A hybrid quantum enabled RBM advantage: convolutional autoencoders for quantum image compression and generative learning

Sleeman, Jennifer, Dorband, John, Halem, Milton

Jennifer Sleeman, John Dorband, Milton Halem, "A hybrid quantum enabled RBM advantage: convolutional autoencoders for quantum image compression and generative learning," Proc. SPIE 11391, Quantum Information Science, Sensing, and Computation XII, 113910B (20 May 2020); doi: 10.1117/12.2558832

SPIE.

Event: SPIE Defense + Commercial Sensing, 2020, Online Only

A Hybrid Quantum Enabled RBM Advantage: Convolutional Autoencoders for Quantum Image Compression and Generative Learning

Jennifer Sleeman[a], John Dorband[b], and Milton Halem[c]

[a]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, USA
[b]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, USA
[c]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, USA

ABSTRACT

Understanding how the D-Wave quantum computer could be used for machine learning problems is of growing interest. Our work explores the feasibility of using the D-Wave as a sampler for a machine learning task. We describe a hybrid method that combines a classical deep neural network autoencoder with a quantum annealing Restricted Boltzmann Machine (RBM) using the D-Wave for image generation. Our method overcomes two key limitations in the 2000-qubit D-Wave processor, namely the limited number of qubits available to accommodate typical problem sizes for fully connected quantum objective functions, and samples that are binary pixel representations. As a consequence of these limitations we are able to show how we achieved nearly a 22-fold compression factor of grayscale 28 x 28 sized images to binary 6 x 6 sized images with a lossy recovery of the original 28 x 28 grayscale images. We further show how generating samples from the D-Wave after training the RBM, resulted in 28 x 28 images that were variations of the original input data distribution, as opposed to recreating the training samples. We evaluated the quality of this method by using a downstream classification method. We formulated a MNIST classification problem using a deep convolutional neural network that used samples from the quantum RBM to train the MNIST classifier and compared the results with a MNIST classifier trained with the original MNIST training data set, as well as a MNIST classifier trained using classical RBM samples. We also explored using a secondary dataset, the MNIST Fashion dataset and demonstrate the first quantum-generated fashion. Our hybrid autoencoder approach indicates advantage for RBM results relative to the use of a current RBM classical computer implementation for image-based machine learning and even more promising results for the next generation D-Wave quantum system. Our method for compression and image mappings is not constrained to RBMs, the autoencoder part of this method could be coupled with other quantum-based algorithms.

Keywords: Quantum Computing, Quantum Annealing, D-Wave, Restricted Boltzmann Machine, Autoencoder, Deep Learning, Data Compression

1. INTRODUCTION

The D-Wave 2000Q quantum annealing system is an adiabatic quantum system,[1] which is composed of a 2048-qubit processor. The quantum annealing algorithm can be used to achieve a performance improvement for a number of optimization problems.[2–4] Recent work has explored both the theoretical prospects of machine learning on the D-wave[5–7] and using the D-Wave for machine learning applications.[8,9] Building on previous work, we describe a hybrid system that uses a deep convolutional autoencoder neural network that provides a way to compress images to be used for quantum machine learning. We demonstrate this compression technique using a Restricted Boltzmann Machine (RBM) and the D-Wave 2000Q. The D-Wave 2000Q is limited by the number of qubits available, which limits the overall size of the problem that can be embedded on the D-Wave. In addition, using the D-Wave for image-based sampling presents another challenge in that only binary information can be sampled. We overcome both of these issues in our work by using a convolutional autoencoder to provide a translation from

Quantum Information Science, Sensing, and Computation XII, edited by Eric Donkor, Michael Hayduk, Proc. of SPIE Vol. 11391, 113910B · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2558832

images represented on the classical machine and image representation learned using the D-Wave. In doing so, we were able to achieve compression from an image size of 28 x 28 grayscale to an image size of 6 x 6 binary and recover the original 28 x 28 grayscale (with some loss). We show how to train a RBM using the D-Wave as the sampler and how sampling from the D-Wave provides enough low-level noise to obtain new image variants. We compare the results of a hybrid quantum RBM with a hybrid classical RBM using a downstream classification problem. This approach for quantum compression and image mapping is not constrained to quantum RBMs, but could be coupled with other quantum-based algorithms.

2. BACKGROUND

RBMs have a long history dating back to the original paper introducing Boltzmann machines.[10] Boltzmann machines are known for their intractability on classical machines[11] due to the connectivity among units. In addition to every unit in the visible layer being connected to every unit in the hidden layer, the Boltzmann machine also contains connections between units within the same layer. RBMs are restricted in that there is no connectivity among units within a given layer. An early example of an autoencoder, the RBM is a simple neural network. It is distinct from other neural networks in that it is probabilistic and represents an undirected graphical model.

Due to this property, RBMs can be used to learn the stochastic representation of its input, modeling the input distribution.[11] During training, the model parameters are changed so that the probability distribution fits the input data.[12]

The general structure of a RBM is one visible layer, one hidden layer, and two corresponding bias vectors as shown in Figure 1. RBMs in the standard form have binary values for the v visible and h hidden units.

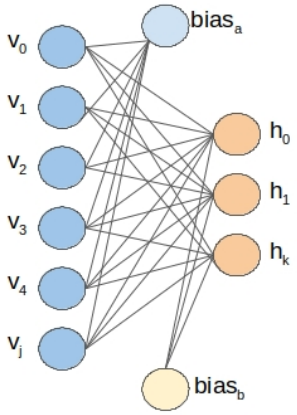

Figure 1. Restricted Boltzmann Machine Network Diagram.

2.1 Energy Function

The RBM is considered an energy based model where it uses energy as a metric for measuring the quality of the model.[13] The energy of a given combination of v and h units is defined by the following equation:

$$E(v,h) = \sum_i a_i v_i - \sum_j b_j h_j - \sum \sum v_i h_j w_{i,j} \tag{1}$$

Where v represents the visible units and h represents the hidden units, a and b represent the bias, and w represents the weight matrix. E defines a probability distribution over v and h.

