Deep Learning Assignment 3

Anonymous Author(s)
Affiliation
Address
email

1 General Questions

(a)

(b)

2 Softmax regression gradient calculation

(a) By chain rule, we have:

$$\frac{\partial l}{\partial W_{i,j}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{i,j}} \tag{1}$$

$$\frac{\partial l}{\partial \hat{y}_i} = \frac{\partial}{\partial \hat{y}_i}(-\sum\nolimits_j y_j \log \hat{y}_j) \tag{2}$$

When $i \neq j$, the elements in the loss function are constant compared to y_i. We can just consider the element that $i = j$.

$$\frac{\partial l}{\partial \hat{y}_i} = \frac{\partial}{\partial \hat{y}_i}(-y_i \log \hat{y}_i) = \frac{-y_i}{(\ln 10)\hat{y}_i} = \frac{-y_i}{\hat{y}_i} \tag{3}$$

From assignment 1, we have:

$$(X_{out})_i = \frac{exp(\beta(X_{in})_i)}{\sum_j exp(\beta(X_{in})_j)} \tag{4}$$

and also

$$\frac{\partial (X_{out})_i}{\partial (X_{in})_i} = \frac{\partial}{\partial (X_{in})_i} \frac{exp(\beta(X_{in})_i)}{\sum_j exp(\beta(X_{in})_j)} = \beta(X_{out})_i(1 - (X_{out})_i) \tag{5}$$

So

$$\frac{\partial \hat{y}_i}{\partial W_{i,j}} = \frac{\partial}{\partial W_{i,j}} \frac{exp(W_i x + b_i)}{\sum_k exp(W_k x + b_k)} = x_j \hat{y}_i(1 - \hat{y}_i) \tag{6}$$

Now we can calculate $\frac{\partial l}{\partial W_{i,j}}$

$$\frac{\partial l}{\partial W_{i,j}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{i,j}} = \frac{-y_i}{\hat{y}_i} x_j \hat{y}_i(1 - \hat{y}_i) = -y_i x_j(1 - \hat{y}_i) \tag{7}$$

Submitted to 29th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

(b) When $y_{c1} = 1$ and $\hat{y_{c2}} = 1$ and $c1 \neq c2$, the model has wrong output with confidence 100% for example:
model output = [1, 0]
true label = [0, 1]

From the loss function in part(a), we will get a very very large loss at $y_{c1} \log \hat{y_{c1}}$ and get 0 error in other positions. As for the gradient, since

$$\frac{\partial l}{\partial W_{i,j}} = -y_i x_j (1 - \hat{y}_i) \tag{8}$$

when $i = c1$, it will increase the W value that contribute to the correct label, otherwise the W value remain unchanged. With the softmax function, the $\hat{y}_i$ for the wrong label will be reduced and the correct the $\hat{y}_i$ will increase in next forward propagation.

3 Chain rule

(a) Let

$$f = \frac{a}{b} \tag{9}$$

$$a = x^2 + \sigma(y) \tag{10}$$

$$b = 3x + y - \sigma(x) \tag{11}$$

$$\frac{\partial f}{\partial x} = \frac{\frac{\partial a}{\partial x} \cdot b - \frac{\partial b}{\partial x} \cdot a}{b^2} = \frac{2x \cdot b - (3 - \frac{\partial \sigma(x)}{\partial x}) \cdot a}{b^2} \tag{12}$$

$$\frac{\partial f}{\partial y} = \frac{\frac{\partial a}{\partial y} \cdot b - \frac{\partial b}{\partial y} \cdot a}{b^2} = \frac{\frac{\partial \sigma(y)}{\partial y} \cdot b - a}{b^2} \tag{13}$$

Where

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x) \cdot (1 - \sigma(x)) \tag{14}$$

(b) For $x = 1, y = 0$,

$$\sigma(x) = \sigma(1) = 0.731 \tag{15}$$

$$\sigma(y) = \sigma(0) = 0.5 \tag{16}$$

$$a = 1 + \sigma(0) = 1.5 \tag{17}$$

$$b = 3 * 1 + 0 - \sigma(1) = 2.269 \tag{18}$$

So we can calculate derivitive of f by:

$$\frac{\partial f}{\partial x} = \frac{2 * 1 * 2.269 - (3 - 0.731 * (1 - 0.731)) * 1.5}{2.269^2} = 0.0646 \tag{19}$$

$$\frac{\partial f}{\partial y} = \frac{0.5 * (1 - 0.5) * 2.269 - 1.5}{2.269^2} = -0.181 \tag{20}$$

4 Variants of pooling

(a) SpatialMaxPooling SpatialAveragePooling SpatialAdaptivePooling

(b)

(c)

5 Convolution

(a) Assume we use zero padding and step size = 1, then: $(5-3+1)*(5-3+1) = 9$ values will be generated.

