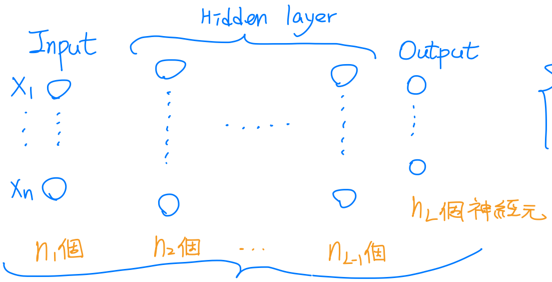
ML 第二週 筆記整理



$$\begin{cases} z^{[l]} = W^{[l]} a^{[l]} + b^{[l]} \\ a^{[l]} = \sigma(z^{[l]}) \end{cases}$$

$$h(x): \mathbb{R}^{n_1} \longrightarrow \mathbb{R}^{n_L}, \quad h(x) = a^{[L]}$$

Assignment #1 : Let $n_L = 1$, 計算梯度 $\nabla h = \left[\frac{\partial h}{\partial x_1} \cdots \frac{\partial h}{\partial x_{n_1}}\right]^T$

Defined : $\delta_j^{[l]} := \dfrac{\partial h}{\partial z_j^{[l]}}$

Then $\begin{cases} \text{output} : \delta_j^{[L]} = \dfrac{\partial h}{\partial z_j^{[L]}} = \dfrac{\partial(\sigma(z_j^{[L]}))}{\partial z_j^{[L]}} = \sigma'(z_j^{[L]}) \\ \text{hidden} : \delta_j^{[l-1]} = \dfrac{\partial h}{\partial z_j^{[l-1]}} = \dfrac{\partial h}{\partial z_j^{[l]}} \dfrac{\partial z_j^{[l]}}{\partial z_j^{[l-1]}} \end{cases}$

Classification

Data : $\{(\vec{x}, y)\}_{i=1}^{N}$, $y^i \in \{0, 1\}$

Method I :

Learn a function $h: \mathbb{R}^2 \to \mathbb{R}$. 再設閾值 (ex if $h(x^i) > \frac{1}{2}$. then $y^i = 1$)

缺 : 函數可能非平滑 , 在邊界附近可能不連續

Method II : One-hot encoding

將 y^i 以向量表示 (ex $y^i \in \{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\}$)

則 $h(x): \mathbb{R}^2 \to \mathbb{R}^2$. $h(\vec{x}) = \begin{bmatrix} h_0(\vec{x}) \\ h_1(\vec{x}) \end{bmatrix}$

$y^i = \operatorname{argmax}(h(\vec{x}^i))$ (有利於使用 softmax 層)

Supervised Learning 監督式學習

給 $\{(x^i, y^i)\}_{i=1}^n$，學習 h，使 $h(x^i) \approx y^i$

Regression
$$\begin{cases} y^i \text{ 為連續值} \\ \text{常用 Loss function (MSE): } \sum_i \|y^i - h(x^i)\|^2 \end{cases}$$

Classification
$$\begin{cases} y^i \text{ 為離散} \\ \text{常用 cross-entropy} \end{cases}$$

Assignment (Programing)

目標：Learn $h(x)$ s.t. $h(x) \approx f(x) = \frac{1}{1+25x^2}$，$x \in [-1, 1]$

延伸問題：$h'(x) \approx f'(x)$?

Maximum likelihood estimation (MLE)

目標：給 $\{x_i\}_{i=1}^n$，求 μ, σ^2 where $x \sim N(\mu, \sigma^2)$

$x \sim N(\mu, \sigma^2)$，$p(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Defined Likelihood function

$$L(\theta) := \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} e^{\sum_i -\frac{(x_i-\mu)^2}{2\sigma^2}}$$

The goal is find $\theta = \arg\max_\theta L(\theta)$

It is equal to the log-likelihood $\ln L(\theta) = -\frac{n}{2}(\ln 2\pi + 2\ln\sigma) - \sum_i \frac{(x_i-\mu)^2}{2\sigma^2}$

$$\begin{cases} \frac{\partial}{\partial\mu} \ln L(\theta) = \sum_{i=1}^n \frac{x_i-\mu}{\sigma^2} = \frac{1}{\sigma^2} \Sigma(x_i-\mu) = 0 & \Rightarrow \mu^* = \frac{1}{n}\sum_{i=1}^n x_i \\ \frac{\partial}{\partial\sigma^2} \ln L(\theta) = -\frac{n}{\sigma} + \sum_{i=1}^n \frac{(x_i-\mu)^2}{\sigma^3} = 0 & \Rightarrow \sigma^{*2} = \frac{1}{n}\sum_{i=1}^n (x_i-\mu)^2 \end{cases}$$