The joint probability distribution is defined in terms of the following equation (Gibbs distribution):

$$p(v,h) = \frac{e^{-E(v,h)}}{Z} \tag{2}$$

Z is the summation over all pairs of visible v and hidden h vectors and acts as the normalizing constant for this equation.

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

Hidden and visible unit states are independent and calculated using conditional probabilities, where the conditional probability of the visible units state is conditioned upon the current state of the hidden units, and conditional probability of the hidden units state is conditioned upon the current state of the visible units. This independence is due to the fact that there are no connections between units of a given layer. This improves the Gibbs sampling method because the states for a given layer can be jointly sampled.[12]

2.2 Learning

Maximizing the probability of the training data for the model can be compared to the maximum likelihood estimation, where a likelihood function is maximized by some configuration of the state space. To maximize the likelihood function, the gradients of the log-likelihood are needed. Calculating these gradients is not tractable, therefore typically Gibb's sampling,[14] a Markov Chain Monte Carlo (MCMC) method, is used to sample from the joint Boltzmann distribution. Often the Gibb's sampling requires many steps and can become computationally expensive. Therefore, a method known as Contrastive Divergence[13] can be used to perform only n steps of the Gibb's sampling. It has been shown that even in some cases 1 step is enough for training a RBM.[13] Given a set of training samples the RBM is trained to learn how to adjust model state such that the probability distribution is fit to the training data probability distribution.

2.3 Generative Models

Generative models model the joint probability distribution. RBMs learn a joint probability distribution $P(v, h)$, where v represents the visible units and h represents the hidden units. The joint distribution is used to calculate the probability of a configuration. Configurations of v and h are fixed during training but are not fixed after training. Given $P(v, h)$, sampling from this distribution, could enable generated output that is not necessarily a recreation of a sample from the input distribution. Previous work has used RBMs for generative sampling.[15–17]

2.4 RBMs on the D-Wave

In order to implement a RBM using the D-Wave, the problem to be solved on the D-Wave needs to be expressed as a QUBO objective function.[18] We express the QUBO in terms of a chimera graph[19] which is the architecture formed by D-Wave qubit connectivity. This architecture (a 2-D grid) entails groupings of 'unit cells' that are connected, where each cell contains a set of qubits that have bipartite connectivity locally and is connected to qubits in other sets through couplers.[20] Minor embedding entails mapping logical qubits to physical qubits where nodes and edges map to logical qubits and couplers in the chimera graph. This architecture is similar to the architecture of a RBM. Bias noise in this model are different than bias noise in classical RBMs, in that quantum RBM bias variables are random.[21]

We follow the approach of learning image representation using the RBM, having each pixel of the image represented by a visible unit of the RBM. However, since images go through the autoencoder before RBM processing, we formulate an encoding based on the number of pixels and a compression size.

3. RELATED WORK

Early work by Dorband[22] explored a RBM implementation using the D-Wave. This implementation used a different approach and was not necessarily identifying generative differences between a classical and quantum approach. Generative sampling using the D-Wave and a similar approach for the RBM, in the past, yielded poor results. In the work by Thulasidasan et al.,[23] generative sampling after training MNIST down-scaled samples were not visibly distinguishable. Previous work by Adachi et al.[24] described a generative RBM with 32 visible

nodes and 32 hidden nodes using 512 qubits for sampling. They showed that the quantum sampling was able to achieve comparable accuracy to the classical system with fewer iterations.

Work by Romero et al.[5] described the need for tools which reduce experimental overhead as advantageous. They describe the idea of a quantum autoencoder. Though they describe using the quantum autoencoder for compression, they solve a different problem than what we propose. Work by Ni et al.[25] performed a comparison between a classical RBM and a quantum RBM using binary problems and saw some improvements using the quantum RBM. Most experiments were simple binary problems. We build on this work and extend it to support RBMs for real problems (such as MNIST). Recent work by Amin et al.[6] explored using a quantum RBM for generative sampling. Interestingly, when experimenting with their model of fully-connected 8 inputs and 3 outputs of binary data, they showed that the distribution learned using the quantum method when compared with the classical for a small test was very different from the actual distribution. Recent work by Khoshaman et al.[7] used a quantum Boltzmann machine to generate the latent space for a Variational autoencoder, showing state of the art results using the MNIST dataset. This work is most closely related to our work in that we both explore using the D-Wave for sampling to generate latent space. However, our approach includes providing a hybrid classical quantum approach to overcome quantum hardware limitations that affect the number of qubits available to represent problems. Research that has been quickly working towards progress related to generative learning includes our work[26] and work by Vinci et al.[27] which explored a theoretical understanding of generative-based learning using variational autoencoders on the D-Wave.

To summarize, many of the early theoretical contributions have shown that using the D-Wave for generative sampling can enable faster learning of the latent space. However, as Amin et al.[6] concluded, the learned distribution can deviate significantly from a given actual distribution. We believe the classical autoencoder performing the translation between the classical system and the quantum system acts as a stabilizer for the latent space since part of the latent space is captured on the classical side.

4. APPROACH

Our approach is designed to address the challenges of image sampling using the D-Wave with regards to the number of qubits available by simultaneously mapping binary output from the D-Wave to floating point values and the original image space to a compressed image space. The overall architecture of our approach is shown in Figure 2 using the MNIST dataset as an example dataset.

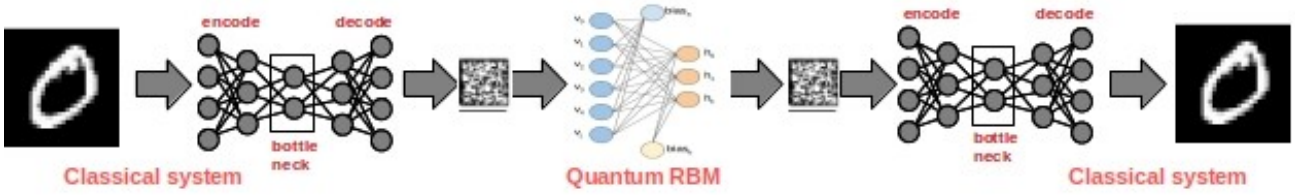

Figure 2. Hybrid Approach that used a Classical Autoencoder to map the Image Space to a Compressed Space.

