Why Deep Learning Works?

Manvi Gupta

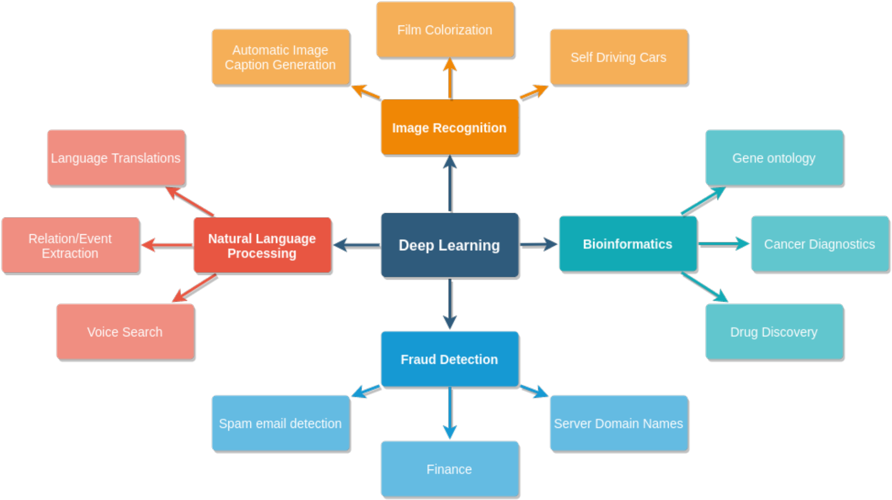
Indian Institute of Technology Mandi

b17092@students.iitmandi.ac.in

June 2021

Overview

Deep Learning Application Areas



Introduction (Contd.)

But, many questions still remain unanswered...

- Why does one model outperform the other?
- How many layers should be used?
- What should be the minimum size of the dataset?
- ... and so on

Information Theoretic Approach

- Based on *Rate Distortion Theory*.
- **Information Bottleneck Trade off**:
  - Compression Vs. Prediction

The IB Principle

Extract relevant information that an input random variable X contains about the output random variable Y.

Problem: No established theory to prove if the Information Bottleneck Principle is universally true.

Objective & Scope of Work

Long Term Objective:
Computational approach to evaluate the universality of the *Information Bottleneck Principle*, by varying the DL Architecture, Activation Function, Dataset and MI Estimator.

Part of current Project:
Mutual Information & estimation techniques

- Varying the MI Estimator, keeping other parameters fixed.
- Implementation of EDGE estimator.
- Comparing the Mutual Information trends.

Mutual Information

Entropy

Measure of uncertainty of a random variable.
$H(X) = -\sum_{x \in X} p(x) log_2(p(x))$.

Mutual Information (Relative Entropy)

The amount of information a random variable contains about another.

$$I(X;Y) = D_{KL}[p(x,y)||p(x)p(y)] = \sum_{x \in X, y \in Y} p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$$
$$= H(X) + H(Y) - H(X,Y)$$

- X and Y are the random variables with a joint distribution of $p(x,y)$.
- $D_{KL}[p||q]$ = Kullback-Liebler divergence of the distributions p and q.
- $H(X)$ = entropy of X.

The EDGE Estimator

Scalable Mutual Information Using Dependence Graphs

The paper[1] proposes *Ensemble Dependency Graph Estimator (EDGE)*

- Achieve optimal computational complexity of precise MI estimation.
- Combines LSH and dependence graphs
- Data points are mapped to integer values using a randomized LSH function.

---

[1] M. Noshad and Alfred O. Hero III, "Scalable Mutual Information Estimation Using Dependence Graphs",
CoRR, vol. abs/1801.09125, 2018.

Dependence Graph[2]

- Each node is assigned a weight, $w_i \propto$ no. of hash collisions.
- Each edge between the vertices v_i and u_j also has a weight, $w_{ij} \propto$ the no. of (X_k, Y_k) pairs mapped to (v_i, u_j).

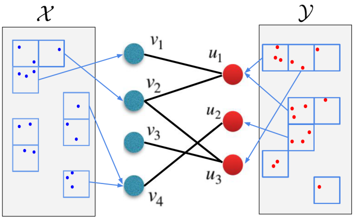


Figure: Sample dependence graph with 4 and 3 respective distinct hash values of X and Y data jointly encoded with LSH, and the corresponding dependency edges.

[2] Figure taken from [Noshad and Hero, 2018]

Hashing in EDGE

Hashing forms a major part of EDGE.

