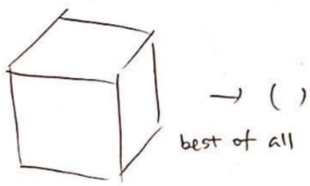
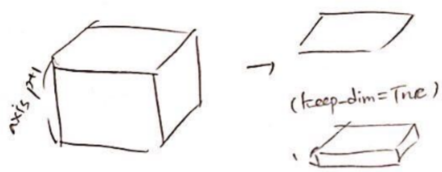
Statistics

`torch.max`, `torch.min`

<table>
<tr><td colspan="2"><code>torch.max(input) or .min</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td></td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>$|y| = ()$</td></tr>
<tr><td>Comment</td></tr>
<tr><td>주어진 텐서에서 최대/최소 값을 ()-shape 텐서로 반환(스칼라 X)</td></tr>
</table>

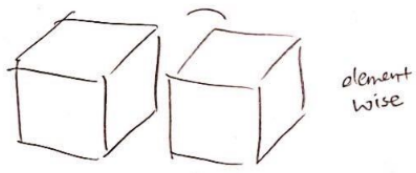
$$\forall \texttt{ft} \in \{\texttt{min}, \texttt{max}\}, \qquad \frac{\sigma \vdash E \Rightarrow \_, c}{\sigma \vdash \texttt{ft}(E) \Rightarrow (), c}$$

<table>
<tr><td colspan="2"><code>torch.max(input, dim, keep_dim=False, out=None) or .min</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>$|input| = (d_1, d_2, \ldots, d_k)$ $k \geq 1$, $0 \leq dim < k$</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>$|y| = |z| = (d_1, d_2, \ldots, d_{dim}, d_{dim+2}, \ldots, d_k)$인 (y, z) 튜플 반환 세 번째 인자 $keep_dim$이 $True$이면 $|y| = |z| = (d_1, d_2, \ldots, d_{dim}, 1, d_{dim+2}, \ldots, d_k)$로 처리</td></tr>
<tr><td>Comment</td></tr>
<tr><td>주어진 텐서에서 축 상의 최대/최소값과, 그에 해당하는 인덱스 번호를 쌍으로 엮어 튜플로 반환
out-텐서 인자가 있는 함수</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{min}, \texttt{max}\}, \qquad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@e[n+2{:}k] \\ c' = \{(k \geq 1) \land (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, n) \Rightarrow (e', e'), c \cup c'} \qquad \text{tuple 형태로 반환}$$

$$\forall \texttt{ft} \in \{\texttt{min}, \texttt{max}\}, \qquad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@(1)@e[n+2{:}k] \\ c' = \{(k \geq 1) \land (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, n, True) \Rightarrow (e', e'), c \cup c'} \qquad \text{tuple 형태로 반환}$$

$$\forall \texttt{ft} \in \{\texttt{min}, \texttt{max}\}, \qquad \frac{\sigma \vdash \texttt{ft}(E, n) \Rightarrow (e, e), c}{\sigma \vdash \texttt{ft}(E, n, False) \Rightarrow (e, e), c} \qquad \text{tuple 형태로 반환}$$

<table>
<tr><td colspan="2"><code>torch.max(input, other, out=None) or .min</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>

- $broadcastable(|input|, |other|)$

</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>

- $broadcast(|input|, |other|)$

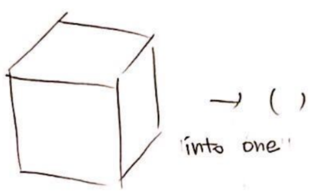
</td></tr>
<tr><td>Comment</td></tr>
<tr><td>

- 두 텐서의 elementwise 최대/최소
- *out*-텐서 인자가 있는 함수

</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{min}, \texttt{max}\}, \qquad \frac{\sigma \vdash E_1 \Rightarrow e_1, c_1 \qquad \sigma \vdash E_2 \Rightarrow e_2, c_2}{\sigma \vdash \texttt{ft}(E_1, E_2) \Rightarrow broadcast(e_1, e_2), c_1 \cup c_2 \cup broadcastable(e_1, e_2)}$$

`torch.sum`, `torch.mean`

<table>
<tr><td colspan="2"><code>torch.sum(input, dtype=None) or .mean</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>

