DL-HW4

Jianing Zhang jz5212

Mar 16th 2024

1.1 Energy Based Models Intuition

(a)

Energy-based models operate on a different principle compared to traditional mapping models. Instead of directly associating an input x_i with a single output y_i, EBMs assign an energy score to each (x_i, y_i) pair through a function $F_W(x, y)$. The core idea is that configurations (x, y) that are more probable or desirable are assigned lower energy values. This framework naturally accommodates situations where an input x_i could correspond to multiple plausible outputs y_i, as each of these output candidates can be associated with a low energy score. Therefore, for a given x, there can exist several y values, each with their own low energy scores, enabling the model to represent many-to-one mappings effectively.

(b)

Its output is not $p(y|x)$, which is the probability of input x having the label y, it outputs the unnormalized score $F_W(x, y)$.

(c)

By using Gibbs distribution to convert the value to distribution, the probability of $p(y|x)$ is

$$p(y|x) = \frac{\exp(-\beta F_W(x, y))}{\int_{y'} \exp(-\beta F_W(x, y'))}$$

(d)

Loss function is used to learn the energy function, which are used for inference. The energy function is utilized during both the training and inference phases. During inference, it is minimized to find the correct target y for a given input x, where the correct target is associated with the lowest energy. During training, the energy function is shaped and refined through the application of the loss function, which measures and optimizes the energy values produced by F over the training dataset. The loss function is designed to reduce the energy of correct (x, y) pairs, thereby increasing the model's ability to identify and favor more probable configurations. Simultaneously, it may penalize incorrect pairs by increasing their energy, effectively pushing the model to differentiate between correct and incorrect answers.

(e)

Using only positive examples for energy minimization can lead to a problem known as "energy collapse," where the model assigns low energy to all possible configurations, not just the correct ones, making it unable to discriminate between correct and incorrect inputs. This can result in a lack of contrast between the energy of good and bad configurations. To avoid this, negative samples or contrasting examples can be included in training to push up the energy of incorrect inputs, ensuring the model learns to distinguish between different types of inputs by maintaining a gap in the energy landscape.

(f)

Contrastive Methods: This method relies on using positive examples, which the model is encouraged to give low energy, alongside negative examples, which are assigned higher energy. The model is trained to differentiate between these contrasting sets, effectively creating a gap in the energy landscape.

Regularization Methods: These techniques involve adding an additional term to the loss function to prevent model collapse. An example is the use of L1 regularization on the model's hidden layers, which promotes sparsity in the representations. This prevents the model from assigning low energy to every possible input by limiting the usable representation space.

Architectural Methods: By designing the model architecture with constraints, such as a bottleneck in an autoencoder with a compact hidden layer, the model's capacity to represent data is limited. Consequently, the model cannot assign low energy to all inputs indiscriminately, as the constrained representation space cannot perfectly encode every input.

(g)

For example: Negative-log likelihood

$$\ell_{nll}(x, y, W) = F_W(x, y) + \log \int_{y'} \exp(-\beta F_W(x, y'))$$

(h)

Inference: The output $\tilde{y}$ that minimizes the energy function for a given input x is given by:

$$\hat{y} = \arg\min_y F(x, y)$$

When including a latent variable z, and the energy function is $G(x, y, z)$, the inference process accounts for this additional variable, leading to:

$$\hat{z} = \arg\min_z G(x, y, z)$$

1.2 Negative log-likelihood loss

(i)

$$p(y|x) = \frac{\exp(-\beta F_W(x, y))}{\int_{y'} \exp(-\beta F_W(x, y'))}$$

(ii)

$$\begin{aligned} \ell(x, y, W) &= -logp(y|x) \\ &= -\log\left(\frac{\exp(-\beta F_W(x, y))}{\int_{y'} \exp(-\beta F_W(x, y'))}\right) \\ &= -\log(\exp(-\beta F_W(x, y))) + \log \int_{y'} \exp(-\beta F_W(x, y')) \\ &= \beta F_W(x, y) + \log \int_{y'} \exp(-\beta F_W(x, y'))\, dy' \end{aligned}$$

We can divide the loss by beta:

$$\ell(x, y, W) = F_W(x, y) + \frac{1}{\beta} \log \int_{y'} \exp(-\beta F_W(x, y'))$$

(iii)

$$\begin{aligned} \frac{\partial \ell(x, y, W)}{\partial W} &= \frac{\partial F_W(x, y)}{\partial W} + \frac{\partial}{\partial W}(\frac{1}{\beta} \log \int_{y'} \exp(-\beta F_W(x, y')) \\ &= \frac{\partial F_W(x, y)}{\partial W} + \frac{1}{\beta}\frac{\partial}{\partial W} \log \int_{y'} \exp(-\beta F_W(x, y')) \\ &= \frac{\partial F_W(x, y)}{\partial W} + \frac{1}{\beta}\frac{1}{\int_{y'} \exp(-\beta F_W(x, y'))}\frac{\partial}{\partial W}\left(\int_{y'} \exp(-\beta F_W(x, y'))\right) \\ &= \frac{\partial F_W(x, y)}{\partial W} + \frac{1}{\beta}\frac{\int_{y'} \exp(-\beta F_W(x, y'))(-\beta)\frac{\partial F_W(x, y')}{\partial W}}{\int_{y'} \exp(-\beta F_W(x, y'))} \\ &= \frac{\partial F_W(x, y)}{\partial W} - \int_{y'} P_W(y'|x)\frac{\partial F_W(x, y')}{\partial W} \end{aligned}$$

This may be intractable because of the integral over y'.
We can use Markov Chain Monte Carlo methods to solve it.

