Message in a Bottle

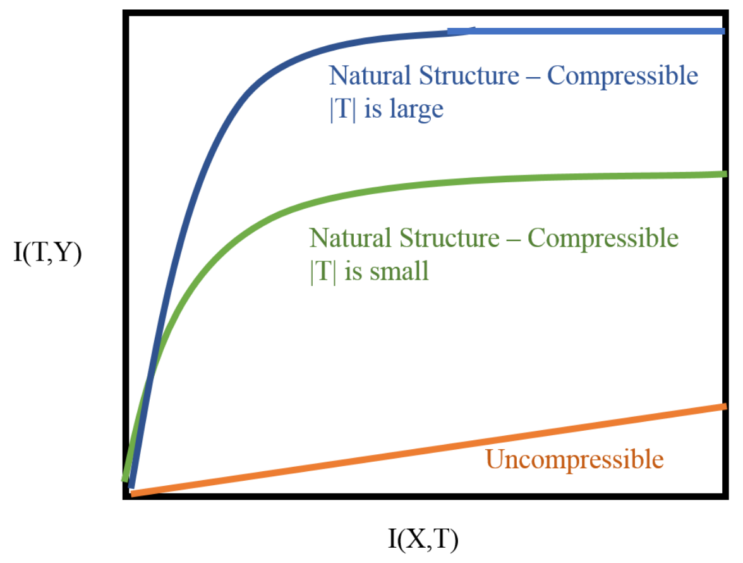
learning dynamics in the information plane

Daniel Kunin (kunin@stanford.edu)
Mansheej Paul (mansheej@stanford.edu)
Matt Bull (bullm@stanford.edu)

1. Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. ICLR 2017. URL https://arxiv.org/abs/1612.00410.
2. David Jordan, Seppe Kuehn, Eleni Katifori and Stan Liebler. PNAS 2013 14018–14023, doi: 10.1073/pnas.1308282110
3. Github Repository for neural network code: https://github.com/ravidziv/IDNNs
4. Moon Dataset: http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html
5. Ravid Shwartz-Ziv & Naftali Tishby. Opening the black box of deep neural networks via information. arXiv: 1803.00810 & \verb"https://github.com/ravidziv/IDNNs"
6. Slonim, Noam. "The Information Bottleneck: Theory and Applications." PhD Thesis Hebrew University, 2002, pp. 27–27., www.yaroslavvb.com/papers/slonim-information.pdf.
7. Suzuki, Taiji, et al. "Mutual Information Approximation via Maximum Likelihood Estimation of Density Ratio." 2009 IEEE International Symposium on Information Theory, 2009, doi:10.1109/isit.2009.5205712.
8. Tishby, Naftali, et al. "The Information Bottleneck Method" 1999 arXiv:physics/0004057v1

Introduction

The fundamental challenge of supervised learning is to navigate a trade off between compressing the representation of input features and preserving the meaningful information for prediction of the output responses. Naftali Tishby et al. describe this process as squeezing ``the information that X provides about Y through a bottleneck", which they termed the Information Bottleneck (IB) Method [8]. This method has provided a new perspective on the recent successes of Deep Learning [5].



Above is a image that depicts the achievable information bound for compressing the information in the inputs while retaining the information about the outputs [6].

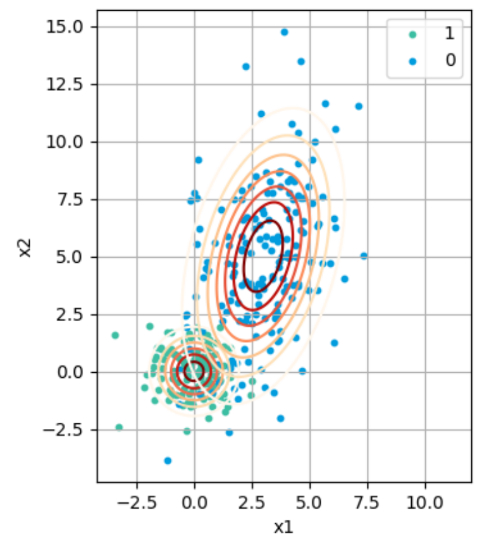
In this project we set out to explore the Information Bottleneck Method and address the following four major gaps in the current discourse:

