PADDING AND POOLING

① DNA can be expressed as a sequence of letters from the alphabet ⟨A, C, G, T⟩ ("adenine", "cytosine", "guanine", and "thymine"), e.g.

G A T T A C A A

C A G T T A A G

A A T A A A C C

G C A T T C A G

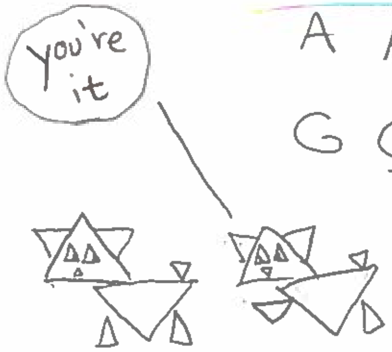
② Suppose that all DNA sequences of the form *CA*TT*AG*, where * is an arbitarily long (possibly empty) sequence, are responsible for some bodily function:

G A T T A C A A

$\underline{\text{C A}}$ G $\underline{\text{T T}}$ A $\underline{\text{A G}}$ ← positive

A A T A A A C C

G $\underline{\text{C A}}$ $\underline{\text{T T}}$ C $\underline{\text{A G}}$ ← positive



PADDING AND POOLING

③ We can imagine building a CNN with three convolution kernels: one to detect "CA", one to detect "TT", and one to detect "AG".

$$X: \quad \begin{array}{c|cccccccc} & C & A & G & T & T & A & A & G \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ A & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ C & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ T & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}$$

$$\vee\ \vee\ \vee\ \vee\ \vee\ \vee\ \vee$$

$$\dot{X}: \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \leftarrow \text{"CA" detector} \\ \leftarrow \text{"TT" detector} \\ \leftarrow \text{"AG" detector} \end{array}$$

④ The first thing to notice is that $\dot{X}$ has one fewer column than X. This disparity will end up being rather annoying.

Padding and Pooling

⑤ We can remedy this by simply "padding" the end of the DNA sequence with a fifth letter, which we'll call "0".

$$\begin{array}{cccccccccc}
 & C & A & G & T & T & A & A & G & (0) \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
X: & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 & \vee & \vee & \vee & \vee & \vee & \vee & \vee & \vee &
\end{array}$$

$$\dot{X}: \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{array}{l} \leftarrow \text{"CA"} \\ \leftarrow \text{"TT"} \\ \leftarrow \text{"AG"} \end{array}$$

⑥ If we run this convolution layer on other positive sequences:

$$\begin{array}{cccccccc}
G & C & A & T & T & C & A & G \\
\vee & \vee & \vee & \vee & \vee & \vee & \vee & \vee
\end{array}
\qquad
\begin{array}{cccccccc}
C & A & T & T & C & A & G & C \\
\vee & \vee & \vee & \vee & \vee & \vee & \vee & \vee
\end{array}
\qquad
\begin{array}{cccccccc}
T & T & C & A & T & T & A & G \\
\vee & \vee & \vee & \vee & \vee & \vee & \vee & \vee
\end{array}$$

$$\dot{X}: \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

PADDING AND POOLING

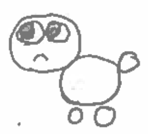
7. Observe that the next layer of the CNN is still faced with a relatively daunting task. If

$$\dot{X} = \begin{bmatrix} \dot{x}_{11} & \cdots & \dot{x}_{18} \\ \dot{x}_{21} & \cdots & \dot{x}_{28} \\ \dot{x}_{31} & \cdots & \dot{x}_{38} \end{bmatrix},$$

then we want to predict a positive response in each of the following cases:

