UNIVERSITY OF CALIFORNIA SAN DIEGO

A Statistical Investigation of Model Quality in Generative Systems

A thesis submitted in partial satisfaction of the
requirements for the degree
Masters of Science

in

Electrical and Computer Engineering
(Machine Learning and Data Science)

by

Alexander Potapov

Committee in charge:

Professor Ken Kreutz-Delgado, Chair
Professor Piya Pal
Professor Michael Yip

2018

The thesis of Alexander Potapov is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

DEDICATION

To my mother and grandmother, for introducing me to the wonders of science, math and history from an early age.

EPIGRAPH

It is change, continuing change, inevitable change, that is the dominant factor in society today. No sensible decision can be made any longer without taking into account not only the world as it is, but the world as it will be ... —*Isaac Asimov*

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Epigraph . . . v

Table of Contents . . . vi

List of Figures . . . viii

List of Tables . . . ix

Acknowledgements . . . x

Vita . . . xi

Abstract of the Thesis . . . xii

Chapter 1 Introduction . . . 1

Chapter 2 Generative Models . . . 5
2.1 Restricted Boltzmann Machines . . . 6
2.2 Deep Belief Networks . . . 7
2.3 Quantization of the Model . . . 7

Chapter 3 Maximum Mean Discrepancy . . . 10
3.1 Problem Definition: What is the test to do? . . . 11
3.2 A Direct Alternative . . . 13
3.3 Results as Achieved by Baseline MMD . . . 16
3.4 Expanding on MMD . . . 18

Chapter 4 Permutation Testing . . . 19
4.1 Defining the Permutation Test . . . 19
4.2 Executing Permutation Testing . . . 23
4.2.1 Initial Approach: Whole Set Testing . . . 23
4.2.2 Second Approach: Label-Wise Testing . . . 25
4.2.3 Multiple Hypothesis Testing . . . 28
4.3 Conclusions of Permutation Testing . . . 30

Chapter 5 Annealed Importance Sampling For Quantization Selection . . . 32
5.1 Integration with Permutation Testing . . . 33
5.2 AIS For Bitwidth Selection . . . 34

Chapter 6 Unification through Workflow 37
6.1 Description 37
6.2 Operation 39
6.3 Results 40
6.4 Conclusions 44

Chapter 7 Conclusions and Future Work 45
7.1 General Thoughts 46
7.2 HaarPSI Wavelet Metrics 47
7.2.1 Human vs Computer Vision 48
7.2.2 Integration with Permutation Testing 48
7.3 Scaling to Other Systems 49
7.4 Final Thoughts 50

Appendix A Annealed Importance Sampling 51
A.1 General Overview 51
A.2 Derivation 52
A.3 AIS for MNIST 53

Bibliography 55

LIST OF FIGURES

Figure 3.1: Results of Running MMD on the Primary Models 17

Figure 4.1: Mean Image Comparison 25
Figure 4.2: MMD and PVal of Each Label 27
Figure 4.3: Mean MMD and P-Value of all Labels 28

Figure 5.1: Quantization Optimization 34

Figure 6.1: Flowchart of Work Process 38
Figure 6.2: Workflow Training Steps 41
Figure 6.3: Sample Results at Passing Bitwidths 43
Figure 6.4: Sample Results at Failing Bitwidths 43

LIST OF TABLES

Table 2.1: Initial Quantization Structure 8

Table 3.1: Curse of Dimensionality 12

Table 4.1: Whole Set Permutation Testing (Gaussian) 24
Table 4.2: Kernel Permutation Testing Results 24
Table 4.3: FWER vs Conventional 29

Table 5.1: Optimal Quantization Structure 35

Table 6.1: Bitwidth Results for 1000 Epochs 42

ACKNOWLEDGEMENTS

Thank you Professor Kenneth Kreutz-Delgado, for helping make this project a great experience at every step of the way.

Thank you to Professors Piya Pal and Micheal Yip for graciously serving on my committee.

Thank you Srinjoy Das, for helping me in the process of refining and understanding all the math and code in such a long-reaching project.

Thank you Professor Alexander Cloninger, for facilitating an interdisciplinary approach in integrating statistics with machine learning in a new and innovative way.

Thank you to John Graham for setting up and managing all of the Calit2 infrastructure at UCSD, allowing for a much faster resolution of the many tests that had to be run.

A big thank you to my mother, Olga Potapova, and to my good friend, Kendrick Mausolf, for helping me revise my thesis.

Thank you to Alexandria Do, for being a great help in pushing through the tough spots of implementing a major project like this one.

Thank you to Ian Colbert for helping with the power and control measurements for this project as a whole.

Thank you to Chih-Yin Kan for setting up the infrastructure for this project.

And a thank you to everyone else at UCSD who helped me in preparation for the experience of writing a paper like this one.

VITA

2017	B.S. in Computer Engineering, University of California San Diego
2018	M.S. in Machine Learning and Data Science, University of California San Diego

ABSTRACT OF THE THESIS

A Statistical Investigation of Model Quality in Generative Systems

by

Alexander Potapov

Masters of Science in Electrical and Computer Engineering
(Machine Learning and Data Science)

University of California San Diego, 2018

Professor Ken Kreutz-Delgado, Chair

Machine Learning is a powerful tool for both processing and generating data. It has been demonstrated to be more efficient than humans at distinguishing hundreds of different types of images. Such classification and game-based metrics are easily quantifiable, making it simple to demonstrate improvements in efficiency and quality over previous iterations. In contrast, Generative Models (GMs) that create synthetic samples by simulating a distribution are much harder to evaluate cleanly. When looking at images, an approach known as Maximum Mean Discrepancy (MMD) has recently become quite popular, as it

can non-parametrically compare samples and assign a similarity score between their relative distributions [GBR+12]. A major flaw that MMD has is that there is no accepted approach for defining a score that would deem the generated images as similar enough to real images for them to pass an independent analysis. Introducing humans has been a stopgap solution, but this adds a subjective element to the process workflow. An ideal solution would wholly remove the human element and determine an appropriate MMD-based quality target value using solely the data provided. In this thesis, we present a solution for this situation, as we introduce a novel statistical test that can more accurately compare distributions of data and determine a target score using solely the sample sets. By inspecting the quality of this solution on various models, we can train and analyze models that perform at sufficiently good levels via a fully automated procedure.

Chapter 1

Introduction

Generative Models refer to models of probability distributions that are now being trained and used to describe the distributional behavior of very high-dimensional data. They have an immense amount of potential, especially with the revolutionary amount of data ("big data") now available to train them with. Nevertheless, they have failed to become as mainstream as their counterparts, the Discriminative Models.

Discriminative Models have advantages in their relative ease of training, comparison, and use. They can be trained by using labeled data and running it through a feed-forward deterministic Neural Network. Their loss value is directly related to how well they perform, giving a simple way to compare otherwise disparate networks. Furthermore, they can also be compared through relative accuracy and false positive rates. Using them is as simple as just running an input image and receiving a one dimensional output declaring the label.

In contrast, Generative Models (GMs) are implemented on stochastic networks, have a more complex training process, and are more difficult to statistically compare. Training a

GM involves running large quantities of unlabeled data through a network and depending on the network to to statistically encode the ensemble patterns of randomly occurring in the data, as opposed to using predefined, labelled training patterns. To statistically encode these patterns, training a Generative Model requires significantly more data than training a discriminative model, since the patterns have to be found naturally by processing large quantities of training data. Even once we have trained a Generative Model, comparing it to another Generative Model is not as simple as comparing outputs directly. Since there is no obvious right or wrong generated image, comparing the quality of two models, or even of one model against itself becomes a troubling proposition. Devising an efficient and powerful method of comparison is essential for Generative Models to become more feasible for common use.

Since Generative Models are inherently tricky to work with, we choose to focus on the relatively tractable family of Restricted Boltzmann Machines (RBMs) in this thesis. They have a reasonably robust operating flow and can give results across a spectrum of possibilities. By using their strengths in simplicity, we can develop a scheme that will be easily extensible to more complex Deep Belief Networks (DBNs) and subsequently to the rich and interesting class of Generative Adversarial Networks (GANs) [RS08]. To make sure that the work is as extensible as possible, we have chosen to use the MNIST dataset, composed of hand-drawn digits, to ensure that there are a variety of labels and image styles [Den12]. This seemingly simple and yet fundamentally complex dataset acts as a very useful benchmark for the tests that we design. To be able to understand the interaction between model quality and its score in various metrics, we also chose to introduce bitwidth

of the numerical parameters in the model as our independent variable [WBSG16]. This systematic approach allows us to properly devise tests and determine the qualities and problems of the samples that we generate.

A Generative Model is a system that seeks to emulate a probability distribution. When we compare class-dependent distributions for classification purposes, we often seek to design hyperplanes that cleanly separate the data generated by one distribution from another. However, as we scale the complexity of the distribution, even with something as simple as MNIST, we are faced with a 784-dimensional array with many locality based interactions [Den12]. Separating such a pair would require inspecting an intractable amount of possible solutions, forcing us to adopt alternative approaches. Thus, this thesis will focus primarily on the Maximum Mean Discrepancy (MMD) method, with a cursory overview of the alternative Annealed Importance Sampling (AIS) method, where each of these methods indirectly compare probability distributions by avoiding absolute solutions.

When operating a non-parametric test akin to the Kolmogorov-Smirnov test, the primary result is a 1-dimensional score that clearly declares the goodness of fit between the true distribution and the simulated distribution. However, when performing generative tasks through probability distribution framework, these direct distance metrics prove insufficient to properly determine efficacy. In the immediate context, using MMD to achieve a score is insufficient to receive a meaningful result as to the categorical effectiveness of the image generation framework. We need to perform a higher level statistical analysis in order to get a proper score of the quality of these images and their passability relative to the real images provided for training. Even if the MMD score is low, if the images are clearly

distinguishable from one another, then the results are pointless and do not fulfill their purpose. By implementing a Permutation Test system, we can directly compare two sets of samples and determine the likelihood that they were drawn from different distributions [KR]. By analyzing these results, we can determine proper thresholds for the generated images insofar as to their ordinal quality relative to the real images.

