Deep Learning HW4

yg2709

March 2024

1 Theory

1.1 Energy Based Models Intuition

(a) Energy-based models computes the energy of all possible y, and then finds the one with lowest energy using `argmin`.

(b) The goal of energy-based models and models that output probabilities are different. For energy-based models, the purpose of training is minimize the energy of the desired one, and maximize the energy of others. While for models that output probabilities, the training goal is correctly predicting the probability distribution.

(c) We can simply turn the output of energy function into a probability distribution. $p(y|x) = softargmin_\beta(F_W(x, y))$.

(d) Energy function computes the 'energy' score of each possible y. Loss function is used to find the best performing energy function.

(e) By only pushing down energy of correct inputs, other data are neglected during training. This might lead the model to collapse, which produces the same output regardless of inputs. To avoid this, we can push up the energy of incorrect inputs.

(f) The first method is contrastive. We push down the energy of positive samples, and pull up the energy of negative samples.

The second method is regularization. We add some constrains to the model such that it can behave in a desired way.

The third method is architectural. We design the architecture of the whole model such that the overall energy is low.

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

(h) For $F(x, y)$, the inference function is $\hat{y} = argmin_y(x, y)$. For $G(x, y, z)$, the inference function is $\hat{y} = argmin_{y,z}(x, y, z)$.

1.2 Negative log-likelihood loss

(a) $P(y_i) = \frac{exp(-\beta F_W(x,y_i)}{\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))}$

(b) First we take the negative loss of P(y).

$$-log(P(y|x) = -log(\frac{exp(-\beta F_W(x,y)}{\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))}) \tag{1}$$

$$= \beta F_W(x,y) + log(\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))) \tag{2}$$

$$\tag{3}$$

Then we get the loss function

$$L(x,y,W) = \frac{1}{\beta}(\beta F_W(x,y) - \Sigma_{j=1}^n(\beta F_W(x,y_j))) \tag{4}$$

$$= F_W(x,y) + \frac{1}{\beta} log(\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))) \tag{5}$$

(c)

$$\frac{\partial L}{\partial W} = \frac{\partial F_W(x,y)}{\partial W} + \frac{\partial(\frac{1}{\beta} log(\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))))}{\partial W} \tag{6}$$

$$= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\beta}\frac{1}{\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))}\frac{\partial(\Sigma_{j=1}^n exp(-\beta F_W(x,y_j)))}{\partial W} \tag{7}$$

$$= \frac{\partial F_W(x,y)}{\partial W} + \frac{1}{\beta}\frac{-\beta}{\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))}\Sigma_{j=1}^n exp(-\beta F_W(x,y_j))\frac{\partial F_W(x,y_j}{\partial W} \tag{8}$$

$$= \frac{\partial F_W(x,y)}{\partial W} - \Sigma_{j=1}^n P(y_j|x)\frac{\partial F_W(x,y)}{\partial W} \tag{9}$$

This becomes intractable if y is continuous. The gradient will be $\frac{\partial F_W(x,y)}{\partial W} - \int_{y'} P(y'|x)\frac{\partial F_W(x,y)}{\partial W}$. In this case, the integral is difficult to compute when y' is a high dimensional vector. To get around, we can sample points from y' to make it discrete.

(d) For the force that pushes the energy, it's not proportional to the correct label y. but proportional to each of the y'. This means that even two close examples could be pushed to very different directions.

1.3 Comparing Contrastive Loss Functions

(a) Simple loss function

$$\frac{\partial l_{simple}(x, y, \bar{y}, W)}{\partial W} = \frac{\partial [F_W(x, y)]^+}{\partial W} + \frac{\partial [m - F_W(x, \bar{y})]^+}{\partial W} \tag{10}$$

$$\frac{\partial [F_W(x, y)]^+}{\partial W} = \begin{cases} 0 & \text{if } F_W(x, y) < 0 \\ \frac{\partial F_W(x,y)}{\partial W} & \text{in other cases} \end{cases} \tag{11}$$

$$\frac{\partial [m - F_W(x, \bar{y})]^+}{\partial W} = \begin{cases} 0 & \text{if } m - F_W(x, \bar{y}) < 0 \\ -\frac{\partial F_W(x,y)}{\partial W} & \text{in other cases} \end{cases} \tag{12}$$

(b) Log loss

$$\frac{\partial l_{log}(x, y, \bar{y}, W)}{\partial W} = \frac{exp(F_W(x, y) - F_W(x, \bar{y})}{1 + exp(F_W(x, y) - F_W(x, \bar{y})} (\frac{\partial F_W(x, y)}{\partial W} - \frac{\partial F_W(x, \bar{y})}{\partial W}) \tag{13}$$

(c) Square-Square loss

$$\frac{\partial l_{square-square}(x, y, \bar{y}, W)}{\partial W} = 2([F_W(x, y)]^+) \frac{\partial [F_W(x, y)]^+}{\partial W} + 2([m - F_W(x, \bar{y})]^+) \frac{\partial [m - F_W(x, \bar{y})]^+}{\partial W} \tag{14}$$

$$\frac{\partial [F_W(x, y)]^+}{\partial W} = \begin{cases} 0 & \text{if } F_W(x, y) < 0 \\ \frac{\partial F_W(x,y)}{\partial W} & \text{in other cases} \end{cases} \tag{15}$$

$$\frac{\partial [m - F_W(x, \bar{y})]^+}{\partial W} = \begin{cases} 0 & \text{if } m - F_W(x, \bar{y}) < 0 \\ -\frac{\partial F_W(x,y)}{\partial W} & \text{in other cases} \end{cases} \tag{16}$$

(d) (i) For the three losses above, the energy of negative examples are pushed up once at a time, while NLL loss pushes up the energy of all negative examples each time.

(ii) Both log loss and hinge loss try to keep positive and negative samples apart in a certain distance. Compared with the constant margin m in hinge loss, the 'margin' in log loss is smoother, thus can be called 'soft-hinge' loss. The advantage of log loss is that it's easier to derive.

(iii) Simple and square-square loss minimize the energy of positive examples to be close to 0. However, the hinge/loss loss only cares about the relative energy between positive and negative examples, so the energy of positive examples might not necessarily be close to 0. For simple loss and square-square loss, since the pushing force of square-square loss is quadratic, we prefer simple loss when we don't want to the model to be robust to outliers, and prefer square-square loss in other cases.