Deep Learning

2022-02-11 Creation

1 方程式推導

1.1 Loss Functions

$\hat{\mathbf{y}}$ is prediction, $\mathbf{y}$ is ground true, **epsilon** $= 10 \times e^{-5}$

1.1.1 MSE

$$\mathrm{MSE}(\hat{y},\ y) = \frac{1}{n}(\hat{y} - y)^2, \qquad \text{n is size of y and prediction } \hat{y}$$

$$\frac{\partial \mathrm{MSE}}{\partial \hat{y}} = \frac{2}{n}(\hat{y} - y)$$

1.1.2 Cross Entropy with Softmax

$$\mathrm{Softmax}(\hat{y}) = \frac{e^x}{\sum e^x}$$

$$\mathrm{CossEntropy}(\hat{y}, y) = -\sum y \log_e(\hat{y} + \mathrm{epsilon})$$

$$\mathrm{CossEntropyWithSoftmax}(\mathrm{Softmax}(\hat{y}), y) = \mathrm{CossEntropy}(\mathrm{Softmax}(\hat{y}), y)$$

$$\frac{\partial \mathrm{CossEntropyWithSoftmax}}{\partial \hat{y}} = \hat{y} - y$$

You can see the proof in https://deepnotes.io/softmax-crossentropy

1.2 $\frac{x}{sum(x)}$ gradient

$$f(x) = \frac{x}{\mathrm{sum}(x)}$$

$$\begin{cases} \dfrac{\partial f_i}{\partial x_j} = \dfrac{\mathrm{sum}(x) \cdot 1 - x_i \cdot 1}{\mathrm{sum}(x)^2} = \dfrac{\mathrm{sum}(x) - x_i}{\mathrm{sum}(x)^2}, & i = j \\ \dfrac{\partial f_i}{\partial x_j} = -\dfrac{x_i}{\mathrm{sum}(x)^2}, & i \neq j \end{cases}$$

$$\frac{\partial L}{\partial f}\frac{\partial f}{\partial x_i} = \frac{\partial L}{\partial f_i}\frac{\partial f_i}{\partial x_i} + \sum_{k\neq i}\frac{\partial L}{\partial f_k}\frac{\partial f_k}{\partial x_i} = \frac{\partial L}{\partial f_i}\frac{\mathrm{sum}(x) - x_i}{\mathrm{sum}(x)^2} + \sum_{k\neq i}\frac{\partial L}{\partial f_k}\frac{-x_k}{\mathrm{sum}(x)^2}$$
$$= \frac{\partial L}{\partial f_i}\frac{\mathrm{sum}(x) - x_i}{\mathrm{sum}(x)^2} + \sum_{k}\frac{\partial L}{\partial f_k}\frac{-x_k}{\mathrm{sum}(x)^2} - \frac{\partial L}{\partial f_i}\frac{-x_i}{\mathrm{sum}(x)^2}$$
$$= \frac{\partial L}{\partial f_i}\frac{\mathrm{sum}(x) - x_i + x_i}{\mathrm{sum}(x)^2} + \sum_{k}\frac{\partial L}{\partial f_k}\frac{-x_k}{\mathrm{sum}(x)^2}$$
$$= \frac{\partial L}{\partial f_i}\frac{1}{\mathrm{sum}(x)} + \sum_{k}\frac{\partial L}{\partial f_k}\frac{-x_k}{\mathrm{sum}(x)^2}$$

1.3 Proof for Matrix's Gradient

Example 1

b is batch, ***n*** is input features, ***m*** is output features, $\boldsymbol{\partial Y}$ is a matrix all contains ***1***.

$$X_{b,n}W_{n,m} = Y_{b,m}$$
$$\left(\frac{\partial Y}{\partial W}\right)_{n,m} = (X^T)_{n,b}(\partial Y)_{b,m}$$
$$\left(\frac{\partial Y}{\partial X}\right)_{b,n} = (\partial Y)_{b,m}(W^T)_{m,n}$$

Gradient calculation

$Y_{b,m} = \sum_{k=1}^{n} X_{b,k}W_{k,m}$, for a value $W_{k,i}$, multiplied by values in vector $X_{b,k}$. for a value $X_{i,k}$, multiplied by values in vector $W_{k,m}$.