An autoencoder is trained on the original input, a bottleneck is defined in terms of the compression size, and the autoencoder learns how to map compressed encodings to the original grayscale full resolution images. We used a convolutional autoencoder and experimented with compression sizes of 6 x 6 and 7 x 7. The bottleneck of the autoencoder was used to generate the binary compressions that could be decoded. The classical autoencoder architecture is shown in Figure 3.

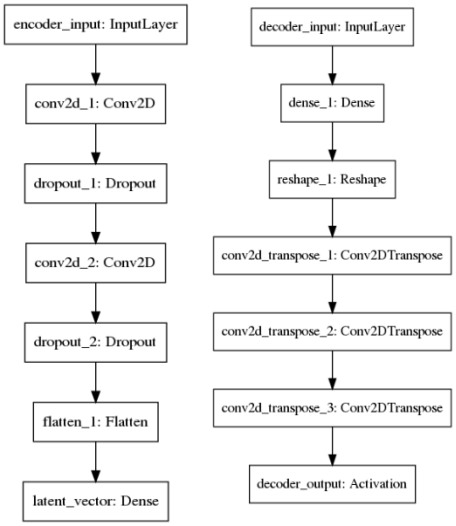

Figure 3. Classical Autoencoder Architecture.

After training the autoencoder, we took the encoded data and used it to train the RBM. Each pixel of the encoded input was represented by a unit in the visible layer. For example, a 6 x 6 compression would have 36 units in the visible layer. To create a mapping of this for use on the D-Wave we created an embedding which had 36 visible units, 18 hidden units, and connectivity from each visible unit to each hidden unit. Hence, with 2048 qubits available (not all are available) on the D-Wave 2000Q system, if we compress larger than a 7 x 7 size we go beyond the capacity of the D-Wave.

For the quantum RBM, we used the D-Wave API and the D-Wave 2000Q solver. We created a BinaryQuadraticModel for the QUBO and used minorminer to create the embedding. Samples were obtained using the DWaveSampler. An example of an embedding created for a 6 x 6 problem is shown in Figure 4.

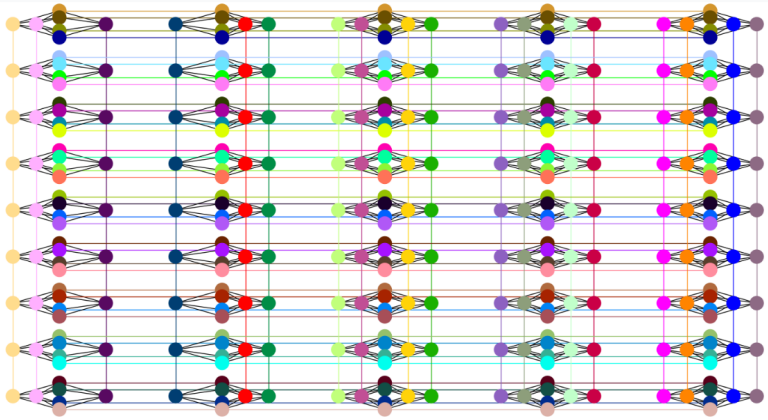

Figure 4. Logical Embedding on D-Wave for 6 x 6 Problem.

The RBM was trained to learn the best configuration for recreating the compressed encodings, i.e. the learned data distribution moves towards the actual training data distribution. After convergence (the learned distribution no longer improves), we used the trained RBM for sampling binary encodings (invoking the D-Wave sampler). These encodings were then decoded by the (classical) autoencoder to obtain images in the original representation space, both in size and in pixel values (grayscale rather than binary).

5. EXPERIMENTAL SETUP

The MNIST dataset was used to evaluate the sampled output. The autoencoder was executed on a classical machine (Intel Core i7-7700HQ CPU 2.8GHz x 8) with 32 GBs of memory, using a GeForce GTX 1060 GPU and received as input grayscale MNIST digits at a size of 28 x 28. The output after encoding was in the representation of compressed encodings of a designated size (7 x 7 or 6 x 6). The autoencoder was trained using the 60,000 MNIST training digits. The RBM was trained using the encodings of these 60,000 digits. The D-Wave was used for sampling to obtain learned binary encodings. The samples obtained from the D-Wave were then decoded using the classical autoencoder to recover 28 x 28 sized grayscale MNIST images. To assess the quality of the MNIST generated images, we performed downstream classification experiments. Note, these experiments did not include the new D-Wave Hybrid API (our experimentation began before the Hybrid API was available).

A comparable classical RBM was also built and trained on the same encodings. The classical RBM was built to be as similar as possible to the quantum RBM. Both methods used a sampling method and contrastive divergence. Both had a visible layer with unit size equal to the number of encoded values (36 for 6 x 6 compressions and 49 for 7 x 7 compressions) and a hidden layer that was half of the size of the visible layer. However, due to the nature of the D-Wave, numeric representations vary.

After training the quantum RBM, the D-Wave sampler was used to obtain 60,000 quantum RBM samples. Each class was balanced based on the original size of the MNIST training data. We used samples to train a downstream deep convolutional neural network to classify 10,000 unseen MNIST test digits. We compared these results with training a downstream deep convolutional neural network, using the original MNIST training data, and classifying the same 10,000 unseen MNIST test digits. We also compared these results with training a downstream deep convolutional neural network, using the classical RBM-recreated samples as training data, and classifying the same 10,000 unseen MNIST test digits. With the classical RBM, Gibbs sampling was used.

