1 Auto Encoders

Using activation function $\sigma(z) = \tanh(z)$, as thus its derivative is $\sigma'(z)$ as the following,

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \qquad \Longrightarrow \qquad \sigma'(z) = \frac{4e^{2x}}{(e^{2z}+1)^2}$$

and the activation $\sigma(z) = z$ at output layer, so that building a network $F(\underline{x})$ to minimize the loss

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left\|\underline{x}^i - F(\underline{x}^i)\right\|^2$$

Calculate ∇L:

Let X_i be inputs, Y_j be hidden layer, Z_i be output layer, for $i = 30,\ j < k$.

Add $X_0 = Y_0 = Z_0 = 1$ to help calculation. Let $w_0, \dots, w_{30}, v_0, \dots, v_k$ be the gradients, where

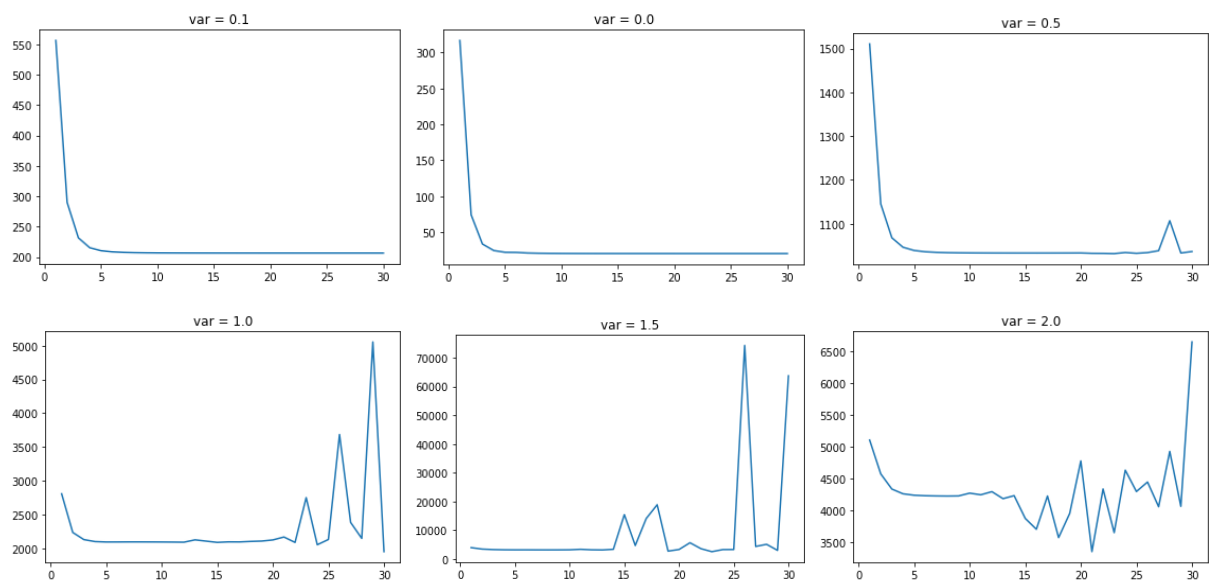
$$Y_j = \tanh\left(\sum_{i=0}^{30} w_i^j x_i\right), \qquad Z_i = \sum_{j=0}^{k} v_j^i y_j$$

$$\frac{\partial L}{\partial v_j^i} = \frac{\partial L}{\partial Z_i}\cdot\frac{\partial Z_i}{\partial v_j^i} = 2(Z_i - X_i)\cdot y_j$$

$$\frac{\partial L}{\partial w_i^j} = \frac{\partial L}{\partial Y_j}\cdot\frac{\partial Y_j}{\partial w_i^j} = \left(\sum_{d=1}^{30}\frac{\partial L}{\partial Z_d}\cdot\frac{\partial Z_d}{\partial Y_j}\right)\cdot\frac{\partial \sigma}{\partial w_i^j} = \left(\sum_{d=1}^{30}\frac{\partial L}{\partial Z_d}\cdot\frac{\partial Z_d}{\partial Y_j}\right)\cdot X_i\cdot\sigma'(\underline{w}^j.\underline{X})$$

For $k \in \{1, 2, \dots, 30\}$, train an auto encoder on the data, and plot the final loss as a function of k

- *Plot the final loss as a function of k for initial $\sigma^2 = 0.1$*
- *How does this change as σ^2 changes $\sigma^2 \in [0,2]$?*



Around $k \geq 7$, the variation become small, which implies that the dimension of this set is around 7.

