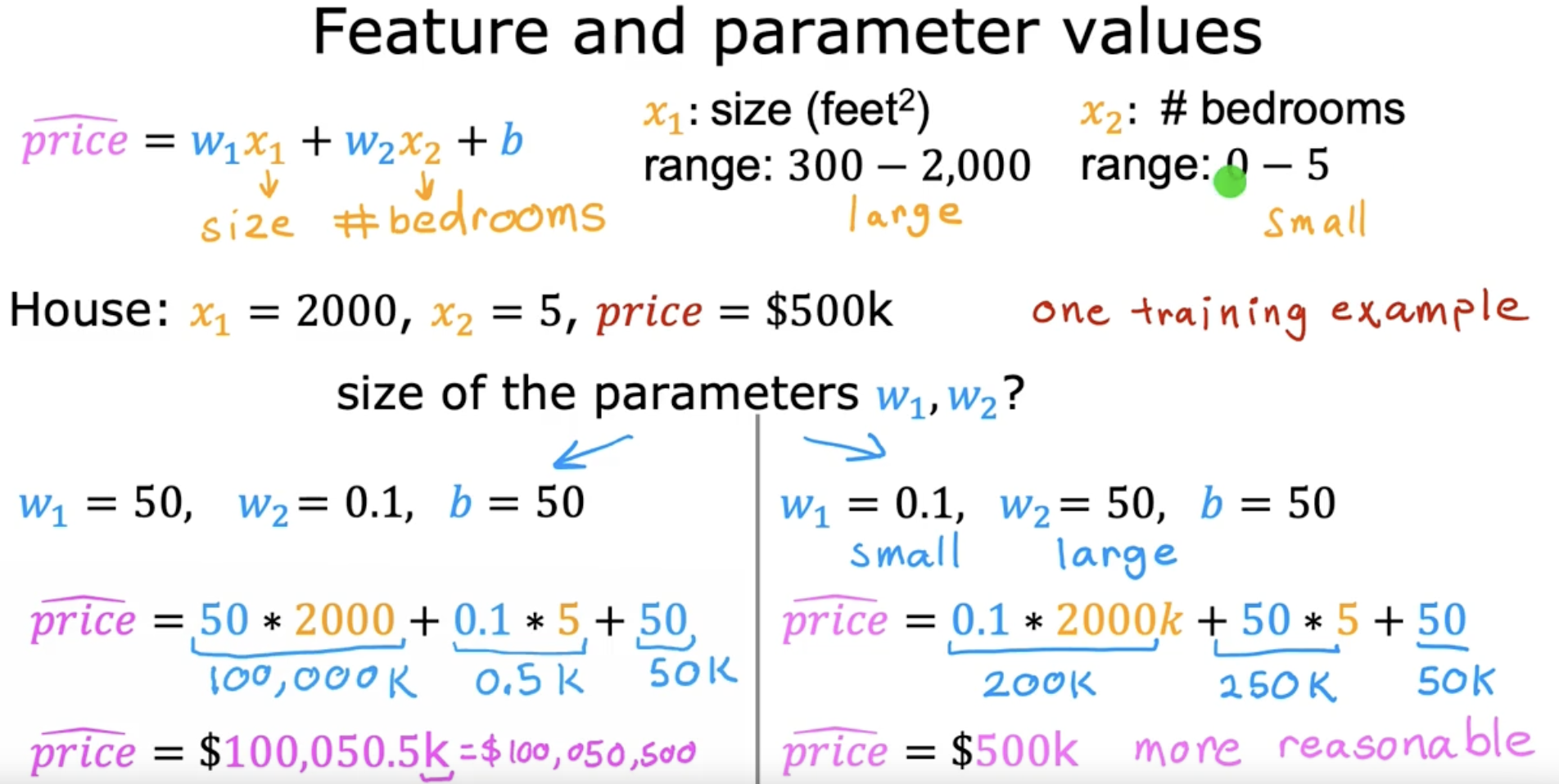
