Hand-written Part

学号: t11902210 姓名: 张一凡 (交换生)

code question report 在后面

Problem 1

Swish is an activation function that can be viewed as a "modification" of the logistic function. Swish has been shown to be better than ReLU in some deep learning models. Recall that the logistic function is defined as

$$\theta(s) = \frac{1}{1+\exp(-s)}$$

Now Swish is defined as $\varphi(s) = s \cdot \theta(s)$

What is $\varphi'(s)$?

解:

$$\varphi(s) = s \cdot \theta(s)$$

$$\varphi'(s) = \theta(s) + s \cdot \frac{d(\theta(s))}{ds} \quad \Leftarrow \quad \frac{d(\theta(s))}{ds} = \theta'(s) = \left(\frac{1}{1+e^{-s}}\right)'$$

$$= \frac{1}{1+e^{-s}} + \frac{s \cdot e^{-s}}{(1+e^{-s})^2} \qquad = -\frac{e^{-s}\cdot(-1)}{(1+e^{-s})^2} = \frac{e^{-s}}{(1+e^{-s})^2}$$

$$= \frac{1+e^{-s}+s\cdot e^{-s}}{(1+e^{-s})^2}$$

$$= \frac{1+(s+1)e^{-s}}{(1+e^{-s})^2} \qquad \varphi'(s) = \frac{1+(s+1)e^{-s}}{(1+e^{-s})^2}$$

Problem 2

In the class, we briefly talked about the famous PageRank algorithm, which assigns a rank to each web page. Given a set of pages: $P = \{P_1, P_2, \dots, P_n\}$ and their incoming links: $in(P_j) | P_j$ has a link to $P_i, i \neq j\}$. The basic idea is that a web page should be ranked higher if it has 1) more incoming hyperlinks and 2) incoming links from high-ranked pages. The PageRank can be computed through the following iterative algorithm.

1. Initialize $PageRank_0(P_i) = \frac{1}{n}$ for each page.
2. Compute the updated PageRank for each page: $PageRank_1(P_i) = \sum_{P_j \leftarrow in(P_j)} \frac{PageRank_0(P_j)}{|out(P_j)|}$
   where $|out(P_j)|$ denotes the out-degree of P_j.
3. Repeat step 2 until the ranks converge.

Now, consider the following example with 3 pages:

$P = \{P_1, P_2, P_3\}$ $in(P_1) = \{P_2, P_3\}$ $in(P_2) = \{P_3\}$ $in(P_3) = \{P_1\}$

We define the following variables:

$$V_t = (PageRank_t(P_1), PageRank_t(P_2), PageRank_t(P_3))^T$$

$$P = \begin{bmatrix} 0 & 1 & 0.5 \\ 0 & 0 & 0.5 \\ 1 & 0 & 0 \end{bmatrix}.$$

where V_t is the ranks after the t-th iteration, and P is the transition matrix between pages. We can rewrite the update rule above as $v_t = Pv_{t-1}$. Answer the following questions:

(A) Based on the iterative algorithm above, please compute the values of v_1, v_2, v_3, v_4 and v_5.

Ans. according to the alegrithm.

Step 1: $P = \{P_1, P_2, P_3\} \Rightarrow n = 3 \Rightarrow V_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$

initialize PageRank$_0(P_i) = \frac{1}{3}$ $(i = 1, 2, 3)$ $\Rightarrow V_0 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}$

Step 2-1: $V_{01} = P \cdot V_{0-1} = PV_0 = \begin{pmatrix} 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{6} \\ \frac{1}{3} \end{pmatrix}$

Step 2-2: $V_2 = PV_1 = \begin{pmatrix} 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{1}{6} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{6} \\ \frac{1}{3} \end{pmatrix}$

Step 2-3: $V_3 = PV_2 = \begin{pmatrix} 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{6} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{5}{12} \\ \frac{1}{4} \\ \frac{1}{3} \end{pmatrix}$

Step 2-4: $V_4 = PV_3 = \begin{pmatrix} 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{5}{12} \\ \frac{1}{4} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{5}{12} \\ \frac{1}{6} \\ \frac{5}{12} \end{pmatrix}$

Step 2-5 $V_5 = PV_4 = \begin{pmatrix} 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{5}{12} \\ \frac{1}{6} \\ \frac{5}{12} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{5}{24} \\ \frac{5}{12} \end{pmatrix}$

In conclusion
$$\begin{cases} V_{01} = (\frac{1}{2}, \frac{1}{6}, \frac{1}{3})^T \\ V_2 = (\frac{1}{3} \ \frac{1}{6} \ \frac{1}{2})^T \\ V_3 = (\frac{5}{12} \ \frac{1}{4} \ \frac{1}{3})^T \\ V_4 = (\frac{5}{12} \ \frac{1}{6} \ \frac{5}{12})^T \\ V_5 = (\frac{3}{8} \ \frac{5}{24} \ \frac{5}{12})^T \end{cases}$$

(B) Recall that in the class, we mentioned that the algorithm converges when $v^* = Pv^*$ solve this equation to derive V^*. Note that you should provide a normalized v^* as answer.

