1.1

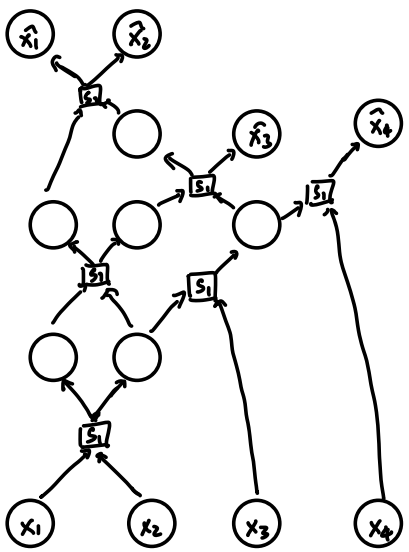
$$W^{(1)} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \qquad b^{(1)} = b^{(2)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\phi^{(1)}(z) = |z|, \quad \phi^{(1)}(z) = z$$

$$W^{(2)} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

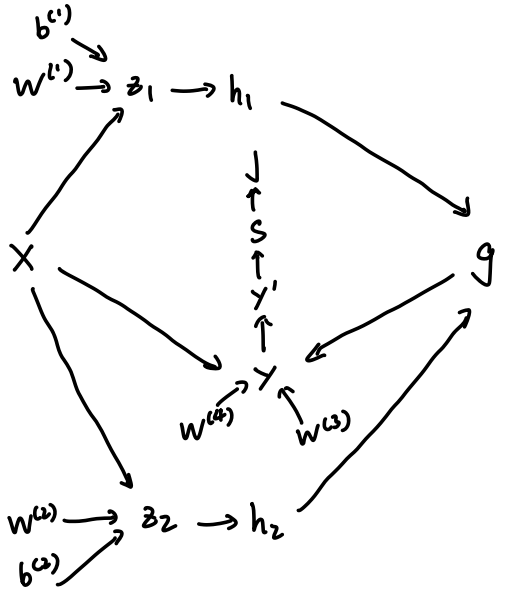
1.2

I will choose the bubble sort. The computation graph looks as follows:



2.1.2

First we have the following computation graph:



So by the backprop algo in lec 2, we have:

$$\bar{J} = 1$$

$$\bar{S} = -1$$

$$\bar{y}' = \bar{S}\frac{\partial S}{\partial y'}^{T} = \begin{bmatrix} 0 \\ \vdots \\ -\frac{1}{y_i'} \\ \vdots \\ 0 \end{bmatrix}$$

here $\frac{\partial S}{\partial y'}^{T}$ is computed as first we compute $\frac{\partial S}{\partial y_i'} = \frac{\partial}{\partial y_i'}\sum_{k=1}^{N} 1(t=k)\log(y_k')$

$$= \sum_{k=1}^{N} 1(t=k)\frac{\partial}{\partial y_i'}\log(y_k')$$

$$= 1(t=i)\frac{1}{y_i'}$$

So only when $i=t$ we have $1(t=i)=1$, and so it will be a one-hot vector with 1 on the t-th position & 0 in the rest positions.

$$\bar{y} = \text{softmax}'(y)\,\bar{y}'$$

softmax'(y) is $N \times N$ symmetric matrix as follow:

$$\begin{bmatrix} y_1'(1-y_1') & -y_1'y_2' & \cdots & -y_1'y_N' \\ -y_2'y_1' & y_2'(1-y_2') & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ -y_N'y_1' & -y_N'y_2' & \cdots & y_N'(1-y_N') \end{bmatrix}$$

$$\bar{g} = W^{(2)^T}\bar{y}$$
$$\overline{W^{(2)}} = \bar{y}g^T$$
$$\overline{W^{(1)}} = \bar{y}x^T$$

For $\bar{h}_1$, we see

$$\frac{\partial g}{\partial h_1} = \begin{bmatrix} h_{21} & 0 & \cdots & 0 \\ 0 & h_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{2N} \end{bmatrix}$$