Though classical RBMs and quantum RBMs are different in the way they learn, this comparison provides insight into how the D-Wave sampling method compares to the Gibb's sampling method performed on the classical system. These experiments are intended to be the overfit case, in that we trained on a set of images and tried to regenerate those training samples. The training data in each experiment consisted of 60,000 samples (with a subset reserved for validation) and the test set consisted of 10,000 samples (unseen). We trained the classifier for 5 epoch, since accuracy can reach 100% with the original data set in 5 epoch.

6. EXPERIMENTAL RESULTS

As shown in Table 1, the classifier that was trained with the original MNIST training data achieved ∼99% accuracy (ExpID 1). This measure serves as a benchmark for maximum accuracy that could be achieved. We then used samples from a classical RBM after training it on the original 28 x 28 grayscale MNIST digits. The learned samples were then used to train the downstream MNIST classifier. In this case, the result was ∼98% (ExpID 2), as shown in Table 1. We used this as a baseline for what the classical RBM, trained on the original data, could achieve.

ExpID	**Training Data Variants**	**Accuracy On Test Set**
1	Original MNIST 28 x 28 digits (baseline)	0.99
2	Sampled from Classical RBM MNIST 28 x 28 digits (no encodings)	0.98

Table 1. Comparing MNIST classification accuracy scores when using the original training data and samples generated from a classical RBM.

The next set of experiments, as shown in Table 2, shows the results of the classical autoencoder binary compression recovery of the original MNIST data representation. ExpID 3 evaluated MNIST digits that were converted from grayscale to binary. ExpID 3 conveys an evaluation of compression only, the method of converting from grayscale to binary and back to the original sized grayscale images. The evaluation in this case resulted in

∼97% accuracy (averaged over three runs). This was compared to compressing the digits to a binary 16 x 16 size then decoding back to 28 x 28 grayscale (ExpID 4). When using these images for training the classifier, it achieved ∼95% accuracy. The importance of a 16 x 16 compression is that this provides an anticipated size that the next generation D-Wave will support, given an increase in the number of qubits that will be available. We treat this accuracy as a potential upper bound for what could be achieved when a 16 x 16 sized compression could be supported on the D-Wave. ExpID 5 evaluated MINST digits compressed to a binary 6 x 6 size and decoded back to the 28 x 28 grayscale representation, achieving ∼91% accuracy. These encoding/decoding results provide a secondary baseline for results, in that they provide an upper bound for accuracy that could be obtained using encoded learned samples from a RBM.

ExpID	**Training Data Variants**	**Accuracy On Test Set**
3	Encoded/Decoded Binary Translated MNIST 28 x 28 digits (no RBM, no compression, just binary to grayscale mappings)	0.97
4	Encoded/Decoded Binary MNIST 16 x 16 decoded to 28 x 28 digits (no RBM)	0.95
5	Encoded/Decoded Binary MNIST 6 x 6 decoded to 28 x 28 digits (no RBM)	0.91

Table 2. Comparing MNIST classification accuracy when using different compressed encodings that are decoded to recover the original MNIST representations.

Early examples of 6 x 6 grayscale encoding/decoding results are shown in Figure 5 and the more recent examples are shown in Figure 6, where the first row represents the original 28 x 28 MNIST digits, the second row represents the 6 x 6 grayscale encodings, the third row represents the 6 x 6 binary encodings and the fourth row represents the recovered 28 x 28 digits. It is observed from the decoded digits that there are times when the digits are incorrectly decoded in the case of 6 x 6. This is due to the loss incurred when compressing from the 28 x 28 to 6 x 6 binary representation. With more recent results, we improved our overall performance in recovering digits but as shown in Figure 6, there were samples which decoded to the wrong digit. Future versions will explore using a variational autoencoder.

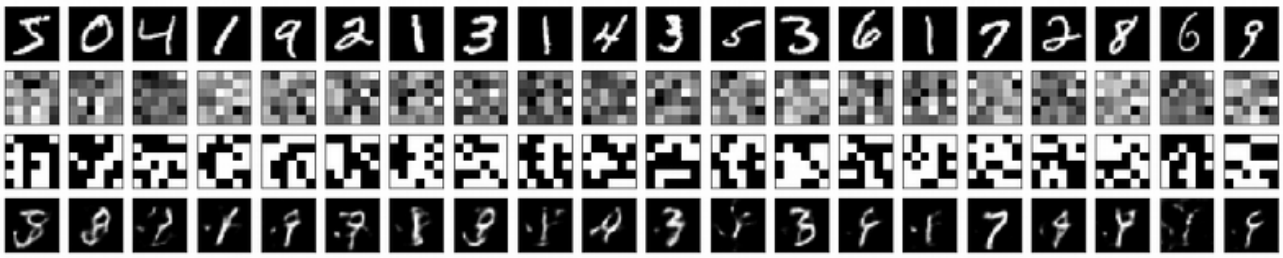

Figure 5. Early Results: Examples of 6 x 6 encoding and binary decoding to recover 28 x 28 grayscale MNIST digits.

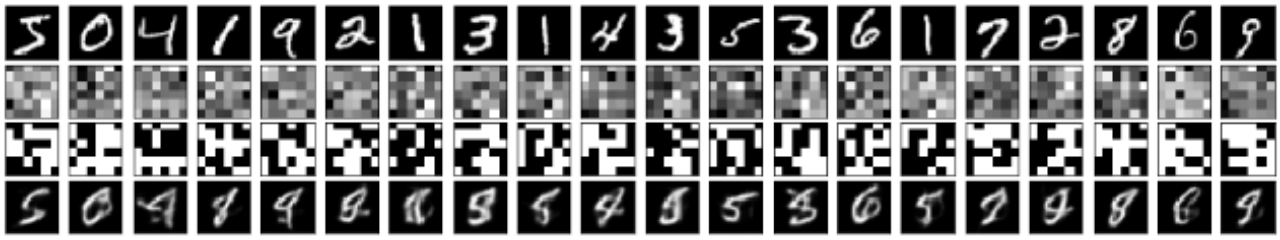

Figure 6. More Recent Results: Examples of 6 x 6 encoding and binary decoding to recover 28 x 28 grayscale MNIST digits.