- $\dot{x}_{11} = 1$ and $\dot{x}_{23} = 1$ and $\dot{x}_{35} = 1$
- $\dot{x}_{11} = 1$ and $\dot{x}_{23} = 1$ and $\dot{x}_{36} = 1$
- $\dot{x}_{\underline{1}1} = 1$ and $\dot{x}_{\underline{2}3} = 1$ and $\dot{x}_{3\underline{7}} = 1$
- $\dot{x}_{11} = 1$ and $\dot{x}_{24} = 1$ and $\dot{x}_{36} = 1$
- $\dot{x}_{11} = 1$ and $\dot{x}_{24} = 1$ and $\dot{x}_{37} = 1$
- $\dot{x}_{11} = 1$ and $\dot{x}_{25} = 1$ and $\dot{x}_{36} = 1$
- $\dot{x}_{12} = 1$ and $\dot{x}_{24} = 1$ and $\dot{x}_{36} = 1$
- $\dot{x}_{12} = 1$ and $\dot{x}_{24} = 1$ and $\dot{x}_{37} = 1$
- $\dot{x}_{12} = 1$ and $\dot{x}_{25} = 1$ and $\dot{x}_{37} = 1$
- $\dot{x}_{13} = 1$ and $\dot{x}_{25} = 1$ and $\dot{x}_{37} = 1$

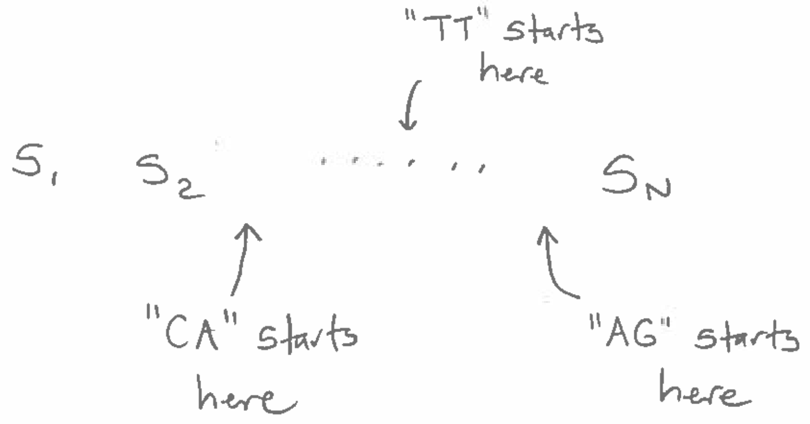
e.g., read this line as: "CA" starts at position 1, "TT" starts at position 3, and "AG" starts at position 7



There's no way for it to generalize between these cases, so the training data would need to contain several instances of each case.

PADDING AND POOLING

(8) This only gets worse as the length of the DNA sequence grows. If its length is N, then the number of different cases to consider is cubic in N:



(9) One way to tackle this issue is by creating a layer that summarizes the results of the detectors over subregions:

C A G T T A A G

$$\dot{X}: \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\ddot{X}: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"AG" starts at position 7 or 8

C A T T C A G C

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

e.g. "CA" starts at position 5 or position 6

PADDING AND POOLING

(10) This technique:

$$\dot{X}: \begin{bmatrix} \dot{x}_{11} & \dot{x}_{12} & \dot{x}_{13} & \dot{x}_{14} & \dot{x}_{15} & \dot{x}_{16} & \dot{x}_{17} & \dot{x}_{18} \\ \dot{x}_{21} & \dot{x}_{22} & \dot{x}_{23} & \dot{x}_{24} & \dot{x}_{25} & \dot{x}_{26} & \dot{x}_{27} & \dot{x}_{28} \\ \dot{x}_{31} & \dot{x}_{32} & \dot{x}_{33} & \dot{x}_{34} & \dot{x}_{35} & \dot{x}_{36} & \dot{x}_{37} & \dot{x}_{38} \end{bmatrix}$$

$$\ddot{X}: \begin{bmatrix} \max(\dot{x}_{11},\dot{x}_{12}) & \max(\dot{x}_{13},\dot{x}_{14}) & \max(\dot{x}_{15},\dot{x}_{16}) & \max(\dot{x}_{17},\dot{x}_{18}) \\ \max(\dot{x}_{21},\dot{x}_{22}) & \max(\dot{x}_{23},\dot{x}_{24}) & \max(\dot{x}_{25},\dot{x}_{26}) & \max(\dot{x}_{27},\dot{x}_{28}) \\ \max(\dot{x}_{31},\dot{x}_{32}) & \max(\dot{x}_{33},\dot{x}_{34}) & \max(\dot{x}_{35},\dot{x}_{36}) & \max(\dot{x}_{17},\dot{x}_{38}) \end{bmatrix}$$

is called <u>max-pooling</u>.