Mean Square Error (MSE)

Data : $\{(x_i, y_i)\}$

Loss $:= \frac{1}{n}\sum_{i=1}^{n} \| y_i - h(x_i)\|^2$

從高斯假設推出 MSE

$$\text{Let } \overset{\text{真實值}}{y_i} = \underset{\text{模型預測}}{h(x_i)} + \overset{\text{誤差項}}{\varepsilon_i} , \quad \varepsilon_i \sim N(0, \sigma^2)$$

$$\text{Since } \varepsilon_i \sim N(0,\sigma^2), \quad P(\varepsilon_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\varepsilon_i^2}{2\sigma^2}}$$

$$\text{Likelihood function } L(\theta) = \prod P(y_i \mid x_i, \theta) = \prod_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\varepsilon_i^2}{2\sigma^2}}$$

$$\text{log-Likelihood function } \ell(\theta) = \ln L(\theta) = -\frac{n}{2}(\ln 2\pi + \ln\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n} \| y_i - \underset{\substack{\text{hypothesis function} \\ \text{with parameters } \theta}}{h_\theta(x_i)} \|^2$$

$$\theta^* = \arg\max_\theta L(\theta) = \arg\max_\theta \ell(\theta)$$

$$= \arg\max_\theta \Sigma \| y_i - h_\theta(x_i)\|^2$$

$$= \arg\max_\theta \underbrace{\frac{1}{n}\sum_{i=1}^{n} \| y_i - h_\theta(x_i)\|^2}_{\text{MSE}} \quad \text{正規化}$$

Let $\sigma(x) = \tanh(x)$

性質

$\sigma = \tanh$ 為光滑奇函數

$\sigma(0)=0$, $\sigma'(0)=1$, $\sigma''(0)=0$

在 0 附近的泰勒展開 : $\sigma(x) = x + \frac{\sigma^{(3)}(0)}{3!}x^3 + \frac{\sigma^{(5)}(0)}{5!}x^5 + \cdots$

單項式的逼近 (用少量神經元) (這邊 h 代表很小的量 而 H(x) 代表神經網路)

逼近 x : 設 $h>0$ 很小

$$\frac{1}{h}\sigma(hx) = x + \frac{\sigma^{(3)}(0)}{6}h^2x^3 + O(h^4) \quad , \text{where } |x| \le 1$$

$\Rightarrow \frac{1}{h}\sigma(hx)$ 以 $O(h^2)$ 誤差逼近 x

逼近 x^3 : — 利用係數消去

$$\sigma(2hx) = 2hx + \frac{\sigma^{(3)}(0)}{6}(2hx)^3 + \cdots$$

$$\text{Let } C(x) = \sigma(2hx) - 2\sigma(hx) = \left[\frac{\sigma^{(3)}(0)}{6}(8h^3-2h^3)\right]x^3 + O(h^5)$$

$$= \sigma^{(3)}(0)\, h^3x^3 + O(h^5)$$

$$\text{Let } H(x) = \frac{C(x)}{\sigma^{(3)}(0)\, h^3} = x^3 + O(h^2)$$

$\Rightarrow$ 用 2 個權重為 $h, 2h$ 的 tanh 做線性組合. 即可以 $O(h^2)$ 逼近 x^3

逼近 x^2: $\sigma(h(x+\alpha)) = h(x+\alpha) + \frac{\sigma^{(3)}(0)}{6} h^3(x+\alpha)^3 \cdots$

$$\sigma(h(x-\alpha)) = h(x-\alpha) + \frac{\sigma^{(3)}(0)}{6} h^3(x-\alpha)^3 \cdots$$

$$\frac{\sigma(h(x+\alpha)) - \sigma(h(x-\alpha))}{6\sigma^{(3)}(0)h^3\alpha} - \frac{\alpha^2}{3} = x^2 + O(h^2)$$

⇒ 只須 4個隱藏神經元即可同時逼近 x、x^2、x^3 (以 $O(h^2)$ 誤差)