So $\bar{h}_1 = \frac{\partial g}{\partial h_1}^T \bar{g}$, but it is just the element-wise product, and so

$$\bar{h}_1 = h_2 \circ \bar{g}$$
$$\bar{h}_2 = h_1 \circ \bar{g}$$
$$\bar{z}_2 = \bar{h}_2 \circ \sigma'(z_2) = \bar{h}_2 \circ [\sigma(z_2) \circ (1-\sigma(z_2))]$$
$$\bar{z}_1 = \bar{h}_1 \circ \mathrm{ReLU}'(z_1) = \bar{h}_1 \circ \mathbb{1}(z_1 > 0)$$

For $\sigma'(z_2)$, since for a scalar value s, $\sigma(s) = \sigma(s)(1-\sigma(s))$, we have

$$\sigma'(z_2) = \sigma(z_2) \circ (1 - \sigma(z_2))$$

For $\mathrm{ReLU}'(z_1)$, we have for a scalar value s, $\mathrm{ReLU}'(s) = \begin{cases} 1, & s>0 \\ 0, & s<0 \end{cases}$, and so

$$\mathrm{ReLU}'(z_1) = \mathbb{1}(z_1 > 0)$$

Finally, we have

$$\bar{x} = W^{(1)^T}\bar{z}_1 + W^{(1)^T}\bar{z}_2 + W^{(2)^T}\bar{y}$$

2.2.1

Once we draw the computational graph, we see: $W^{(1)}$ and $W^{(2)}$ feed into $x \to z \to h \to y$

So by the back-propagation, we have $\frac{\partial l}{\partial W^{(1)}} = \overline{W^{(1)}} = (\bar{z}x^T)^T$ & $\frac{\partial l}{\partial W^{(2)}} = \overline{W^{(2)}} = (\bar{y}h^T)^T$

Now, $\bar{z} = \bar{h} \circ ReLU'(z)$, and since $z = \begin{pmatrix} 1 & 2 & 1 \\ -2 & 1 & 0 \\ 1 & -2 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 1 \\ -6 \end{pmatrix}$, we have

$ReLU(z) = \begin{pmatrix} 8 \\ 1 \\ 0 \end{pmatrix}$, and so $ReLU'(z) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$, and $\bar{h} = W^{(2)T}\bar{y} = \begin{pmatrix} -2 & 1 & -3 \\ 4 & -2 & 4 \\ 1 & -3 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

$= \begin{pmatrix} -4 \\ 6 \\ 4 \end{pmatrix}$

So $\bar{z} = \begin{pmatrix} -4 \\ 6 \\ 0 \end{pmatrix}$, and so

$\overline{W^{(1)}} = \left[\begin{pmatrix} -4 \\ 6 \\ 0 \end{pmatrix}(1 \;\; 3 \;\; 1)\right]^T = \begin{pmatrix} -4 & 6 & 0 \\ -12 & 18 & 0 \\ -4 & 6 & 0 \end{pmatrix}$ & $W^{(1)} = \left[\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}(8 \;\; 1 \;\; 0)\right]^T = \begin{pmatrix} 8 & 8 & 8 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

So $\|\overline{W^{(1)}}\|_F^2 = trace\left[\begin{pmatrix} -4 & -12 & -4 \\ 6 & 18 & 6 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} -4 & 6 & 0 \\ -12 & 18 & 0 \\ -4 & 6 & 0 \end{pmatrix}\right] = trace\left[\begin{pmatrix} 176 & -264 & 0 \\ -264 & 396 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right] = 176 + 396 = 572$

$\|\overline{W^{(2)}}\|_F^2 = trace\left[\begin{pmatrix} 8 & 1 & 0 \\ 8 & 1 & 0 \\ 8 & 1 & 0 \end{pmatrix}\begin{pmatrix} 8 & 8 & 8 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}\right] = trace\left[\begin{pmatrix} 65 & 65 & 65 \\ 65 & 65 & 65 \\ 65 & 65 & 65 \end{pmatrix}\right] = 65 \times 3 = 195$

