Dynamic learning rate using Mutual Information

Shrihari Vasudevan
IBM Research - India
shrivasu@in.ibm.com

Abstract

This paper demonstrates dynamic hyper-parameter setting, for deep neural network training, using Mutual Information (MI). The specific hyper-parameter studied in this paper is the learning rate. MI between the output layer and true outcomes is used to dynamically set the learning rate of the network through the training cycle; the idea is also extended to layer-wise setting of learning rate. Two approaches are demonstrated - tracking relative change in mutual information and, additionally tracking its value relative to a reference measure. The paper does not attempt to recommend a specific learning rate policy. Experiments demonstrate that mutual information may be effectively used to dynamically set learning rate and achieve competitive to better outcomes in competitive to better time.

1 Introduction

Hyper-parameter selection in deep neural networks is mostly done by experimentation for different data-sets and models. A key example of one such hyper-parameter that is the subject of this paper is the learning rate. High learning rates, particularly in early training stages, can result in instabilities and fluctuations in the parameter search process. Established procedures to set learning rate to a low value at the beginning and then gradually "warm-up" to the desired learning rate have been used effectively [1, 2]. These approaches require the a-priori definition of a policy or schedule and the learning rate changes according to that fixed policy. The fixed policy may not be suited for different data-sets or model architectures which may be very different in complexity. The same policy may also not suffice in case the compute resources available, or other factors, necessitate changes in other training hyper-parameters (e.g. batch size) during training. Dynamically setting the learning rate through the training cycle is one way of handling such issues; this paper studies the feasibility of Mutual Information (MI) [3] as a metric to realize this objective.

2 Related work

Adaptive learning rate schedules based on gradients have been proposed in various gradient-descent based optimization algorithms used for training deep neural networks [4]. These include the likes of AdaGrad, AdaDelta, RMSprop, Adam and some more recent algorithms. These set learning rates at the level of individual parameters by considering the frequency or magnitude of updates; slow or infrequent updates characterized by smaller past gradients get more importance than fast or frequent updates characterized by larger past gradients. Depending on the data-set and model complexity, careful initial selection of the learning rate may be required.

Recent works of Tishby et al [5, 6] have attempted to explain deep learning through the Information Bottleneck (IB) [7] basis. In particular, the recent paper [6] made strong and wide-ranging claims on aspects relating to phases in deep learning, causal relationship between compression and generalization and the basis for compression in deep learning. Some of these claims were subsequently countered in [8], while acknowledging the potential of the more general MI and IB concepts. The current paper builds on this body of literature.
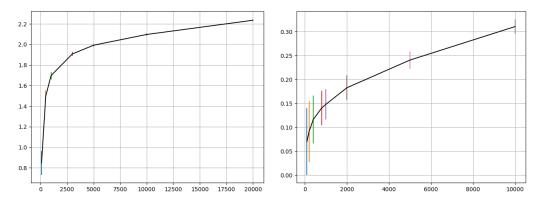
Figure 1: MI (of input and output training data) vs sample size for MNIST (left) and CIFAR-10 (right) as computed using the KSG estimator. The figures show estimated mean and standard deviation (error bar) for each sample size tested. Experiments in this paper use a sample size of 1000.

An important and related problem is that of estimation of MI. The problem has been widely studied and several estimation methods exist [9]. One such method is that of nearest neighbor approaches; these have been shown [10, 11] to be effective with high dimensional data and at large sample sizes. A widely cited example of this class of algorithms is the Kraskov-Stögbauer-Grassberger (KSG) estimator [12]. This approach is used for MI estimation in this paper; other algorithms can also be used. Recent developments in the area include [13] that estimates MI using pairwise distances between Gaussian mixture components and [14] that estimates MI through gradient descent over neural networks. Their properties and suitability to the context of this paper will be studied in future.

This work seeks to understand the operational utility of MI as a metric for deep learning and specifically, for dynamic hyper-parameter (in this paper, the learning rate) setting. For this paper, the essential core of a typical deep neural network training pipeline remains unchanged i.e. the use of mini-batch Stochastic Gradient Descent (SGD) optimization to optimize (minimize) the training cost (e.g. mis-classification error) is maintained but the learning rate is set adaptively considering the MI of hidden layer activations with the true output. The paper effectively demonstrates that an information driven "warm-up" and subsequent "cool-down" of learning rate can produce competitive outcomes on standard data-sets; it does not attempt to prescribe a specific learning rate policy.

