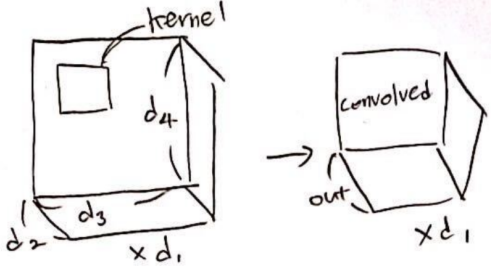


Matmul Layers

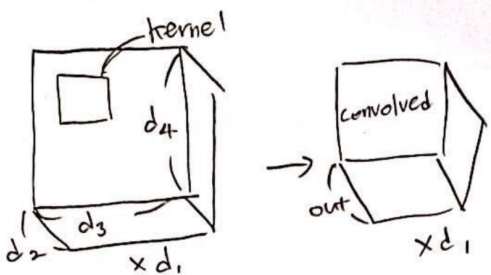
torch.nn.Linear

torch.nn.Linear(in, out, bias=True)(x)	
	Require
	<ul style="list-style-type: none"> • $x = (d_1, d_2, \dots, d_k)$ • $\text{rank}(x) \geq 1$ • $d_k = \text{in}$
	Guarantees
	<ul style="list-style-type: none"> • $y = (d_1, d_2, \dots, d_{k-1}, \text{out})$
	Comment
	<ul style="list-style-type: none"> • $y = xA^T + b$를 계산하는 dense 레이어 • 1차원인 경우에도 잘 작동합니다. • <i>bias</i> 옵션은 출력 shape에 영향을 주지 않습니다.

$$\begin{array}{c}
 \sigma \vdash E \Rightarrow e, c \\
 e' = e[1 : k - 1]@(\text{out}) \\
 c' = \{(\text{rank}(e) \geq 1) \wedge (d_k = \text{in})\} \\
 \hline
 \sigma \vdash \text{Linear}(\text{in}, \text{out}, \text{bias} = \text{True})(E) \Rightarrow e', c \cup c'
 \end{array}$$

Activations

torch.nn.ReLU, torch.nn.ReLU6, torch.relu, torch.nn.functional.relu

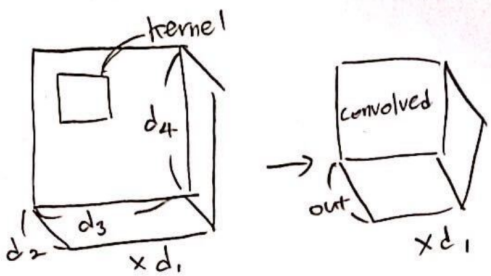
torch.nn.ReLU(inplace=True)(x)	
	Require
	Guarantees
	<ul style="list-style-type: none"> • $y = x$ (same shape)
	Comment
	<ul style="list-style-type: none"> • <i>inplace</i> 옵션은 shape에 영향을 주지 않습니다. • ReLU6도 ReLU와 똑같은 방식으로 shape 계산 • Builtins인 torch.relu와 torch.nn.functional.relu는 같은거

$$\forall \text{ft} \in \{\text{ReLU}, \text{ReLU6}\}, \quad \frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \text{ft}(\text{inplace} = \text{True})(E) \Rightarrow e, c}$$

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \text{relu}(E, \text{inplace} = \text{True}) \Rightarrow e, c}$$

Technique

`torch.nn.Dropout`, `torch.dropout`, `torch.nn.functional.dropout`

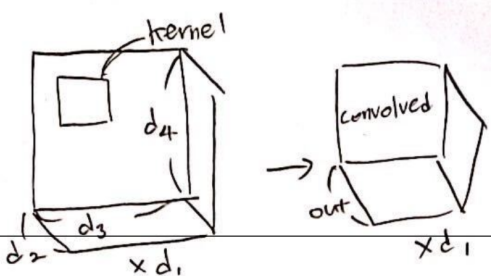
<code>torch.nn.Dropout(...)(x)</code>	
	Require
	Guarantees
	<ul style="list-style-type: none">• $y = x$ (same shape)
	Comment
	<ul style="list-style-type: none">• 모든 옵션은 shape에 영향을 주지 않습니다.• Bultins인 <code>torch.dropout</code>와 <code>torch.nn.functional.dropout</code>는 서로 역할이 같습니다.

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \text{Dropout}(\dots)(E) \Rightarrow e, c}$$

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \text{dropout}(E, \dots) \Rightarrow e, c}$$

Wrapper

`torch.nn.Sequential`

<code>torch.nn.Sequential(l1, l2, l3, ..., ln)(x)</code>	
	Require
	<ul style="list-style-type: none">• 순차적으로 shape이 맞아떨어져야함
	Guarantees
	<ul style="list-style-type: none">• $y = l_n \circ l_{n-1} \circ \dots \circ l_1(x)$
$\frac{\sigma \vdash l_n \circ l_{n-1} \circ \dots \circ l_1(E) \Rightarrow e, c}{\sigma \vdash \text{Sequential}(l_1, l_2, \dots, l_n)(E) \Rightarrow e, c}$	