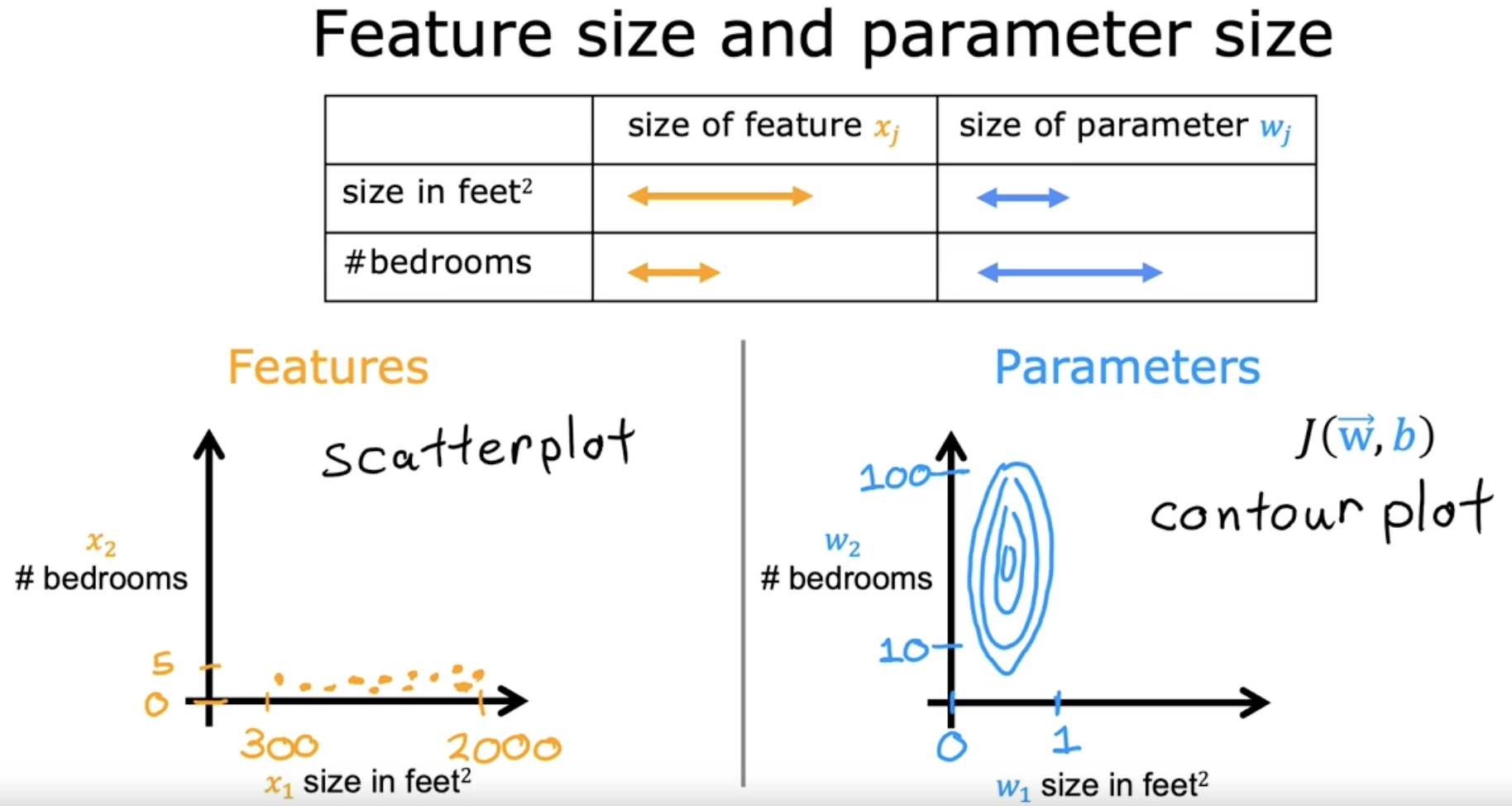