In Table 3, results are conveyed that are related to using MNIST digits recovered from the encodings of the size required for embedding the RBM model on the D-Wave. These results include experimentation related to MNIST digits encoded to a 6 x 6 representation. We trained a purely classical RBM and also the quantum RBM. The number of visible layers was composed of 36 units. The number of hidden layers was composed of 18 units for both RBMs. We set the learning rate for the classical RBM to be the same as the quantum RBM.

We achieved ∼75% accuracy using the downstream classifier with the samples obtained from Gibbs sampling using the classical RBM (ExpID 6). We achieved ∼72% accuracy using the downstream classifier with the samples obtained from the quantum RBM (ExpID 7). By modifying the autoencoder to be a denoising autoencoder, by modifying dropout layer parameters, and by increasing the RBM epoch, we were able to achieve ∼12% increase in downstream classification results which gave us ∼72% accuracy (previously we achieved ∼60% accuracy on average).

ExpID	**Training Data Variants**	**Accuracy On Test Set**
6	Encoded/Decoded Binary MNIST 6 x 6 to translated to 28 x 28 digits classical RBM using Gibbs sampling	0.75
7	Encoded/Decoded Binary MNIST 6 x 6 to translated to 28 x 28 digits RBM using the D-Wave as a sampler	0.72

Table 3. Comparing MNIST classification accuracy when using classical and quantum RBM samples.

For the quantum RBM, using 6 x 6 binary encodings, we were able to recover MNIST digits, as shown in Figure 7 (early results), when sampling from the D-Wave after training and using the classical autoencoder to decode sampled encodings. Early results showed the quality of the digits tended to have a better appearance when using 7 x 7 binary encodings sampled from the D-Wave and decoded using the classical autoencoder, as shown in Figure 8.

Figure 7. Early Results: Recovered MNIST digits from the quantum RBM after a 6 x 6 binary encodings.

Figure 8. Early Results: Recovered MNIST digits from the quantum RBM after a 7 x 7 binary encodings.

A sample of the results after improvements in the autoencoder and RBM showed improved digits for both 6 x 6 and 7 x 7 encodings, as shown in Figure 9. Though this selection of samples are visually similar to the original MNIST digits, there are also samples that were decoded to incorrect digits or distorted versions of the digits. Downstream classification results were not significantly better using 7 x 7 decoded samples.

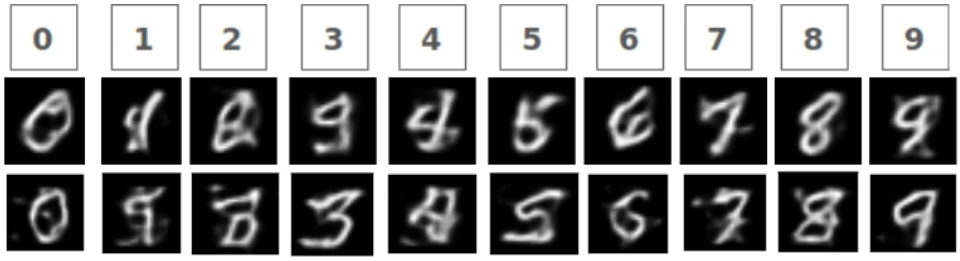

Figure 9. More Recent Results: Recovered MNIST digits from the quantum RBM after a 6 x 6 binary encodings (middle row) and 7 x 7 binary encodings (bottom row).

6.1 Generating MNIST Samples: Taking Advantage of the D-Wave's Inherent Noise

We examined individual MNIST digits of large samples taken from the D-Wave. As an example, a large sampling of the MNIST digit 3, after training the quantum RBM using the D-Wave as the sampler, is shown in Figure 10. We sampled 100,000 3's in this exploration task.

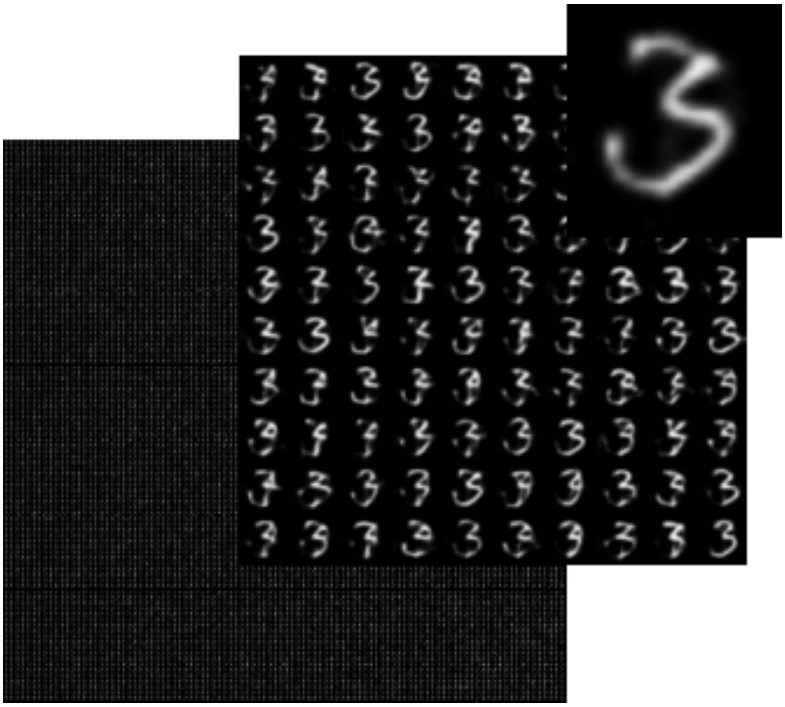

Figure 10. D-Wave samples of the digit 3 after training the RBM.

We also show binary output and recovered digits in Figure 11 from the D-Wave sampling. As can be seen in Figure 11, both binary output and recovered digits represent a distinct set of 3's that we believe were not necessarily part of the original training distribution. We were able to produce these variations in samples trained on hundreds of samples of the original training digits. Reproducing these variations in digits was repeatable on the original D-Wave 2000Q system.

