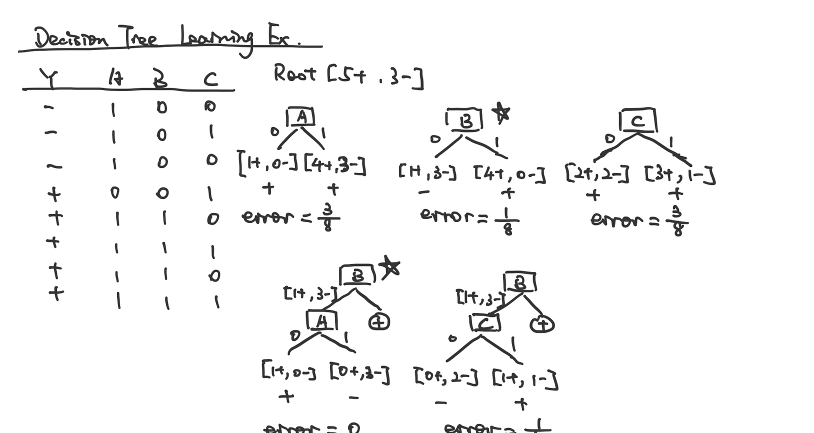
Midterm Cheatsheet

Wednesday, October 10, 2018 4:29 PM

Decision Tree Learning Ex.



Ex. Loss Function

Regression $\ell(\hat{y}, y) = (y-\hat{y})^2$

Binary classification $\ell(\hat{y}, y) = 0$ if $\hat{y}=y$, 1 else

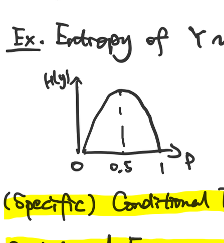
Information Theory

Let X be a random variable, where $x \in \mathcal{X}$

Let Y be a random variable, where $y \in \mathcal{Y}$

(Marginal) Entropy $H(Y) = -\sum_{y\in\mathcal{Y}} P(Y=y)\log_2 P(Y=y)$

Ex. Entropy of $Y \sim \text{Bernoulli}(p)$



(Specific) Conditional Entropy: $H(Y|X=x) = -\sum_{y\in\mathcal{Y}} P(Y=y|X=x)\log_2 P(Y=y|X=x)$

Conditional Entropy: $H(Y|X) = \sum_{x\in\mathcal{X}} P(X=x) H(Y|X=x)$

Mutual Information: $I(Y;X) = H(Y) - H(Y|X)$

(Information Gain)

ID3 Algorithm - greedy search, maximize mutual information at each split

Inductive bias of ID3 - search for the smallest tree that is consistent with the training data

Occam's Razor: prefer the simplest hypothesis that explains the data

Inductive bias of kNN

- Similar (nearby) points should have similar labels
- All dimensions are created equally!

(Online) Perceptron Algorithm

Initialize parameters: $\vec{\theta} = [\theta_1, \dots, \theta_M]^T = \vec{0}$, $b = 0$

While not converged: for $t = 1, 2, 3, \dots$

① Receive next example $(\vec{x}^{(t)}, y^{(t)})$

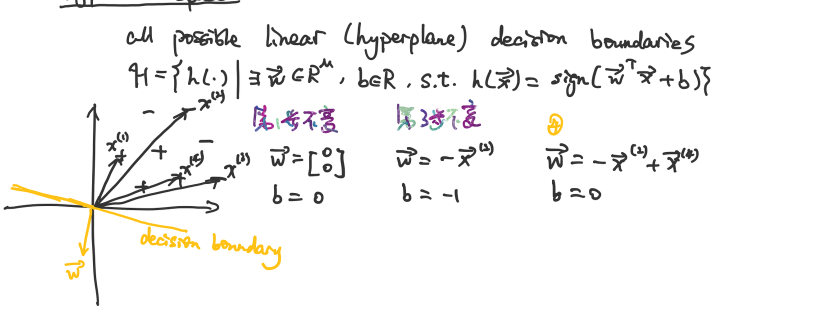
② Predict $\hat{y} = \text{sign}(\vec{\theta}^T\vec{x}^{(t)} + b)$ where $\text{sign}(a) = \begin{cases} 1 & \text{if } a \ge 0 \\ -1 & \text{otherwise}\end{cases}$

