Homework 3: Energy-Based Models

CSCI-GA 2572 Deep Learning

Spring 2021

The goal of homework 3 is to test your understanding of Energy-Based Models, and to show you one application in structured prediction.

In the theoretical part, we'll mostly test your intuition. You'll need to write brief answers to questions about how EBMs work. In part 2, we will implement a simple optical character recognition system.

In part 1, you should submit all your answers in a pdf file. As before, we recommend using LaTeX.

For part 2, you will implement some neural networks by adding your code to the provided ipynb file.

As before, please use numerator layout.

The due date of homework 3 is 11:55pm `03/28`. Submit the following files in a zip file `your_net_id.zip` through NYU classes:

- `hw3_theory.pdf`
- `hw3_practice.ipynb`

Note: we will subtract points for Campuswire posts containing solutions to problems. Campuswire shouldn't be a platform where you can get your solution checked, the goal is to help you with any misunderstandings associated with the homework.

The following behaviors will result in penalty of your final score:

1. 10% penalty for submitting your file without using the correct naming format (including naming the zip file, PDF file or python file wrong, adding extra files in the zip folder, like the testing scripts in your zip file).
2. 20% penalty for late submission within the first 24 hours after the deadline. We will not accept any late submission after the first 24 hours.
3. 20% penalty for code submission that cannot be executed following the steps we mentioned.

1 Theory (50pt)

1.1 Energy Based Models Intuition (15 pts)

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?

Answer: Energy-based models don't map input x to output y, they map pairs x, y to energy value, with most likely values having the lowest energy. For one x, multiple different values of y may have low energy $F_W(x, y)$.

(b) (2pts) How do energy-based models differ from models that output probabilities? **Answer:** Energy-based models don't output the probability $p(y \mid x)$ of input x having label y, they output the unnormalized score $F_W(x, y)$.

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

Answer: In order to calculate the probability, we'd need to convert the values to distribution by using Gibbs distribution:

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

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

Answer: Energy function is a measure of compatibility of variables, and loss function is what we use to shape the energy function.

(e) (2pts) Can loss function be equal to the energy function?

Answer: Yes, it's a loss function known ass energy function.

$$\ell_{\text{energy}}(x, y, W) = F_W(x, y)$$

(f) (2pts) Why using only positive examples for energy (pushing down energy of correct inputs only) may lead to a degenerate solution?

Answer: Not pushing the energy of incorrect values up may lead to a flat energy surface, where energy is 0 everywhere. This happens if the model is not restricted in lowering the energy, and can minimize the loss by making energy 0 everywhere.

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

Answer:

- Contrastive methods. They shape energy function by having positive examples, the energy of which is pushed down, and negative examples, energy of which is pushed up.
- Regularization methods. Regularization methods introduce a loss term that prevents the model from collapsing. One example is introducing L1 loss on the hidden representation, making the model choose sparse representations, preventing it from using the entire hidden representation space and getting low energy everywhere.
- Architectural methods. These methods limit the volume of the space that can have low energy by means of architecture. One example is an autoencoder with a small hidden representation. Because the hidden representation is small, the model can't represent any data perfectly and get low energy everywhere.

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

Answer: Negative-log likelihood is an example of a loss with negative examples:

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

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: $y \in \{1, \ldots, n\}$.

(i) (2 pts) 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.

Answer:

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

(ii) (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 (don't copy expressions from the slides, 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 $\frac{1}{\beta}$.

Answer:

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

We can divide the loss by beta to have a simpler expression later.

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

(iii) (8 pts) 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?

Answer:

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

This may be intractable because of the integral over y'. One way to get around the intractability is to use Markov Chain Monte Carlo methods.

(iv) (5 pts) 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).

Answer: Negative log likelihood pushes the energy up with a force not proportional to the distance from the correct y, but proportional to the probability of that particular y'. This means that no matter how close two values are in continuous y case, the force will be the same. Therefore the end result should be a very sharp drop in the energy surface.

1.3 Comparing Contrastive Loss Functions (15 pts)

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: $\ell_{example}(x, y, \bar{y}, W) = F_W(x, y)$.

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

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

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.

Answer:

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

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

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

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

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

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_{hinge} with respect to W.

Answer:

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

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

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

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 $\ell_{\text{square-square}}$ with respect to W.

Answer:

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

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

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

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

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

(ii) (2 pts) What is the role of the margin in hinge loss? Why do we take only the positive part of $F_W(x, y) - F_W(x, \bar{y}) + m$?

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

Answer:

(i) NLL doesn't just push up one negative example, it pushes all examples up. Because of that NLL is a loss that demands an integral calculation. The other loss functions don't need that. The integral can be intractable in continuous cases.

(ii) The margin in hinge loss is the desired value by which the positive examples should differ from negative examples. We only take the positive part of that expression because once the difference is greater than m, the expression will be negative. Since we don't want to do anything if the margin is already greater than m, we only take the positive part.

(iii) Hinge loss differs from the other two because it only enforces relative values. The correct values' energy will not necessarily be 0, but it will be lower than incorrect values' energy by at least m. Simple loss and square-square loss differ in how they push the positive energy to 0 and negative energy to at least m. Simple loss pushes linearly, while square-square pushes quadratically. Similarly to L1 and L2 losses, we'd want to use L1 if we don't want to be susceptible to outliers, but in general L2 is the more popular choice.

2 Implementation (50pt)

Please add your solutions to this notebook `hw3_practice.ipynb` . **Plase use your NYU account to access the notebook.** The notebook contains parts marked as `TODO`, where you should put your code or explanations. The notebook is a Google Colab notebook, you should copy it to your drive, add your solutions, and then download and submit it to NYU Classes. You're also free to run it on any other machine, as long as the version you send us can be run on Google Colab.