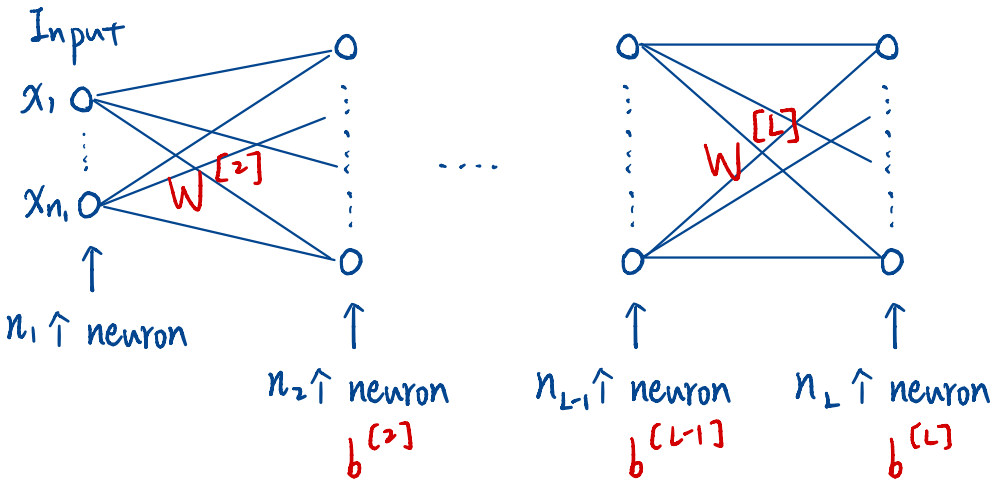


$$z^{[2]} = W^{[2]} x + b^{[2]}, \quad a^{[2]} = \sigma(z^{[2]}) \in \mathbb{R}^{n_2}$$

$z^{[2]} \in \mathbb{R}^{n_2}$, $W^{[2]} \in M_{n_2 \times n_1}$, $x \in \mathbb{R}^{n_1}$, $b^{[2]} \in \mathbb{R}^{n_2}$

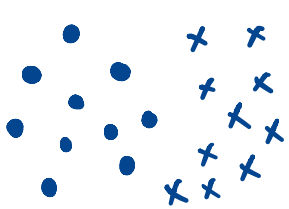
$$z^{[L]} = W^{[L]} a^{[L-1]} + b^{[L]}, \quad a^{[L]} = \sigma(z^{[L]})$$

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

1. $n_L = 1$, $H : \mathbb{R}^{n_1} \to \mathbb{R}^1$. $\nabla H = ?$

Classification

data : $\{(\vec{x}_i, c_i)\}$, $c_i \in \{0, 1\}$.



Method 1: Find a function : $\mathbb{R}^2 \to \mathbb{R}$ s.t. $H(\vec{x}_i) = C_i$.

Method 2: One-hot encoding $C_i \in \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$

Find a function : $\mathbb{R}^2 \to \mathbb{R}^2$ s.t. $H(\vec{x}_i) = C_i$.

Supervised learning

- Regression : Target takes continuous value

  ex : House price
- Classification : Target takes discrete value

MSE loss

$$L = \sum_i \| C_i - H(x_i) \|^2$$

Note : This is uncommon in classification.

2 Programming assignment

$$f(x) = \frac{1}{1+25x^2} \quad . \quad x \in [-1, 1]$$

Function approximation / Regression Hypothesis ?

Extra : Accuracy of $H'(x)$?

Maximum Likelyhood Estimation (MLE)

The probability density function of a normal distribution is

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

μ: mean, σ^2: varience, σ: standard deviation

Given data drown from $N(\mu, \sigma^2)$, can we recover μ and σ?

data: $\{x_i\}_{i=1}^n$ (assume independent and drawn from identical normal distribution)

likelyhood function

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

To find μ and σ s.t. $L(\mu, \sigma)$ is maximized.

