MFDNN

Homework 6

24/04/18

$\rightarrow$ i.e. not for Sigmoid (since not homogeneous: $\frac{1}{1+e^{-5x}} \neq \frac{5}{1+e^{-x}}$

1. Equivalent only for (a) and (c) since these are non-negative homogeneous, preserving the 0 value dropout can output with probability p.

Explicitly, for some individual neuron value $x \in \mathbb{R}$,

If $x > 0$, $\text{dropout}(\text{ReLU}(x)) = \begin{cases} 0 & \text{with probability } p \\ \frac{x}{1-p} & \text{otherwise} \end{cases}$; $\text{ReLU}(\text{dropout}(x)) = \text{ReLU}\left(\begin{cases} 0 & \text{with prob } p \\ \frac{x}{1-p} & \text{otherwise} \end{cases}\right)$

equal since $\frac{x}{1-p} > 0$ still

If $x \leq 0$, $\text{dropout}(\overbrace{\text{ReLU}(x)}^{0}) = 0 = \text{ReLU}(\overbrace{\text{dropout}(x)}^{0 \text{ or } x \leq 0})$

Equivalently for LeakyReLU,

If $x > 0$, $\text{dropout}(\text{LeakyReLU}(x)) = \begin{cases} 0 & \text{''} \\ \frac{x}{1-p} & \text{''} \end{cases}$; $\text{LeakyReLU}(\text{dropout}(x)) = \text{LeakyReLU}\left(\begin{cases} 0 & \text{''} \\ \frac{x}{1-p} & \text{''} \end{cases}\right)$

equal since $\frac{x}{1-p} > 0$ still

If $x \leq 0$, $\text{dropout}(\text{LeakyReLU}(x)) = \begin{cases} 0 & \text{''} \\ \frac{\alpha x}{1-p} & \text{''} \end{cases}$

$$\text{LeakyReLU}(\text{dropout}(x)) = \text{LeakyReLU}\left(\begin{cases} 0 & \text{''} \\ \frac{x}{1-p} & \text{''} \end{cases}\right) = \begin{cases} 0 & \text{''} \\ \frac{\alpha x}{1-p} & \text{''} \end{cases} \quad \text{since } \frac{x}{1-p} \leq 0 \text{ too}$$

Meanwhile, $\text{Sigmoid}(0) = 0.5$

So $\text{sigmoid}(\text{dropout}(x)) = \begin{cases} 0.5 & \text{''} \\ \text{sigmoid}\left(\frac{x}{1-p}\right) & \text{''} \end{cases}$

clearly not equal

$\text{dropout}(\text{sigmoid}(x)) = \begin{cases} 0 & \text{''} \\ \frac{\text{sigmoid}(x)}{1-p} & \text{''} \end{cases}$

2. PyTorch defaults: $(A_\ell)_{ij} \sim \mathcal{U}\left(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$ ← a, ← b

uniform dist.

$(b_\ell)_i \sim \mathcal{U}\left(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$ where n = num input features (x dimension)

$1 \le i \le n_\ell$, $1 \le j \le n_{\ell-1}$

$$\mu_{uniform} = \frac{1}{2}(a+b) = 0$$
$$\sigma^2_{uniform} = \frac{1}{12}(b-a)^2 = \frac{1}{12}\left(\frac{2}{\sqrt{n}}\right)^2 = \frac{1}{3n}$$

For R.V. W either $(A_\ell)_{ij}$ or $(b_\ell)_i$, $\mathbb{E}[W]=0$, $\mathrm{Var}[W] = \frac{1}{3n_{\ell-1}} = \mathbb{E}[W^2]$

$\mathbb{E}[x_i]=0$, $\mathbb{E}[x_i^2]=1$

$$\mathbb{E}[y_L] = \mathbb{E}[A_L y_{L-1} + b_L] = \mathbb{E}\left[\sum_k^{n_{L-1}} (A_L)_{1k}(y_{L-1})_k\right] + \mathbb{E}[b_L]$$
$$= \sum_k^{n_{L-1}} \underbrace{\mathbb{E}[(A_L)_{1k}]}_{0}\mathbb{E}[(y_{L-1})_k] + 0 \quad \text{by assumed indp. of } A \,\&\, y_{L-1}$$
$$= 0 \quad \Rightarrow \mu(y_L) = 0$$