③ If positive mistake ($\hat{y} \ne y^{(t)}$, $y^{(t)} = 1$): $\vec{\theta} \leftarrow \vec{\theta} + \vec{x}^{(t)}$, $b \leftarrow b + 1$

If negative mistake ($\hat{y} \ne y^{(t)}$, $y^{(t)} = -1$): $\vec{\theta} \leftarrow \vec{\theta} - \vec{x}^{(t)}$, $b \leftarrow b - 1$

$h(\vec{x}) = \text{sign}(\vec{\theta}^T\vec{x} + b)$

Hypothesis Space



Perceptron Mistake Bound

Guarantee: If data has margin γ and all points inside a ball of radius R, then Perceptron makes $\le (R/\gamma)^2$ mistakes.

Linear Regression as Function Approximation

① Assume D is generated:

$x^{(i)} \sim p^*(\cdot)$

$y^{(i)} = c^*(x^{(i)})$

② Choose Hypothesis Space H

$H = \{h_\theta \mid h_\theta(\vec{x}) = \vec{\theta}^T\vec{x}, \vec{\theta} \in \mathbb{R}^M\}$

③ Choose an Objective Function

Goal: minimize the mean squared error

$J_D(\theta) = \frac{1}{N}\sum_{i=1}^N e_i^2 = \frac{1}{N}\sum_{i=1}^N (y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})^2$

④ Solve optimization problem

$\hat{\theta} = \underset{\theta}{\text{argmin}}\, J_D(\theta)$

⑤ Predict: given new $\vec{x}$

$y = h_{\hat{\theta}}(\vec{x}) = \hat{\theta}^T\vec{x}$

Closed-form Solution to Linear Regression

$J(\theta) = \frac{1}{2}\sum_{i=1}^N (y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})^2 = \frac{1}{2}(X\vec{\theta} - \vec{y})^T(X\vec{\theta} - \vec{y})$

$\nabla J(\theta) = X^TX\vec{\theta} - X^T\vec{y} = 0$

Solution = the Normal Equation

$\Rightarrow \underset{\theta}{\text{argmin}}\, J(\theta) = \hat{\theta} = (X^TX)^{-1}X^T\vec{y}$

Gradient Descent Algorithm

Goal: $\vec{\theta}^* = \underset{\vec{\theta}}{\text{argmin}}\, J(\vec{\theta})$, $\vec{\theta} \in \mathbb{R}^M$

① Choose an initial point $\vec{\theta}$

② Repeat

a) Compute the gradient $\nabla J(\vec{\theta})$

b) Choose a step size γ

c) update $\vec{\theta} \leftarrow \vec{\theta} - \gamma\nabla J(\vec{\theta})$

③ Return $\vec{\theta}$ when stopping criterion satisfied

Algorithm 1 Gradient Descent

1: **procedure** GD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
3: **while** not converged **do**
4: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
5: **return** $\boldsymbol{\theta}$

$J(\theta) = \frac{1}{N}\sum_{i=1}^N \frac{1}{2}(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})^2$

$= \frac{1}{N}\sum_{i=1}^N J_i(\theta)$ $\quad y^{(i)} - \sum_{j=1}^M \theta_j x_j$

$\frac{\partial J_i(\theta)}{\partial \theta_m} = (y^{(i)} - \theta^T\vec{x}^{(i)})\frac{\partial}{\partial\theta_m}(y^{(i)} - \theta^T\vec{x}^{(i)})$

