Skepticism in Neural Networks: Experiments in [Variational] Information Bottleneck Theory

Ilan Sharon

University of Michigan
isharon@umich.edu

December 15, 2025

Abstract

Deep neural networks are powerful function approximators but often suffer from overfitting and a tendency to memorize noise. The Information Bottleneck (IB) theory frames learning as a trade-off between compression and prediction (forgetting irrelevant information while retaining enough for prediction). This project investigates the Variational Information Bottleneck (VIB) as a practical, approximate, application of this principle. Through a series of experiments on the MNIST dataset, we evaluate VIB's robustness to Gaussian pixel noise and label corruption compared to a deterministic baseline. Our results suggest that VIB is not a perfect solution for all purposes. It can, however, be helpful in specific yet realistic situations, significantly outperforming standard models in rejecting corrupted labels by penalizing the information cost of memorization.

1 Introduction

It is no secret that neural networks have grown increasingly powerful in recent years. In doing so, however, they have also grown more complex, meaning we know less and less about their inner workings. One useful framework for observing these black box models uses information theory to illustrate information flow through the network. We follow the inception of this theory, and look into some practical extensions. The field has progressed quite a bit beyond the scope of this project in recent years, but we are nonetheless able to make interesting observations regarding some of the early findings.

2 Theoretical Foundations

The following claims in this study are built on Information Theory foundations such as Entropy and Mutual Information. Before getting into the theory itself, we introduce the mathematical tools which are fundamental to understanding what follows, both conceptually and mathematically. We treat a neural network as a series of random variables: X representing the input, T the hidden layers (with Z representing specifically our hidden bottleneck layer), and Y the output.

2.1 Information Theoretic Basics

To quantify the information content of these variables, we utilize Shannon Entropy and Mutual Information. For a random variable X with probability distribution $p(x)$, the entropy $H(X)$ measures the uncertainty associated with X:

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x) \tag{1}$$

Building on this, Mutual Information $I(X;Y)$ quantifies the reduction in uncertainty about one variable given knowledge of another:

$$I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = H(X) - H(X|Y) \tag{2}$$

This value is most fundamental to the theory. Mutual Information can be thought of as a measure of how much one variable knows about another. It is symmetric, so:

$$I(X;Y) = I(Y;X) \tag{3}$$

Probabilistically, it can be viewed as how far two marginal distributions are from being independent.

2.2 The Information Bottleneck Principle

Tishby et al. [1] formulated supervised learning as an optimization problem such that an optimal representation Z must satisfy two competing goals:

1. **Sufficiency:** It must aim to capture information in X that is relevant to predicting Y (maximize $I(Z;Y)$).

2. **Minimality:** It should discard information in X that is irrelevant to Y (minimize $I(X;Z)$).

This trade-off is formalized as the minimization of the Information Bottleneck Lagrangian:

$$\mathcal{L}_{IB} = I(X;Z) - \beta I(Z;Y) \tag{4}$$

where β is like a Lagrange multiplier controlling the trade-off. However, directly optimizing Equation 4 is computationally intractable for most scenarios in a deep neural network, as it requires knowledge of the true joint distribution $p(X,Y,Z)$.

2.3 The Information Plane

This plot of the 'Information Plane' is helpful for visualizing learning dynamics in the network. We can clearly see that accuracy increases throughout the training process, but there is an interesting regime shift on the compression axis. After some time training, the model undergoes a 'forgetting phase', during which it is able to discard the redundant information about its inputs, and hopefully become more robust.

