DS-GA 1008 HW 4

Long Chen
Center for Data Science, New York University
lc3424@nyu.edu

Fall 2021

1 Theory

1.1 Energy Based Models Intuition

a)

EBM is an implicit funtion that captures dependencies between x and y, and thus there could be multiple "compatible" y's with a single x.

b)

Energy in EBM is essentially an unnormalized negative log probability. Either conditional or unconditional, energy captures the dependency without considering all possible y's.

c)

We could convert energy to probability using Gibbs-Boltzmann Distribution:

$$p(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}},$$

where β is a positive constant.

d)

An energy function measures the "compatibility" of x and y, whereas a loss function measures the "desirability" of y with respect to x in the context of ground truth (just like general loss functions) or other y's (e.g. triplet loss).

e)

Yes, when loss does not consider margin (e.g. energy loss).

f)

Say we have a great amount of positive examples. If we push down the energy function at positive examples, it's possible that for every y, we will eventually have energy of negative infinity, thus leading to a degenerate solution.

g)

- **Contrastive Methods:** learning for larger contrast between positive and negative samples
- **Regularized Latent Variable:** use techniques to make volume of low energy space bounded, then regularize the volume so that we learn a representation $G(x, y)$ that is, to the greatest degree, constant w.r.t y, then constructing the energy function $F_W(x, y) = C(y, G(x, y))$.
- **Generative method:** use generative model to predict $\hat{y}$ and compare with y 's in order to minimize energy for positive samples.

h)

Consider **minimum classification loss:**

$$\mathcal{L}_{MCE}(, y, x, w) = \sigma\left[F_w(x, y) - F_w(x, y')\right],$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and y' is the nearest energy negative sample.

1.2 Negative Log-Likelihood Loss

a)

$$p(y|x) = \frac{e^{-\beta F_w(x,y)}}{\int_{y'} e^{-\beta F_w(x,y')}},$$

b)

$$\begin{aligned}
\mathcal{L}(, x, y, W) &= -\frac{1}{\beta} \log P_W(y|x) \\
&= -\frac{1}{\beta} \left(-\beta\ F_W(x, y)\right) - \left(-\log\left[\int_{y'} e^{-\beta F_W(x,y')}\right]\right) \\
&= F_W(x, y) + \log\left[\int_{y'} e^{-\beta F_W(x,y')}\right]
\end{aligned}$$

c)

$$\begin{aligned}
\frac{\partial \mathcal{L}(,x,y,W)}{\partial W} &= \frac{\partial F_W(x,y)}{\partial W} + \frac{\partial}{\partial W}\left[\log \int_{y'} e^{-\beta F_W(x,y')}\right] \\
&= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\int_{y'} e^{-\beta F_W(x,y')}} \frac{\partial}{\partial W}\left[\int_{y'} e^{-\beta F_W(x,y')}\right] \\
&= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\int_{y'} e^{-\beta F_W(x,y')}} \int_{y'} e^{-\beta F_W(x,y')} \left(-\frac{\partial F_W(x,y')}{\partial W}\right) \\
&= \frac{\partial F_W(x,y)}{\partial W} - \int_{y'} \frac{e^{-\beta F_W(x,y')}}{\int_{y'} e^{-\beta F_W(x,y')}} \frac{\partial F_W(x,y')}{\partial W} \\
&= \frac{\partial F_W(x,y)}{\partial W} - \int_{y'} \frac{\partial F_W(x,y')}{\partial W}\ P_W(y'|x)
\end{aligned}$$

This is in general intractable as the latter part could be non-differentiable to have a closed-form solution. Even differentiable, calculation could be extremely sophisticated. A workaround could be using Monte Carlo method to sample y' from $P_W(y|x)$ and estimate the integral part using a subset of y 's.

d)

As we can see from the previous expression in c), partial derivative of loss function is proportional only to $\frac{\partial F_W(x,y)}{\partial W}$ and is negatively proportional to all other y's. If we do the "push up and down" according to NLL, the only correct sample will be pushed downwards (or towards negative infinity) and all the others (even the closest one) will be pushed upwards (or towards positive infinity). Imagine when the model "saturates" in the sense of learning, intuitively, we will observe very sharp energy surface.

1.3 Comparing Contrastive Loss Functions

Note: $\mathbb{1}(\cdot)$ is the indicator function.

a)

$$\frac{\partial \ell_{simple}}{\partial W} = \mathbb{1}\left[m > F_W(x,\overline{y})\right]\left[-\frac{\partial F_W(x,\overline{y})}{\partial W}\right] + \mathbb{1}\left[F_W(x,y) > 0\right]\left[\frac{\partial F_W(x,y)}{\partial W}\right]$$

b)

$$\frac{\partial \ell_{hinge}}{\partial W} = \begin{cases} \frac{\partial F_W(x,y)}{\partial W} - \frac{\partial F_W(x,\overline{y})}{\partial W}, & \text{if } F_W(x,y) - F_W(x,\overline{y}) \geq -m, \\ 0, & \text{otherwise.} \end{cases}$$

c)

$$\frac{\partial \ell_{square\_square}}{\partial W} = \mathbb{1}\left[F_W(x,y) \geq 0\right]\left[2\ F_W(x,y)\frac{\partial F_W(x,y)}{\partial W}\right] + \mathbb{1}\left[m - F_W(x,\overline{y}) > 0\right]\left[-2(m - F_W(x,\overline{y}))\frac{\partial F_W(x,\overline{y})}{\partial W}\right]$$

d)

a) NLL is continuous and, in most cases, differentiable, while the three losses above is not continuous or differentiable.

b) The margin acts like a constraint to better "separates" positive y and other undesired $\overline{y}$'s.

c) The "push up" and "push down" operations in simple loss and square-square loss happen separately based on conditions, while the two operations in hinge loss happen simultaneously (if any).

2 Coding

All writeups of coding part is inside the jupyter notebook attached. Please note that, for the "finding bad energy path" question, the presented two paths are worse than the optimal path but may not be that bad, since I did not set global random seed for pytorch and the trained model varies from run to run.