$= -(y^{(i)} - \theta^T\vec{x}^{(i)}) \cdot x_m^{(i)}$

$\nabla J_i(\theta) = \begin{bmatrix} \frac{\partial J_i(\theta)}{\partial\theta_1} \\ \vdots \\ \frac{\partial J_i(\theta)}{\partial\theta_m}\end{bmatrix} = \underbrace{-(y^{(i)} - \theta^T\vec{x}^{(i)})}_{\text{scalar}}\underbrace{\vec{x}^{(i)}}_{\text{vector}}$

$\nabla J(\theta) = \frac{1}{N}\sum_{i=1}^N J_i(\theta) = \frac{1}{N}\sum_{i=1}^N -(y^{(i)} - \theta^T\vec{x}^{(i)})\vec{x}^{(i)}$

Algorithm 2 Stochastic Gradient Descent (SGD)

1: **procedure** SGD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
3: **while** not converged **do**
4: **for** $i \in \text{shuffle}(\{1, 2, \dots, N\})$ **do**
5: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda\nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
6: **return** $\boldsymbol{\theta}$

Principle of Maximum Likelihood Estimation (MLE) 最大似然估计

Choose the parameters that makes the data "most likely"

Assumption $\mathcal{D} = \{\vec{v}^{(i)}\}_{i=1}^N$ → independent and identically distributed

Data generated iid from distribution $p^*(\vec{v}|\vec{\theta}^*)$

$\vec{v} \sim p^*(\cdot|\vec{\theta}^*)$

and p^* comes from a family of distributs parameterized by $\theta^* \in \Theta$

Formally $\hat{\theta} = \underset{\theta\in\Theta}{\text{argmax}}\, p(D|\theta) = \underset{\theta\in\Theta}{\text{argmax}}\, \log p(D|\theta)$

likelihood, log-likelihood

$P(D|\theta) = \prod_{i=1}^N p(v^{(i)}|\theta)$ 似然函数.

①求似然函数 $L(\theta) = L(\theta; \vec{x}) = \begin{cases} \prod_{i=1}^n p(x_i;\theta) & \text{离散型} \\ \prod_{i=1}^n f(x_i;\theta) & \text{连续型}\end{cases}$

②求出 $\ln L(\theta)$ 并整理

③求似然方程 $\frac{\partial}{\partial\theta} L(\theta) = 0$.

④解上述方程得最大似然估计值 $\hat{\theta}(x_1, x_2 \dots x_n)$ 或最大似然估计量 $\hat{\theta}(X_1 \dots X_n)$

Dataset

i	y	x_1	...	x_M
1	1	0.5	...	0.7
2	0	3	...	4
3	1	2	...	1
4	1	1	...	-3

Bad Idea #1: Bernoulli Classifier

① ignore $\vec{x}$

② Model: $y \sim \text{Bernoulli}(\varphi)$

$p(y|\vec{x}) = \begin{cases}\varphi & \text{if } y=1 \\ 1-\varphi & \text{if } y=0\end{cases}$

③ ~~Likelihood~~ conditional log likelihood

$L(\varphi) = \log\prod_{i=1}^N p(y^{(i)}|\vec{x}^{(i)}) = \sum_{i=1}^N \log p(y^{(i)}|x^{(i)})$

$= \log\varphi + \log(1-\varphi) + \log(\varphi) + \log(\varphi)$

$= 3\log\varphi + \log(1-\varphi)$

④ $\hat{\varphi} = \underset{\varphi\in[0,1]}{\text{argmax}}\, L(\varphi)$

$\frac{d\, 3\log\varphi + \log(1-\varphi)}{d\varphi} = 0 \Rightarrow \varphi = \frac{3}{4}$

⑤ Bayes Classifier

$h(\vec{x}) = \underset{y\in\{0,1\}}{\text{argmax}}\, p(y|\vec{x}) = 1$ majority vote

Binary Logistic Regression

① Use $\vec{x}$

② Model: $\varphi = \sigma(\vec{\theta}^T\vec{x})$ where $\sigma(u) = \frac{1}{1+\exp(-u)}$