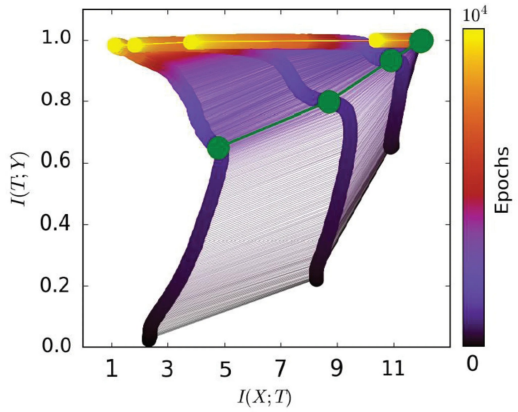


Figure 1: The Information Plane

2.4 Variational Information Bottleneck (VIB)

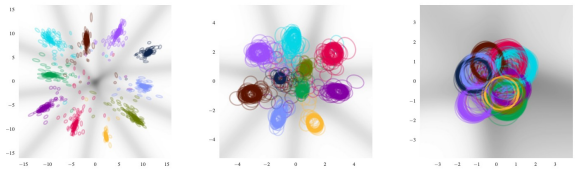
To make the IB principle practical for deep neural networks, Alemi et al. [2] introduced a variational approximation, the 'Variational Information Bottleneck'. The model is the same up to the bottleneck layer. Upon reaching Z, instead of a typical linear transformation, we take statistics on the distribution $p(z|x)$, and sample Z as a Gaussian with these statistics. The mutual information term $I(X;Z)$ is then bounded by the Kullback-Leibler (KL) divergence between the encoder posterior $p(z|x)$ and a marginal prior $r(z)$, typically chosen as a standard Gaussian $\mathcal{N}(0,I)$.

This yields the tractable VIB loss function used in our experiments:

$$J_{VIB} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\epsilon}[-\log q(y_n|z_n)] + \beta \mathrm{KL}[p(z|x_n)||r(z)] \tag{5}$$

The first term corresponds to the standard Cross-Entropy loss (prediction), while the second term acts as a regularizer, penalizing the complexity of the representation by forcing it towards the prior.

The choice to introduce stochasticity directly into the training process here is not an intuitive one. A useful analogy comes from a visual interpretation of Z. Imagine flattening the many-dimensional vector Z into two dimensions, and plotting it as a point in the plane:



(a) $\beta = 10^{-3}$, $\mathrm{err_{mc}} = 3.18\%$, $\mathrm{err_1} = 3.24\%$ (b) $\beta = 10^{-1}$, $\mathrm{err_{mc}} = 3.44\%$, $\mathrm{err_1} = 4.32\%$ (c) $\beta = 10^{0}$, $\mathrm{err_{mc}} = 33.82\%$, $\mathrm{err_1} = 62.81\%$.

Figure 2: Z-clouds across β

Really, these are not just Z but $Z|X$. Each different color represents being conditioned on a different X. As we can see, increasing our compression parameter, β, causes the clouds to cluster more towards the center. This is because in our VIB loss function, the KL divergence term punishes them for drifting too far from the origin.

3 Methodology

To isolate the effects of introducing VIB into our training process, we train two neural networks on the MNIST dataset. Our base model is a traditional Multilayer Perceptron. Our VIB model is essentially the same, with the key alteration of the non-deterministic Z sampling allowing for the use of Information Bottleneck in the loss function.

3.1 Model Architectures

Both models share an identical backbone to ensure fair comparison. The encoder consists of an input layer (784 units), followed by two hidden layers (256 and 128 units) with ReLU activations.

- **Baseline (MLP):** The bottleneck layer Z is a deterministic vector of dimension $d = 8$. The network directly maps input x to a point z.

- **VIB Model:** The bottleneck layer outputs parameters for a Gaussian distribution: a mean vector μ and a variance vector σ, allowing for the z-sampling to occur.

3.2 Training and Evaluation Protocol

Both of the models are trained using the Adam optimizer with a learning rate of 10^{-3}. For the VIB model, we introduce the bottleneck coefficient β (typically set to 10^{-2}) to best balance the trade-off between the Cross-Entropy loss and the KL Divergence penalty.

Stochastic Inference: Unlike the traditional MLP, the VIB model is slightly probabilistic. During evaluation, we perform Monte Carlo sampling ($N = 12$ samples per image) to approximate the predictive distribution $p(y|x) \approx \frac{1}{N} \sum p(y|z_i)$. This ensures that we evaluate the robustness of the full learned distribution, not just the mean. This technique is adopted directly from Alemi et al. [2]

4 Experiments and Results

4.1 Clean Training

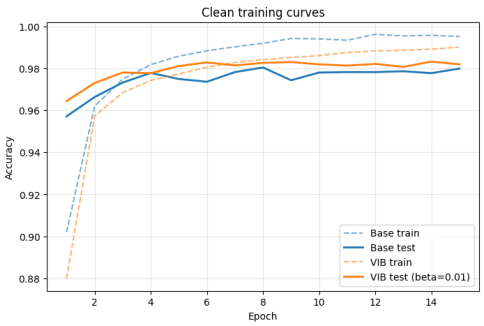


Figure 3: Clean Training Accuracies

Before getting into experiments, we investigate a clean version of both models. Over 15 epochs of training, we can see that the base model outperforms VIB on training data, but underperforms on test data. The numbers are not strong here, but this is nonetheless a promising sign that the VIB model may be more resistant to overfitting.