- `.mean`에 대해서는 텐서 타입이 floating이어야 함

</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>

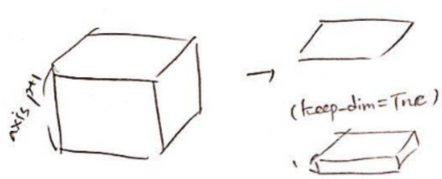
- $|y| = ()$

</td></tr>
<tr><td>Comment</td></tr>
<tr><td>

- 주어진 텐서의 합계/평균값을 ()-shape 텐서로 반환(스칼라 X)

</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{sum}, \texttt{mean}\}, \qquad \frac{\sigma \vdash E \Rightarrow \_, c}{\sigma \vdash \texttt{ft}(E) \Rightarrow (), c}$$

<table>
<tr><td colspan="2"><code>torch.sum(input, dim, keep_dim=False, dtype=None) or .mean</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>

- $|input| = (d_1, d_2, \ldots, d_k)$
- $k \geq 1,\ 0 \leq dim < k$
- `.mean`에 대해서는 텐서 타입이 floating이어야 함

</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>

- $|y| = (d_1, d_2, \ldots, d_{dim}, d_{dim+2}, \ldots, d_k)$
- 세 번째 인자 $keep_dim$이 $True$이면 $|y| = (d_1, d_2, \ldots, d_{dim}, 1, d_{dim+2}, \ldots, d_k)$

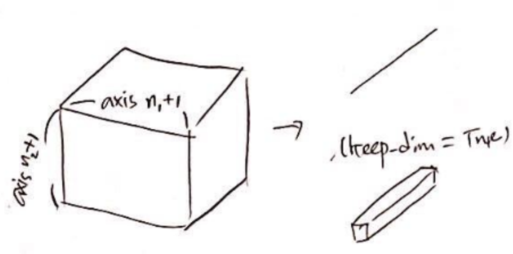
</td></tr>
<tr><td>Comment</td></tr>
<tr><td>

- 주어진 텐서에서 축 상의 합/평균을 반환

</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{sum}, \texttt{mean}\}, \qquad \frac{\begin{array}{l}\sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@e[n+2{:}k] \\ c' = \{(k \geq 1) \wedge (0 \leq n < k)\}\end{array}}{\sigma \vdash \texttt{ft}(E, n) \Rightarrow e', c \cup c'}$$

$$\forall \texttt{ft} \in \{\texttt{sum}, \texttt{mean}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@(1)@e[n+2{:}k] \\ c' = \{(k \geq 1) \land (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, n, True) \Rightarrow e', c \cup c'}$$

<table>
<tr><td colspan="2"><code>torch.sum(input, [n1, n2, ..., nl], keep_dim=False, dtype=None) or .mean</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>

- $|x| = (d_1, d_2, \ldots, d_k)$
- $k \geq 1,\ 0 \leq n_1, n_2, \ldots, n_l < k$
- `.mean`에 대해서는 텐서 타입이 floating이어야 함

</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>

- $|y| = (d_{i_1}, d_{i_2}, \ldots, d_{i_{k-l}})$
  - $1, 2, \ldots, k$에서 $n_1+1, n_2+1, \ldots, n_l+1$번째 항이 지워진 shape
- 세 번째 인자 $keep_dim$이 $True$이면 $n_1+1, n_2+1, \ldots, n_l+1$번째 항은 삭제되지 않고 1로 남음

</td></tr>
<tr><td>Comment</td></tr>
<tr><td>

- 주어진 텐서에서 여러 축을 통합한 합/평균을 반환
- $[n_1, n_2, \ldots, n_l]$에 중복된 원소가 들어가도 상관없이 잘 작동함

</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{sum}, \texttt{mean}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e_1 = \texttt{if } 0 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[1]) \\ e_2 = \texttt{if } 1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[2]) \\ \quad \ldots \\ e_k = \texttt{if } k-1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[k]) \\ e' = e_1@e_2@\cdots@e_k \\ c' = \{(k \geq 1) \land (\forall i = 1, 2, \ldots, r,\ 0 \leq n_i < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, (n_1, n_2, \ldots, n_r)) \Rightarrow e', c \cup c'}$$

$$\forall \texttt{ft} \in \{\texttt{sum}, \texttt{mean}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e_1 = \texttt{if } 0 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } (1) \texttt{ else } (e[1]) \\ e_2 = \texttt{if } 1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } (1) \texttt{ else } (e[2]) \\ \quad \ldots \\ e_k = \texttt{if } k-1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } (1) \texttt{ else } (e[k]) \\ e' = e_1@e_2@\cdots@e_k \\ c' = \{(k \geq 1) \land (\forall i = 1, 2, \ldots, r,\ 0 \leq n_i < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, (n_1, n_2, \ldots, n_r), True) \Rightarrow e', c \cup c'}$$