3   Approach

In (deep) neural network models, MI lends a layer specific measure that may be utilized in multiple ways. It may potentially be used as a metric for parameter optimization through standard optimization methods used for the purpose [5]; works such as [15] suggested MI as providing an upper bound for prediction error. It may serve as a basis for dynamic tuning of network-level hyper-parameters. Further, as a layer-specific measure, it may be utilized for layer-wise dynamic tuning of hyper-parameters. Based on the information metric per epoch, interventions in hyper-parameters may be used to steer the learning process towards efficient and effective deep learning.

Computing MI is generally computationally expensive; performing MI computation after each epoch in deep neural net training can prove to be infeasible. This paper relies on two ideas to effectively use MI in training with large data sets - (1) use a randomly selected subset of data for MI computation - plotting the MI vs sample size curve for different data sets (see Fig 1) enables informed selection of an appropriate subset sample-size for per-epoch MI computation and (2) this approximate MI value may suffice if the relative measures can be utilized for the problem being addressed.

Given input X, output Y and hidden layer activations H_i, MI between input and output, $I(X;Y)$, is denoted as IXY. MI between hidden layer activations and output, $I(H_i;Y)$, is denoted by as IHY; specifically, MI between last (output) layer activations and output is denoted as $IHYLL$. Finally, MI between hidden layer activations and input, $I(H_i;X)$, is denoted by as IHX.

In the context of neural networks, the Data Processing Inequality (DPI) [3] effectively provides an upper bound to the information that each layer (including output) of a neural network can capture. It suggests that successive layers H_i operating on the input data X cannot increase its information content relative to the output data i.e. $IHY \leq IXY$; as a specific case, $IHYLL \leq IXY$.

With IXY computed from input data X and output data Y, this inequality may hold true for dense neural networks but will not hold true for convolution neural networks (CNNs). The use of multiple (say n) convolution filters in CNNs is akin to treating n image inputs tiled together with the respective convolution filter weights being mapped to the weights of a much larger dense neural network. Thus, a reasonable estimate of the upper bound IXY may be obtained by repeating or tiling X and Y, n times, and then computing the estimate IXY.

The work [6] suggests $IHYLL/IXY$ as a ratio of the amount of information captured by the model. The KSG estimator [12] incorporates a small amount of noise (a jitter in the order of 10^{-10}) in MI computation to overcome degeneracies in data. Saxe et al [8] observe that DPI will not hold when noise is added for the purposes of measuring MI. As a consequence of the DPI not being valid, IXY may not be an upper bound and $IHY > IXY$ is a possible outcome. This paper proposes that the upper bound IXY may be used as a "soft" criterion for dynamic learning rate setting using MI. Specifically, this paper proposes to increase the learning rate towards achieving the soft upper bound of IXY and using the DPI violation condition ($IHY > IXY$) as a signal to reduce learning rate.

While adaptive learning rates schedules have been developed using gradients [4] and can, in principle, be developed using other measures such as validation accuracy, the use of MI as a criterion for dynamic tuning of hyper-parameters is motivated by this measure being able to capture both linear and nonlinear dependence between the quantities of interest (in our case, hidden layer activations and the true output) and offer a layer-wise measure of optimality (IHY) with respect to a reference measure (IXY). This paper uses standard deep neural net training design choices (e.g. mini-batch SGD to minimize misclassification error) while setting the learning rate to maximize information with respect to true outcomes i.e. maximize IHY.

Given the soft upper bound of IXY, training the neural network increases the information of the last layer with respect to the true outcomes $IHYLL$ until it finally saturates to a maximum. This trend is also observed for previous layers though the change over the training cycle may be less dramatic and the IHY value is typically lower. This observation serves as the basis for dynamically setting the learning rate (LR) in this paper. Two approaches are explored -

1. Tracking the change in $IHYLL$, denoted by $\delta IHYLL$, relative to its value

   - This basically uses behavior that when the information measure saturates, change in the measure diminishes. The relative change measure $\frac{\delta IHYLL}{IHYLL} < \epsilon$ as $IHYLL$ saturates; ϵ is a small number.

