

# Regression using Energy-Based Models and Noise Contrastive Estimation

---

Fredrik K. Gustafsson  
Uppsala University

SysCon  $\mu$ -seminar  
February 12, 2021

## **Energy-Based Models for Deep Probabilistic Regression**

*Fredrik K. Gustafsson, Martin Danelljan, Goutam Bhat, Thomas B. Schön*

The European Conference on Computer Vision (ECCV), 2020

## **How to Train Your Energy-Based Model for Regression**

*Fredrik K. Gustafsson, Martin Danelljan, Radu Timofte, Thomas B. Schön*

The British Machine Vision Conference (BMVC), 2020

## **Accurate 3D Object Detection using Energy-Based Models**

*Fredrik K. Gustafsson, Martin Danelljan, Thomas B. Schön*

Preprint

## **Deep Energy-Based NARX Models**

*Johannes Hendriks, Fredrik K. Gustafsson, Antônio Ribeiro, Adrian Wills, Thomas B. Schön*

Preprint

An **energy-based model (EBM)** specifies a probability distribution  $p(x; \theta)$  over  $x \in \mathcal{X}$  directly via a parameterized scalar function  $f_\theta : \mathcal{X} \rightarrow \mathbb{R}$ :

$$p(x; \theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$$

By defining  $f_\theta(x)$  using a deep neural network (DNN),  $p(x; \theta)$  becomes expressive enough to learn practically any distribution from observed data.

An EBM specifies a probability distribution  $p(x; \theta)$  directly via a parameterized scalar function  $f_\theta(x)$ ,

$$p(x; \theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

where  $f_\theta(x)$  commonly is defined using a DNN.

The EBM  $p(x; \theta) = e^{f_\theta(x)} / \int e^{f_\theta(\tilde{x})} d\tilde{x}$  is thus a highly expressive model that puts minimal restricting assumptions on the true distribution  $p(x)$ .

**Drawback:** the normalizing partition function  $Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$  is intractable, which complicates evaluating or sampling from  $p(x; \theta)$ .

(Compare with normalizing flows which are specifically designed to be easy to both evaluate and sample. EBMs instead prioritize maximum expressivity)

The definition of an EBM  $p(x; \theta)$ ,

$$p(x; \theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

includes the intractable  $Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$ .

This complicates evaluating or sampling from  $p(x; \theta)$ .

In particular, EBMs are challenging to train. A variety of different approaches have therefore been explored in literature.

A very recent tutorial on the subject:

## **How to Train Your Energy-Based Models**

*Yang Song, Diederik P. Kingma*

arXiv:2101.03288

**Regression:** learn to predict a continuous target  $y^* \in \mathcal{Y} = \mathbb{R}^K$  from a corresponding input  $x^* \in \mathcal{X}$ , given a training set  $\mathcal{D}$  of i.i.d. input-target pairs,  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ ,  $(x_i, y_i) \sim p(x, y)$ .

We address this task by modelling the distribution  $p(y|x)$  with a *conditional* EBM:

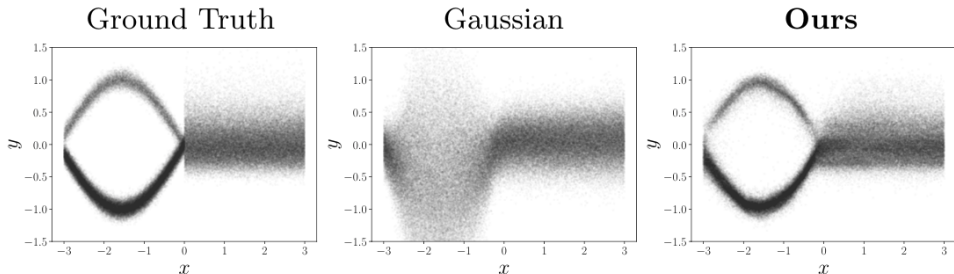
$$p(y|x; \theta) = \frac{e^{f_\theta(x,y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

Here,  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  is a DNN that maps any input-target pair  $(x, y) \in \mathcal{X} \times \mathcal{Y}$  directly to a scalar  $f_\theta(x, y) \in \mathbb{R}$ , and  $Z(x, \theta)$  is the input-dependent partition function.