`torch.prod`

<table>
<tr><td colspan="2"><code>torch.prod(input, dtype=None),</code>
<code>torch.prod(input, dim, keepdim=False, dtype=None)</code></td></tr>
<tr><td rowspan="2"></td><td>Comment</td></tr>
<tr><td>• 대부분 <code>sum</code>과 똑같음
• 하지만, dim 부분에 튜플값이 들어갈 수 없음. 즉 여러 축에 대한 곱을 계산할 수 없음</td></tr>
</table>

$$\frac{\sigma \vdash E \Rightarrow \_, c}{\sigma \vdash \texttt{prod}(E) \Rightarrow (), c}$$

$$\frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@e[n+2{:}k] \\ c' = \{(k \geq 1) \wedge (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{prod}(E, n) \Rightarrow e', c \cup c'}$$

$$\frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@(1)@e[n+2{:}k] \\ c' = \{(k \geq 1) \wedge (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{prod}(E, n, True) \Rightarrow e', c \cup c'}$$

`torch.var`, `torch.std`

<table>
<tr><td colspan="2"><code>torch.var(input, unbiased=True) or .std</code></td></tr>
<tr><td rowspan="6"></td><td>Require</td></tr>
<tr><td>• 텐서 타입이 floating이어야 함</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>• $|y| = ()$</td></tr>
<tr><td>Comment</td></tr>
<tr><td>• 주어진 텐서의 분산/표준편차를 ()-shape 텐서로 반환(스칼라 X)</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{var}, \texttt{std}\}, \quad \frac{\sigma \vdash E \Rightarrow \_, c}{\sigma \vdash \texttt{ft}(E) \Rightarrow (), c}$$

<table>
<tr><td colspan="2"><code>torch.var(input, dim, keepdim=False, unbiased=None, out=None)</code> or
<code>torch.std(input, dim, unbiased=True, keepdim=False, out=None)</code></td></tr>
<tr><td rowspan="6"></td><td>Require</td></tr>
<tr><td>• $|input| = (d_1, d_2, \ldots, d_k)$
• $k \geq 1$, $0 \leq dim < k$
• 텐서 타입이 floating이어야 함</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>• $|y| = (d_1, d_2, \ldots, d_{dim}, d_{dim+2}, \ldots, d_k)$
• $keepdim$이 $True$이면 $|y| = (d_1, d_2, \ldots, d_{dim}, 1, d_{dim+2}, \ldots, d_k)$</td></tr>
<tr><td>Comment</td></tr>
<tr><td>• 주어진 텐서에서 축 상의 분산/표준편차를 반환
• <code>sum</code>, <code>mean</code>과 마찬가지로 dim 부분은 튜플로 쓰일 수도 있음 (여러 축에 대한 분산/표준편차)</td></tr>
</table>

$$\forall \texttt{ft} \in \{\texttt{var}, \texttt{std}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@e[n+2{:}k] \\ c' = \{(k \geq 1) \land (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, n) \Rightarrow e', c \cup c'}$$

$$\forall \texttt{ft} \in \{\texttt{var}, \texttt{std}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e' = e[1{:}n]@(1)@e[n+2{:}k] \\ c' = \{(k \geq 1) \land (0 \leq n < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, n, True) \Rightarrow e', c \cup c'}$$

