1. Hard-Coding Networks

1.1 Sort Two Numbers

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

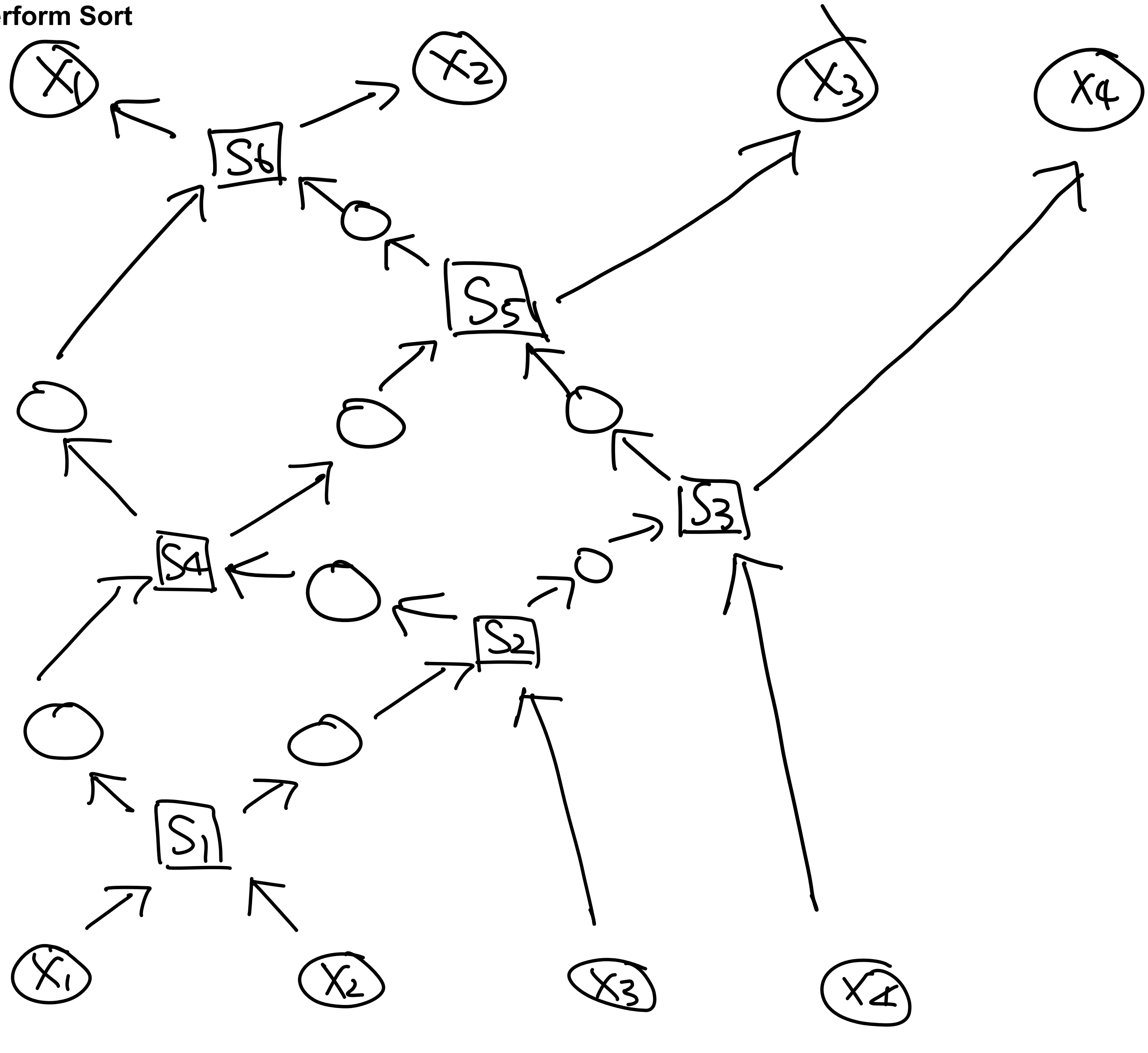
$$\mathbf{W}^{(2)} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$b^{(1)} = b^{(2)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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

$$\phi^{(2)}(z) = z$$

1.2 Perform Sort



2. Backpropagation

2.1.2 Backward Pass

$$\bar{J} = 1$$

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

$$\overline{y^{'}} = \bar{S}\frac{\partial S}{\partial y^{'}} = -\bar{J}\sum_{k=1}^{N}\frac{I(t=k)}{(y_k')}$$

