On Learning Energy-based Model in Supervised Learning Tasks

Hao Xin

September 9, 2019

1 Introduction

Deep energy-based models (EBMs by LeCun) have been widely applied in many model-driven machine learning tasks, such as image generation, language modeling and even robotics. A major advantage of EBM is its freedom of choice on energy function which can formulate a probability density over high-dimensional data that can capture variety dependencies among variables. Like a double-edged sword, the flexibility over energy function often require wisely designed parameterization structure which can often lead to gradient without direct computational form. Approximation is needed when performing maximum likelihood estimation for unknown model parameters. Thus, energy-based models are more commonly used in unsupervised tasks where understanding from distribution level is crucial. On the other hand, supervised learning tasks, are often dealt by loss-based approach that training and prediction are done in a more straightforward fashion compared to model-based learning.

In essence, model-based and loss-based learning are equivalent. Take regression as example, given explanatory feature x and target y, loss-based learning tries to minimize some pre-defined loss function $\mathcal{L}(\hat{y}, y)$ with some learned mapping function $g(\cdot)$. So the loss based approach essentially suggest the following relationship:

$$\|y - g(x)\| = \mathcal{L}(g(x), y)$$

This is equivalent to an maximum likelihood estimation with the following conditional distribution

$$P(y|x) = \frac{1}{Z(x)} \exp\Big(- \mathcal{L}(g(x), y)\Big)$$

where $\mathcal{L}(g(x), y)$ is also recognized as an energy function. That is, it tends to have high energy for incredible predictions while assigning low energy to more trustworthy predictions.

Monocular depth estimation is an important problem in scene understanding, depth information from single RGB image can be useful for 3D reconstruction. One fundamental question of monocular depth learning is how much depth information contained in RGB images. If it is rich enough so that one would be possible to propose a well designed model to learn depth information from raw images in an unsupervised manner. Otherwise, supervision is required for finding some mapping function from RGB channels to depth map.

2 Energy based model

2.1 Model for Depth Map

So far, the state-of-art methods of depth map regression are mainly loss-based, which are usually harder to acquire distribution over images. Due to the grid structure of images, typical probabilistic models for image modeling are usually based on random field models like:

- FRAME (Markov random field) model.
- CRF (Conditional random filed) model.

A non-trivial problem for random field model on images is to define neighborhood relation on image pixels. CRF model defines graph neighbors over nearby pixels or formed super-pixels. FRAME model, bypass clique potentials used in traditional random field model by using expressive or even trainable filters to capture pixel relations.

After all, these distributions are all belong to the general family of energy based model having the following form:

$$P(Y) = \frac{1}{Z(\theta)} \exp\left(-f_\theta(Y)\right)$$

where $f_\theta(\cdot)$ is some (non-negative) energy function and θ is the unknown parameters. This general parametric model can be trained by maximum likelihood estimation on θ which is equivalent to minimize the KL-divergence between the true data distribution p_d and parametric distribution p_θ. The biggest difficulty to train such a model is the unknown normalizing constant $\mathbb{Z}(\theta)$, which makes the gradient for updating MLE estimator uncomputable. To resolve this issue, the most common approach is to generate MCMC samples and compute an approximate gradient. We will use depth prediction as an example to illustrate how the MLE procedure is performed.

Let $\mathbf{D}$ be the depth map defined on spatial domain $\mathcal{D}$ and $\mathbf{X}$ be the input RGB image. A conditional random field model defines a probability measure $P_\theta(\mathbf{D}|\mathbf{X}) : \mathbb{R}_+^{|\mathcal{D}|} \to [0,1]^{|\mathcal{D}|}$, where θ are unknown underlying model parameters. For a given model, one would want to find the maximum likelihood estimate that

$$\hat{\mathbf{D}} = \arg\max_{\mathbf{D}} P_\theta(\mathbf{D}|\mathbf{X})$$

We consider the non-stationary inhomogeneous FRAME model on texture which can be generalized to a model on depth map in the following form:

$$P(\mathbf{D};\lambda) = \frac{1}{Z(\lambda)} \exp\left[-\sum_{k=1}^{K}\sum_{y\in\mathcal{D}} \lambda_{k,y}\big([F_k * \mathbf{D}](y), \mathbf{X}\big)\right] q(\mathbf{D}) \tag{1}$$

where function $\lambda_{k,y}(\cdot, \mathbf{X})$ depend on spatial position y and $\{F_k : k = 1, \ldots, K\}$ is a bank of filters. $q(\mathbf{D})$ works as a reference distribution here, in Bayesian inference $q(\mathbf{D})$ is the prior distribution on $\mathbf{D}$. For simplifying learning, we use

