CSCI-GA 2572 Deep Learning
Homework 3: Energy-Based Models

Chuanyang Jin

March 12, 2023

1.1 Energy Based Models Intuition (15pt)

This question tests your intuitive understanding of Energy-based models and their properties.

(a) (1pts) How do energy-based models allow for modeling situations where the mapping from input x_i to output y_i is not 1 to 1, but 1 to many?

Solution.

Energy-based models can allow for modeling situations where the mapping from input x_i to output y_i is one-to-many, by assigning a low energy to each compatible y_i for a given input x_i, and a high energy to each incompatible y_i for a given input x_i. □

(b) (2pts) How do energy-based models differ from models that output probabilities?

Solution.

Models that output probabilities usually model the conditional probability distribution over the possible outputs given the inputs. The output with the highest probability is typically chosen as the prediction.

On the other hand, energy-based models assign an energy according to the input and the output based on their compatibility. They are more flexible, and we can use the energy function to calculate a probability as well. □

(c) (2pts) How can you use energy function $F_W(x, y)$ to calculate a probability $p(y|x)$?

Solution.

For continuous y,

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

For discrete y,

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

where β is a positive constant (e.g. $\beta = 1$). □

(d) (2pts) What are the roles of the loss function and energy function?

Solution.

The loss function measures the discrepancy between the predicted output and the true output for a given input.

On the other hand, the energy function assigns an energy to the data based on their compatibility. □

(e) (2pts) What problems can be caused by using only positive examples for energy (pushing down energy of correct inputs only)? How can it be avoided?

Solution.

By pushing down the energies of the correct inputs only, the model may inadvertently push down the energies of the incorrect inputs as well, since it has not been penalized for assigning low energies to them. It may assign low energies to everyone, making it hard to distinguish between positive and negative examples. This can lead to overfitting and a failure to generalize well.

To address this issue, a common approach is to use contrastive methods, where the model is trained not only on positive examples but also on negative, or ”contrastive” examples. In this case, the model is trained to assign lower energies to correct inputs and higher energies to incorrect inputs, which encourages the model to distinguish between them and better generalize to new inputs. □

(f) (2pts) Briefly explain the three methods that can be used to shape the energy function.

Solution.

Architectural shaping involves designing the model architecture in a way that captures the relevant information from the input data. Contrastive shaping encourages the model to push positive examples closer together and negative examples further apart in the energy space. Regularized shaping involves adding regularization terms to the energy function in order to constrain the model behavior. □

(g) (2pts) Provide an example of a loss function that uses negative examples. The format should be as follows $l_{example}(x, y, W) = F_W(x, y)$

Solution.

Provided the input sample x, the positive example y, and a negative example $\bar{y}$, a simple contrastive loss function is defined as follows:

$$l(x, y, \bar{y}, W) = [F_W(x, y)]^+ + [m - F_W(x, \bar{y})]^+$$

The first term in the loss function encourages the model to learn to correctly classify the positive example as similar to the input sample, and the second term encourages the model to learn to correctly classify the negative example as dissimilar to the input sample. □

(h) (2pts) Say we have an energy function $F(x, y)$ with images x, classification for this image y. Write down the mathematical expression for doing inference given an input x. Now say we have a latent variable z, and our energy is $G(x, y, z)$. What is the expression for doing inference then?

Solution.

Assuming that $F(x, y)$ is an energy function that takes an image x and a classification score y, the mathematical expression for doing inference can be written as:

$$\hat{y} = \underset{y}{\operatorname{argmin}} F(x, y)$$

Now, if we have a latent variable z and an energy function $G(x, y, z)$, the expression for doing inference would involve minimizing $F_\infty(x, y)$:

$$\hat{y} = \underset{y}{\operatorname{argmin}} F_\infty(x, y) = \underset{y}{\operatorname{argmin}} \left(\min_z G(x, y, z) \right)$$

□

1.2 Negative log-likelihood loss (20 pts)

Let's consider an energy-based model we are training to do classification of input between n classes. $F_W(x, y)$ is the energy of input x and class y. We consider n classes: $y \in \{1, ..., n\}$.

(a) (2pts) For a given input x, write down an expression for a Gibbs distribution over labels y that this energy-based model specifies. Use β for the constant multiplier.

Solution.

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

where β is a positive constant. □

(b) (5pts) Let's say for a particular data sample x, we have the label y. Give the expression for the negative log likelihood loss, i.e. negative log likelihood of the correct label (show step-by-step derivation of the loss function from the expression of the previous subproblem). For easier calculations in the following subproblem, multiply the loss by $\frac{1}{\beta}$.

Solution.

