Deep Neural Networks (DNN) with Energy-Based Learning

Anthony, Faustine

Data Scientist, CeADAR

sambaiga anthony.faustiner@ucd.ie

sambaiga.github.io/sambaiga/

CeADAR
Ireland's Centre for Applied AI

TECHNOLOGY CENTRE
ENTERPRISE IRELAND
IDA IRELAND SUPPORTED

Outline

Introduction

Energy Based Model (EBM)

EBMs Learning

DNN-EBM applications

Conclusion

Deep Learning Success

Automatic Colorization

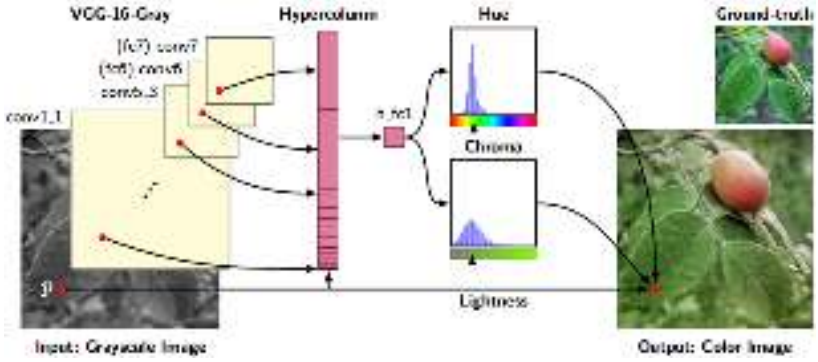


Figure 1: Automatic colorization

Game



Object Classification and Detection

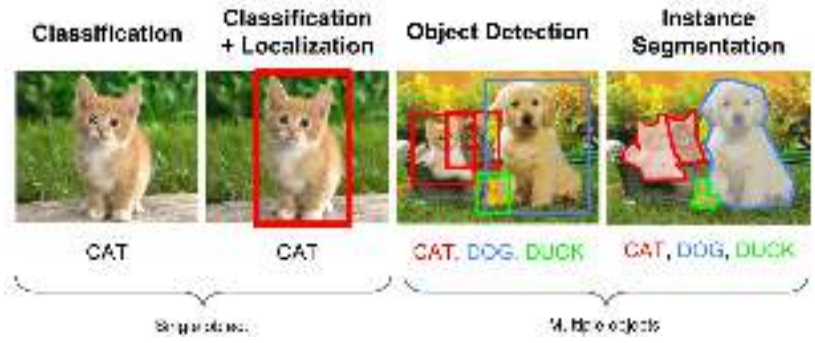
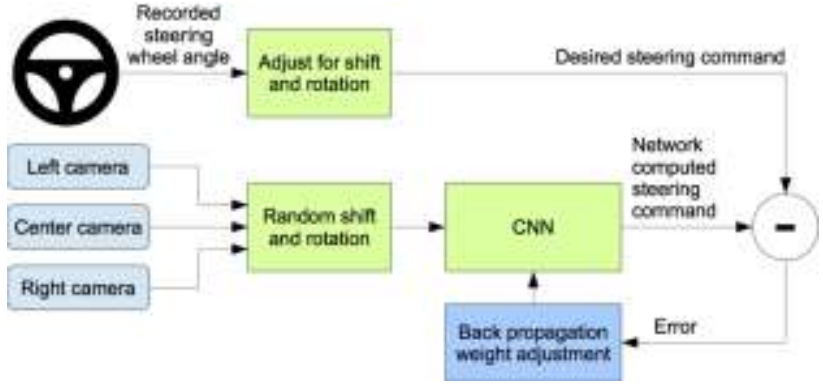


Figure 2: Object recognition

Self driving car



Motivation

Deep Learning use finite number of computational steps (stacked layers) to produce a single prediction.

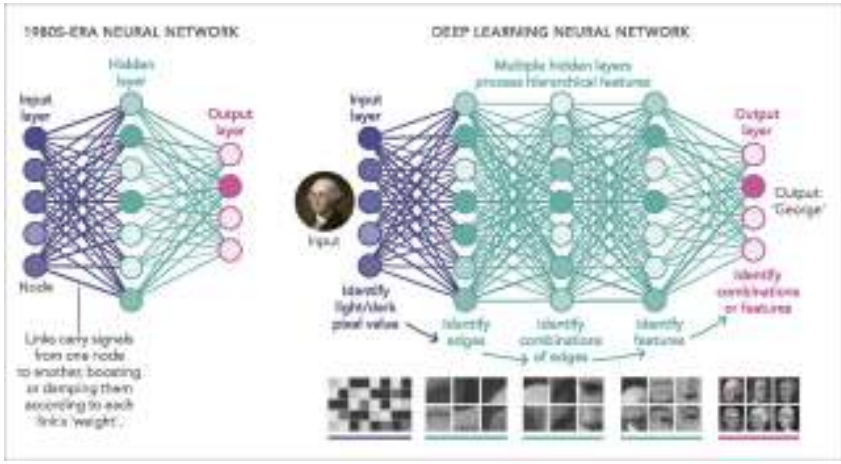


Figure 3: Deep learning: *credit:M. Mitchell Waldrop*

Issues:

- When the computed output require a complex computations (complex inference).
- When we need multiple possible outputs eg. predicting video frames.
- When labeled data is not enough.
- How to deal with uncertainty in the prediction?.

