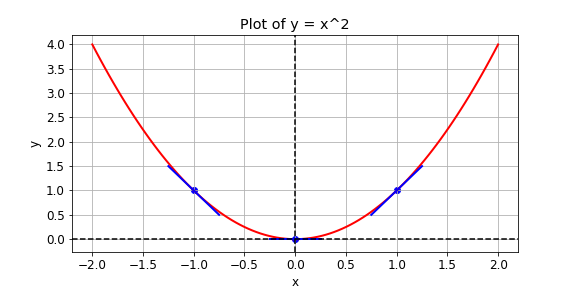
**Optimizing with gradients**

You are given a loss function, y=x2, which you want to minimize. You can do this by computing the slope using the GradientTape() operation at different values of x. If the slope is positive, you can decrease the loss by lowering x. If it is negative, you can decrease it by increasing x. This is how gradient descent works.



In practice, you will use a high level tensorflow operation to perform gradient descent automatically. In this exercise, however, you will compute the slope at x values of -1, 1, and 0. The following operations are available: GradientTape(), multiply(), and Variable().

**Instructions**

**100 XP**

* Define x as a variable with the initial value x0.
* Set the loss function, y, equal to x multiplied by x. Do not make use of operator overloading.
* Set the function to return the gradient of y with respect to x.

def compute\_gradient(x0):

# Define x as a variable with an initial value of x0

x = Variable(x0)

with GradientTape() as tape:

tape.watch(x)

# Define y using the multiply operation

y = multiply(x, x)

# Return the gradient of y with respect to x

return tape.gradient(y, x).numpy()

# Compute and print gradients at x = -1, 1, and 0

print(compute\_gradient(-1.0))

print(compute\_gradient(1.0))

print(compute\_gradient(0.0))

Excellent work! Notice that the slope is positive at x = 1, which means that we can lower the loss by reducing x. The slope is negative at x = -1, which means that we can lower the loss by increasing x. The slope at x = 0 is 0, which means that we cannot lower the loss by either increasing or decreasing x. This is because the loss is minimized at x = 0.