P1 ReLU는 nonnegative homogeneous하다. (by midterm)

LeakyReLU는 nonnegative homogeneous하다 증명

LeakyReLU

$$\sigma(x) = \begin{cases} ax & x<0 \\ x & x\geq 0 \end{cases}$$

$\alpha \geq 0$이면 $\sigma(\alpha x) = \begin{cases} a\alpha x & x<0 \\ \alpha x & x\geq 0 \end{cases}$ 이므로 $\sigma(\alpha x) = \alpha\sigma(x)$

dropout은 element에 0이 곱해지거나 $\frac{1}{1-p}$이 곱해지는 것이므로 σ가 nonnegative homogeneous하면

dropout-σ 와 σ-dropout은 equivalent하다.

Sigmoid의 경우 linear를 통과한 벡터의 element 하나를 고려하면

	dropout $\to\sigma$		$\sigma\to$ dropout
non-drop	$\dfrac{1}{1+e^{-\frac{x}{1-p}}}$	$\neq$	$\dfrac{1}{(1-p)(1+e^{-x})}$
drop	$\dfrac{1}{1+e^{-0}} = \dfrac{1}{2}$	$\neq$	0

Sigmoid는 순서를 바꾸면 equivalent하지 않다.

P2 default initialization은

$$A_l \sim \text{Uniform}\left(-\frac{1}{\sqrt{n_{l-1}}}, \frac{1}{\sqrt{n_{l-1}}}\right) \quad \text{mean}: 0, \ \text{Var}: \frac{1}{3n_{l-1}}$$

$$b_l \sim \text{Uniform}\left(-\frac{1}{\sqrt{n_{l-1}}}, \frac{1}{\sqrt{n_{l-1}}}\right) \quad \text{mean}: 0, \ \text{Var}: \frac{1}{3n_{l-1}}$$

$$E[y_1] = E[A_1 x + b_1] = E[A_1]E[x] + E[b_1] = 0$$

$E[y_l] = 0$ 이라 가정하면

$$E[y_{l+1}] = E[A_l]E[y_l] + E[b_l] = 0$$

즉, $E[y_L]$도 0이다.

$$\text{Var}(y_{l_i}) = \text{Var}\left(\sum_{j=1}^{n_{l-1}} A_{l_{ij}} y_{l-1_j}\right) + \text{Var}(b_{l_i})$$

$$= \sum_{j=1}^{n_{l-1}} \text{Var}(A_{l_{ij}} y_{l-1_j}) + \text{Var}(b_{l_i})$$

A_{ij}와 y_j 모두 zero mean이고 서로 독립이므로

$$= \sum_{j=1}^{n_{l-1}} \text{Var}(A_{l_{ij}}) \text{Var}(y_{l-1_j}) + \text{Var}(b_{l_i})$$

$$= n_{l-1} \cdot \frac{1}{3n_{l-1}} \text{Var}(y_{l-1_j}) + \frac{1}{3n_{l-1}}$$

$$= \frac{1}{3} \text{Var}(y_{l-1}) + \frac{1}{3n_{l-1}}$$

$$\text{Var}(y_1) = \frac{1}{3} \cdot 1 + \frac{1}{3n_0}$$

$$\therefore \text{Var}(y_L) = \frac{1}{3^L} + \sum_{k=0}^{L-1} \frac{1}{3^{L-k} \cdot n_k}$$

$$\text{Var}(y_2) = \frac{1}{3}\text{Var}(y_1) + \frac{1}{3n_1}$$

$$\text{Var}(y_3) = \frac{1}{3}\text{Var}(y_2) + \frac{1}{3n_2}$$

$$\vdots$$

$$\text{Var}(y_L) = \frac{1}{3n_{L-1}} + \frac{1}{3 \cdot 3n_{L-2}} + \cdots + \frac{1}{3^{L-2} \cdot 3n_1} + \frac{1}{3^{L-2}} \cdot \frac{1}{3}\left(\frac{1}{3} + \frac{1}{3n_0}\right)$$