解：设 $V^*=\alpha\begin{pmatrix}v_1\\v_2\\v_3\end{pmatrix}$ (非0) $PV^*=V^* \Rightarrow (P-E)V^*=0$ (E为单位阵)

$$P=\begin{pmatrix}0&1&\frac{1}{2}\\0&0&\frac{1}{2}\\1&0&0\end{pmatrix} \qquad (P-E)=\begin{pmatrix}-1&1&\frac{1}{2}\\0&-1&\frac{1}{2}\\1&0&-1\end{pmatrix}$$

$$|P-E|=\begin{vmatrix}0&1&\frac{1}{2}\\0&-1&\frac{1}{2}\\1&0&-1\end{vmatrix}=\begin{vmatrix}0&1&-\frac{1}{2}\\0&-1&\frac{1}{2}\\1&0&0\end{vmatrix}=\begin{vmatrix}1&-\frac{1}{2}\\-1&\frac{1}{2}\end{vmatrix}=1\neq 0$$

$$(P-E)V^*=\alpha\begin{pmatrix}-v_1+v_2+\frac{1}{2}v_3\\-v_2+\frac{1}{2}v_3\\v_1-v_3\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

$$\Rightarrow\begin{cases}-v_1+v_2+\frac{1}{2}v_3=0\\-v_2+\frac{1}{2}v_3=0\\v_1-v_3=0\end{cases}\Rightarrow\begin{cases}2v_2=v_3\\v_1=v_3\end{cases}$$

$$V^*=\alpha\begin{pmatrix}2v_2\\v_2\\2v_2\end{pmatrix}=\alpha\cdot v_2\begin{pmatrix}2\\1\\2\end{pmatrix}$$

V^* ⇒ normalized $\sqrt{2^2+1^2+2^2}=3$

则得 $V^*=\begin{pmatrix}\frac{2}{3}\\\frac{1}{3}\\\frac{2}{3}\end{pmatrix}$

Problem 3

For a $d^{(0)}-d^{(1)}-d^{(2)}-\cdots-d^{(L)}$ fully-connected neural network, the total number of neurons is $D=\sum_{l=1}^{L}d^{(l)}$ and the total number of weight is $\sum_{l=1}^{L}d^{(l-1)}d^{(l)}$ which ignore the so-called 'bias' terms in the network. Let the number of inputs $d^{(0)}=10$ and the total number of neurons $D=100$. Over all possible choices of L, $d^{(1)}, d^{(2)}, \cdots d^{(L)}$. What is the maximum total number of weights (and what neural network architecture does it correspond to)? What is the minimum total number of weights (and what neural network architecture does it correspond to)?

Consider $L\in\{1,2,3\}$ and

解：maximum：选用较大的 $d(l)$ 和较大 L. ~~使每层 $d(1)=d(2)=\cdots=d(L-1)=100$~~

$\sum_{l=1}^{L}d^{(l-1)}d^{(l)}=$ ~~$10\cdot 10\cdot\cdots\cdot 100$~~ L

$d(0)d(1)+d(1)d(2)+\cdots+d(L-1)d(L)$

整一个式子 $a+b=100$ $(a>0, b>0)$

$10000=(a+b)^2=a^2+2ab+b^2\geq a^2+b^2$.

因此在总 D 数之情况下，分层越多 $\sum_{l=1}^{L}d^{(l-1)}d^{(l)}$ 越小，

~~最大情况下的设计都为 $d(1)$~~ 但是时 $d(0)$ 给定，我们还能分别讨论几个 ($L=1,2,3$)

$L\in\{1,2,3\}$

直接考虑所有

① $L=1$ $\sum_{l=1}^{L}d^{(l-1)}d^{(l)}=10\times 100=1000$

② $L=2$ 时 $\sum_{i=1}^{2} d(i-1)d(i) = d(0)d(1)+d(1)d(2)$

设 $d(1)=x$, $d(2)=y$ $w_{max} = 10x+xy = 10x+x(100-x)$

$= -x^2+110x = -(x-55)^2+3025$

≤ 3025

③ $L=3$ 时 $w_{max} = 10d(1)+d(1)d(2)+d(2)d(3)$

设 $x.y.z = d_1.d_2.d_3$ $x+y+z=100$

$w = 10x+xy+yz = 10x+xy+(100-x-y)y$

$= 10x+100y-y^2 = 10\cdot(100-y-z)$

$+xy-y^2$

$= -(y-50)^2+2500+10x$

$= -(y+x-100)100 - y^2-10z+90y$

$= 3025-(y-45)^2-10z$

$= 3025-0-10 = 3015$

$d(1)>0$

$\frac{\partial w}{\partial x} = 10-y>0$ $\frac{\partial w}{\partial y} = -y+90>0$

