Lecture 3

Cross Entropy

$L = -\sum y_j \log S_j$

S_j — estimation

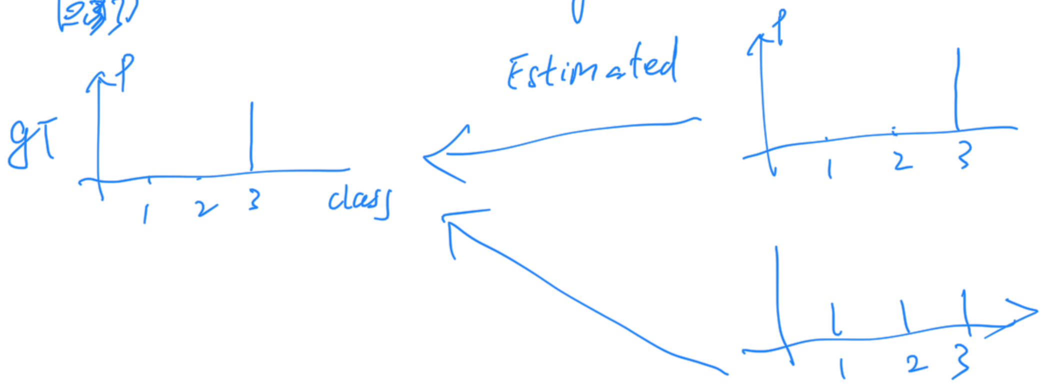
y_j — growth truth

$$\begin{array}{cc} GT & S\text{-outp} \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0.002 \\ 0.02 \\ 0.978 \end{bmatrix} \end{array}$$

$$= -(0 + 0 + \log 0.978) = 0.02$$

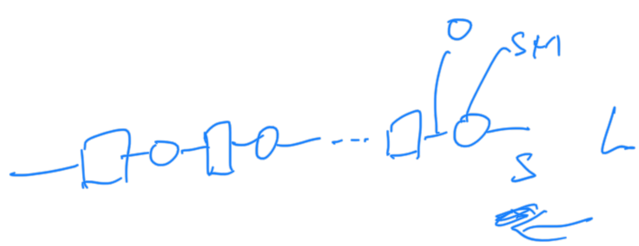
S-out

$$\begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix} \quad L = -(0 + 0 + \log 0.33) = 1.1$$



$$\underline{\sum L_i(S_i, Y_i)}$$

$$\sum (S - y)^2$$



$$\frac{\partial L}{\partial O_i} = \frac{\partial L}{\partial S} \frac{\partial S}{\partial O_i}$$

$$L = -\sum_j y_j \log S_j$$

cross entropy

softmax

$$S_j = \frac{e^{O_j}}{\sum_{i=1}^{N} e^{O_i}}$$

cross entropy

$$\frac{\partial s_j}{\partial o_j} = s_j(1-s$$

$$= -\left(\sum y_j \frac{\partial \log s_j}{\partial s_j} \frac{\partial s_j}{\partial o_i}\right)$$

$$\frac{\partial \log(x)}{\partial x} = \frac{1}{x}$$

$$= -\sum\left(y_j \frac{1}{s_j} \frac{\partial s_j}{\partial o_i}\right)$$

$$\frac{\partial \sum f(x_i)}{\partial x_i} = \sum \frac{\partial f(x_i)}{\partial x_i}$$

$i \neq j$

$i = j$

$$= -y_i \frac{s_i(1-s_i)}{s_i} - \sum_{i \neq j} y_j \frac{1}{s_j} \underbrace{\frac{\partial s_j}{\partial o_i}}_{-s_i s_j}$$

$$= -y_i(1-s_i) - \sum_{i \neq j} y_j(-s_i)$$

=

$$\frac{\partial s_j}{\partial o_i} \text{ where } i \neq j \qquad \sum e^{o_k} = c + e^{o_i}$$

$$= \frac{\partial \frac{e^{o_j}}{\sum e^{o_k}}}{\partial o_i} = \frac{\partial \frac{e^{o_j}}{c + e^{o_i}}}{\partial o_i}$$

$$= e^{o_j} \frac{\partial \frac{1}{c + e^{o_i}}}{\partial o_i}$$

$$= -e^{o_j} e^{o_i} / (c + e^{o_i})^2$$

$$C + e^{o_i} = \sum_k e^{o_k}$$

$$= -\frac{e^{o_j}}{C + e^{o_i}} \cdot \frac{e^{o_i}}{C + e^{o_i}}$$

$$= -\underbrace{\frac{e^{o_j}}{\sum_k e^{o_k}}}_{= S_j} \; \underbrace{\frac{e^{o_i}}{\sum_k e^{o_k}}}_{= S_i}$$