4.2 Exp 1: Generalization and Input Noise

The first experiment we investigate is how the clean-trained model can handle noise in its inputs during evaluation. Simply put, we trained the model on the dataset without any noise. Then evaluated the model on data with an increasing amount of Gaussian noise in its inputs. We plot model performances with respect to the amount of noise (standard deviation):

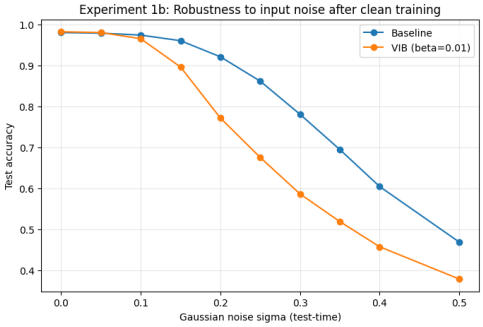


Figure 4: Experiment 1: Clean Training + Noisy Eval

Here, the base model significantly outperforms VIB. This result seems counterintuitive, but it is possible that forcing the model to forget input information can make it more sensitive to small pixel shifts.

4.3 Exp 2: Adaptation to Noisy Training

In the second experiment, we extend the first to also include noise during the training process. The VIB loss function is specifically built to filter out noisy data. Training the model on perfectly clean data, and evaluating on noise, almost entirely defeats the purpose of VIB. So we repeat the same experiment, with the caveat that now the X-axis represents noise included in inputs throughout both the training and evaluation phases:

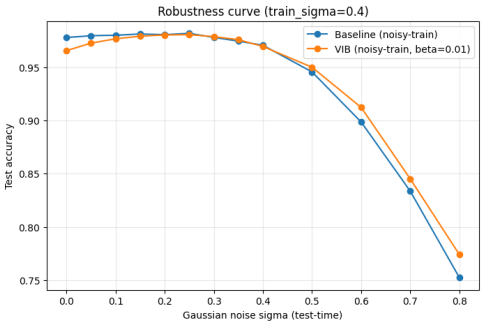


Figure 5: Experiment 2: Noisy Training + Noisy Eval

We are starting to see a stronger result here. The models perform quite similarly on this data, which is at the very least an improvement from the first experiment. There is a slight trend of higher accuracy in the VIB model throughout the noisier runs, but overall performance is similar.

4.4 Exp 3: Robustness to Label Corruption

The third and final experiment truly probes a model's ability to remain "skeptical". We implant a series of lies in the data. An image that should be labeled 1, for example, will be changed to a different label randomly

with probability p. We sweep over values of p, and evaluate the ability of our models to adjust to the lies, and identify the true images in the evaluation phase:

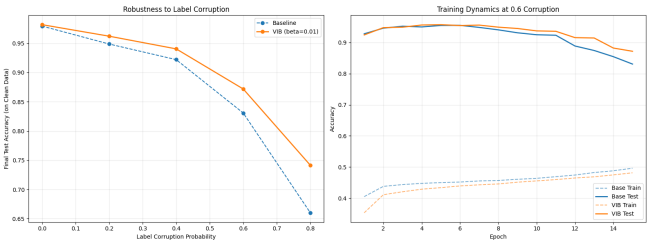


Figure 6: Experiment 3: Corrupted Training

This is the strongest result thus far. The VIB model is clearly able to resist memorizing incorrect labels more so than the base model. Furthermore, we can see that during training, the base model performs better than VIB on the training data with $p = 0.6$, but is outperformed by VIB on test data. This is a real application, it is often the case that data is riddled with errors, and VIB has shown an ability to filter these errors out at a higher rate than a traditional model.

5 Conclusions

It is clear that VIB is not a universal solution. We have identified instances in which VIB can have the opposite intended effect - increased sensitivity to noise, and instances in which it can perform exactly as intended.

It is reasonable to assume that while compression makes the model more efficient, it also makes it more brittle to perturbations it has not been trained on. The intention of VIB, however, is more focused on training noise out *during* the training process. When testing this, we found that as expected, it is able to generalize better than a base model training purely on cross entropy.

Ultimately, if nothing else, the information bottleneck is a useful tool in illuminating the mind of a neural network during training, allowing us to build models that don't just learn data, but understand it.

References

[1] Tishby, N., & Zaslavsky, N. (2015). *Deep learning and the information bottleneck principle*. IEEE Information Theory Workshop.

[2] Alemi, A. A., et al. (2017). *Deep Variational Information Bottleneck*. ICLR.

[3] Saxe, A. M., et al. (2019). *On the information bottleneck theory of deep learning*. ICLR.