Outline

Introduction

Energy Based Model (EBM)

EBMs Learning

DNN-EBM applications

Conclusion

Energy Based Model (EBM)

EBM encode dependencies between variables (x, y) by associating a scalar parametric energy function $E_\theta(.)$ to each of the variables.

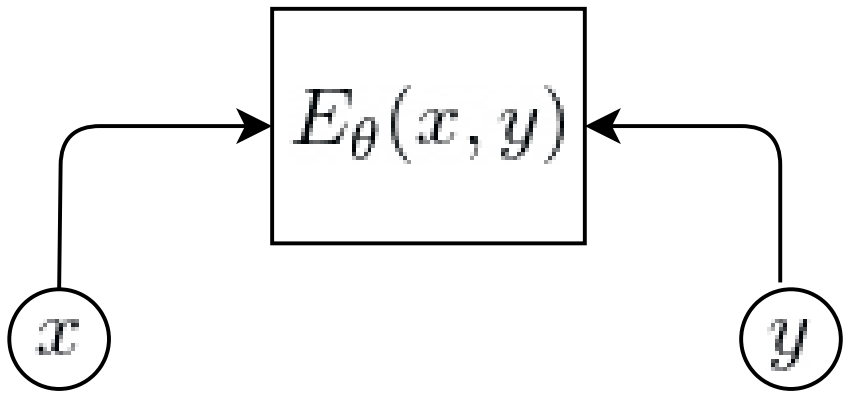
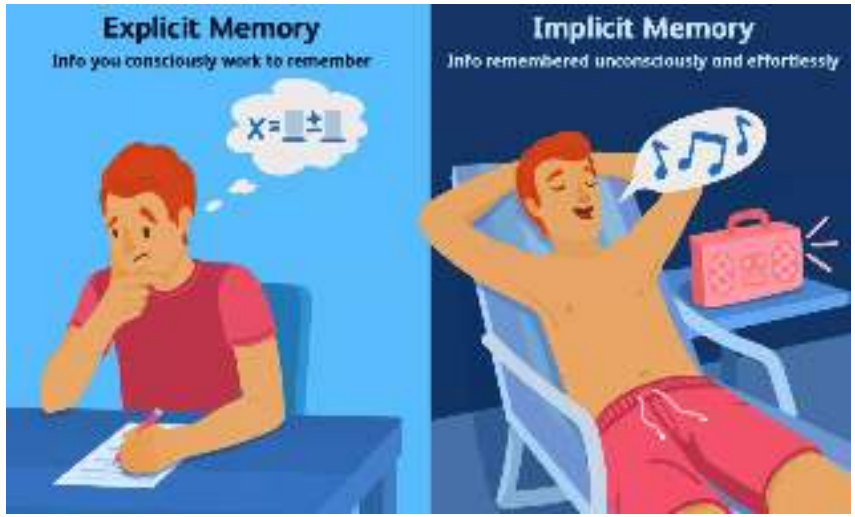


Figure 4: Energy function

- Learn to find if y is compatible to x eg. Is y an accurate high-resolution image of x ?
- $E_\theta(x, y)$ captures some statistical property of the input data.
- $E_\theta(x, y)$ takes low values when y is compatible with x and higher values when y is less compatible with x.

EBM vs Neural Networks

- A feed-forward model is an explicit function that computes y from x.
- An EBM is an implicit function that captures the dependency between x and y



EBM Inference

The energy $E_\theta(.)$ is used for inference, not for learning.

Conditional Energy: $E_\theta(x, y)$ vs Unconditional Energy: $E_\theta(x)$
Inference: find values of y that make $E_\theta(x, y)$ small.

$$\hat{y} = \arg\min_y E_\theta(x, y) \tag{1}$$

The EBM model could be used for:

- Prediction, classification, and decision-making *which value of y is most compatible with this x*
- Ranking: *is y_1 or y_2 more compatible with this x*
- Conditional density estimation: *what is the conditional probability distribution over $\mathcal{Y}$ given x*

EBM as Probabilistic Model

$E_\theta(x)$ can be turned into a normalized joint probability distribution $p_\theta(x)$ through the Gibbs distribution:

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)} \tag{2}$$

where $Z(\theta) = \int_{x \in \mathbf{x}} \exp(-E_\theta(x) dx$ is is the normalizing constant. Pros:

- Extreme flexibility: can use pretty much any function $-E_\theta$ you want.

