

제1 고지 : 미분 자동 계산

STEP 2 : 변수를 낳는 함수

그림 2-1 변수와 함수의 관계



- 함수(f) : 어떤 변수(x)부터 다른 변수(y)로의 대응 관계
- 모든 함수(Function)가 공통적으로 제공하는 기능은 다음과 같이 정의 될 수 있다.
 - 입력 변수 : Variable
 - 구체적 계산 : forward()
 - 출력 변수 : Variable

In []:

```
import torch
import numpy as np
import torch.nn as nn

class Variable:
    def __init__(self, data: np.ndarray) -> None:
        self.data = data

class Function:
    """
    Function Base Class
    """

    def __call__(self, input: Variable) -> Variable:
        x = input.data # 입력 변수
        y = self.forward(x) # 구체적 계산
        return Variable(y) # 출력 변수

    def forward(self, x):
        """
        구체적인 함수 계산 담당
        # NOTE : 0차원의 ndarray 의 경우 np.float64로 변환되는데(넘파이가 의도)
        """
        raise NotImplementedError

class Square(Function):
    """
    y= x ** 2
    """

    def forward(self, x: np.ndarray) -> np.ndarray:
        return x**2

class Sigmoid(Function):
    """
    y = 1 / (1 + e ^(-x))
    """
```

```

"""

def forward(self, x: np.ndarray) -> np.ndarray:
    return 1 / (1 + np.exp(-x))

```

In []:

```

# Dezero
x = Variable(np.array(10))
f = Square()
y = f(x)
print(type(y))
print(y.data)

```

```

<class '__main__.Variable'>
100

```

In []:

```

# Dezero ~ Pytorch
## Dezero
x = Variable(np.array(1))
f = Sigmoid()
y = f(x)
print(f"Dezero : {y.data}")

## Pytorch
x = torch.Tensor([1])
f = nn.Sigmoid()
y = f(x)
print(f"PyTorch : {y.data}")

```

```

Dezero : 0.7310585786300049
PyTorch : tensor([0.7311])

```