Using a combination of these methods, we may be able to develop a workflow that determines the proper model quality needed to produce passable samples in a generative framework. Once we have such a workflow, it will be possible to extend and modify it to work on more complex datasets and models. Effectively, it will be possible to directly compare the quality of these models amongst each other and maintain a relative comparison basis that can compete with the systems in place for comparing Discriminative Models.

After a general overview of Restricted Boltzmann Machines (RBMs) and the design decisions inherent to them in Chapter 2, an explanation of the methodology and purpose of Maximum Mean Discrepancy (MMD) is provided in Chapter 3. That will be followed by an overview of Permutation Testing and the tradeoffs it entails in Chapter 4. Chapter 5 will contain an explanation of the use of Annealed Importance Sampling (AIS) for determining the truly optimal quantization structure of each model. Subsequently, we will then demonstrate a generalized workflow design, as outlined and described in Chapter 6. Finally, we will see a review of the Haar Perceptual Similarity Index (HaarPSI) as an other potential pathway forward in Chapter 7, alongside a more general and unified conclusion of the results previously described.

Chapter 2

Generative Models

When looking at the most common and systematic Machine Learning approaches, Discriminative Neural Networks are currently at the forefront of collective thought in the Machine Learning community. They are powerful enough to distinguish subtle changes in millions of images as seen in the annual ImageNet Competition [DDS+09]. These networks are simultaneously capable of massive throughput that outstrips conventional classification methods such as manual feature selection and human based sample gathering. Generative Models, even Generative Neural Networks, are frequently relegated to a secondary position due to their relative complexity in training, analysis, and operation. Generative Models are designed to generate samples from n-dimensional distributions that are inherently unknowable due to their natural complexity. Through stochastic generation of hopefully similar approximations of the samples that it was presented during training, the model simulates the process of creation and seeks to distributionally sample from otherwise intractably complex dimensions. Due to the high-dimensionality of even simple images,

(n=784 for the MNIST dataset), the sampling range of the distribution is immense and cannot be properly quantified using a conventional sum of Gaussians approach. Since these samplings are difficult to simulate and understand, we chose to focus on a relatively simple system that would allow us to more carefully manipulate the hyperparameters inherent to the problem.

2.1 Restricted Boltzmann Machines

When looking at common neural networks, the conventional approach is now a relatively architecturally-complicated "black-box" multi-layer convolutional neural network. These are advantageous with how straightforward they are to train and operate. They have a major downside in internal complexity, as they are difficult to perform minor manipulations on in order to derive useful internal metrics that could facilitate optimization and space savings. Restricted Boltzmann Machines (RBMs) are a style of generative stochastic neural networks that are optimized to simulate distributions while not having overcomplicated internal parts [RS08]. These features make them ideal for the purpose of performance-quality analysis. RBMs are effectively composed of a bipartite graph of neurons with a visible image layer and a hidden layer. When appropriately initialized to a highly corrupted version of an image, over the course of a set number of iterations, a weight network between the two layers oscillates values and generates the original image, either denoising it or completing an unfinished or occluded variant. Alternatively, an RBM can be used to generate an entirely novel image. In all such scenarios, the final generated image

is drawn from a probability distribution over images that has been encoded into the RBM via training over representative image samples. In order to further optimize the internal characteristics of the system for purposes of clearer performance analysis in our research, we mandate that the input image has to be wholly binarized, allowing for a cleaner shift from unfinished to finished image. All the initial work that is done in this thesis focuses on a Generative RBM trained on the first 5000 images of the MNIST database. This relatively small sample size allows for a better analysis of the effects of model quality on results.

2.2 Deep Belief Networks

While Restricted Boltzmann Machines have a single layer of internal neurons, Deep Belief Networks (DBNs) are a more complex version of the same basic idea [RS08]. Deep Belief Networks consist of a single input layer with any number of hidden layers stacked to form a more intricate and powerful generative system. These networks are inherently more effective at simulating more complex distributions. Fundamentally, DBNs are quite similar to RBMs. Thusly, results that are derived for RBMs will also translate to DBNs.

2.3 Quantization of the Model

In order to have a greater nuance in the effects of model quality on result quality, the model parameters are quantized using a fixed bit structure [Con]. This quantization intrinsically results in lower power and memory usage at lower bitwidths due to the simpler

calculations. Thus, quantizing these parameters also leads to an understanding of the power-generative capacity tradeoff in these GMs. Each model initially experimented upon was quantized at six different levels: 4-bits, 8-bits, 12-bits, 16-bits, 32-bits, and 64-bits. These initial quantization levels were chosen in an iterative manner in previous studies [yK18]. Their structure is simple with three factors for each bitwidth: the sign bit; n the number of bits before the decimal; and m, the number of bits after the decimal. Thus, each free parameter of a model is quantized to a signed $n.m$ floating point number.

Table 2.1: Initial Quantization Structure

Bitwidth	Sign-bit	n-bits	m-bits
4-bits	1	2	1
8-bits	1	3	4
12-bits	1	7	4
16-bits	1	7	8
32-bits	1	15	16
64-bits	1	55	8

When looking at each of these bitwidths (Table 2.1), the true decimal range can be determined by executing the simple operation $[-(2^n), 2^n + (1 - 0.5^m)]$ [Con]. 2^n is thus declared as the range of the model. Accordingly, 0.5^m is known as the resolution of the model. By understanding the limits placed on the model by these two factors, each model will have its own optimal bit scheme for more efficient and powerful operation.

Furthermore, by using this quantization scheme, it is possible to clearly demonstrate the relationship between model performance quality and parameter quantization (and thereby potential energy efficiency of hardware implementation).

Chapter 3

Maximum Mean Discrepancy

When comparing two sets of samples, it is important to have a powerful and efficient test that can give a score that measures the similarity between them. While various norms are a useful conventional approach to comparing images, they only ordinally declare how identical two images or samples are. The goal of a generative system is to simulate a distribution, not to identically copy a single sample and achieve one-to-one parity. Furthermore, mean-squared error only works when comparing images that are intentionally supposed to be similar. Since there are many different image sample realizations than can be drawn from the same distribution, it is not optimal to have an image-to-image comparison. Instead, it is important to compare the distribution of the images as opposed to actual images themselves. Maximum Mean Discrepancy is a potential solution to these problems.

3.1 Problem Definition: What is the test to do?

Suppose that we have two distributions, p and q. Given these two distributions, we can draw m iid (independent and identically distributed) samples from each distribution and label these sets as X and Y, respectively. We now have a two sets of samples that can be compared using different methods. Maximum Mean Discrepancy is a test designed to utilize these two sets to provide a score of the similarity of these two distributions. Thus, the problem that MMD solves can be described as follows:

$$\text{Given } X := \{x_1, ..., x_n\} \sim p$$

$$Y := \{y_1, ..., y_m\} \sim q$$

$$\text{Test Whether } p = q$$

From a simple, indirect basis, the solution is defined as such:

1. Estimate $\hat{\text{p}}$ and $\hat{\text{q}}$ from the obervations.

2. Compute the distance between $\hat{\text{p}}$ and $\hat{\text{q}}$.

This approach is conventionally done using something similar to a mixture of gaussians for the first step and a L_2 norm or KL divergence for the second. Usually, these tools are sufficient, as reported previously in designing estimators and distance measures [Das99] . Nevertheless, there are major problems when using these approaches; the two biggest ones are the dimensionality of the individual samples and the sampling bias of the iid sampling

technique. Dimensionality is a major problem when attempting to design estimators for images due to their inherently large size. Even a simple image like those in MNIST is 784 dimensional, so even after a binarization procedure, there are still 2^{784} possible images, making a naive hyperplane division computationally impossible.

Table 3.1: Curse of Dimensionality

Dimensions	Possibilities	Time to Generate at 10^9 samples/second
1	2	2 ns (nanoseconds)
8	256	0.256 μs (microseconds)
32	4,294,967,296	4.295 seconds
64	$\approx 1.8447 * 10^{19}$	584.6 average Gregorian years
Age of the Universe		$\approx 1.4 * 10^{10}$ average Gregorian years
128	$\approx 3.4028 * 10^{38}$	$1.078 * 10^{22}$ average Gregorian years

Thus, due to the dimensionality of even simple images, as seen in Table 3.1, it is unreasonable to directly estimate their distributions. Furthermore, this dimensionality leads to an immense amount of sampling bias even within the sample space available in the MNIST database [Den12]. Correcting for this bias and dimensionality requires a different approach that is more robust to the changes and randomness inherent to real-world samples.

All of these problems sum up to a conclusion that solutions focused on using density estimation are ineffective when attempting to solve such problems. No matter how many

samples are available, it will require a infeasible amount of effort and processing capabilities to create the estimates and derive a proper solution.