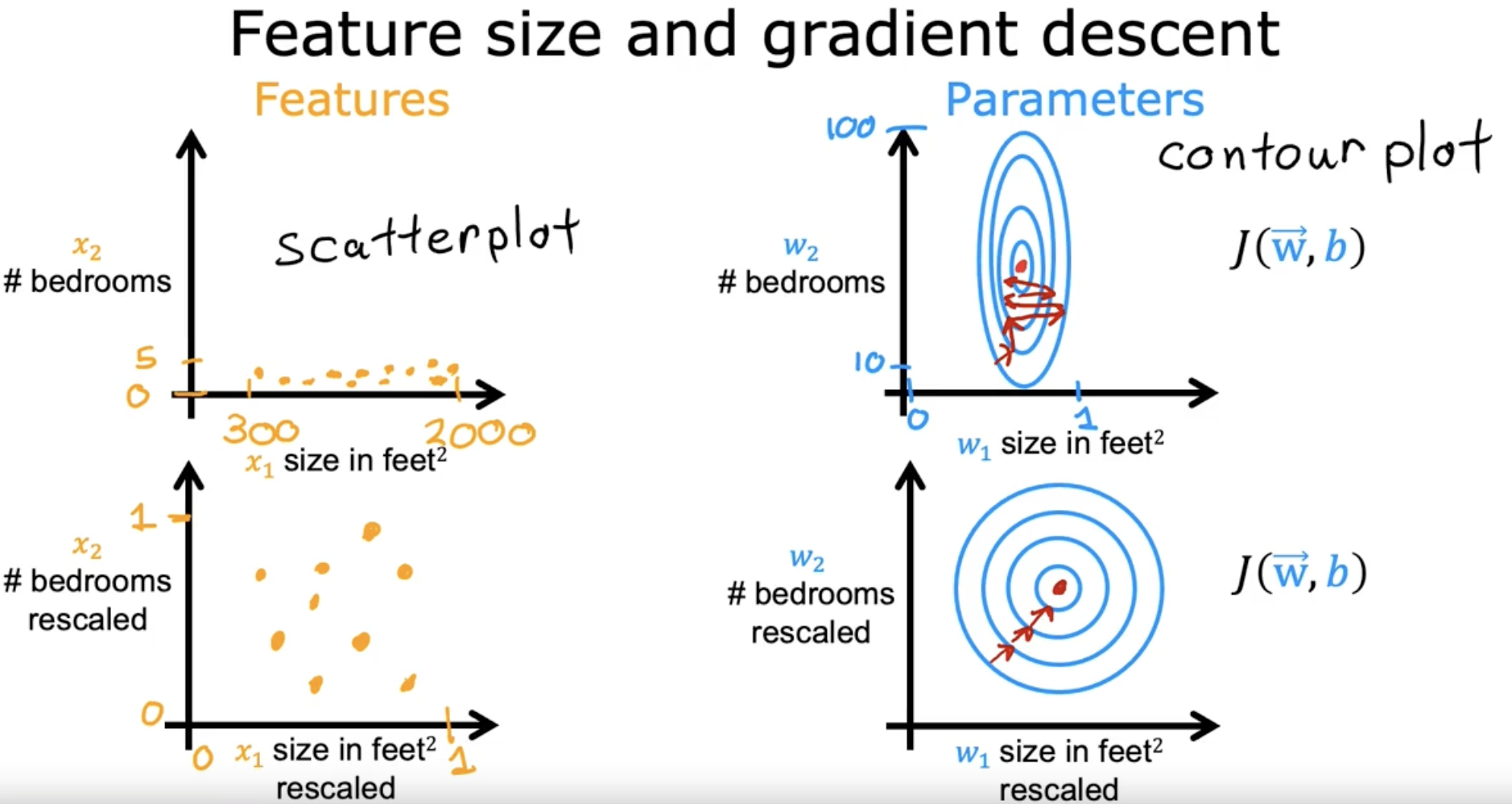
# **Feature Scaling part-1**

