Homework 3: Energy-Based Models

CSCI-GA 2572 Deep Learning

Fall 2024

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.

The due date of homework 3 is 11:55pm `10/20`. Submit the following files in a zip file `your_net_id.zip` through NYU classes:

- `hw3_theory.pdf`
- `hw3_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. 10% penalty for every extra day of lateness. Up to 4 days max (after that we won't accept submission).
3. 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, or even 1 to an infinite continuum of y?

> We are mapping pairs (x, y) to a scalar energy value and find that the most likely values of y have a low $F(x, y)$. We then observe that, for each x, we can have several different values y that have this low energy.
> Note a good definition from DeepAI: "Energy-Based Models (EBMs) discover data dependencies by applying a measure of compatibility (scalar energy) to each configuration of the variables. For a model to make a prediction or decision (inference) it needs to set the value of observed variables to 1 and finding values of the remaining variables that minimize that energy level. In the same way, machine learning consists of discovering an energy function that assigns low energies to the correct values of the remaining variables, and higher energies to the incorrect values. A so-called loss functional, that is minimized during training, is used to measure the quality of the energy functions. Within this framework, there are many energy functions and loss functionals allows available to design different probabilistic and non-probabilistic statistical models."
> Words from Alf: "We would like the energy function to be smooth and differentiable so that we can use it to perform the gradient-based method for inference. In order to perform inference, we search this function using gradient descent to find compatible y's. There are many alternate methods to gradient methods to obtain the minimum."

(b) (1pts) How do energy-based models differ from models that output probabilities?

> As is the key to their flexibility, we need not concern ourselves with normalization as EBMs output an unnormalized scalar (score) of $F(x, y)$ as opposed to conditional probabilities (i.e $\mathbb{P}(y|x)$ would later require an estimate of normalization).

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

We can view energies as unnormalised negative log probabilities, and use Gibbs-Boltzmann distribution to convert from energy to probability (with normalization and calibrated β):

$$\mathbb{P}(y|x) = \frac{\exp(-\beta F(x,y))}{\int_{y'} \exp(-\beta F(x,y'))}$$

Note: β is postive constant and larger values produce models with more variance whereas smoother ones are produced with smaller values.

(d) (1pt) Is there a way to control the smoothness of the probability distribution $p(y|x)$ estimated from the energy function $F_W(x,y)$? How do we reduce the variance of $p(y|x)$?

The temperature affects the smoothness of the distribution: higher temperatures make the distribution smoother by spreading the probabilities more evenly across the possible outcomes y. Lower temperatures make the distribution sharper by increasing the contrast between higher and lower probability outcomes. Accordingly, variance can be reduced by increasing the temperature, which makes the distribution smoother and reduces overfitting to specific outcomes.

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

The energy function is a measure of incompatibility between variables (for us, usually the input x and output y) whereas the loss function is used to mold the energy function (we minimize loss to end up with a well-behaved energy function). Note that the cost is how far prediction $\hat{y}$ is from target y. As Yann mentions: A loss functional, minimized during learning, is used to measure the quality of the available energy functions. A distinction should be made between the energy function, which is minimized by the inference process, and the loss functional, which is minimized by the learning process.

(f) (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?

We may get a case of having energy be 0 everywhere, which is a valid minimization of the (flat) energy surface under this constraint. As this flat model can reach every location of the space, the distance between any two points (such as the length of the latent vector spanning the embedded model manifold's reconstructed $\tilde{y} = Wz$ to observed target y) is 0, hence at the minimum energy by default. To avoid this state of collapse case, we can augment $y = [1, y]^T$ to give an additional degree of freedom to dictionary $W = [1, W]^T$, so that we can now intersect any point in the 2D space but only at those points located at the specific height (here, 1) that gives the minimum energy near 0.

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

Regularization Methods: if the latent variable z is too expressive power in producing the final prediction $\tilde{y}$ then every true output y will be a perfect reconstruction from input x at the optimized latent $\check{z}$. We can then limit the volume of space of z (say, with L1 loss to promote sparsity) and therby reduce the regions of y with low energy, preventing the case of getting energy 0 everywhere.

Contrastive Methods: Push down the energy of training data points, $F(X_i, Y_i)$, while pushing up energy on everywhere else, $F(X_i, Y')$.

Architectural Methods: The manifold is of lower dimension than the ambient space so the data cannot be reconstructred perfectly. Autoencoders can reduce the dimensionality of the input in the hidden layer (under-complete) and thus cannot reconstruct data perfectly, preventing collapse. If the hidden space is over-complete, we can use encoding with higher dimensionality than the input to make optimization easier.

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

$$\ell_{NLL}(x, y, W) = -\log\left(\frac{\exp(-\beta F(x, y))}{\int_{y'} \exp(-\beta F(x, y'))}\right)$$

(i) (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?

$$\check{y} = \arg\min_{y} F(x, y)$$

$$\check{z}, \check{y} = \arg\min_{y,z} G(x, y, z)$$

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

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

$$F_\beta(x, y) = -\frac{1}{\beta} \log \int_z \exp(-\beta G(x, y, z))$$

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

$$\begin{aligned}
\ell_{NLL}(x, y, W) &= -\log\left(\frac{\exp(-\beta F_W(x, y))}{\int_{y'} \exp(-\beta F_W(x, y'))}\right) \\
&= \log\left(\int_{y'} \exp(-\beta F_W(x, y'))\right) - \log(\exp(-\beta F_W(x, y)) \\
&= \log\left(\int_{y'} \exp(-\beta F_W(x, y'))\right) + \beta F_W(x, y)) \\
&\implies \frac{1}{\beta} \log\left(\int_{y'} \exp(-\beta F_W(x, y'))\right) + F_W(x, y)
\end{aligned}$$

Where the last step was from dividing by β.