(b) Let X be a 3x3 matrix on Image Matrix and W is the 3x3 convolution filter.
According to the definition of convolution, each element in the output is the point product of these two matrix.

$$F = \sum W \cdot *X \tag{21}$$

For example: $F_1 1 = 4*4+3*5+3*2+5*3+5*3+5*2+2*4+4*3+3*4 = 109$

So we have the output F = $\begin{pmatrix} 109 & 92 & 72 \\ 108 & 85 & 74 \\ 110 & 74 & 79 \end{pmatrix}$

(c) Let $\frac{\partial E}{\partial X^(i-1)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

The definition of F = $\begin{pmatrix} \sum\limits_{i=1}^{3}\sum\limits_{j=1}^{3} X_i j W_i j & \sum\limits_{i=2}^{4}\sum\limits_{j=1}^{3} X_i j W_i j & \sum\limits_{i=3}^{5}\sum\limits_{j=1}^{3} X_i j W_i j \\ \sum\limits_{i=1}^{3}\sum\limits_{j=2}^{4} X_i j W_i j & \sum\limits_{i=2}^{4}\sum\limits_{j=2}^{4} X_i j W_i j & \sum\limits_{i=3}^{5}\sum\limits_{j=2}^{4} X_i j W_i j \\ \sum\limits_{i=1}^{3}\sum\limits_{j=3}^{5} X_i j W_i j & \sum\limits_{i=2}^{4}\sum\limits_{j=3}^{5} X_i j W_i j & \sum\limits_{i=3}^{5}\sum\limits_{j=3}^{5} X_i j W_i j \end{pmatrix}$

$$\frac{\partial E}{\partial X_i j} = \frac{\partial}{\partial X_i j} ... = \sum W_i j \frac{\partial E}{\partial X^(i-1)} \tag{22}$$

result = $\begin{pmatrix} 4 & 7 & 10 & 6 & 3 \\ 9 & 17 & 25 & 16 & 8 \\ 11 & 23 & 34 & 23 & 11 \\ 7 & 16 & 24 & 17 & 8 \\ 2 & 6 & 9 & 7 & 3 \end{pmatrix}$

6 Optimization

(a)

(b)

(c)

(d)

7 Top-k error

Top-k error: the fraction of test images for which the correct label is not among the k labels considered most probable by the model.

$$E = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{R_i > k} \tag{23}$$

Where function R counts the number of probability for prediction $\hat{y}_c$ that's larger than the probability of true label $\hat{y}_L$ for each test image.

$$R = \sum\nolimits_{c \in C} \mathbb{1}_{\hat{y}_c > \hat{y}_L} \tag{24}$$

The top-1 errors represents the error rate that the prediction is not the same with true label. This is useful to compare the performance of different models. However, it cannot show how how good the model is in general. Sometimes the categories in a image may be ambiguous and it may be described as multiple labels. The order of several possible labels is ambiguous. Using top-5 errors provides a more general understanding of learning ability of the model.

8 t-SNE

(a) The crowding problem happens when we try to project high dimentional dataset into lower dimentional space. The distance between records in lower dimentional space will be much smaller than in high dimentional space. In some clustering algorithms, the records will form a large group in the center and fail to produce nature clustering. t-SNE use heavy-tailed distribution to reduce this problem. It convert distances into probabilities using a Guassian distribution in high-dimentional space. Then use a much heavier tails probability distribution in the low-dimentional map. This allows a moderate distance in the high-dimentional space to be modeled by a much larger distance in low dimention.

(b)

$$\frac{\partial C}{\partial y_i} = \frac{\partial}{\partial X_i j} ... = \sum W_i j \frac{\partial E}{\partial X^(i-1)} \tag{25}$$

9 Proximal gradient decent

(a)

(b)

(c)

(d)