$\partial W_{n,m} = \sum_{p=1}^{b} X_{n,p}^T\, \partial Y_{p,m}$, gradient is a summation of $X_{b,k} \odot \partial Y_{b,k}$ for a value $w_{i,j}$.

$\partial X_{b,n} = \sum_{p=1}^{m} \partial Y_{b,p} W_{p,n}^T$, gradient is a summation of $W_{k,m} \odot \partial Y_{k,m}$ for a value $X_{i,j}$.

Example 2

$X_{b,n}W_{n,m} = Y_{b,m}$, $f(Y_{b,m}) = {Y_{b,m}}^2$

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} x_{11}w_{11} + x_{12}w_{21} & x_{11}w_{12} + x_{12}w_{22} \\ x_{21}w_{11} + x_{22}w_{21} & x_{21}w_{12} + x_{22}w_{22} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}$$

$$f\left(\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}\right) = \begin{bmatrix} {y_{11}}^2 & {y_{12}}^2 \\ {y_{21}}^2 & {y_{22}}^2 \end{bmatrix},\ f'\left(\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}\right) = \begin{bmatrix} 2y_{11} & 2y_{12} \\ 2y_{21} & 2y_{22} \end{bmatrix}$$

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} \\ \frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial y_{11}}\frac{\partial y_{11}}{\partial x_{11}} + \frac{\partial f}{\partial y_{12}}\frac{\partial y_{12}}{\partial x_{11}} & \frac{\partial f}{\partial y_{11}}\frac{\partial y_{11}}{\partial x_{12}} + \frac{\partial f}{\partial y_{12}}\frac{\partial y_{12}}{\partial x_{12}} \\ \frac{\partial f}{\partial y_{21}}\frac{\partial y_{21}}{\partial x_{21}} + \frac{\partial f}{\partial y_{22}}\frac{\partial y_{22}}{\partial x_{21}} & \frac{\partial f}{\partial y_{21}}\frac{\partial y_{21}}{\partial x_{22}} + \frac{\partial f}{\partial y_{22}}\frac{\partial y_{22}}{\partial x_{22}} \end{bmatrix}$$
$$= \begin{bmatrix} (2y_{11})w_{11} + (2y_{12})w_{12} & (2y_{11})w_{21} + (2y_{12})w_{22} \\ (2y_{21})w_{11} + (2y_{22})w_{12} & (2y_{21})w_{21} + (2y_{22})w_{22} \end{bmatrix}$$
$$= (2_{b,m} \odot Y_{b,m})W_{m,n}^T$$

Theorem 1

$AB = C, \qquad f(C) = Y$

$$\frac{\partial f}{\partial A} = \frac{\partial f}{\partial C}\frac{\partial C}{\partial A} = \frac{\partial f}{\partial C}B^T, \qquad \frac{\partial f}{\partial B} = \frac{\partial f}{\partial C}\frac{\partial C}{\partial B} = A^T\frac{\partial f}{\partial C}$$

Theorem 2

$ABC = D, \qquad f(D) = Y$

$$\frac{\partial f}{\partial A} = \frac{\partial f}{\partial D}\frac{\partial D}{\partial A} = \frac{\partial f}{\partial D}(BC)^T = \frac{\partial f}{\partial D}C^TB^T$$

$$\frac{\partial f}{\partial B} = \frac{\partial f}{\partial D}\frac{\partial D}{\partial B} = A^T\frac{\partial f}{\partial D}C^T$$

$$\frac{\partial f}{\partial C} = \frac{\partial f}{\partial D}\frac{\partial D}{\partial C} = (AB)^T\frac{\partial f}{\partial D} = B^TA^T\frac{\partial f}{\partial D}$$

Theorem 3

$X_1X_2X_3 \ldots X_n = Y_1, \qquad Y_2 = f(Y_1)$

$$\frac{\partial Y_2}{\partial X_i} = (X_1X_2 \ldots X_{i-1})^T\frac{\partial Y_2}{\partial Y_1}(X_{i+1}X_{i+2} \ldots X_n)^T$$

2 經驗

- Batch Normalization 影響輸出是否為爆炸型，並影響訓練的收斂速度
- 深層網路收斂與訓練速度較淺層網路快速，記憶體用量也更大，但表現力更強
- Pytorch 僅更改 Module Class 的名字後，仍然能夠使用