$$\mathrm{Var}[y_1] = \mathbb{E}[y_1^2] - \underbrace{\mathbb{E}[y_1]^2}_{0}$$
$$= \sigma_{A_1}^2 n_0 \sigma_{y_0}^2 + \sigma_{b_1}^2 \quad \text{by lecture slides chp. 3 slide 70}$$
$$= \frac{1}{3n_0}\cdot n_0 \cdot 1 + \frac{1}{3n_0}$$
$$= \frac{1}{3n_0} + \frac{1}{3}$$

$$\mathrm{Var}[y_2^2] = \mathbb{E}[y_2^2]$$
$$= \sigma_{A_2}^2 n_1 \sigma_{y_1}^2 + \sigma_{b_2}^2$$
$$= \frac{1}{3n_1}\cdot n_1 \cdot \left(\frac{1}{3n_0} + \frac{1}{3}\right) + \frac{1}{3n_1}$$
$$= \frac{1}{3n_1} + \frac{1}{3}\left(\frac{1}{3n_0} + \frac{1}{3}\right)$$

$$\mathrm{Var}[y_3^2] = \sigma_{A_3}^2 n_2 \sigma_{y_2}^2 + \sigma_{b_3}^2$$
$$= \frac{1}{3n_2}\cdot n_2 \cdot \left(\frac{1}{3n_1} + \frac{1}{3}\left(\frac{1}{3n_0} + \frac{1}{3}\right)\right) + \frac{1}{3n_2}$$
$$= \frac{1}{3n_2} + \frac{1}{3}\left(\frac{1}{3n_1} + \frac{1}{3}\left(\frac{1}{3n_0} + \frac{1}{3}\right)\right)$$

So $\mathrm{Var}[y_L] = \frac{1}{3n_{L-1}} + \frac{1}{3}\left(\frac{1}{3n_{L-2}} + \frac{1}{3}\left(\ldots + \frac{1}{3}\left(\frac{1}{3n_1} + \frac{1}{3}\left(\frac{1}{3n_0} + \frac{1}{3}\right)\right)\ldots\right)\right)$?

Assume true for $\ell = k$,

$$\mathrm{Var}[y_{k+1}] = \sigma_{A_{k+1}}^2 n_k \sigma_{y_k}^2 + \sigma_{b_{k+1}}^2$$
$$= \frac{1}{3n_k}\cdot n_k \cdot \left(\frac{1}{3n_{k-1}} + \frac{1}{3}\left(\frac{1}{3n_{k-2}} + \frac{1}{3}\left(\ldots + \frac{1}{3}\left(\frac{1}{3n_1} + \frac{1}{3}\left(\frac{1}{3n_0} + \frac{1}{3}\right)\right)\ldots\right)\right)\right) + \frac{1}{3n_k}$$
$$= \frac{1}{3n_k} + \frac{1}{3}\left(\frac{1}{3n_{k-1}} + \quad '' \quad \ldots\right)\right)$$

which is expected form so true by induction?

3. (i) $\frac{\partial y_L}{\partial y_{L-1}} = A_L$

$$\frac{\partial (y_\ell)_i}{\partial (y_{\ell-1})_j} = \frac{\partial}{\partial (y_{\ell-1})_j}\left(\sigma\left(\sum_k^m (A_\ell)_{ik}(y_{\ell-1})_k + (b_\ell)_i\right) + (y_{\ell-1})_i\right)$$
$$= \sigma_i'(A_\ell y_{\ell-1} + b_\ell)(A_\ell)_{ij} + \delta_{ij}$$

For $\ell = 2, \ldots, L-1$

$$\frac{\partial y_\ell}{\partial y_{\ell-1}} = \frac{\partial}{\partial y_{\ell-1}}\left(\sigma(A_\ell y_{\ell-1} + b_\ell) + y_{\ell-1}\right)$$

$$= \operatorname{diag}(\underbrace{\sigma'(A_\ell y_{\ell-1} + b_\ell)}_{\text{by HW 4.6}})\overset{m\times m}{A_\ell} + \overset{m\times m}{I}$$ where $I \in \mathbb{R}^{m\times m}$ is the identity matrix

