1. Hard-Coding Networks.

1.1. Verify Sort

$$W^{(1)} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad b^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

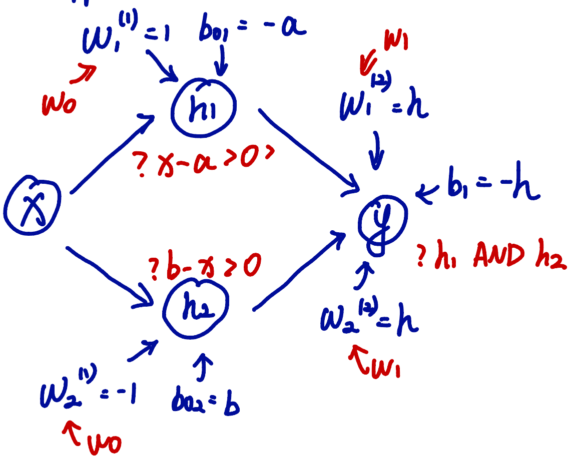
$$W^{(2)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad b^{(2)} = -2.5$$

1.2. Perform Sort:

```
P ← permutation (x1, x2, x3, x4).
for p in P:
    if verifysort(p) > 0:
        return p
```

1.3. Universal Approximation Theorem.

1.3.1.



$w_1 a(w_0 x + b_0) + b_1 = h\,\mathbb{I}(a < x < b)$

$w_0 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ $b_0 = \begin{bmatrix} -a \\ b \end{bmatrix}$. $a = \mathbb{I}(y \geq 0)$

$w_1 = [h, h]$. $b_1 = -h$

1.3.2. $\|f - \hat{f}_1\| = \int_I |f(x) - \hat{f}_1(x)|dx = \int_{-1}^{1} |-x^2+1 - 0 - g(h_1, a_1, b_1, x)|\,dx$ ①, suppose $[a, b] \in I$

① $\Rightarrow$ $|-x^2+1-h| > 0$

$= \int_{-1}^{a} -x^2+1\,dx + \int_b^1 -x^2+1\,dx + \int_a^b |-x^2+1-h|\,dx$

$= \int_{-1}^{1} -x^2+1\,dx + \int_a^b -h\,dx$ $\qquad \int_I -x^2+1 = -\frac{x^3}{3} + x\Big]_{-1}^{1} = \frac{4}{3} = \|f - \hat{f}_0\|$

$= \frac{4}{3} - h(b-a) < \frac{4}{3}$

$h(b-a) > 0$

$|-x^2+1-h| < 0$
① $\Rightarrow$

$\int_{-1}^{a} -x^2+1\,dx + \int_b^1 -x^2+1\,dx + \int_a^b x^2-1+h\,dx$

$= -\frac{x^3}{3} + x\Big]^{[-1, a]}_{[b, 1]} + \frac{x^3}{3} - x + hx\Big]_a^b$

$= -\frac{a^3}{3} + a - \left[\frac{1}{3} - 1\right] + -\frac{1}{3} + 1 - \left[-\frac{b^3}{3} + b\right] + \frac{b^3}{3} - b + hb - \left[\frac{a^3}{3} - a + ha\right]$

$= -\frac{2}{3}a^3 + \frac{2}{3}b^3 + 2a - 2b + hb - ha + \frac{4}{3} < \frac{4}{3}$

$(b-a)(h-2) < \frac{2}{3}(a^3 - b^3)$

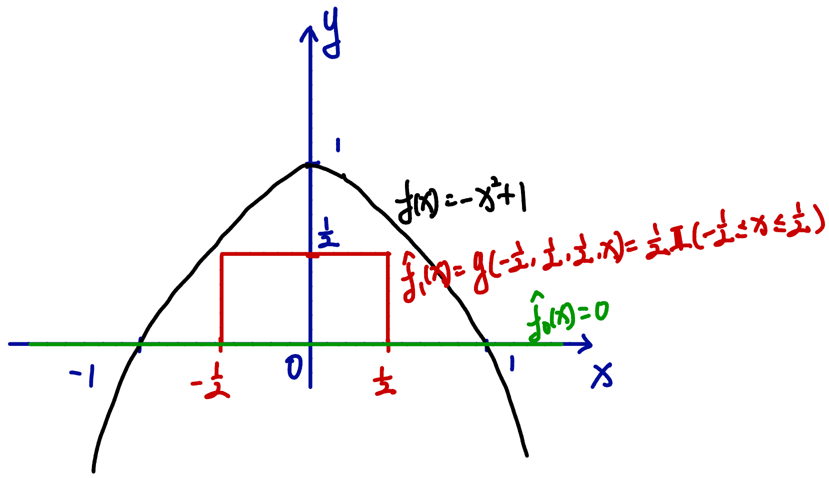
take $\hat{f}_1 = g(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, x)$.

