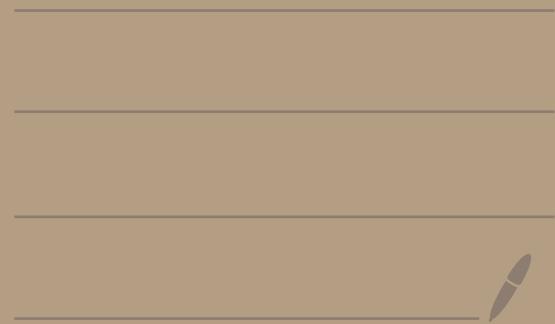
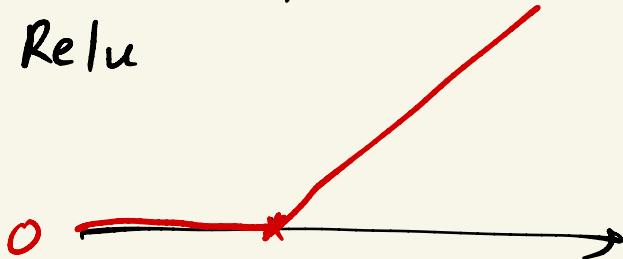


Lecture 19 Neural Arch



Fourier transform

Relu



$$\langle (1, 1, 1), (-1, 0, 1) \rangle = 0$$

$$(1, 1, 1)$$

$$(1, 1, 2)$$

n, m



r

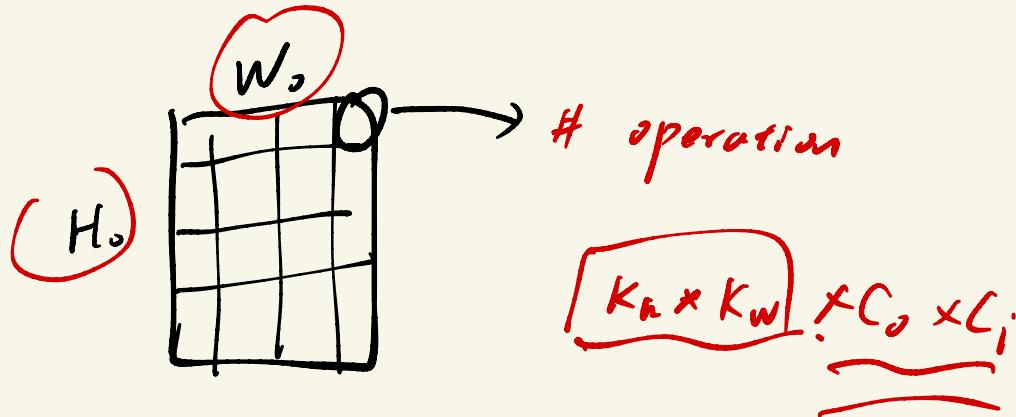
kernel filter $\sum C_o \times C_i$



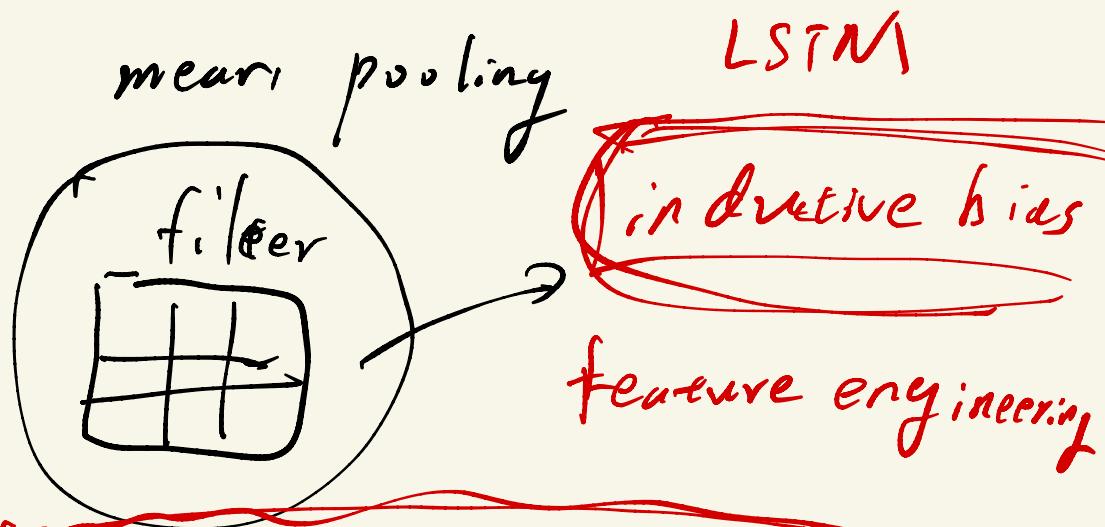