Let $\theta = (\mu, \sigma)$, $\theta^* = \underset{\theta}{\operatorname{argmax}}\, L(\theta) = \underset{\theta}{\operatorname{argmax}}\, \ln(L(\theta))$

$$l(\theta) = \ln(L(\theta)) = \frac{-n}{2}(\ln 2\pi + 2\ln\sigma) - \frac{1}{2\sigma^2}\sum_i (x_i-\mu)^2$$

$$\frac{\partial l}{\partial \mu} = 0 \Leftrightarrow -\frac{1}{2\sigma^2}\sum_i (-2)(x_i-\mu) = 0$$

$$\Leftrightarrow \sum_i (x_i-\mu) = 0$$

$$\therefore \mu^* = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\frac{\partial l}{\partial \sigma} = 0 \Leftrightarrow \frac{-n}{\sigma} + \frac{1}{\sigma^3}\sum_i (x_i-\mu)^2 = 0$$

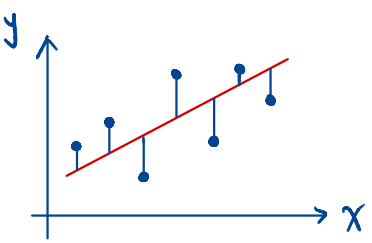
$$\Leftrightarrow \sigma^2 = \frac{1}{n}\sum (x_i-\mu)^2$$

$$\therefore \sigma^* = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

mean square error (MSE) loss

$\{(x_i, y_i)\}_i$

$$\text{Loss} = \frac{1}{n}\sum_i \| y_i - H(x_i) \|^2$$



Assume $y_i = H(x_i) + \varepsilon_i$, $\varepsilon_i \sim N(0, \sigma^2)$

$$P(\varepsilon) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\varepsilon^2}{2\sigma^2}} \qquad L(\theta) = \prod_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\varepsilon_i^2}{2\sigma^2}}$$

$$l(\theta) = \ln(L(\theta)) = \frac{-n}{2}(\ln 2\pi + 2\ln\sigma) - \frac{1}{2\sigma^2}\sum_i \varepsilon_i^2$$

$$= \frac{-n}{2}(\ln 2\pi + 2\ln\sigma) - \frac{1}{2\sigma^2}\sum_i \|y_i - H_\theta(x_i)\|^2$$

$$\theta^* = \arg\max_\theta \ell(\theta) = \arg\min_\theta \frac{1}{2\sigma^2}\sum_i \|y_i - H_\theta(x_i)\|^2$$

$$= \arg\min_\theta \underbrace{\frac{1}{n}\sum_i \|y_i - H_\theta(x_i)\|^2}_{\text{MSE loss}}$$

CS229 Ch 1

Locally weighted linear regression (LWLR)

For a given point x.

$$w_i = \exp\left(\frac{-(x_i - x)^2}{2\tau^2}\right)$$

loss: $\sum_i w_i \|y_i - (ax_i + b)\|^2$ (weighted inner product)

Ryck et al.

On the approximation of functions, by tanh neural networks.

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

σ : odd function $\Rightarrow \sigma^{(2n)}(0) = 0$

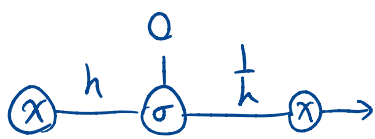
Method 1.

$$\sigma(x) = \cancel{\sigma(0)} + \sigma'(0)x + \cancel{\frac{\sigma''(0)}{2}x^2} + \frac{\sigma'''(0)}{6}x^3 + \cancel{\frac{\sigma^{(4)}(0)}{4!}x^4} + \cdots$$

$$= x + \frac{\sigma'''(0)}{6}x^3 + \frac{\sigma^{(5)}(0)}{120}x^5 + \cdots$$

Approximate x'

$$\sigma(hx) = hx + \frac{\sigma'''(\xi)}{6}(hx)^3$$



Assume $|\sigma'''(x)| < M_3$, $-1 \le x \le 1$.

$$\Rightarrow \left|\frac{1}{h}\sigma(hx) - x\right| \le \frac{M_3}{6}h^2 < \varepsilon \Rightarrow h \approx O(\sqrt{\varepsilon})$$

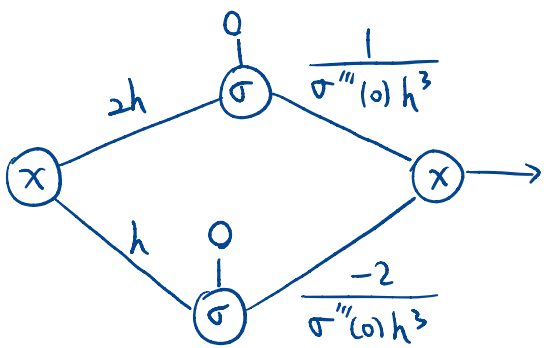
Conclusion : $\tanh(x)$ neural network with 1-neuron in the hidden and output layer can approximate $f(x) = x$, $x \in [-1, 1]$ to any desired accuracy.