$$= -S_j S_i$$

$$= -y_i(1 - S_i) + \sum_{i \neq j} y_j S_i$$

$$= -y_i + y_i S_i + \sum_{i \neq j} y_j S_i$$

$$= -y_i + \sum_j y_j S_i$$

$$= -y_i + S_i \underline{\sum y_j}$$

$$= S_i - y_i$$

Training

- preprocessing
- overfitting, Generalization
- Large data set

input X_i D-dimension

$i = 1 - N$, N samples

$$\mu_j = \frac{1}{N} \sum_{i=1}^{N} X_{ij} \quad \leftarrow \text{mean}$$

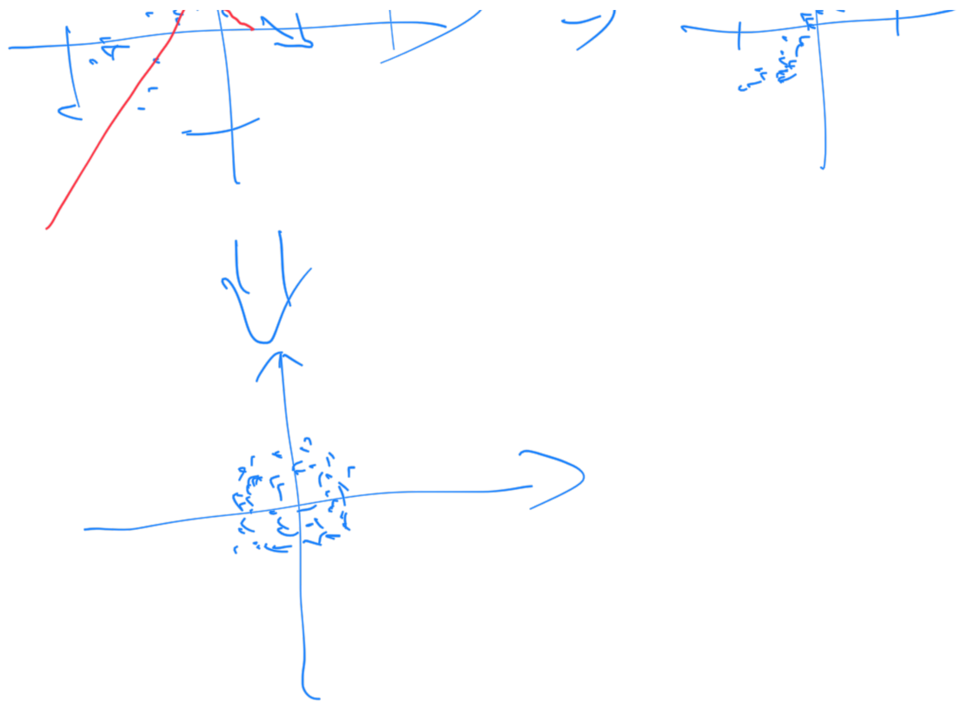
$j = 1 \to D$

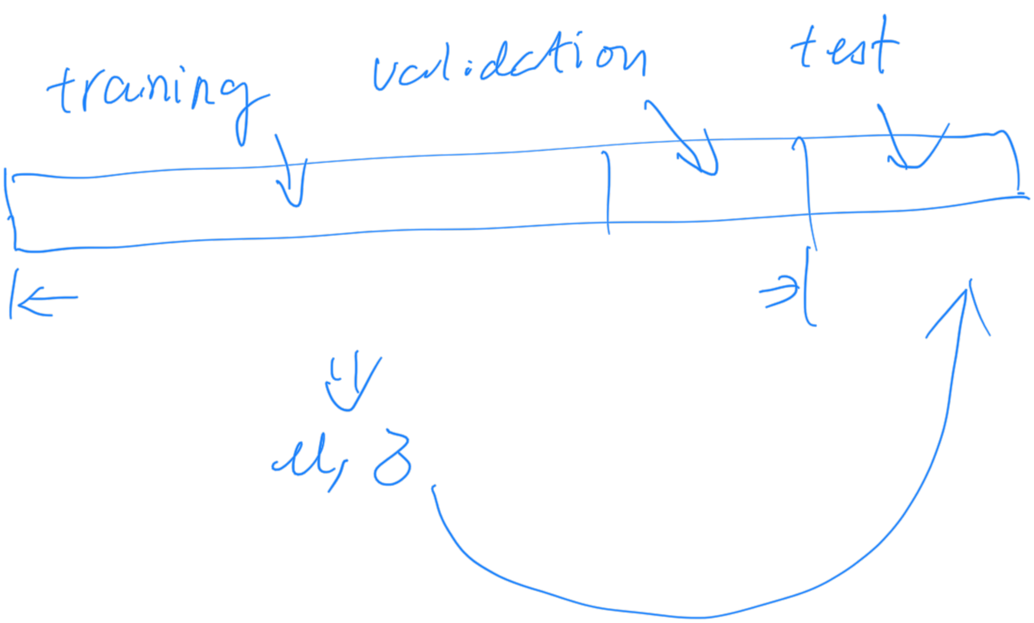
$$\sigma_j = \frac{1}{N} \sum_{i=1}^{N} (X_{i,j} - \mu_j)^2 \quad \leftarrow \text{variance}$$

