Regularization and Bias/Variance

We've seen how regularization can help prevent over-fitting. But how does it affect the bias and variances of a learning algorithm?

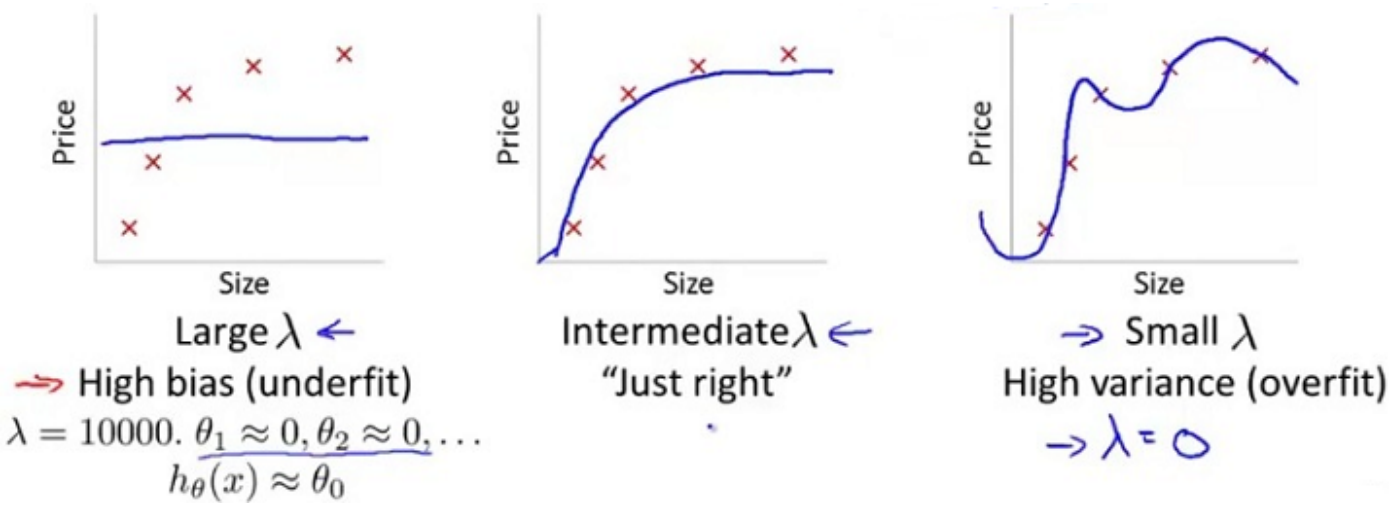
Linear Regression with regularization

Suppose we're fitting a high order polynomial, but to prevent over-fitting we're going to use regularization.

$$\text{Model: } h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

Let's consider three cases. The first is the case of the very large value of the regularization parameter lambda, such as if lambda were equal to 10,000. Some huge value.



All of the parameters, $\theta_1, \theta_2, \theta_3, \ldots, \theta_n$ would be heavily penalized and so we end up with most of the parameter values being closer to zero, and the hypothesis $h_\theta(x)$ will be just equal or approximately equal to θ_0, and so this hypothesis has **high bias** and it badly under-fits the data set (is just not a very good model for the data set).

- **Large λ = high bias -> Under fitting data (all θ values are heavily penalized)**.
  - Most parameters end up being close to zero
  - Hypothesis ends up being close to 0

If we have a very small value of lambda, such as if lambda were equal to zero. In that case, given that we're fitting a high order polynomial, this is a usual **over-fitting setting**. In that case, given that we're fitting a high-order polynomial, **without regularization** or with very minimal regularization, we end up with our usual **high-variance**, over fitting setting, basically if λ is equal to zero, we're just fitting with our regularization, so that over fits the hypothesis $h_\theta(x)$.

- **Small λ = high variance -> Get overfitting ($\lambda = 0$)**

Finally, it's only if we have some intermediate value of λ that is neither too large not too small that we end up with parameters θ that give us a reasonable fit to the data.

- λ = **intermediate -> Only this values gives the fitting which is reasonable**

Choosing the regularization parameter λ

So, how can we automatically choose a good value for the regularization parameter? To do this we define another function $J_{train}(\theta)$ which is the optimization function without the regularization term (average squared errors).

