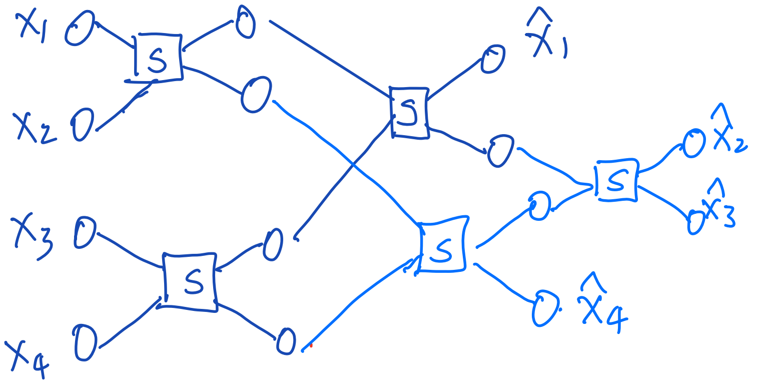
Q1

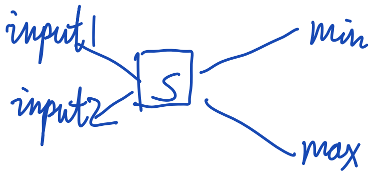
1.1: given input $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ as column vector:

$$W^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad b^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \phi^{(1)}(z) = \begin{bmatrix} |z_1| \\ |z_2| \end{bmatrix} = |z|$$

$$W^{(2)} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \quad b^{(2)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \phi^{(2)}(z) = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z$$

1.2





Q2

2.1.2

$\bar{J} = 1;$

$\bar{S} = \bar{J} \cdot \frac{\partial J}{\partial S} = -J = -1$

$\bar{y'} = \bar{S} \cdot \frac{\partial S}{\partial y'} = \bar{S} \cdot \left[\frac{\partial S}{\partial y'_1} \cdots \frac{\partial S}{\partial y'_N}\right]^T, \quad \frac{\partial S}{\partial y'_i} = \begin{cases} \frac{1}{y_i} & \text{if } I(t=i)=1 \\ 0 & \text{if } I(t=i)=0 \end{cases}$

$\bar{y} = \bar{y'} \frac{\partial y'}{\partial y} = \bar{y'}^T \cdot \text{softmax}'(y)$

$\bar{g} = \bar{y} \cdot \frac{\partial y}{\partial g} = W_3^T \cdot \bar{y}$

$\bar{h_1} = \bar{g} \cdot \frac{\partial g}{\partial h_1} = \bar{g} \circ h_2 \qquad \bar{h_2} = \bar{g} \frac{\partial g}{\partial h_2} = \bar{g} \circ h_1$

$$\sigma'(z) = \left(\frac{1}{1+e^{-z}}\right)' = \frac{-(-e^{-z})}{(1+e^{-z})(1+e^{-z})} = \frac{1+e^{-z}-1}{(1+e^{-z})(1+e^{-z})} = \sigma(z)\circ(1-\sigma(z))$$

$$\bar{z}_2 = \bar{h}_2 \cdot \frac{\partial h_2}{\partial z_2} = \bar{h}_2 \circ \sigma(z_2)\circ(1-\sigma(z_2))$$

$$\bar{z}_1 = \bar{h}_1 \frac{\partial h_1}{\partial z_1} = \bar{h}_1 \circ \mathrm{Relu}'(z_1) \quad \text{where } \mathrm{Relu}'(z_{1i}) = \begin{cases} 1 & z_{1i} > 0 \\ 0 & z_{1i} < 0 \end{cases}$$

$$\bar{x} = \bar{y}\frac{\partial y}{\partial x} + \bar{z}_2 \frac{\partial z_2}{\partial x} + \bar{z}_1 \frac{\partial z_1}{\partial x}$$

$$= W_4^T \bar{y} + W_2^T \bar{z}_2 + W_1^T \bar{z}_1$$

2.2.1 forward pass:

$$z = W_1 x = \begin{bmatrix} 1 & 2 & 1 \\ -2 & 1 & 0 \\ 1 & -2 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 8 \\ 1 \\ -6 \end{bmatrix}$$

$$h = \mathrm{Relu}(z) = \begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix}$$

$$y = W_2 h = \begin{bmatrix} -2 & 4 & 1 \\ 1 & -2 & -3 \\ -3 & 4 & 6 \end{bmatrix}\begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -12 \\ 6 \\ -20 \end{bmatrix}$$

Backward pass

$$\bar{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\overline{W_2} = \bar{y} \cdot \left(\frac{\partial y}{\partial W_2}\right)^T$$

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 8 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 1 & 0 \\ 8 & 1 & 0 \\ 8 & 1 & 0 \end{bmatrix} ☆$$

$$\bar{h} = \bar{y}\frac{\partial y}{\partial h}$$

$$= (\bar{y}^T) W_2$$

$$= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} -2 & 4 & 1 \\ 1 & -2 & -3 \\ -3 & 4 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} -4 & 6 & 4 \end{bmatrix}$$

$$\bar{z} = \bar{h} \cdot \mathrm{Relu}'(z)$$

$$= \begin{bmatrix} -4 & 6 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -4 & 6 & 0 \end{bmatrix}$$

$$\overline{W_1} = \bar{z}^T x^T$$

$$= \begin{bmatrix} -4 \\ 6 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -4 & -12 & -4 \\ 6 & 18 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

