-3021	-0940
v_4^I	0	0	0	8349	-5823	-0585	-3950	-5099	6058	-1130	2169	-7715	4310	-6061	-0156	2999	6243
v_5^O	0	0	0	0	4358	9143	4360	3205	-4261	-0752	-0595	-1433	0219	4846	-1109	0457	6712
v_6^O	0	0	0	0	0	-5516	1903	-3466	0996	2507	0742	4412	3241	-0128	-4531	-0171	-0898
v_7^O	0	0	0	0	0	0	1380	-0413	-4725	-5755	2427	4392	-4545	5817	-4866	-2200	-7568
h_1	0	0	0	0	0	0	0	-1569	4807	-0577	-0395	-0796	-2633	3799	4702	-1537	3292
h_2	0	0	0	0	0	0	0	0	1274	3303	4542	2661	2408	-0423	3445	2003	-1049
h_3	0	0	0	0	0	0	0	0	0	9421	4099	1761	-4561	-0824	-3707	2405	-1580
h_4	0	0	0	0	0	0	0	0	0	0	-9482	-4292	3708	-5064	5116	-3102	-6825
h_5	0	0	0	0	0	0	0	0	0	0	0	-10693	-4500	-5952	4610	-1515	-4579
h_6	0	0	0	0	0	0	0	0	0	0	0	0	10075	-2927	-5589	2537	-8234
h_7	0	0	0	0	0	0	0	0	0	0	0	0	0	-4459	-0662	3866	-4704
h_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0137	-0330	-0786
h_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4713	4098
h_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6301

Table 5: Parameters for Boltzmann machine (10^{-4} units) representing Adder circuit trained at $\beta \sim 5$ as a distribution. The diagonal terms represent the field strength. The bounds for of training were taken as $H_{max} = 2$ and $J_{max} = 1$. Due to symmetric relation of interaction the lower diagonal is taken to be zero.