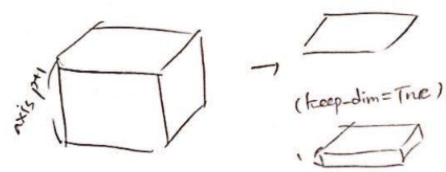
$keepdim$이 $True$인 상황에 대한 proof tree

$$\forall \texttt{ft} \in \{\texttt{var}, \texttt{std}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \texttt{rank}(e) \\ e_1 = \texttt{if } 0 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[1]) \\ e_2 = \texttt{if } 1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[2]) \\ \quad \cdots \\ e_k = \texttt{if } k-1 \in \{n_1, n_2, \ldots, n_r\} \texttt{ then } () \texttt{ else } (e[k]) \\ e' = e_1 @ e_2 @ \cdots @ e_k \\ c' = \{(k \geq 1) \land (\forall i = 1, 2, \ldots, r,\ 0 \leq n_i < k)\} \end{array}}{\sigma \vdash \texttt{ft}(E, (n_1, n_2, \ldots, n_r)) \Rightarrow e', c \cup c'}$$

$$
\forall \mathtt{ft} \in \{\mathtt{var}, \mathtt{std}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \mathtt{rank}(e) \\ e_1 = \mathtt{if}\ 0 \in \{n_1, n_2, \ldots, n_r\}\ \mathtt{then}\ (1)\ \mathtt{else}\ (e[1]) \\ e_2 = \mathtt{if}\ 1 \in \{n_1, n_2, \ldots, n_r\}\ \mathtt{then}\ (1)\ \mathtt{else}\ (e[2]) \\ \quad \cdots \\ e_k = \mathtt{if}\ k-1 \in \{n_1, n_2, \ldots, n_r\}\ \mathtt{then}\ (1)\ \mathtt{else}\ (e[k]) \\ e' = e_1 @ e_2 @ \cdots @ e_k \\ c' = \{(k \geq 1) \wedge (\forall i = 1, 2, \ldots, r,\ 0 \leq n_i < k)\} \end{array}}{\sigma \vdash \mathtt{ft}(E, (n_1, n_2, \ldots, n_r), True) \Rightarrow e', c \cup c'}
$$

$keepdim$이 $True$인 상황에 대한 proof tree

torch.mode, torch.median

<table>
<tr><td colspan="2"><code>torch.mode(input, dim=-1, keep_dim=False, out=None) or torch.median</code></td></tr>
<tr><td rowspan="6">
</td><td>Require</td></tr>
<tr><td>

- $|input| = (d_1, d_2, \ldots, d_k)$
- $k \geq 1$
- $0 \leq dim < k$ ($dim \leftarrow k-1$ if $dim = -1$)

</td></tr>
<tr><td>Guarantees</td></tr>
<tr><td>

- $|y| = |z| = (d_1, d_2, \ldots, d_{dim}, d_{dim+2}, \ldots, d_k)$인 (y, z) 튜플 반환
- 세 번째 인자 $keep_dim$이 $True$이면 $|y| = |z| = (d_1, d_2, \ldots, d_{dim}, 1, d_{dim+2}, \ldots, d_k)$

</td></tr>
<tr><td>Comment</td></tr>
<tr><td>

- 입력받은 축을 기준으로한 통계적 최빈값/중간값 계산 함수
- (결과값, 인덱스) 튜플 형태로 반환하기 때문에, `out`인자도 들어가는 텐서도 2-원소 튜플 형태로 들어감

</td></tr>
</table>

$$
\forall \mathtt{ft} \in \{\mathtt{mode}, \mathtt{median}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \mathtt{rank}(e) \\ e' = e[1{:}n] @ e[n+2{:}k] \\ c' = \{(k \geq 1) \wedge (0 \leq n < k)\} \end{array}}{\sigma \vdash \mathtt{ft}(E, n = -1, out = None) \Rightarrow (e', e'), c \cup c'}
$$

tuple 형태로 반환

$$
\forall \mathtt{ft} \in \{\mathtt{mode}, \mathtt{median}\}, \quad \frac{\begin{array}{l} \sigma \vdash E \Rightarrow e, c \\ k = \mathtt{rank}(e) \\ e' = e[1{:}n] @ (1) @ e[n+2{:}k] \\ c' = \{(k \geq 1) \wedge (0 \leq n < k)\} \end{array}}{\sigma \vdash \mathtt{ft}(E, n = -1, True, out = None) \Rightarrow (e', e'), c \cup c'}
$$

tuple 형태로 반환

$$
\forall \mathtt{ft} \in \{\mathtt{mode}, \mathtt{median}\}, \quad \frac{\sigma \vdash \mathtt{ft}(E, n) \Rightarrow (e, e), c}{\sigma \vdash \mathtt{ft}(E, n = -1, False, out = None) \Rightarrow (e, e), c}
$$

tuple 형태로 반환