$y \sim \text{Bernoulli}(\varphi)$

$p(y|\vec{x}) = \begin{cases}\sigma(\vec{\theta}^T\vec{x}) & \text{if } y=1 \\ 1-\sigma(\vec{\theta}^T\vec{x}) & \text{if } y=0\end{cases}$

③ $L(\theta) = \sum_{i=1}^N \log p(y^{(i)}|\vec{x}^{(i)})$ $J_i(\theta)$ for SGD

$J(\theta) = -\frac{1}{N}L(\theta) = \frac{1}{N}\sum_{i=1}^N -\log p(y^{(i)}|\vec{x}^{(i)})$

④ $\hat{\theta} = \text{argmin}\, J(\theta) = \text{argmax}\, L(\theta)$.

☆ need derivatives/gradient

$\frac{\partial J_i(\theta)}{\partial\theta_m} = \frac{\partial \log p(y^{(i)}|\vec{x}^{(i)}, \vec{\theta})}{\partial\theta_m}$

$= \begin{cases}\frac{\partial}{\partial\theta_m} -\log\sigma(\vec{\theta}^T\vec{x}^{(i)}) & \text{if } y=1 \\ \frac{\partial}{\partial\theta_m} -\log(1-\sigma(\vec{\theta}^T\vec{x}^{(i)})) & \text{if } y=0\end{cases}$

$= \dots$

$= -(y^{(i)} - \sigma(\vec{\theta}^T x^{(i)}))\, x_m^{(i)}$ ← m^{th} feature

$\nabla J_i(\theta) = \underbrace{-(y^{(i)} - \sigma(\vec{\theta}^T x^{(i)}))}_{\text{scalar}}\underbrace{\vec{x}^{(i)}}_{\text{vector}} = \begin{bmatrix}\cdot \\ \cdot \\ \cdot \\ \cdot\end{bmatrix}$

Use Gradient Descent or SGD to optimize

Multiclass Classification

$\vec{x} \in \mathbb{R}^M$, $y \in \{1, \dots k\}$

Model $p(y|\vec{x}) = \frac{\exp\{\vec{\theta}_y^T\vec{x}\}}{\sum_{k=1}^K \exp\{\theta_k^T\vec{x}\}}$ $\quad \theta = \begin{bmatrix} - \theta_1^T - \\ - \theta_2^T - \\ \vdots \\ - \theta_K^T - \end{bmatrix}_{K\times M}$

Multinomial Lin Reg is different from Binary Logistic Regression even if $K=2$

Sampling $\vec{\phi} = [\phi_1, \phi_2 \dots \phi_K]$ where $\phi_k = p(y=k|\vec{x})$

$y \sim \text{Multinomial}(\vec{\phi}, 1)$

Conditional Log Likelihood

$L(\theta) = \log\prod_{i=1}^N p(y^{(i)}|x^{(i)})$

$J(\theta) = -L(\theta)$

$\hat{\theta} = \underset{\theta}{\text{argmin}}\, J(\theta)$

$\frac{\partial \log p(y^{(i)}|x^{(i)})}{\partial\theta_{km}} = \dots = \mathbb{I}(y=k)x_m - p(y|\vec{x})x_m$ indicator function

$\nabla_{\theta_m}\log$

L0, L1, L2 Regularization

Given objective function $J(\theta)$

Goal: find $\hat{\theta} = \underset{\theta}{\text{argmin}}\, J(\theta) + \lambda\underbrace{r(\theta)}_{\text{regulizer}}$ → keep model simple

Key: Choose $r(\theta)$ s.t. we tradeoff between fitting the data and keeping model simple

