Overfitting & Dropout

Dr. Atul Nautiyal
ADAPT Centre, TCD

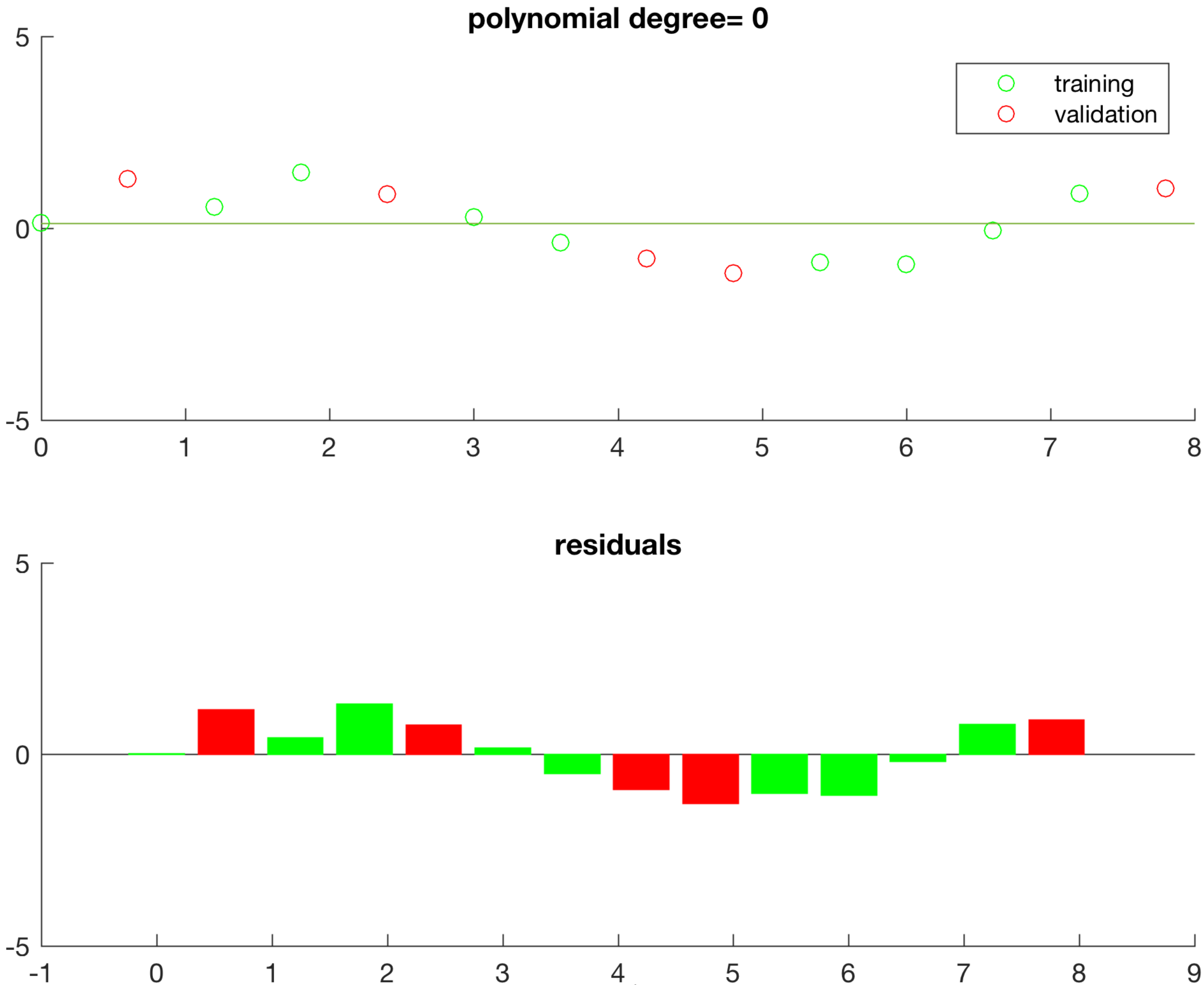
Machine Learning

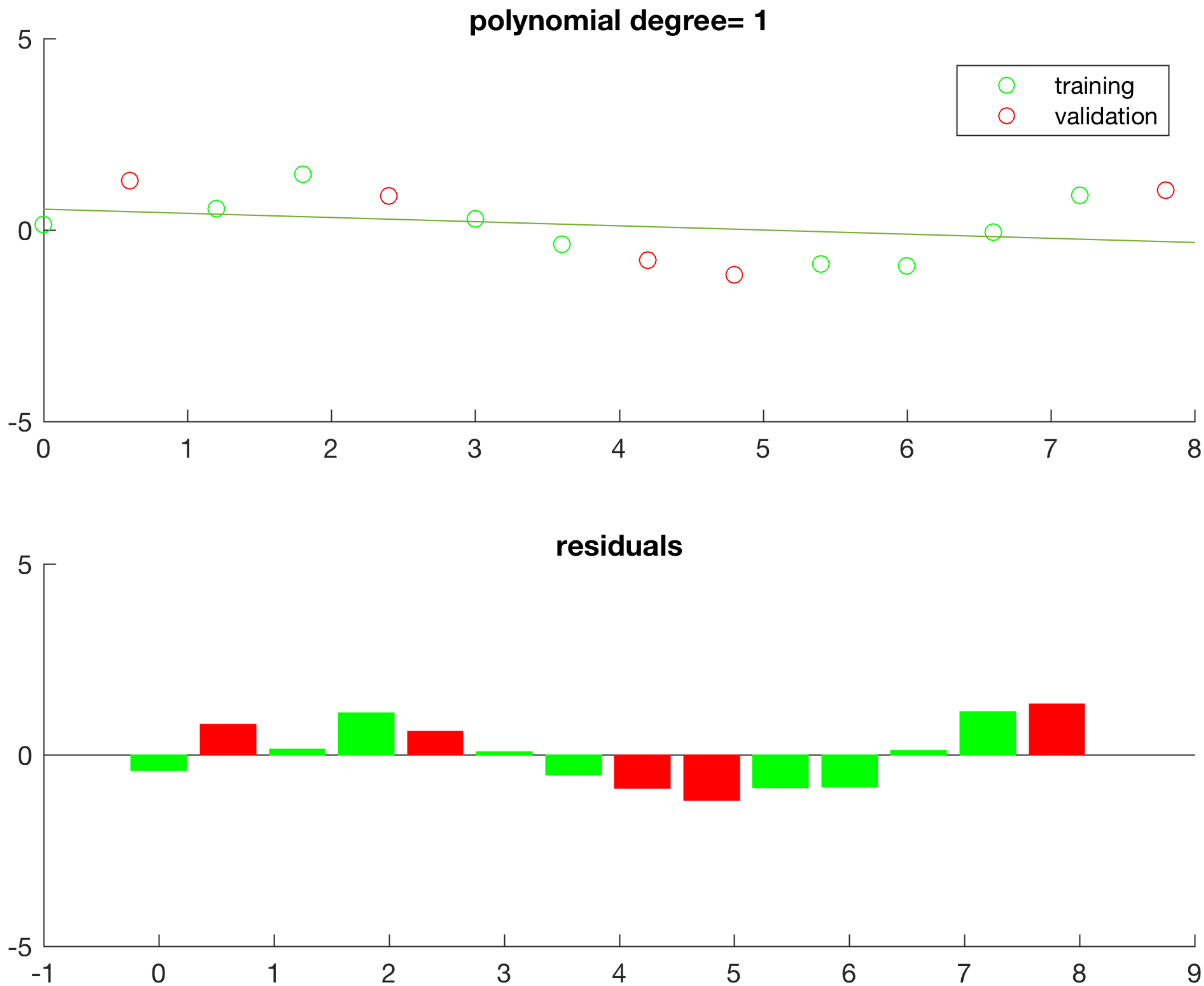
- Goal: Learn relationships between input and output.