Cons:

- Sampling from $p_\theta(x)$ is hard.
- Evaluating and optimizing likelihood $p_\theta(x)$ is hard (learning is hard)
- No feature learning (but can add latent variables)

EBM with latent variable

Latent EBM: The output y depends on x as well as an extra variable z (the latent variable)

$$E_\theta = E_\theta(x, y, z) \tag{3}$$

Given z the $E_\theta(x, y, z))$ can be used for both generation of x and identification of a y implicitly.

$$\hat{x} = \arg\min_x E_\theta(x, y, z) \tag{4}$$

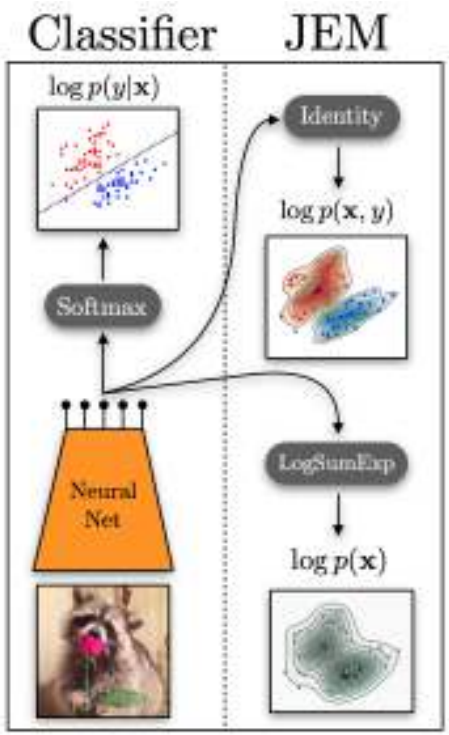
$$\hat{y} = \arg\min_y E_\theta(x, y, z) \tag{5}$$

Allows a machine to produce multiple outputs, not just one.

Neural Network as Energy Function

$E_\theta(x)$ can be parameterized by neural networks for a wide variety of tasks.

- Defining $E_\theta(x, y)$ as DNN allow to exploit the predictive power of DNN and the benefits of EBMs.



Consider a DNN $f_\theta(\mathbf{x}[y]) \Longrightarrow$ map (x, y) to a scalar value.

Re-interpret $f_\theta(\mathbf{x}[y])$ as the negative energy $E_\theta = -f_\theta(\mathbf{x}[y])$.

$$p_\theta(\mathbf{x}, y) = \frac{\exp(f_\theta(\mathbf{x})[y]))}{Z(\theta)} \tag{6}$$

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \sum_y \frac{\exp(f_\theta(\mathbf{x}))}{Z(\theta)} \tag{7}$$

$$p_\theta(y|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, y)}{p_\theta(\mathbf{x})} \tag{8}$$

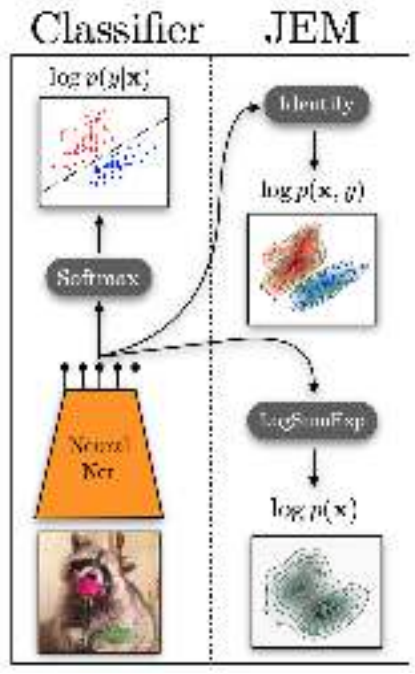
Neural Network as Energy Function

The energy function of a data point x can thus be defined as

$$E_\theta(x) = -\text{LogSum}_y f_\theta(\mathbf{x}) = -\log \sum_y \exp(f_\theta(\mathbf{x})) \tag{9}$$

Optimize:

