Noname manuscript No.
(will be inserted by the editor)

Entropy loss function for unsupervised deep learning

Dylan M. Paiton
Sheng Y. Lundquist

Abstract Here we describe a loss function for training a typical deep network in an unsupervised way. The objective of the loss is to encourage the deep network to minimize the entropy of its output while still conveying information. This should result in an intelligent clustering of object categories without the need of human generated object labels.

Keywords entropy · deep network · loss functions · unsupervised learning

1. A Few Notes

The input to the Entropy function is X, a vector of N neurons. The energy loss will be accumulated over a batch of M images.

2. Useful Derivations

The following are a handful of derivatives that have to be taken multiple times for the entropy gradient calculation.

$$\frac{\partial \sum_i X_i}{\partial X_k} = \frac{\partial X_k}{\partial X_k} + \frac{\partial \sum_{i \neq k} X_i}{\partial X_k} = 1 \tag{1}$$

$$\frac{\partial \sum_i e^{-\beta X_i}}{\partial X_k} = \frac{\partial e^{-\beta X_k}}{\partial X_K} + \frac{\partial \sum_{i \neq k} e^{-\beta X_i}}{\partial X_k} = -\beta e^{-\beta X_k} \tag{2}$$

DM Paiton
Vision Science Graduate Group
Redwood Center for Theoretical Neuroscience
University of California, Berkeley
E-mail: dpaiton@berkeley.edu

$$\begin{aligned}\frac{\partial \sum_i \beta X_i e^{-\beta X_i}}{\partial X_k} &= \frac{\partial \beta X_k e^{-\beta X_k}}{\partial X_k} + \frac{\partial \sum_{i \neq k} \beta X_i e^{-\beta X_i}}{\partial X_k} \\ &= \frac{\partial \beta X_K}{\partial X_k} e^{-\beta X_k} + \beta X_k \frac{\partial e^{-\beta X_k}}{\partial X_k} \\ &= \beta e^{-\beta X_k} + \beta X_k (-\beta e^{-\beta X_k}) \\ &= \beta e^{-\beta X_k} (1 - \beta X_k)\end{aligned} \tag{3}$$

$$\frac{\partial \ln \sum_j e^{-\beta X_j}}{\partial X_k} = \frac{1}{\sum_j e^{-\beta X_j}} \frac{\partial \sum_j e^{-\beta X_j}}{\partial X_k} = \frac{-\beta e^{-\beta X_k}}{\sum_j e^{-\beta X_j}} \tag{4}$$

$$\begin{aligned}q_i &= \frac{e^{-\beta X_i}}{\sum_j e^{-\beta X_j}} \\ \frac{\partial q_i}{\partial X_i} &= \frac{-\sum_j e^{-\beta X_j} \beta e^{-\beta X_i} + \beta (e^{-\beta X_i})^2}{(\sum_j e^{-\beta X_j})^2} \\ \frac{\partial q_i}{\partial X_i} &= \beta q_i \left(q_i - 1\right)\end{aligned} \tag{5}$$

3. Forward Function - Entropy Computation

Our loss should include a term for minimizing entropy as well as a term for preserving information. We want the network output distribution for a given image to be a delta function, although we want the average distribution over all images to be uniform. Thus, we want

$$\underset{X(\phi)}{\text{argmin}} < H[q(c|I)] >_I \text{subject to} < q(c|I) >_I = 1/N,$$

where N is the number of nodes, I is an image, c is an object category, $H[]$ is the entropy, q is the output distribution, $<>_I$ indicates an average over all images in a mini-batch, and the minimization is performed over all network parameters, $X(\phi)$.

First we will derive the equation for entropy. We define our output probability from the network as

$$q_i = \frac{e^{-\beta X_i}}{\sum_j e^{-\beta X_j}}, \tag{6}$$

where q is computed per neuron for a single input image. For $\beta = -1$, equation 6 is the softmax function on X. This can then plug into the entropy equation:

$$H = -\sum_i q_i log(q_i), \tag{7}$$

which expands to:

$$H = \sum_i \left(\frac{e^{-\beta X_i}}{\sum_j e^{-\beta X_j}} \left(\beta X_i + \ln \sum_j e^{-\beta X_j} \right) \right)$$

$$H = \sum_i \frac{\beta X_i e^{-\beta X_i}}{\sum_j e^{-\beta X_j}} + \sum_i \frac{e^{-\beta X_i}}{\sum_j e^{-\beta X_j}} \ln \sum_j e^{-\beta X_j}$$

$$H = \overbrace{\sum_i \frac{\beta X_i e^{-\beta X_i}}{\sum_j e^{-\beta X_j}}}^{\text{Left Term}} + \overbrace{\frac{\ln \sum_j e^{-\beta X_j}}{\sum_j e^{-\beta X_j}} \sum_i e^{-\beta X_i}}^{\text{Right Term}}. \tag{8}$$