$w \times fb$

A diagram illustrating a convolutional layer. It shows an input tensor with dimensions $3 \times 64 \times 64$ being processed by a kernel filter with dimensions $C \times H \times W$. The output dimensions are indicated as $3 \times 64 \times 64$.

(R, G, B)

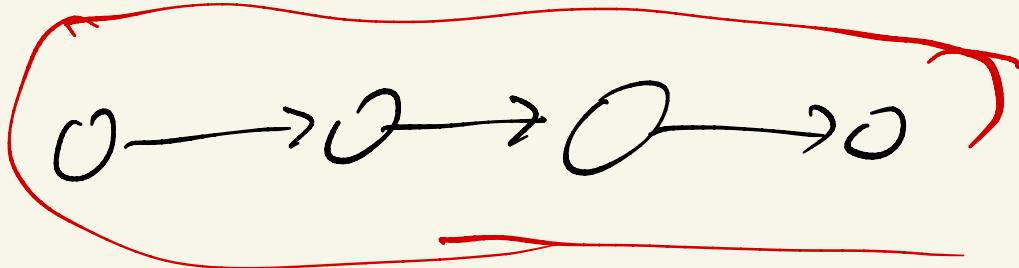
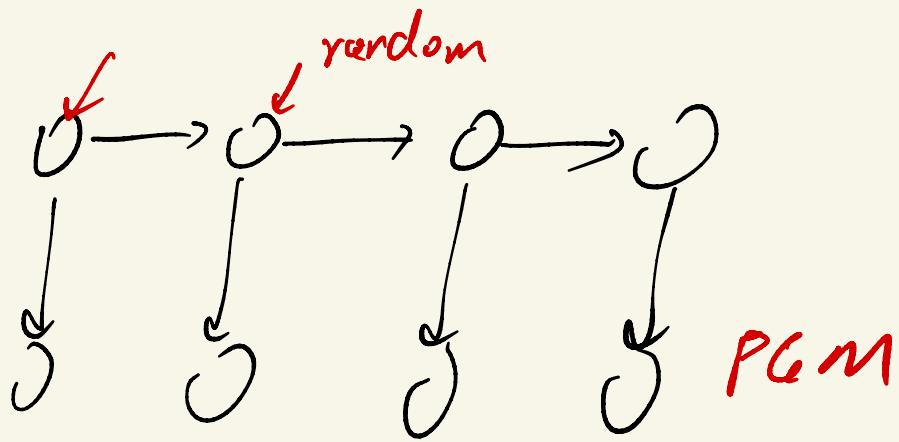


