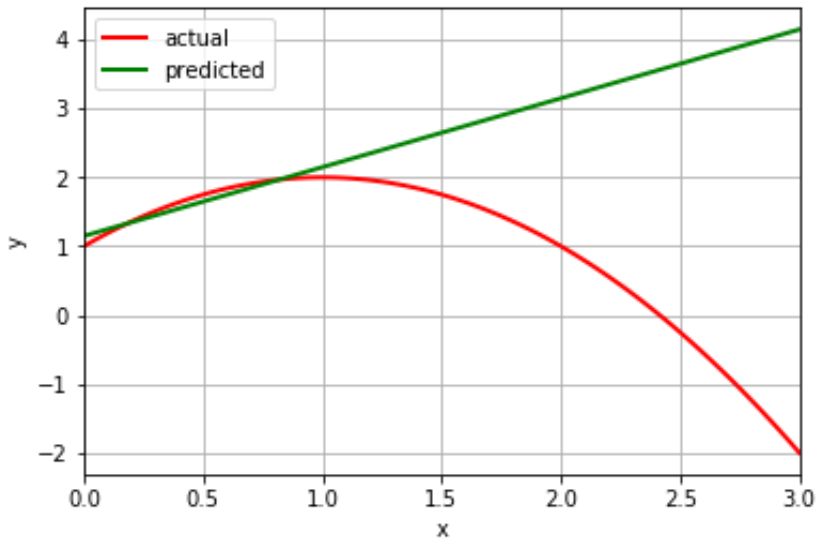
In [8]:
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model, preprocessing
import numpy.polynomial.polynomial as poly
%matplotlib inline


beta = np.array([1, 2, -1])
nsamp = 10
xdat = np.linspace(0, 1, nsamp)
ydat = poly.polyval(xdat, beta)
d = 1
beta_hat = poly.polyfit(xdat, ydat, d)
```

```
[ 1.14814815  1.        ]
```

In [14]:
```
xp = np.linspace(0, 3, 100)
yp = poly.polyval(xp, beta)
yp_hat = poly.polyval(xp, beta_hat)
```

In [21]:
```
plt.xlim(0, 3)
plt.plot(xp, yp, 'r-', linewidth=2)
plt.plot(xp, yp_hat, 'g-', linewidth=2)
plt.legend(['actual', 'predicted'], loc = 'upper left')
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
```

Out[21]: <matplotlib.text.Text at 0x1151fc400>



In [22]:
```
bias = np.zeros(100)
for i in range(100):
    bias[i] = (yp_hat[i] - yp[i])**2
bias_order = np.argsort(-bias)
print(xp[bias_order[0]])
```

```
3.0
```

In [ ]:
```

```