🡪 to run gradient descent run much faster  




Horizontal axis 🡪 Narrow range  
Vertical axis 🡪 Wider range

Small change in w1, causes large change in cost.  
Small change in w2, won’t affect the change in cost.



When you have different features that take on very different ranges of values, it can cause gradient descent to run slowly but re-scaling the different features so they all take on comparable range of values. Because speed, upgrade and dissent significantly.

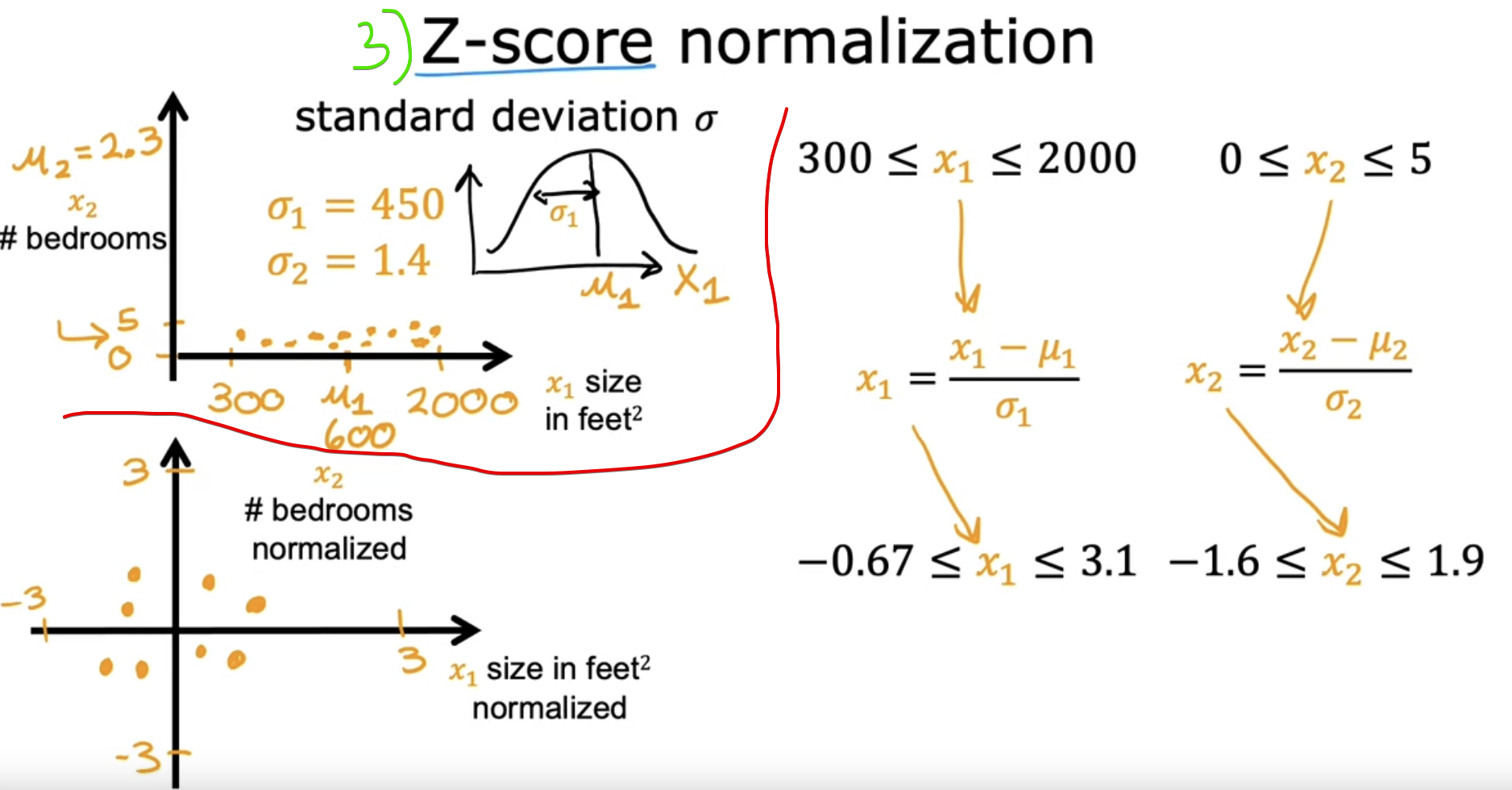
# **Feature Scaling part-2**

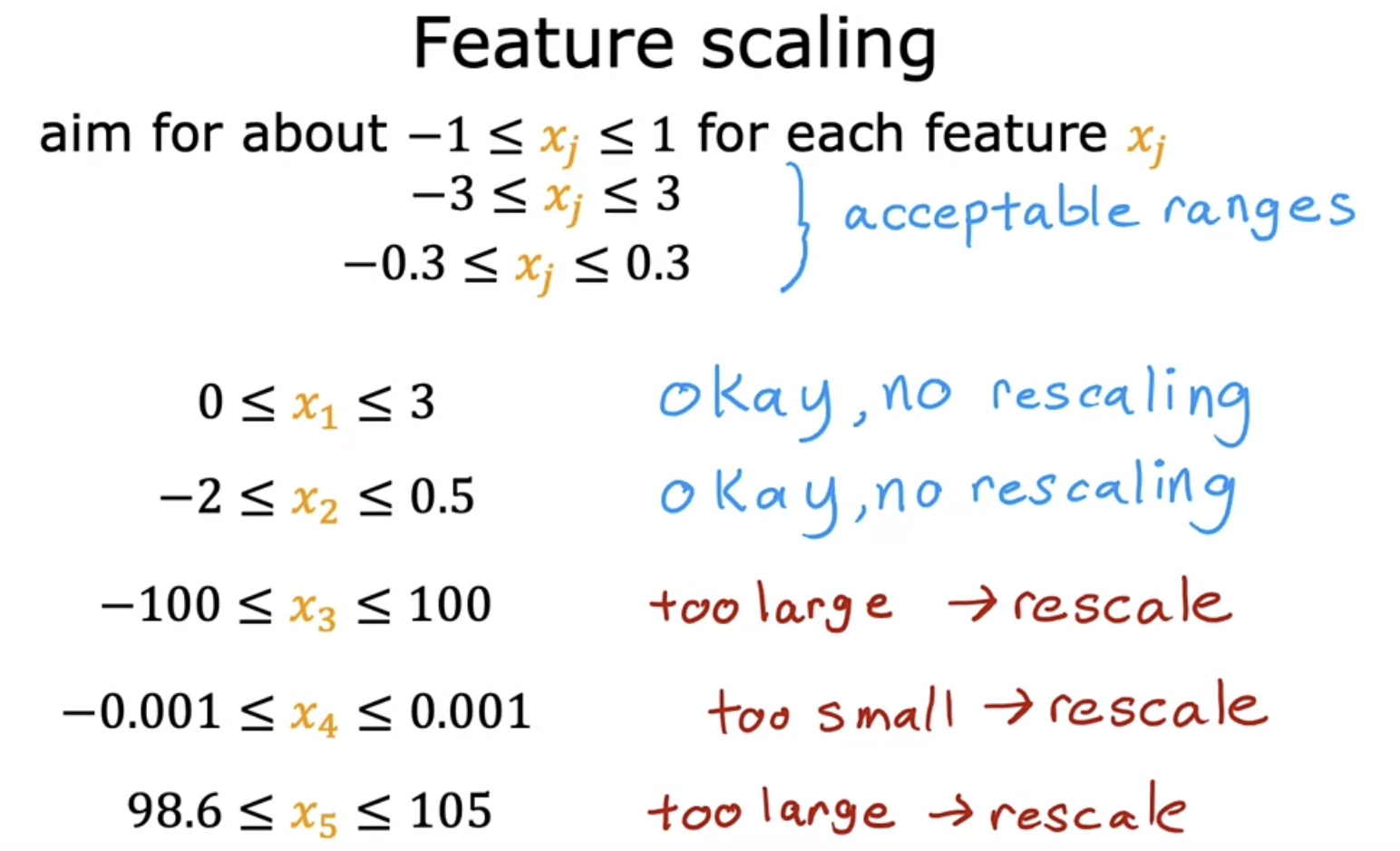
A picture containing diagram

Description automatically generated

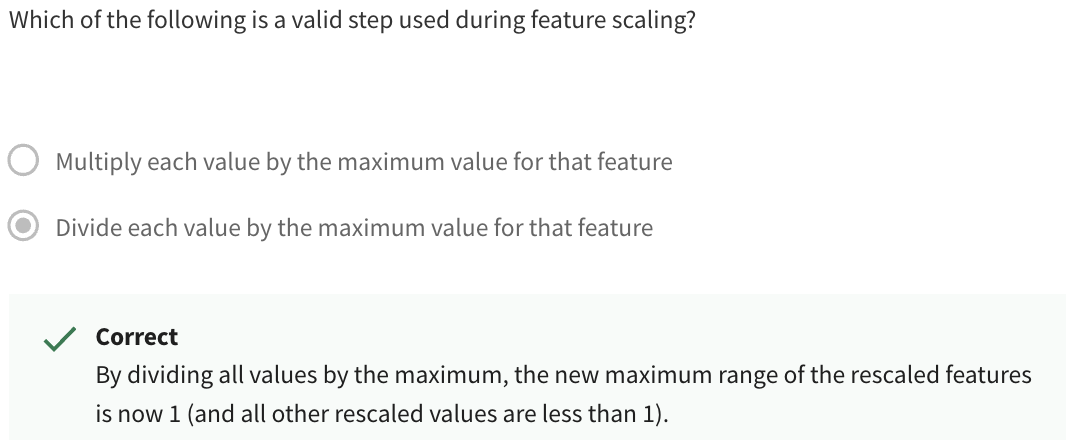
Diagram

Description automatically generated





Patient’s temperature 🡪 also very large to be scalable



Lets see whether this feature scaling helps in Gradient Descent converging/diverging ???