- $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

And similarly we're then also going to define the **cross validation set error** and the **test set error** (without the regularization term).

- $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$
- $J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$

Choosing λ

- Have a set or range of λ values to use (Often increment by factors of 2):

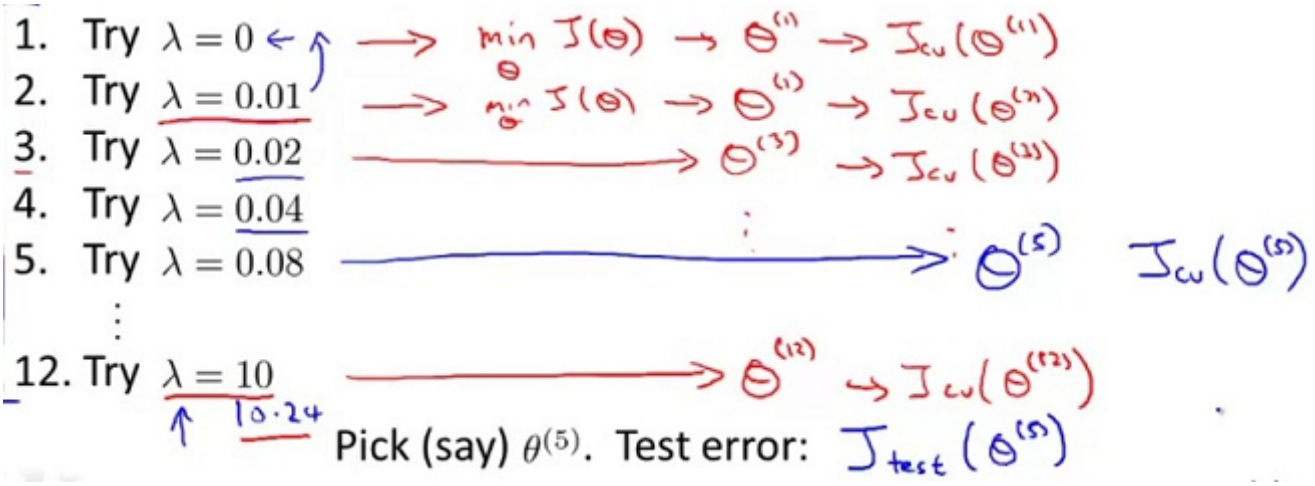
1. Try $\lambda = 0$
2. Try $\lambda = 0.01$
3. Try $\lambda = 0.02$
4. Try $\lambda = 0.04$
5. Try $\lambda = 0.08$

   $\vdots$

12. Try $\lambda = 10$

So, this give us a number of models which have different λ. Given each of these 12 models, what we can do is take each the model and minimize the cost function $J(\theta)$ of each model and this will generate some parameter vector and now we have a set of parameter vectors corresponding to models with different λ values.

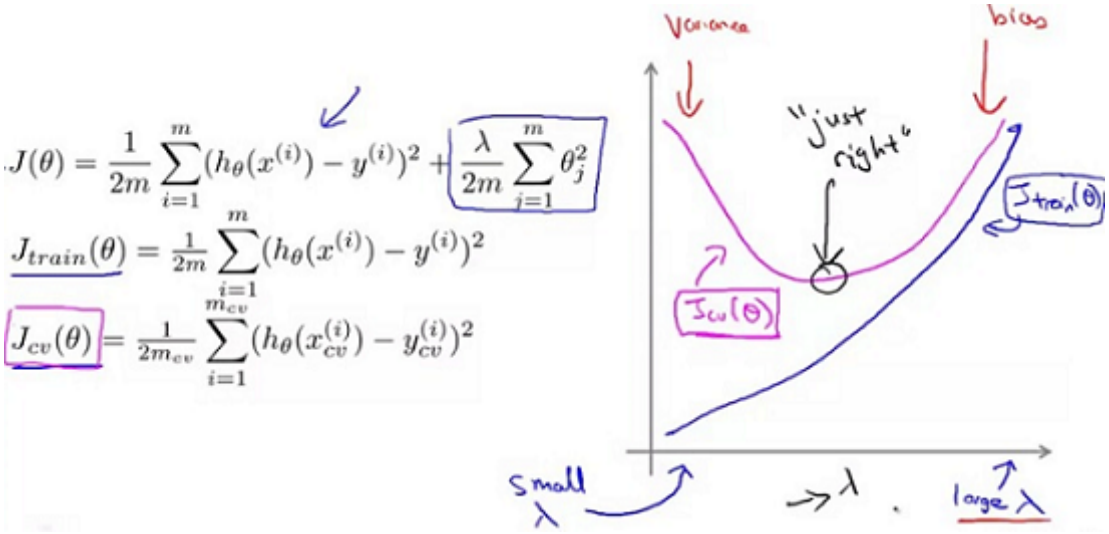
Next we can take all of the hypothesis, all of the parameters and use the cross validation set to **validate them**, measure average squared error on cross validation set and pick the model which gives **the lowest error**.