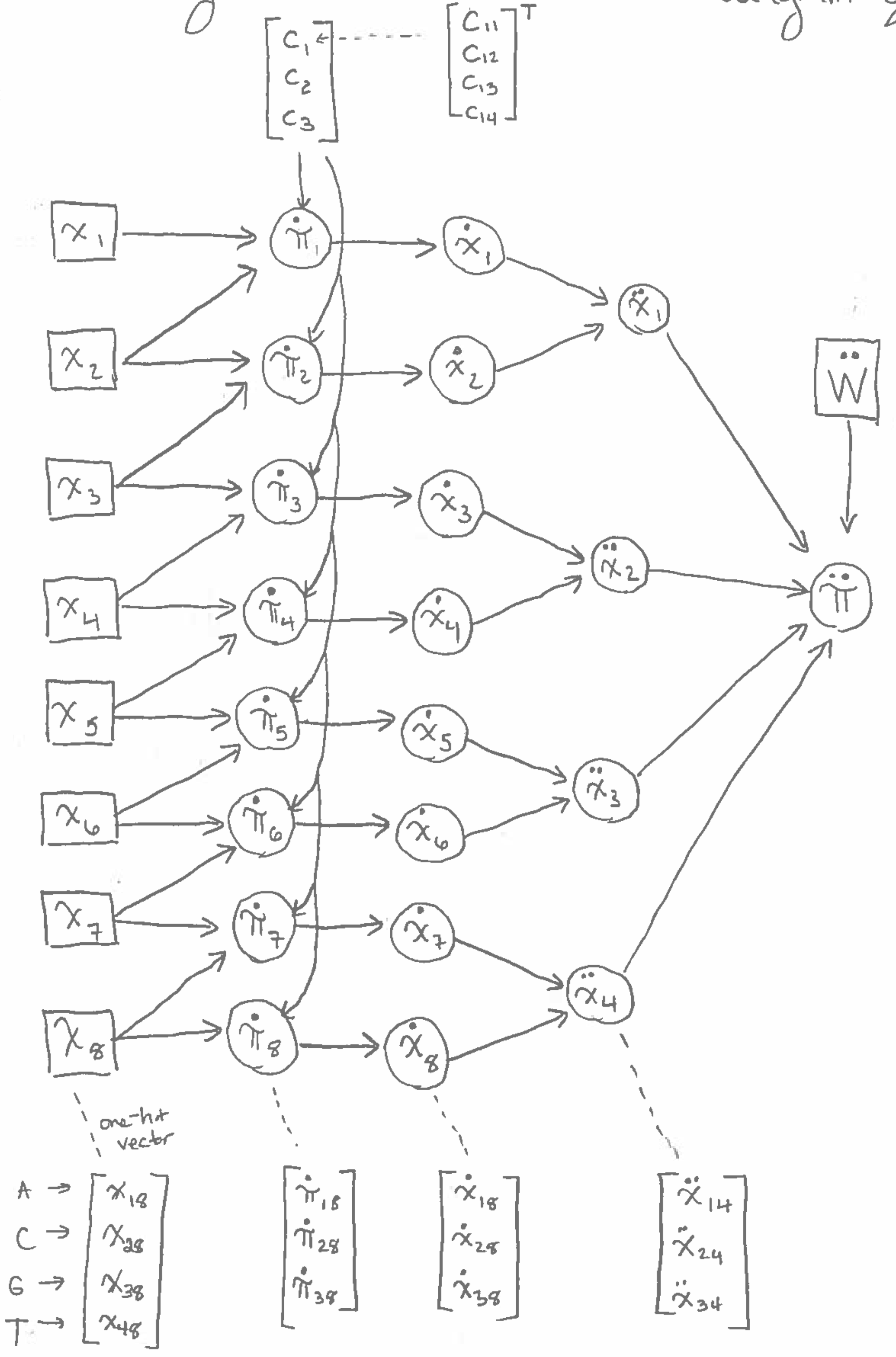
(11) Notice that max-pooling greatly simplifies the next layer of the CNN. If $\ddot{X} = \begin{bmatrix} \ddot{x}_{11} & \ddot{x}_{12} & \ddot{x}_{13} & \ddot{x}_{14} \\ \ddot{x}_{21} & \ddot{x}_{22} & \ddot{x}_{23} & \ddot{x}_{24} \\ \ddot{x}_{31} & \ddot{x}_{32} & \ddot{x}_{33} & \ddot{x}_{34} \end{bmatrix}$, then

