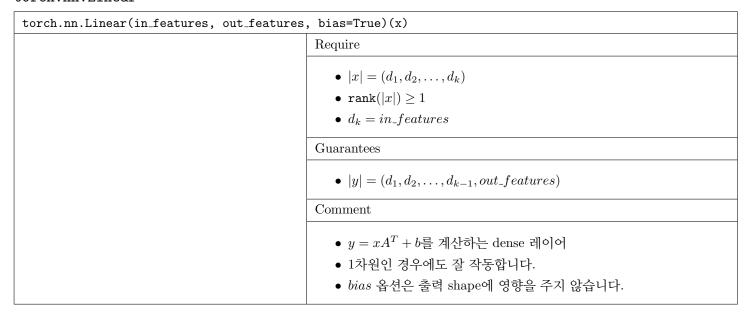
Matmul Layers

torch.nn.Linear



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

Activations

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

torch.nn.ReLU(inplace=True)(x)		
	Require	
	Guarantees	
	• $ y = x $ (same shape)	
	Comment	
	 inplace 옵션은 shape에 영향을 주지 않습니다. ReLU6도 ReLU와 똑같은 방식으로 shape 계산 	
	• Bulitins인 torch.relu와 torch.nn.functional.relu는 같은거	

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

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

End Points

torch.nn.CrossEntropyLoss

torch.nn.CrossEntropyLoss(weight=None, size_average=None, ignore_index=-100,
reduction='mean')(input, target)

Require

- $|input| = (n, c, d_1, d_2, \dots, d_k)$ $(rank(|input|) \ge 2)$
- $|target| = (n, d_1, d_2, \dots, d_k)$
 - $-\operatorname{rank}(|target|) + 1 = \operatorname{rank}(|input|)$
 - $|target|[1] = |input|[1], |target|[2] = |input|[3], \dots$
- if $weight \neq None$, then |weight| = (c)

Guarantees

• $|y| = \text{if } reduction == `none' \text{ then } (n, d_1, d_2, \dots, d_k) \text{ else } ()$

Comment

- size_average, reduce 인자는 deprecated로 도큐먼트되어 있습니다.
- *ignore_index*는 *reduction*이나 *size_average* 옵션에서 평균을 계산 하면서 생략하는 인덱스 번호를 지정할 때 쓰는 것으로, shape에 영향을 주지 않습니다.

$$\begin{split} \sigma \vdash E &\Rightarrow e, c_e \\ \sigma \vdash T \Rightarrow t, c_t \\ \sigma \vdash weight \Rightarrow w, c_w & \text{if } weight \neq None, \text{ otherwise } c_w = \emptyset \\ c_{dim} &= \{ (\texttt{rank}(e) \geq 2) \land (\texttt{rank}(e) = \texttt{rank}(t) + 1) \} \\ c_{elt} &= \{ (e[1] = t[1]) \land (e[3] = t[2]) \land (e[4] = t[3]) \land \cdots \} \\ c_{weight} &= \{ (weight = None \lor w = (e[2])) \} \\ e' &= \text{if } reduction = `none' \text{ then } t \text{ else } () \end{split}$$

 $\overline{\sigma \vdash \mathtt{CrossEntropyLoss}(weight = None, ..., reduction = `mean')(E, T) \Rightarrow e', c_e \cup c_t \cup c_w \cup c_{dim} \cup c_{elt} \cup c_{weight}}$

torch.nn.TripletMarginLoss

NOT DONE YET!!

Reading Papers...

- •	rgin=1.0, p=2.0, eps=1e-6, swap=False, size_average=False,
duce=None, reduction='mean')(anchor, positive, negative)
	Require
	$\bullet \ broadcastable(anchor , positive , negative)$
	$\bullet \ \mathtt{rank}(broadcast(anchor , positive , negative)) \geq 2$
	• if $swap$ is $True$ then,
	$- \hspace{0.1cm} \texttt{rank}(anchor), positive , negative)$
	$- \ \texttt{rank}(broadcast(anchor , positive , negative)) \geq 2$
	• $ target = (n, d_1, d_2, \dots, d_k)$
	$-\ \mathtt{rank}(target) + 1 = \mathtt{rank}(input)$
	$-\ target [1] = input [1],\ target [2] = input [3],\ \dots$
	• if $weight \neq None$, then $ weight = (c)$
	Guarantees
	• $ y = \text{if } reduction == `none' \text{ then } (n, d_1, d_2, \ldots, d_k) \text{ else } ()$
	Comment
	• size_average, reduce 인자는 deprecated로 도큐먼트되어 있습니

- *ignore_index*는 *reduction*이나 *size_average* 옵션에서 평균을 계산 하면서 생략하는 인덱스 번호를 지정할 때 쓰는 것으로, shape에 영향을 주지 않습니다.
- 토치 구현의 버그인지 모르겠는데, *swap*이 *True*인 경우가 너무 복잡합니다.. 도큐먼트에도 논문 하나만 첨부되어 있습니다.

```
\begin{split} \sigma \vdash E \Rightarrow e, c_e \\ \sigma \vdash T \Rightarrow t, c_t \\ \sigma \vdash weight \Rightarrow w, c_w & \text{if } weight \neq None, \text{ otherwise } c_w = \emptyset \\ c_{dim} = \{(\texttt{rank}(e) \geq 2) \land (\texttt{rank}(e) = \texttt{rank}(t) + 1)\} \\ c_{elt} = \{(e[1] = t[1]) \land (e[3] = t[2]) \land (e[4] = t[3]) \land \cdots \} \\ c_{weight} = \{(weight = None \lor w = (e[2]))\} \\ e' = \texttt{if } reduction = `none' \text{ then } t \text{ else } () \end{split}
```

 $\overline{\sigma \vdash \mathtt{CrossEntropyLoss}(weight = None, ..., reduction = `mean')(E, T) \Rightarrow e', c_e \cup c_t \cup c_w \cup c_{dim} \cup c_{elt} \cup c_{weight}}$

Technique

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

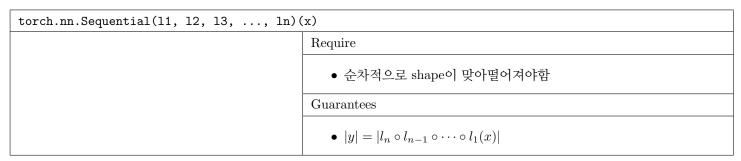
torch.nn.Dropout()(x)	
	Require
	Guarantees
	• $ y = x $ (same shape)
	Comment
	 모든 옵션은 shape에 영향을 주지 않습니다. Bulitins인 torch.dropout와 torch.nn.functional.dropout는 서로 역할이 같습니다.

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \mathtt{Dropout}(...)(E) \Rightarrow e, c}$$

$$\frac{\sigma \vdash E \Rightarrow e, c}{\sigma \vdash \mathtt{dropout}(E, \ldots) \Rightarrow e, c}$$

Wrapper

torch.nn.Sequential



$$\frac{\sigma \vdash l_n \circ l_{n-1} \circ \cdots l_1(E) \Rightarrow e, c}{\sigma \vdash \mathtt{Sequential}(l_1, l_2, \dots, l_n)(E) \Rightarrow e, c}$$