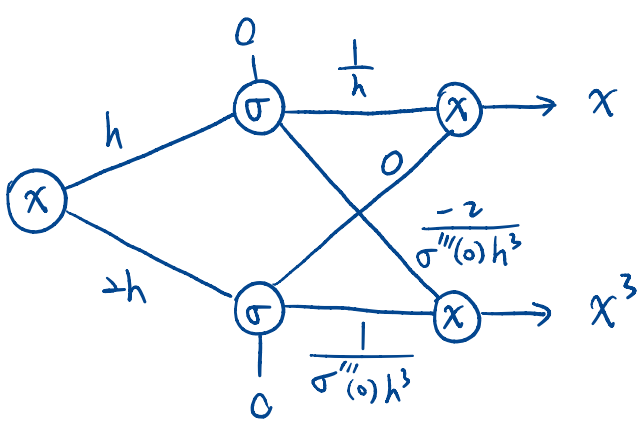
$$\sigma(hx) = hx + \frac{\sigma'''(0)}{6}(hx)^3 + \frac{\sigma^{(5)}(\xi_1)}{120}(hx)^5 \quad \cdots ②$$

$$\sigma(2hx) = 2hx + \frac{\sigma'''(0)}{6}(2hx)^3 + \frac{\sigma^{(5)}(\xi_2)}{120}(2hx)^5 \quad \cdots ③$$

Assume $|\sigma^{(5)}(x)| \le M_5$, $-1 \le x \le 1$.

$$③ - 2\times② : \left| \frac{\sigma(2hx) - 2\sigma(hx)}{\sigma'''(0)h^3} - x^3 \right| \le \frac{3M_5}{4|\sigma'''(0)|}h^2$$





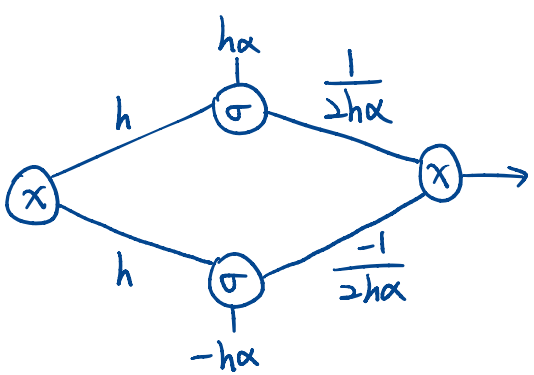
Conclusion : $\tanh(x)$ neural network with n-neuron in the hidden and output layer can approximate x^{2p-1}, $p = 1, 2, \cdots, n$, $x \in [-1, 1]$ to any desired accuracy.

Approximate x^0

$$\sigma(h(x+\alpha)) = h(x+\alpha) + \frac{\sigma'''(\xi_1)}{6}(h(x+\alpha))^3$$

$$\sigma(h(x-\alpha)) = h(x-\alpha) + \frac{\sigma'''(\xi_2)}{6}(h(x-\alpha))^3$$

$$\Rightarrow \left| \frac{1}{2h\alpha}\left(\sigma(hx+h\alpha) - \sigma(hx-h\alpha)\right) - 1 \right| \leq \tilde{M}h^2$$



Approximate x^2

$$\frac{\sigma(2h(x+\alpha)) - 2\sigma(h(x+\alpha))}{\sigma'''(0)\,h^3} = (x+\alpha)^3 + \frac{\sigma^{(5)}(\xi_1)}{4\sigma'''(0)}(x+\alpha)^5 h^2$$

$$-) \quad \frac{\sigma(2h(x-\alpha)) - 2\sigma(h(x-\alpha))}{\sigma'''(0)\,h^3} = (x-\alpha)^3 + \frac{\sigma^{(5)}(\xi_2)}{4\sigma'''(0)}(x-\alpha)^5 h^2$$

$$\div 6\alpha \Rightarrow \left| \frac{\sigma(2h(x+\alpha)) - 2\sigma(h(x+\alpha)) - \sigma(2h(x-\alpha)) + 2\sigma(h(x-\alpha))}{\sigma'''(0)\,h^3 \cdot 6\alpha} - \frac{\alpha^2}{3} - x^2 \right| \leq \tilde{\tilde{M}} h^2$$

Conclusion : $\tanh(x)$ neural network with $3n+2$ neuron in hidden layer and $2n+1$ neuron in output layer can approximate x^p, $p = 0, 2, \cdots, 2n$, $x \in [-1, 1]$ to any desired accuracy.

Week 3

#1. Lemma 3.1, Lemma 3.2