Your Classifier is Secretly an Energy Based Model and You Should Treat It Like One

Maria Kovaleva

MIPT, 2022

November 6, 2022

Basic Concept of EBM

Energy-Based Models (EBMs) capture dependencies by associating a scalar energy to each configuration of the variables
Learning: finding an energy function that associates low energies to correct values, and higher energies to incorrect values
Inference: finding that minimize the energy
Pros: no requirement for proper normalization, in comparison with probabilistic models

Probability density

$p(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^D$: $p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}$, where $E_\theta(\mathbf{x})$ is energy function and $Z(\theta) = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x}))d\mathbf{x}$ is normalizing constant or partition function

Learning of EBM

Estimate $Z(\theta)$ can be challenging for most energy functions. Derivative of the log-likelihood for a single example x with respect to θ:
$\frac{\partial \log p_\theta(\mathbf{x})}{\partial \theta} = \mathbb{E}_{p_\theta(\mathbf{x}')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right] - \frac{\partial E_\theta(x)}{\partial \theta}$

Learning

Approximate the expectation in derivative of the log-likelihood using a sampler based on Stochastic Gradient Langevin Dynamics (SGLD):
$\mathbf{x}_0 \sim p_0(\mathbf{x})$, $\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\alpha}{2}\frac{\partial E_\theta(\mathbf{x}_i)}{\partial \mathbf{x}_i} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \alpha)$,
where $p_0(\mathbf{x})$ is typically a Uniform distribution, α - step-size (should be decayed following a polynomial schedule).
Practically α and ϵ is often chosen separately leading to a biased sampler which allows for faster training

In classifiers

Classification: D parameters, K classes, parametric function, $f_\theta : \mathbb{R}^D \to \mathbb{R}^K$ return logits

Categorical distribution parameterized by f_θ via Softmax transfer function: $p_\theta(y|\mathbf{x}) = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])}$

Reinterpretation of the logits

Let's define an energy based model with $E_\theta(\mathbf{x}, y) = -f_\theta(\mathbf{x})[y]$ and unknown normalizing constant Z_θ :

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

Then

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)}$$

Conclusion

The LogSumExp$(\cdot)$ of the logits of any classifier can be re-used to define the energy function as
$E_\theta(\mathbf{x}) = LogSumExp(f_\theta(\mathbf{x})[y]) = -\log \sum_y \exp f_\theta(\mathbf{x})[y]$
A generative model hidden within every standard discriminative model!

Optimization

$\log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x})$
Where:

1. $p(y|\mathbf{x})$ optimized using standard cross-entropy
2. $\log p(\mathbf{x})$ optimized with SGLD where gradients are taken with respect to $LogSumExp(f_\theta(\mathbf{x})[y])$ as it described for EBMs

Remarks

1. We don't know Z_θ, because of it we use SLGD sampler
2. The estimator of $\log p(\mathbf{x})$ will be biased when using a MCMC sampler with a finite number of steps

Experiment

CIFAR10, CIFAR100, SVHN datasets were used
All architectures used are based on Wide Residual Networks where we have removed batch-normalization to ensure that our models' outputs are deterministic functions of the input
Comparison with Resudal Flow[1], Glow[2], IGEBM[3], SNGAN[4], NCSN[5] and other models

[1] Ricky TQ Chen, Jens Behrmann, David Duvenaud, and Jorn-Henrik Jacobsen. Residual flows for invertible generative modeling

[2] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions

[3] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks

[5] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution

Results

1. Performance rivaling the state of the art in both discriminative and generative modeling achieved
2. Increase in inception scores (IS) and Frechet Inception Distance (FID) for generative modeling
3. Increase in accuracy for discriminative problem
4. Decrease in Expected Calibration Error (ECE) for calibration [6]
5. Better out-of-distribution (OOD) detection [7] results for $s_\theta(x) = \max_y p_\theta(y|\mathbf{x})$ and for $s_\theta = -\left\| \frac{\partial \log p_\theta(\mathbf{x})}{\partial \mathbf{x}} \right\|_2$
6. Better robustness

[6] A classifier is considered calibrated if its predictive confidence, $\max_y p(y|\mathbf{x})$, aligns with its misclassification rate. Thus, when a calibrated classifier predicts label y with confidence 0.9 it should have a 90% chance of being correct.

[7] Out-of-distribution (OOD) detection is a binary classification problem. $s_\theta \in \mathbb{R}$ We desire that the scores for in-distribution examples are higher than that out-of-distribution examples

Literature

[1] Will Grathwohl et al. "Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One". In: *CoRR* abs/1912.03263 (2019). arXiv: 1912.03263. URL: http://arxiv.org/abs/1912.03263.

[2] Yann Lecun et al. "A Tutorial on Energy-Based Learning". In: Jan. 2006.

[3] Erik Nijkamp et al. *Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model*. 2019. DOI: 10.48550/ARXIV.1904.09770. URL: https://arxiv.org/abs/1904.09770.