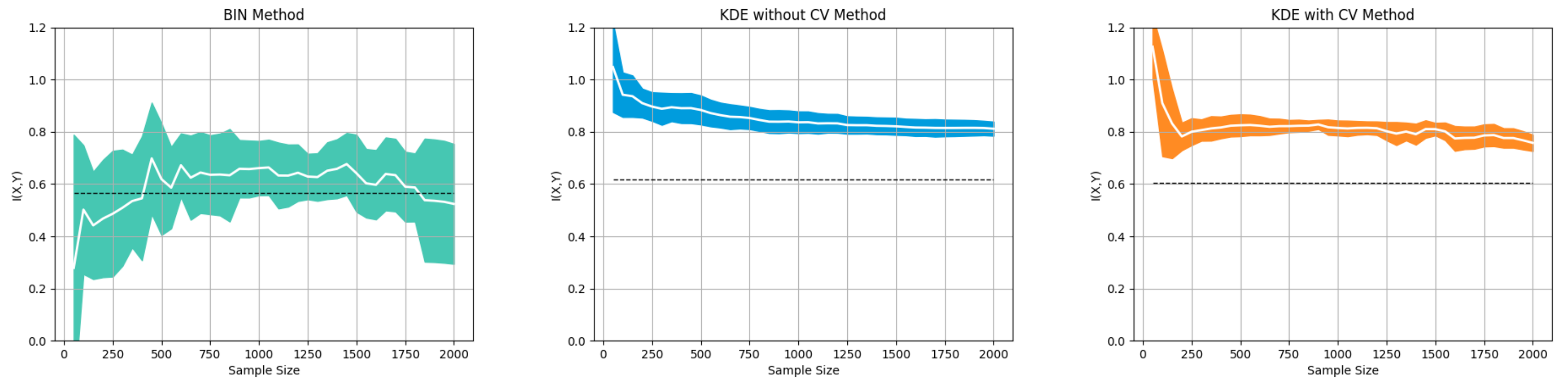
1. Discuss mutual information (MI) estimates in the context of supervised learning.
2. Explore a range of learning algorithms in the information plane and investigate the relationship between MI and training error.
3. Investigate bias vs. variance and generalization vs. compression in the information plane.
4. Use the information bottleneck to improve neural network performance by adapting the learning strategy to the learning phase.

Mutual Information Estimation

The greatest challenge to using the Information Bottleneck Method is constructing consistent, distribution invariant, high dimensional estimators of MI. We found seven distinct methods for MI estimation in the literature [7].

Below are the results of two estimation methods we implemented and a cross validated variant. We evaluated the performance of these estimators using data simulated from a mixture of Gaussians. To the right is the dataset that we used for these experiments.