For variances small, $\sigma^2 \in [.01, .5]$, the losses are decreasing as k increases. And the curve of k is relatively smooth. While for larger σ^2, larger k makes the fluctuation larger, so as the value of losses.

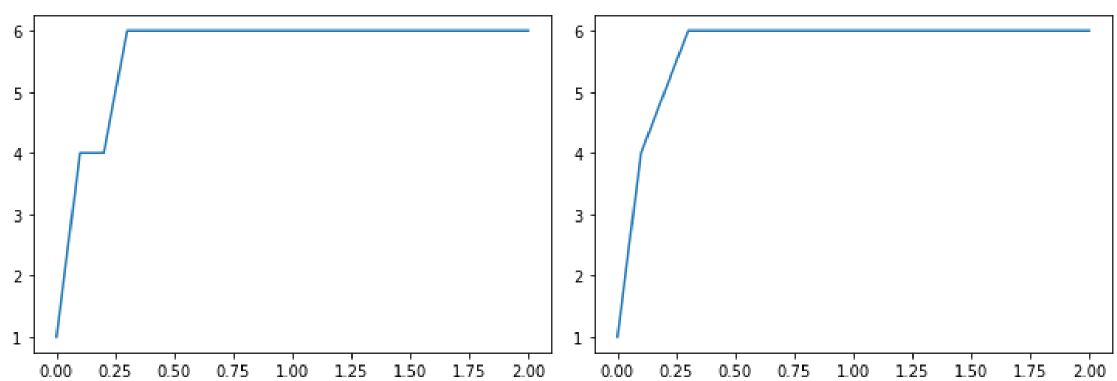
The reason might be the following: the data (X_i) itself is depends on each other. The smaller variance implies a stronger connection, the data points is closer to each other. Then, using the auto-encoder, it is easier to generate the gradients that closer to "True" results. However, the larger variance makes the connections weaker, which also reduces the dependency in each other, and the dimension become larger. Thus, it is possible that make the nodes more depends on the constant w_0, v_0 or other noises that should not be use in calculation. Therefore, larger σ^2 make the auto-encoder less accurate.

2 PCA

For the initial $\sigma^2 = 0.1$, the dimensionality of the data set is 4. This result is robust as get the same result doing again on a different data set.

```
try =  0 , dim =  4
try =  1 , dim =  4
try =  2 , dim =  4
```

For all $\sigma^2 \in [0, 2]$ we choose, there are 30 eigenvalues corresponding to 30 eigenvectors.



For small variance, i.e., $\sigma^2 = 0$, it is reasonable to see the dimensionality as 1 as mainly depends on X_1. While the variance is around $[0.25, 0.5]$, the result is variance between 4 and 5. And for larger variance, the dimensionality is 6.

3 Correlation Graphs

The following is dependency graph, connecting each feature to the other two / three / four features, where the same color represents the "True" dependency structure should be.

Here, using 2 features is robust as getting the same result when doing it again on a different data set. However, when using larger number of other features, it becomes less robust.