Having done that, finally, if we want to report the test error, we take the model we've selected (the model with the lowest error) and test it with the test set, and look at how well it does on our test set.

Bias/variance as a function of the regularization parameter λ

To get a better understanding of how cross validation and training error vary as we vary the regularization parameter λ, what we're going to do is plot the J_{train} and plot J_{CV} meaning just how well does our hypothesis do on the training set and how does our hypothesis do on our cross validation set as we vary our regularization parameter λ (for this purpose we're going to define training error without using a regularization parameter, and cross validation error without using the regularization parameter).



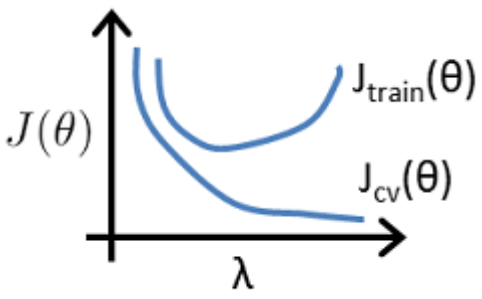
- J_{train}
  - When λ is small we get a small value (regularization basically goes to 0).
  - When λ is large we get a large vale corresponding to high bias (underfitting).
- J_{CV}
  - When λ is small we see high variance (we over fit the data).
  - When λ is large we end up underfitting, so this is bias
    - So cross validation error is high

Such a plot can help show us we're picking a good value for λ.

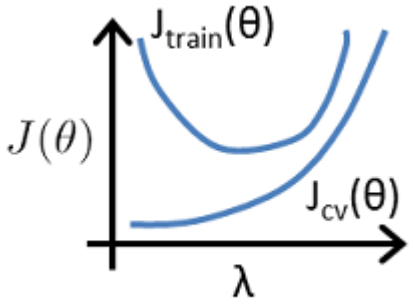
Video Question: Consider regularized logistic regression. Let

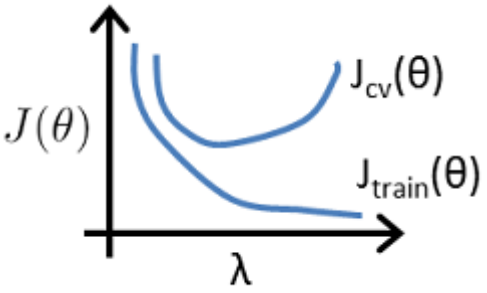
- $J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=2}^{n}\theta_j^2]$
- $J_{\text{train}}(\theta) = \frac{1}{2m_{\text{train}}}[\sum_{i=1}^{m_{\text{train}}}(h_\theta(x_{\text{train}}^{(i)}) - y_{\text{train}}^{(i)})^2]$
- $J_{\text{CV}}(\theta) = \frac{1}{2m_{\text{CV}}}[\sum_{i=1}^{m_{\text{CV}}}(h_\theta(x_{\text{CV}}^{(i)}) - y_{\text{CV}}^{(i)})^2]$

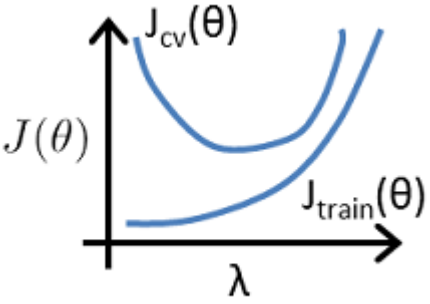
Suppose you plot J_{train} and J_{CV} as a function of the regularization parameter λ. which of the following plots do you expect to get?

