

# Nariflow의 자료 구조

NariFlow

- 수치를 저장하는 Parameter와 연산을 저장하는 Function으로 이분화

Parameter	Function
<div>Data</div> <ul style="list-style-type: none"><li>- 초기는 랜덤 초기화된 가중치</li><li>- 이후엔 학습중 가중치</li></ul>	<div>Forward</div> <ul style="list-style-type: none"><li>- Data를 가져와 정전파 연산 수행</li></ul>
<div>Grad</div> <ul style="list-style-type: none"><li>- 초기는 None</li><li>- 이후엔 계산된 그래디언트</li></ul>	<div>Backward</div> <ul style="list-style-type: none"><li>- Data와 Gradient를 가져와 역전파 연산 수행</li></ul>
<div>Generation</div> <ul style="list-style-type: none"><li>- 해당 Parameter 계산이 몇 세대 연산인지 기억</li><li>- 역전파시 순차적 계산을 보장</li></ul>	<div>Making Tape</div> <ul style="list-style-type: none"><li>- 정전파 시 Parameter, Function의 메모리 주소를 Tape에 기록</li></ul>

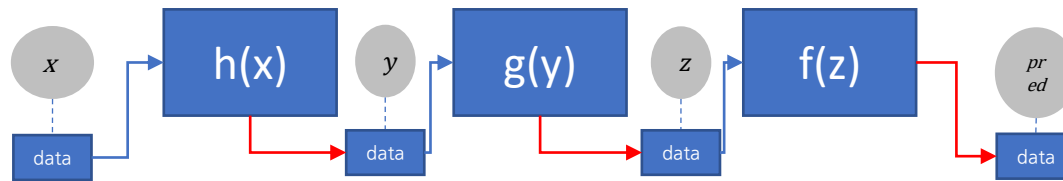
# Nariflow의 연산 구조

NariFlow

- Function Class의 Forward 함수에 Data를 투입하여 통해 정전파 수행
- Function Class의 Backward 함수에 Data와 Grad를 투입하여 역전파 수행