The negative log likelihood loss of the correct label y for a particular data sample x is given by:

$$\begin{aligned}
\mathcal{L} &= -\frac{1}{\beta} \log p(y|x) \\
&= -\frac{1}{\beta} \log \frac{e^{-\beta F_W(x,y)}}{\sum_{y'} e^{-\beta F_W(x,y')}} \\
&= -\frac{1}{\beta}(-\beta F_W(x, y)) + \frac{1}{\beta} \log \sum_{y'} e^{-\beta F_W(x,y')} \\
&= F_W(x, y) + \frac{1}{\beta} \log \sum_{y'} e^{-\beta F_W(x,y')}
\end{aligned}$$

□

(c) (8pts) Now, derive the gradient of that expression with respect to W (just providing the final expression is not enough). Why can it be intractable to compute it, and how can we get around the intractability?

Solution.

The gradient of that expression with respect to W is given by:

$$\begin{aligned}\nabla_W \mathcal{L} &= \nabla_W F_W(x,y) + \frac{1}{\beta}\nabla_W \log \sum_{y'} e^{-\beta F_W(x,y')} \\ &= \nabla_W F_W(x,y) + \frac{1}{\beta}\frac{\nabla_W \sum_{y'} e^{-\beta F_W(x,y')}}{\sum_{y'} e^{-\beta F_W(x,y')}} \\ &= \nabla_W F_W(x,y) + \frac{1}{\beta}\frac{\sum_{y'} e^{-\beta F_W(x,y')}\nabla_W(-\beta F_W(x,y'))}{\sum_{y'} e^{-\beta F_W(x,y')}} \\ &= \nabla_W F_W(x,y) - \frac{\sum_{y'} e^{-\beta F_W(x,y')}\nabla_W F_W(x,y')}{\sum_{y'} e^{-\beta F_W(x,y')}} \\ &= \nabla_W F_W(x,y) - \sum_{y'} \frac{e^{-\beta F_W(x,y')}}{\sum_{y''} e^{-\beta F_W(x,y'')}}\end{aligned}$$

The gradient of $F_W(x,y)$ can be computed by:

$$\nabla_W F_W(x,y) = \nabla_W(\boldsymbol{w}_{\boldsymbol{y}}^\top \boldsymbol{\phi}(x)) = \boldsymbol{\phi}(x)\boldsymbol{e}_{\boldsymbol{y}}^\top$$

where $\boldsymbol{\phi}(x)$ is the feature vector representation of the input x, $\boldsymbol{w}_{\boldsymbol{y}}$ is the weight vector for class y, and $\boldsymbol{e}_{\boldsymbol{y}}$ is a one-hot encoding vector with a 1 in the y'-th or y-th position and 0 everywhere else.

It can be intractable to compute the gradient because it involves computing the normalization term $\sum_{y''=1}^{n} e^{-\beta F_W(x,y'')}$ for all possible labels, which can be computationally expensive when the number of labels n is large.

One way to get around this intractability is to randomly sample a mini-batch of the training examples instead of computing the gradient over the entire set. □

(d) (5pts) Explain why negative log-likelihood loss pushes the energy of the correct example to negative infinity, and all others to positive infinity, no matter how close the two examples are, resulting in an energy surface with really sharp edges in case of continuous y (this is usually not an issue for discrete y because there's no distance measure between different classes).

Solution.

From the expression of the negative log-likelihood loss function:

$$\mathcal{L} = -\frac{1}{\beta}\log p(y|x) = F_W(x,y) + \frac{1}{\beta}\log \sum_{y'=1}^{n} e^{-\beta F_W(x,y')}$$

we can see that minimizing the loss function is equivalent to pushing down the energy $F_W(x,y)$ of the correct label y and pushing up the energy $F_W(x,y')$ of all other labels y'.

As a result, the negative log-likelihood loss function creates a sharp discontinuity in the case of continuous labels, pushing the energy of the correct labels very low and the energies of the other neighboring labels very high. This can result in a very steep energy surface with sharp edges, making it difficult for the model to generalize well to new, unseen examples, as small changes in the input can result in large changes in the predicted label due to the sharp energy surface. □

1.3 Comparing Contrastive Loss Functions (15pts)

In this problem, we're going to compare a few contrastive loss functions. We are going to look at the behavior of the gradients, and understand what uses each loss function has. In the following subproblems, m is a margin, $m \in \mathbb{R}$, x is input, y is the correct label, $\bar{y}$ is the incorrect label. Define the loss in the following format: $l_{\text{example}}(x, y, \bar{y}, W) = F_W(x, \bar{y})$.