2. Tracking $IHYLL$ and $\delta IHYLL$ relative to IXY

   - LR increases and decreases are set in terms of the relative measure $\frac{IHYLL}{IXY}$ so as to maximize $IHYLL$ relative to IXY; this measure coupled with the relative change $\frac{\delta IHYLL}{IXY}$ between epochs are used to decide on increases or decreases in LR.

Both approaches require the specification of a minimum and maximum LR (upper and lower bounds) and begin from the minimum value; experiments in this paper set these bounds around the desired LR. The first approach increases LR while the relative change in $IHYLL$ is significant (to a threshold, ϵ); thereafter LR decreases. The second approach tracks both the value $IHYLL$ and the change in $IHYLL$ between epochs, relative to IXY. LR increases while significant changes in $IHYLL$ occur and $IHYLL \leq IXY$ and thereafter, it decreases. In both cases, LR changes incrementally to enable gradual changes. LR increases may be performed at the same rate as decreases or may be dampened to control the max LR reached. LR policies used in the experiments are provided in the appendix. Note that this paper does not propose a specific learning rate policy; it focuses on demonstrating that MI can be used to dynamically set the LR to achieve competitive outcomes.
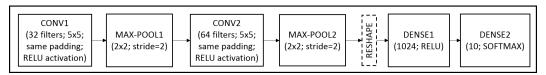
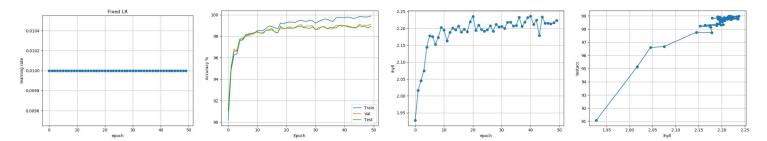Figure 2: Model used for experiments on MNIST data-set

4 Experiments

Two standard data-sets were used to demonstrate dynamic LR setting using MI - MNIST [16] and CIFAR-10 [17]. The emphasis (model and design choices) of these experiments was *not* maximizing accuracy for the data-set but to understand relative performance of dynamic LR setting (using MI) in comparison to alternatives. The model used for the experiments with MNIST data is shown in Figure 2; these experiments used standard off-the-shelf mini-batch SGD to minimize mis-classification error (categorical cross entropy). For CIFAR-10 experiments, the model used was based on [18]. The specific model implementation used dropout (50%) only after max-pooling layers, no L2 regularization for weights, a fixed momentum value of 0.9 and Nesterov acceleration. With a continuously decaying LR beginning at 0.01, it resulted in sufficiently close maximum test accuracies of 88.21% and 91.23%, with and without data augmentation respectively, to the respective outcomes (90.92% and 92.75%) over 350 epochs in the referenced paper. Experiments of this paper used the implementation without any data augmentation to enable a fair comparison between methods.
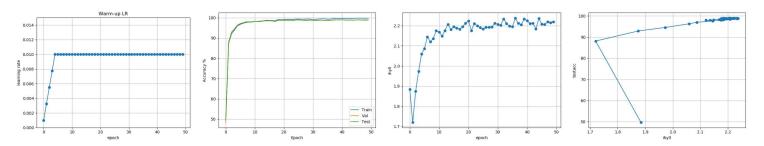
Figures 3 and 4 show outcomes of various LR selection methods applied to the MNIST and CIFAR-10 data-sets respectively; this experiment compared the use of a fixed (desired) LR, a warm-up beginning from a lower LR and increasing to the desired LR and the dynamic LR methods tracking either the change in $IHYLL$ relative to its value or tracking both the value and the change in $IHYLL$ relative to the reference measure of IXY. The CIFAR-10 experiment also adds a decaying LR policy into the comparison. The following trends were observed -

- Training deep neural networks results in increasing $IHYLL$. This is and intuitive and expected outcome of successful training.

- There is a non-linear and increasing relationship between test accuracy and $IHYLL$. Increasing $IHYLL$ results in higher test accuracy.

- Achieving maximum $IHYLL$ does not necessarily guarantee (from existing plots) maximum test accuracy but definitely gives a very competitive test accuracy. Per the Information Bottleneck approach, the same value of IHY may be associated with different IHX and it is likely that a more compressed (lower IHX) model generalizes better.

- It appears that an effective warm-up should ideally result in significant increase in $IHYLL$ towards IXY. The warm-up policy (linear increase to desired learning rate in 5 epochs) seems effective for MNIST and insufficient for CIFAR-10.

