Machine Learning

March 16, 2019

Assignment 02: Taylor Approximation
Name: Joon Ho Yang
ID: 20141043

```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
        from scipy.misc import derivative
```

1. Define a differentiable function that maps from real number to real number.

$f(x) = x^2 + 5$

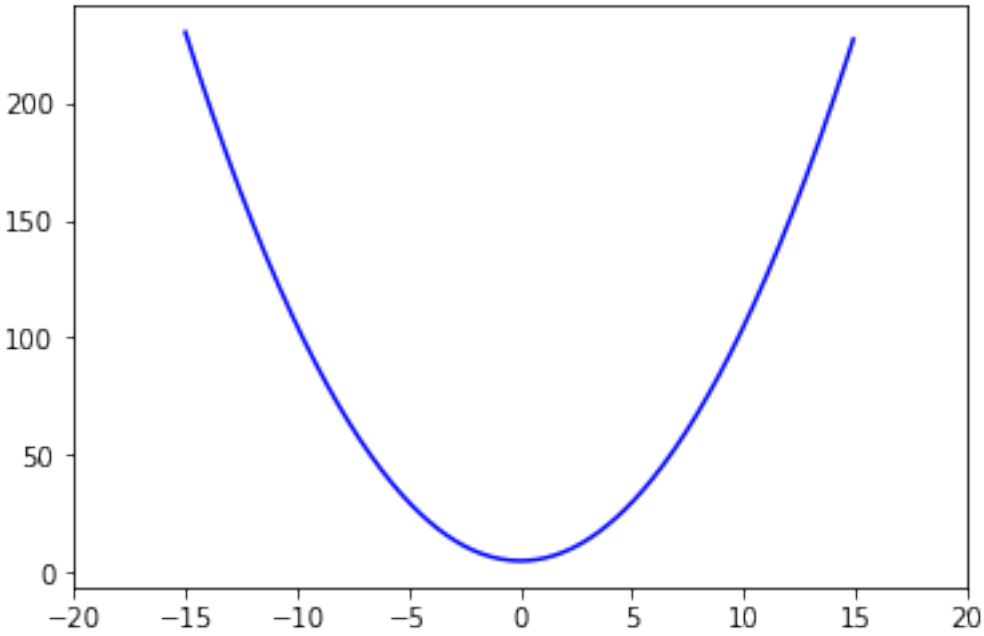
```
In [3]: def givenFunction(x):
            f = x ** 2 + 5
            return f
```

2. Define a domain of the function.

```
In [4]: x = np.arange(-15, 15, 0.1)
```

3. Plot the function.

```
In [5]: f = givenFunction(x)
        plt.figure(1)
        plt.plot(x, f, 'b')
        plt.xlim(-20, 20)
        plt.show()
```



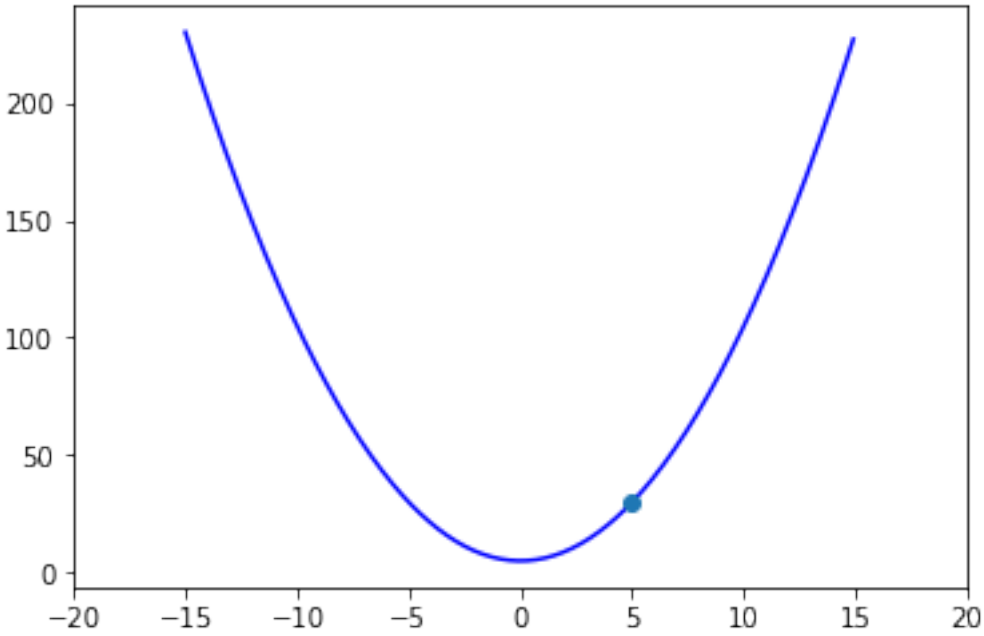
4. Select a point within the domain.