⇒ 一般化：透過 h、$2h$、$4h$‥組合與線性組合，即可以 $3n+2$個隱藏神經元

同時以任意精度逼近 $[x^0, x^1 \cdots, x^{2n}]$

(都是以 $O(h^2)$ 為誤差逼近的，取 $h \sim \sqrt{\varepsilon}$ 即可達到誤差小於 ε)

Deep Learning: An Introduction for Applied Mathematicians 筆記

Sigmoid function $\sigma(x) = \dfrac{1}{1+e^{-x}}$, $\sigma'(x) = \sigma(x)[1-\sigma(x)]$

↓

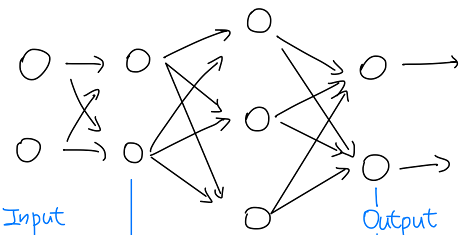
好處：$\sigma : \mathbb{R} \to [0,1]$ 可視作機率

平滑，單調，非線性

權重 W $\begin{cases} 列 \Rightarrow 當前層神經元數 \\ 行 \Rightarrow 前一層神經元數 \end{cases}$

b：分量數為當前層神經元數

↳ 決定臨界點位子



$h(x) : \mathbb{R}^2 \to \mathbb{R}^2$

輸出：

$\sigma(W^{[2]}_{2\times2} x + b^{[2]})$

$\sigma(W^{[3]}_{3\times2}\,\sigma(W^{[2]}_{2\times2}\, x + b^{[2]}) + b^{[3]})$

$W^{[4]}_{2\times3}\,\sigma(W^{[3]}_{3\times2}\,\sigma(W^{[2]}_{2\times2}\, x + b^{[2]}) + b^{[3]}) + b^{[4]}$

ex. 鑽油井例子：

有 10 個資料點 $\{x^i\}_{i=1}^{10}$，目標為把 x^i 分成 A, B 兩類

假設神經網路為上面那個

則 Cost function $:= \mathrm{Cost}(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]})$

∥

objective function $= \frac{1}{10} \cdot \frac{1}{2} \sum_{i=1}^{10} \| y(x^i) - h(x^i) \|_2^2$

(使用 Matlab 的 nonlinear least-squares solver — lsqnonlin)

⇒ 問題：
- 無法窮舉 23維空間找最小值 ← 若將矩陣每項都視為變數，則有23個
- 無法保證非凸函數具全域最小

⇒ 解決方法：梯度下降

The General Setup

有 L 層，每層有 n_i 個神經元

$a^{[1]} = x \in \mathbb{R}^{n_1}$ （輸入）

$a^{[l]} = \sigma(W^{[l]} a^{[l-1]} + b^{[l]}) \in \mathbb{R}^{n_l}$, with $W^{[l]} \in \mathbb{R}^{n_l} \times \mathbb{R}^{n_{l-1}}$, $b^{[l]} \in \mathbb{R}^{n_l}$

$F: \mathbb{R}^{n_1} \longrightarrow \mathbb{R}^{n_L}$

Gradient Descent

Let $p \in \mathbb{R}^s$, $\text{Cost}: \mathbb{R}^s \to \mathbb{R}$

$$\text{Cost}(p + \Delta p) = \text{Cost}(p) + \sum_{i=1}^{s} \frac{\partial \text{Cost}(p)}{\partial p_i} \Delta p_i + O(\Delta p^2)$$

Defined $(\nabla \text{Cost}(p))_i = \dfrac{\partial \text{Cost}(p)}{\partial p_i}$

$$\Rightarrow \text{Cost}(p + \Delta p) = \text{Cost}(p) + (\nabla \text{Cost}(p))^T \Delta p$$

By Cauchy-Schwarz inequality $|f^T g| \le \|f\|_2 \|g\|_2$, we have

$$-|(\nabla \text{Cost}(p))^T \Delta p| \ge -\|\nabla \text{Cost}(p)\|_2 \|\Delta p\|_2$$

when Δp 為 $\nabla \text{Cost}(p)$ 的反方向時，$(\nabla \text{Cost}(p))^T \Delta p = -\|\nabla \text{Cost}(p)\|_2 \|\Delta p\|_2$

$$\Rightarrow \Delta p = -\underbrace{\alpha}_{\text{learning rate}} \nabla \text{Cost}(p)$$