	first try		second try								
	2 other features		**2 other features**		**3 other features**			**4 other features**			
	1st	2nd	1st	2nd	1st	2nd	3rd	1st	2nd	3rd	4th
X1	X2	X4	X4	X2	X2	X4	X3	X4	X2	X3	X11
X2	X5	X1	X1	X5	X5	X1	X6	X1	X5	X3	X15
X3	X6	X1	X6	X1	X6	X1	X10	X6	X1	X2	X11
X4	X7	X1	X7	X1	X7	X1	X23	X7	X1	X23	X15
X5	X2	X8	X2	X8	X2	X8	X19	X2	X8	X11	X4
X6	X3	X9	X3	X9	X3	X9	X30	X3	X9	X18	X10
X7	X4	X10	X4	X10	X4	X10	X17	X4	X10	X14	X21
X8	X11	X5	X11	X5	X11	X5	X27	X11	X5	X12	X20
X9	X12	X6	X6	X12	X12	X6	X16	X6	X12	X1	X15
X10	X13	X7	X13	X7	X13	X7	X28	X13	X7	X1	X18
X11	X8	X14	X14	X8	X8	X14	X4	X14	X8	X3	X5
X12	X9	X15	X15	X9	X9	X15	X22	X15	X9	X17	X23
X13	X10	X16	X10	X16	X10	X16	X12	X10	X16	X19	X26
X14	X11	X17	X11	X17	X11	X17	X13	X11	X17	X4	X2
X15	X12	X18	X18	X12	X12	X18	X20	X18	X12	X20	X1
X16	X13	X19	X13	X19	X13	X19	X4	X13	X19	X3	X24
X17	X14	X20	X20	X14	X14	X20	X5	X20	X14	X12	X21
X18	X15	X21	X21	X15	X15	X21	X14	X21	X15	X6	X12
X19	X16	X22	X22	X16	X16	X22	X18	X22	X16	X21	X2
X20	X23	X17	X17	X23	X23	X17	X10	X17	X23	X16	X25
X21	X18	X24	X18	X24	X18	X24	X29	X18	X24	X27	X6
X22	X19	X25	X19	X25	X19	X25	X12	X19	X25	X23	X24
X23	X20	X26	X20	X26	X20	X26	X25	X20	X26	X4	X19
X24	X21	X27	X21	X27	X21	X27	X12	X21	X27	X1	X16
X25	X28	X22	X22	X28	X28	X22	X23	X22	X28	X3	X17
X26	X29	X23	X29	X23	X29	X23	X10	X29	X23	X7	X19
X27	X30	X24	X30	X24	X30	X24	X26	X30	X24	X4	X15
X28	X25	X2	X25	X22	X25	X2	X22	X25	X22	X11	X13
X29	X26	X2	X26	X1	X26	X2	X23	X26	X1	X14	X16
X30	X27	X23	X27	X6	X27	X23	X11	X27	X6	X18	X1

It is possible to reconstruct the true dependency graph, but the linear regression should be more accurate. Here is my trying.

```
X 1
-> X 2
-> X 5
-> X 8
-> X 11
-> X 14
-> X 17
-> X 20
-> X 23
-> X 26
-> X 29
```

```
X 1
-> X 3
-> X 6
-> X 9
-> X 12
-> X 15
-> X 18
-> X 21
-> X 24
-> X 27
-> X 30
```

```
X 1
-> X 11
-> X 14
-> X 17
-> X 20
-> X 23
-> X 26
-> X 29
-> X 16
-> X 13
-> X 10
```

Using 2 features should be enough.