Overfitting

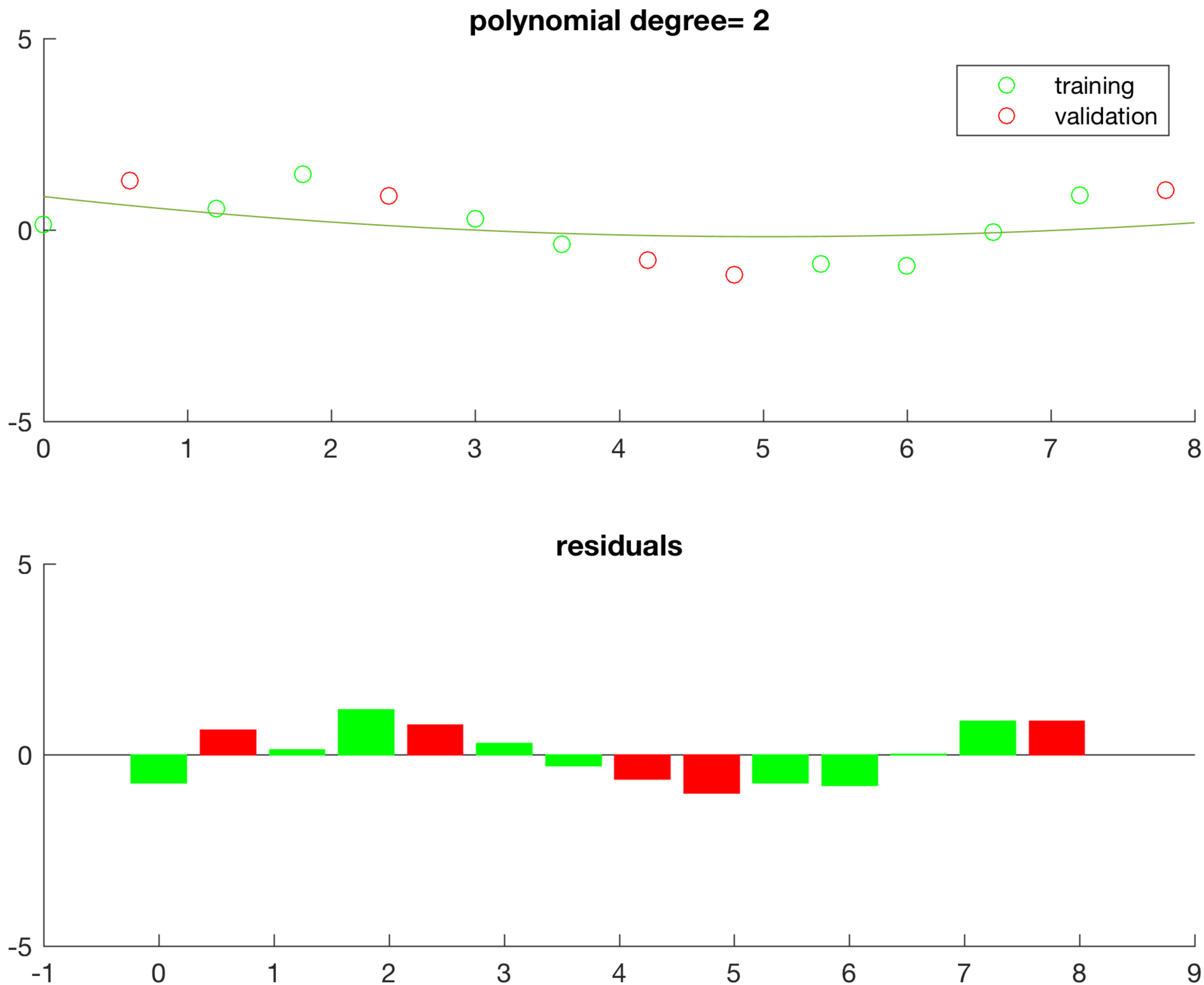
- Neural network with multiple non-linear hidden layers can learn complicated relationships between input and output. However, with limited training data, these networks may learn many false relationships in training data which are not present in validation or test data.
- As a result, networks give very high accuracy score on training data but fail miserably on validation or test data.
- Some counter measures are early stopping, regularisation, dropout, etc.