(ii) $$\frac{\partial y_L}{\partial b_\ell} = \frac{\partial y_L}{\partial y_\ell}\frac{\partial y_\ell}{\partial b_\ell} = \frac{\partial y_L}{\partial y_{L-1}}\frac{\partial y_{L-1}}{\partial y_{L-2}}\cdots\frac{\partial y_{\ell+1}}{\partial y_\ell}\frac{\partial y_\ell}{\partial b_\ell}$$ by the chain rule

$$= \quad " \quad " \quad \cdots \quad " \quad \operatorname{diag}(\sigma'(A_\ell y_{\ell-1} + b_\ell))$$ once again by HW 4.6 since b_ℓ differentiated not affected by residual connections (which only include 'earlier' b_i's)

for $\ell = 1, \ldots, L-1$. And $\frac{\partial y_L}{\partial b_L} = 1$.

$$\frac{\partial y_L}{\partial A_\ell} = \operatorname{diag}(\sigma'(A_\ell y_{\ell-1} + b_\ell))\left(\frac{\partial y_L}{\partial y_\ell}\right)^T y_{\ell-1}^T$$ by HW 4.6 (same logic)

for $\ell = 1, \ldots, L-1$. And, $\frac{\partial y_L}{\partial A_L} = y_{L-1}^T$

(iii) $$\frac{\partial y_L}{\partial b_i} = \frac{\partial y_L}{\partial y_{L-1}}\overset{j=L-1}{\frac{\partial y_{L-1}}{\partial y_{L-2}}}\cdots\overset{j=\ell+1}{\frac{\partial y_{\ell+1}}{\partial y_\ell}}\frac{\partial y_\ell}{\partial y_{\ell-1}}\cdots\frac{\partial y_{i+1}}{\partial y_i}\frac{\partial y_i}{\partial b_i}$$

If $A_j = 0$ for some $j \in \{\ell+1, \ldots, L-1\}$
or $\sigma'(A_j y_{j-1} + b_j) = 0$ for " , in a non-residual network one of the orange terms would be 0 meaning that $\frac{\partial y_L}{\partial b_i} = 0$ (would vanish). But the residual connection introduces the identity matrix addition so the jth term doesn't vanish and instead becomes the identity so no vanishing occurs.
multiplying by 0

Similarly for $\frac{\partial y_L}{\partial A_i}$: its expr also includes this $\frac{\partial y_L}{\partial y_\ell}$ term containing the orange potentially vanished terms in a regular network, but with the residual connection the identity prevents vanishing.

4.(a) Convolutional layer num parameters: $(C_{in} k^2 + 1) C_{out}$ ← bias

Original

$(256 \times 1^2 + 1) \times 128 +$
$(128 \times 3^2 + 1) \times 128 +$
$(128 \times 1^2 + 1) \times 256$
$= 213{,}504$

Split-transform-merge

$((256 \times 1^2 + 1) \times 4 +$
$(4 \times 3^2 + 1) \times 4 +$
$(4 \times 1^2 + 1) \times 256) \times 32$
$= 78{,}592$

(b)

```
...
super(STMConvLayer, self).__init__()
self.conv1 = nn.Conv2d(256,4,1)
self.conv2 = nn.Conv2d(4,4,3, padding=1)
self.conv3 = nn.Conv2d(4,256,1)
def forward(self, x):
    out = torch.zeroes(x.shape)
    for path_ in range(32):
        path_out = nn.functional.relu(self.conv1(x))
        path_out = nn.functional.relu(self.conv2(path_out))
        path_out = nn.functional.relu(self.conv3(path_out))
        out += path_out
    return out
```

5. (Workings)

'Training' (LA)

Min when $\nabla_\theta \left(\frac{1}{2} \| \tilde{X}\underline{\theta} - \underline{y} \|^2 + \frac{\lambda}{2} \| \underline{\theta} \|^2 \right) = \underline{0}$ $\qquad \tilde{x}_i := \mathrm{ReLU}(W x_i)$

Optimal $\underline{\theta}^*$,

$$\tilde{X}^T(\tilde{X}\underline{\theta}^* - \underline{y}) + \lambda \underline{\theta}^* = \underline{0} \quad \text{by HW1.1}$$

$$\tilde{X}^T\tilde{X}\underline{\theta}^* - \tilde{X}^T\underline{y} + \lambda \underline{\theta}^* = \underline{0}$$

$$(\tilde{X}^T\tilde{X} + \lambda I)\,\underline{\theta}^* = \tilde{X}^T\underline{y}$$ → tensor.linalg.solve(...)

$$\Rightarrow \underline{\theta}^* = (\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T\underline{y}$$