we want to predict a positive response in these cases:

- $\ddot{x}_{11} = 1$ and $\ddot{x}_{22} = 1$ and $\ddot{x}_{33} = 1$
- $\ddot{x}_{11} = 1$ and $\ddot{x}_{22} = 1$ and $\ddot{x}_{34} = 1$
- $\ddot{x}_{11} = 1$ and $\ddot{x}_{23} = 1$ and $\ddot{x}_{34} = 1$
- $\ddot{x}_{12} = 1$ and $\ddot{x}_{23} = 1$ and $\ddot{x}_{34} = 1$

Padding and Pooling

(12) Summarizing all this as a causal diagram gives us:



c_1, c_2, c_3
c_{11}, c_{12}, c_{13}, c_{14}, T
x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8
$\dot{\pi}_1$, $\dot{\pi}_2$, $\dot{\pi}_3$, $\dot{\pi}_4$, $\dot{\pi}_5$, $\dot{\pi}_6$, $\dot{\pi}_7$, $\dot{\pi}_8$
$\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$, $\dot{x}_4$, $\dot{x}_5$, $\dot{x}_6$, $\dot{x}_7$, $\dot{x}_8$
$\ddot{x}_1$, $\ddot{x}_2$, $\ddot{x}_3$, $\ddot{x}_4$
$\ddot{W}$
$\ddot{\pi}$
one-hot vector
A →, C →, G →, T →
x_{18}, x_{28}, x_{38}, x_{48}
$\dot{\pi}_{18}$, $\dot{\pi}_{28}$, $\dot{\pi}_{38}$
$\dot{x}_{18}$, $\dot{x}_{28}$, $\dot{x}_{38}$
$\ddot{x}_{14}$, $\ddot{x}_{24}$, $\ddot{x}_{34}$

PADDING AND POOLING

(13) How does one backpropagate through a maxpool layer? Consider the following simple example:

$$\begin{tikzcd} & x_1 \arrow[dr] & \\ w \arrow[ur] \arrow[dr] & & \dot{x} \\ & x_2 \arrow[ur] & \end{tikzcd}$$

where $\dot{x} = \max(x_1, x_2)$, and suppose we need to compute $\frac{\partial \dot{x}}{\partial w}$. Assume x_1, x_2 are scalars.

(14) From the Chain Rule:

$$\frac{\partial \dot{x}}{\partial w} = \frac{\partial \dot{x}}{\partial x_1}\frac{\partial x_1}{\partial w} + \frac{\partial \dot{x}}{\partial x_2}\frac{\partial x_2}{\partial w}$$

We know:

$$\dot{x} = \begin{cases} x_1 & \text{if } x_1 \geq x_2 \\ x_2 & \text{if } x_2 \geq x_1 \end{cases}$$

So:

$$\frac{\partial \dot{x}}{\partial x_1} = \begin{cases} 1 & \text{if } x_1 > x_2 \\ 0 & \text{if } x_2 < x_1 \\ \text{undefined} & \text{if } x_1 = x_2 \end{cases} \qquad \frac{\partial \dot{x}}{\partial x_2} = \begin{cases} 1 & \text{if } x_2 > x_1 \\ 0 & \text{if } x_1 > x_2 \\ \text{undefined} & \text{if } x_1 = x_2 \end{cases}$$