Ex. A commo choice

$r(\vec{\theta}) = \|\vec{\theta}\|_q = \left(\sum_{m=1}^M |\theta_m|^q\right)^{1/q}$

q	$r(\vec{\theta})$	Preference for...	Notes
0	$\|\vec{\theta}\|_0 = \sum_m \mathbb{I}(\theta_m \ne 0)$	zero values	L0 reg. no good computational solutions
1	$\|\vec{\theta}\|_1 = \sum_m \|\theta_m\|$	Small, preferably zero	L1 reg, subdifferentiable SGD
2	$(\|\theta\|_2)^2 = \sum_m \|\theta_m\|^2$	Small values	L2 reg. differentiable

Small = absolute value close to zero

$$\frac{\partial J(\boldsymbol{\Theta})}{\partial\boldsymbol{\theta}_k} = \frac{\partial(-\sum_{i=1}^N \log p(y^{(i)}\mid\mathbf{x}^{(i)}, \boldsymbol{\Theta}))}{\partial\boldsymbol{\theta}_k}$$

$$= \frac{\partial(-\sum_{i=1}^N \log\frac{\exp(\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)})}{\sum_{j=1}^K\exp(\boldsymbol{\theta}_j\mathbf{x}^{(i)})})}{\partial\boldsymbol{\theta}_k}$$

$$= \frac{\partial}{\partial\boldsymbol{\theta}_k}(-\sum_{i=1}^N \log\exp(\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)}) + \sum_{i=1}^N\log\sum_{j=1}^K\exp(\boldsymbol{\theta}_j\mathbf{x}^{(i)}))$$

$$= -(\frac{\partial}{\partial\boldsymbol{\theta}_k}\sum_{i=1}^N\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)} + \sum_{i=1}^N -\mathbf{x}^{(i)}\cdot\frac{\exp(\boldsymbol{\theta}_k\mathbf{x}^{(i)})}{\sum_{j=1}^K\exp(\boldsymbol{\theta}_j\mathbf{x}^{(i)})})$$

$$\mathbf{T}_{i,k} = \begin{cases} 1 & y^{(i)} = k \\ 0 & y^{(i)} \ne k\end{cases}$$

$$\frac{\partial}{\partial\boldsymbol{\theta}_k}\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)} = \begin{cases}\mathbf{x}^{(i)} & y^{(i)} = k \\ 0 & y^{(i)} \ne k\end{cases}$$

$$\frac{\partial}{\partial\boldsymbol{\theta}_k}\sum_{i=1}^N\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)} = \sum_{i=1}^N\mathbf{T}_{i,k}\cdot\mathbf{x}^{(i)}$$

$$\frac{\partial J(\boldsymbol{\Theta})}{\partial\boldsymbol{\theta}_k} = -(\frac{\partial}{\partial\boldsymbol{\theta}_k}\sum_{i=1}^N\boldsymbol{\theta}_{y^{(i)}}\mathbf{x}^{(i)} + \sum_{i=1}^N -\mathbf{x}^{(i)}\cdot\frac{\exp(\boldsymbol{\theta}_k\mathbf{x}^{(i)})}{\sum_{j=1}^K\exp(\boldsymbol{\theta}_j\mathbf{x}^{(i)})})$$

$$= -(\sum_{i=1}^N\mathbf{T}_{i,k}\cdot\mathbf{x}^{(i)} + \sum_{i=1}^N -\mathbf{x}^{(i)}\cdot\frac{\exp(\boldsymbol{\theta}_k\mathbf{x}^{(i)})}{\sum_{j=1}^K\exp(\boldsymbol{\theta}_j\mathbf{x}^{(i)})})$$

$$= -\sum_{i=1}^N\left(\mathbf{T}_{i,k} - p\left(C_k\mid\mathbf{x}^{(i)}, \boldsymbol{\Theta}\right)\right)\mathbf{x}^{(i)}$$