- Experiments reported in this paper and other attempts suggest that the MNIST was able to produce good outcomes with an aggressive (relative to CIFAR-10) warm-up and cool-down LR policy; CIFAR-10 on the other hand required the use of a slow warm-up and lower LR values over-all.

- It is clear that mutual information of hidden layer activations with respect to output may be useful for dynamically setting learning rate through the training process to obtain competitive to better test accuracies in competitive to better time. A policy involving both change in $IHYLL$ and its value relative to IXY produces better outcomes than tracking the former alone. In both MNIST (Figure 3) and CIFAR-10 (Figure 4), dynamic LR using both the change and the value of $IHYLL$ resulted in top accuracy levels being achieved in roughly half the number of epochs as compared to the corresponding fixed LR policy.

- A dynamic LR policy based on MI makes training easier in the sense that it moves this hyper-parameter selection problem one level up; the problems of specifying a single optimal LR for the entire training cycle or specifying an optimal warm-up policy to a "good" LR are overcome by automatically adjusting the LR every epoch, between bounds, to effectively realize an information driven warm-up and cool-down of the LR. Starting from a
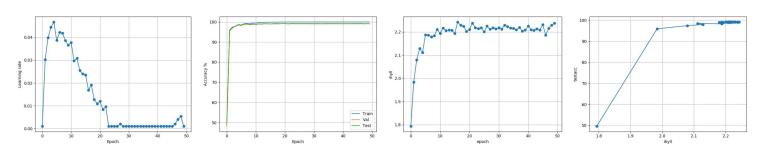
4

Figure 3: MNIST - comparison of fixed, warm-up and dynamic LR methods. Each row shows four plots - LR vs epoch, accuracy vs epoch, IHYLL vs epoch and test-accuracy vs IHYLL. Dynamic LR is able to start low and gradually explore a larger range of learning rates as required by the data/model and then cools down to produce better outcomes in better time. Tracking both change and value of $IHYLL$ relative to IXY performed better than tracking the relative change in $IHYLL$ alone.

Figure 4: CIFAR-10 - comparison of fixed, warm-up, decay and dynamic LR methods using a model based on [18]. Each row shows four plots - LR vs epoch, accuracy vs epoch, IHYLL vs epoch and test-accuracy vs IHYLL. Dynamic LR is able to start low and gradually explore a larger range of learning rates as required by the data/model to produce better outcomes in better time. Tracking both change and value of $IHYLL$ relative to IXY performed better than tracking the relative change in $IHYLL$ alone.

low value of learning rate results in a stable search process and outcomes. Competitive outcomes are achieved by exploring a larger space of learning rates than the fixed and warm-up strategies which are both essentially fixed learning rate policies; it is also also possible to achieve competitive outcomes using a smaller learning rate for a longer length of time.

Figure 5 shows an application experiment where the dynamic LR concept may be useful. During a training run, if the situation (e.g. availability of compute resources or simply, a training design choice) requires an increase in one hyper-parameter such as the batch-size (BS), the LR would have to be suitably increased or a drop in accuracy may occur. There are guidelines on managing the LR in such scenarios. This experiment however, demonstrates that dynamic LR based on MI can be effectively used to automatically adjust LR in such scenarios. The LR policy used here tracked only the value of $IHYLL$ relative to IXY for a few epochs before resuming the tracking of both change and value; this was done to enable the growth of both LR and $IHYLL$ as a consequence of the increased batch size before resuming the tracking of both its change and value. Note the higher LR reached in this experiment as compared to the fixed batch size run in Figure 3. Competitive to better outcomes were achieved in competitive to better time. The availability of a reference measure (a soft upper bound) IXY enables MI to be particularly suited to handle such scenarios, as compared to other more readily available measures.

The use of MI of the last layer alone $IHYLL$ provides a network-level intervention i.e. dynamic LR setting for all layers. The key property that mutual information affords is a layer-wise measure of optimality. The methods demonstrated thus far were extended to a layer-wise intervention i.e. dynamic LR setting of individual layers; this is demonstrated in Figure 6 where each layer's MI with the true outcomes IHY relative to the reference measure IXY enables the setting of a layer-specific LR. Competitive outcomes were obtained in competitive time, compared to the outcomes of Figure 3.