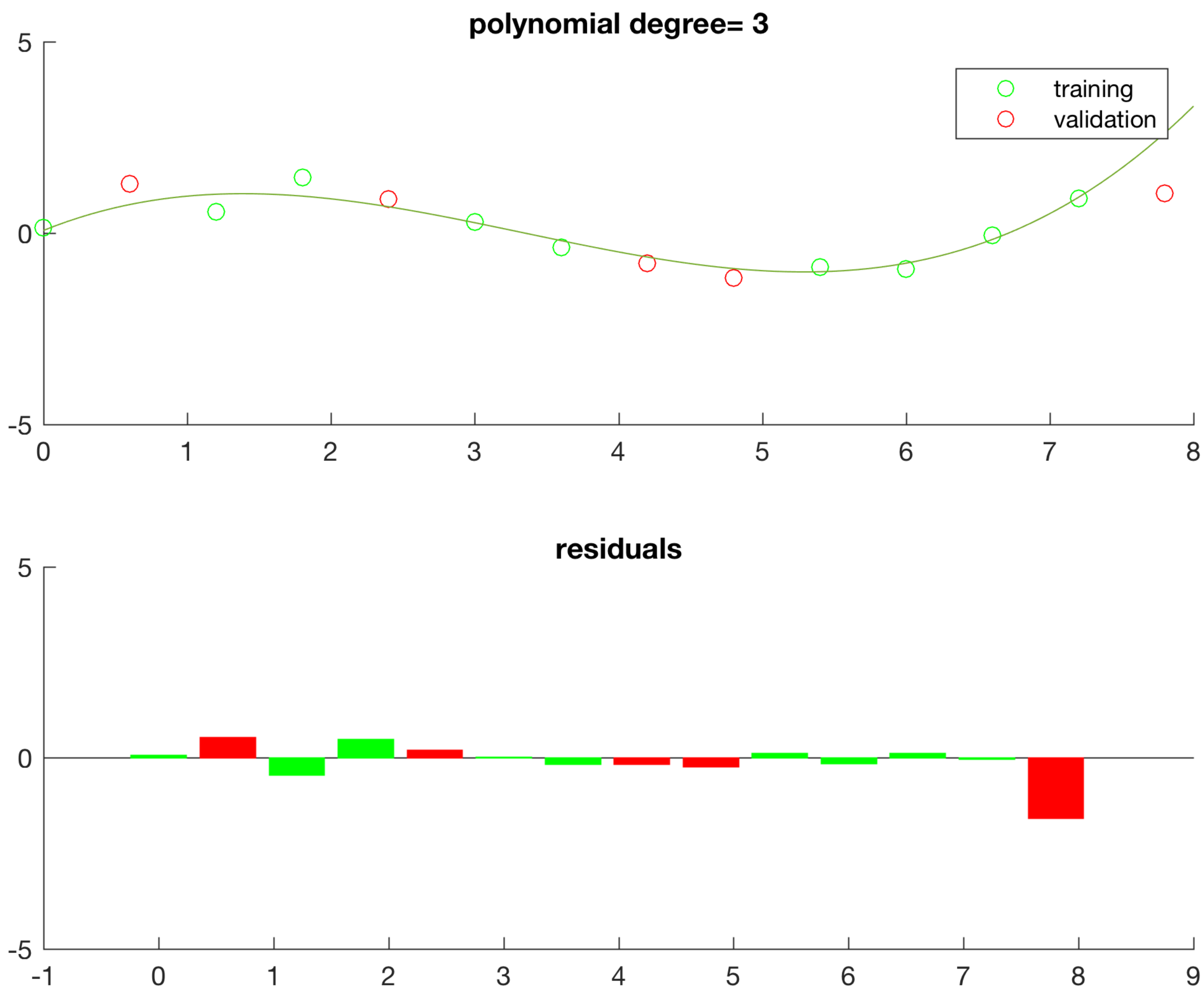
Overfitting example

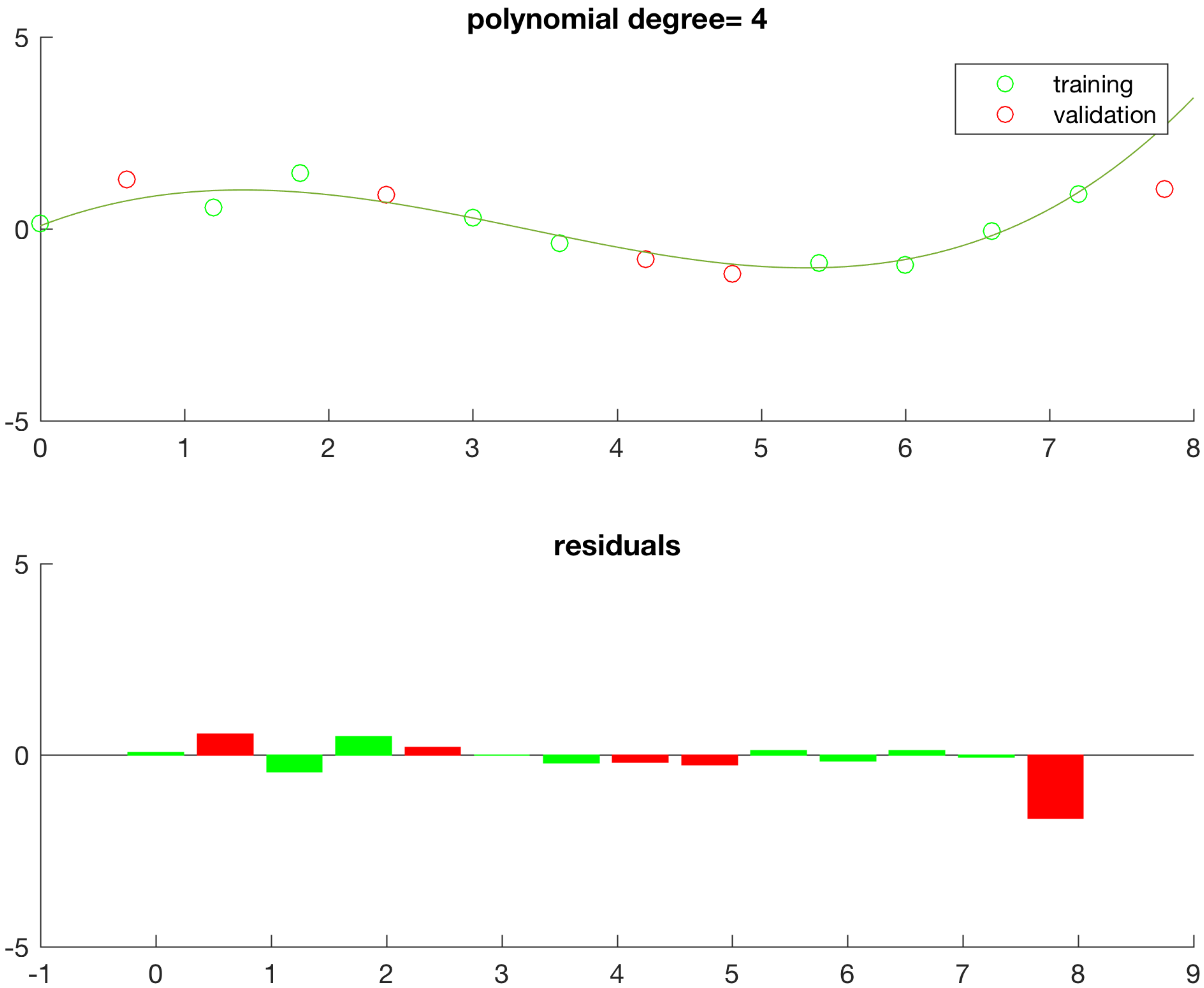




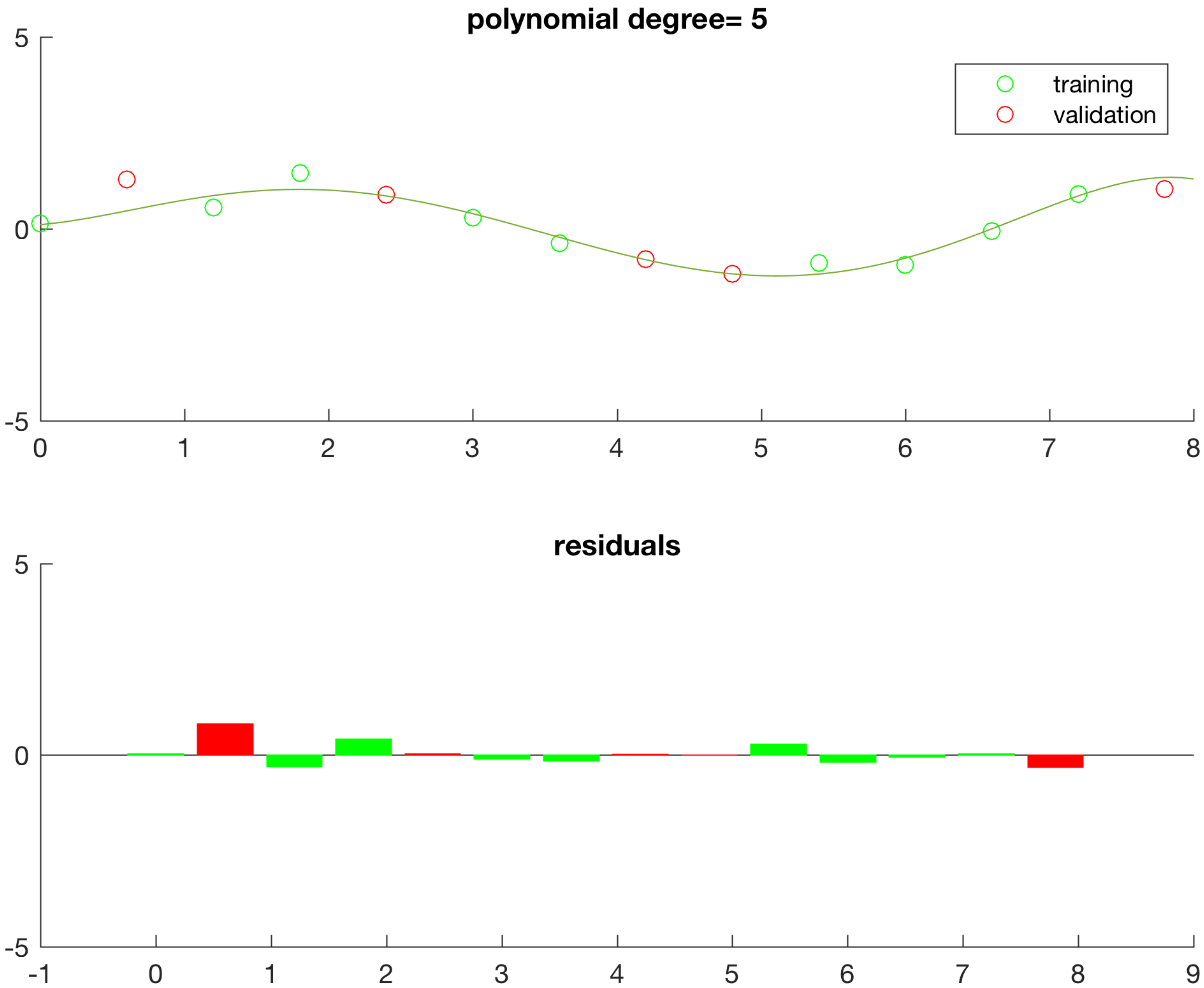
polynomial degree= 1
training
validation
residuals


polynomial degree= 2
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8
residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9

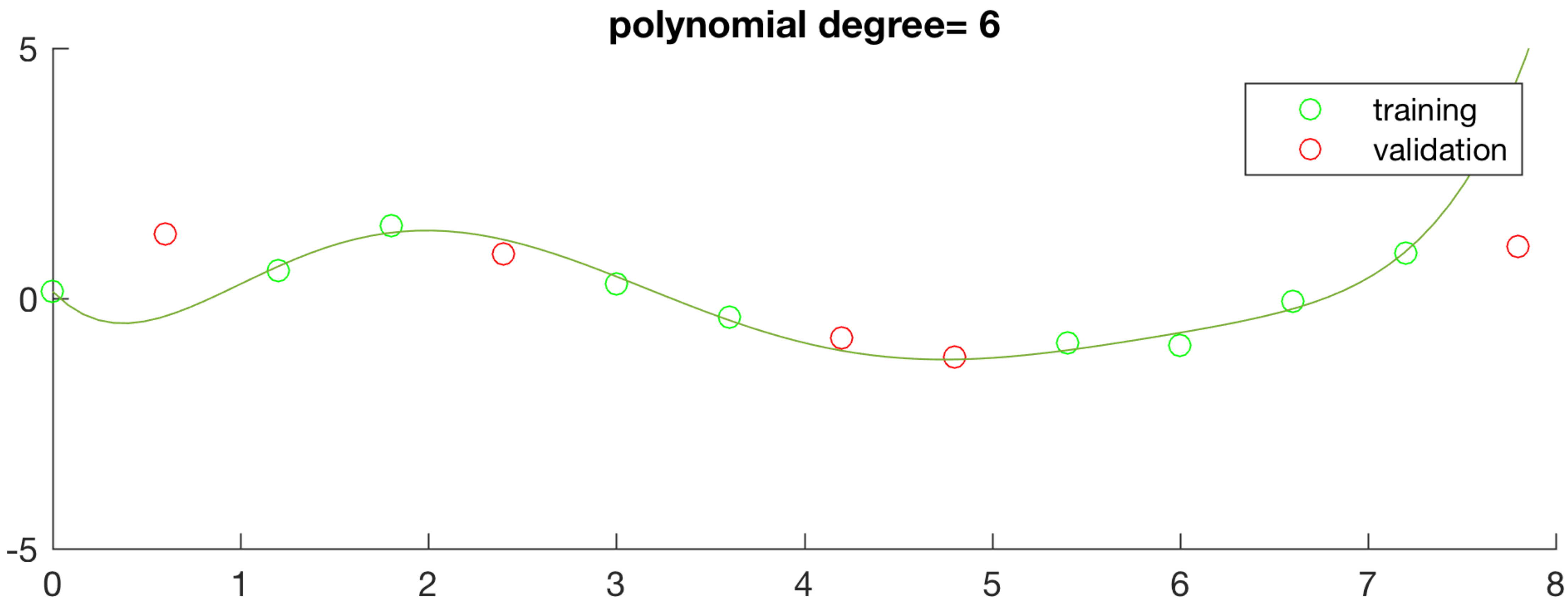

polynomial degree= 3
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8
residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9