2.2.2

a) naive: $||W_2||^2 = 8^2+8^2+8^2+3\cdot 1 = 195$

$||W_1||^2 = (-4)^2\cdot 2 + (-12)^2 + (6)^2\cdot 2 + 18^2 = 572$

b) efficient: $||W_2||^2 = ||h||^2 \cdot ||\bar{y}||^2 = (8^2+1^2)\cdot(1^2+1^2+1^2)$

$= 65\cdot 3 = 195$

$||W_1||^2 = ||\bar{z}||^2 ||x||^2 = (16+36)(1+9+1) = 572$

2.2.3:

	Time naive.	Time efficient.	Space naive.	Space efficient
Forward pass	ND^2K	ND^2K	$O(KD^2+DNK)$	$O(NKD+KD^2)$
Backward pass	$2ND^2K$	$D^2N\cdot K$	$O(NKD^2)$	$O(NKD)$
Norm	ND^2K	$NK(2D+1)$	$O(NKD^2)$	$O(NKD)$

Q3

3.2.1

Assuming training converges.

Realizing that the training cost function: $\frac{1}{n}||Xw-t||^2$ is convex.

Thus if gradient descent converges, the resulting weight must be the one yielding smallest weight.

By lecture 1 part 1 notes, when the loss is at minimum, the weight is: $W = (X^TX)^{-1}X^Tt$.

Thus if gradient descent makes training converges, the weight will be $(X^TX)^{-1}X^Tt$.

3.2.2

$\frac{1}{n}||X\hat{w}-t||^2$ plug in $\hat{w}=(X^TX)^{-1}X^Tt$.

Thus $L=\frac{1}{n}||X(X^TX)^{-1}X^Tt-t||^2 \Rightarrow \frac{1}{n}||(X(X^TX)^{-1}X^T-I)t||^2$

Now consider training error:

error is defined as $t-Xw^{*T}$, which equals ε;

Thus $Err=\frac{1}{n}||(X(X^TX)^{-1}X^T-I)(t-Xw^{*T})||^2$

$$=\frac{1}{n}||(X(X^TX)^{-1}X^T-I)\varepsilon||^2$$

To find $E(Err)$: write $||\,||^2$ as a product of vectors;

$[(X(X^TX)^{-1}X^T-I)\varepsilon]^T=\varepsilon^T(X(XX^T)^{-1}X^T-I)$

(since $[(X^TX)^{-1}]^T=[(X^TX)^T]^{-1}$)

realizing that $Err=trace(Err)$ since both is a scalar

$$\Rightarrow \varepsilon^T(X(XX^T)^{-1}X^T-I)(X(X^TX)^{-1}X^T-I)\varepsilon$$

$$=\varepsilon^T(\cancel{(X(XX^T)^{-1}\underline{X^TX}(\underline{X^TX})^{-1}X^T)}-\cancel{(X(XX^T)^{-1}X^T)}-X(X^TX)^{-1}X^T+I)\varepsilon$$

$$=\varepsilon^T(I-X(X^TX)^{-1}X^T)\varepsilon$$

?

3.3.2:

Suppose gradient descent converges. As the loss function is convex, as gradient descent trains the weight, the final loss can only converge when the loss is approaching it's minimum.

Thus by taking the derivative of loss function and setting it to zero, the critical point would be found, and by convexity, it has to be global minimum of loss function;

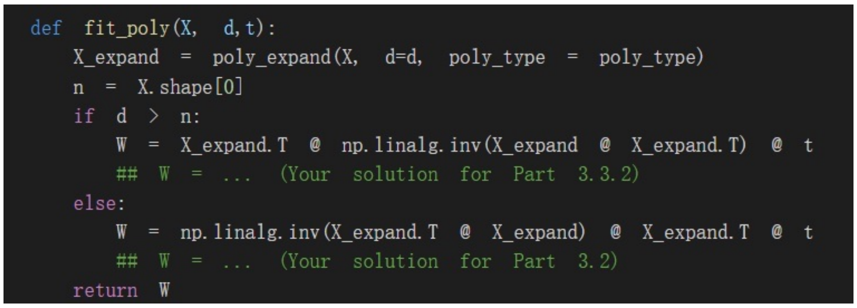
$\frac{\partial}{\partial w}\frac{1}{n}||Xw-t||^2 = 2(Xw-t)X\cdot\frac{1}{n}$; set to zero:

$\frac{2}{n}(Xw-t)X=0$; as X is non-zero, $Xw-t$ must be zero $\Rightarrow$

$X\cdot X^Ta-t=0 \Rightarrow a=(XX^T)^{-1}\cdot t$;

thus $w=X^T(XX^T)^{-1}t$;

3.3.4:



```
def fit_poly(X, d,t):
    X_expand = poly_expand(X, d=d, poly_type = poly_type)
    n = X.shape[0]
    if d > n:
        W = X_expand.T @ np.linalg.inv(X_expand @ X_expand.T) @ t
        ## W = ... (Your solution for Part 3.3.2)
    else:
        W = np.linalg.inv(X_expand.T @ X_expand) @ X_expand.T @ t
        ## W = ... (Your solution for Part 3.2)
    return W
```

By visualizing the loss function (for test), although the loss is dramatically high around 20-90 degrees, the loss drops significantly when degree is greater than 100.

Thus overparameterization doesn't always lead to overfitting.

3.3.2

n X d

$d > n$

d: features;

d w 1

n 1

$(Xw - t)^T X$

$\frac{1}{2n} \sum_{i=1}^{n}$

---

follow hint:

$$XX^T \cdot a - t = 0$$

$$\Rightarrow a = (XX^T)^{-1} t$$

$$\Rightarrow W = X^T a = X^T (XX^T)^{-1} t$$

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

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