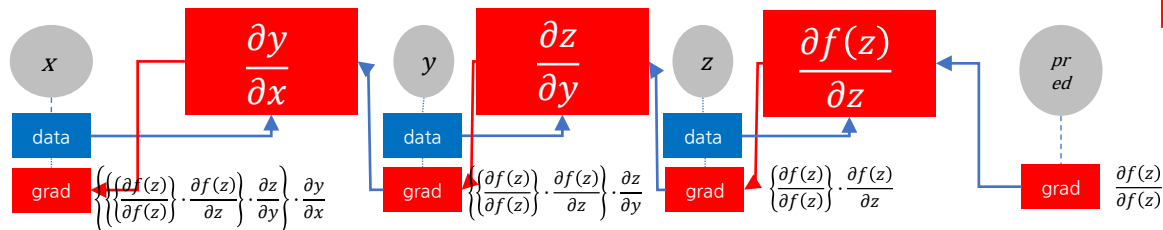
## 정전파

$$pred = f(g(h(x)))$$



## 역전파

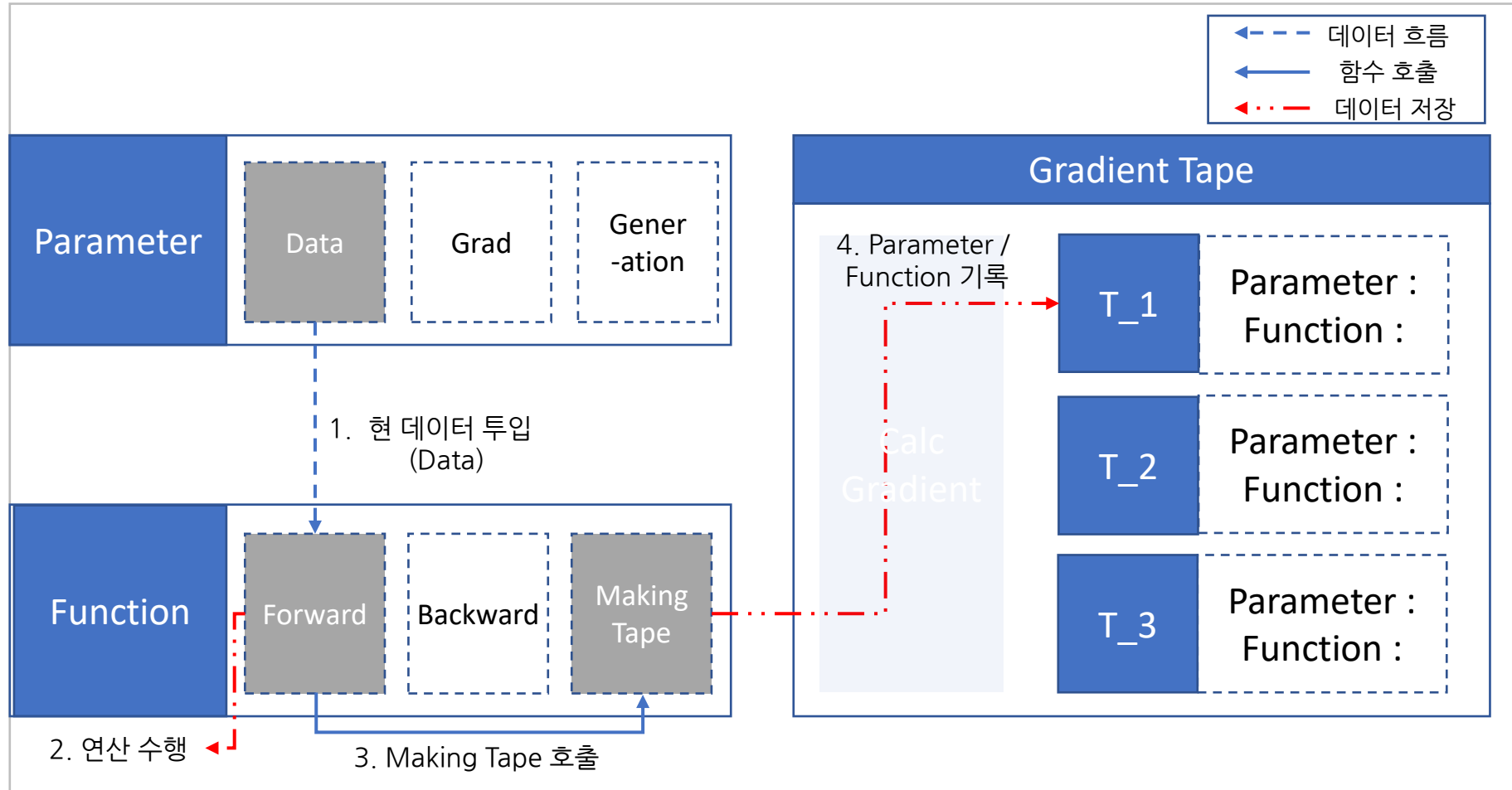
$$\frac{\partial pred}{\partial x} = \left\{ \left\{ \left\{ \frac{\partial f(z)}{\partial f(z)} \right\} \cdot \frac{\partial f(z)}{\partial z} \right\} \cdot \frac{\partial z}{\partial y} \right\} \cdot \frac{\partial y}{\partial x}$$



# Nariflow의 연산 구조

NariFlow

- Parameter에 저장된 Data를 가져와 Function의 Forward에서 연산 수행
- 연산 수행 Log를 Gradient Tape에 저장



# Nariflow의 연산 구조

- GradientTape에 저장된 연산 기록을 역순으로 Load
- Parameter에 저장된 Data와 Grad를 가져와 Function의 Backward에서 연산 수행