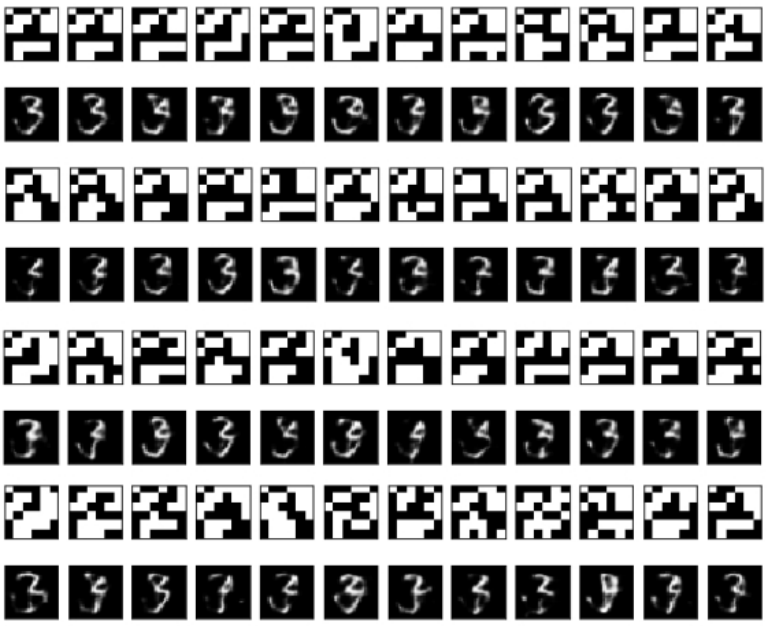

Figure 11. D-Wave samples of the digit 3 after training the RBM.

However, when the D-Wave 2000Q was replaced with the D-Wave 2000Q lower-noise system, (holding all parameters constant), we were not able to reproduce these results, as shown in Figure 12. We concluded from these results that there was enough thermal temperature fluctuations that enabled the D-Wave sampling to result in variations in sampled encodings, and digits after decoding the generated encodings.

Figure 12. Using Low Noise D-Wave Quantum Annealing for RBM sampling training on the digit 3.

With the D-Wave 2000Q lower-noise system, after performing extensive experimentation, we found that when we reduced the learning rate and increased the number of epoch, we were able to again achieve these variations in samples.

6.2 Measuring Image Similarity

Since we observed that the encodings that were sampled from the D-Wave were resulting in new image variants, we developed experiments to understand how much duplication existed within the generated data itself. We used a metric to measure image structural similarity,[28] defined by the following equation:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma xy + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4}$$

Where x,y are the two images to be compared and μ_x is the average of x, μy is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σxy is the correlation coefficient of x and y.

We used this method to measure how much similarity there is among samples within a dataset. Using SSIM for the whole dataset would require comparing each image against 60,000 images, a n^2 complexity. This computation would be extremely computer intensive. Instead, given we have a dataset, $mnist_{generated}$, we randomly selected n set of images from $mnist_{generated}$ and compared each image with the rest of the $59,999$ generated images and averaged the results to calculate the SSIM.

To evaluate what is the best n, we experimented with different values for n. In Table 4, results related to the average SSIM score are shown, given the number of samples used to calculate it for a given dataset of size 60,000 samples. Given 10 samples, 100 samples, or 1000 samples, a stable average of how much similarity there is across digits is consistent. Therefore, we used 100 samples to measure similarity among images in a given dataset.

Dataset	**Average Image Sample Similarity score**	**Number of Samples to Compare**
Original MNIST 28 x 28 digits	0.56	10
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits (no RBM)	0.55	10
Original MNIST 28 x 28 digits	0.57	100
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits (no RBM)	0.55	100
Original MNIST 28 x 28 digits	0.56	1000
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits (no RBM)	0.55	1000

Table 4. Measuring how similar a sample of decodings is to the remaining decodings in the dataset after encoding/binary decoding and comparing this to measuring similarity among the original MNIST digits.

To understand if measuring image overlap is consistent with human intuition, we measured image overlap for a given class, for each of the 10 classes in the MNIST dataset. We calculated SSIM measures for each class of the original training data set, using 100-sized sample sets. The digits which are classified as the number 1 tended to have a higher average SSIM score, as would be expected since the number 1 tends to be more simplistic than the other digits $(0 - 9)$. These results are shown in Table 5.

Digit	Total Size	Average Image Sample Similarity score
0	5923	0.58
1	6742	0.78
2	5958	0.58
3	6131	0.60
4	5842	0.61
5	5421	0.59
6	5918	0.62
7	6265	0.66
8	5851	0.61
9	5949	0.65

Table 5. Measuring dataset similarity using SSIM by Digit using the original MNIST dataset.

We compared this measure for the original MNIST dataset with digits that were decoded after encoded and compressed down to a 7 x 7 size (since they appeared to be visually more similar to the original MNIST digits than the 6 x 6 compressed encodings after decoding). For 7 x 7 after encoding and binary decoding to the original grayscale 28 x 28 representation, image sample similarity scores shows only a $\sim$.01 difference from the original MNIST to the encoded/decoded recovered digits. Table 6 conveys a comparison of image sample similarity scores for the original MNIST dataset, the encoded/decoded recovered digits, and the classical RBM learned encoded/decoded recovered digits. As observed, there tends to be more duplication among what is learned using the classical RBM. With the D-Wave generated 7 x 7 encodings, after decoding to the 28 x 28 images, we measured on average about 59% structural similarity. With 6 x 6 encodings, after decoding to the 28 x 28 images, we measured on average about 61% structural similarity. Overlap measurements can vary dependent upon autoencoder and RBM parameters, and can be useful as a metric in support of parameter tuning. Future experiments will explore these measures further.