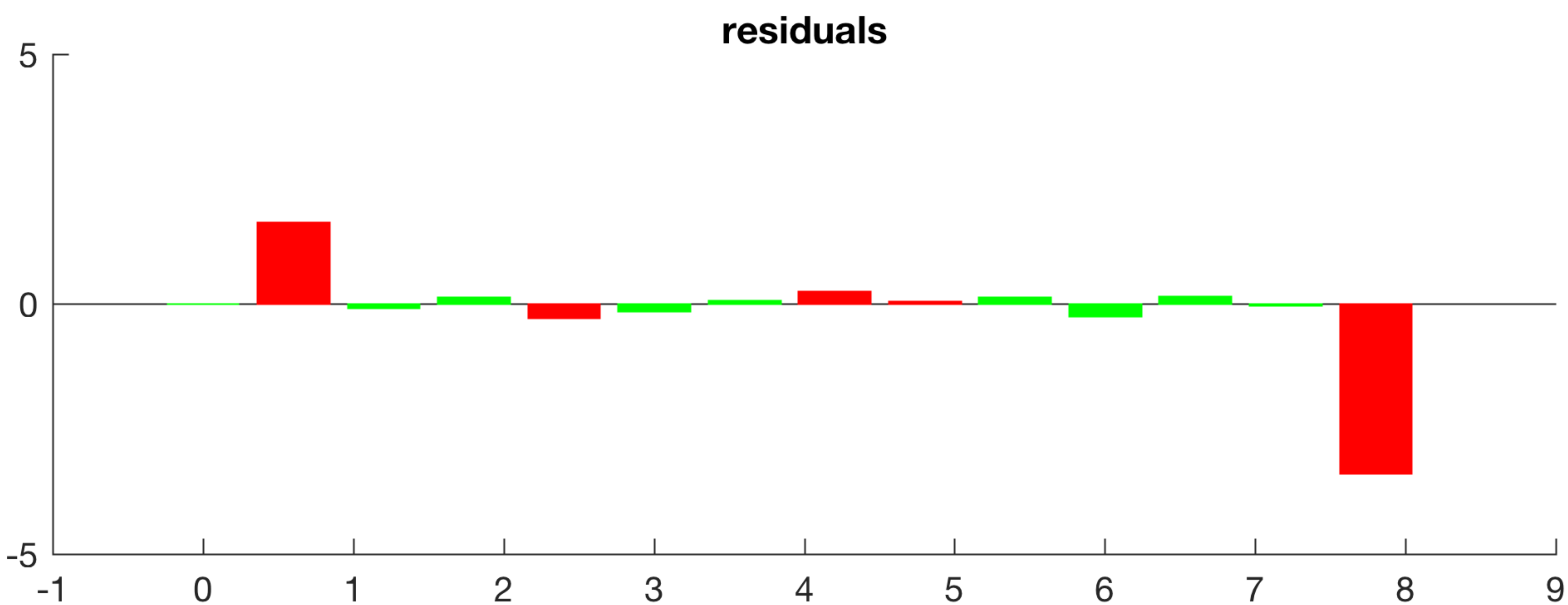
polynomial degree= 4
training
validation
residuals



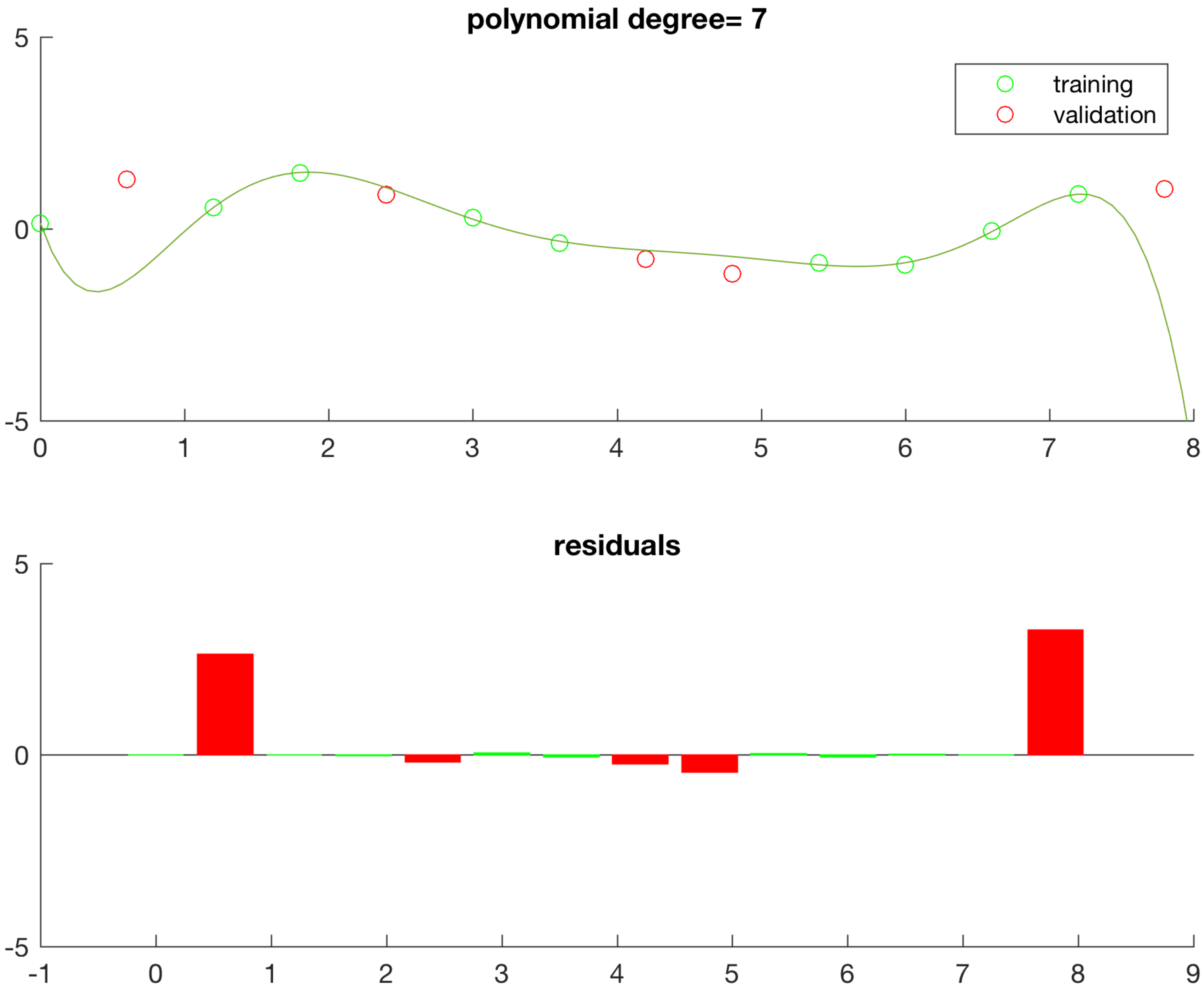
polynomial degree= 5
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8
residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9

