Prechastic Coding: An Alternative Approach to Neural Network Description Lengths

Anonymous Author(s)
Affiliation
Address
email

Abstract

The minimum description length (MDL) principle has a rich history of informing neural network research and there are numerous algorithms for developing efficient neural network description lengths. Of these methods, prequential coding, based on the prequential approach to statistics, has proven to be highly successful. Despite its achievements, general prequential coding limits learning at each increment to a prefix of a given dataset - a constraint which is potentially misaligned with an effective learning process. In this paper we introduce prechastic coding, an alternative to the prequential approach which is based on a guided, noisy sequence of intermediate learning steps. In our experiments we determine that the prechastic coding can challenge prequential coding in certain scenarios, whilst also leaving significant potential for further improvement.

1 Introduction

Pioneered by Jorma Rissanen [1, 2, 3, 4], the MDL principle has a rich history of informing and advancing machine learning, underlying important work on topics such as variational inference for neural networks [5, 6]. At a high level, the MDL principle advocates for model selection based on measures of both model performance and model complexity. This viewpoint can be informally expressed by the notion that a good model of some data allows for the efficient transmission of both the data and the model. Intuitively, overly complex models which fit the data extremely well are not desirable as, while the model itself can achieve highly compressed lossless encodings of the data, the combined cost of communicating the data and the model is large.

At first, neural networks (particularly deep learning models) appear to stand in contrast to the MDL philosophy as they often demonstrate compelling performance whilst having extremely high parameter counts. This misconception stems from a naive coding scheme where parameters are passed as raw floating point numbers before the lossless transmission of data. Alternate schemes which can be used to develop far better code lengths than the naive encoding include network compression, intrinsic dimension and variational approaches [5, 6, 7, 8, 9, 10]. However, Blier and Ollivier [10] demonstrated that all of these schemes are inferior to a method known as prequential coding.

Prequential coding stems from the prequential approach to statistics [11] and works by sending data incrementally and updating the model after each transmission (see Figure 1 for a high-level visual diagram of an iteration). As a result, the prequential scheme leverages a model's ability to generalise from limited data. Blier and Ollivier [10]'s work established the pre-eminence of prequential coding for description lengths of deep learning models. Subsequently prequential coding has also facilitated state-of-the-art results in compression as the Large Text Compression Benchmark [12], a competition to compress one gigabyte of English Wikipedia test, is currently topped by the `nncp` algorithm which is largely a prequential approach with some extra features [13, 14].

Submitted to the Machine Learning and Compression Workshop at the 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

More recently, work by Bornschein et al. [15] found that the block transmission approach to prequential coding, used by [10] to lower computational costs, could be improved using techniques from continuous learning. Additionally there has been significant research on the use of prequential code lengths as an evaluation metric for various criteria [16, 17, 18]. Despite this popularity, the prequential approach is not without its drawbacks as a description length/compression mechanism. Many datasets might not be presented in an ordering particularly conducive for learning; for example, one might conjecture that in many scenarios incrementally learning a body of text would be better done by increasing the level of abstract complexity of the concept at each iteration, rather than progressing through the text one word at a time. This begets the question - can one find a better general method of computing description lengths for neural networks? In this paper we introduce a challenger to prequential coding termed prechastic coding (a portmanteau of predictive and stochastic). Prechastic coding shifts the concept of intermediate training datasets from the prequential viewpoint of cumulative, sequential partitions to noisy views of the full dataset by allowing fake labels at broadly diminishing rates across the scheme. Rather than predicting subsequent individual labels in the data sequence, the prechastic method uses the model to iteratively denoise the stochastic, yet curated, intermediate datasets. In our experiments we find that in select scenarios a greedy version of the prechastic code approaches the performance of the prequential. However, the prechastic approach as presented herein also allows for significant future improvement as a core component of the method, *i.e.* the selection of guiding distributions, is left as an open-ended topic of discussion.

