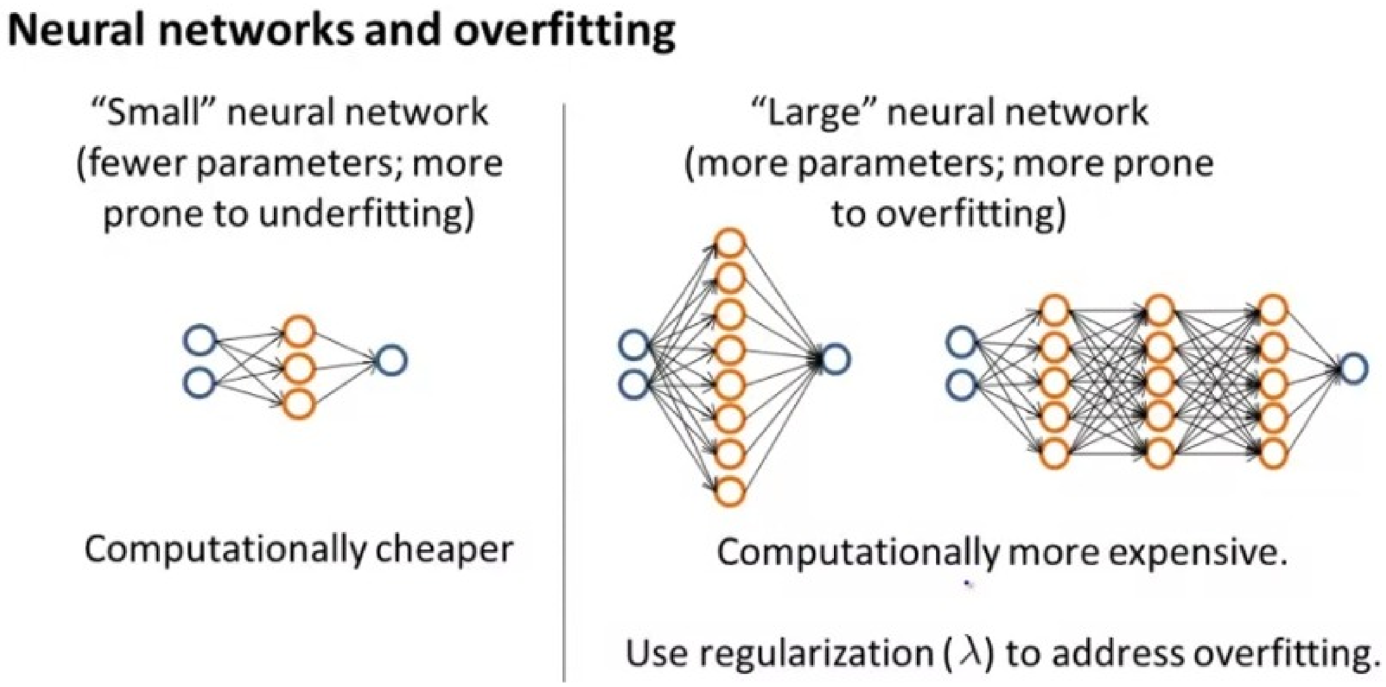
Deciding What to Do Next Revisited

Debugging a learning algorithm:

Suppose we have implemented regularized linear regression to predict housing prices. However, when we test our hypothesis in a new set of houses, we find that it makes unacceptably large errors in its prediction. What should we try next?

- **Get more training examples:** Fixes high variance
  - Not good if we have high bias
- **Try smaller sets of features:** Fixes high variance
  - Not good if we have high bias
- **Try getting additional features:** Fixes high bias
  - Because hypothesis is too simple, make hypothesis more specific
- **Try adding polynomial features (x_1^2, x_2^2, x_1, x_2, etc.):** Fixes high bias
- **Try decreasing λ:** Fixes high bias
- **Try increasing λ:** Fixes high variance.



Relating it all back to neural networks - selecting a network architecture

- One option is to use a small neural network
  - Few (maybe one) hidden layer and few hidden units
  - Such networks are prone to under fitting
  - But they are computationally cheaper
- Using a single hidden layer is good default
  - Also try with 1, 2, 3, see which performs best on cross validation set
  - So like before, take three sets (training, cross validation)
- More units
  - This is computational expensive
  - Prone to over-fitting
  - Use regularization to address over fitting

Video Question: Suppose you fit a neural network with one hidden layer to a training set. You find that the cross validation error $J_{\text{CV}}(\theta)$ is much larger than the training error $J_{\text{train}}(\theta)$. Is increasing the number of hidden units likely to help?

- Yes, because this increases the number of parameters and lets the network represent more complex functions.
- Yes, because it is currently suffering from high bias.
- No, because it is currently suffering from high bias, so adding hidden units is unlikely to help.

> No, because it is currently suffering from high variance, so adding hidden units is unlikely to help.

Summary

Our decision process can be broken down as follows:

- **Getting more training examples:** Fixes high variance
- **Trying smaller sets of features:** Fixes high variance
- **Adding features:** Fixes high bias
- **Adding polynomial features:** Fixes high bias
- **Decreasing λ:** Fixes high bias
- **Increasing λ:** Fixes high variance.

Diagnosing Neural Networks

- A neural network with fewer parameters is **prone to underfitting**. It is also **computationally cheaper**.
- A large neural network with more parameters is **prone to overfitting**. It is also **computationally expensive**. In this case you can use regularization (increase λ) to address the overfitting.

Using a single hidden layer is a good starting default. You can train your neural network on a number of hidden layers using your cross validation set. You can then select the one that performs best.

Model Complexity Effects:

- Lower-order polynomials (low model complexity) have high bias and low variance. In this case, the model fits poorly consistently.
- Higher-order polynomials (high model complexity) fit the training data extremely well and the test data extremely poorly. These have low bias on the training data, but very high variance.
- In reality, we would want to choose a model somewhere in between, that can generalize well but also fits the data reasonably well.