Dataset	Average Image Sample Similarity score (Using 100 samples)
Original MNIST 28 x 28 digits	0.57
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits (no RBM)	0.55
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits classical RBM using Gibbs sampling	0.74
Encoded/Decoded Binary MNIST 7 x 7 to translated to 28 x 28 digits quantum RBM using D-Wave sampling	0.59
Encoded/Decoded Binary MNIST 6 x 6 to translated to 28 x 28 digits quantum RBM using D-Wave sampling	0.61

Table 6. Measuring dataset similarity using SSIM and comparing the original MNIST dataset with the encoded/binary decoded dataset, with the Classical RBM learned encoded/binary decoded dataset, and with the Quantum RBM learned binary encoded/decoded dataset.

6.3 The First Quantum Generated Fashion

In addition to the MNIST dataset, we also trained the quantum-based RBM using the MNIST-Fashion dataset.[29] Designed to be similar to the MNIST dataset, however more challenging to classify, the Fashion MNIST dataset provides another dataset for experimenting with the D-Wave. There are 10 classes, 60,000 images in total, and the images are grayscale sized at 28 x 28.

Figure 13. Fashion MNIST[29]

In Figure 14 the first row is the original grayscale fashion images, the second row is the 7 x 7 encoded fashion, the third row is the 7 x 7 binary encoded fashion and the final row is the 28 x 28 grayscale decoded fashion. As can be observed, the autoencoder tends to have a harder time generating details on shirts, shoes, and handbags.

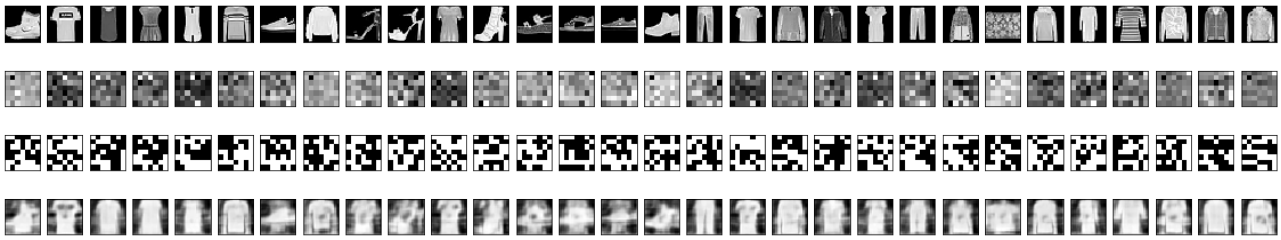

Figure 14. Fashion MNIST Generated By Sampling the D-Wave After Training the Quantum RBM and Decoding the 7 x 7 Samples.

In Figure 15, decoded samples from the D-Wave after training the quantum RBM are shown. Future experiments will include classification of generated Fashion MNIST samples.

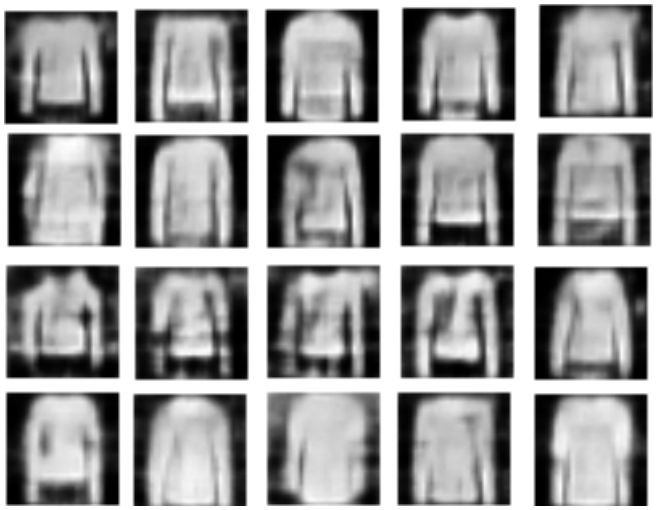

Figure 15. New Shirts - Fashion MNIST Generated By Sampling the D-Wave After Training the Quantum RBM and Decoding the 7 x 7 Samples.

7. OBSERVATIONS AND CONCLUSIONS

The method we described provides a way to overcome the limitations of the D-Wave 2000Q by providing a hybrid method from mapping original data representations to a representation that could be processed using quantum annealing.

Our prior attempts using the classical autoencoder for this work were unsuccessful, where learned encodings from the D-Wave did not decode to digits. Initially we began with a fully connected autoencoder, which showed only hints of a digit recovered. We saw improvement when moving to a convolutional neural network for the autoencoder. However, the real improvements came when using a denoising autoencoder for this work. By applying Gaussian noise to the images prior to encoding them (and modifying dropout parameters), we saw improved quality when decoding the quantum sampled learned encodings.

Using a classical RBM with Gibb's sampling did not produce the same sort of variations (as expected) as when the D-Wave was used as the sampler. In fact, when running a number of experiments varying the epoch and learning rate on the classical RBM, we often saw the network learning encodings that resulted in duplicate digits after decoding. We saw this more frequently using 6 x 6 compressed encodings. As we increased the compression size to 16 x 16 encodings we could consistently see the classical RBM learning different variants of digits across varying parameters. We conclude from these experiments that the quantum RBM was able to tolerate and learn from these highly compressed encodings whereas the classical RBM was less tolerable. This can in part be due to the fact that though we were providing the classical RBM with binary encodings, during training there is a difference in numeric precision than on the D-Wave. Though the encoded samples from the classical RBM differ from each other, the decoded MNIST resulted in duplicated digits more frequently. This behavior is reflected in the SSIM results. High classification scores for the classical RBM were still achieved, this is in part due to the simplicity of the MNIST dataset and the fact that the classical RBM did have digits that were visually more closely related to the actual MNIST, as shown in Figure 16. Though they are visually better recreations of the MNIST, there was higher duplication across decoded digits.

Figure 16. Recreated MNIST digits from the classical RBM after a 7 x7 binary encoding.

