

모멘텀은 운동량을 뜻하는 단어로, 신경망에서의 모멘텀 기법은 아래 수식과 같이 표현할 수 있습니다.

$$v \leftarrow \alpha v + \eta \frac{\partial L}{\partial W}$$

모멘텀의 속도 갱신 수식

$$W \leftarrow \cancel{W - \alpha v} \quad W - v$$

모멘텀 가중치 갱신 수식

SGD에서와 마찬가지로  
**W**는 갱신할 가중치 매개변수,  
**L**은 손실함수를 나타내고  
**η**는 학습률 learning rate,  
**∂L/∂W**은 **W**에 대한 손실함수의 기울기를 나타냅니다.

SGD와 달리 변수 **v**가 등장하는데 물리에서 운동량을 나타내는 식은  $p = mv$ , 질량  $m$ , 속도  $v$ 이므로  
위 수식에서도 **v**는 속도를 의미합니다.  
매개변수 **α**를 **v**에 곱해서  $\alpha v$  항은 물체가 아무 힘도 받지 않을 때도 서서히 하강시키는 역할을 하게 됩니다. 물리에서의 마찰력이라고 생각하면 편할 것 같습니다. 하강시키기 위해 **α**를 0.9등의 값으로 설정합니다.

속도  $v$ 를

모멘텀을 파이썬 코드로 구현하면 다음과 같습니다.

```
import numpy as np
```

```
class Momentum:
```

```
    def __init__(self, lr=0.01, momentum=0.9):
```

```
        self.lr = lr # $\eta$ 
```

```
        self.momentum = momentum # $\alpha$ 
```

```
        self.v = None
```

```
    def update(self, params, grads):
```

# update()가 처음 호출될 때 v에 매개변수와 같은 구조의 데이터를 딕셔너리 변수로 저장

```
        if self.v is None:
```

```
            self.v = {}
```

```
            for key, val in params.items():
```

```
                self.v[key] = np.zeros_like(val)
```

```
        for key in params.keys():
```

```
            self.v[key] = self.momentum*self.v[key] +
```

```
self.lr*grads[key]
```

```
            params[key] params[key] = self.v[key]
```