$$\overline{y} = \overline{y^{'}}\frac{\partial y^{'}}{\partial y} = \overline{y^{'}} softmax'(y)$$

$$\overline{g} = \overline{y}\frac{\partial y}{\partial g} = \overline{y}W^{(3)}$$

$$\overline{h_1} = \overline{g}\frac{\partial g}{\partial h_1} = \overline{g} \cdot diag(h_2)$$

$$\overline{h_2} = \overline{g}\frac{\partial g}{\partial h_2} = \overline{g} \cdot diag(h_1)$$

$$\overline{z_1} = \overline{h_1}\frac{\partial h_1}{\partial z_1} = \overline{h_1}ReLU'(z) = \overline{h_1}\begin{bmatrix} I(z_{11} > 0) \\ I(z_{12} > 0) \\ \dots \\ I(z_{1N} > 0) \end{bmatrix}$$

$$\overline{z_2} = \overline{h_2}\frac{\partial h_2}{\partial z_2} = \overline{h_2}[\frac{1}{1+e^{-z}}]' = \overline{h_2}\frac{e^{-z}}{(1+e^{-z})^2}$$

$$\overline{x} = \overline{y}\frac{\partial y}{\partial x} + \overline{z_1}\frac{\partial z_1}{\partial x} + \overline{z_2}\frac{\partial z_2}{\partial x} = \overline{y}W^{(4)} + \overline{z_1}W^{(1)} + \overline{z_2}W^{(2)}$$

2.2.1 Naive Computation

Forward Pass:

$$z = \mathbf{W}^{(1)}x = \begin{bmatrix} 1 & 2 & 1 \\ -2 & 1 & 0 \\ 1 & -2 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ -6 \end{bmatrix}$$

$$h = ReLU(z) = ReLU(\begin{bmatrix} 8 \\ 1 \\ -6 \end{bmatrix}) = \begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix}$$

$$y = \mathbf{W}^{(2)} h = \begin{bmatrix} -2 & 4 & 1 \\ 1 & -2 & -3 \\ -3 & 4 & 6 \end{bmatrix} \begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -12 \\ 6 \\ -20 \end{bmatrix}$$

Backward Pass:

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

$$\bar{h} = \mathbf{W}^{(2)^T} \bar{y} = \begin{bmatrix} -2 & 1 & -3 \\ 4 & -2 & 4 \\ 1 & -3 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 6 \\ 4 \end{bmatrix}$$

$$\bar{z} = \bar{h} \circ \frac{\partial h}{\partial z} = \begin{bmatrix} -4 \\ 6 \\ 4 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ 6 \\ 0 \end{bmatrix}$$

$$\overline{\mathbf{W}^{(1)}} = \bar{z} x^T = \begin{bmatrix} -4 \\ 6 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}^T = \begin{bmatrix} -4 & -12 & -4 \\ 6 & 18 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial J}{\partial \mathbf{W}^{(1)}} = (\bar{z} x^T)^T = \begin{bmatrix} -4 & 6 & 0 \\ -12 & 18 & 0 \\ -4 & 6 & 0 \end{bmatrix}$$

$$\overline{\mathbf{W}^{(2)}} = \bar{y} h^T = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix}^T = \begin{bmatrix} 8 & 1 & 0 \\ 8 & 1 & 0 \\ 8 & 1 & 0 \end{bmatrix}$$

$$\frac{\partial J}{\partial \mathbf{W}^{(2)}} = (\bar{y} h^T)^T = \begin{bmatrix} 8 & 8 & 8 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Square of Frobenius Norm:

$$
\begin{aligned}
&||\overline{\mathbf{W}^{(1)}}||_F^2 \\
&= tr(\overline{\mathbf{W}^{(1)}}^T \overline{\mathbf{W}^{(1)}}) \\
&= tr(\begin{bmatrix} -4 & 6 & 0 \\ -12 & 18 & 0 \\ -4 & 6 & 0 \end{bmatrix} \begin{bmatrix} -4 & -12 & -4 \\ 6 & 18 & 6 \\ 0 & 0 & 0 \end{bmatrix}) \\
&= tr(\begin{bmatrix} 52 & 156 & 52 \\ 156 & 468 & 156 \\ 52 & 156 & 52 \end{bmatrix} \\
&= 52 + 468 + 52 \\
&= 572
\end{aligned}
$$