3.2 A Direct Alternative

Using a statistical perspective, an alternative approach may present itself. Instead of utilizing a density estimator, it is possible to bypass it and use the mean differences directly between samples [GBR+12]. By suggesting a metric such as

$$D(p, q, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbf{E}_p(f(x)) - \mathbf{E}_q(f(y))$$

subject to the condition that

$$D(p, q, \mathcal{F}) = 0 \iff p = q$$

but only when

$$\mathcal{F} = C^0(\mathcal{X})$$

is the space of continuous bounded functions on $\mathcal{X}$, since the function needs to have a valid output for any possible sample from $\mathcal{X}$, as $\mathcal{X}$ represents the arbitrary sample space which we are working within. By designing this metric, we effectively declare that it will only equal zero when two distributions are identical. Thus, this function will indicate that two distributions trend towards being identical as it approaches zero. It further declares that it can work for any two distributions that have the same sample dimensionality, since they need to have samples that are directly comparable using a function $\mathcal{F}$ [GBR+12]. In

order to effectively utilize this suggestion, it is necessary to define

$$C^0(\mathcal{X}) = \{f | ||f||_{\mathcal{H}} \leq 1\}$$

which is a unit ball in a reproducing Hilbert Kernel Space, as long as $\mathcal{H}$ is universal [GBR+12]. Effectively, this approach allows for transforming the real distribution into a Hilbert Kernel Space and then operating upon it within that assumption. Once the comparison is performed, it can be returned to a regular space with relevant results. Thus, the initial steps of the derivation for Maximum Mean Discrepancy are:

$$\begin{aligned} D(p, q, C^0(\mathcal{X})) &= \sup_{||f|| \leq 1} \mathbf{E}_p - \mathbf{E}_q \\ \sup_{||f|| \leq 1} \mathbf{E}_p - \mathbf{E}_q &= \sup_{||f|| \leq 1} \langle \mu_p - \mu_q, f \rangle \\ &= ||\mu_p - \mu_q||_{\mathcal{H}} \end{aligned}$$

At this point, it is necessary to introduce statistical kernels, which are effectively probability density functions with their normalization component removed. By using these kernel functions to map high-dimensional data to a single-dimensional output, it is more possible to operate on the specific comparisons. Subsequently, the next steps are to determine the value of $\|\mu_p - \mu_q\|_{\mathcal{H}}$, which for $\mathcal{H}$ being a RHKS takes the form [GBR+12]:

$$\begin{aligned} ||\mu_p - \mu_q||^2_{\mathcal{H}} &= \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle \\ &= \mathbf{E}_{p,p} \langle k(x, \bullet), k(x', \bullet) \rangle - 2\mathbf{E}_{p,q} \langle k(x, \bullet), k(y, \bullet) \rangle + \mathbf{E}_{q,q} \langle k(y, \bullet), k(y', \bullet) \rangle \\ &= \mathbf{E}_{p,p} k(x, x') - 2\mathbf{E}_{p,q} k(x, y) + \mathbf{E}_{q,q} k(y, y') \end{aligned}$$

Given the depth of the previous explanation, when actually implementing the test, the algorithm itself is relatively simple to describe. The kernel that is used is a simple gaussian kernel, defined as:

$$k(x, x') = e^{-\frac{(x-x')^2}{2\sigma^2}}$$

Where x and x' are two samples being compared and σ is a normalization constant that is also known as the bandwidth of the kernel. This kernel function will be run on each pair between and within the two sets of samples. This operation will result in a large $(m+n)$ by $(m+n)$ matrix where each index holds the result of the kernel function on a respective pair of samples. Each of these results is understood as a score of the similarity of the two relevant samples. Upon constructing this matrix, it is possible to determine the Mean Discrepancy within each set of data by getting the mean of each quadrant. Thus, the Mean Discrepancy between X and X' would be located in the top left quadrant. To determine the Maximum Mean Discrepancy between X and Y, we then add the two self comparison quadrants and subtract the two cross comparison quadrants. As the two distributions become more similar, the cross comparisons will become closer to the self comparisons and the MMD score will trend towards 0. To this end, the actual mathematical operation being performed is:

$$\mathbf{MMD}_u[\mathcal{F}, X, Y]^2 = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j)$$

where x_i, y_i are the i-th elements of X and Y, respectively.

Within this formulation, the main other factor is the bandwidth of the kernel to be used.

In this project, while working with the MNIST database through an RBM framework, we found a bandwidth of 64 to be the optimal constant [SSL+17]. To have an efficient and powerful MMD metric, we chose to use the Shogun machine learning toolbox, which has implemented a quadratic time C++ version of the MMD operation [SSL+17].

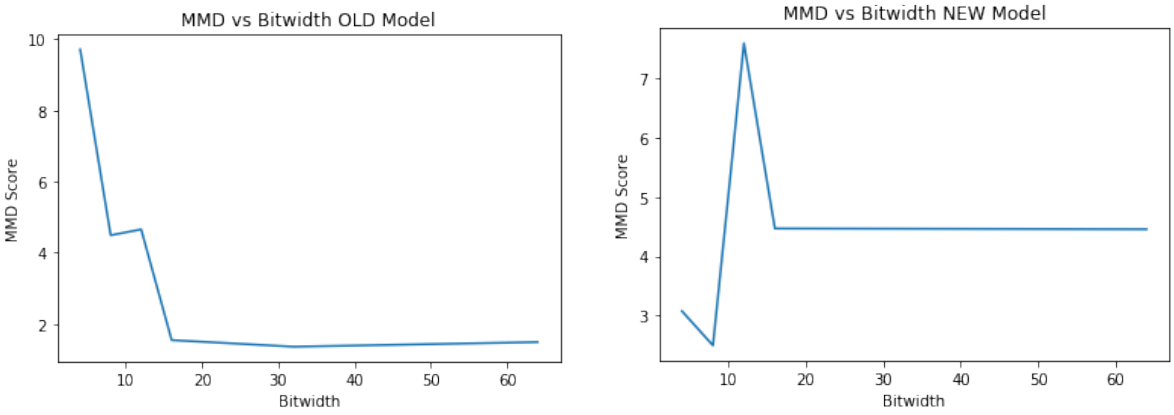
Effectively, it is possible to use this direct calculation of the MMD as a score to quantify the likelihood that two specific sets of data were drawn from the same distribution. Therefore, an early step in analyzing the tradeoffs between was performing this conventional MMD test on the generated samples across all of the quantized bitwidths.

3.3 Results as Achieved by Baseline MMD

When executing MMD in the predefined manner, the relevant result is the score received. Thus, a good early trial to compare the effectiveness of various models is to directly compare the scores received. The major problem with this approach is that there are no ordinal scores of good versus bad. When looking at two models with different scores, all that can be said is that one model provides a better closeness score over the other given a specific kernel. This score is purely relative and only applies for the specific test data, quantity of generated data, and bandwidth used in the given experiment. In spite of this limitation, it is still quite useful in this respect, as it can effectively compare the generated distributions against the approximated real image distribution.

To set up for this operation, 1,000 images were generated from each of two models, one trained for 100 epochs and the other trained for 1,000 epochs. These models were each

composed of 441 hidden neurons and were trained using the masked images at 100 generative steps [yK18]. These generations were performed for each bitwidth previously declared (4,8,12,16,32,64). Upon completing the generation, MMD was performed comparing the generated images against the test sub-dataset of 1,000 images from MNIST.



(a) 100 Epoch Trained Model

(b) 1000 Epoch Trained Model

Figure 3.1: Results of Running MMD on the Primary Models

When looking at the results from this operation in Figure 3.1, it is possible to see an obvious pattern in the 100 epoch model. There is a clear inverse correlation between MMD score and the bitwidth, indicating that as the model quality increases, the closeness of the two distributions increases as well. In contrast, the 1000 epoch model is distinctly overtrained, as there are no trends in the MMD changes across all of the bitwidths, suggesting that the model is overfitted to the training data. However, there are no conclusions regarding the quality of the model in comparison to the real distribution, as the MMD score cannot be used as a definitive cutoff of quality alone.

3.4 Expanding on MMD

After performing this initial MMD operation, it is evident that it is a potentially suitable method for comparing models and determining their relative qualities. To be able to construct a more deterministic system for comparing models and simultaneously determining whether they are of sufficient quality will require a more powerful system that can achieve both a score of comparison for model against model and a probability value for comparing the distribution itself against the real data. When looking at suitable tests, the permutation test presents itself as a valid and powerful solution [ET94].

Chapter 4

Permutation Testing

When looking at p-testing, there is a necessity for powerful, robust, and efficient tests that can indicate whether two sets of data are from the same distribution or at least from two indistinguishable distributions. The Permutation Test fulfills all of these requirements [KR]. It is able to integrate with any valid metric. It is relatively efficient, as it simply requires the baseline metric to be run a sufficient amount of times. It is powerful, as it is able to give a proper p-value that can distinguish from two distributions based solely upon the prescribed metric.

4.1 Defining the Permutation Test

The Permutation Test is part of a group of nonparametric statistics that are able to compare sampling distributions through a test statistic [KR]. This nonparametric quality is advantageous, as it allows the test to be executed using data sampled from a distribution

instead of the distributions themselves. When looking to define the test, we first need to declare the initial conditions and the relevant null hypothesis.

$$F \to \mathbf{z} = \{z_1, \ldots, z_n\}$$

$$G \to \mathbf{y} = \{y_1, \ldots, y_m\}$$

$$H_0 : F = G.$$

When looking at these definitions, we understand F and G to be the two distributions, real and generative, that are being compared. Consequently, $\mathbf{z}$ and $\mathbf{y}$ are the sampled images from each distribution, effectively the real and generated images themselves. Following, the null hypothesis is that F and G are the same distribution, subsequently putting the burden on the test to disprove that claim and distinctly declare that the real and synthetic images are drawn from sufficiently different distributions.

When performing a permutation test, there needs to be a baseline result to compare the permutations against. Thus, the next step is setting up the initial metric which will operate as a δ. This δ will be used as the baseline test statistic result against which all the other test statistic results will be compared. The MMD of the $\mathbf{z}$ and $\mathbf{y}$ sets, having been previously computed, is declared to be this δ. At this point, the permutation component of the permutation test begins. Initially, the two groups of samples are pooled together into $\mathbf{x}$, a set of all the available samples. Subsequently, $\mathbf{z}^*$ and $\mathbf{y}^*$ are randomly sampled from $\mathbf{x}$ without replacement, while also maintaining that $\mathbf{z}^*$ and $\mathbf{y}^*$ have the same number of samples as $\mathbf{z}$ and $\mathbf{y}$, respectively. This random sampling results in two new sets of samples

that should represent a similar MMD score if they are from the same or a sufficiently similar distribution.