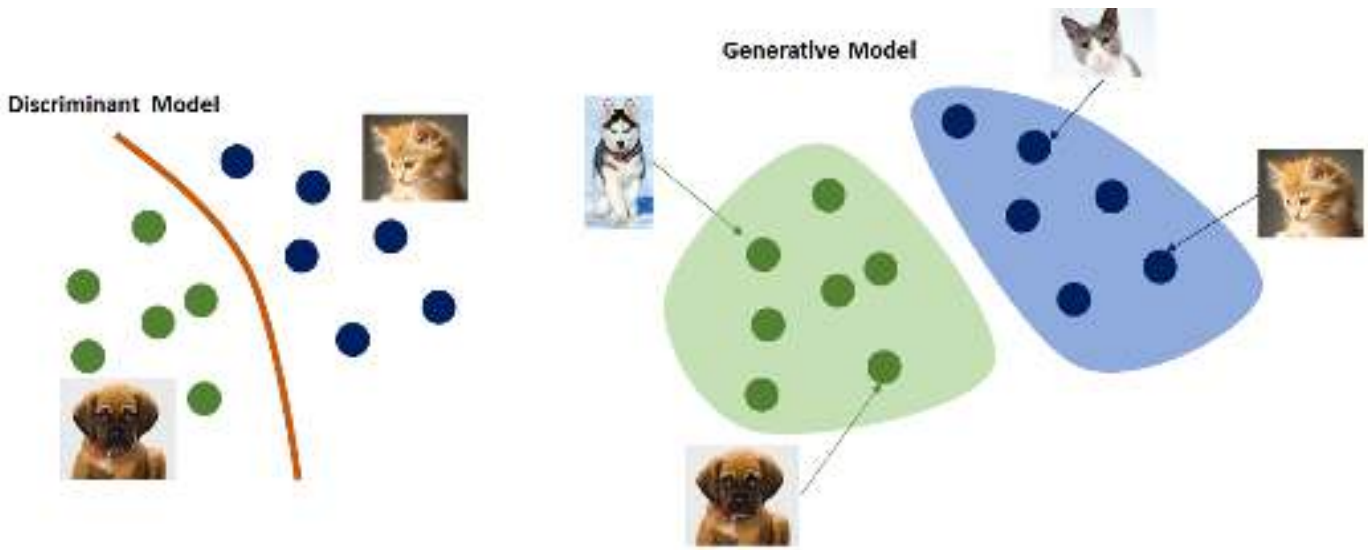
$$\arg\min_\theta \mathbb{E}_{p_D}\left[-\log p_\theta(\mathbf{x}, y)\right]$$
$$= \arg\min_\theta -\mathbb{E}_{p_D}\left[\log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x}, \theta)\right]$$



EBM advantages

Provide unified framework for probabilistic and non-probabilistic learning approaches.

- Proper normalization is not required, $\Rightarrow$ EBMs don't have the issues arising from estimating the normalization constant in probabilistic models.
- Allows for much more flexibility in the design of learning machines.



Outline

Introduction

Energy Based Model (EBM)

EBMs Learning

DNN-EBM applications

Conclusion

EBM: learning

Learning: finding an energy function which gives lower energies to observed configurations than unobserved ones

- Assigns low E_θ values to inputs in the data distribution and high E_θ values to other inputs.

$$p_\theta(x) = \frac{\exp(-E_\theta(x)}{Z(\theta)} \tag{10}$$

- The log-likelihood of $E_\theta(x)$

$$\log p_\theta(x) = -E_\theta - \log \mathbb{E}_{p(x)} \exp(-E_\theta(x)) \tag{11}$$

- For most choices of E_θ, it is hard to estimate $Z(\theta) \Rightarrow$ intractable
- If x is 16×16 RGB image
  - Computing $Z(\theta) \longrightarrow$ summation over $(256 \times 256 \times 256)^{16 \times 16}$ terms.

EBM: MLE

- In MLE, we seek to maximize the log-likelihood function $\Rightarrow$ equivalent to minimizing the Kullback-Leibler divergence $KL(p_D||q_\theta)$
- The derivative of the log-likelihood for a single example x with respect to θ

$$\frac{\partial \log p_\theta(x)}{\partial \theta} = \mathbb{E}_{p_\theta(x')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right] - \frac{\partial E_\theta(x)}{\partial \theta} \tag{12}$$

$$-\frac{\partial KL(p_D||q_\theta)}{\partial \theta} = \frac{\partial E_\theta(x)}{\partial \theta} - \mathbb{E}_{p_\theta(x')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right] \tag{13}$$

- $\mathbb{E}_{p_\theta(x')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right]$ is intractable.
- Can be approximated through samples (Langevin Dynamics or MCMC).

EBM MLE Sampling: SGLD

- Stochastic Gradient Langevin Dynamics (SGLD) [1]–[3] use of the gradient of $E_\theta(.)$ to undergo sampling such as

$$\mathbf{x}'_k = \mathbf{x}'_{k-1} - \frac{\alpha}{2}\frac{\partial E_\theta(\mathbf{x}'_{k-1})}{\partial \theta} + \epsilon_k \tag{14}$$

  where $\mathbf{x}_0 \sim p_0(\mathbf{x})$ and $\epsilon_k \sim \mathcal{N}(0, \alpha)$
- SGDLD sampling define a distribution q_θ such that $\mathbf{x}'_k \sim q_\theta$.
- As as $K \to \infty$ and $\alpha \to 0$ then $q_\theta \sim p_\theta$.
- Samples are generated from the distribution defined by $E_\theta(.)$

EBM: Noise contrastive estimation

Given

$$p_\theta(x) = \frac{\exp -E_\theta(x)}{Z(\theta)} \tag{15}$$

Can we learn $Z(\theta)$ instead of computing it ? $\implies c = \log Z(\theta)$ [4], [5].