2 Prechastic Coding

In this section we will describe the specifics of the general prechastic approach along with variants of interest. Before we proceed, we first describe the standard supervised learning compression scenario. In line with Blier and Ollivier [10]'s work on prequential coding we will use this setting throughout the remainder of the paper in order to develop the prechastic approach (although we note that this is by no means a necessity and is simply for pedagogical reasons). Consider a sender and a receiver who both have a copy of a sequence of N inputs $x_{1:N}$ and have agreed to some identically initialised learning model. The latter agreement often involves a high level description of an architecture and initialisation procedure along with a mutual seed for a pseudorandom number generator. The input data is randomly ordered yet identical for both the sender and the receiver. Each x_i has a corresponding label $y_i \in \{1, 2, \dots K\}$ which are, initially, only known to the sender. The sender would like to transmit $y_{1:N}$ to the receiver using as little information as possible.

Consider a sequence of probability distributions $Q_1, Q_2, \dots Q_T$ each defined over $\{1, 2, \dots K\}^N$. Each individual Q_i assigns probabilities to all K^N possible permutations of labels, both true or false, to the dataset $x_{1:N}$. Samples from $Q_1, Q_2, \dots Q_T$ constitute successive, intermediate datasets for the selected model to train on and we shall therefore refer to them as the *guiding distributions*. Since, until the final transmission, only the sender knows all of the true labels $y_{1:N}$, the sender must use this information to compute guiding distributions which lead to an efficient code length.

To initiate the general prechastic algorithm, the receiver creates predictions for all the labels for the dataset from its copy of the untrained model. We will denote these predictions as P_1 and note that in many circumstances P_1 is likely to be an approximate uniform distribution over all K^N possible predictions. The sender computes an identical copy of P_1 and transmits some sample $\boldsymbol{q_1} \sim Q_1$ using $\mathcal{O}(\text{KL}[Q_1 \parallel P_1])$ bits; possible machinery for this is discussed below in the section on relative entropy coding. Q_1 could be pre-determined or computed once P_1 has been calculated. The receiver then uses the noisy labels $\boldsymbol{q_1}$ to train the model and create an updated set of predictions P_2.

This process is repeated for a total of T iterations with the sender transmitting some sample from Q_i at each step to the receiver who, subsequently, uses this sample to train their model and form updated predictions P_{i+1}. The sum cost of these transmissions, including a final lossless encoding of $y_{1:N}$, constitutes the total code length for the prechastic approach. A full diagram of the process in comparison with the prequential approach can be found in Figure 1. A summary of the general prechastic coding algorithm is given in Algorithm 1.

The difference between the prequential and prechastic methods boils down to a difference in how the intermediate training datasets are viewed: in the prequential approach such intermediate datasets are expanding restrictions of the original dataset; in the prechastic approach they are noisy views of the entire original dataset with an overall trend towards less noise. While deep learning models have