After completing the random sampling, the MMD metric is determined for the resampled datasets $\mathbf{z}^*$ and $\mathbf{y}^*$ to receive a similarity score between the randomly sampled sets. This similarity score is directly relevant to the original δ and and is stored. Since we know that this metric indicates the similarity of the underlying distributions of the two sets of samples, if the resampled versions of the datasets achieve a significantly different score (in either direction) than the original datasets, there is a strong suggestion that the original datasets are less close than the metric may seem to suggest. This operation of mixing, sampling, and metric determining is repeated a series of times. In this project, the operation is repeated for N = 1,000 times.

$$\delta = \text{MMD}(\mathbf{z}, \mathbf{y})$$

$$\theta = [](\text{an empty array})$$

for N Iterations :

$$\mathbf{x}^* = \text{shuffle}([\mathbf{z}, \mathbf{y}])$$

$$\mathbf{z}^* = \mathbf{x}^*[1 : n]$$

$$\mathbf{y}^* = \mathbf{x}^*[-m :]$$

$$\theta_i = \text{MMD}(\mathbf{z}^*, \mathbf{y}^*)$$

Upon completing these permutation iterations, there is a set of N MMD scores that are all compared to the original δ score. To complete the permutation test, the number

of scores less than the value of δ are counted, indicating the number of permutations that are more similar than the two original sets. Since the original datasets are assumed to be drawn from the same distribution under the null hypothesis, if there is an abundance of resampled datasets that are more similar than the original datasets, there is an indication that similarities within the datasets may be closer than across them. This number is divided by N to receive a ratio of more similar permutations, which is subsequently subtracted from 1 to receive a p-value.

$$\text{estimates} = \text{N MMD Trials}$$

$$\text{count} = \text{estimates} \leq \delta$$

$$p = 1 - \text{count}/N$$

This p-value can then be compared to a standard significance value of 0.05 to compare the two distributions in closeness. If the p-value is less than 0.05, H_0 is rejected and the implication is that the two distributions are so different that permutations of the two sample sets are closer to each other than the two original sets in an overwhelming number of combinations. Otherwise, if the null hypothesis is not rejected, the implication is that the two distributions are sufficiently close that there is no strong indication that the two distributions are distinguishably far apart [ET94].

4.2 Executing Permutation Testing

When looking at the distributions inherent to MNIST, there are two relevant approaches. The distribution of hand-written digits can be treated as one big distribution, from which all numbers are drawn or as a collection of conditional distributions, one for each label, from which each example of that label is drawn. This duality allows for a two-pronged approach to the matter, comparing both the whole dataset and in contrast zooming in and focusing on these individual labels.

4.2.1 Initial Approach: Whole Set Testing

The first trials undertaken treated the whole set of generated images as being drawn from one distribution. These efforts focused on computing p-values for the various bitwidths and observing the effects that bitwidth and MMD score had on the p-values.

As evidenced by the p-values received for all of the bitwidths regardless of MMD score (see Table 4.1), approaching permutation testing by treating the generative distribution as one general distribution was not effective. Acknowledging these results, there was also an attempt to try different types of kernels from the basic gaussian kernel to see if there was any possibility to keep the approach as general as possible by treating all the images as being drawn from this same natural distribution.

While trying other kernels that were considered to be more inline with the human methodology of distinguishing images (see Table 4.2), the p-values remained at 0.00, further implying that while using MMD, it was not feasible to treat this generative system

Table 4.1: Whole Set Permutation Testing (Gaussian)

Bitwidth	MMD Score	P-value
4	≈ 9.71	0.0
8	≈ 4.49	0.0
12	≈ 4.65	0.0
16	≈ 1.54	0.0
32	≈ 1.36	0.0
64	≈ 1.49	0.0
Passing	≈ 0.89	0.05

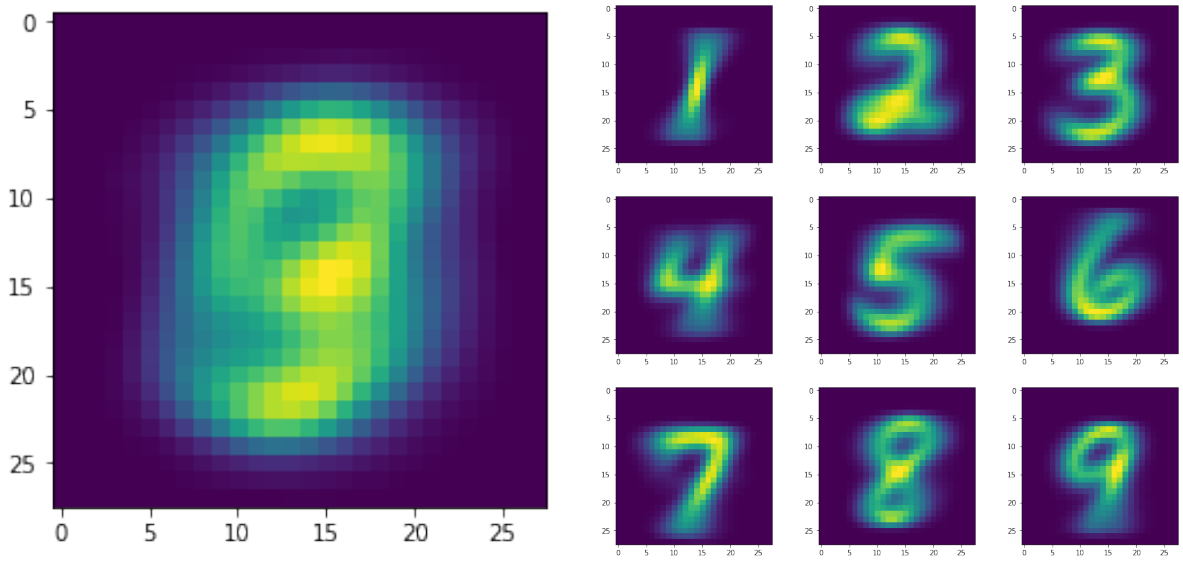
Table 4.2: Kernel Permutation Testing Results

Kernel Name	MMD Score	P-value	Computation Expense	Pass
Gaussian	≈ 1.36	0.0	Low	No
Gaussian Shift	≈ 3.03	0.0	Medium	No
Chi Squared	≈ 2.36	0.0	Low	No
CANOVA	$\approx 3.4 * 10^{22}$	0.0	Very High	No

as one distribution.

4.2.2 Second Approach: Label-Wise Testing

By treating the natural distribution as a mixture of distributions rather than a single over-arching distribution, it is possible to divide the original problem into ten sub-problems, one for each digit. This approach derives from understanding that MNIST is not just a set of images, but rather a set of sets of images [Den12]. With this logical step, it is possible to devise a new system for understanding how similar the real and generated distributions really are.



(a) Mean of All Digits

(b) Mean of Each Digit

Figure 4.1: Mean Image Comparison

By observing the similarities within each digit (see Figure 4.1), in contrast to the relatively messy mean of all digits, it is reasonable to assume that these generated images are not drawn from one overarching distribution, but rather from a class-conditional distribution for each digit. From the per-class distributions $P(\text{image}|\text{class})$ you can find the

"overarching distribution" by marginalization $P(\text{image}) = \sum_c P(\text{image}|\text{class} = c)$. This mixture of complex distributions can thus be better compared by looking at the quality of each label as opposed to the system as a whole.

When looking at complex distributions, one common modeling technique is to decompose it into a mixture of Gaussians [Das99]. The mixture of Gaussians allows seemingly intractable distribution to be modeled in a simple and effective manner. However, when looking at real-world distributions, they are even more complex and thus need a more nuanced representation than as a mixture of Gaussians, let alone as a single mass distribution.

To properly evaluate this approach, it was necessary to make sure that all the labels of the generated and real data were known. RBMs have the advantage of allowing for a front end encoder to be added to the visible layer, thereby, allowing the output's label to be declared in the generative step. Using this functionality, the generated images were divided into ten subsets of approximately 100 generated images. The test dataset was also split along the same lines. After this division, the permutation test was evaluated on each subset.