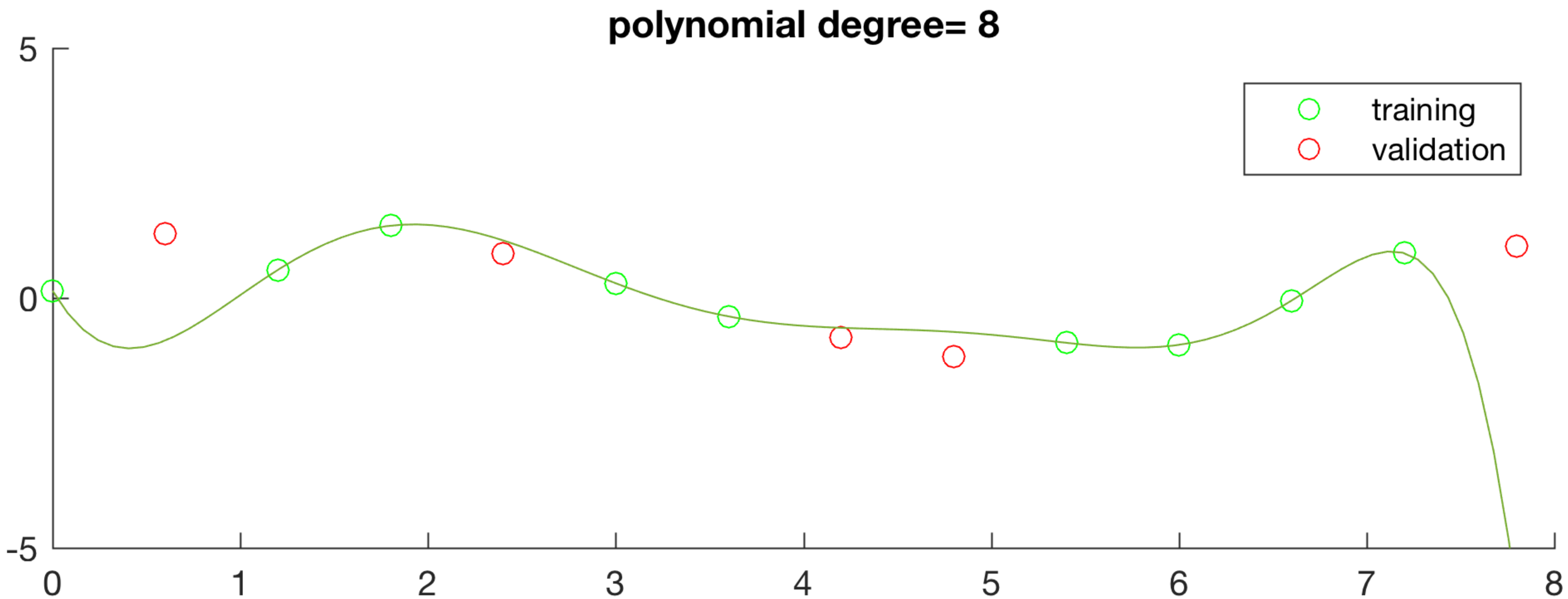


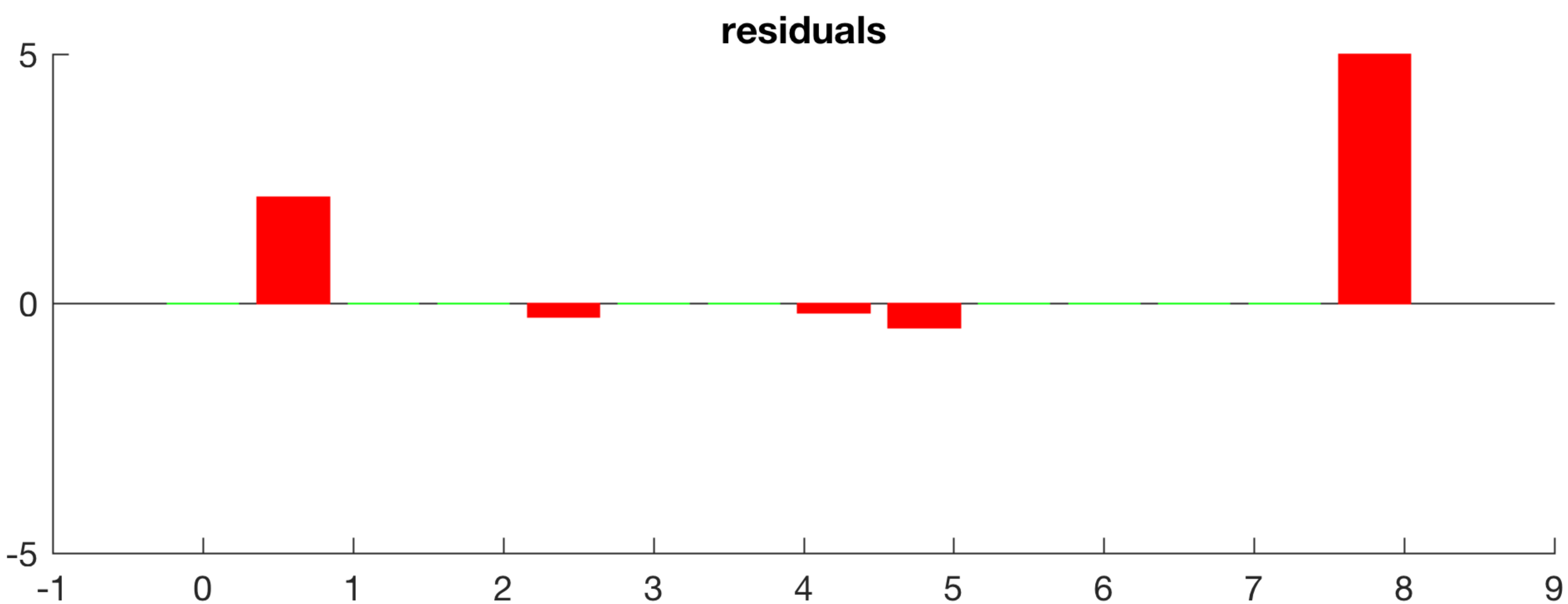
polynomial degree= 6
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8