$x = 5$
$y = 30$

```
In [6]: point = 5
```

5. Mark the selected point on the function.

```
In [7]: plt.figure(1)
        plt.plot(x, f, 'b')
        plt.xlim(-20, 20)
        plt.plot(point, givenFunction(point),'o')
        plt.show()
```

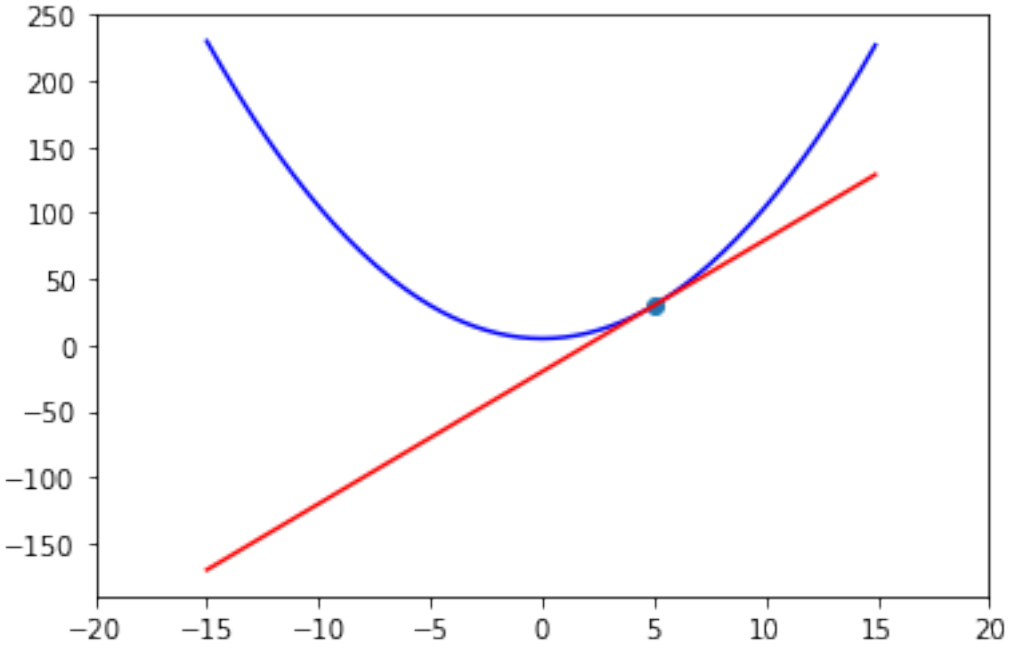


6. Define the first-order Taylor approximation at the selected point.

```
In [8]: def firstDerivative(x):
            return derivative(givenFunction, x, dx=1e-4)
        def taylorApproximation(x, xPoint, yPoint):
            f = firstDerivative(xPoint)*(x-xPoint)+yPoint
            return f
        taF=taylorApproximation(x, point, givenFunction(point))
```

7. Plot the Taylor approximation with the same domain of the original function.

```
In [9]: plt.figure(1)
        plt.plot(x, f, 'b')
        plt.xlim(-20, 20)
        plt.plot(point, givenFunction(point),'o')
        plt.plot(x, taF, 'r')
        plt.show()
```


250
200
150
100
50
0
−50
−100
−150
−20
−15
−10
−5
0
5
10
15
20