- $p_\theta(x) = \exp\left[-E_\theta(x) - c\right]$ c is now treated as a free parameter.
- Introduce a noise distribution $q(x)$ turn EBM estimation into classification problem

$$\mathbb{J}(\theta) = \mathbb{E}_{p_{\mathbb{D}}}\left[\log \frac{p_\theta(x)}{p_\theta(x) + q(x)}\right] + \mathbb{E}_q\left[\log \frac{q(x)}{p_\theta(x) + q(x)}\right] \tag{16}$$

- Strictly requirements on $q(x)$
  1. Analytically tractable expression density.
  2. Easy to draw samples from.
  3. Close to data distribution $\Rightarrow$ Flow Contrastive Estimation [5].

Outline

Introduction

Energy Based Model (EBM)

EBMs Learning

DNN-EBM applications

Conclusion

DNN-EBM: Generative modeling

EBM is used to model the underlying data distribution [3], [5][1]

- EBM does not require an explicit neural network to generate samples (unlike GANs, VAEs, and Flow-based models).

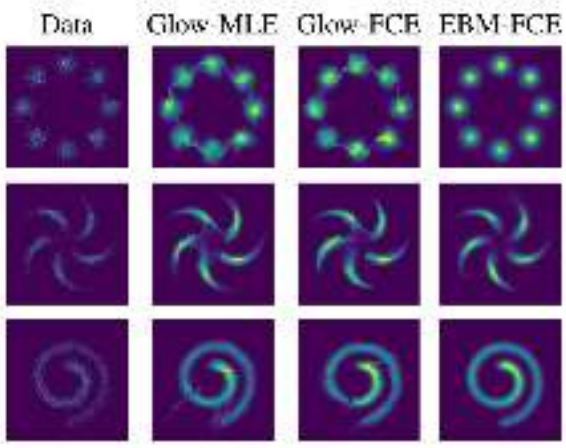


Figure 1: Comparison of trained EBM and Glow models on 2-dimensional data distributions.



(a) GLOW Model (b) EBM

Figure 5: Comparison of image generation techniques on unconditional CIFAR-10 dataset [2]

EBMs are effective generative models for multi-dimensional inputs like images [3], [5].

[1] http://www.stat.ucla.edu/ ruiqigao/fce/main.html

[2] https://github.com/openai/ebm_code_release

DNN-EBM: Semi-supervised learning

EBMs can be generalized to perform semi-supervised learning.

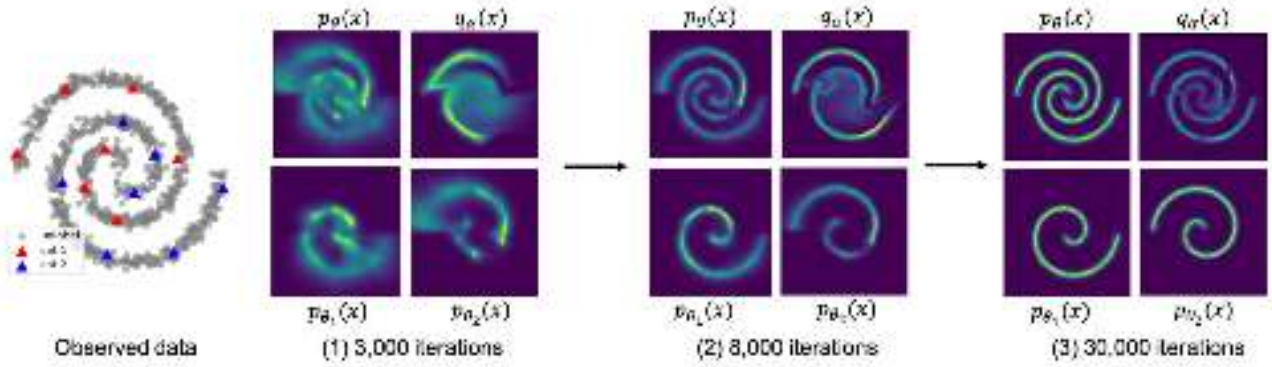
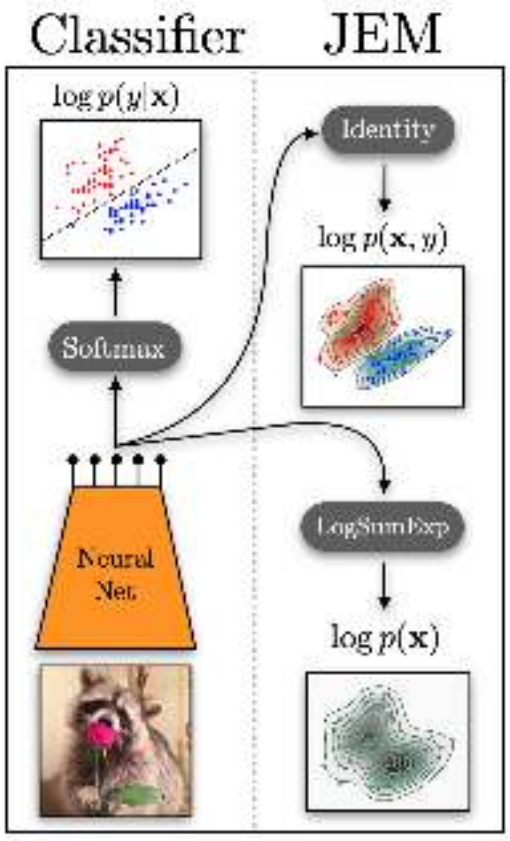