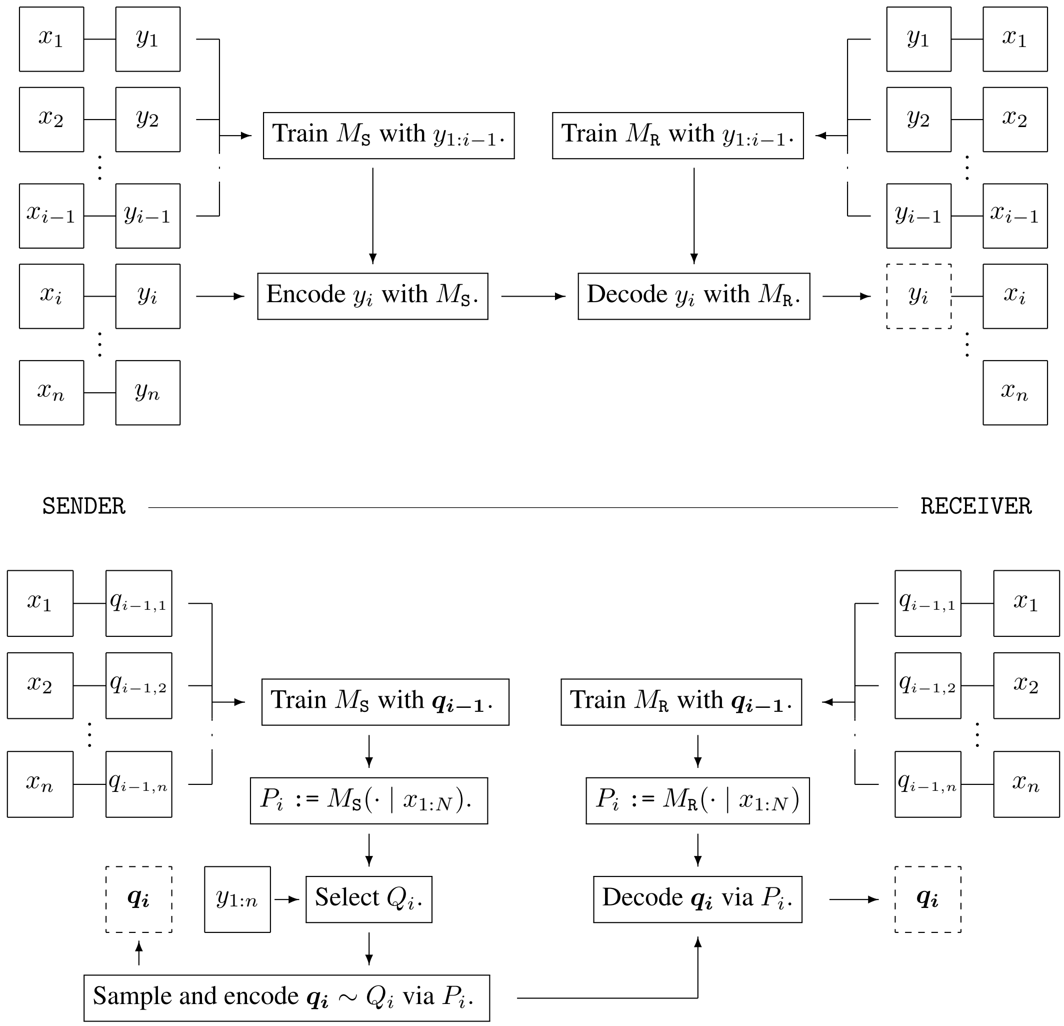

Figure 1: High level overview of an iteration of prequential coding (top) and prechastic coding (bottom). Both the sender and the receiver model are trained identically, *i.e.* $M_{\mathtt{S}} \equiv M_{\mathtt{R}}$. This diagram slightly differs from the presentation in Algorithm 1 as each iteration begins with model training.

the ability to reach negligible levels of training error on noisy labels, training regimens are typically crafted for performance on unseen data. For this reason, if the guiding distributions are selected carefully, we should in expectation see improvements between the prechastic iterations. We explore quantitative results further in Section 3; however, we will first discuss relative entropy coding as well as specific prechastic algorithms.

Relative Entropy Coding Consider the following communication scenario: a sender would like to send a sample from a distribution Q to a receiver who only has access to a distribution P. The sender does not care which particular sample is transmitted, only that it comes from the distribution Q. Relative entropy coding (REC) algorithms communicate such a sample with an expected code length of $\mathcal{O}(\mathrm{KL}[Q \parallel P])$ [19, 20]. Initial work by Harsha et al. [21] proposed a computationally intractable rejection sampling algorithm; later, Havasi et al. [22] used an importance sampling approach which first generates $M = \lceil 2^{\mathrm{KL}[Q \parallel P]} \rceil$ samples from the P distribution.[1] Each sample x is weighted according to the ratio $Q(x)/P(x)$ and, after normalisation, the resulting categorical distribution is sampled to select an index from 1 to M. This index is then transmitted at a cost of $\log_2(M) \approx \mathrm{KL}[Q \parallel P]$. Critically, Havasi et al. [22] demonstrated that, via a result from Chatterjee

[1] Note that in all instances in this paper, the Kullback-Leibler divergence is given in bits - many of the papers referenced in this section instead use nats.