When looking at the label-wise split (see Figure 4.2), it is evident that there is a crossing point that shows the closeness of the real and generated images. There is a consistent correlation in the MMD and the p-values, further implying that it is possible to systematically evaluate the absolute quality of the generative model. Once the null hypothesis is not rejected, we can conclude that the distributions of the real and generated images for each label are sufficiently close that the computerized MMD algorithm cannot

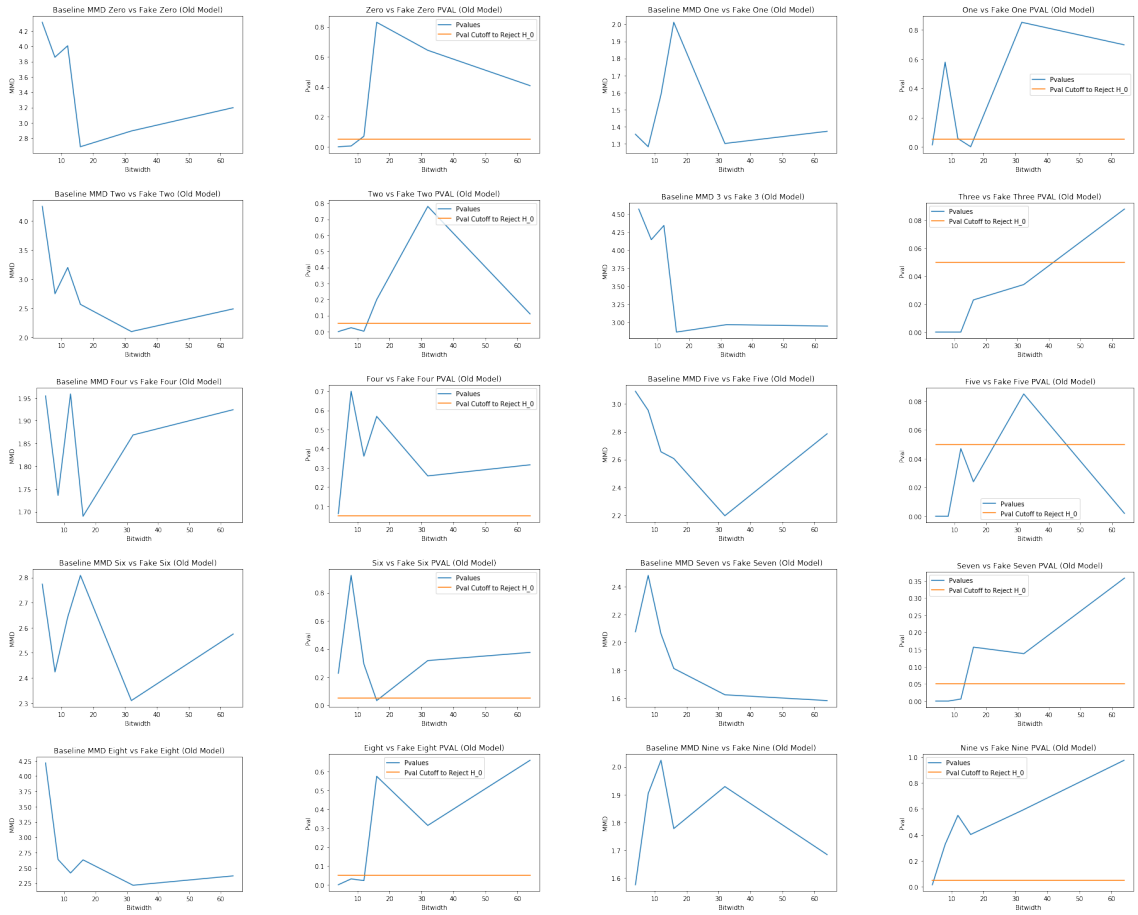


Figure 4.2: MMD and PVal of Each Label

distinguish them.

Furthermore, when retaking the average of all of the labels ad combining their relative results on one graph, there is a clear correlation between their MMD scores and their p-values as seen in Figure 4.3. The R-Squared value for a linear regression between the MMD and the p-values is 0.919. When looking at the prior graphs for each label, there is the problem of insufficient data (about 100 generated images for each label), while the total number generated was 1000 images. By increasing the number of datapoints in future trials, the graphs should be much clearer and have significantly less fluctuation.

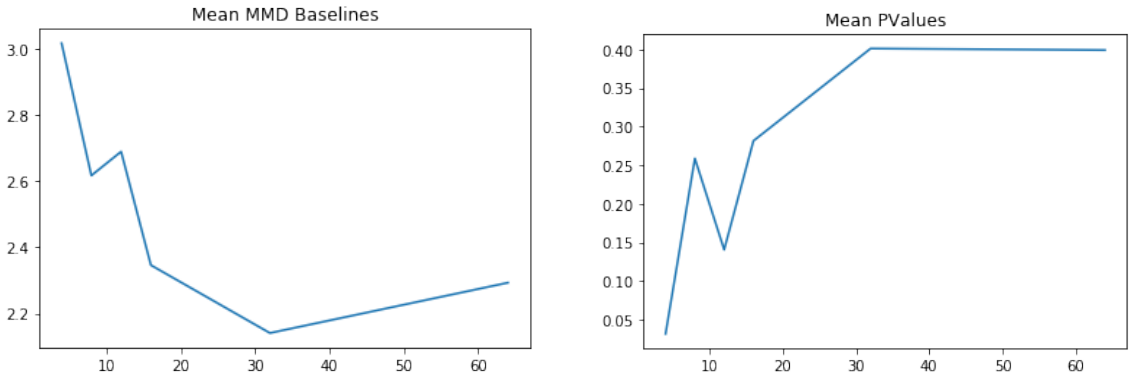


Figure 4.3: Mean MMD and P-Value of all Labels

4.2.3 Multiple Hypothesis Testing

Regardless of these initial results using multiple labels, well-known pitfalls cause a need for the statistical approach to be improved to account for the negatives of using multiple hypotheses to make a single claim [Sch15]. To combat this overzealous acceptance or rejection of null hypotheses, it may be necessary to implement a form of Multiple Hypothesis Testing. The proposed solution is the Family-Wise Error Rate.

Family-Wise Error Rate (FWER)

In order to integrate all of these hypotheses, it is necessary to effectively treat them as a family of tests. The objective of FWER is to minimize the number of false rejections by adjusting the confidence interval based on the number of trials being performed [LR05]. Effectively, this entails using Bonferroni's Method to only reject the i-th class null hypothesis $H_{0,i}$ if $p_i \leq \alpha/N$ where α is the initial confidence level and N is the number of trials being performed [Bon36]. What this does to the trials performed prior is change the

confidence level from 0.05 to 0.005 for the same fundamental effect. Under this methodology, if any of the null hypotheses are rejected, it is also necessary to reject the general null hypothesis regarding the set as a whole.

Potential Negatives of the Use of FWER

Since Multiple Hypothesis Testing is inherently conservative to rejecting the null hypothesis through its generous allowances in confidence, it is more likely to let through a given model, as fewer trials need to pass the relative similarity threshold. This makes it easier for a given model to pass general testing. In practice this proved to occasionally be relevant, as models might get a p-value of 0.01 for a specific digit, failing to reject the null hypothesis under FWER, but rejecting it without it.

Evaluations with FWER

Table 4.3: FWER vs Conventional

Bitwidth	4	8	12	16	32	64
FWER	Fail	Fail	Fail	**Pass**	Pass	Pass
Conventional	Fail	Fail	Fail	**Fail**	Pass	Pass

The only difference when performing these initial trials was that FWER made it slightly easier to pass through a model (see Table 4.3). Effectively, the main purpose of multiple hypothesis testing in such an environment would be to make sure that the conclusions

are not overly pessimistic. Nonetheless, if each label is treated as its own independent distribution, the conventional approach proves more useful, as it is more likely to catch a model that produces samples of an insufficient quality, minimizing type one errors, while maximizing type two errors. Type two errors are more favorable in this situation, as if an insufficiently high quality model made it through testing, it would be easy to detect in practice.

4.3 Conclusions of Permutation Testing

Permutation Testing allowed for a statistical and objective analysis of the quality of the models investigated in this project. A general overview of the model's quality is not feasible using the conventional kernels available at this time, indicating that a label-wise approach is more powerful and simultaneously more efficient. By comparing real and generated examples of each type of label available, it is possible to make a systematic conclusion as to the quality of the model at hand. After evaluating whether a model is capable of passing permutation testing, running the same trial on each possible bitwidth allows for an understanding of the minimum viable model quality for generation of sufficiently high fidelity images that can pass a systematic computerized evaluation.

Thus, by integrating permutation and maximum mean discrepancy, it is possible to systematically determine whether a model is of sufficiently quality to pass a censor while removing any human element from the system. This functionality enables an expansion of understanding in the categorical quality of a model, rather than an abstract relative

quality in human observed MMD and other arbitrarily chosen cutoff metrics.

Chapter 5

Annealed Importance Sampling For Quantization Selection

While looking for a metric that is more deterministic than MMD, Annealed Importance Sampling (AIS) presents itself as a viable alternative [RS08]. Fundamentally, AIS uses the weights of an RBM in tandem with a dataset to eliminate the need to generate synthetic data in order to produce a likelihood score of generating the given samples using a given set of weights [Kri]. In this project's case, the samples would be the MNIST dataset and the weights would be that of the trained model. The process behind AIS is defined more in depth in the Appendix. Here AIS can be used to deterministically evaluate the quality of a model by providing a likelihood score for the generation of data given certain weight values. To the degree that a set of weight values has a higher likelihood, it is a better model for the original distribution. In the framework of the workflow described afterwards, AIS was used solely for determining optimal bitwidths. Regardless, AIS has potential for

being an alternative metric to MMD for the trials described.

5.1 Integration with Permutation Testing

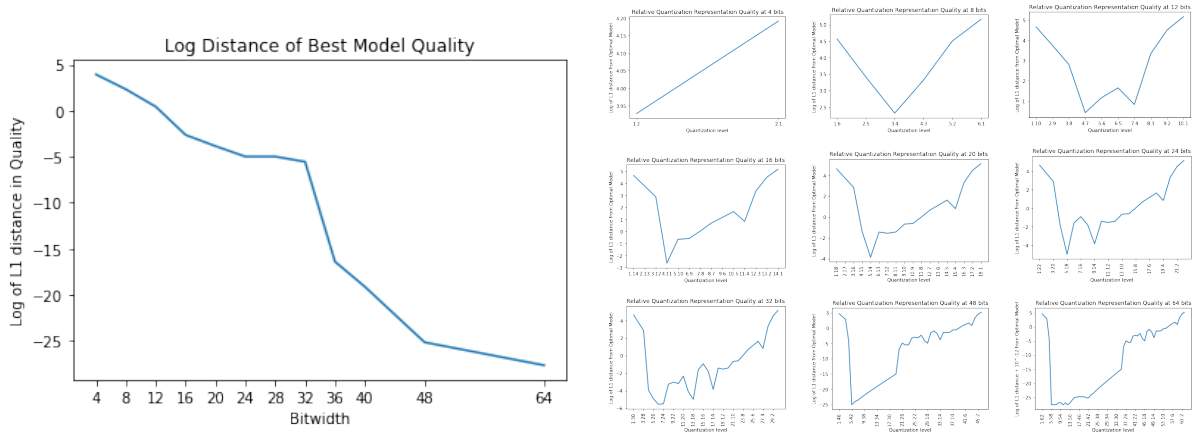
The major problem with AIS is that it is model based, while permutation testing is designed for a parametric evaluation based on sets of a samples [KR]. AIS only takes in one set of samples and outputs a likelihood score, while permutation testing depends on allowing a metric to take in various permutations of the available samples and return a metric based on each pairing [RS08]. Fundamentally, there may be a way to reconcile this issue, but for now AIS is best suited for determining optimal bitwidths. Here is the proposed procedure:

1. Generate a Model using the Parameters of Choice
2. For Each Relevant Bitwidth:
    (a) For Each Possible Quantization
        i. Run AIS
        ii. Determine log-likelihood of Data
    (b) Take log-likelihood closest to Un-quantized Data
    (c) Use Related Quantization Scheme
3. Use each best-likely Quantization Scheme

5.2 AIS For Bitwidth Selection

Despite the limitations of AIS, we can demonstrate see that it is quite useful for selecting the optimal $n.m$ quantization at each bitwidth. Since it takes in a model and a set of data that the likelihood will be determined of, it is possible to determine which quantization for the model will result in the highest likelihood for the given bitwidth.