(a) (3pts) **Simple loss function** is defined as follows:

$$l_{\text{simple}}(x, y, \bar{y}, W) = [F_W(x, y)]^+ + [m - F_W(x, \bar{y})]^+$$

Assuming we know the derivative $\frac{\partial F_W(x,y)}{\partial W}$ for any x, y, give an expression for the partial derivative of the l_{simple} with respect to W.

Solution.

$$\begin{aligned} \frac{\partial l_{\text{simple}}}{\partial W} &= \frac{\partial [F_W(x,y)]^+}{\partial F_W(x,y)} \cdot \frac{\partial F_W(x,y)}{\partial W} - \frac{\partial [m - F_W(x,\bar{y})]^+}{\partial F_W(x,\bar{y})} \cdot \frac{\partial F_W(x,\bar{y})}{\partial W} \\ &= [F_W(x,y) > 0] \cdot \frac{\partial F_W(x,y)}{\partial W} - [F_W(x,\bar{y}) < m] \cdot \frac{\partial F_W(x,\bar{y})}{\partial W} \end{aligned}$$

where the square brackets "[·]" represent the Iverson brackets, which evaluates to 1 if the statement inside is true, and 0 otherwise. □

(b) (3pts) **Log loss** is defined as follows:

$$l_{\log}(x, y, \bar{y}, W) = \log(1 + e^{F_W(x,y) - F_W(x,\bar{y})})$$

Assuming we know the derivative $\frac{\partial F_W(x,y)}{\partial W}$ for any x, y, give an expression for the partial derivative of the $l_{\log}$ with respect to W.

Solution.

Using the chain rule, the partial derivative of $l_{\log}$ with respect to W can be written as:

$$\begin{aligned} \frac{\partial l_{\log}}{\partial W} &= \frac{e^{F_W(x,y) - F_W(x,\bar{y})}}{1 + e^{F_W(x,y) - F_W(x,\bar{y})}} \cdot \frac{\partial}{\partial W}(F_W(x,y) - F_W(x,\bar{y})) \\ &= \frac{e^{F_W(x,y) - F_W(x,\bar{y})}}{1 + e^{F_W(x,y) - F_W(x,\bar{y})}} \cdot \left(\frac{\partial F_W(x,y)}{\partial W} - \frac{\partial F_W(x,\bar{y})}{\partial W} \right) \end{aligned}$$

□

(c) (3pts) **Square-Square loss** is defined as follows:

$$l_{\text{square-square}}(x, y, \bar{y}, W) = ([F_W(x, y)]^+)^2 + ([m - F_W(x, \bar{y})]^+)^2$$

Assuming we know the derivative $\frac{\partial F_W(x,y)}{\partial W}$ for any x, y, give an expression for the partial derivative of the $l_{\text{square-square}}$ with respect to W.

Solution.

$$\frac{\partial l_{\text{square-square}}}{\partial W} = 2[F_W(x,y)]^+ \frac{\partial F_W(x,y)}{\partial W} - 2[m - F_W(x,\bar{y})]^+ \frac{\partial F_W(x,\bar{y})}{\partial W}$$

□

(d) (6pts) **Comparison.**

(i) (2pts) Explain how NLL loss is different from the three losses above.

Solution.

The Negative Log Likelihood (NLL) loss is commonly used in classification problems, and measures how well the model's predicted probability distribution over labels matches the true labels.

In contrast, the three contrastive losses above are used to learn embeddings for input examples such that examples with the same labels are embedded close to each other in the embedding space, while examples with different labels are embedded far apart. □

(ii) (2pts) The hinge loss $[F_W(x,y) - F_W(x,\bar{y}) + m]^+$ has a margin parameter m, which gives 0 loss when the positive and negative examples have energy that are m apart. The log loss is sometimes called a "soft-hinge" loss. Why? What is the advantage of using a soft hinge loss?

Solution.

The log loss is a "soft" version of the hinge loss because it assigns a non-zero loss even when the positive and negative examples have energy that are more than m apart rather than completely ignoring them as the hinge loss does.

The advantage of using the soft hinge loss is that it allows for non-zero gradients and continuous optimization. While the hinge loss may produce models with sharp decision boundaries that are sensitive to small changes in the data, the log loss can produce more robust models with smoother decision boundaries that are less likely to overfit. □

(iii) (2pts) How are the simple loss and square-square loss different from the hinge/log loss? In what situations would you use the simple loss, and in what situations would you use the square-square loss?

Solution.

Simple loss and square square loss attempts to obtain specific energies for positive and negative examples, while hinge loss/log loss only consider the relative relationship between them.

The simple loss function may be preferred when the data contains a large number of outliers, as it is less sensitive to extreme errors than the square-square loss function. On the other hand, the square-square loss function may be preferred when the data is relatively noise-free and the goal is to avoid extreme errors. □