Algorithm 1 The generic prechastic coding algorithm.

S initialises a model $M_{\texttt{S}}$.
R initialises an identical model $M_{\texttt{R}}$.
for i := 1 to T **do**
  $P_i := M(\cdot \mid x_{1:N})$ where $M \equiv M_{\texttt{R}} \equiv M_{\texttt{S}}$.
  S generates a sample $\boldsymbol{q_i} \sim Q_i$.
  S transmits $\mathcal{O}(\mathrm{KL}[Q_i \parallel P_i])$ bits which enable R to recreate $\boldsymbol{q_i}$.
  $M_{\texttt{R}}$ is trained on $\boldsymbol{q_i}$.
  $M_{\texttt{S}}$ is trained on $\boldsymbol{q_i}$ in an identical manner.
end for
S encodes $y_{1:N}$ using $M_{\texttt{S}}(y_{1:N} \mid x_{1:N})$.
S transmits the code for $y_{1:N}$ to R.
R decodes $y_{1:N}$ using $M_{\texttt{R}}(y_{1:N} \mid x_{1:N})$.

and Diaconis [23], setting $M = \lceil 2^{\mathrm{KL}[Q \parallel P]} \rceil$ was a sufficiently large sample size to keep the bias in the sampling low. In our experimental section we make use of the importance sampling procedure in a first-pass approach to selecting the guiding distributions. Further work on REC and REC-related methods include: Flamich et al. [19] who suggested a method of dividing the transmission process into a sequence of intermediary steps, Flamich et al. [20], who introduced approaches based on A* sampling [24], Li and Gamal [25]'s research on Poisson functional representation, and Theis and Ahmed [26]'s Ordered Random Coding method (presented under the framework of the related reverse channel coding problem [27]). In some sense, carefully selecting guiding distributions can achieve a similar intermediary effect to the auxiliary variable method of Flamich et al. [19] as it also mitigates much of the exponential runtime effects.

2.1 The Greedy Prechastic Algorithm

We shall now describe an effective greedy approach to the prechastic approach which considers potential P_{i+1} values. Rather than choosing a specific Q_i from a process such as the minimisation of an optimisation problem as in Appendix A, the greedy approach generates G samples from P_i then trains a model on each of these samples independently. The sample whose model, after training, minimises the cost of encoding the true values is then encoded via index at a cost of $\mathcal{O}(\log(G))$ bits. Note that the greedy approach, which is outlined in Algorithm 2, is a slight deviation from the general prechastic scheme as it does not explicitly choose a guiding distribution Q_i.

3 Experiments

In the following experiments we evaluate the greedy prechastic algorithm in comparison with the prequential approach as well as two variations of the first-pass convex method described in Appendix A. We operate under the supervised learning scenario described in Section 2 and apply it to the MNIST [28] and Fashion-MNIST [29] datasets. Learning is conducted using a simple MLP with two 128 neuron hidden layers as well as a convolutional LeNet-style network [28].

These models were chosen in part to accommodate for the computational cost of computing a full, non-batched prequential code which requires $\mathcal{O}(N^2)$ items of data to be processed per epoch of training. In comparison, if we consider time contributions as primarily determined by training, the general prechastic scheme is $\mathcal{O}(TN)$ whilst the greedy scheme is $\mathcal{O}(TNG)$. For practical values of T and G that produce efficient code lengths, we found that our experiments had to restrict both datasets to smaller sizes of $N = 128, 256,$ and 512 in order to lower runtimes. Despite this constraint, Bornschein et al. [30] found that model selection based on small dataset restrictions may give similar results to model selection which uses the entire dataset.