**EBMs for Regression:** train a DNN  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  to predict a scalar value  $f_\theta(x, y)$ , then model  $p(y|x)$  with the conditional EBM  $p(y|x; \theta)$ :

$$p(y|x; \theta) = \frac{e^{f_\theta(x, y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

The EBM  $p(y|x; \theta)$  can learn complex target distributions directly from data:



**EBMs for Regression:** train a DNN  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  to predict a scalar value  $f_\theta(x, y)$ , then model  $p(y|x)$  with the conditional EBM  $p(y|x; \theta)$ :

$$p(y|x; \theta) = \frac{e^{f_\theta(x, y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems:

- Age estimation,  $\mathcal{Y} = \mathbb{R}$ .
- Head-pose estimation,  $\mathcal{Y} = \mathbb{R}^3$ .
- 2D bounding box regression (object detection, visual tracking),  $\mathcal{Y} = \mathbb{R}^4$ .
- 3D bounding box regression (3D object detection in LiDAR point clouds),  $\mathcal{Y} = \mathbb{R}^7$ .
- System identification,  $\mathcal{Y} = \mathbb{R}$ .



**EBMs for Regression:** train a DNN  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  to predict a scalar value  $f_\theta(x, y)$ , then model  $p(y|x)$  with the conditional EBM  $p(y|x; \theta)$ :

$$p(y|x; \theta) = \frac{e^{f_\theta(x, y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

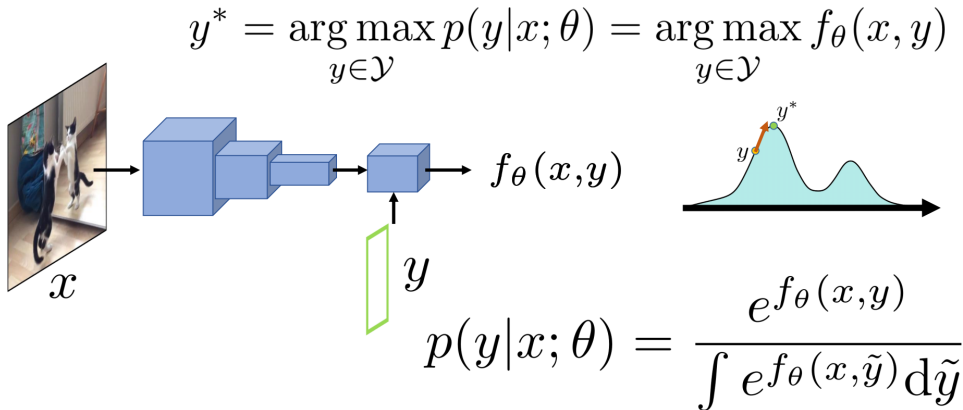
Given an input  $x^*$  at test time, we usually predict  $y^*$  by maximizing  $p(y|x^*; \theta)$ :

$$y^* = \operatorname{argmax}_y p(y|x^*; \theta) = \operatorname{argmax}_y f_\theta(x^*, y)$$

In practice,  $y^* = \operatorname{argmax}_y f_\theta(x^*, y)$  is approximated by refining an initial estimate  $\hat{y}$  via  $T$  steps of gradient ascent,

$$y \leftarrow y + \lambda \nabla_y f_\theta(x^*, y),$$

thus finding a local maximum of  $f_\theta(x^*, y)$ . Evaluation of the partition function  $Z(x^*, \theta)$  is therefore *not* required.



**EBMs for Regression:** train a DNN  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  to predict a scalar value  $f_\theta(x, y)$ , then model  $p(y|x)$  with the conditional EBM  $p(y|x; \theta)$ :

$$p(y|x; \theta) = \frac{e^{f_\theta(x, y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

The DNN  $f_\theta(x, y)$  can be trained using various methods for fitting a density  $p(y|x; \theta)$  to observed data  $\{(x_i, y_i)\}_{i=1}^N$ .

Generally, the most straightforward such method is probably to minimize the negative log-likelihood  $\mathcal{L}(\theta) = -\sum_{i=1}^N \log p(y_i|x_i; \theta)$ , which for the EBM  $p(y|x; \theta)$  is given by,

$$\mathcal{L}(\theta) = \sum_{i=1}^N \log \left( \int e^{f_\theta(x_i, y)} dy \right) - f_\theta(x_i, y_i).$$

$$p(y|x; \theta) = \frac{e^{f_\theta(x,y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = - \sum_{i=1}^N \log p(y_i|x_i; \theta) = \sum_{i=1}^N \log \left( \int e^{f_\theta(x_i, y)} dy \right) - f_\theta(x_i, y_i).$$

The integral  $\int e^{f_\theta(x_i, y)} dy$  is however intractable, preventing exact evaluation of  $\mathcal{L}(\theta)$ .