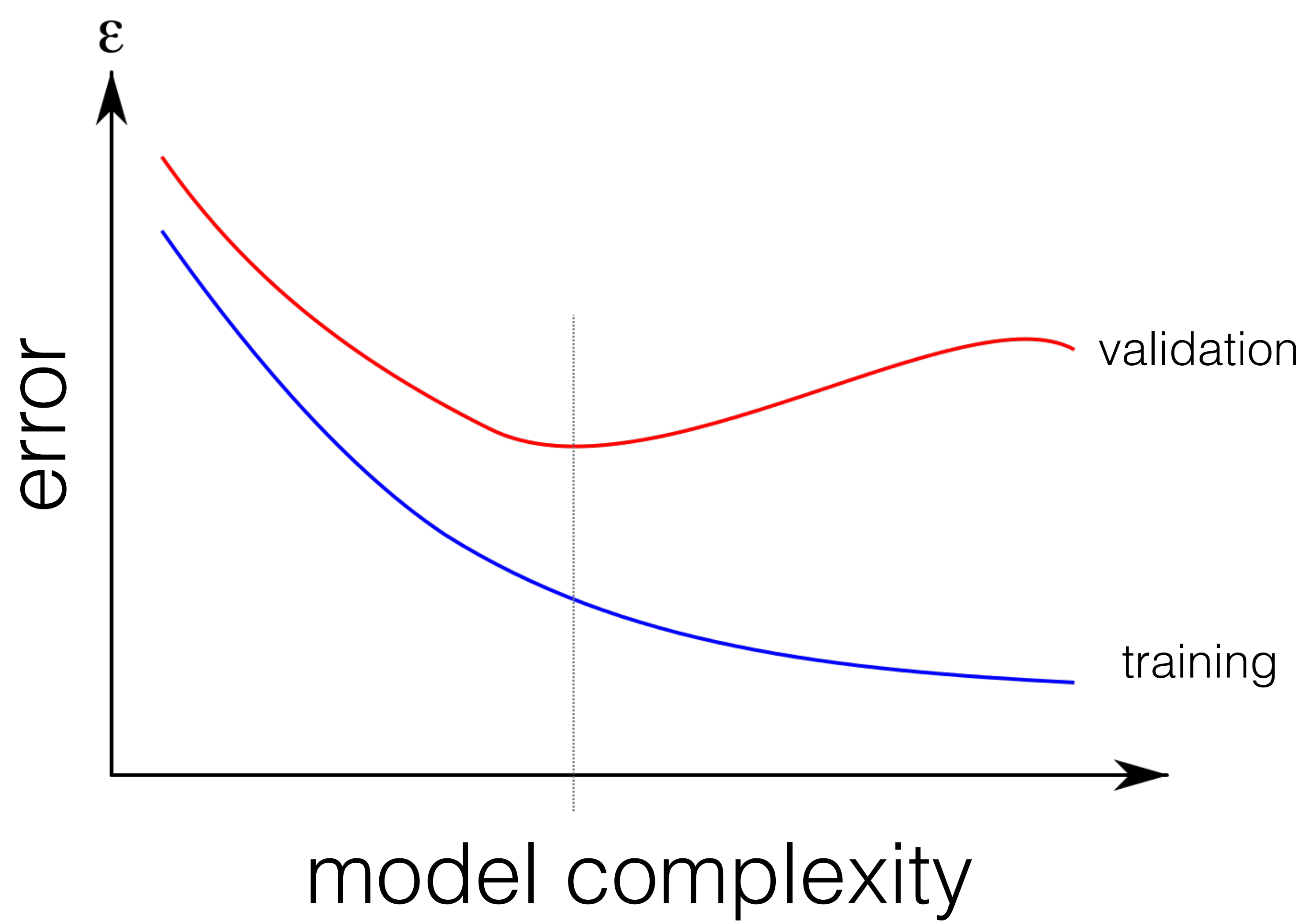
residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9


polynomial degree= 7
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8
residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9


polynomial degree= 8
training
validation
5
0
-5
0
1
2
3
4
5
6
7
8


residuals
5
0
-5
-1
0
1
2
3
4
5
6
7
8
9

Error curve: Training vs Validation



Learned weights

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
M_0	0.1262								
M_1	-0.1087	0.5464							
M_2	0.0412	-0.4154	0.8748						
M_3	0.0688	-0.6893	1.5153	0.0744					
M_4	0.0007	0.0583	-0.6414	1.4482	0.0845				
M_5	-0.0048	0.0876	-0.4864	0.7259	0.3182	0.1147			
M_6	0.0044	-0.1041	0.9445	-3.9410	7.0977	-3.8522	0.1424		
M_7	-0.0029	0.0788	-0.8713	4.9069	-14.6290	21.2075	-10.9026	0.1490	
M_8	-0.0005	0.0123	-0.1098	0.3725	0.2513	-4.8297	10.5808	-6.3613	0.1485

Table: Learned weights for regression example

- Weights increase with model complexity.

L1 & L2 Regularisation

- Penalise large weights using an additional term with the error function.

$$\widetilde{E}(w) = \frac{1}{2}\sum_{n=1}^{N}\{f(x_n, w) - y\}^2 + \frac{\lambda}{2}||w||^l$$

where l = {1,2} and λ term controls the relative importance of regularisation term.

- L1 regularisation (l = 1)
- L2 regularisation (l = 2)
- L2 regularisation also known as weight decay.

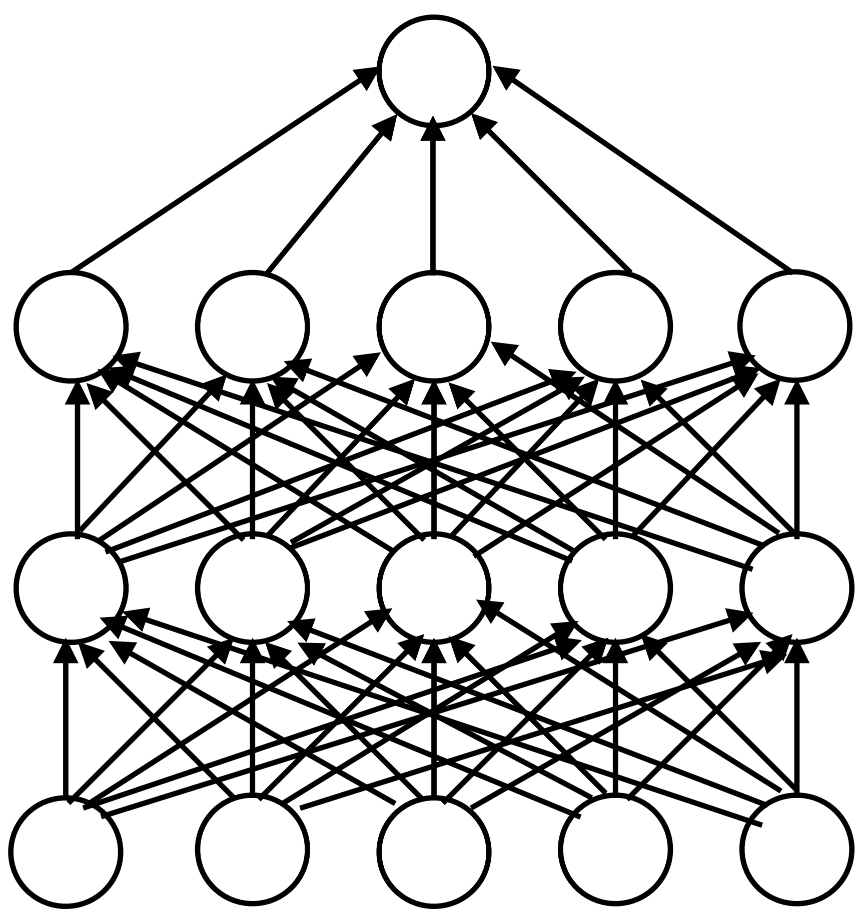
Dropout

- In a neural network, the derivative received by each unit pushes it in a direction so that the final loss function is reduced given what all other units are doing.
- Units may change in a way to fix up mistakes of the other units.
- Leads to complex co-adaptations which terns to overfitting.
- Dropout tries to break this co-adaptions by randomly dropping the hidden units at training time.
- Therefore, units cannot rely on other units to correct their mistakes.