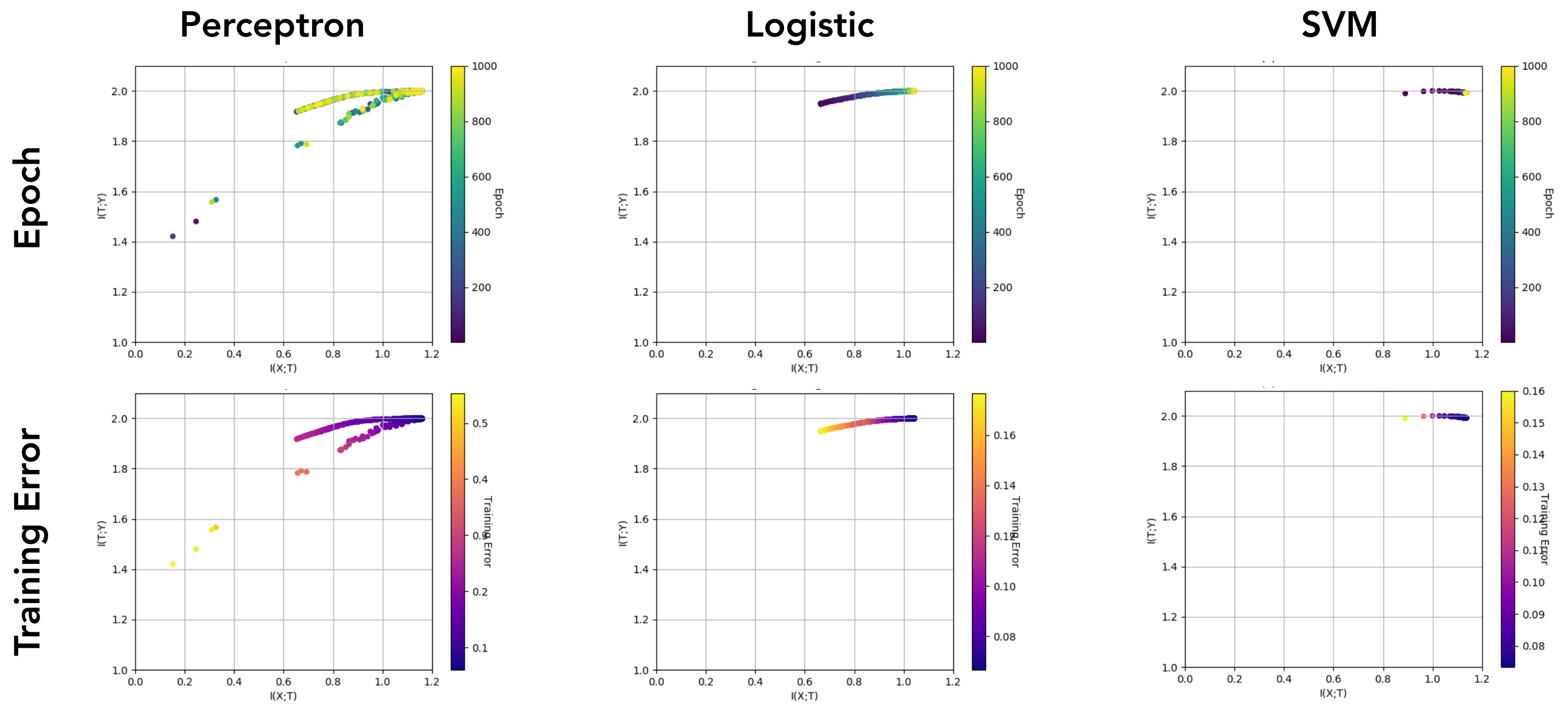




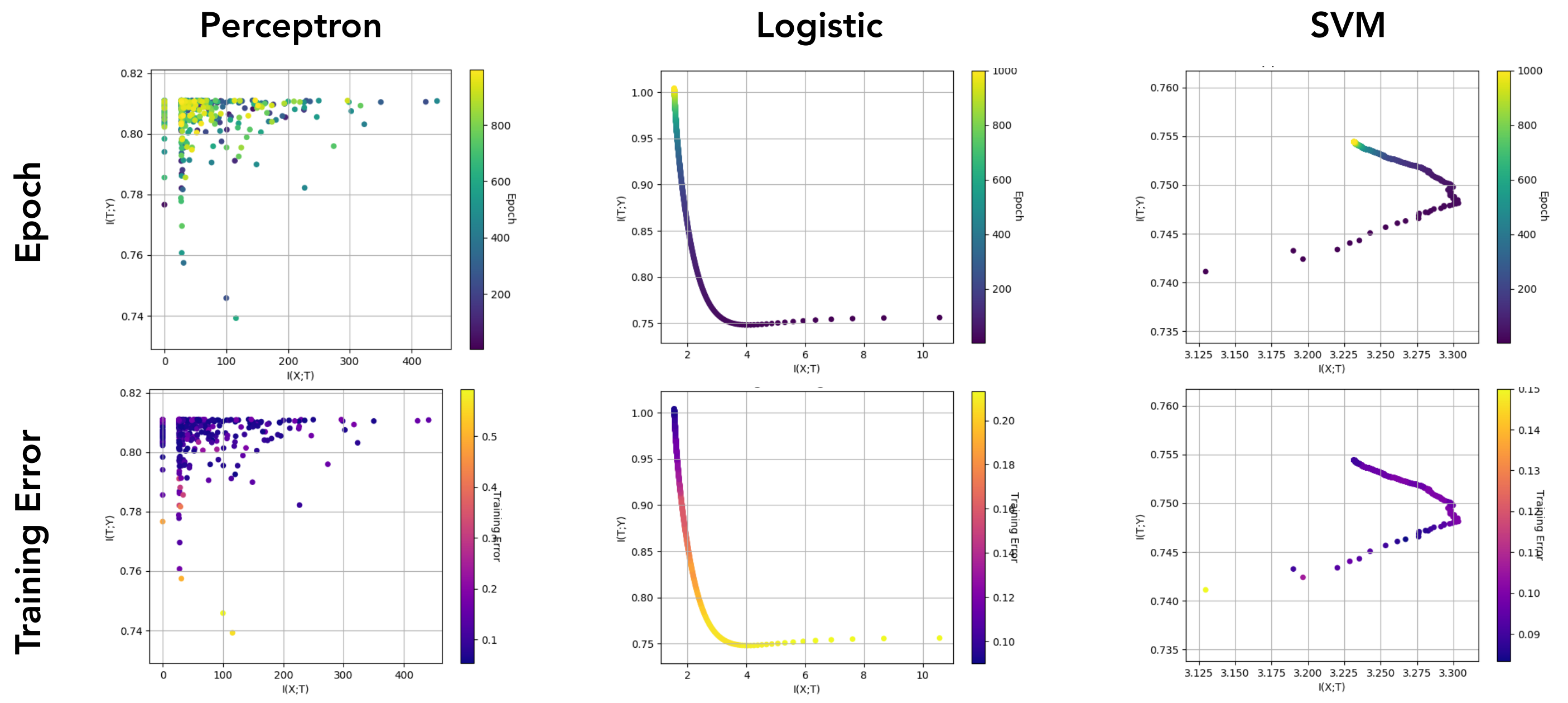
The dotted line is the true MI solved by a Monte Carlo integrator. Left: the adaptive binning estimate with a first order correction [2] is unbiased but suffers from large variance. Center: a simple kernel density estimate shows significant bias. Right: a cross validated KDE improves the estimate, but suffers from residual bias. For the next section we use the simple KDE estimator because of its consistency and speed of computation. For the neural networks, we build upon the code base at [3].