$b = \frac{1}{2}$, $a = -\frac{1}{2}$, $h = \frac{1}{2}$

$(\frac{1}{2} + \frac{1}{2}) \cdot (\frac{1}{2} - 2) < \frac{2}{3}((-\frac{1}{2})^3 - (\frac{1}{2})^3)$

$-\frac{3}{2} < -\frac{1}{6}$

$\frac{1}{2} \cdot (\frac{1}{2} + \frac{1}{2}) > 0$ $\Rightarrow$ checks.



1.3.3. $\hat{f}_1(x) = \hat{f}_0(x) + g(-\frac{1}{2^N}, \frac{1}{2^N}, \frac{1}{2^N}, x)$.

$\hat{f}_2(x) = \hat{f}_1(x) + g(-\frac{1}{2^{N-1}}, \frac{1}{2^{N-1}}, \frac{1}{2^{N-1}}, x)$

$\vdots$

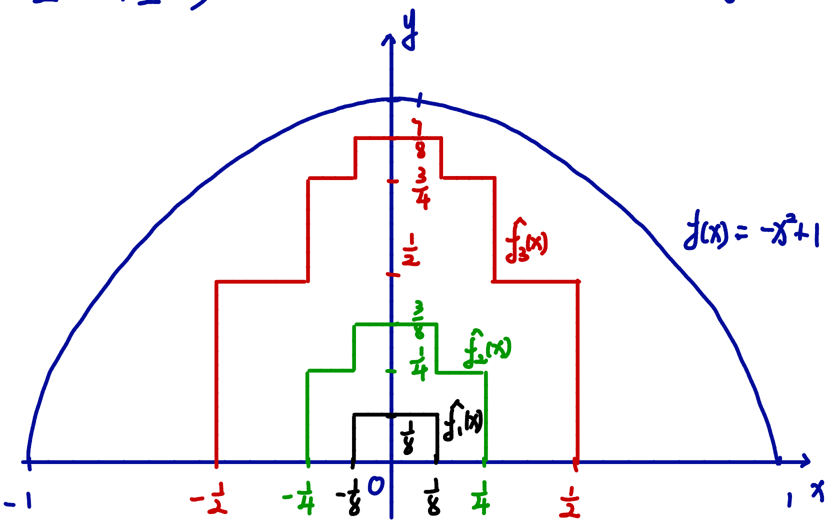
$\hat{f}_N(x) = \hat{f}_{N-1}(x) + g(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, x)$

$\Rightarrow$

$N=3$. $\hat{f}_1(x) = g(-\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, x)$

$\hat{f}_2(x) = g(-\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, x) + g(-\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, x)$

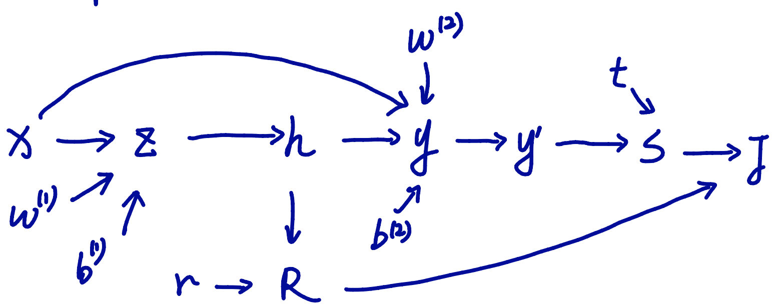
$\hat{f}_3(x) = g(-\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, x) + g(-\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, x) + g(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, x)$



2. Backprop.

2.1 Computational Graph

2.1.1.



2.1.2.

$\bar{J} = dJ/dJ = 1$

$\bar{S} = dJ/dS = 1$

$\bar{y}' = dJ/dy' = dJ/dS \cdot dS/dy' = \bar{S} \cdot \mathbb{I}(t=k)$

$\bar{y} = dJ/dy = dJ/dy' \cdot dy'/dy = \bar{y}' \cdot \text{softmax}'(y)$

$\bar{R} = dJ/dR = 1$

$\bar{h} = dJ/dh = dJ/dy \cdot dy/dh + dJ/dR \cdot dR/dh = W^{(2)T}\bar{y} + \bar{R}r$

$\bar{z} = dJ/dz = dJ/dh \cdot dh/dz = \begin{cases} \bar{h} & z > 0 \\ 0 & z \le 0 \end{cases}$

$\bar{x} = dJ/dx = dJ/dz \cdot dz/dx + dJ/dy \cdot dy/dx$

$= W^{(1)T}\bar{z} + \bar{y}$

2.2. VJPs.

2.2.1. $f(x) = vv^T x = \begin{bmatrix}1\\2\\3\end{bmatrix}[1\ 2\ 3]\, x = \begin{bmatrix}1&2&3\\2&4&6\\3&6&9\end{bmatrix} x$

$\nabla f(x) = J = \begin{bmatrix}1&2&3\\2&4&6\\3&6&9\end{bmatrix}$

2.2.2.