Figure 5: Illustration of FCE for semi-supervised learning on a 2D example, where the data distribution is two spirals belonging to two categories. Within each panel, the top left is the learned unconditional EBM. The top right is the learned Glow model. The bottom are two class-conditional EBMs. For observed data, seven labeled points are provided for each category.

EBM tends to learn a smoothly connected cluster, which is often what we desire in semi-supervised learning [5].

DNN-EBM: Classification



- Joint Energy based Model applying SGLD[3] [2]
- Hybrid Discriminative Generative Energy-based Model(HDGE)[4]: optimize Supervised learning and contrastive learning: [6] .

Dataset	Method				
	Supervised Learning	Supervised Contrastive	JEM	HDGE (ours)	HDGE ($\log q_\theta(x\|y)$ only)
CIFAR10	95.8	96.3	94.4	**96.7**	96.4
CIFAR100	79.9	80.5	78.1	**80.9**	80.6

Table 1: **Comparison on three standard image classification datasets**: All models use the same batch size of 256 and step-wise learning rate decay, the number of training epochs is 200. The baselines Supervised Contrastive [21], JEM [8], and our method HDGE are based on WideResNet-28-10 [50].

EBM results into improved uncertainty quantification, model-calibrated out-of-distribution detection (OOD), and robustness to adversarial examples.

[3] https://wgrathwohl.github.io/JEM/
[4] https://github.com/lhao499/HDGE

DNN-EBM: Model calibration

For calibrated model the predictive confidence $\arg\max_y p(y|x)$, aligns with its misclassification rate.

- when predicts label y with 0.9 confidence it should have a 90% chance of being correct.
- important feature for a model to have when deployed in real-world scenarios.
- Usually evaluated in terms of the Expected Calibration Error (ECE)

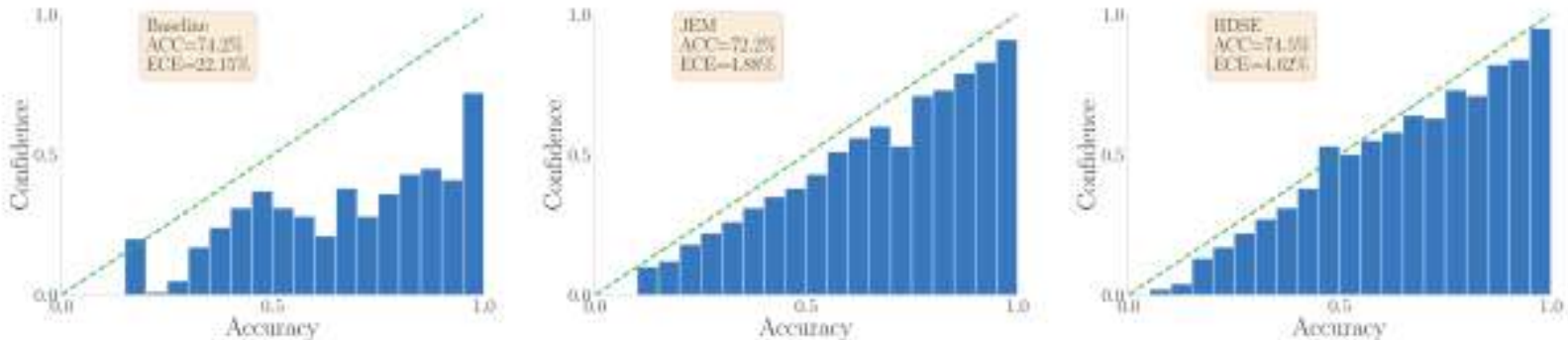
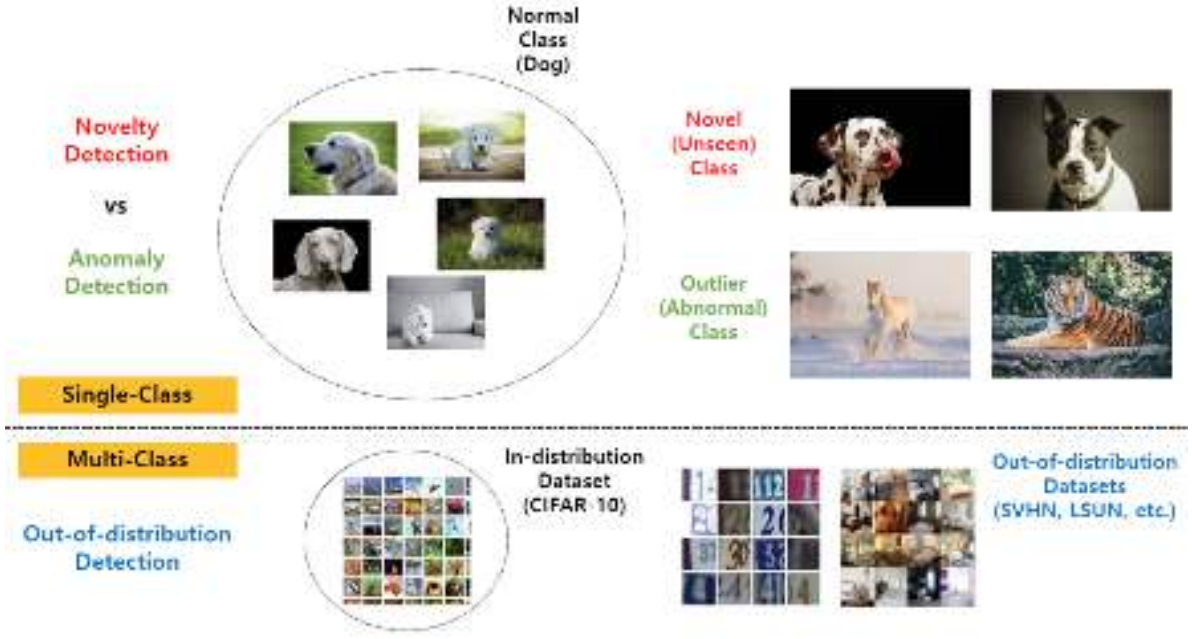