P3. (i) $y_l = \sigma(A_l y_{l-1} + b_l) + y_{l-1}$

$y_l^* = \sigma(A_l y_{l-1} + b_l)$ 이라하면

$y_{l\,i}^* = \sigma\left(\sum_{j=1}^{m} A_{l\,ij}\, y_{l-1\,j} + b_{l\,i}\right)$ 이므로

$$\left(\frac{\partial y_l^*}{\partial y_{l-1}}\right)_{ij} = \sigma'\left(\sum_{j=1}^{m} A_{l\,ij}\, y_{l-1\,j} + b_{l\,i}\right) \cdot A_{l\,ij}$$

$$\left(\frac{\partial y_l^*}{\partial y_{l-1}}\right) = \mathrm{diag}\left(\sigma'(A_l y_{l-1} + b_l)\right) A_l$$

$$y_l = y_l^* + y_{l-1}$$

$$\frac{\partial y_l}{\partial y_{l-1}} = \mathrm{diag}\left(\sigma'(A_l y_{l-1} + b_l)\right) A_l + I$$

(ii) $$\frac{\partial y_L}{\partial b_l} = \frac{\partial y_L}{\partial y_l}\frac{\partial y_l}{\partial b_l}$$

$$= \frac{\partial y_L}{\partial y_l}\cdot \operatorname{diag}\left(\sigma'(A_l y_{l-1} + b_l)\right)$$

$$\left(\frac{\partial y_L}{\partial A_l}\right)_{ij} = \frac{\partial y_L}{\partial y_l}\frac{\partial y_l}{(\partial A_l)_{ij}} \quad \left(\rightarrow \begin{bmatrix} 0 \\ \vdots \\ \sigma'((A_l y_{l-1}+b_l)_i)\, y_{l-1,j} \\ \vdots \\ 0 \end{bmatrix} \right)$$

$$= \left(\frac{\partial y_L}{\partial y_l}\right)_i \sigma'\left((A_l y_{l-1} + b_l)_i\right)(y_{l-1})_j$$

$$\therefore \frac{\partial y_L}{\partial A_l} = \operatorname{diag}\left(\sigma'(A_l y_{l-1} + b_l)\right)\left(\frac{\partial y_L}{\partial y_l}\right)^T y_{l-1}^T$$

(iii) $\frac{\partial y_l}{\partial y_{l-1}}$에 $\underbrace{\operatorname{diag}\left(\sigma'(A_l y_{l-1} + b_l)\right)A_l}$ 항 외에 I가 포함되어

있으므로 $\frac{\partial y_L}{\partial y_l}$도 need not vanish하다.

$\frac{\partial y_L}{\partial b_i}$, $\frac{\partial y_L}{\partial A_i}$ 모두 $\frac{\partial y_L}{\partial y_l}$항을 포함하고 있다.

P4

(a)

	My ConvLayer	STM ConvLayer:
Conv1	$256\times128+128$	$256\times4\times32+4\times32$
Conv2	$128\times128\times9+128$	$4\times4\times9\times32+4\times32$
Conv3	$128\times256+256$	$4\times256\times32+256\times32$
total	213504	78592

(b)

```python
import torch
import torch.nn as nn


class STMConv(nn.Module):
    def __init__(self):
        super(STMConv, self).__init__()
        self.layer1 = nn.ModuleList([nn.Conv2d(256, 4, kernel_size=1, stride=1) for _ in range(32)])
        self.layer2 = nn.ModuleList([nn.Conv2d(4, 4, kernel_size=3, stride=1, padding=1)for _ in range(32)])
        self.layer3 = nn.ModuleList([nn.Conv2d(4, 256, kernel_size=1, stride=1)for _ in range(32)])
    def forward(self, x):
        result = []
        for i in range(32):
            x = torch.nn.functional.relu(self.layer1[i](x))
            x = torch.nn.functional.relu(self.layer2[i](x))
            x = torch.nn.functional.relu(self.layer3[i](x))
            result.append(x)
        return sum(result)
```