-


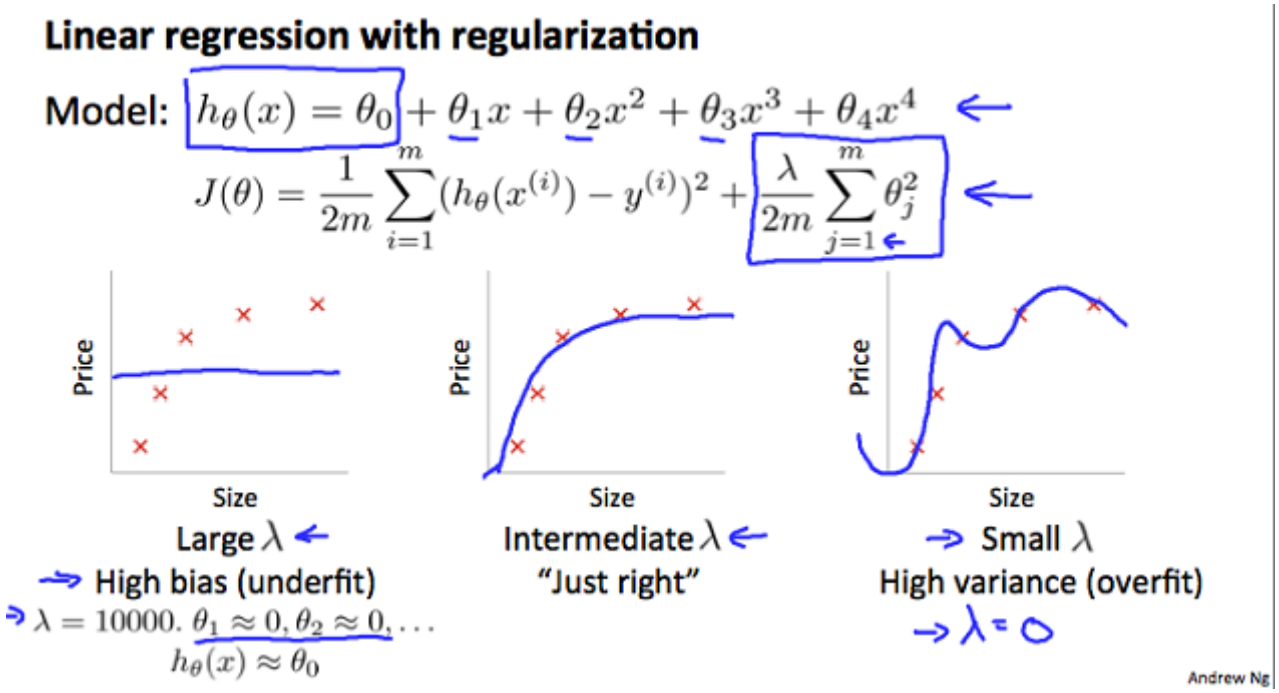
-



- 



Summary



In the figure above, we see that as λ increases, our fit becomes more rigid. On the other hand, as λ approaches 0, we tend to over overfit the data.

So how do we choose our parameter λ to get it 'just right'? In order to choose the model and the regularization term λ, we need to:

1. Create a list of lambdas (i.e. $\lambda \in 0, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 2.56, 5.12, 10.24$);
2. Create a set of models with different degrees or any other variants.
3. Iterate through the λs and for each λ go through all the models to learn some Θ.
4. Compute the cross validation error using the learned Θ (computed with λ) on the $J_{CV}(\Theta)$ without regularization or $\lambda = 0$.
5. Select the best combo that produces the lowest error on the cross validation set.
6. Using the best combo Θ and λ, apply it on $J_{test}(\Theta)$ to see if it has a good generalization of the problem.