Figure 2: **CIFAR-100 calibration results**. The model is WideResNet-28-10 (without BN). Expected calibration error (ECE) [10] on CIFAR-100 dataset under various training losses.

EBMs significantly improves the calibration of classifier

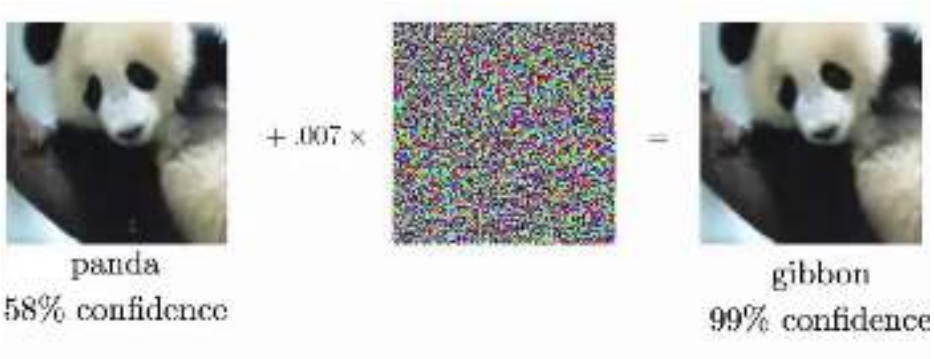
DNN-EBM: OOD



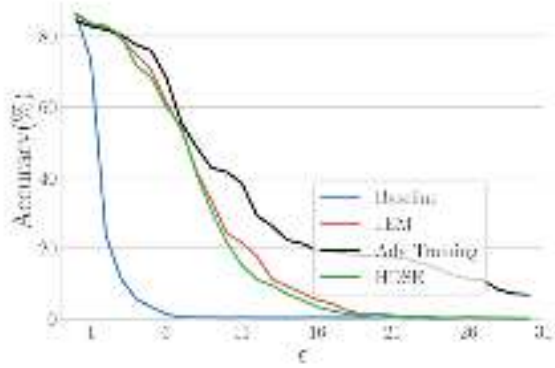
Model	PixelCNN++	Glow	EBM (ours)
SVHN	0.32	0.24	**0.63**
Textures	0.33	0.27	**0.48**
Constant Uniform	0.0	0.0	**0.30**
Uniform	1.0	1.0	**1.0**
CIFAR10 Interpolation	**0.71**	0.59	0.70
Average	0.47	0.42	**0.62**

Figure 10: AUROC scores of out of distribution classification on different datasets. Only our model gets better than chance classification.

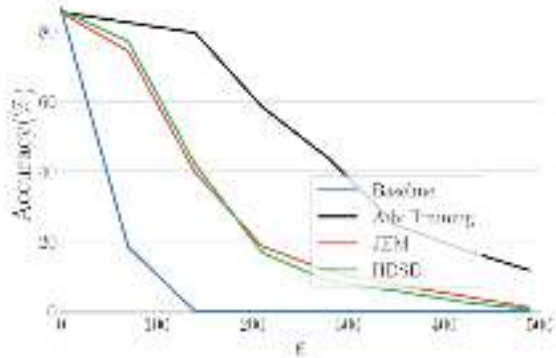
DNN-EBM: Adversial Attack



- DNN are sensitive to perturbation-based adversarial examples.
- DNN-EBMs exhibit adversarial robustness without explicit adversarial training.