We observed a trend that with increased bitwidth, the likelihood of the optimal quantization becomes closer and closer to the original model, becoming nearly indistinguishable at the higher bitwidths (Fig. 5.1). The distance of the likelihood from the original model's likelihood for each tested bitwidth at each relevant quantization is shown in Fig 5.1b. As the distance drops, the quantized model provides a better approximation of the original model, while retaining the power savings provided by the lower bitwidth. The graph in Fig 5.1a was generated by taking the lowest distance for each bitiwdth, demonstrating a constant downwards trend in distance as bitwidth increased.



(a) Best Distance of Each Bitwidth

(b) Loglikelihood of Each Bitwidth

Figure 5.1: Quantization Optimization

From experimental data shown in Figure 5.1b, we can see that there is a quantization where the likelihood distance from the full model is minimized. This lowest quantization is also the optimal quantization for the given model and bitwidth, since that specific quantization provides the best approximation of the given model. These optimal bitwidths are shown in Table 5.1.

Table 5.1: Optimal Quantization Structure

Bitwidth	Sign-bit	n-bits	m-bits
4-bits	1	1	2
8-bits	1	3	4
12-bits	1	4	7
16-bits	1	4	11
24-bits	1	5	18
32-bits	1	6	25
48-bits	1	5	42
64-bits	1	5	58

By comparing Table 5.1 and Table 2.1, we can conclude that beyond 8 bits, the original bitwidth assumptions are extremely far from the AIS calculated optimal bitwidths. Thus, it is evident that the initial assumptions regarding quantization were not optimal. This initial bad quantization also directly leads to all of the models initially failing the permutation test for the whole set version. While iterating through the workflow described in the

following chapter, the models quantized using the above $n.m$ approaches performed much more effectively at the single set permutation test, making the label-wise permutation test unnecessary. Therefore, AIS proves most useful in a quantization context, but that is not to say that it is not potentially feasible in other contexts as well.

Chapter 6

Unification through Workflow

Upon completing the process of permutation testing, it became evident that there was the potential to unify the components of training and testing the generative model into one general workflow. This workflow would be able to account for every step of the process from collecting data to outputting a properly trained and quantized model for maximum generative effectiveness and power-wise efficiency. With this unified workflow, we aim to completely remove the human element beyond the collection and labelling of data in preparation for the training of the system.

6.1 Description

To create a workflow that results in a well trained and fitted model, it is necessary to avoid a few important pitfalls:

1. Excessive epochs of training results in overfitting;

2. Insufficient sample size results in noisy p-values and MMD scores;

3. Improper Quantization results in inefficient power consumption.

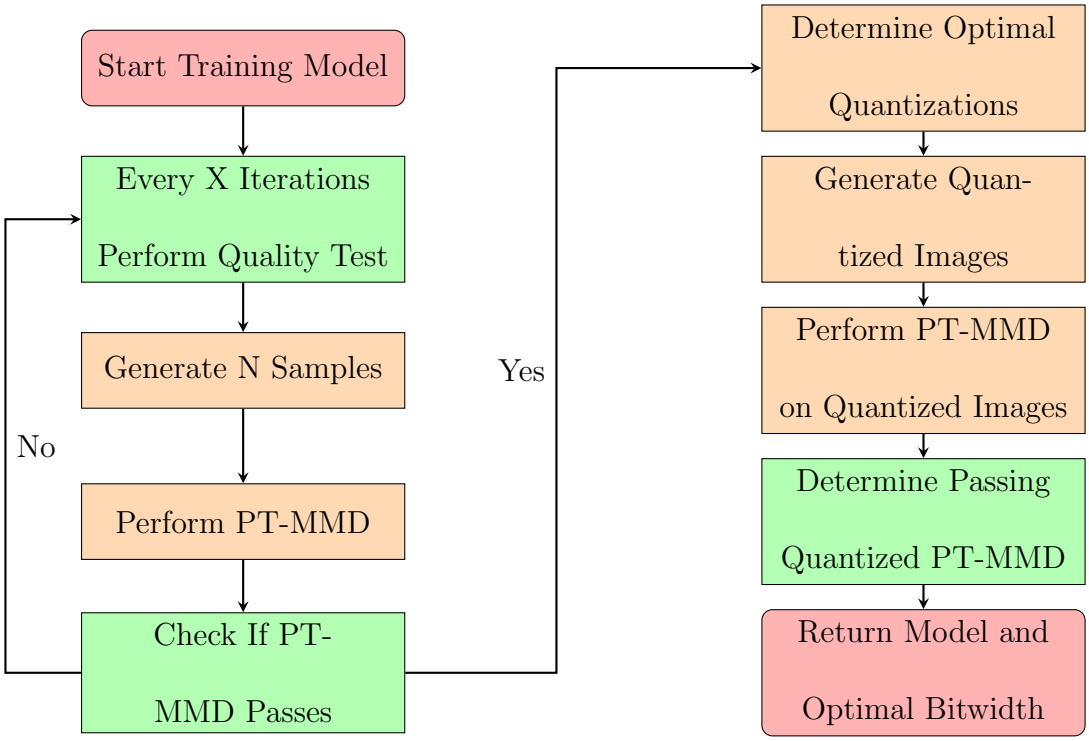


Figure 6.1: Flowchart of Work Process

Within the flowchart (Figure 6.1), red nodes are initial and terminal, green nodes involve some sort of decision, and orange nodes describe processes. By minimizing the number of epochs trained in the initial steps of the Work Process , it is possible to minimize the potential for overfitting. As soon as the model is deemed viable, it is advised to continue to the later stages, as excessive training is computationally wasteful and is likely to result in an overfitted weights matrix. Further, by tuning N, it is possible to avoid the second problem and minimize the noise in the MMD scores through having sufficient samples created. Since creating samples can prove computationally expensive, minimizing

training epochs benefits the computation cost. Finally, by performing an iterative procedure starting from a maximum bitwidth, it is possible to determine a minimally viable bitwidth in the fewest steps, while simultaneously developing a model that performs well at a low level of power use.

6.2 Operation

To properly execute the workflow, it is necessary to have a sufficiently large set of samples so they can be split into training, testing, and validation sets. After training the generative system on the training dataset for some predetermined number of epochs, it is appropriate to perform the permutation test to evaluate the quality of this model. First, it is necessary to generate a sufficient number of samples for each label in order to perform a permutation test between those generated samples and the test dataset. If all of the labels perform at a passing grade, the model is sufficiently trained and does not need further training. If any of the labels failed the permutation test, it is necessary to continue training the model.

Once there is a passing model, we can to continue to the next phase of testing, which is quantization analysis. Since this project uses a fixed point quantization approach, it is possible to determine the optimal bitwidth for any model using the AIS framework. Subsequently, it is required to test each bitwidth starting with the maximum tolerable bitwidth using the same permutation test methodology, but with the validation set in place of the test set. Once a bitwidth is reached which causes the permutation test to fail,

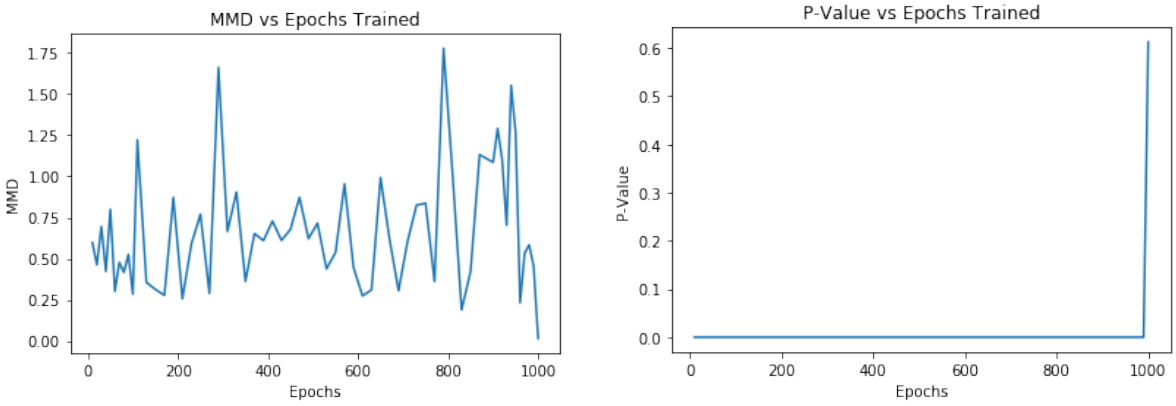
the minimum viable bitwidth is found and can be returned from the workflow. Thus, by systematically following this flowchart previously shown, we create a generative model that can pass a computerized censor and is efficient in both power consumption and training time.