$J = v \cdot v^T \Rightarrow$ time: n^2 multiplication

space: n^2

2.2.3.

$J^T y = [vv^T]^T y = v^T v y = v^T y\, v \Rightarrow \underbrace{a = v^T y}_{\substack{\text{linear} \\ \text{time + space}}},\ \underbrace{J^T y = a \cdot v}_{\substack{\text{linear} \\ \text{time + space}}}$ a is a scalar.

linear time + space + linear time + space $\Rightarrow$ Linear time + space.

$z = J^T y \quad a = v^T y = [1, 1, 1]\begin{bmatrix}1\\2\\3\end{bmatrix} = 6$

$z = a \cdot v = 6v = \begin{bmatrix}6\\12\\18\end{bmatrix} \quad z^T = [6, 12, 18].$

3. Linear Regression

3.1. $L = \frac{1}{n}\|X\hat{w} - t\|_2^2 = \frac{1}{n}(X\hat{w}-t)^T(X\hat{w}-t) = \frac{1}{n}(\hat{w}^T X^T - t^T)(X\hat{w} - t) = \frac{1}{n}(\hat{w}^T X^T X \hat{w} - t^T X \hat{w} - \hat{w}^T X^T t + t^T t)$

$\partial L / \partial \hat{w} = \frac{1}{n}(2X^T X \hat{w} - 2X^T t) = \frac{2}{n} X^T (X\hat{w} - t)$

3.2.

3.2.1. $\partial L/\partial \hat{w} = 0 \Rightarrow X^T(X\hat{w} - t) = 0 \quad X^T X \hat{w} = X^T t$

$\hat{w} = (X^T X)^{-1} X^T t$

3.2.2. $t_i = w^{*T} x_i. \Rightarrow t = X w^*$

$\hat{w} = (X^T X)^{-1} X^T X w^* \Rightarrow \hat{w} = w^*$

3.3.

3.3.1. $w = \begin{bmatrix}w_1\\w_2\end{bmatrix} \quad w^T x_1 = t. \quad 2w_1 + w_2 = 2 \Rightarrow$ infinity many w_1, w_2 satisfies.

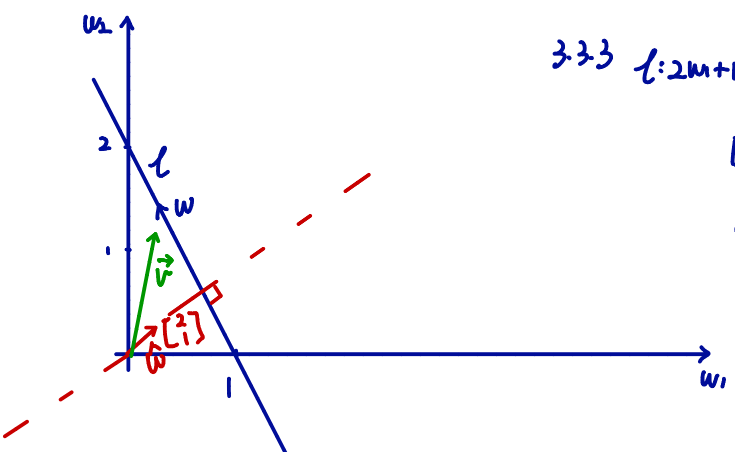
3.3.2. $\partial L/\partial \hat{w} = \frac{2}{n} X^T (X\hat{w} - t) = 2 \cdot \begin{bmatrix}2\\1\end{bmatrix}\left([2, 1]\begin{bmatrix}0\\0\end{bmatrix} - 2\right) = -4\begin{bmatrix}2\\1\end{bmatrix}.$

$w_1 \leftarrow w_0 - \alpha\, \partial L/\partial \hat{w} \qquad w_1 = c \cdot \begin{bmatrix}2\\1\end{bmatrix}.$

$\partial L/\partial \hat{w} = 2 \cdot \begin{bmatrix}2\\1\end{bmatrix} \cdot \left([2, 1] \cdot c \begin{bmatrix}2\\1\end{bmatrix} - 2\right) = k\begin{bmatrix}2\\1\end{bmatrix}$