# **Checking gradient descent for convergence**

Good Gradient Descent  
After every iteration, the cost function must be reduced.

If the cost function increases after iterations, then the learning rate must be high.

Text

Description automatically generated with low confidence

By the way, the number of iterations that gradient descent takes a conversion can vary

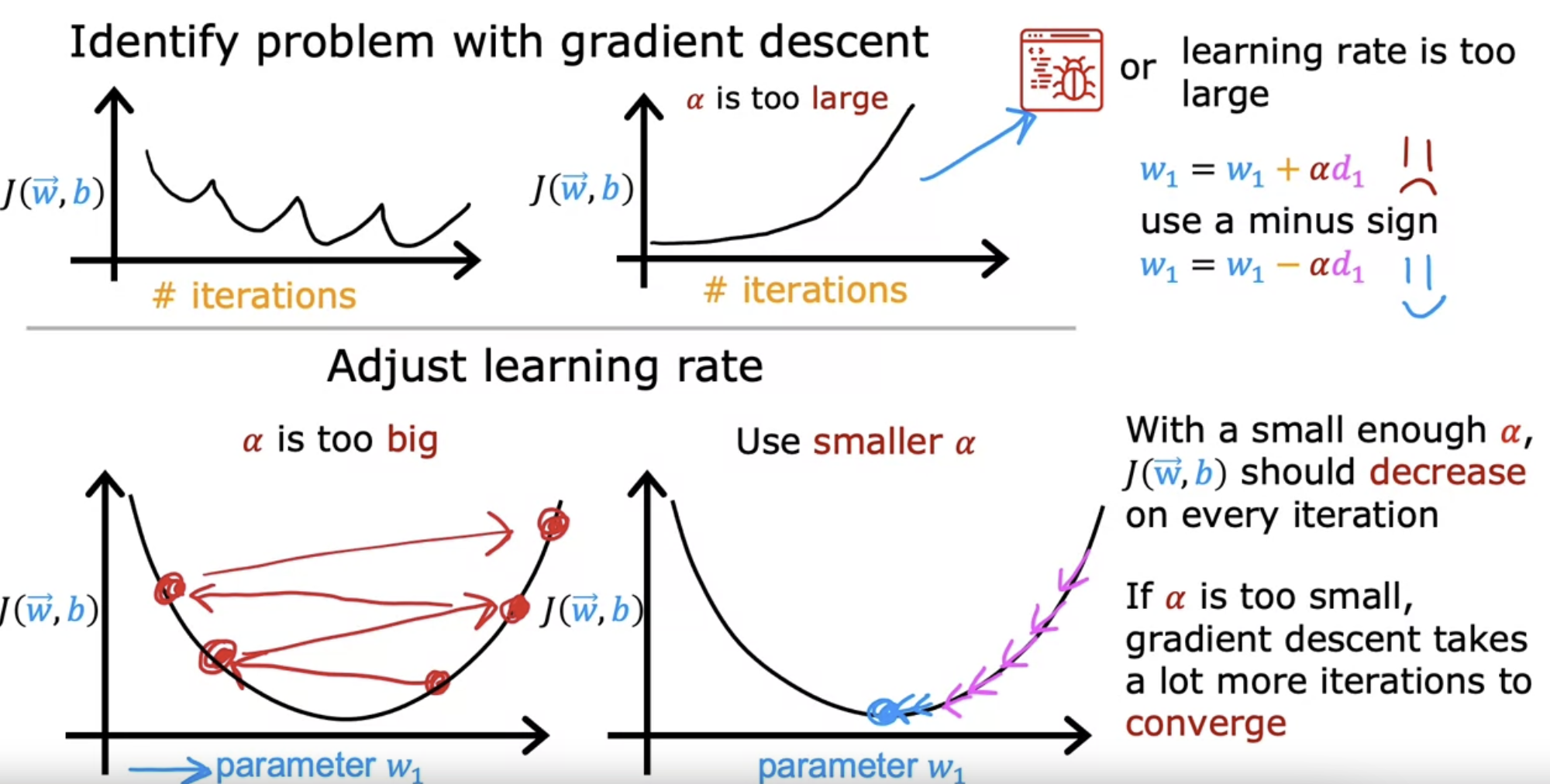
a lot between different applications.