6.3 Results

For validating this workflow, the MNIST dataset was used in the same RBM infrastructure as during the design process. The MNIST was decomposed into 50000 training images, 1000 testing images, and 1000 validation images. These numbers were chosen as they allowed for a balance between database scale and computation time. The epoch frequency for analysis was set to be 10. N was set to be 100 for each label. The maximum viable bitwidth was set to 64 bits, as that is a conventional upper limit for number quality.

During this trial, several key distinctions emerged from prior runs. Most notably, by implementing a proper methodology for determining quantization at each bitwidth, as described in Chapter 5, it was no longer necessary to perform the label-wise analysis. Even these simple models had sufficient distinctions in their quality to guarantee optimization through our proposed workflow.

Based on the results in Figure 6.2, we can conclude that solely using MMD as a score of quality does not result in good models and that by integrating permutation testing, it is possible to with better certainty determine when a model is sufficiently trained. Without utilizing PT-MMD, it may have been considered reasonable to cut off testing at some



(a) MMD Score over the Course of Training

(b) p-value over the Course of Training

Figure 6.2: Workflow Training Steps

earlier epoch based on the relatively low mmd scores, but by performing a permutation test, it is evident that this model would have failed a further statistical analysis. Thus, at the end of the first phase of our testing, it is obvious that this specific 1000 epoch model is the first model of sufficient quality to perform further analysis.

The next stage of analysis was used to determine the minimum viable bitwidth for the derived 1000 epoch model. First, samples were generated using the validation set at 4, 8, 12, 16, 24, 32, 48, and 64 bits wide. Each bitwidth had an empirically derived quantization scheme, as described in the previous chapter. After generating these samples, a permutation test was performed once more, in order to further determine which bitwidths were optimal for this specific model.

From results shown in Table 6.1, it is evident that having a higher bitwidth does not guarantee that the generated samples will be strictly better. Due to the interplay between overly precise models and overfitting, the best results were achieved at either full

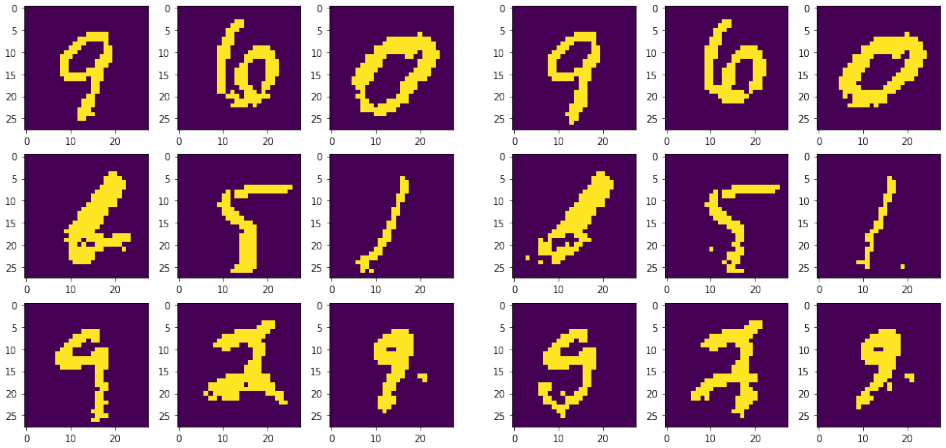
Table 6.1: Bitwidth Results for 1000 Epochs

Bitwidth	Delta	Pvalue	PT-MMD
64	0.01513	0.636	Pass
48	0.01516	0.588	Pass
32	0.01880	0.192	Pass
24	0.01655	0.436	Pass
16	0.08454	0.000	Fail
12	0.05643	0.000	Fail
8	1.14608	0.000	Fail
4	2.5226	0.000	Fail

bitwidth (64 bits) or at relatively lower bitwidths (48 and 24). 32 bits wide performs at a passing rate, but is worse than the lower 24 bit version. There are at least two reasonable conclusions that can be derived from this trial:

1. For optimal image quality, it would be best to use the 64 bitwidth model, as it provides the best approximation of the real distribution;

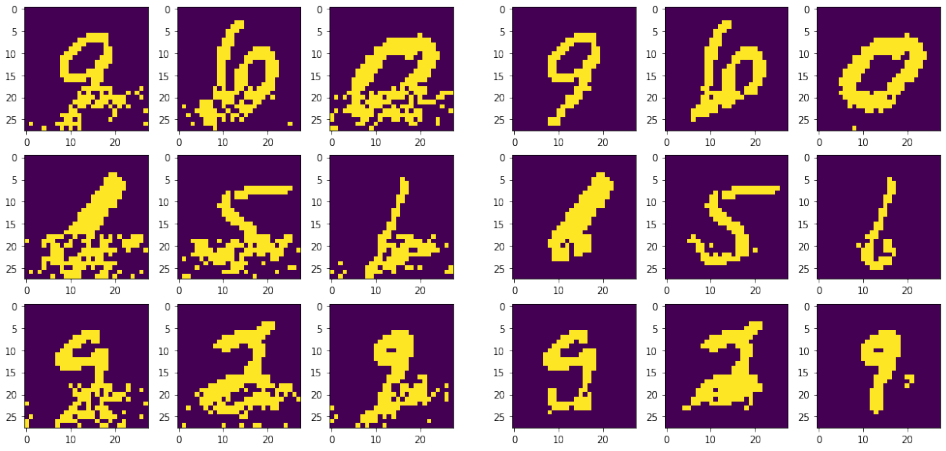
2. The 24 bit model will provide for optimal power consumption, as it is still of passing quality and consumes 2.666x less power than the 64 bit model in the course of generating its images, due to the linear relationship between power consumption and bitwidth [yK18].



(a) Generated Images at 24 Bits (b) Generated Images at 64 Bits

Figure 6.3: Sample Results at Passing Bitwidths

As shown in Figure 6.3, the images at 64 bits are of slightly higher quality than those generated at 24 bits, but both sets of images are passable. By potentially further implementing a smoothing function and accounting for the logical images, it is feasible to recognize these as real, hand-drawn digits.



(a) Generated Images at 4 Bits (b) Generated Images at 16 Bits

Figure 6.4: Sample Results at Failing Bitwidths

In contrast, in the images generated at 4 bits (Figure 6.4a), it is clear that these images are too noisy and of suboptimal quality to pass a human censor. Furthermore, even when generating at 16 bits (Figure 6.4b), there are sufficient numbers of irregular errors, that even with the high quality images, there are enough very low quality images, resulting in a failure of the whole image set.

6.4 Conclusions

Based on these results, we can draw the conclusion that it would be advisable to follow an optimized workflow presented in this thesis to achieve a model that can generate images of sufficient quality to pass the scrutiny of a human censor, while simultaneously minimizing the power output necessary for the generation of these images. For our specific case of the MNIST database, we found that 1000 epochs of training proved sufficient for a passing model. For that specific model, it was found that the minimum viable bitwidth was 24 bits wide. It was also determined, not surprisingly, that 64 bits produced viable images, yet the intermediate bitwidths did not necessarily prove better. Understanding this nuance is important, since we can now predict that increasing bitwidth does not necessarily monotonically lead to strictly better results due to a possibility of bad balance between overfitting and generalization. This workflow is an efficient pipeline for generating a powerful model while simultaneously minimizing the necessary expenditures of time and computational resources on training and analyzing models.

Chapter 7

Conclusions and Future Work

In this work we propose a systematic and efficient methodology for evaluating the performance qualities of generative systems. This approach grew from the initial numerical metrics that were only suitable for relative comparisons and only if performed at identical bandwidths, sample sizes, and approaches (three important hyperparameters for MMD)[GBR+12]. Using our novel integration of maximum mean discrepancy with permutation testing (titled PT-MMD), it is possible to perform a powerful and innovative analysis of the qualities of these models. The proposed model even allows us to determine when these models reach a level of quality that they cannot be distinguished by a human censor.

7.1 General Thoughts

Combining Permutation Testing with Maximum Mean Discrepancy (PT-MMD) results in an effective way to determine the quality of a model. When looking at generative systems, the most important quality they can possess is the ability to generate samples that are sufficiently close to those generated from the real distribution that they cannot be distinguished. Once a system reaches this point and it is sufficiently good, any further enhancement of the score results in overfitting. Having a methodology that can evaluate these generated samples proves as important as having a validation set in conventional discriminative models. Validation sets are used to ensure that a discriminative model is not overfitted and making false conclusions. In the same way, a permutation test evaluation can confirm that a generative model is making high quality generated samples and is robust enough to withstand a rigorous test.

By observing the consistent results demonstrated by the permutation test, it is clear that the permutation test is a sufficiently accurate statistical measure of the quality of the samples generated by a given model. Furthermore, manipulating the distance measurement in the permutation test-enhanced MMD (PT-MMD) allows for an increase in comprehensibility and for a nimble reaction to different datatypes and generative systems. Combining these permutation tests with a training infrastructure can lead to an effective workflow for creating new models from large datasets. This workflow will lead to an efficient and error-free methodology for training new generative systems.

Effectively, it is now possible to evaluate existing generative systems using a statisti-

cally sound approach. Furthermore, integrating this statistical approach into the training methodology helps design these systems in a power-efficient manner. Knowing how good a system is helps in making the system more power-efficient, which simultaneously helps it scale into micro-devices that need effective results while minimizing power consumption. The presented results cement our proposed workflow as a powerful and effective technique for developing such systems.