$L(\Theta|T, X) = \prod_{i=1}^N p(y^{(i)}|x^{(i)}, \Theta)$

$= \prod_{i=1}^N\sum_{j=1}^K T_{i,j}\, p(C_j|x^{(i)}, \Theta)$

$J(\Theta) = -\sum_{i=1}^N\log\sum_{j=1}^K T_{i,j}\, p(C_j|x^{(i)}, \Theta)$

$$a_j = \alpha_0 + \sum_{i=1}^6 \alpha_{j,i} * x_i,\ \forall j \in \{1, \dots, 4\}$$

Sigmoid

$$z_j = \sigma(a_j) = \frac{1}{1+\exp(-a_j)},\ \forall j \in \{1, \dots, 4\}$$

$$b_k = \beta_0 + \sum_{j=1}^4 \beta_{k,j} * z_j,\ \forall k \in \{1, \dots, 3\}$$

softmax

$$\hat{y}_k = \frac{\exp(b_k)}{\sum_{l=1}^3\exp(b_l)},\ \forall k \in \{1, \dots, 3\}$$

cross entropy

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{i=1}^3 y_i\log(\hat{y}_i)$$

Algorithm 3 Forward Computation

1: **procedure** NNFORWARD(Training example $(\mathbf{x}, \mathbf{y})$, Parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$)
2: $\mathbf{a} = \boldsymbol{\alpha}\mathbf{x}$
3: $\mathbf{z} = \sigma(\mathbf{a})$
4: $\mathbf{b} = \boldsymbol{\beta}\mathbf{z}$
5: $\hat{\mathbf{y}} = \text{softmax}(\mathbf{b})$
6: $J = -\mathbf{y}^T\log\hat{\mathbf{y}}$
7: $\mathbf{o} = \texttt{object}(\mathbf{x}, \mathbf{a}, \mathbf{z}, \mathbf{b}, \hat{\mathbf{y}}, J)$
8: **return** intermediate quantities $\mathbf{o}$

Algorithm 4 Backpropagation

1: **procedure** NNBACKWARD(Training example $(\mathbf{x}, \mathbf{y})$, Parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, Intermediates $\mathbf{o}$)
2: Place intermediate quantities $\mathbf{x}, \mathbf{a}, \mathbf{z}, \mathbf{b}, \hat{\mathbf{y}}, J$ in $\mathbf{o}$ in scope
3: $\mathbf{g}_{\hat{\mathbf{y}}} = -\mathbf{y} \div \hat{\mathbf{y}}$
4: $\mathbf{g}_{\mathbf{b}} = \mathbf{g}_{\hat{\mathbf{y}}}^T(\text{diag}(\hat{\mathbf{y}}) - \hat{\mathbf{y}}\hat{\mathbf{y}}^T)$ $\quad g_b = \hat{y} - y$
5: $\mathbf{g}_{\boldsymbol{\beta}} = \mathbf{g}_{\mathbf{b}}\mathbf{z}^T$
6: $\mathbf{g}_{\mathbf{z}} = \boldsymbol{\beta}^T\mathbf{g}_{\mathbf{b}}$
7: $\mathbf{g}_{\mathbf{a}} = \mathbf{g}_{\mathbf{z}} \odot \mathbf{z} \odot (1-\mathbf{z})$
8: $\mathbf{g}_{\boldsymbol{\alpha}} = \mathbf{g}_{\mathbf{a}}\mathbf{x}^T$
9: **return** parameter gradients $\mathbf{g}_{\boldsymbol{\alpha}}, \mathbf{g}_{\boldsymbol{\beta}}$

$y = f(x, z) = \underbrace{\exp(xz)}_{d} + \underbrace{\frac{xz}{\log(x)}}_{e} + \underbrace{\frac{\sin(\log(x))}{xz}}_{f}$

Forward

Given $x=2$, $z=3$

① $a = xz$

② $b = \log(x)$

③ $c = \sin(b)$

④ $d = \exp(a)$

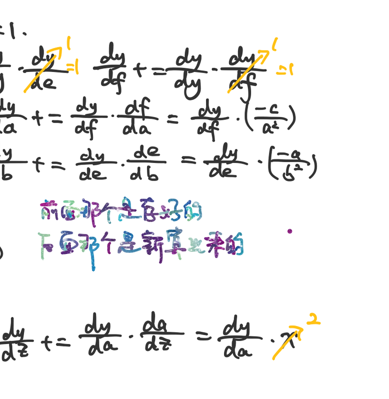
⑤ $e = a/b$

⑥ $f = c/a$

⑦ $y = d+e+f$

反向传播.