Ten trials of prequential experiments were run for each model and dataset combination using a batch size of 32. The models were trained for a total of five epochs and codelengths were computed using the final model at the end of the fifth epoch. We did not use a scheme which evaluated code lengths at the end of each epoch and subsequently took the best performing model as this would have required transmissions of the epoch index and consequently incurred a large penalty. It was determined that five epochs produced reasonably efficient prequential codelengths for the models and datasets used.

Algorithm 2 The greedy prechastic coding algorithm.

S initialises a model $M_{\mathtt{S}}$.
R initialises an identical model $M_{\mathtt{R}}$.
for i := 1 to T **do**
    $P_i := M(\cdot \mid x_{1:N})$ where $M \equiv M_{\mathtt{R}} \equiv M_{\mathtt{S}}$.
    R and S generate identical sets of G samples $\boldsymbol{p_i^{(1)}}, \boldsymbol{p_i^{(2)}}, \ldots, \boldsymbol{p_i^{(G)}} \sim P_i$.
    S initializes a model M'.
    $V := \infty$, $g^* := 1$
    for j := 1 to G **do**
        S creates a clone of M' and trains it on $\boldsymbol{p_i^{(j)}}$, obtaining M'_j.
        if $-\log_2(M'_j(y_{1:N} \mid x_{1:N})) < V$ **then**
            $V := -\log_2(M'_j(y_{1:N} \mid x_{1:N}))$
            $g^* := j$
        end if
    end for
    S transmits g^* to R at a cost of $\mathcal{O}(\log(G))$ bits.
    $M_{\mathtt{R}}$ is trained on $\boldsymbol{p_i^{(g^*)}}$.
    $M_{\mathtt{S}}$ is trained on $\boldsymbol{p_i^{(g^*)}}$ in an identical manner.
end for
S encodes $y_{1:N}$ using $M_{\mathtt{S}}(y_{1:N} \mid x_{1:N})$.
S transmits the code for $y_{1:N}$ to R.
R decodes $y_{1:N}$ using $M_{\mathtt{R}}(y_{1:N} \mid x_{1:N})$.

The prechastic experiments were conducted on the greedy prechastic algorithm along with two variants of the first pass approach from Appendix A which were iteratively solved using the CVX package [31, 32]. The first variant, `FPC-Q` directly sampled each Q_i whilst the second version, `FPC-R` used the importance sampling REC procedure of Havasi et al. [22] to indirectly sample each Q_i through its respective P_i. Splitting the first pass approach into these two variants was done in order to quantify the bias affects from the importance sampling procedure. Note that because `FPC-Q` uses direct samples it is a thought experiment and not a practical compression algorithm. A running average of up to five of the most recently communicated samples were used as a training signal in order to improve stability. β was set to 7 and the cost of each iteration was logged as $\log_2(\lceil 2^{\beta_i^*} \rceil)$ (note that this does not communicate the size of β_i^* itself; in practice it is likely better to simply use β and communicate it once). For the $128, 256$, and 512 count dataset sizes, we used maximum iteration counts of $25, 50$, and 100, respectively (the cost of transmitting the best index was included in the code lengths).

For the greedy prechastic experiments, the hyper-parameter G was set to 128, *i.e.* 7 bits of information was transmitted per iteration. To increase stability in the face of small datasets, multiple samples were generated from P_i for each of the G trials and the average values were used for training. Note that while there are still only G averaged options to choose from (and thus there is still only 7 bits of data transmitted per iteration) larger numbers of multiple samples drive the training signal towards P_i. In order to balance this effect with the desired stability, for the $128, 256$, and 512 count dataset sizes multiple sample values of $25, 5$, and 2 were used, respectively. The larger datasets also required higher maximum iteration counts of $20, 40$, and 60 for the $128, 256$, and 512 count dataset sizes, respectively. The best result from across these iterations was taken as the code length (including the cost of transmitting this index). The results from the prequential, first pass, and greedy prechastic experiments are presented in Table 1. All code was executed on a consumer-grade build (Intel i7-4790k and an Nvidia RTX 3060) and the longer experiments typically took on the rough order of hours to a day.

