Understanding Learning Dynamics of Binary Neural Networks via Information Bottleneck

Seminar for the requirement of EE6999 and EE7999

February 4, 2024

Indian Institute of Technology Madras, India

Outline

Information Bottleneck (IB) Principle

Information Bottleneck

- Information Bottleneck method is information theoretic principle for extracting relevant information that an input RV $X \in \mathcal{X}$ contains about an output RV $Y \in \mathcal{Y}$
- Given the joint distribution $p(X, Y)$, the relevant information is $I(X, Y)$
- X and Y are dependent, so mutual information $I(X; Y) > 0$
- Optimal representation of X would capture relevant features of X for predicting Y, and compress irrelevant features

Minimal Sufficient Statistics

- In supervised learning, we are interested in a good representation $S(X)$ which is the relevant part of X with respect to Y
- In the DNN setting, $S(X)$ is the partition of X that has all the information X has on Y, i.e., $I(S(X); Y) = I(X; Y)$
- The optimal representation is best characterized by minimal sufficient statistics, the coarsest partition of input space X wrt Y
- Finding the **minimal sufficient statistics** $T(X)$ is:

$$T(X) = \underset{S(X): I(S(X);Y) = I(X;Y)}{\arg\min} I(S(X); X). \tag{1}$$

- Exact minimal sufficient statistics may not exist

Minimal Sufficient Statistics (Contd.)

- Relaxed optimization problem is to find the approximate minimal sufficient statistics that captures as much $I(X;Y)$ as possible
- Trade-off between the compression of X and the prediction of Y
- Pass the information that X provides about Y through a **bottleneck**[1] formed by the compact summaries in $T(X)$
- Finding the compressed representation T of X becomes minimizing the below functional:

$$\mathcal{L} = I(T;X) - \beta I(T;Y), \tag{2}$$

  where β is the Lagrange multiplier
- Here $\beta = \infty$ implies no compression and vice versa

[1] Tishby, Naftali, Fernando C. Pereira, and William Bialek. "The information bottleneck method." arXiv preprint physics/0004057 (2000).

IB principle for Deep Neural Networks

- Structure of the DNN is reviewed as a Markov cascade of intermediate representations between input and output layers[2]

$$Y \rightarrow X \rightarrow T_1 \rightarrow T_2 \rightarrow \cdots \rightarrow T_j \rightarrow T_i \rightarrow \cdots \rightarrow \hat{Y} \quad (3)$$

- Let the set of hidden layers in a DNN is defined by $\mathcal{T}$ and T_i denotes i^{th} hidden layer and $i > j$, then according to data processing inequality (DPI)

$$I(Y;X) \geq I(Y;T_1) \geq I(Y;T_2) \geq \ldots I(Y;T_j) \geq I(Y;T_i) \cdots \geq I(Y,\hat{Y}) \quad (4)$$

- Achieving equality is possible iff each layer is a sufficient statistic of its input

[2] Tishby, Naftali, and Noga Zaslavsky. "Deep learning and the information bottleneck principle." In 2015 ieee information theory workshop (itw), pp. 1-5. IEEE, 2015.

Learning dynamics of Deep Neural Networks

Information plane dynamics of DNNs

- Information Plane: The plane of the Mutual Information values that each layer preserves on the input and output variables
- Goal of the network is to optimize the Information Bottleneck (IB) tradeoff between compression and prediction
- By using tanh activation, Deep networks are shown to undergo two distinct phases[3]
  - Empirical Risk Minimization phase where the stochastic gradient descent (SGD) algorithm generates high valued gradients, the loss rapidly decreases
  - Compression phase where the efficient representation of the intermediate layers are learned - higher variance gradient

[3] Shwartz-Ziv, Ravid, and Naftali Tishby. "Opening the black box of deep neural networks via information." arXiv preprint arXiv:1703.00810 (2017).

Information plane dynamics of DNNs (Contd.)