And,

$$
\begin{aligned}
&||\overline{\mathbf{W}^{(2)}}||_F^2 \\
&= tr(\overline{\mathbf{W}^{(2)}}^T \overline{\mathbf{W}^{(2)}}) \\
&= tr(\begin{bmatrix} 8 & 8 & 8 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 8 & 1 & 0 \\ 8 & 1 & 0 \\ 8 & 1 & 0 \end{bmatrix}) \\
&= tr(\begin{pmatrix} 192 & 24 & 0 \\ 24 & 3 & 0 \\ 0 & 0 & 0 \end{pmatrix}) \\
&= 192 + 3 + 0 \\
&= 195
\end{aligned}
$$

2.2.2 Efficient Computation

As given, $||\frac{\partial J}{\partial \mathbf{W}^{(1)}}||_F^2 = ||x||_2^2||\bar{z}||_2^2$. Thus, plugging in x and $\bar{z}$, we get

$$
\begin{aligned}
||\frac{\partial J}{\partial \mathbf{W}^{(1)}}||_F^2 &= ||x||_2^2||\bar{z}||_2^2 \\
&= (\sqrt{1^2 + 3^2 + 1^2})^2 \cdot (\sqrt{-4^2 + 6^2 + 0^2})^2 \\
&= 11 \left(\sqrt{(-4)^2 + 6^2 + 0^2} \right)^2 \\
&= 11 \cdot\ 52 \\
&= 572
\end{aligned}
$$

Similarly, we have

$$
\begin{aligned}
||\frac{\partial J}{\partial \mathbf{W}^{(2)}}||_F^2 &= trace(\frac{\partial J}{\partial \mathbf{W}^{(2)}}^T \frac{\partial J}{\partial \mathbf{W}^{(2)}}) \\
&= trace(\bar{y}h^T h\bar{y}^T) \\
&= trace(h^T h\bar{y}^T\bar{y}) \\
&= (h^T h)(\bar{y}^T\bar{y}) \\
&= ||h||_2^2||\bar{y}||_2^2 \\
&= (\sqrt{8^2+1^2+0^2})^2 \cdot (\sqrt{1^2+1^2+1^3})^2 \\
&= 65\left(\sqrt{1^2+1^2+1^3}\right)^2 \\
&= 195
\end{aligned}
$$

Notice that the results we get here is the same as what we got in part 2.2.1 😃

2.2.3 Complexity Analysis

	T (Naive)	T (Efficient)	M (Naive)	M (Efficient)
Forward Pass	$O(N^{k-1}D)$	$O(KND)$	$O(KND+2N)$	$O(KND+N)$
Backward Pass	$O(N^{k-1}D^2)$	$O(KND)$	$O(KND+2N)$	$O(KND+N)$
Gradient Norm Computation	$O((K-1)^4)$	$O((K-1)^2)$	$O(2(K-1)^2)$	$O((K-1)^2+(K-1))$

3. Linear Regression

3.1 Deriving the Gradient

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

3.2 Underparameterized Model

3.2.1

As stated in the quesion, let's assume that training converges.
Since we know that gradient must be zero at the optimum (by reading L01a), let's set what we got from 3.1 to 0.

$$\begin{aligned}\frac{2}{n}(X^TX\hat{w} - X^Tt) &= 0 \\ X^TX\hat{w} - X^Tt &= 0 \\ X^TX\hat{w} &= X^Tt \\ (X^TX)^{-1}X^TX\hat{w} &= (X^TX)^{-1}X^Tt \\ \hat{w} &= (X^TX)^{-1}X^Tt\end{aligned}$$

3.2.2

By hint, subsitute $\hat{w} = (X^TX)^{-1}X^Tt$ into $\frac{1}{n}||X\hat{w} - t||_2^2$, we get Error = $\frac{1}{n}||X(X^TX)^{-1}X^Tt - t||_2^2$

Note that $t_i = {w^*}^T x_i + \epsilon_i$. Thus, $t = Xw^* + \epsilon$.