2.2.2

Since $\bar{z} = \begin{pmatrix} -4 \\ 6 \\ 0 \end{pmatrix}$ & $x = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$, we have $\|x\|_2^2 \|\bar{z}\|_2^2 = 52 \times 11 = 572$, which coincides with what we get in 2.2.1. For $\frac{\partial l}{\partial W^{(1)}}$, we have

$$\left\|\frac{\partial l}{\partial W^{(1)}}\right\|_F^2 = trace\left(\frac{\partial l}{\partial W^{(1)}}^T \frac{\partial l}{\partial W^{(1)}}\right) \quad (Def)$$

$$= trace(h\bar{y}^T\bar{y}h^T)$$

$$= trace(\bar{y}^T\bar{y}h^Th) \quad (cyclic\ prop.\ of\ Trace)$$

$$= (\bar{y}^T\bar{y})(h^Th) \quad (Scalar\ multi)$$

$$= \|\bar{y}\|_2^2 \|h\|_2^2 = 3 \times 65 = 195$$

with $\bar{y} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ & $h = ReLU(z) = \begin{pmatrix} 8 \\ 1 \\ 0 \end{pmatrix}$. This is the same as what we computed in 2.2.1 as well.

2.2.3

	T(Naive)	T(Efficient)	M(Naive)	M(Efficient)
Forward Pass	NKD^2	NKD^2	$O(KD^2+NKD)$	$O(NKD)$
Backward Pass	$N(2K-1)D^2$	$N(K-1)D^2$	$O(NKD^2)$	$O(NKD)$
Gradient Norm Computation	NKD^3	$NK(2D+1)$	$O(NKD^2)$	$O(NKD)$

3.2.1

Let $J = \frac{1}{n}\|X\hat{w} - t\|_2^2$, then by gradient descent, we have

$$w := w - \alpha \frac{\partial J}{\partial \hat{w}} \quad (\alpha \text{ is the constant learning rate})$$

and when it converges, we have $\frac{\partial J}{\partial \hat{w}} = 0$. Now,

$$\frac{\partial J}{\partial \hat{w}} = X^T \frac{2}{n}(X\hat{w} - t) \quad \text{(by backpropagation result in lec2)}$$

$$= \frac{2}{n} X^T (X\hat{w} - t) = 0$$

$$\Rightarrow X^T X \hat{w} = X^T t$$

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

since we are assuming $d < n$, $X^T X$ is invertible.

3.2.2

Substituting $\hat{w} = (X^T X)^{-1} X^T t$ into the Error function, we have

$$\frac{1}{n}\|X\hat{w}-t\|_2^2 = \frac{1}{n}\|X(X^TX)^{-1}X^Tt-t\|_2^2$$
$$= \frac{1}{n}\|(X(X^TX)^{-1}X^T-I)t\|_2^2$$

For $(X(X^TX)^{-1}X^T-I)t$, it is equivalent to $(X(X^TX)^{-1}X^T-I)(Xw^*+\varepsilon)$

$$= X(X^TX)^{-1}\cancel{X^TX}w^* - Xw^* + (X(X^TX)^{-1}X^T-I)\varepsilon$$
$$= Xw^* - Xw^* + (X(X^TX)^{-1}X^T-I)\varepsilon$$
$$= (X(X^TX)^{-1}X^T-I)\varepsilon$$

So the error is $\frac{1}{n}\|(X(X^TX)^{-1}X^T-I)\varepsilon\|_2^2$. To get the expectation, we see the error is expanded to be $\frac{1}{n}((X(X^TX)^{-1}X^T-I)\varepsilon)^T((X(X^TX)^{-1}X^T-I)\varepsilon)$

$$= \frac{1}{n}(\varepsilon^TX(X^TX)^{-1}X^T-\varepsilon^T)(X(X^TX)^{-1}X^T\varepsilon-\varepsilon)$$
$$= \frac{1}{n}(\varepsilon^TX(X^TX)^{-1}X^TX(X^TX)^{-1}X^T\varepsilon - \varepsilon^TX(X^TX)^{-1}X^T\varepsilon - \varepsilon^TX(X^TX)^{-1}X^T\varepsilon + \varepsilon^T\varepsilon)$$
$$= \frac{1}{n}(\varepsilon^T\varepsilon - \varepsilon^TX(X^TX)^{-1}X^T\varepsilon).$$
$$= \frac{1}{n}(\text{trace}(\varepsilon^T\varepsilon) - \text{trace}(\varepsilon^TX(X^TX)^{-1}X^T\varepsilon)) \quad (\text{trace of the scalar is scalar})$$

Now, by the cyclic property, $\text{trace}(\varepsilon^TX(X^TX)^{-1}X^T\varepsilon) = \text{trace}(X^T\varepsilon\varepsilon^TX(X^TX)^{-1})$.