SGD $\begin{cases} \text{有放回抽樣} \\ \text{沒 〃} \\ \text{小批次 (mini-batch)} \end{cases}$

Defined 單筆資料成本 : $C_{x^i} = \frac{1}{2}\| y^i - a^{[L]}(x^i) \|_2^2$

隨機取 i

更新參數 $p \to p - \alpha \nabla C_{x^i}(p)$

反向傳播 (Back Propagation)

(i) 簡寫成 $C = \frac{1}{2}\| y - a^{[L]} \|_2^2$

Def 加權輸入 weight input $z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ — (ii)

$z_j^{[l]}$ 第 l 層第 j 個加權輸入

$\Rightarrow$ 前向傳遞 $a^{[l]} = \sigma(z^{[l]})$ ——— (iii)

Def 誤差變數 error term $\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}}$ — (iv)

Def Hadamard 乘積 : $x, y \in \mathbb{R}^n$. $(x \circ y)_i = x_i y_i$

證明以下結果:

① $\delta^{[L]} = \sigma'(z^{[L]}) \circ (a^{[L]} - y)$

② $\delta^{[l]} = \sigma'(z^{[l]}) \circ (W^{[l+1]})^T \delta^{[l+1]}$, $2 \le l \le L-1$

③ $\frac{\partial C}{\partial b_j^{[l]}} = \delta_j^{[l]}$ $\quad 2 \le l \le L$

④ $\frac{\partial C}{\partial w_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]}$ $\quad 2 \le l \le L$

Proof ①

由 (iv) 得 $a^{[L]} = \sigma(z^{[L]}) \Rightarrow a_j^{[L]} = \sigma(z_j^{[L]})$

$$\Rightarrow \frac{\partial a_j^{[L]}}{\partial z_j^{[L]}} = \sigma'(z_j^{[L]})$$

由 (i) . $C = \frac{1}{2}\sum_{k=1}^{n_L}(y_k - a_k^{[L]}) \Rightarrow \frac{\partial C}{\partial a_j^{[L]}} = -(y_j - a_j^{[L]})$

由 (iv) $\delta_j^{[L]} := \frac{\partial C}{\partial z_j^{[L]}} = \frac{\partial C}{\partial a_j^{[L]}} \frac{\partial a_j^{[L]}}{\partial z_j^{[L]}} = (a_j^{[L]} - y_j)\sigma'(z_j^{[L]})$

Proof ②

$$z_k^{[l+1]} = \sum_{s=1}^{n_l} w_{ks}^{[l+1]}\sigma(z_s^{[l]}) + b_k^{[l+1]}$$

$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \frac{\partial C}{\partial z_k^{[l+1]}} \frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \frac{\partial C}{\partial z_k^{[l+1]}} \cdot w_{kj}^{[l+1]}\sigma'(z_j^{[l]})$$

$$= \sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} \cdot w_{kj}^{[l+1]}\sigma'(z_j^{[l]})$$

$$\Rightarrow \delta_j^{[l]} = \sigma'(z_j^{[l]})\sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} \cdot w_{kj}^{[l+1]} = \sigma'(z_j^{[l]})\left((W^{[l+1]})^T\delta^{[l+1]}\right)_j$$

Proof ③

$$z_j^{[l]} = (W^{[l]}a^{[l-1]})_j + b_j^{[l]} \Rightarrow \frac{\partial z_j^{[l]}}{\partial b_j^{[l]}} = 1$$

$$\frac{\partial C}{\partial b_j^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial b_j^{[l]}} = \delta_j^{[l]}$$

Proof ④

$$z_j^{[l]} = \sum_{k=1}^{n_{l-1}} w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]} \Rightarrow \frac{\partial z_j^{[l]}}{\partial w_{jk}^{[l]}} = a_k^{[l-1]} \text{ and } \frac{\partial z_s^{[l]}}{\partial w_{jk}^{[l]}} = 0 \quad s \neq j$$

$$\frac{\partial C}{\partial w_{jk}^{[l]}} = \sum_{s=1}^{n_l} \frac{\partial C}{\partial z_s^{[l]}} \frac{\partial z_s^{[l]}}{\partial w_{jk}^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial w_{jk}^{[l]}} = \frac{\partial C}{\partial z_j^{[l]}} a_k^{[l-1]} = \delta_j^{[l]} a_k^{[l-1]}$$