4 Conclusion

The greedy prechastic algorithm performed well in our experiments, approaching and demonstrating comparable performance in the MNIST/LeNet testing suite. Further results across the remainder of experiments were competitive although the prequential tests consistently produced the best code

SIZE	CODING	MNIST	
		MLP	LENET
128	PREQ.	0.896 ± 0.008 (380.8 ± 3.2)	0.895 ± 0.005 (380.6 ± 2.1)
	FPC-Q	0.981 ± 0.006 (417.0 ± 2.5)	0.997 ± 0.008 (424.1 ± 3.3)
	FPC-R	1.002 ± 0.005 (425.9 ± 2.1)	1.014 ± 0.004 (431.3 ± 2.1)
	GREEDY	0.948 ± 0.006 (402.9 ± 2.4)	0.933 ± 0.006 (396.5 ± 2.4)
256	PREQ.	0.726 ± 0.006 (617.7 ± 5.0)	0.710 ± 0.004 (603.6 ± 3.8)
	FPC-Q	0.879 ± 0.012 (747.3 ± 10.3)	0.805 ± 0.012 (684.6 ± 10.1)
	FPC-R	0.971 ± 0.006 (826.2 ± 4.9)	0.935 ± 0.010 (795.2 ± 8.6)
	GREEDY	0.744 ± 0.010 (632.6 ± 7.7)	0.696 ± 0.006 (592.0 ± 5.3)
512	PREQ.	0.540 ± 0.006 (917.9 ± 9.4)	0.512 ± 0.005 (870.0 ± 7.8)
	FPC-Q	0.762 ± 0.013 (1296.9 ± 22.1)	0.622 ± 0.007 (1057.8 ± 11.1)
	FPC-R	0.919 ± 0.006 (1563.2 ± 9.3)	0.770 ± 0.008 (1309.0 ± 13.9)
	GREEDY	0.620 ± 0.006 (1055.3 ± 9.8)	0.517 ± 0.004 (878.7 ± 6.0)

SIZE	CODING	FASHION-MNIST	
		MLP	LENET
128	PREQ.	0.718 ± 0.011 (305.2 ± 4.7)	0.836 ± 0.008 (355.4 ± 3.2)
	FPC-Q	0.909 ± 0.013 (386.5 ± 5.4)	0.980 ± 0.009 (416.5 ± 3.9)
	FPC-R	0.994 ± 0.013 (422.6 ± 5.3)	1.018 ± 0.007 (433.1 ± 3.1)
	GREEDY	0.851 ± 0.007 (361.7 ± 3.0)	0.926 ± 0.007 (393.7 ± 2.9)
256	PREQ.	0.555 ± 0.010 (472.4 ± 8.3)	0.715 ± 0.004 (608.3 ± 3.3)
	FPC-Q	0.746 ± 0.010 (634.3 ± 8.9)	0.869 ± 0.008 (739.3 ± 7.0)
	FPC-R	0.897 ± 0.011 (763.1 ± 9.5)	0.952 ± 0.009 (809.5 ± 7.4)
	GREEDY	0.619 ± 0.007 (526.7 ± 6.0)	0.743 ± 0.007 (631.6 ± 5.6)
512	PREQ.	0.447 ± 0.005 (761.0 ± 8.4)	0.577 ± 0.004 (980.8 ± 7.2)
	FPC-Q	0.688 ± 0.008 (1170.5 ± 13.2)	0.756 ± 0.007 (1285.5 ± 11.2)
	FPC-R	0.815 ± 0.008 (1386.1 ± 12.9)	0.868 ± 0.007 (1476.6 ± 12.1)
	GREEDY	0.494 ± 0.004 (840.8 ± 7.6)	0.619 ± 0.004 (1053.5 ± 6.4)