Since $E(\varepsilon^T\varepsilon) = E(\sum_{i=1}^n \varepsilon_i^2) = n\sigma^2$, and $E(\varepsilon\varepsilon^T) = E\begin{pmatrix} \varepsilon_1^2 & \varepsilon_1\varepsilon_2 & \cdots & \varepsilon_1\varepsilon_n \\ \vdots & \varepsilon_2^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \varepsilon_n\varepsilon_1 & \cdots & \cdots & \varepsilon_n^2 \end{pmatrix}$

$$= \sigma^2 I \quad (E(\varepsilon_i\varepsilon_j)=0 \text{ when } i\neq j \text{ due to indep})$$

we have $E[\frac{1}{n}(\varepsilon^T\varepsilon - \varepsilon^TX(X^TX)^{-1}X^T\varepsilon)] = \frac{1}{n}[E(\varepsilon^T\varepsilon) - E(\text{trace}(\varepsilon^TX(X^TX)^{-1}X^T\varepsilon))]$

$$= \frac{1}{n}[n\sigma^2 - E(\text{trace}(X^T\varepsilon\varepsilon^TX(X^TX)^{-1}))]$$
$$= \sigma^2 - \frac{1}{n}\text{trace}(X^TE(\varepsilon\varepsilon^T)X(X^TX)^{-1})$$
$$= \sigma^2 - \frac{1}{n}\sigma^2\,\text{trace}(X^TX(X^TX)^{-1})$$
$$= \sigma^2 - \frac{d}{n}\sigma^2 = (1-\frac{d}{n})\sigma^2$$

3.3.2

Back to $\frac{\partial J}{\partial \hat{w}}$, we have $\frac{\partial J}{\partial \hat{w}} = 0 \Rightarrow X^T \frac{2}{n}(X\hat{w} - t) = 0 \Rightarrow X^T(X\hat{w} - t) = 0$. Since now we are assuming that XX^T is invertible, we have

$$XX^T(X\hat{w} - t) = 0 \quad \text{(multiply both sides with 0)}$$

and so by the fact from linear algebra. we must have $X\hat{w} - t = 0$. With the assumption that $\hat{w} = X^T a$, we have

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

$$\Leftrightarrow \quad XX^T a = t$$

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

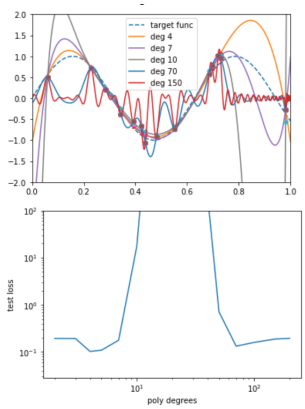
So we have a unique solution for a to solve $X\hat{w} - t = 0$. which means that we have a unique sol for $\hat{w} = X^T a = X^T(XX^T)^{-1} t$

3.3.4

From the loss graph generated by the code, we see the loss stays quite small when poly degrees are < 10 and approximately $> 10^2$, and only increases to very huge and decreases later in the rough range $(10^1, 10^2)$ for poly degrees. Therefore, overparametrization does not always lead to overfitting.

```
# 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)
  n = X.shape[0]
  if d > n:
    W = np.matmul(np.matmul(X_expand.T, linalg.inv(np.matmul(X_expand, X_expand.T))), t)
  else:
    W = np.matmul(np.matmul(linalg.inv(np.matmul(X_expand.T, X_expand)), X_expand.T), t)
  return W
```


target func
deg 4
deg 7
deg 10
deg 70
deg 150
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.2
0.4
0.6
0.8
1.0

test loss
poly degrees