Information and Error

We examined the Percepton, Logistic Regression and Support Vector Machine in the Information Bottleneck framework with the same sample dataset shown perviously.

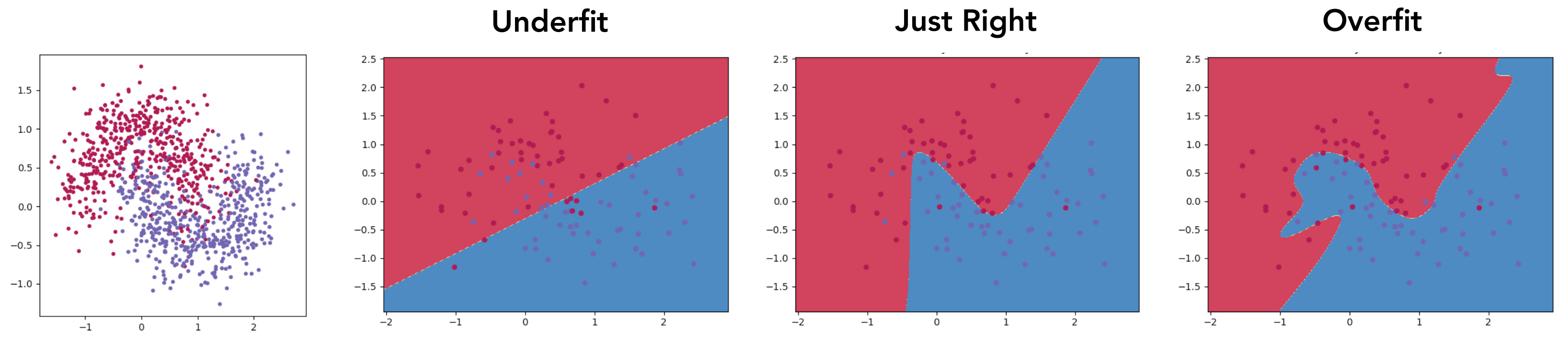


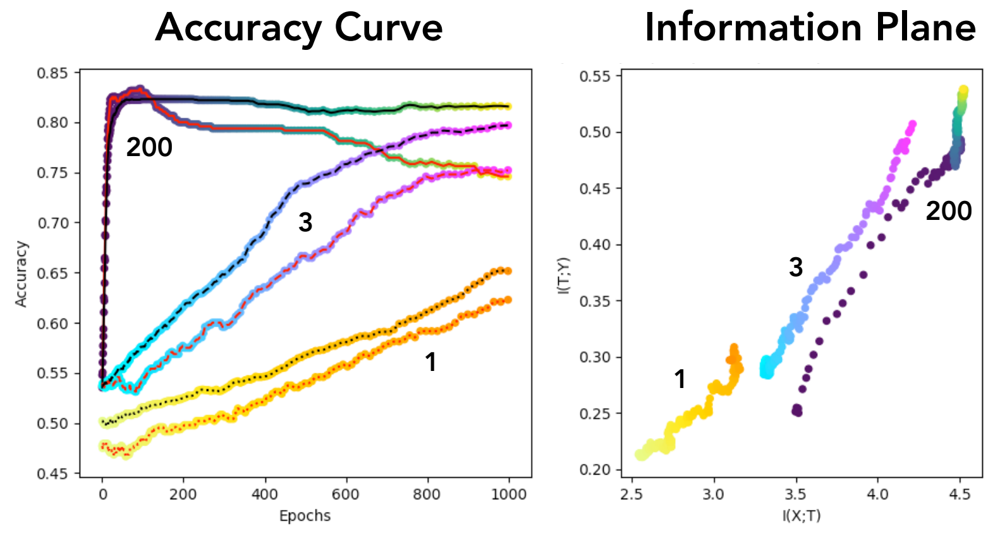
The top plots show the information plane where the representations, T, are the predictions. In the bottom plots, more interesting dynamics emerge when we use the hidden state (logistics, perceptron) and margin (SVM) as the representations. In these, we observe compression of the inputs.



Bias vs Variance for IB

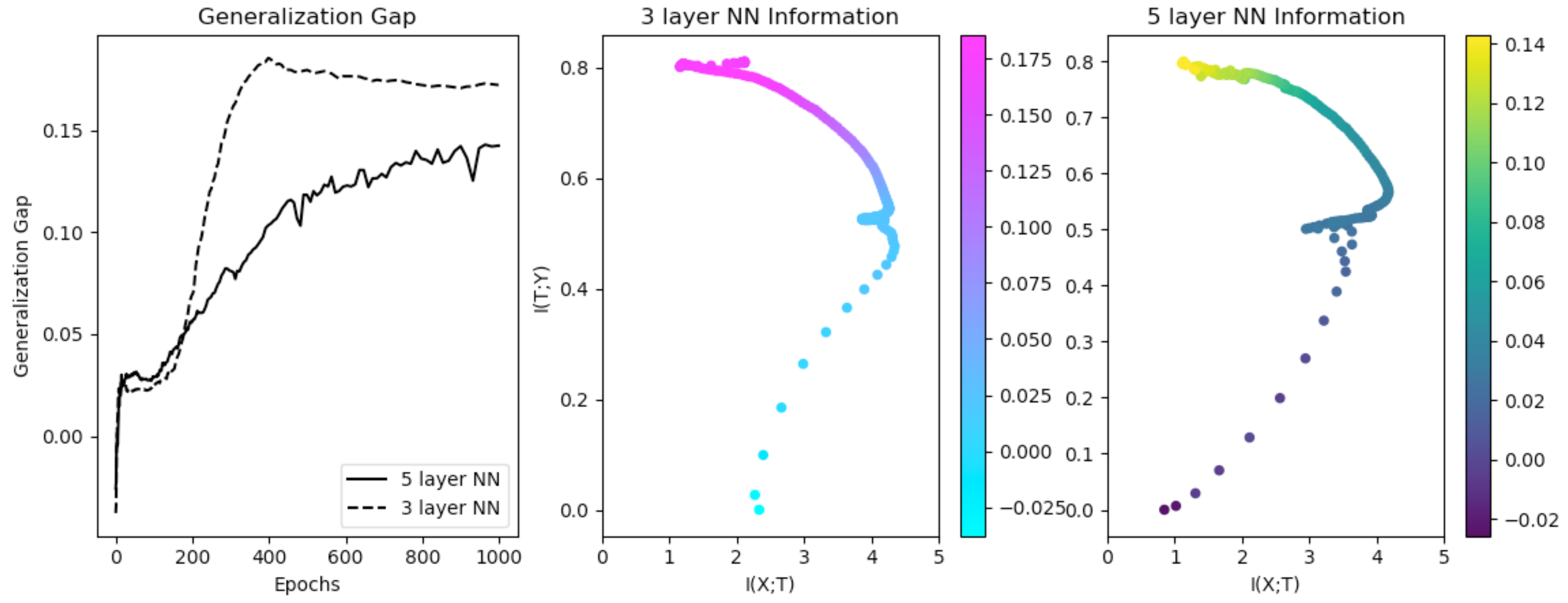
We use the moon dataset from Scikit-learn [4] and 1 hidden layered Neural Networks with 200, 3 and 1 hidden neurons (averaged over 25 runs each) to demonstrate overfit, a "just right" fit and underfit respectively in the information plane.