$\Rightarrow w_2 = d\begin{bmatrix}2\\1\end{bmatrix} \cdots \Rightarrow W_i.$ for $i \in \mathbb{Z}^+$, $W_i = a\begin{bmatrix}2\\1\end{bmatrix}$. $a \in \mathbb{R}$

$\therefore$ all weight in direction of $\hat{w} = \begin{bmatrix}2\\1\end{bmatrix}$



3.3.3 $l: 2w_1 + w_2 = 2$ has direction: $w = \begin{bmatrix}1\\-2\end{bmatrix}$

$[1, -2] \cdot \begin{bmatrix}2\\1\end{bmatrix} = 0$

for any $\vec{v} = w - \hat{w}$

when $\vec{v} \perp w$, $\vec{v} = \hat{w}$, and has smallest length/Euclidean norm by $\triangle$ enclosed by $\vec{v}, \hat{w}, w$

3.4.

3.4.1

$$\partial L/\partial w = \frac{2}{n}\sum_{1}^{n} x_i^T (w^T x_i - t_i)$$

$$\hat{w}_0 = 0$$

$$\partial L/\partial w = \frac{2}{n}\sum_{1}^{n} - t_i x_i^T$$

$$\hat{w}_1 = \hat{w}_0 - \alpha\, \partial L/\partial w = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = X^T \cdot C$$

$$\partial L/\partial w = \frac{2}{n}\sum_{1}^{n} x_i^T (C^T X x_i - t_i) = X^T \cdot D \quad \cdots$$

For each $\hat{w}_i$, $i \in \mathbb{Z}^+$, $\hat{w}_i = X^T \cdot C$, $C \in \mathbb{R}^{n \times 1}$

$$\hat{w} = X^T C \qquad X\hat{w} = t$$

$$\Rightarrow X X^T C = t \qquad C = (XX^T)^{-1} t$$

$$\hat{w} = X^T (XX^T)^{-1} t$$

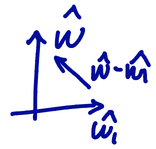
3.4.2. $X\hat{w}_1 = t \qquad \hat{w}_1 = X^{-1} t \qquad \hat{w}_1^T = t^T (X^{-1})^T$

$$\hat{w}^T = t^T [X^T (XX^T)^{-1}] = t^T [(XX^T)^{-1}]^T X$$

$$\hat{w}^T \hat{w} = t^T [(XX^T)^{-1}]^T X X^T (XX^T)^{-1} t = t^T (XX^T)^{-1} (XX^T)(XX^T)^{-1} t = t^T (XX^T)^{-1} t$$

$$\hat{w}_1^T \hat{w} = t^T (X^{-1})^T X^T (XX^T)^{-1} t = t^T (XX^T)^{-1} t$$

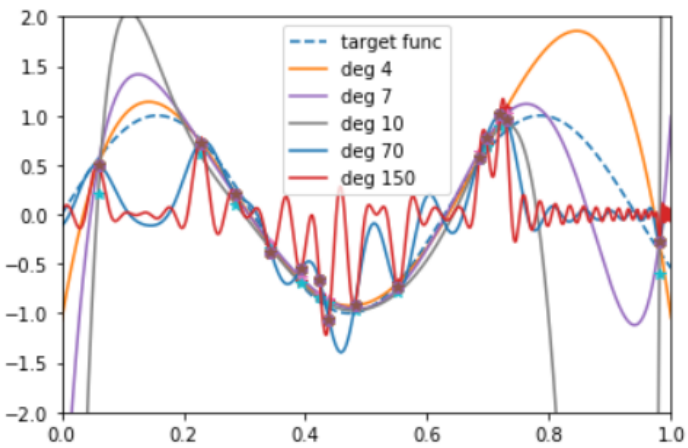
$$\therefore (\hat{w} - \hat{w}_1)^T \hat{w} = \hat{w}^T \hat{w} - \hat{w}_1^T \hat{w} = 0$$



$\therefore$ $\hat{w}$ has smallest distance / Euclidean norm among all possible $w \Rightarrow \hat{w} \perp (\hat{w} - \hat{w}_1)$

```
# to be implemented; fill in the derived solution for the underparameterized (d<n) and overparameterized (d>n) problem

def fit_poly(X, d, t):
    X_expand = poly_expand(X, d=d, poly_type = poly_type)
    if d > n:
        W = np.dot(np.dot(np.transpose(X_expand), np.linalg.inv(np.dot(X_expand, np.transpose(X_expand)))), t)
    else:
        W = np.dot(np.dot(np.linalg.inv(np.dot(np.transpose(X_expand), X_expand)), np.transpose(X_expand)), t)
    return W
```



Higher degree polynomial / Overparameterization does not always leads to overfitting.

degree 70, 100, 150, 200 shows better generalization and smaller error than degree 10-50.