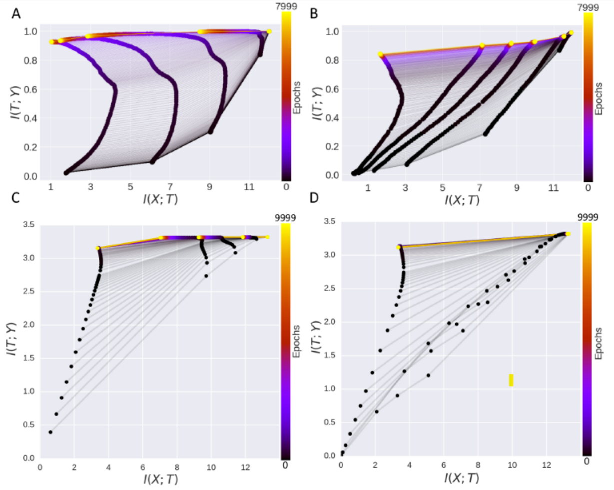


Figure 1: Information plane dynamics and neural nonlinearities. A. Tanh, binning; B. ReLU, binning; C. Tanh, KDE; D. ReLU, KDE

Information plane dynamics of DNNs (Contd.)

- Information plane trajectory is a function of the neural nonlinearities[4]:
  - double-sided saturating nonlinearities like tanh yield a compression phase
  - no evident causal connection between compression and generalization

[4] Saxe, Andrew M., Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. "On the information bottleneck theory of deep learning." Journal of Statistical Mechanics: Theory and Experiment 2019, no. 12 (2019): 124020.

Does compression really depend on double saturating non-linearity?

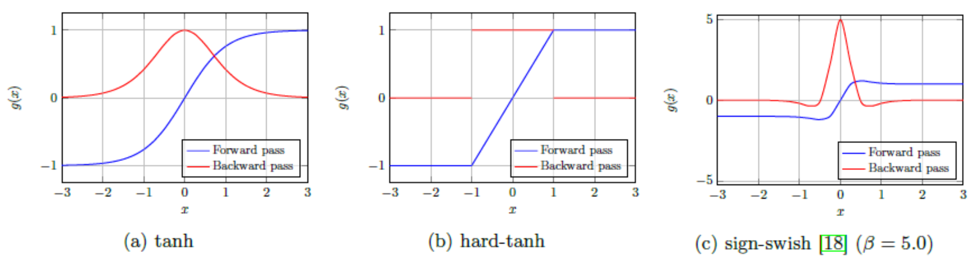


Figure 2: Activations considered

- We do not observe a compression phase for double saturating sign-swish activation

Information plane behaviour for DNN

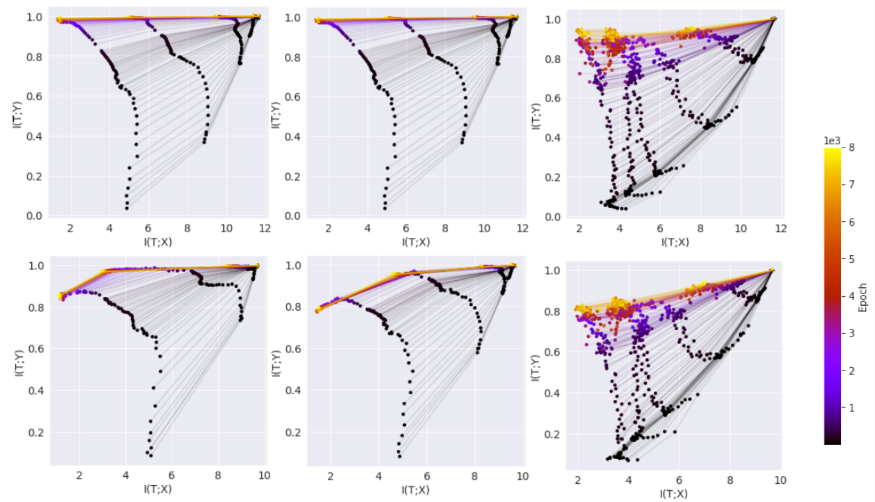


Figure 3: Top: Training data, bottom: test data; Left: Tanh, Middle: Hard-tanh, Right: Sign-swish