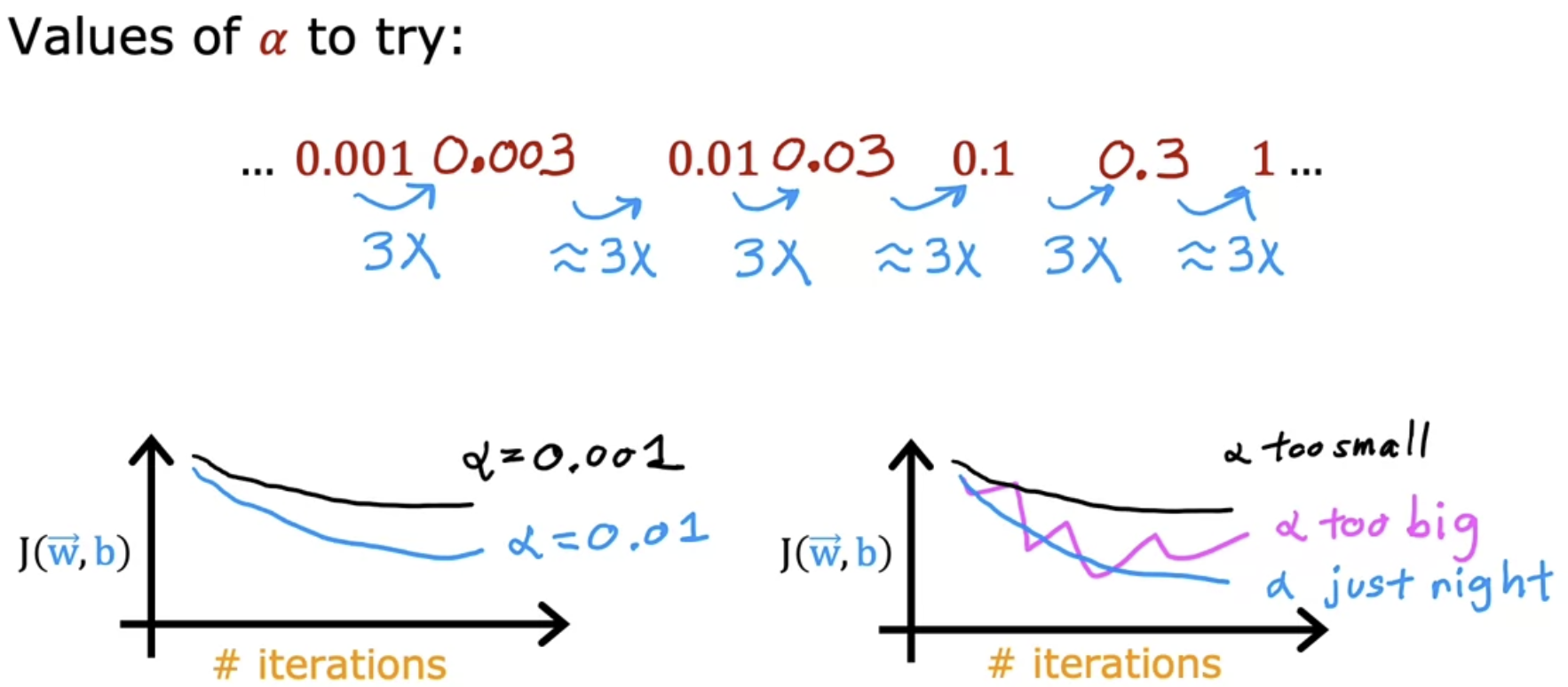
In one application, it may converge after just 30 iterations.

For a different application, it could take 1,000 or 100,000 iterations.

It turns out to be very difficult to tell in advance how many iterations gradient descent needs to converge, which is why you can create a graph like this, a learning curve.

# **Choosing the learning rate**





After multiplying with 3, cost function tremendously decreased.

Graphical user interface, text, application

Description automatically generated  
Graphical user interface, text

Description automatically generated

Now, there are couple more ideas that you can use to make multiple linear regression much more powerful.

That is choosing custom features, which will also allow you to fit curves,  not just a straight line to your data.

Refer Feature\_Scaling\_and\_Learning\_Rate\_Soln.ipynb file

# **Feature Engineering**

# **Polynomial Regression**

Feature Scaling part-1