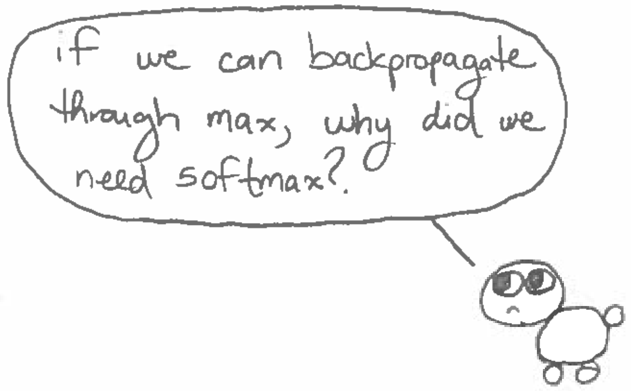
The functions are piecewise differentiable, much like ReLU:

$$\frac{d}{dz} a(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z < 0 \\ \text{undefined} & \text{if } z = 0. \end{cases}$$

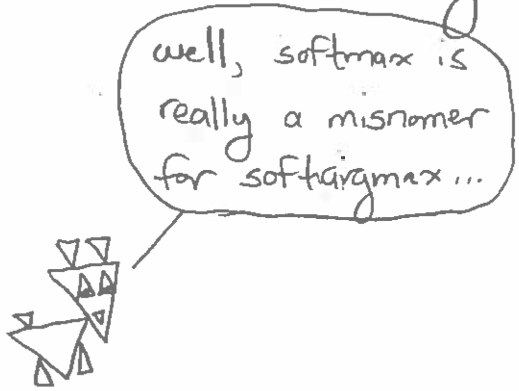
PADDING AND POOLING

(15) As with ReLU, we just need to hope that we don't reach a place in our weight space (during gradient descent) where $x_1 = x_2$. Since they're both real numbers, chances are low.

(16) It could be asked:


if we can backpropagate through max, why did we need softmax?

The answer lies in the fact that softmax is really a misnomer for softargmax.



PADDING AND POOLING

(17) Recall that softmax takes a vector of reals and produces a skewed distribution of the same dimension:

e.g.

$$\begin{bmatrix} 1.6 \\ -0.2 \end{bmatrix} \longrightarrow \begin{bmatrix} .86 \\ .14 \end{bmatrix}$$

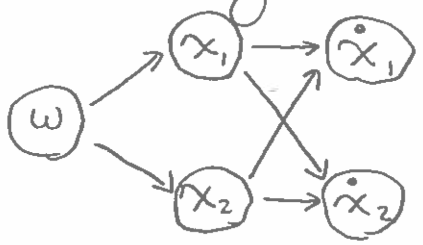
This is an approximation of an argmax function that produces a one-hot vector:

$$\begin{bmatrix} 1.6 \\ -0.2 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

It is not an approximation of the max function:

$$\begin{bmatrix} 1.6 \\ -0.2 \end{bmatrix} \longrightarrow 1.6$$

(18) Let's try implementing this argmax function:



where: $\dot{x}_1 = \begin{cases} 1 & \text{if } x_1 \geq x_2 \\ 0 & \text{if } x_2 > x_1 \end{cases}$ $\quad \dot{x}_2 = \begin{cases} 1 & \text{if } x_2 \geq x_1 \\ 0 & \text{if } x_1 > x_2 \end{cases}$

PADDING AND POOLING

(19) That means:

$$\frac{\partial \dot{x}_1}{\partial x_1} = 0 \qquad \frac{\partial \dot{x}_1}{\partial x_2} = 0$$

$$\frac{\partial \dot{x}_2}{\partial x_1} = 0 \qquad \frac{\partial \dot{x}_2}{\partial x_2} = 0$$

All the derivatives are degenerate and useless.