有 z 取最小值 $z\geq 0$. $y=45$. $z=1$

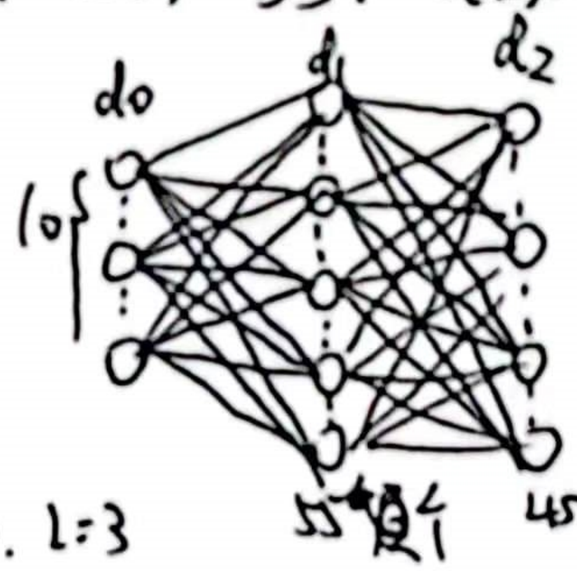
$x=54$

$1000 < 3025$

$3015 <$

因此在 $L=2$ 时 设 $d(1)=55$. $d(2)=45$ 取最大值

神经网络架构:



$\sum_{i=1}^{L} d(i-1)d(i) = 3025$

(最大值)

minimize (最小值): 也就是 $L=1$. $L=2$. $L=3$

① $L=1$ 时. 与求最大值相同 $\sum_{i=1}^{1} d(i-1)d(i) = 10\times 100 = 1000$

② $L=2$ 时 同样设 $d(1)=x$ $d(2)=y$

$w_{min} = -(x-55)^2+3025$ 求 $|x-55|$ 最大

$x>0$ $x=1$ $w_{min} = -(1-55)^2+3025 = -1+110\times 1 = -1+110 = 109$

③ $L=3$ 时 同样设 $d(x.y.z) = d_1.d_2.d_3$

$w = -(y-45)^2-10z+3025$ $x.y.z>0$

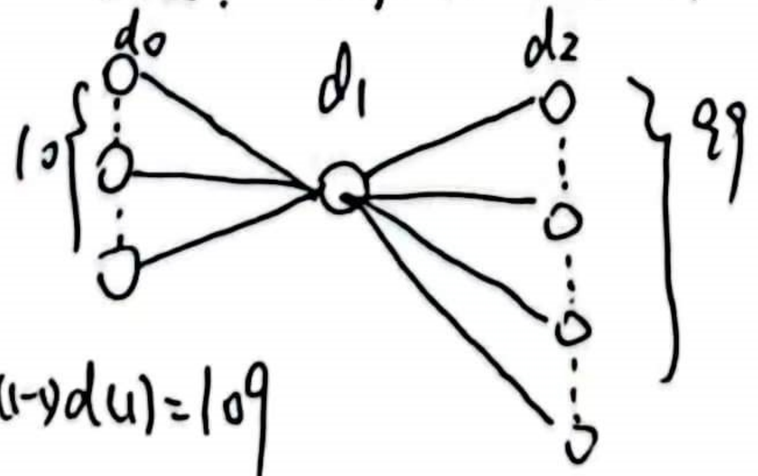
z 的系数 $-10<0$, $|y-45|$ 取大值. z 取大值 $x=1$. $y=1$

$w = 10\times 1+1\times 1+1\times 98 = 109$ $z = 100-2 = 98$

$1000 > 109$ 故 $=109$ 最小值有两种情况

情况1: $L=2$. $d(1)=1$. $d(2)=99$

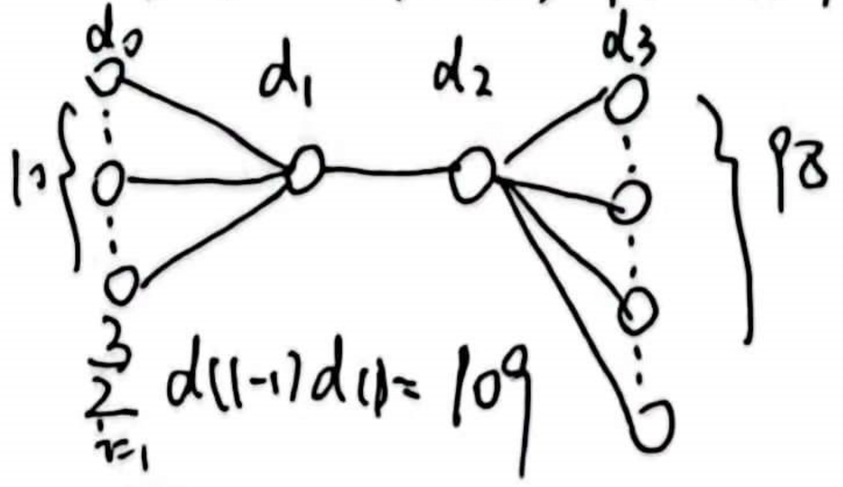
网络架构:



$\sum_{i=1}^{2} d(i-1)d(i) = 109$

情况2: $L=3$ $d(1)=d(2)=1$. $d(3)=98$

架构:



$\sum_{i=1}^{3} d(i-1)d(i) = 109$