Computation Graph



$\frac{dy}{dd} = 1 \quad \frac{dy}{de} = 1 \quad \frac{dy}{df} = 1$

$\frac{dy}{da} = \frac{dy}{dd}\cdot\frac{dd}{da} + \frac{dy}{de}\cdot\frac{de}{da} + \frac{dy}{df}\cdot\frac{df}{da}$

Backpropagation

⓪ Initialize $\frac{dy}{dv} = 0$, $\forall v$, $\frac{dy}{dy} = 1$.

① $\frac{dy}{dd} += \frac{dy}{dy}\cdot\cancel{\frac{dy}{dd}}^{1} = 1$ $\quad \frac{dy}{de} += \frac{dy}{dy}\cdot\cancel{\frac{dy}{de}}^{1} = 1$ $\quad \frac{dy}{df} += \frac{dy}{dy}\cdot\cancel{\frac{dy}{df}}^{1} = 1$

② $\frac{dy}{dc} += \frac{dy}{df}\cdot\frac{df}{dc} = \frac{dy}{df}\cdot\frac{1}{a}$ $\quad \frac{dy}{da} += \frac{dy}{df}\cdot\frac{df}{da} = \frac{dy}{df}\cdot\left(\frac{-c}{a^2}\right)$

③ $\frac{dy}{da} += \frac{dy}{de}\cdot\frac{de}{da} = \frac{dy}{de}\cdot\frac{1}{b}$ $\quad \frac{dy}{db} += \frac{dy}{de}\cdot\frac{de}{db} = \frac{dy}{de}\cdot\left(\frac{-a}{b^2}\right)$

④ $\frac{dy}{da} += \frac{dy}{dd}\cdot\frac{dd}{da} = \frac{dy}{dd}\cdot\exp(a)$ 前面那个是存好的

⑤ $\frac{dy}{db} += \frac{dy}{dc}\cdot\frac{dc}{db} = \frac{dy}{dc}\cdot\cos(b)$ 下面那个是新算出来的

⑥ $\frac{dy}{dx} += \frac{dy}{db}\cdot\frac{db}{dx} = \frac{dy}{db}\cdot\frac{1}{x}$

⑦ $\frac{dy}{dx} += \frac{dy}{da}\cdot\frac{da}{dx} = \frac{dy}{da}\cdot\cancel{z}^{3}$ $\quad \frac{dy}{dz} += \frac{dy}{da}\cdot\frac{da}{dz} = \frac{dy}{da}\cdot\cancel{x}^{2}$

5. $J = -\sum_{k=1}^K y_k\log\hat{y}_k$

6. $g_{\hat{y}_k} = -\frac{y_k}{\hat{y}_k}$

4. $\hat{y}_k = \frac{\exp(b_k)}{\sum_{l=1}^K\exp(b_l)}$

7. $g_{b_k} = \sum_{l=1}^K g_{\hat{y}_l}\hat{y}_l(\mathbb{I}[k=l] - \hat{y}_k)$

3. $b_k = \sum_{j=0}^D \beta_{kj}z_j$

8. $g_{\beta_{kj}} = g_{b_k}z_j$

$g_{z_j} = \sum_{k=1}^K g_{b_k}\beta_{kj}$

2. $z_j = \frac{1}{1+\exp(-a_j)}$

9. $g_{a_j} = g_{z_j}z_j(1-z_j)$

1. $a_j = \sum_{m=0}^M \alpha_{jm}x_m$

10. $g_{\alpha_{ji}} = g_{a_j}x_i$