Table 1: Results from prechastic and prequential experiments on restrictions of the MNIST and Fashion-MNIST datasets. The average compression ratio and the average size in bits are presented along with their corresponding standard error values.

lengths. However, across the largest datasets of size 512, the absolute compression ratio of the greedy approach was never more than eight percent greater than the prequential (less than fifteen percent in terms of relative performance to the prequential). Expectedly, as a first-pass at selecting the guiding distributions, the convex results fared worse than the greedy algorithm. However, the comparison between `FPC-Q` and `FPC-R` performance did yield insights into the underlying REC algorithm used in `FPC-R`. As there was a large drop-off from the `FPC-Q` results down to the `FPC-R` the importance sampling approach of Havasi et al. [22] clearly introduced a significant amount of sampling bias.

Looking beyond these experiments, the prechastic approach is a highly flexible coding scheme with potential for further development and improvement. Because the general prechastic scheme does not prescribe a specific method for selecting the guiding distributions, future work should investigate more advanced selection techniques to further improve prechastic code lengths. The greedy method could also potentially be improved by considering higher order decisions at each iteration. Future research might also consider the reduction of computational costs which partially necessitated dataset size restrictions during the experiments. Bornschein et al. [30] found that model selection based on small dataset restrictions may provide similar results to using the entire dataset, however, if one would still like to use large datasets one possible method might be to bootstrap from smaller datasets up to large ones, spreading the prechastic iterations across smaller views of the original dataset.

References

[1] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[2] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984.

[3] Jorma Rissanen. Stochastic complexity and modeling. *The annals of statistics*, pages 1080–1100, 1986.

[4] Jorma Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(3):223–239, 1987.

[5] Geoffrey E. Hinton and Drew van Camp. Keeping the Neural Networks Simple By Minimizing the Description Length of the Weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, page 5–13. Association for Computing Machinery, Aug 1993.

[6] Alex Graves. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[7] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[8] Ting-Bing Xu, Peipei Yang, Xu-Yao Zhang, and Cheng-Lin Liu. Margin-Aware Binarized Weight Networks for Image Classification. In *Image and Graphics: 9th International Conference, ICIG 2017*, pages 590–601. Springer, 2017.

[9] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the Intrinsic Dimension of Objective Landscapes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[10] Léonard Blier and Yann Ollivier. The description length of deep learning models. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2220–2230, 2018.

[11] A. P. Dawid. Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach. *Journal of the Royal Statistical Society. Series A (General)*, 147(2):278–292, 1984.

[12] Matt Mahoney. Large Text Compression Benchmark, n.d. URL `http://mattmahoney.net/dc/text.html`.

[13] Fabrice Bellard. Lossless Data Compression with Neural Networks. Technical report, Self-published, May 2019. URL `https://bellard.org/nncp/nncp.pdf`.

[14] Fabrice Bellard. NNCP v2: Lossless Data Compression with Transformer. Technical report, Self-published, Feb 2021. URL `https://bellard.org/nncp/nncp_v2.1.pdf`.

[15] Jörg Bornschein, Yazhe Li, and Marcus Hutter. Sequential learning of neural networks for prequential MDL. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=dMMPUvNSYJr`.

[16] Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373, 2019. URL `http://arxiv.org/abs/1901.11373`.

[17] Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. Predicting Inductive Biases of Pre-Trained Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[18] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. Rissanen Data Analysis: Examining Dataset Characteristics via Description Length. In *Proceedings of the 38th International Conference on Machine Learning*, page 8500–8513. PMLR, Jul 2021.

[19] Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. Compressing images by encoding their latent representations with relative entropy coding. In *NeurIPS*, 2020.