Information plane behaviour for DNN

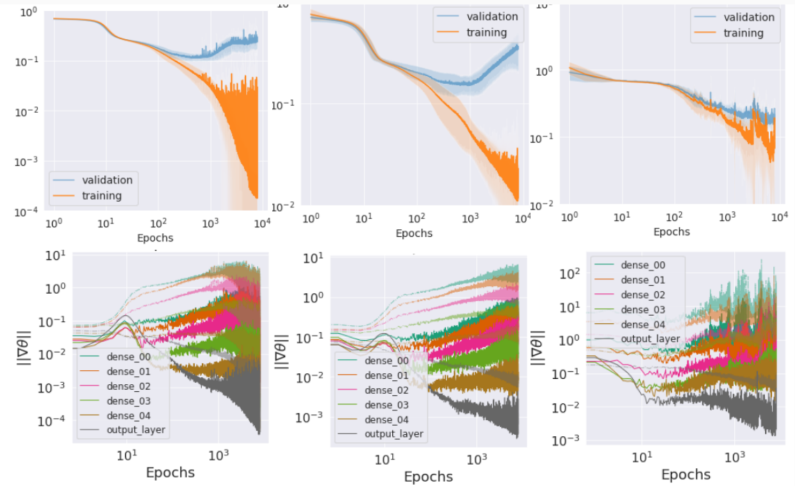


Figure 3: Top: Loss, bottom: gradient; Left: Tanh, Middle: Hard-tanh, Right: Sign-swish

Information plane behaviour for DNN (Contd.)

- Decrease in $I(T;Y)$ is prominent after 1000 epochs for both the activations tanh and hard-tanh
- Increase in validation loss around 1000 epochs - overfitting
- In sign-swish, over-fitting is less
- DNNs first increase both $I(T;X)$ and $I(T;Y)$ followed by a separate representation compression (RC) phase where $I(T;X)$ decreases
- Representation compression phase is slow process in DNN, and often happens once loss starts saturating
- When we use methods like early stopping, practical models may never get to the compression phase
- The compression phase of DNNs is seen as generalization and this is not achievable unless models are trained well beyond loss saturation and are at the risk of overfitting

Learning dynamics of Binary Neural Networks

Information bottleneck for BNN

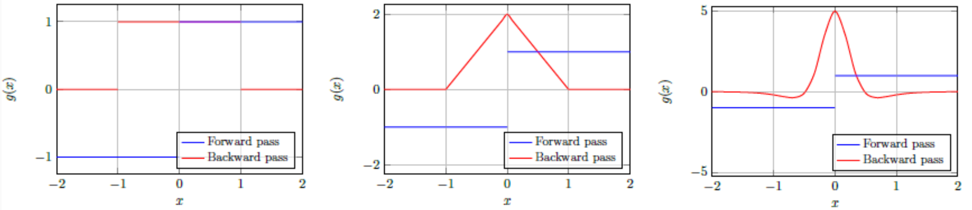
- The intermediate representation T of a real-valued neural network can have high precision and hence can accommodate any shorter representation of the input - information flows without any hindrance
- $I(T;Y)$ needs to be kept at a certain level for correct prediction of Y
- Representation capability of T is limited due to binary activation in BNNs - free flow of complete information is suppressed
- It is of immense interest to study the learning dynamics of BNNs

Binary Neural Networks

- The binary activation function $g(\cdot)$:

$$g(x) = \begin{cases} -1 & ; x \leq 0, \\ +1 & ; x > 0, \end{cases} \quad (5)$$

- backpropagation requires differentiable activation functions



(a) STE [8] (b) Approximate sign [17] (c) Swish sign [18] ($\beta = 5.0$)

Figure 4: Activations considered in BNN

Information plane behaviour for BNN

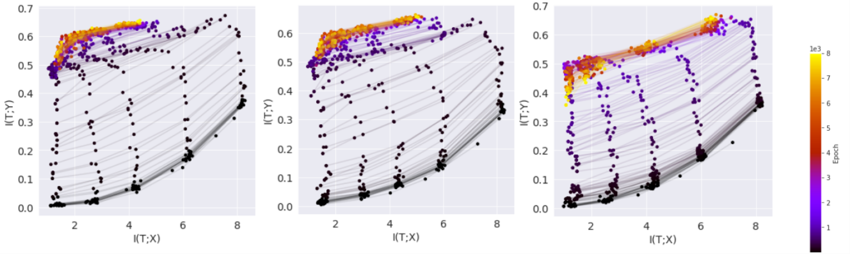


Figure 5: Left: STE activation, Middle: Approximate sign activation, Right: Swish sign activation