Though RBMs are generative, using a RBM to generate new images from a latent representation, as opposed to recreating training samples, is not typically performed on the classical computer but has been achieved, for example with deep layered RBMs.[30] By taking advantage of the inherent noise on the D-Wave and the natural quantum properties of the D-Wave, we have been able to successfully use it to generate images. By generating images from sampling the D-Wave, using the 2000Q lower-noise system, we conclude that the variations in images were not solely due to thermal temperature fluctuations. However, more experiments are required to prove this claim completely. Though there were runs when the quantum D-Wave RBM learned only a single digit, we were able to successfully, repeatably create these variations. In addition, we observed when epoch reaches a certain threshold, the samples tend to collapse to a single image.

Given the samples collected from the D-Wave after training the RBM using the MNIST dataset encodings and also after training the RBM using the MNIST-Fashion dataset, we show early quantum image generation results. Thus, paraphrasing Biamonte et. al., "if a small quantum [D-Wave 2000Q] processor can produce statistical patterns that are computationally difficult to be produced by a classical computer, then larger quantum annealers [perhaps the 5000 qubit D-Wave processor using the above hybrid RBM method] might recognize patterns that are significantly more difficult to recognize classically".

ACKNOWLEDGMENTS

This work was supported by D-Wave for system access and support as member in NSF-funded CHMPR, and through the NASA grant for Quantum, grant number #NNH16ZDA001N-AIST16-0091.

REFERENCES

[1] Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M., "Quantum computation by adiabatic evolution," *arXiv preprint quant-ph/0001106* (2000).

[2] Hashizume, Y., Koizumi, T., Akitaya, K., Nakajima, T., Okamura, S., and Suzuki, M., "Singular-value decomposition using quantum annealing," *Physical Review E* **92**(2), 023302 (2015).

[3] Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., and Parney, B., "Traffic flow optimization using a quantum annealer," *Frontiers in ICT* **4**, 29 (2017).

[4] Ushijima-Mwesigwa, H., Negre, C. F., and Mniszewski, S. M., "Graph partitioning using quantum annealing on the d-wave system," in [*Proceedings of the Second International Workshop on Post Moores Era Supercomputing*], 22–29, ACM (2017).

[5] Romero, J., Olson, J. P., and Aspuru-Guzik, A., "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology* **2**(4), 045001 (2017).

[6] Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., and Melko, R., "Quantum boltzmann machine," *Physical Review X* **8**(2), 021050 (2018).

[7] Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., and Amin, M. H., "Quantum variational autoencoder," *Quantum Science and Technology* **4**(1), 014001 (2018).

[8] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S., "Quantum machine learning," *Nature* **549**(7671), 195 (2017).

[9] O'Malley, D., Vesselinov, V. V., Alexandrov, B. S., and Alexandrov, L. B., "Nonnegative/binary matrix factorization with a d-wave quantum annealer," *PloS one* **13**(12), e0206653 (2018).

[10] Ackley, D. H., Hinton, G. E., and Sejnowski, T. J., "A learning algorithm for boltzmann machines," *Cognitive science* **9**(1), 147–169 (1985).

[11] Salakhutdinov, R. and Hinton, G., "Deep boltzmann machines," in [*Artificial intelligence and statistics*], 448–455 (2009).

[12] Fischer, A. and Igel, C., "Training restricted boltzmann machines: An introduction," *Pattern Recognition* **47**(1), 25–39 (2014).

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

[14] Carter, C. K. and Kohn, R., "On gibbs sampling for state space models," *Biometrika* **81**(3), 541–553 (1994).

[15] Taylor, G. W., Hinton, G. E., and Roweis, S. T., "Modeling human motion using binary latent variables," in [*Advances in neural information processing systems*], 1345–1352 (2007).

[16] Schmah, T., Hinton, G. E., Small, S. L., Strother, S., and Zemel, R. S., "Generative versus discriminative training of rbms for classification of fmri images," in [*Advances in neural information processing systems*], 1409–1416 (2009).

[17] Sukhbaatar, S., Makino, T., Aihara, K., and Chikayama, T., "Robust generation of dynamical patterns in human motion by a deep belief nets," in [*Asian Conference on Machine Learning*], 231–246 (2011).

[18] Boros, E., Hammer, P. L., and Tavares, G., "Local search heuristics for quadratic unconstrained binary optimization (qubo)," *Journal of Heuristics* **13**(2), 99–132 (2007).

[19] Cai, J., Macready, W. G., and Roy, A., "A practical heuristic for finding graph minors," *arXiv preprint arXiv:1406.2741* (2014).

[20] D-Wave, "Wave qpu architecture: Chimera."

[21] Dumoulin, V., Goodfellow, I. J., Courville, A., and Bengio, Y., "On the challenges of physical implementations of rbms," in [*Twenty-Eighth AAAI Conference on Artificial Intelligence*], (2014).

[22] Dorband, J. E., "A boltzmann machine implementation for the d-wave," in [*2015 12th International Conference on Information Technology-New Generations*], 703–707, IEEE (2015).

[23] Thulasidasan, S., "Generative modeling for machine learning on the d-wave," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2016).

[24] Adachi, S. H. and Henderson, M. P., "Application of quantum annealing to training of deep neural networks," *arXiv preprint arXiv:1510.06356* (2015).

[25] Ni, S. and Nagayama, S., "Performance comparison on cfrbm between gpu and quantum annealing," tech. rep., Mercari (2018).

[26] Sleeman, J., Halem, M., Dorband, J. E., et al., "Rbm image generation using the d-wave 2000q," in [*D-Wave Qubits North America Conference*], (2019).

[27] Vinci, W., Buffoni, L., Sadeghi, H., Khoshaman, A., Andriyash, E., and Amin, M. H., "A path towards quantum advantage in training deep generative models with quantum annealers," *arXiv preprint arXiv:1912.02119* (2019).

[28] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), 600–612 (2004).

[29] Xiao, H., Rasul, K., and Vollgraf, R., "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747* (2017).

[30] Hu, H., Gao, L., and Ma, Q., "Deep restricted boltzmann networks," *arXiv preprint arXiv:1611.07917* (2016).