In **Energy-Based Models for Deep Probabilistic Regression**, we simply approximated this intractable integral using importance sampling.

$$p(y|x; \theta) = \frac{e^{f_\theta(x,y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = - \sum_{i=1}^N \log p(y_i|x_i; \theta) = \sum_{i=1}^N \log \left( \int e^{f_\theta(x_i, y)} dy \right) - f_\theta(x_i, y_i).$$

Importance sampling:

$$\begin{aligned} -\log p(y_i|x_i; \theta) &= \log \left( \int e^{f_\theta(x_i, y)} dy \right) - f_\theta(x_i, y_i) \\ &= \log \left( \int \frac{e^{f_\theta(x_i, y)}}{q(y)} q(y) dy \right) - f_\theta(x_i, y_i) \\ &\approx \log \left( \frac{1}{M} \sum_{k=1}^M \frac{e^{f_\theta(x_i, y^{(k)})}}{q(y^{(k)})} \right) - f_\theta(x_i, y_i), \quad y^{(k)} \sim q(y). \end{aligned}$$

**EBMs for Regression:** train a DNN  $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  to predict a scalar value  $f_\theta(x, y)$ , then model  $p(y|x)$  with the conditional EBM  $p(y|x; \theta)$ :

$$p(y|x; \theta) = \frac{e^{f_\theta(x, y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

Various alternative techniques could however also be employed to train the DNN  $f_\theta(x, y)$ , including noise contrastive estimation (NCE) and score matching.

In **How to Train Your Energy-Based Model for Regression**, we therefore studied in detail how EBMs should be trained specifically for regression problems.

We compared six methods on the task of 2D bounding box regression, and concluded that a simple extension of NCE should be considered the go-to training method.

$$p(y|x; \theta) = \frac{e^{f_\theta(x,y)}}{Z(x, \theta)}, \quad Z(x, \theta) = \int e^{f_\theta(x, \tilde{y})} d\tilde{y}.$$

**Noise contrastive estimation (NCE)** entails learning to discriminate between observed data examples and samples drawn from a noise distribution.

Specifically, the DNN  $f_\theta(x, y)$  is trained by minimizing the loss  $J(\theta) = -\frac{1}{N} \sum_{i=1}^N J_i(\theta)$ ,

$$J_i(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)}|y_i)\}}{\sum_{m=0}^M \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)}|y_i)\}},$$

where  $y_i^{(0)} \triangleq y_i$ , and  $\{y_i^{(m)}\}_{m=1}^M$  are  $M$  samples drawn from a noise distribution  $q(y|y_i)$  that depends on the true target  $y_i$ .

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^N J_i(\theta), \quad J_i(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)}|y_i)\}}{\sum_{m=0}^M \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)}|y_i)\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^M \sim q(y|y_i) \text{ (noise distribution).}$$

Effectively,  $J(\theta)$  is the softmax cross-entropy loss for a classification problem with  $M + 1$  classes (which of the  $M + 1$  values  $\{y_i^{(m)}\}_{m=0}^M$  is the true target  $y_i$ ?).

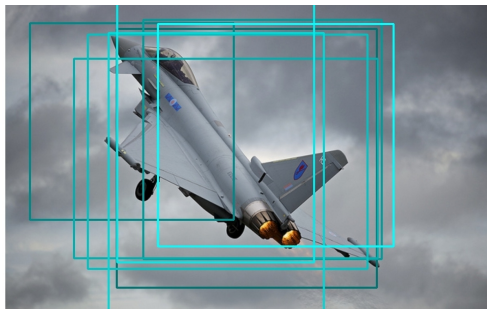
A simple yet effective choice for the noise distribution  $q(y|y_i)$  is a mixture of  $K$  Gaussians centered at  $y_i$ ,

$$q(y|y_i) = \frac{1}{K} \sum_{k=1}^K \mathcal{N}(y; y_i, \sigma_k^2 I).$$



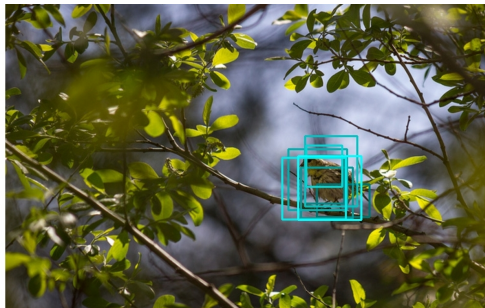
$$J(\theta) = -\frac{1}{N} \sum_{i=1}^N J_i(\theta), \quad J_i(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)} | y_i)\}}{\sum_{m=0}^M \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)} | y_i)\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^M \sim q(y | y_i) \text{ (noise distribution).}$$



$$J(\theta) = -\frac{1}{N} \sum_{i=1}^N J_i(\theta), \quad J_i(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)} | y_i)\}}{\sum_{m=0}^M \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)} | y_i)\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^M \sim q(y | y_i) \text{ (noise distribution).}$$



**Fredrik K. Gustafsson, Uppsala University**

[fredrik.gustafsson@it.uu.se](mailto:fredrik.gustafsson@it.uu.se)

[www.fregu856.com](http://www.fregu856.com)