5   Conclusion

This paper demonstrated that using Mutual Information (MI) to dynamically set learning rate through the training cycle, in the context of deep neural networks, is both feasible and produces competitive to better outcomes in competitive to better time. The paper also demonstrated the application of this idea to automatically respond to changes in other hyper-parameters such as the batch-size and the extension of the idea to a layer-wise dynamic LR tuning through the training cycle. MI lends a layer-wise measure of optimality with respect to a reference value that can be leveraged to effectively steer deep neural network training to competitive/better outcomes.

Appendix

The following policies were used in experiments of this paper. Note that the paper does not attempt to prescribe a specific learning rate policy.
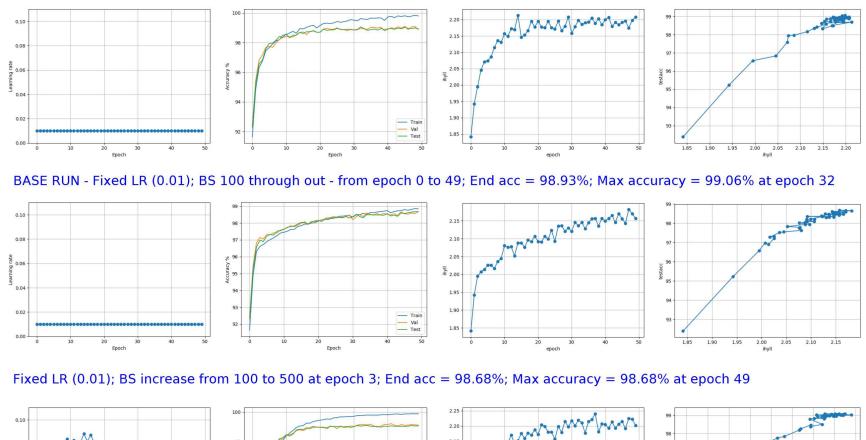
Dynamic LR policy based on change in $IHYLL$

$$\delta_t = |(IHYLL_{t-1} - IHYLL_{t-2})|/IHYLL_{t-1}$$

$$LR_t = \begin{cases} min(LR_{max}, LR_{t-1} + \gamma_1 * \delta_t) & (\delta_t > \epsilon) \\ max(LR_{min}, LR_{t-1} - \gamma_2 * \delta_t) & (\delta_t \leq \epsilon) \end{cases}$$

LR_{min} and LR_{max} are selected by the user for each data set. The LR for the current epoch, LR_t, is decided based on that of the previous epoch, LR_{t-1}, and the relative change in $IHYLL$ with respect to its value. ϵ is a small number e.g. 0.01. The γ parameters allow dampening LR increases relative to decreases, if required; for e.g., $\gamma_1 = 0.1$ and $\gamma_2 = 1$ was used for MNIST and $\gamma_1 = 0.003$ and $\gamma_2 = 0.003$, for CIFAR-10.
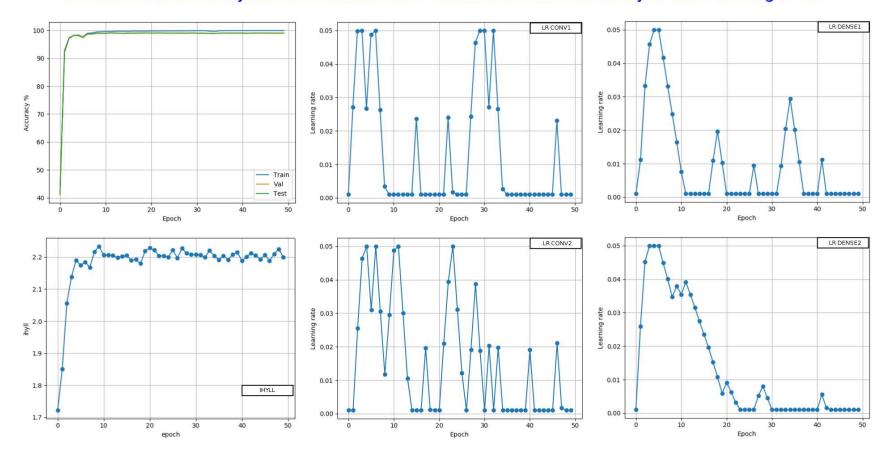
Figure 5: MNIST - dynamic LR using mutual information produces competitive to better outcomes when another hyper-parameter (e.g. batch size BS) changes during training. The base run does not perform a BS change; the other two runs are started from the 3rd checkpoint of this run. Each row in the figure shows LR vs epoch, accuracy vs epoch, IHYLL vs epoch and test-accuracy vs IHYLL.