(iii) (8pts) Now, derive the gradient of that expression with respect to W (just providing the final expression is not enough). Your final answer may contain the expression $\frac{\partial F_W(\dots)}{\partial W}$. Why can it be intractable to compute it, and how can we get around the intractability?

$$
\begin{aligned}
\frac{\partial \ell_N LL(x,y,W)}{\partial W} &= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\beta}\frac{\partial \log \int_{y'} \exp(-\beta F_W(x,y'))}{\partial W} \\
&= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\beta}\frac{\frac{\partial \int_{y'} \exp(-\beta F_W(x,y'))}{\partial W}}{\int_{y'} \exp(-\beta F_W(x,y'))} \\
&= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\beta}\frac{1}{\int_{y'} \exp(-\beta F_W(x,y'))}\int_{y'} -\beta \exp(-\beta F_W(x,y'))\frac{\partial F_W(x,y')}{\partial W} \\
&= \frac{\partial F_W(x,y)}{\partial W} - \int_{y'} \frac{\exp(-\beta F_W(x,y')}{\int_{y''} \exp(-\beta F(x,y''))}\frac{\partial F(x,y')}{\partial W} \\
&= \frac{\partial F_W(x,y)}{\partial W} - \int_{y'} \mathbb{P}_\beta(y'|x)\frac{\partial F_W(x,y')}{\partial W}
\end{aligned}
$$

It may be intractable to compute the integral term over all $y' \in Y$; we can get around this by perhaps by applying Markov chain Monte Carlo (MCMC) methods, sequential Monte Carlo (SMC) methods, importance sampling, or the forward-backward algorithm.

(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).

NLL pulls all negative examples up with force proportional to the probability of the y' in question (see Energy Classification slides). This is not a distance metric so proximity of two examples is irrelevant. For the continuous case, this force remains the same regardless of proximity of examples so we end up with sharp edges over the 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: $\ell_{example}(x, y, \bar{y}, W) = F_W(x, y)$.

(a) (2pts) **Simple loss function** is defined as follows:

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

Where $[z]^+ = max(0, z)$

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.

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

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

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

(b) (2pts) **Hinge Loss** is defined as follows:

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

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_{hinge} with respect to W.

$$\frac{\partial \ell_{\text{hinge}}(x,y,\bar{y},W)}{\partial W} = \frac{\partial F_W(x,y)}{\partial W} - \frac{\partial F_W(x,\bar{y})}{\partial W}.$$

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

(c) (2pts) **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.

$$\frac{\partial \log(1+\exp(F_W(x,y)-F_W(x,\bar{y})))}{\partial W} =$$

$$\frac{(\exp(F_W(x,y)-F_W(x,\bar{y})))\left(\frac{\partial F_W(x,y)}{\partial W}-\frac{\partial F_W(x,\bar{y})}{\partial W}\right)}{1+\exp(F_W(x,y)-F_W(x,\bar{y}))}$$

$$=\frac{\exp(F_W(x,y))\left(\frac{\partial F_W(x,y)}{\partial W}-\frac{\partial F_W(x,\bar{y})}{\partial W}\right)}{\exp(F_W(x,y))+\exp(F_W(x,\bar{y}))}$$

(d) (2pts) **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.

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

$$\frac{\partial(F_W(x,y)^+)^2}{\partial W}+\frac{\partial((m-F_W(x,\bar{y}))^+)^2}{\partial W}$$

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

$$\frac{\partial((m-F_W(x,\bar{y}))^+)^2}{\partial W}=\begin{cases}-2(m-F_W(x,\bar{y}))\frac{\partial F_W(x,\bar{y})}{\partial W} & \text{if } F_W(x,\bar{y})<m\\ 0 & \text{otherwise}\end{cases}$$

(e) (7pts) **Comparison.**

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

NLL loss can be applied in the continuous case, in which it can prove an intractable calculation. Also note that NLL loss pulls up on all negative examples at a time.

(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?

By taking a margin of m, we notice the following:

$$\begin{aligned} &\exp(F_W(x,y) - F_W(x,\bar{y}) + m) \propto (1 + \exp(F_W(x,y) - F_W(x,\bar{y}))) \\ \implies &m \propto \log(1 + \exp(F_W(x,y) - F_W(x,\bar{y}))) - (F_W(x,y) - F_W(x,\bar{y})) \\ &\therefore m + F_W(x,y) - F_W(x,\bar{y}) \propto \log(1 + \exp(F_W(x,y) - F_W(x,\bar{y}))) \end{aligned}$$

Thus we 'soften' the exponential representation by using log to arrive at log loss, noting that it is proportional to hinge loss for this given margin. The soft-hinge loss stabilizes the exponential (hinge loss) representation and is resilient to over/underflow. We also guarantee that the output won't be at a vastly different scale than the input using the log.

(iii) (2pts) How are the simple loss and square-square loss different from the hinge/log loss?

Square-square loss pushes the positive examples' energy towards 0 and pulls negative examples' energy away from 0 quadratically, whereas simple loss does this linearly. In comparison, hinge/log loss concerns itself with the difference in energy between the positive and negative examples. Hinge loss' positive examples may have lower energy than those of negative by at least the margin m so it may not produce energy of 0. Use simple loss if the model does not need to be sensitive to outliers we come across and use square-square loss if it does.

(iv) (1pt) In what situations would you use the simple loss, and in what situations would you use the square-square loss?

2 Implementation (50pt + 10pt extra credit)

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.