Homework 4: Energy-Based Models

Deep Learning

Spring 2024

The goal of homework 4 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.

Submit the following files in a zip file `your_net_id.zip` through NYU classes:

- `hw4_theory.pdf`
- `hw4_impl.ipynb`

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 code submission that cannot be executed following the steps we mentioned.

1 Theory (50pt)

1.1 Energy Based Models Intuition (15pts)

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?

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

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

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

(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?

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

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

(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?

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) (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.

(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 (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 $\frac{1}{\beta}$.

(iii) (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?

(iv) (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).

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

(a) (3pts) **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.

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

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

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 $\ell_{\log}$ with respect to W.

(c) (3pts) **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.

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

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

(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?

(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?

2 Implementation (50pt)

Please add your solutions to this notebook `hw3_impl.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.