1. **Element-wise Hashing:** Maps each component of the vector to a corresponding value in the integer domain

$$h1(x_i) = \left\lfloor x_i * c * HashTableSize \right\rfloor \% HashTableSize \tag{1}$$

2. **Random Hash function for dimension reduction:** Map vector to the range $1, 2, 3, .., F$, where F is a fixed tunable integer. *Our implementation*:

$$H_2(\vec{x}) = x_1 \oplus x_2 \oplus ... \oplus x_d \tag{2}$$

Final Expression

Vector Valued Hashing: $H(x) = H_2(H_1(x))$

Mutual Information Estimation

MI Dependence Graph Estimator

$I(X;Y) := \sum_{e_{ij} \in E} w_i w\prime_j \hat{g}(w_{ij}),$

- E = Set of Edges of dependence graph.
- $\hat{g}(x) = min(g(x), Upperbound)$.
- $w_i = \frac{N_i}{N}$
- $w'_j = \frac{M_j}{N}$
- $w_{ij} = \frac{N_{ij} * N}{N_i * M_j}$

$\rightarrow I(X;Y) := \frac{1}{N} \sum_{e_{ij} \in E} N_{ij} log(w_{ij})$

Theoretical Challenges

- **Discrepancy**: [Morteza 2018] states that the MI estimator has $max(g(x), U)$ instead of *min*, which is against the assumptions.
- **Handling varying dimensions of input**:
  - The paper assumes both the inputs, X and Y, have same size, which is not the case with the hidden layers.
  - Probabilities required to be redefined.

Term	**EDGE version**	**Redefined version**
$\hat{g}(x)$	$min(g(x), U)$	$max(g(x), U)$
N	# elements in X (or Y)	# (X, Y) pairs

Table: The corrections made in EDGE.

Implementation Challenges

- **Choosing the appropriate Hash function,**
  - has uniform density on the output.
  - works for d-dimensional data.
- **Size of the Hash Table:** By experimentation.
- **Parallelisation:** The algorithm proposed needed parallelisation at various levels, like:
  - Iterating over each pair to hash.
  - Counting collisions.
  - Computing weights for each edge.

  - Solved using tensor algebra over GPU, and techniques of vectorisation, function inlining, sampling, etc.

Experimentation and Results

Comparison of EDGE implementation results with the expected output:

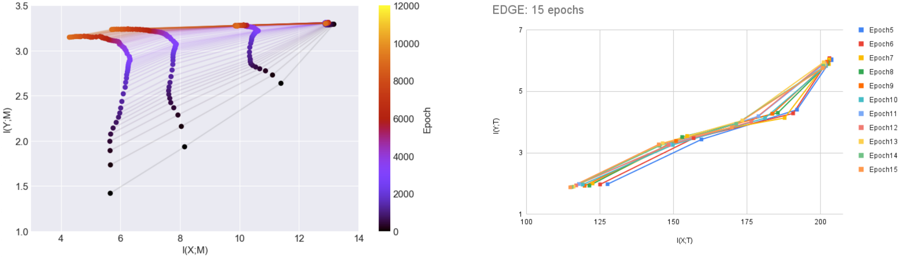


Figure: The expected plots from [Noshad and Hero, 2018] Vs. our EDGE implementation results when evaluating a sample Sequential Model over MNIST Dataset with ReLU Activation.

Comparison of EDGE implementation results with the expected output:

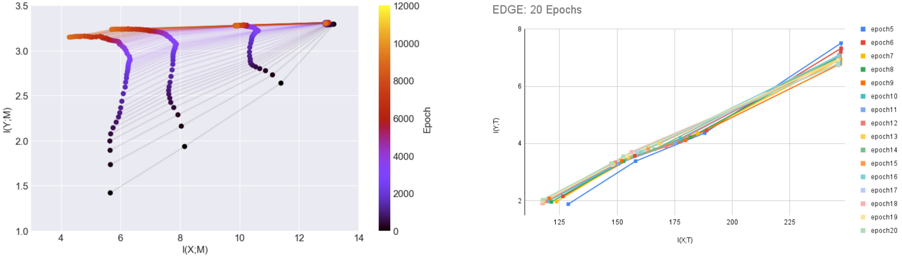


Figure: The expected plots from [Noshad and Hero, 2018] Vs. our EDGE implementation results when evaluating a sample Sequential Model over MNIST Dataset with ReLU Activation.

.

Comparison of EDGE implementation results with the expected output:

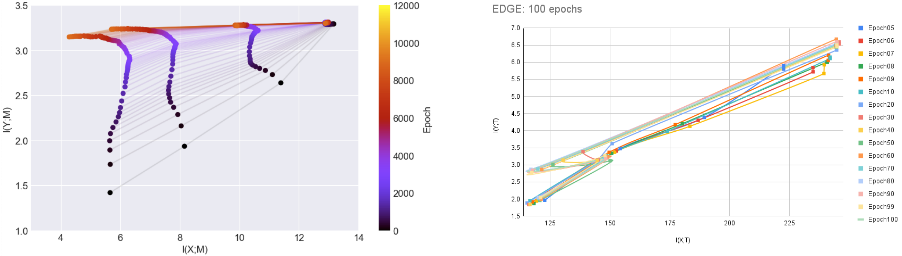
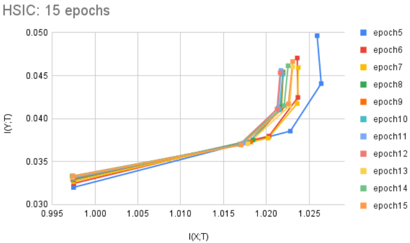


Figure: The expected plots from [Noshad and Hero, 2018] Vs. our EDGE implementation results when evaluating a sample Sequential Model over MNIST Dataset with ReLU Activation.