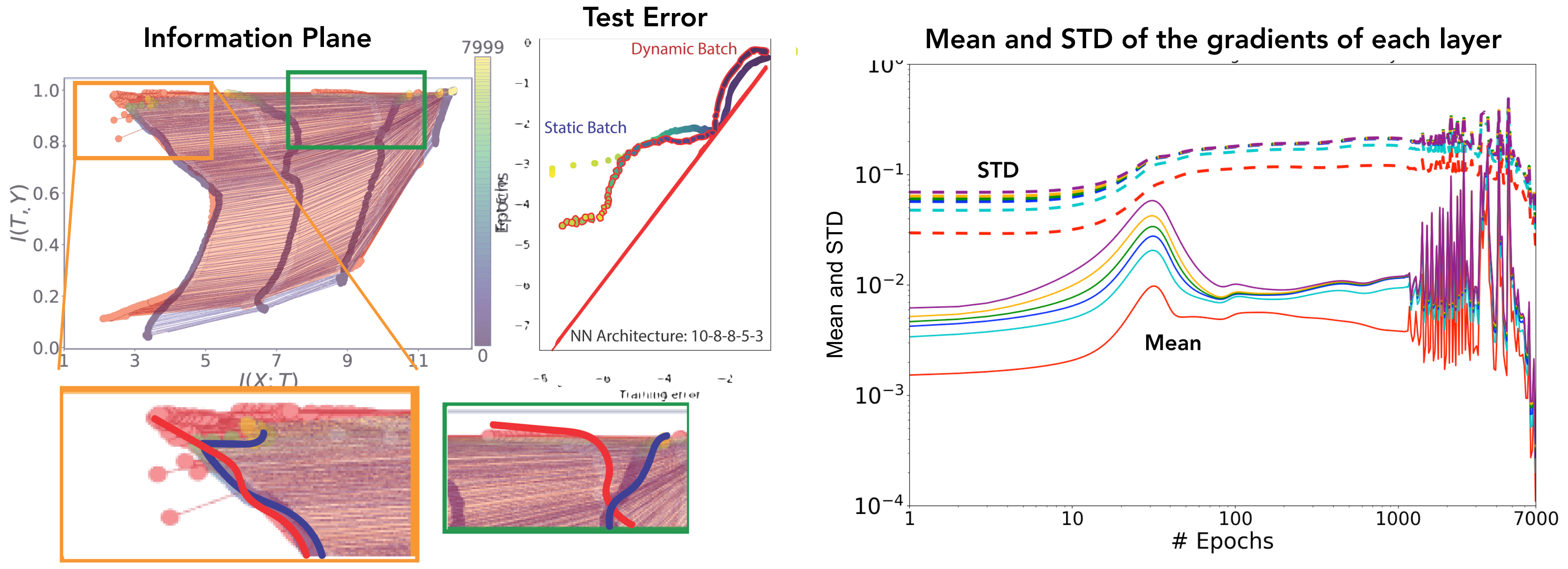
As we tradeoff bias for variance, we observe the learning trajectories move right and up in the information plane. The neural networks with 3 and 200 hidden neurons have similar test error but the network with 3 neurons has lower generalization error and less I(X,T).

Generalization Gap vs Compression



We add 100 dimensions of Gaussian noise to our moon data set and use it to train 3 and 5 layered deep networks. The generalization gap is the train minus test error and is the colorbar metric. To generalize well, the networks need to compress the relevant information from the high dimensional data. We observe that the 3 layered network has worse generalization and ends up with a higher I(X,T); it is not compressing as well. Deeper networks compress better [5]. However, even as generalization worsens the networks compress indicating that this relationship is not straightforward.

Dynamic Batch Algorithm



The information plane suggests a phase change between two phases of learning: rapid error minimization and compression [5]. The signature of compression is an order of magnitude growth of the STD of the gradients. At the critical point, we artificially increase the STD of the gradients by reducing batch size. This led to to a jump decrease in test error (compared to a plateau without the adaptive batch size) and a corresponding decrease in I(X,T) at multiple layers. Intuitively, our method adds noise during the compression phase which serves as regularization.

Future Work

For further work, we propose implementing the variational MI estimation [1], investigating the reliability of the dynamic batch algorithm and exploring other strategies for leveraging the two phases of learning.