(a) L_∞ Robustness



(b) L_2 Robustness

Figure 3: **Adversarial robustness** results with PGD attacks. HDGE adds considerable robustness to standard supervised training and achieves comparable robustness with JEM.

DNN-EBM: compositional learning

Human intelligence is capable to compose complex concepts out of simpler ideas ⇒ rapid learning and adaptation of knowledge.

- DNN not good at compositional learning.
- EBM exhibit compositional learning by directly combining probability distributions [3], [7], [8][5].

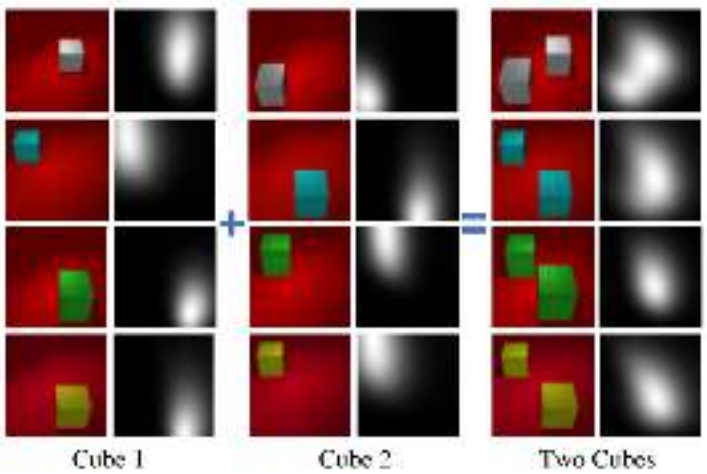


Figure 12: Concept inference of multiple objects with EBM trained on single cubes and tested on two cubes. The color image is the input and in grayscale is the output energy map over all positions. The energy map of two cubes correctly shows the bimodality which is close to the summation of the front two energy maps.

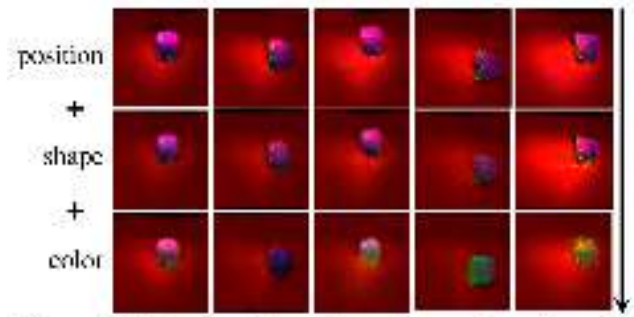


Figure 7: Continual learning of concepts. A position EBM is first trained on one shape (cube) of one color (purple) at different positions. A shape EBM is then trained on different shapes of one fixed color (purple). Finally, a color EBM is trained on shapes of many colors. EBMs can continually learn to generate many shapes (cube, sphere) with different colors at different positions.

[5] https://energy-based-model.github.io/compositional-generation-inference/

Outline

Introduction

Energy Based Model (EBM)

EBMs Learning

DNN-EBM applications

Conclusion

Conclusion

- Energy-based models very flexible class of models.
- Parameterized energy function with DNN provide a unified framework for modeling high-dimensional probability distributions.
- Explore, extend, and understand their applicability in industrial applications.

References I


Max Welling and Yee Whye Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: Proceedings of the 28th International Conference on International Conference on Machine Learning. ICML'11. Bellevue, Washington, USA: Omnipress, 2011, pp. 681–688. ISBN: 9781450306195.


Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, et al. YOUR CLASSIFIER IS SECRETLY AN ENERGY BASED MODEL AND YOU SHOULD TREAT IT LIKE ONE. Tech. rep. arXiv: 1912.03263v2.


Yilun Du and Igor Mordatch. "Implicit Generation and Generalization in Energy-Based Models". In: NeurIPS (2019). arXiv: 1903.08689. URL: http://arxiv.org/abs/1903.08689.

References II


Michael U. Gutmann and Aapo Hyvärinen. "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics". In: Journal of Machine Learning Research 13.11 (2012), pp. 307–361. URL: http://jmlr.org/papers/v13/gutmann12a.html.


Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, et al. "Flow Contrastive Estimation of Energy-Based Models". In: (2020), pp. 7515–7525. DOI: 10.1109/cvpr42600.2020.00754. arXiv: 1912.00589.


Hao Liu and Pieter Abbeel. "Hybrid Discriminative-Generative Training via Contrastive Learning". In: (2020). arXiv: 2007.09070. URL: http://arxiv.org/abs/2007.09070.

References III


Yilun Du, Shuang Li, and Igor Mordatch. "Compositional Visual Generation and Inference with Energy Based Models". In: (2020). arXiv: 2004.06030. URL: http://arxiv.org/abs/2004.06030.


Igor Mordatch. "Concept learning with energy-based models". In: 6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings (2018). arXiv: 1811.02486.