Attempt 1

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
X1	**X2**	X4	X3	X27	X14	X26	X19	X21	X29	X15
X2	**X5**	X1	X6	X28	X29	X9	X7	X18	X23	X13
X3	X6	X1	X10	X9	X18	X23	X4	X11	X7	X5
X4	X7	X1	X23	X24	X2	X3	X28	X18	X25	X9
X5	X2	**X8**	X19	X14	X17	X20	X3	X29	X23	X25
X6	X3	X9	X30	X2	X26	X8	X22	X21	X28	X24
X7	X4	X10	X17	X16	X9	X13	X24	X6	X29	X19
X8	**X11**	X5	X27	X2	X26	X10	X6	X15	X19	X9
X9	X12	X6	X16	X17	X25	X4	X3	X10	X20	X19
X10	X13	X7	X28	X20	X25	X30	X27	X26	X11	X15
X11	X8	**X14**	X4	X20	X5	X17	X7	X26	X1	X25
X12	X9	X15	X22	X23	X13	X28	X24	X14	X21	X8
X13	X10	X16	X12	X14	X29	X1	X2	X23	X19	X17
X14	X11	**X17**	X13	X18	X30	X2	X3	X27	X22	X16
X15	X12	X18	X20	X7	X17	X27	X29	X1	X11	X3
X16	X13	X19	X4	X25	X22	X3	X20	X27	X5	X6
X17	X14	**X20**	X5	X7	X15	X6	X28	X9	X11	X25
X18	X15	X21	X14	X28	X19	X22	X9	X10	X26	X4
X19	X16	X22	X18	X9	X23	X20	X5	X28	X26	X24
X20	**X23**	X17	X10	X14	X24	X16	X3	X18	X11	X6
X21	X18	X24	X29	X10	X27	X7	X5	X2	X23	X20
X22	X19	X25	X12	X21	X28	X29	X2	X17	X14	X16
X23	X20	**X26**	X25	X30	X29	X17	X21	X15	X1	X2
X24	X21	X27	X12	X11	X25	X23	X8	X22	X6	X16
X25	X28	X22	X23	X1	X24	X9	X5	X8	X26	X16
X26	**X29**	X23	X10	X11	X18	X12	X3	X27	X24	X8
X27	X30	X24	X26	X2	X10	X20	X15	X17	X3	X18
X28	X25	X2	X22	X21	X17	X10	X29	X18	X20	X4
X29	X26	X2	X23	X22	X15	X13	X1	X28	X30	X5
X30	X27	X23	X11	X6	X12	X10	X21	X29	X24	X25

For different variance, my trying is the following

```
start var =  0.01

X 1
-> X 27
-> X 22
-> X 21
-> X 4
-> X 10
-> X 17
-> X 16
-> X 29
-> X 12
-> X 15

X 1
-> X 10
-> X 17
-> X 16
-> X 29
-> X 22
-> X 21
-> X 4
-> X 6
-> X 15
-> X 24

X 1
-> X 17
-> X 16
-> X 29
-> X 22
-> X 21
-> X 4
-> X 10
-> X 13
-> X 9
-> X 19

start var =  0.51
X 1
-> X 16
-> X 13
-> X 21
-> X 26
-> X 15
-> X 12
-> X 18
-> X 9
-> X 6
-> X 7
X 1
-> X 27
-> X 16
-> X 13
```

```
-> X 21
-> X 26
-> X 15
-> X 12
-> X 18
-> X 9

-> X 6
X 1
-> X 18
-> X 9
-> X 6
-> X 7

-> X 10
-> X 4
-> X 3
-> X 17
-> X 11
-> X 20

start var =  1.01
X 1
-> X 10
-> X 4
-> X 25
-> X 28
-> X 22
-> X 19
-> X 16
-> X 13
-> X 3
-> X 7

X 1
-> X 20
-> X 17
-> X 4
-> X 25
-> X 28
-> X 22
-> X 19
-> X 16
-> X 13
-> X 3

X 1
-> X 6
-> X 14
-> X 17
-> X 20
-> X 22
-> X 19
-> X 16
-> X 4
-> X 25
-> X 28

start var =  1.51
X 1
-> X 3
-> X 6
-> X 25
-> X 22
-> X 19
-> X 17
-> X 20
-> X 23
-> X 24
-> X 9

X 1
-> X 4
-> X 3
-> X 6
-> X 25
-> X 22
-> X 19
-> X 17
-> X 20
-> X 23
X 1

-> X 2
-> X 5
-> X 3
-> X 6
-> X 25
-> X 22
-> X 19
-> X 17
-> X 20
-> X 23

start var =  2.01
X 1
-> X 5
-> X 8
-> X 14
-> X 11
-> X 22
-> X 25
-> X 28
-> X 17
-> X 20
-> X 23

X 1
-> X 15
-> X 12
-> X 6
-> X 9
-> X 25
-> X 28
-> X 17
-> X 20
-> X 23
-> X 29

X 1
-> X 3
-> X 17
-> X 20
-> X 23
-> X 29
-> X 26
-> X 8
-> X 14
-> X 11
-> X 22
```