Information plane behaviour for BNN

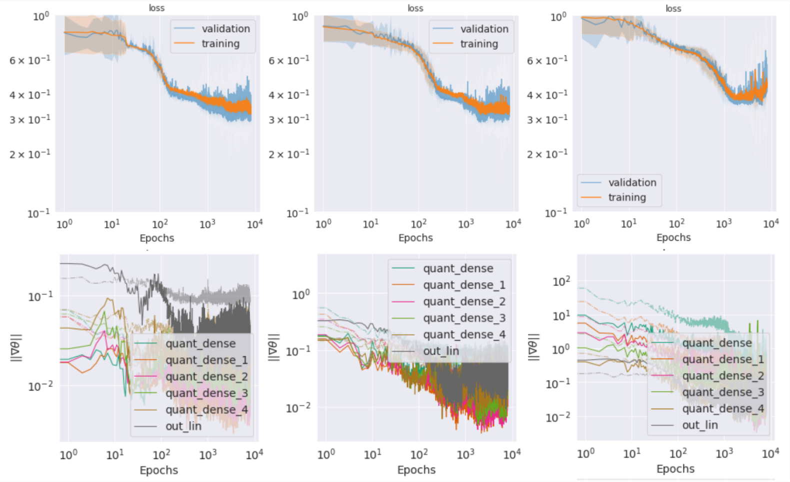


Figure 5: Top: Loss, bottom: gradient; Left: STE activation, Middle: Approximate sign activation, Right: Swish sign activation

Information plane behaviour for BNN (Contd.)

- BNNs start with a low value for $I(T;X)$ and does not show an explicit compression phase
- The behavior of high gradient variance in DNN in each epoch is similar to the noise[5] and this facilitates the generalization in DNNs
- BNNs do not have high gradient variance phase, yet they generalize well
- High variance in gradients alone cannot characterize the representation compression (RC) phase
- No explicit RC phase for BNNs
- BNNs generalize over the dataset rather than extracting features that may be specific for individual samples
- During training they spend time on improving task-relevant mutual information $I(T;Y)$

[5] Shwartz-Ziv, Ravid, and Naftali Tishby. "Opening the black box of deep neural networks via information." arXiv preprint arXiv:1703.00810 (2017).

Conclusion

- Even though the DNNs have a separate empirical risk minimization and representation compression phases, in BNNs, both these phases are simultaneous
- BNNs have a less expressive capacity, they tend to find efficient hidden representations concurrently with label fitting
- Verified across different activation functions

Thank you!

Our study

In this experiment, we study DNN with three activation functions tanh, hard-tanh and sign-swish. The activation hard-tanh is given by

$$g(x) = \begin{cases} -1 & ; x \leq -1 \\ x & ; -1 \leq x \leq +1, \\ +1 & ; x \geq 1. \end{cases} \tag{6}$$

We take another double saturating non-linearity, sign-swish, given by

$$g(x) = 2\sigma(\beta x)\left(1 + \beta x\left(1 - \sigma(\beta x)\right)\right) - 1. \tag{7}$$

where σ is the sigmoid function, β is a tunable parameter.

BNN activations

Straight-Through-Estimator (STE): STE-sign is used in [**courbariaux2016binarized**] to train BNNs using backpropagation. The backward pass for STE is defined as,

$$\frac{d}{dx}g(x) = \begin{cases} 1 & ; -1 \leq x \leq +1, \\ 0 & ; \text{otherwise.} \end{cases} \tag{8}$$

Approximate sign: [**liu2018bi**] introduced Approximate sign (approx-sign) function as a tight approximation to the derivative of the non-differentiable sign function with respect to activation. The backward pass for approximate sign activation function is defined as,

$$\frac{d}{dx}g(x) = \begin{cases} 2 - 2|x| & ; -1 \leq x \leq +1, \\ 0 & ; \text{otherwise.} \end{cases} \tag{9}$$

BNN activations (contd.)

Swish sign: [**darabiregularized**] proposed swish sign activation as another close approximation for the sign function. The backward pass for swish sign activation function is defined as,

$$\frac{d}{dx}g(x) = \frac{\beta\left(2 - \beta \tanh\left(\frac{\beta x}{2}\right)\right)}{1 + \cosh\left(\beta x\right)}. \tag{10}$$