[20] Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Fast relative entropy coding with A* coding. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 6548–6577. PMLR, 2022.

[21] Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 10–23. IEEE, 2007.

[22] Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. In *ICLR (Poster)*. OpenReview.net, 2019.

[23] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

[24] Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *NIPS*, pages 3086–3094, 2014.

[25] Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Trans. Inf. Theory*, 64(11):6967–6978, 2018.

[26] Lucas Theis and Noureldin Y. Ahmed. Algorithms for the communication of samples. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 21308–21328. PMLR, 2022.

[27] Charles H Bennett, Peter W Shor, John A Smolin, and Ashish V Thapliyal. Entanglement-assisted capacity of a quantum channel and the reverse shannon theorem. *IEEE transactions on Information Theory*, 48(10):2637–2655, 2002.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[30] Jorg Bornschein, Francesco Visin, and Simon Osindero. Small data, big decisions: Model selection in the small-data regime. In *International conference on machine learning*, pages 1035–1044. PMLR, 2020.

[31] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `https://cvxr.com/cvx`, March 2014.

[32] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

A First-Pass Convex Approach

One of the central problems left unanswered by the general prechastic approach is how to select the guiding distributions. Choosing efficient $Q_1, Q_2, \ldots Q_T$ is a challenging problem which essentially requires one to design an appropriately difficult curriculum for potentially complex learning models. The following is a rudimentary attempt designed largely to illustrate the difficulties of selecting the guiding distributions. Consider the convex optimization problem

$$
\begin{aligned}
\min_{\beta_i, [\boldsymbol{Q_i}]} \quad & \beta_i - \sum_{j=1}^{N} \log_2 \left([Q_i]_{j,y_j} \right) \\
\text{s.t.} \quad & \sum_{j=1}^{N} \mathrm{KL}[[\boldsymbol{Q_i}]_{\boldsymbol{j}} \parallel [\boldsymbol{P_i}]_{\boldsymbol{j}}] \leq \beta_i \\
& \sum_{k=1}^{K} [Q_i]_{j,k} = 1, \quad \forall j \\
& 0 \leq [Q_i]_{j,k} \leq 1, \quad \forall j, k
\end{aligned}
$$

where $[\boldsymbol{Q_i}]$ is an $N \times K$ matrix which represents N independent categorical distributions across the label classes for each of the inputs. For an input x_j, the prediction rendered by the model P_i is denoted as $[\boldsymbol{P_i}]_{\boldsymbol{j}}$. $[Q_i]_{j,k}$ is the probability of label k given the distribution $[\boldsymbol{Q_i}]_{\boldsymbol{j}}$. Note that the budget variable β_i is implicitly non-negative. The cost function measures the order of information that would have to be transmitted for the receiver to draw single sample from Q_i along with the cost of sending the true labels $y_{1:N}$ if the receiver were to form predictions using Q_i. Naturally, it is unlikely that the receiver will be able to predict in a manner identical to Q_i after training on a single sample; however, Q_i is used as P_{i+1} would require model training.

By minimizing the cost function over β_i and $[\boldsymbol{Q_i}]$, a trade-off is struck between the quality of guidance and the rough cost of communicating a sample. In our experiments we also bound β_i to a hyper-parameter β by introducing the constraint $\beta_i \leq \beta$. This change allowed us to limit the rate of change of the guiding distributions over the iterations and also avoid intractably high computational costs from the REC importance sampling procedure.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the paper's abstract and introduction accurately reflect its contributions and scope without embellishment.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, this is discussed in the concluding section of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: There are no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper contains all the necessary information to reproduce the experimental results to the extent required to justify the claims and conclusions.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code for the relevant algorithms is available upon request.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all the necessary experimental details needed to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The standard error is provided for all experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All computational resources are fully documented in the experiments section of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted for this paper fully conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no direct mechanism for societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All creators are properly credited in the text.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.