1.1.1. Sigmoid

$x \to h_1 \to \cdots \to h_n \to y.$

$h_1 = \sigma(w_1 x + b_1)$

$h_2 = \sigma(w_2 h_1 + b_2)$

$\cdots$

$y = \sigma(w_n h_{n-1} + b_{n-1})$

$w_1 = w_2 = \cdots = w_n = 1$

if $n=1$. $y = \sigma(w_1 x + b_1) = \sigma(x+b_1)$

$dy/dx = \sigma(x+b_1)(1-\sigma(x+b_1))$

$= h_1 \cdot (1-h_1)$

$\because h_1$ is a sigmoid func, $0 < h_1 < 1$. h_1' is max when $x = \frac{1}{2}$

$\therefore 0 \le |dy/dx| \le \frac{1}{4} \Rightarrow |h_1'| \le 1/4$

for $n = i$. $y = \sigma(h_{i-1} + b_{i-1}) = h_i$.

$dy/dx = \sigma(h_{i-1}+b_{i-1})(1-\sigma(h_{i-1}+b_{i-1}))\cdot h_{i-1}'$

$0 \le |dy/dx| = |dh_i/dx| \le \frac{1}{4} h_{i-1}'$

$\Rightarrow$ By induction $\Rightarrow 0 \le |dy/dx| = |dh_n/dx| \le \left(\frac{1}{4}\right)^n$

For dh_1/dx, we have $0 \le |dh_1/dx| \le 1/4$

$\Rightarrow \left|\frac{dh_n}{dh_1}\right| = \left|\frac{dh_n}{dx}\cdot\frac{dx}{dh_1}\right| \Rightarrow 0 \le \frac{dh_n}{dh_1} \le (1/4)^{n-1}$ as $n \to \infty$. $(1/4)^{n-1} \to 0$

$\lim_{n\to\infty} 0 \le \lim_{n\to\infty}\left|\frac{dh_n}{dh_1}\right| \le \lim_{n\to\infty}(1/4)^{n-1} \Rightarrow 0 \le \lim_{n\to\infty}\left|\frac{dh_n}{dh_1}\right| \le 0$

$\Rightarrow \lim_{n\to\infty}\left|\frac{dh_n}{dh_1}\right| = 0 \Rightarrow$ The gradient vanishes for sigmoid activation

1.1.2. Tanh.

$x \to h_1 \to h_2 \cdots \to h_n \to y.$

$h_1 = \tanh(x+b_1)$

$\cdots$

$y = \tanh(h_{n-1} + b_{n-1})$

$n=1$

$dh_1/dx = 1 - \tanh^2(x).$

$-1 < \tanh(x) < 1$

$\Rightarrow 0 \le \tanh(x) < 1$

$0 < |dh_1/dx| \le 1$

$n = i$ $\quad dh_i/dx = h_{i-1}'(1-\tanh^2(h_{i-1}x + b_{i-1}))$

$0 < |dh_i/dx| \le h_{i-1}'$

By induction

$\Rightarrow 0 < |dy/dx| \le h_1' \Rightarrow 0 < |dy/dx| \le 1$

$0 < \left|\frac{dh_n}{dh_1}\right| = \left|\frac{dh_n}{dx}\cdot\frac{dx}{dh_{n-1}}\right| \le 1$

$\Rightarrow$ The gradient does not vanish or explode for Tanh function.

1.2.1.

$$\sigma_{max}(dx_n/dx_1) = \sigma_{max}(dx_n/dx_{n-1} \cdots dx_2/dx_1)$$

$$\le \sigma_{max}(dx_n/dx_{n-1}) \cdot \cdots \cdot \sigma_{max}(dx_2/dx_1)$$

For any i in $n \ldots, 2$.

$$dx_i/dx_{i-1} = d\tanh(Wx_{i-1})/dx_{i-1} = W(1-\tanh^2(Wx_{i-1}))$$

$$\sigma_{max}(dx_i/dx_{i-1}) = \sigma_{max}(W(1-\tanh^2(Wx_{i-1}))) \le \sigma_{max}(W)\,\sigma_{max}(1-\tanh^2(Wx_{i-1}))$$

$\because$ $\sigma_{max}(W) = 1/2$. $\sigma_{max}(0) < \sigma_{max}(1-\tanh^2(Wx_{i-1})) \le \sigma_{max}(1)$

$$\Rightarrow \quad 0 < \sigma_{max}(1-\tanh^2(Wx_{i-1})) \le 1$$

$\therefore$ $0 < \sigma_{max}(dx_i/dx_{i-1}) \le 1/2$

$$\Rightarrow \quad 0 \le \sigma_{max}(dx_n/dx_1) \le (1/2)^n$$

1.2.3.

We know that (from 1.2.1)

$$\begin{cases} \sigma_{min}(z_{t+1}/z_t) \ge 1 - \sigma_{small} \\ \sigma_{max}(z_{t+1}/z_t) \ge \sigma_{big} - 1 \end{cases} \quad \text{and} \quad \begin{cases} \sigma_{small} \ll 1 \\ \sigma_{big} \gg 2. \end{cases}$$

$\Rightarrow$ Similarly to 1.2.1 $\qquad z_{t+1} = z_t + f_t \quad dz_{t+1}/dz_t = 1 + df_t/dz_t$

$$\sigma_{min}(dz_n/dz_1) = \sigma_{min}(dz_n/dz_{n-1} \cdots dz_2/dz_1) \ge \sigma_{min}(dz_n/dz_{n-1}) \cdots \sigma_{min}(dz_2/dz_1)$$

$$\ge (1-\sigma_{small})^{n-1}$$

Since $\sigma_{small} \ll 1$

with $n < \infty$

$\Rightarrow$ $\sigma_{min}(dz_n/dz_1) \ge 1$, is bounded, close to 1. not explode/vanish.

$$\sigma_{max}(dz_n/dz_1) \le \sigma_{max}(dz_n/dz_{n-1}) \cdots \sigma_{max}(dz_2/dz_1) \ge (\sigma_{big}-1)^{n-1}$$

Since $\sigma_{big} \gg 2$

$\sigma_{big} - 1 \gg 1$

$(\sigma_{big}-1)^{n-1} \to \infty$ $\Rightarrow$ $\sigma_{max}(dz_n/dz_1) \to \infty$, not bounded, explodes.

1.3.2.

The first model on the left is favored

Because for it ith layer and $i+1$ th:

$\bar{x}^i = \bar{x}^{i+1}(1 + dF/dx)$. this model is more stable with a "1" in it,

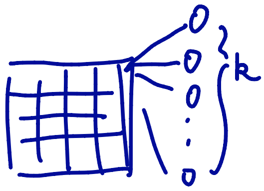
to anti-vanishing to 0.

2.2.1

$O(WHdk^2)$

2.2.2.

$O(d)$



2.3.1 $O(WHdk^2)$

2.3.3.

	Pixel CNN	MDRNN
Computational memory.	weight: $4k^2d \Rightarrow O(k^2d)$	Linear in number of data points and the number of network weights. weight: $(2+2d)dk^2WH \Rightarrow O(WHdk^2)$
Computational complexity. ≈ connections.	$O(WHdk^2)$	$O(WHdk^2)$
Parallelism.	$O(d)$. parallel in each layer	serial in each layer.
Size of context window.	all data unmasked.	adj 2 hidden units.

Conclusion: with similar computational task, pixel CNN is more suitable in paralleism, with less computation memory needed and larger size of context window; while MDRNN may capture more sequential relationships between data.