$j = 1 \to D$

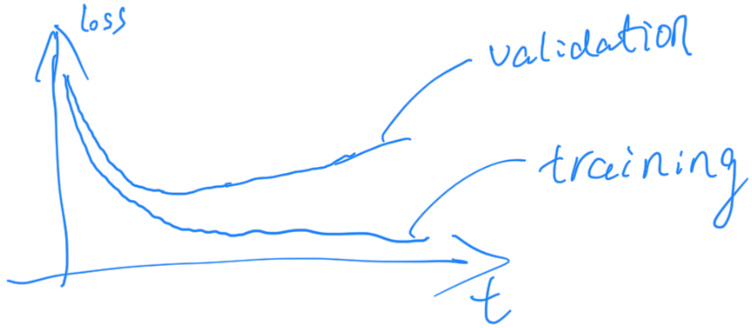
$$\hat{X}_{ij} = \frac{X_{ij} - \mu_j}{\sqrt{\sigma_j + \epsilon}}$$

small #





Early Stopping



$$w_{t+1} = w_t - \lambda \frac{\partial L}{\partial w}$$

keep a copy when validation reach a new low

Regularization

$w^T w$ to be small

$$\sum w_i^2 = w^T w$$

$$L = L + \gamma L(w)$$

L $w^T w$ or $\sum w_i^2$

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w} + r \frac{\partial L(w)}{\partial w}$$

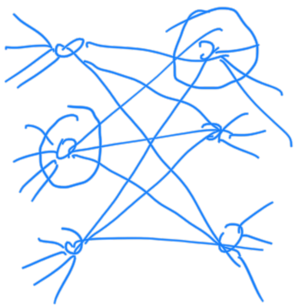
$$\frac{\partial L(w)}{\partial w_j} = \frac{\sum w_i^2}{\partial w_j} = 2 w_j$$

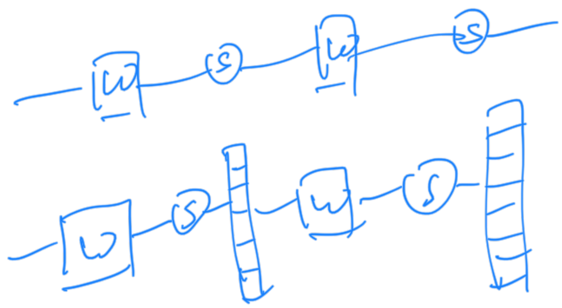
$$w_j^{t+1} = w_j^t - \lambda \left(\frac{\partial L}{\partial w_j} + r 2 w_j^t \right)$$

$$= w_j^t - \lambda \frac{\partial L}{\partial w_j} - 2\lambda\gamma w_j^t$$

$$= w_j^t(1-2\lambda\gamma) - \lambda \frac{\partial L}{\partial w_j}$$

Drop out





$p \begin{cases} 0.5 & \text{hidden layers} \\ 0.8 & \text{input layer} \end{cases}$

Bagging / Ensemble

short for bootstrap aggregation

- Different dataset

  {common data, different data}
- Different initializations
- Different batches

Stochastic Gradient Descent (SGD)

Loop

sample a batch of data from whole

Use the batch to compute feed forward

– – – – – – backpropagation to get

Δw

update $w_{t+1} = w_t - \lambda \Delta w$

Data size = 100,000 , batch size 1,000

- randomize the whole data set in terms of order
- segment it based on batch size
  - run SGD on batch i
  - till the end of all batches

epoch