.

Comparison of EDGE implementation results with that of HSIC estimator:



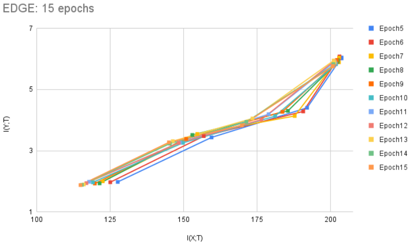


Figure: The results of HSIC Vs. EDGE implementation when evaluating a sample Sequential Model over MNIST Dataset with ReLU Activation, both for 15 epochs.

.

Some inferences:

- **Empirical Error Minimisation:**
  The curves seem to saturate after a certain no. of epochs.

- I_y **increases with no. of epochs**:
  Better training accuracy.

- **Decrease in I_x:**
  Depicts the compression of representation.

- **Absolute value of MI:**
  Overall trends hold higher significance for us since the value is governed by several outer factors, including the estimator used.

Important Conclusions

- *The right hash functions ..?*
  Ans: The estimates are closer when using XOR in H_2. H_1 can be chosen as Multiplicative Hashing or Floor function.
- *How to decide the value of F, or the size of the hash table?*
  Ans: Experimentally, F is a function of input size.
- *How does MI vary across hidden layers? Factors on which it depends?*
  Ans: EDGE shows the representation compression as we go deeper into the hidden layers, and error minimisation as the epochs increase.

  Governing Factors: the estimator used, no. of hidden layers, input size, filters, etc.

Future Work

Can be in different directions:

- Testing EDGE on actual DL Architectures:
  - DenseNet/ResNet
  - Information Theory Based Architecture
- For various datasets like:
  CIFAR10, Cell Histopathology images, etc.
- Testing on higher no. of epochs.

Can start putting down the work as a research paper, once there's sufficient progress in one of these directions by August *:)*

Literature Review

- Understanding of Entropy and Mutual Information [Book, Cover & Thomas, 2006].
- Two phases of the Stochastic Gradient Descent [Shwartz-Ziv and Tishby, 2017].
- Information Plane: Drift and Diffusion Phases [Shwartz-Ziv and Tishby, 2017].
- EDGE implementation [Noshad and Hero, 2018] & [Noshad, GitHub].
- Hash techniques [Datar, 2004].
- Perfect Hashing and Minimal Perfect Hashing [Perfect Hashing, Czech, Havas, 1997].
- GPU Parallelisation: PyTorch Forums & other online resources.

References I

M. Noshad, Y. Zeng, and A. O. Hero III (2018)
"Scalable Mutual Information Estimation using Dependence Graphs"
CoRR, vol. abs/1801.09125, 2018. arXiv: 1801.09125. [Online]. Available: http://arxiv.org/abs/1801.09125.

R. Shwartz-Ziv, and N. Tishby (2017)
"Opening the Black Box of Deep Neural Networks via Information"
CoRR, vol. abs/1703.00810, 2017. arXiv: 1703.00810. [Online]. Available: http://arxiv.org/abs/1703.00810.

Thomas M. Cover, and Joy A. Thomas(2006)
"Elements of Information Theory"
Second Edition, Chapter 2.

M. Noshad, and A. O. Hero III (2018)
"Scalable Hash-Based Estimation of Divergence Measures"
CoRR, vol. abs/1801.00398, 2017. arXiv: 1801.00398. [Online]. Available: http://arxiv.org/abs/1801.00398.

References II

M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni
"Locality-Sensitive Hashing Scheme Based on p-Stable Distributions (2004)"
[Online]. Available: https://doi.org/10.1145/997817.997857.

Z. J. Czech, G. Havas, and B. S. Majewski (1997)
"Fundamental Study Perfect hashing"
Theoretical Computer Science 182 (1997), Chapter 1.

M. Noshad, EDGE, 2018
GitHub Repository
[Online]. Available: https://github.com/mrtnoshad/EDGE/.

P. Mehta, and D. J. Schwab (2014)
"An exact mapping between the Variational Renormalization Group and Deep Learning"
CoRR, vol. abs/1410.3831, 2014. arXiv: 1410.3831. [Online]. Available: http://arxiv.org/abs/1410.3831

References III

Shao-Lun Huang, X. Xu, L. Zheng, and G. W. Wornell (2019)
"An Information Theoretic Interpretation to Deep Neural Networks"
CoRR, vol. abs/1905.06600, 2019. arXiv: 1905.06600. [Online]. Available: http://arxiv.org/abs/1905.06600

N. Tishby, and N. Zaslavsky
"Deep Learning and the Information Bottleneck Principle"
CoRR, vol. abs/1503.02406, 2015. arXiv: 1503.02406. [Online]. Available: http://arxiv.org/abs/1503.02406.