Now, let's simplify $X(X^TX)^{-1}X^Tt - t$

$$\begin{aligned}X(X^TX)^{-1}X^Tt - t &= X(X^TX)^{-1}X^T(Xw^* + \epsilon) - (Xw^* + \epsilon)) \\ &= X(X^TX)^{-1}X^TXw^* + X(X^TX)^{-1}X^T\epsilon - Xw^* - \epsilon \\ &= Xw^* + X(X^TX)^{-1}X^T\epsilon - Xw^* - \epsilon \\ &= X(X^TX)^{-1}X^T\epsilon - \epsilon \\ &= (X(X^TX)^{-1}X^T - I)\epsilon\end{aligned}$$

Therefore, we have

$$\begin{aligned}Error &= \frac{1}{n}||X(X^TX)^{-1}X^Tt - t||_2^2 \\ &= \frac{1}{n}||(X(X^TX)^{-1}X^T - I)\epsilon||_2^2\end{aligned}$$

Then, we are asked to find expexctation of the above training error.
For simplicity, let $A = X(X^TX)^{-1}X^T$. Then, we have $Error = \frac{1}{n}||(A - I)\epsilon||_2^2$.
Then,

$$\begin{aligned}
Error &= \frac{1}{n}||(A-I)\epsilon||_2^2 \\
&= \frac{1}{n}tr([(A-I)\epsilon]^T[(A-I)\epsilon]) \\
&= \frac{1}{n}tr(\epsilon^T(A-I)^T(A-I)\epsilon) \\
&= \frac{1}{n}tr(\epsilon^T\epsilon(A-I)^T(A-I)) && \text{cyclic peoperty of trace: tr(ABC)=tr(ACB)} \\
&= \frac{1}{n}tr(||\epsilon||_2^2||A-I||_F^2) \\
&= \frac{1}{n}\cdot||\epsilon||_2^2\cdot||A-I||_F^2 \\
&= \frac{1}{n}\cdot||\epsilon||_2^2\cdot||(X(X^TX)^{-1}X^T)-I||_F^2
\end{aligned}$$

If so, then

$$\begin{aligned}
E(Error) &= E(\frac{1}{n}\cdot||\epsilon||_2^2\cdot||A-I||_F^2) \\
&= \frac{1}{n}E(||\epsilon||_2^2\cdot||A-I||_F^2) \\
&= \frac{1}{n}E(||\epsilon||_2^2)E(||A-I||_F^2) && \text{Since } \epsilon \text{ is independent} \\
&= \frac{1}{n}\cdot E(\epsilon^T\epsilon)\cdot E(||A-I||_F^2) \\
&= \frac{1}{n}\cdot\sum_{i=1}^n E(\epsilon_i^2)\cdot E(||A-I||_F^2) \\
&= \frac{1}{n}\cdot\sum_{i=1}^n Var(\epsilon_i)\cdot E(||A-I||_F^2) \\
&= \frac{1}{n}\cdot(n\sigma^2)\cdot E(||A-I||_F^2) && \text{Since } Var(\epsilon_i)=\sigma^2 \\
&= \sigma^2E(tr[(A-I)^T(A-I)]) \\
&= \sigma^2tr[E((A-I)^T(A-I))]
\end{aligned}$$

3.3 Overparameterized Model

3.3.2

As hinted, let $\hat{w}=X^Ta$ for some $a\in R^n$
Let's assume convergence of the gradient descent.
Let $\nabla_{\hat{w}}=0$

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

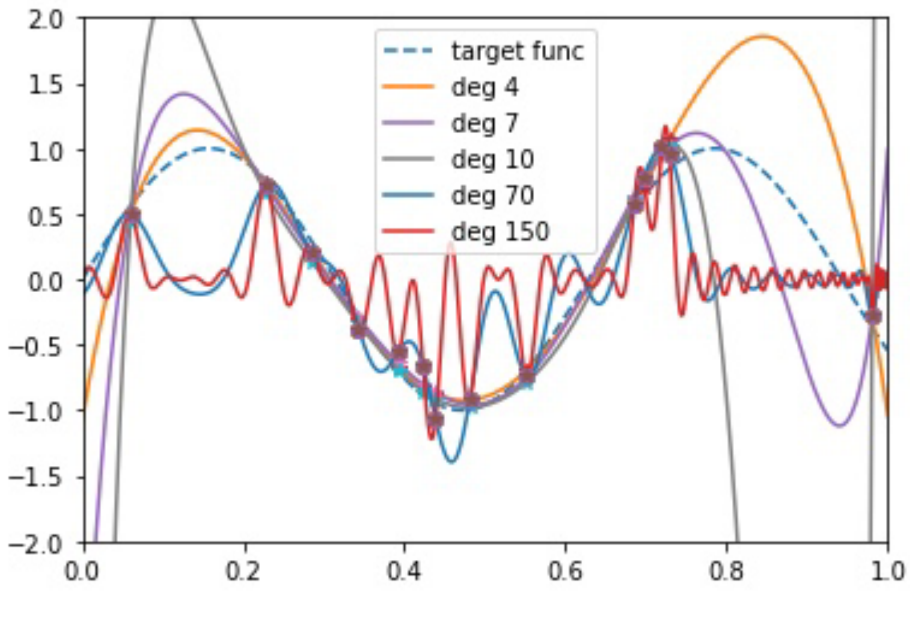
As given by the problem, $n > d$ tells us that XX^T is invertible.
Thus, $(XX^T)^{-1}$ exists.