7.2 HaarPSI Wavelet Metrics

An alternative metric is known as the HaarPSI (Perceptual Similarity Index) [RBKW18]. It is a similarity index that directly compares two images using a series of kernel evaluations and filters that are designed to mimic human sightlines instead of computerized gaussian sightlines. Effectively, they create a non-linear mapping that determines the most likely changes that would affect human vision, while de-emphasizing the small shifts that computers are excellent at detecting and that humans cannot see. HaarPSI has specifically been compared against a series of other metrics, including MMD, to determine which one best represents human vision. It has consistently performed well in such tests, establishing a distinct better system for conferring human vision on an algorithmic basis. Since HaarPSI is effectively a distance metric bounded from 0 to 1, it can be integrated into the kernel-based methodology of MMD.

7.2.1 Human vs Computer Vision

As HaarPSI is optimized to better coordinate with human vision, it can prove to be an interesting alternative to conventional distance measurement if the generative system is presumed to be more likely to interact directly with human users as opposed to as a training platform for other AI [RBKW18]. By manipulating the metric used, it is possible to create slightly different workflows and systems based on the same ideas presented here to tackle different situations [GBR+12]. Tricking humans and computers requires slightly different approaches to sample generation; and understanding this distinction and picking the correct approach is especially important when evaluating various generative systems.

7.2.2 Integration with Permutation Testing

Since the Haar Wavelet PSI effectively acts as a distance measurement, it can be easily integrated into the MMD pipeline. By replacing the basic Euclidean distance used in a Gaussian kernel, it is possible to have what is effectively a Haar Wavelet PSI based kernel [GBR+12].

$$k(x, x') = e^{-\frac{D(x,x')^2}{2\sigma^2}}$$

Upon replacing the D in the kernel above with any distance measurement, it is possible to create a kernel for any suitable function as described in Chapter 3. Since the MMD algorithm can cleanly integrate any distance measurement through a kernel-based approach, it is theoretically possible to try many different distance metrics and choose the best one that is relevant to a specific problem. Since the permutation testing algorithm simply uses

the result of MMD, changing the inner workings of the MMD code does not affect the permutation test, enabling an easily manageable system. One major downside of using Haar Wavelet PSI as a distance measurement is the fact that it is several orders of magnitude more computationally expensive than Euclidean distance. As a counterweight, MMD is highly susceptible to parallelization, as none of the intermediary steps require results from one another. It is also feasible to parallelize the Permutation test, further increasing throughput. Since the computations are relatively complex, both of these parallelizations will most likely require CPU parallelization in the immediate future.

7.3 Scaling to Other Systems

The workflow proposed in this report is suitable for training any type of generative model. Even though the initial results suggested that an encoding scheme was necessary for the permutation test to successfully evaluate the results of a given model, it was subsequently found that it was inherent to the original model and some inferior design solutions. After optimizing the model through the use of superior quantizations, we successfully evaluated the quality of the model without using a label-wise division. Now, it is feasible to scale this approach to any other generative system, as long as there is a suitable distance metric to compare the generated samples against real samples.

7.4 Final Thoughts

Over the course of this project, the complexity of working with generative systems truly became apparent. They are incredibly powerful and yet fundamentally difficult to evaluate and understand without significant human intervention. Removing some human error will pay off by creating more and more robust and powerful generative systems. By being able to directly compare their quality in absolute terms of sufficiency alongside their power consumption and more negative qualities, it will be possible to more systematically design such systems. Having a systematic outlook on machine learning is essential for it to become a more mathematically sound and conclusive field, as a major current problem is that lack of reproducibility due to noise and the wide range of hyperparameters causing major consistency issues. By eliminating the subjective, human introduced cut-off point in training a generative system, this paper proposes a novel and reproducible approach to creating and evaluating generative models. Expanding this system to more types of models will result in more trustworthiness throughout the field, as having a ordinal score of sufficient quality can eliminate this major point of contention seemingly inherent to looking at results produced by generative systems.

Appendix A

Annealed Importance Sampling

Whereas the majority of the math and statistical techniques used throughout this thesis are described in their relative chapters, the methodology behind Annealed Importance Sampling is more tangentially related to the thesis as a specific technique, as opposed to a mathematical approach that was purposefully modified. Thus, its process will be described in this Appendix, so as to not interrupt the flow of the rest of the argument.

A.1 General Overview

Annealed Importance Sampling is a way to simulate a complex super high-dimensional function by approximating it. Conventionally, this approximation is done through importance sampling, a stochastic sampling technique, but standard importance sampling does not scale well to complex distributions due to an inherently difficult-to-optimize hyperparameter. Annealed Importance Sampling eliminates this hyperparameter by replacing

it with a more abstract function and a series of intermediate distributions, that when trained, approximate the correct version of the hyperparameter [Kri]. The end result of this approximation is a likelihood score for a specific set of data given a RBM model. This likelihood score can then be used to compare various models alongside their quantizations in order to provide an accurate measure of relative model quality.

A.2 Derivation

When working with this project, there are two main inputs to AIS. There is the model, which can be quantized or modified however is relevant, and there is the dataset, which is composed of a large portion of MNIST, allowing for a more accurate and relevant likelihood score [RS08]. Using those two components, the steps to receive a likelihood score are as follows.

1. Setup:
    (a) Let $p_0(x) = p(x) \propto f_0(x)$ be the target distribution.
    (b) Let $p_n(x) = p(x) \propto f_n(x)$ be the proposal distribution that can be sampled from.
    (c) Define a sequence of transitions from $p_n(x)$ to $p_0(x)$ called $p_j(x) \propto f_j(x)$.
        i. Their requirement is that $p_j(x) \neq 0$ whenever $p_{j-1}(x) \neq 0$, allowing them to have the same support.
    (d) Define local transition probabilities $T_j(x, x')$

2. Execution:

   (a) It is now possible to sample from $p_0(x)$ if:

      i. Sample an independent point from $x_{n-1} \sim p_n(x)$.

      ii. Sample x_{n-2} from x_{n-1} by doing MCMC w.r.t. T_{n-1}.

      iii. ...

      iv. Sample x_1 from x_2 by doing MCMC w.r.t. T_2.

      v. Sample x_0 from x_1 by doing MCMC w.r.t. T_1.

   (b) Evaluate W:

      i. $W = (f_{n-1}(x_{n-1})/f_n(x_n)) * (f_{n-2}(x_{n-2})/(f_{n-1}(x_{n-1}))\dots$

   (c) Calculate Expectation

      i. $\mathbb{E}_{p(x)}[x] = \frac{1}{\sum_i^N w_i} \sum_i^N x_i w_i$

At this point, there theoretically is an expectation value which can be converted to a likelihood. However, there is a lack of annealing and definition for $f_j(x)$ and for $T_j(x, x')$. To remedy this problem, it is necessary to introduce intermediate functions. Thus, $f_j(x) = f_0(x)^{\beta_j} f_n(x)^{1-\beta_j}$ where $1 = \beta_0 > \beta_1 > \cdots > \beta_n = 0$ [Kri].

A.3 AIS for MNIST

In order to use the previous derivation in the problem at hand, it is necessary to convert some of the components to better fit the relevant paradigm. To this end, $p_0(x)$ is

defined as the underlying true distribution of MNIST, which is unknowable due to the high-dimensionality inherent to it. On the other hand, $p_n(x)$ is defined as the available MNIST dataset, allowing for approximate, if slightly limited, sampling [RS08]. Furthermore, instead of using the expectation, the most relevant metric for MNIST is the likelihood of the data in $p_n(x)$ given the available model. Upon executing these two changes, it is possible to directly use AIS to compare RBM models.

Bibliography

[Bon36] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

[Con] Wikipedia Contributors. Q (number format).

[Das99] Sanjoy Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 634–, Washington, DC, USA, 1999. IEEE Computer Society.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[Den12] Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, nov 2012.

[ET94] Bradley Efron and Tibshirani. *An Introduction to the Bootstrap,Chapman and Hall/CRC Monographs on Statistics and Applied Probability*. 1994.

[GBR+12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, March 2012.

[KR] Thomas Lumley Ken Rice. Permutation tests. http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf.

[Kri] Agustinus Kristiadi. Introduction to annealed importance sampling.

[LR05] E. L. Lehmann and Joseph P. Romano. Generalizations of the familywise error rate. *The Annals of Statistics*, 33(3):1138–1154, jun 2005.

[RBKW18] Rafael Reisenhofer, Sebastian Bosse, Gitta Kutyniok, and Thomas Wiegand. A haar wavelet-based perceptual similarity index for image quality assessment. *Signal Processing: Image Communication*, 61:33–43, feb 2018.

[RS08] Iain Murray Ruslan Salakhutdinov. On the quantitative analysis of deep belief networks. *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[Sch15] Sven Schmit. Multiple hypothesis testing, October 2015.

[SHM+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484 EP –, 01 2016.

[SSL+17] Soeren Sonnenburg, Heiko Strathmann, Sergey Lisitsyn, Viktor Gal, Fernando J. Iglesias García, Wu Lin, Soumyajit De, Chiyuan Zhang, frx, tklein23, Evgeniy Andreev, JonasBehr, sploving, Parijat Mazumdar, Christian Widmer, Pan Deng / Zora, Giovanni De Toni, Saurabh Mahindre, Abhijeet Kislay, Kevin Hughes, Roman Votyakov, khalednasr, Sanuj Sharma, Alesis Novik, Abinash Panda, Evangelos Anagnostopoulos, Liang Pang, Alex Binder, serialhex, and Björn Esser. shogun-toolbox/shogun: Shogun 6.1.0, November 2017.

[WBSG16] Y. Wu, Y. Burda, R. Salakhutdinov, and R. Grosse. On the quantitative analysis of decoder-based generative models. *ArXiv e-prints*, 2016.

[yK18] Chih yin Kan. Power efficient image classification and generation using fixed point gibbs sampling. Master's thesis, University of California San Diego, 2018.