(iv)

The negative log-likelihood function escalates the energy levels in a way that isn't related to the actual proximity to the true y, but rather is tied to the likelihood of the specific predicted y'. This implies that regardless of the proximity between two potential outcomes in a scenario with a continuous y, the exerted force remains constant. As a result, the energy landscape is characterized by very steep declines.

1.3 Comparing Contrastive Loss Functions

(a)

$$\frac{\partial \ell_{\text{simple}}(x,y,\overline{y},W)}{\partial W} = \frac{\partial [F_W(x,y)]^+}{\partial W} + \frac{\partial [m - F_W(x,\overline{y})]^+}{\partial W}$$

$$\frac{\partial [F_W(x,y)]^+}{\partial W} = \begin{cases} 0, & \text{if } F_W(x,y) < 0 \\ \frac{\partial F_W(x,y)}{\partial W}, & \text{otherwise} \end{cases}$$

$$\frac{\partial [m - F_W(x,\overline{y})]^+}{\partial W} = \begin{cases} 0, & \text{if } F_W(x,\overline{y}) > m \\ -\frac{\partial F_W(x,\overline{y})}{\partial W}, & \text{otherwise} \end{cases}$$

(b)

As

$$\ell_{log}(x,y,\overline{y},W) = \log\left(1 + e^{F_W(x,y) - F_W(x,\overline{y})}\right)$$

Therefore,

$$\frac{\partial \ell_{log}(x,y,\overline{y},W)}{\partial W} = \frac{\partial}{\partial W} \log\left(1 + e^{F_W(x,y) - F_W(x,\overline{y})}\right)$$

Applying the chain rule, we get:

$$\frac{\partial \ell_{log}(x,y,\overline{y},W)}{\partial W} = \frac{1}{1 + e^{F_W(x,y) - F_W(x,\overline{y})}} \cdot e^{F_W(x,y) - F_W(x,\overline{y})} \cdot \left(\frac{\partial F_W(x,y))}{\partial W} - \frac{\partial F_W(x,\overline{y})}{\partial W}\right)$$

(c)

Given the square-square loss function defined as follows:

$$\frac{\partial \ell_{\text{square-square}}}{\partial W} = \frac{\partial \left([F_W(x,y)]^+\right)^2}{\partial W} + \frac{\partial \left([m - F_W(x,\overline{y})]^+\right)^2}{\partial W}$$

The partial derivatives are given by:

$$\frac{\partial \left([F_W(x,y)]^+\right)^2}{\partial W} = \begin{cases} 0, & \text{if } F_W(x,y) < 0 \\ 2F_W(x,y)\frac{\partial F_W(x,y)}{\partial W}, & \text{otherwise} \end{cases}$$

$$\frac{\partial\left([m - F_W(x,\overline{y})]^{+}\right)^2}{\partial W} = \begin{cases} 0, & \text{if } F_W(x,\hat{y}) > m \\ -2(m - F_W(x,\overline{y}))\frac{\partial F_W(x,\overline{y})}{\partial W}, & \text{otherwise} \end{cases}$$

(d)

(i)

The Negative Log-Likelihood (NLL) loss function doesn't simply elevate the energy level of a single incorrect prediction; rather, it raises the energy across all predictions. This quality of the NLL means it requires computation of an integral to fully encompass the range of the probability distribution. This requirement is unique to the NLL, as other loss functions do not necessitate such an integral. In situations where the outcomes are continuous, this integral may become exceedingly difficult to compute due to the complexity of accounting for an infinite continuum of outcomes.

(ii)

The hinge loss function includes a margin parameter m that serves to enforce a minimum separation between the correct and incorrect classifications. In the formula $[F_W(x,y) - F_W(x,\overline{y}) + m]^{+}$, the positive part, indicated by the subscript $+$, is taken into account because the goal is to penalize situations where the energy of the correct class $F_W(x,y)$ is not at least m units greater than the incorrect class $F_W(x,\overline{y})$. When the correct classification's confidence exceeds this margin, the loss is zero, indicating that the model's prediction is acceptably strong.

The log loss is sometimes called a "soft hinge" because it provides a continuous, smooth gradient that can penalize misclassifications even within the margin. Unlike the traditional hinge loss, which abruptly ceases to penalize once the margin is achieved, the log loss continues to apply a gradient, thereby ensuring a gradual and more refined model fitting process. This characteristic can lead to better generalization on data where the classes are not perfectly separable or where a strict margin might be too rigid.

(iii)

The simple loss and square-square loss are distinct from hinge/log loss in their approach to handling errors. The hinge loss focuses on maintaining a margin of at least m between the correct and incorrect values' energies, without necessarily driving the correct values' energy to zero. On the other hand, simple loss aims to linearly reduce the energy of correct predictions to zero, and ensure incorrect predictions have energy levels at least m. Square-square loss does something similar but does so in a quadratic fashion.

Regarding when to use each loss function, the simple loss is preferable when the goal is to minimize the impact of outliers, much like the L1 loss, as it does not disproportionately penalize large errors. Square-square loss is analogous to the L2 loss and can be a preferred choice in general because of its propensity to smooth the error surface and provide stable solutions, even though it may be more sensitive to outliers due to its quadratic nature.