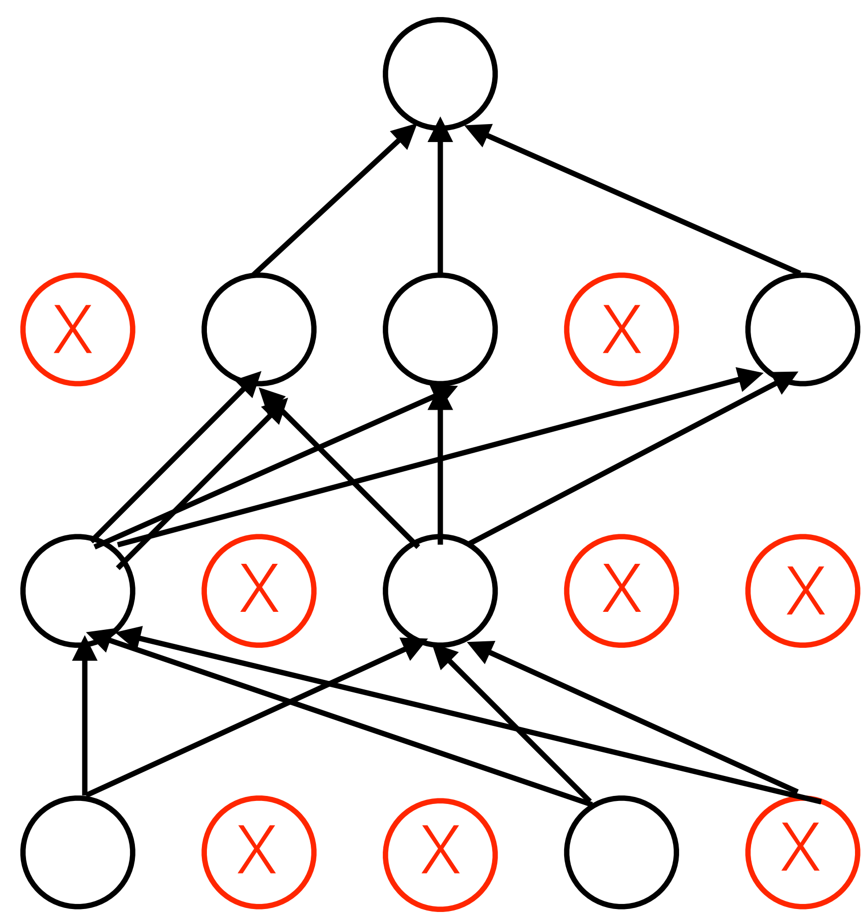
Dropout[1]

- Dropout refers to temporarily removing incoming and outgoing connections of a hidden or input unit.

- Motivated from theory of the role of sex in evolution.



Standard Neural network



Neural network with dropout

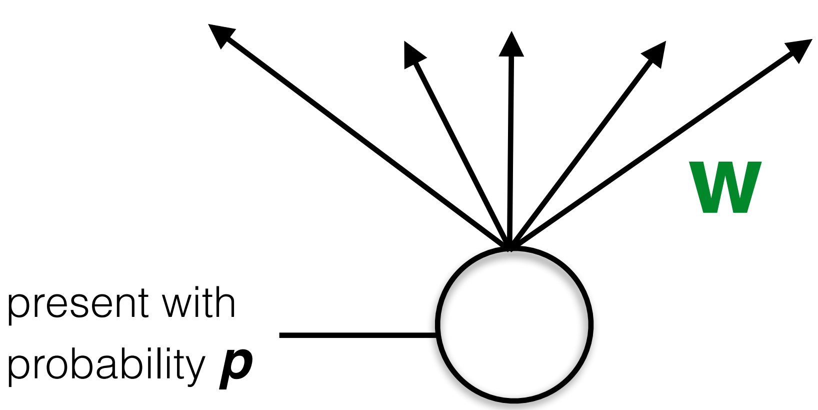
1. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (January 2014), 1929-1958.

Dropout

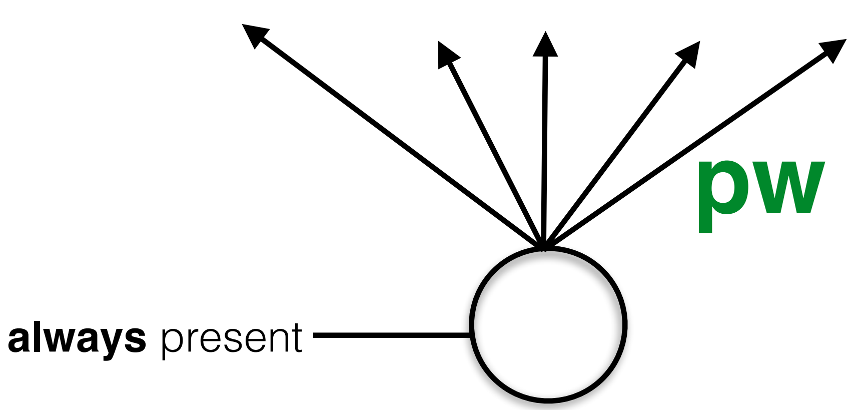
- Create a thinned network with dropout.
- A neural network with ***n*** units can be seen as a collection of 2^n thinned networks.
- Each thin network rarely gets trained.
- Networks share the weights so no increase in total number of parameters.
- Each unit is retained with a fixed probability ***p*** independent of the other units.

Dropout

- At test time, 2^n thinned networks with shared weight are combined to create a single fully connected network.
- Outgoing weights of a unit are multiplied by ***p,*** if the unit was retained with probability ***p*** during the training time.



A unit at **training** time



A unit at **test** time

Dropout: few other points

- Robust features
- Increased training time
- One additional hyper parameter ***p.*** Generally, dropping out 20% of the input units and 50% of the hidden units are often found to be optimal.
- Use with other regularisation methods.

Thanks