$$\lambda_{k,y}(r, \mathbf{X}) = \omega_{k,y}(\mathbf{X}) \cdot r$$

which will have all different pixels contribute, and some may contribute in a different direction. With a with a slight abuse of notation, we will denote both the above two function as the same, so we get the conditional FRAME model for depth:

$$P_\theta(\mathbf{D}|\mathbf{X}) = \frac{1}{Z(W,F)} \exp\left[-\sum_{k=1}^{K} \sum_{y \in \mathcal{D}} \omega_{k,y}(\mathbf{X})[F_k * \mathbf{D}](y) \right] q(\mathbf{D}) \tag{2}$$

W can be learned by maximum likelihood estimation criteria. Given training data $\{(\mathbf{X}_i, \mathbf{D}_i)\}_{i=1}^m$ the gradient of log-likelihood for energy based model (EBM) follows:

$$L'(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta} \log P_\theta(\mathbf{D}_i|\mathbf{X}_i) = \mathbb{E}_{p_{data}}\left[\frac{\partial}{\partial \theta} \log P_\theta(\mathbf{D}|\mathbf{X}) \right] \tag{3}$$

$$= \mathbb{E}_{p_{data}}\left[\frac{\partial}{\partial \theta} f_\theta(\mathbf{D}|\mathbf{X}) - \frac{\partial}{\partial \theta} \log Z(\theta|\mathbf{X}) \right] \tag{4}$$

$$= \mathbb{E}_{p_{data}}\left[\frac{\partial}{\partial \theta} f_\theta(\mathbf{D}|\mathbf{X}) \right] - \mathbb{E}_{p_\theta}\left[\frac{\partial}{\partial \theta} f_\theta(\mathbf{D}|\mathbf{X}) \right] \tag{5}$$

$$\approx \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta} f_\theta(\mathbf{D}_i|\mathbf{X}_i) - \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial}{\partial \theta} f_\theta(\tilde{\mathbf{D}}_{ij}|\mathbf{X}_i) \tag{6}$$

Due to the intractable normalizing constant $Z(\theta|\mathbf{X})$ during computation, we need to use Monte Carlo samples $\tilde{\mathbf{D}}$ from $P_\theta(\mathbf{D}|\mathbf{X})$ to approximate the gradient update for learning $\theta = W$. Following the work (learning FRAME with CNN filters) we have the following learning algorithm:

Algorithm 1: Learning conditional FRAME model.

input : Initial network θ_0, training steps T, training data $\{(\mathbf{X}_i, \mathbf{D}_i)\}_{i=1}^N$, batch size m, variance of noise σ^2, Langevin descretization δ, Langevin steps n, learning rate η

output: Trained network θ_{T+1}

for $t = 0{:}T$ **do**

1. Draw batch samples $\{(\mathbf{X}_i, \mathbf{D}_i)\}$;

2.**for** i=1 : m **do**

Draw n samples $\{\tilde{\mathbf{D}}_i\}$ from $P_{\theta_t}(\mathbf{D}|\mathbf{X}_i)$ using Langevin dynamics:

$$\tilde{\mathbf{D}}_{\tau+1} = \tilde{\mathbf{D}}_\tau + \frac{\delta}{2} f'_{\theta_t}(\tilde{\mathbf{D}}_\tau|\mathbf{X}_i) + \sqrt{\delta} U_\tau$$

where $U_\tau \sim N(0, \sigma^2 I)$ is Gaussian noise.

end

3. Update θ_t by

$$\theta_{t+1} = \theta_t + g(\Delta(\theta_t), \eta, t)$$

where $\Delta(\theta_t)$ is the approximate gradient from (6).

end

Implementation

The datasets used for experiment and simulation are NYUv2 indoor scene data with labeled depth map and CIFAR10 data with images generated by Langevin sampling.

The energy function in the experiments are implemented either by a convolutional neural network with filters or a fully connected feedforward neural network. The training is done on both datasets and are performed either with response (WR) or only training generative model on input images it self with no response (NR).

Results

Table 1: Training losses (MSE) under different experiment settings

	Single Image	Mutiple Images	Dataset
(WR) Fully-connected	70	Do Not Converge	NYUv2
(WR) ConvNet	112	Do Not Converge	NYUv2
(NR) Fully-connected	0	1.42	NYUv2
(NR) ConvNet	0.11	2.64	NYUv2
(WR) ConvNet	NA	Do Not Converge	CIFAR10
(NR) ConvNet	NA	0.05	CIFAR10

References