卷積神經網路 CNNs

200 像素

利用限制權重矩陣來解決輸入項太多 $200 \times 200 \times 3 = 120000$

$$y_k = \sum_{n=1}^{p} x_n g_{k-n}$$

卷積後常接上池化層 (pooling layer) 將小區域像素壓成單一數字

為避免過擬合 ① 將資料分為 $\begin{cases} \text{訓練 training data} \\ \text{驗證 validation data} \end{cases}$

② 隨機刪除部分神經元

$$\sigma(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases}$$ rectified linear unit (ReLU)

訓練資料 $\{x^{\{i\}}\}_{i=1}^{N}$，第 i 個的標籤為 $l_i \in \{1, 2, \dots, K\}$

Let $a^{[L]}(x^{\{i\}}) = v^{\{i\}} \in \mathbb{R}^K$

Defined softmax operation $(v^{\{i\}})_s \mapsto \dfrac{e^{v_s^{\{i\}}}}{\sum_{j=1}^{K} e^{v_j^{\{i\}}}}$

使用對數誤差度量 $\text{Loss} = -\sum_{i=1}^{N} \log\left(\dfrac{e^{v_{l_i}^{\{i\}}}}{\sum_{j=1}^{K} e^{v_j^{\{i\}}}}\right)$

↳ 放大懲罰 ex $\begin{cases} (1-0.001)^2 = 0.998 \\ -\log(0.001) \approx 6.9 \end{cases}$

softmax log loss = 交叉熵 cross-entropy

未涉及內容:

更進一步 [27]

核心概念的總覽 [23]

更詳細之介紹 [30]

歷史 [36]

深度學習廣泛應用 [11, 23, 27, 30, 36]

機器學習優化問題，發展與收斂問題 [3]

線非線性轉換 [25]

證明利用隨機梯度法依然行為良好 [16]

對影像做小擾動，就可能改變結果 [33]

對抗性貼片 (adversarial patch) [4]

介紹一系列數學方法 [35]

非線性函數選擇？

常見的有 $\sigma(x) = \begin{cases} 0 & x \le 0 \\ 1 & x > 0. \end{cases}$ (ReLU)

若飽合（導數很小使梯度更新慢）可使用 Leaky ReLU

$f(x) = \begin{cases} \lambda x & x \le 0 \\ x & x > 0 \end{cases}$ 以在負區域保留非零導數

Stanford CS229 lecture notes 1.3 ~ 1.4

1.3 讓最小平方法合理化的一種假設

Let 目標變數和輸入的關係 $\Rightarrow y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$

↳ 誤差項
假設其獨立同分布
且 $\varepsilon^{(i)} \sim N(0, \sigma^2)$

$$\Rightarrow p(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2}\right)$$

$$\varepsilon^{(i)} = y^{(i)} - \theta^T x^{(i)}$$

$$\Rightarrow p(y^{(i)} \mid x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

給定 X 和 θ，$y^{(i)}$ 的分佈為 $p(\vec{y} \mid X; \theta)$

我們希望它是 θ 的函數，稱為(likelihood function) $L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y} \mid X; \theta)$

由於 $\varepsilon^{(i)}$ 為獨立，給定 $x^{(i)}$ 時，$y^{(i)}$ 也互相獨立

固定但未知

$$\Rightarrow L(\theta) = \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

By 最大似然原理 maximum likelihood principle

我們要尋找 $\hat{\theta} = \arg\max_{\theta} L(\theta)$

我們令 $\ell(\theta) = \ln L(\theta)$，來使問題更好算（log 為單調）

$$\Rightarrow \ell(\theta) = \ln \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= n \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2$$

$\Rightarrow$ 最大化 $\ell(\theta)$ = 最小化 $\frac{1}{2}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2$

1.4 局部加權線性迴歸 Locally Weighted Linear Regression, LWR

一般：擬合 $\sum_i (y^{(i)}-\theta^T x^{(i)})^2$，輸出 $\theta^T x$

LWR：擬合 $\sum_i w^{(i)}(y^{(i)}-\theta^T x^{(i)})^2$ 〃

一種非參數式演算法 non-parameter algorithm

(線性迴歸屬於參數式演算法)

訓練完就不須保留訓練資料.

$w^{(i)}$：權重

↳ 常見選擇：$w^{(i)} = \exp\left(-\frac{(x^{(i)}-x)^2}{2\tau^2}\right)$

x：想預測的點

τ：帶寬參數 bandwidth parameter