Reducing this loss will reduce the total entropy of the output distribution for a given image. If the model is solved via batch learning, a cumulative sum will need to be taken over M batch images. This should encourage the network to cluster inputs into discrete categories and produce a sparse output. However, we also wish to preserve information and avoid the pathological solution of clustering all inputs into a single everything category. To do this, we will introduce an additional loss that will compute the average output over all images in a mini batch, and ensure that it is approximately one over the number of output neurons:

$$A = \sum_i \sum_m log(\frac{N}{M} q_{m,i}) \tag{9}$$

Where q is indexed by the batch elements, $m \in M$, and neuron, $n \in N$. Combining the two terms, we get our total loss function:

$$L = \sum_m H + \lambda A$$

$$L = \sum_m \left(\sum_i \frac{\beta X_{i,m} e^{-\beta X_{i,m}}}{\sum_j e^{-\beta X_{j,m}}} + \frac{\ln \sum_j e^{-\beta X_{j,m}}}{\sum_j e^{-\beta X_{j,m}}} \sum_i e^{-\beta X_{i,m}} \right) + \lambda \sum_{i,m} \ln \frac{N}{M} \frac{e^{-} \beta X_{i,m}}{\sum_j e^{-\beta X_{j,m}}} \tag{10}$$

4. Backward Function - Entropy Gradient

The gradient of the entropy forward function is the partial derivative of equation 8 with respect to an individual element, X_k. First we will take the derivative of the left term, as defined in equation 8, then we will take the derivative of the right term, then we will combine them.

4.1 Left Term Entropy Derivative

$$\frac{\partial H}{\partial X_k} = \frac{\sum_j e^{-\beta X_j} \frac{\partial \sum_i \beta X_i e^{-\beta X_i}}{\partial X_k} - \sum_i \beta X_i e^{-\beta X_i} \frac{\partial \sum_j e^{-\beta X_j}}{\partial X_k}}{\left(\sum_j e^{-\beta X_j}\right)^2} + \dots$$

$$\frac{\partial H}{\partial X_k} = \frac{\sum_j e^{-\beta X_j} \beta e^{-\beta X_k} (1 - \beta X_k) + \beta^2 e^{-\beta X_k} \sum_i X_i e^{-\beta X_i}}{\left(\sum_j e^{-\beta X_j}\right)^2} + \dots$$

$$\frac{\partial H}{\partial X_k} = \frac{\beta e^{-\beta X_k} (1 - \beta X_k)}{\sum_j e^{-\beta X_j}} + \frac{\beta^2 e^{-\beta X_k} \sum_i X_i e^{-\beta X_i}}{\left(\sum_j e^{-\beta X_j}\right)^2} + \dots$$

$$\frac{\partial H}{\partial X_k} = \frac{\beta e^{-\beta X_k}}{\sum_j e^{-\beta X_j}} \left(1 - \beta X_k + \frac{\beta \sum_i X_i e^{-\beta X_i}}{\sum_j e^{-\beta X_j}}\right) + \dots$$

$$\frac{\partial H}{\partial X_k} = \beta q_k \left(1 - \beta X_k + \beta \sum_i X_i q_i\right) + \dots$$

$$\frac{\partial H}{\partial X_k} = \beta q_k - \beta^2 X_k q_k + \beta^2 q_k \sum_i X_i q_i + \dots$$

4.2 Right Term Entropy Derivative

$$\frac{\partial H}{\partial X_k} = \dots + \sum_i e^{-\beta x_i} \left(\frac{\sum_j e^{-\beta X_j} \left(\frac{-\beta e^{-\beta X_k}}{\sum_j e^{-\beta X_j}}\right) + \beta e^{-\beta X_k} \ln \sum_j e^{-\beta X_j}}{\left(\sum_j e^{-\beta X_j}\right)^2} \right) + \frac{\ln \sum_j e^{-\beta X_j}}{\sum_j e^{-\beta X_j}} \left(-\beta e^{-\beta X_k}\right)$$

$$\frac{\partial H}{\partial X_k} = \dots + \frac{-\beta e^{-\beta X_k}}{\sum_j e^{-\beta X_j}} + \frac{\beta e^{-\beta X_k} \ln \sum_j e^{-\beta X_j}}{\sum_j e^{-\beta X_j}} - \frac{\beta e^{-\beta X_k} \ln \sum_j e^{-\beta X_j}}{\sum_j e^{-\beta X_j}}$$

$$\frac{\partial H}{\partial X_k} = \dots - \beta q_k \tag{11}$$

4.3 Combine Left and Right Entropy Derivatives

$$\frac{\partial H}{\partial X_k} = \beta q_k - \beta^2 X_k q_k + \beta^2 q_k \sum_i X_i q_i - \beta q_k$$

$$\frac{\partial H}{\partial X_k} = -\beta^2 X_k q_k + \beta^2 q_k \sum_i X_i q_i$$

$$\frac{\partial H}{\partial X_k} = \beta^2 q_k \left(\sum_i X_i q_i - X_k\right) \tag{12}$$

where q_i is given in equation 6.

4.4 Average Activation Function Derivative

In addition to the gradient on the forward entropy equation (equation 8), we need to compute a gradient on the second term that encourages equal activation across all nodes 9.

$$\begin{aligned}\frac{\partial A}{\partial X_k} &= \lambda \frac{\frac{\partial \frac{N}{M} \sum_m q_{k,m}}{\partial X_k}}{\frac{N}{M} \sum_m q_{k,m}} \\ &= \lambda \sum_m \beta \left(q_{k,m} - 1 \right)\end{aligned} \tag{13}$$

4.5 Loss Derivative

Finally, we combine the derived entropy gradient with the average activity function gradient to get our full loss gradient:

$$\frac{\partial L}{\partial X_k} = \sum_m \beta^2 q_{k,m} \left(\sum_i X_{i,m} q_{i,m} - X_{k,m} \right) + \lambda \sum_m \beta \left(q_{k,m} - 1 \right) \tag{14}$$