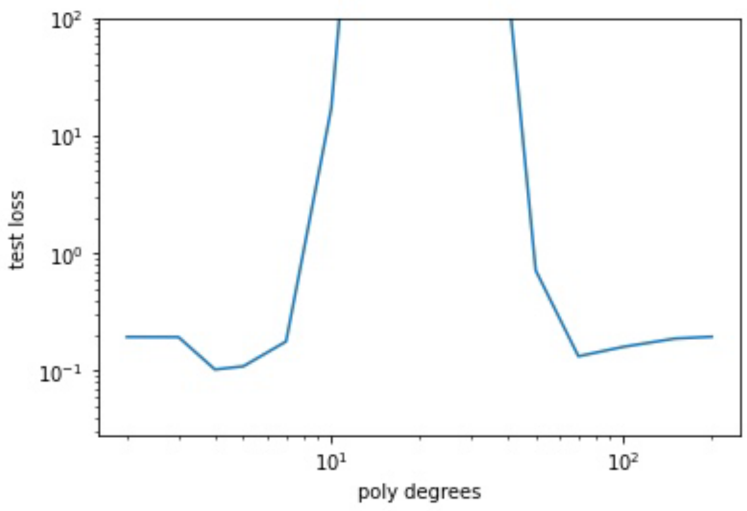
$$X^T(XX^T a - t) = 0$$
$$XX^T(XX^T a - t) = 0$$
$$(XX^T)^{-1}XX^T(XX^T a - t) = 0$$
$$XX^T a - t = 0$$
$$XX^T a = t$$
$$a = (XX^T)^{-1}t$$

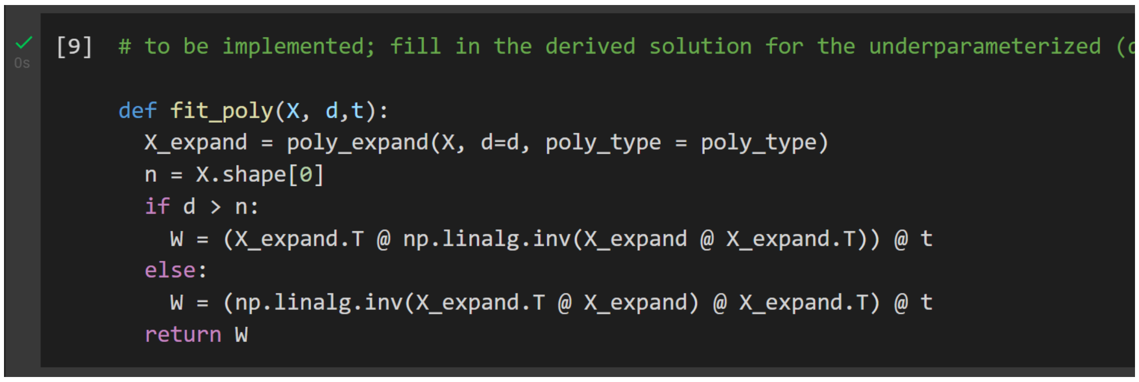
Thus, we have shown that a is unique.

3.3.4

Notice how the visualization shows us that higher degree polynomial does not always lead to larger test error. For example, the test loss incurred when ploy degrees = $10^{1.5}$ is **A LOT GREATER** than the test loss incurred when ploy degree = 10^2.







```
[9]  # to be implemented; fill in the derived solution for the underparameterized (c

     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
       else:
         W = (np.linalg.inv(X_expand.T @ X_expand) @ X_expand.T) @ t
       return W
```