Figure 6: MNIST - layer-wise dynamic LR using MI produces competitive outcomes in competitive time as compared to single LR training. Left column shows accuracy vs epoch and IHYLL vs epoch, middle column shows the LR vs epoch for the convolution layers and right column shows the LR vs epoch for the dense layers.

Dynamic LR policy based on change and value of $IHYLL$ relative to IXY

$$d1_t = 1 - (IHYLL_{t-1}/IXY)$$
$$d2_t = |(IHYLL_{t-1} - IHYLL_{t-2})|/IHYLL_{t-1}$$
$$LR_t = \begin{cases} min(LR_{max}, LR_{t-1} + \gamma_1 * d1_t) & (d1_t > 0 \ \& \ d2_t > \epsilon) \\ max(LR_{min}, LR_{t-1} - \gamma_2 * d1_t) & (d1_t > 0 \ \& \ d2_t \leq \epsilon) \\ max(LR_{min}, LR_{t-1} + \gamma_3 * d1_t) & (d1_t \leq 0 \ \& \ d2_t > \epsilon) \\ max(LR_{min}, LR_{t-1} + \gamma_3 * d1_t) & (d1_t \leq 0 \ \& \ d2_t \leq \epsilon) \end{cases}$$

The terms are defined as before. There are effectively two LR regimes governed by $d1_t$ being > 0 or ≤ 0; in the former the LR may increase or decrease depending on whether $d2_t$ saturates ($\leq \epsilon$) or not; the latter case involves LR reductions only. For both MNIST and CIFAR-10, LR increases occurred at the same rate as decreases - for MNIST, $\gamma_1 = \gamma_2 = 0.1$ and for CIFAR-10, $\gamma_1 = \gamma_2 = 0.001$; for both MNIST and CIFAR-10, γ_3 was set to 0.1.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL http://arxiv.org/abs/1603.05027.

[2] P. Goyal, P. Dollar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training Imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. URL http://arxiv.org/abs/1706.02677.

[3] T.M. Cover and J.A. Thomas. *Elements of Information Theory, 2nd edition*. Wiley, July 2006.

[4] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. URL http://arxiv.org/abs/1609.04747.

[5] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *in: IEEE Information Theory Workshop (ITW)*, 2015.

[6] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017. URL http://arxiv.org/abs/1703.00810.

[7] N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. In *in: The 37'th Allerton Conference on Communication, Control, and Computing*, 1999.

[8] A.M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B.D. Tracey, and D.D. Cox. On the information bottleneck theory of deep learning. In *in: International Conference on Learning Representations (ICLR)*, 2018.

[9] J. Walters-Williams and Y. Li. Estimation of Mutual Information: A Survey. In *Rough Sets and Knowledge Technology*. Springer Berlin Heidelberg, 2009.

[10] G. Doquire and M. Verleysen. A comparison of multivariate mutual information estimators for feature selection. In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, 2012.

[11] S. Khan, S. Bandyopadhyay, A.R. Ganguly, S. Saigal, D.J. Erickson III, V. Protopopescu, and G. Ostrouchov. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Physical Review E*, 76.026209, 2007.

[12] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69: 066138, 2004.

[13] A. Kolchinsky and B.D. Tracey. Estimating mixture entropy with pairwise distances. *CoRR*, abs/1706.02419, 2017. URL http://arxiv.org/abs/1706.02419.

[14] I. Belghazi, S. Rajeswar, A. Baratin, R.D. Hjelm, and A.C. Courville. MINE: mutual information neural estimation. *CoRR*, abs/1801.04062, 2018. URL http://arxiv.org/abs/1801.04062.

[15] O. Shamir, S. Sabato, and N. Tishby. Learning and generalization with the information bottleneck. *Theor. Comput. Sci.*, 411 (29-30):2696–2711, 2010.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE, 86(11):2278-2324*, November 1998.

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[18] J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The All Convolutional Net. *CoRR*, abs/1412.6806, 2015. URL `http://arxiv.org/abs/1412.6806`.