sliding window
convolution is linear

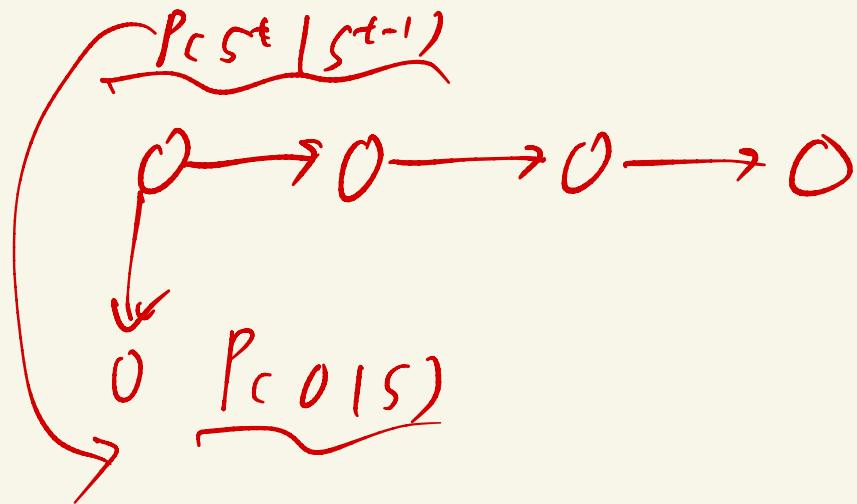


IBM translation

Every time I fire a language
my machine translates - - -
data-driven



hidden states



$U \rightarrow U \rightarrow O$

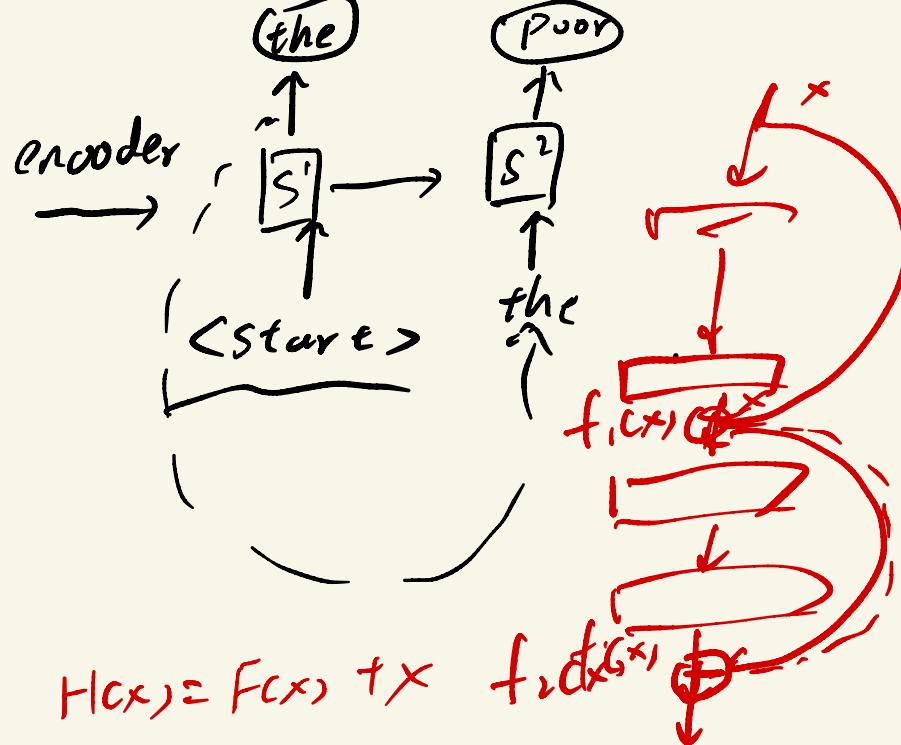
$f(s^e | \theta)$

$$a = W s^{(e-1)} + b + U x^{(e)}$$

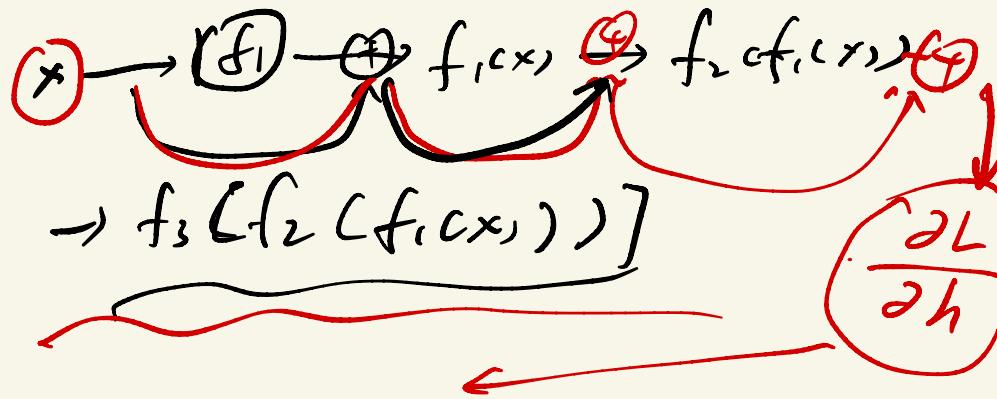
$$s^{(e)} = \tanh(a)$$

$$o^{(e)} = V s^{(e)} + c$$

y



$$H(x) = F(x) + x \quad f_2(x)$$



Layer 1: $x + f_1(cx)$

Layer 2: $x + f_2(cx + f_1(cx)) + f_1(cx)$

Layer 3: $x + f_3(cx + f_2(cx + f_1(cx))) + f_2(cx + f_1(cx)) + f_1(cx)$

(Red lines indicate the flow of gradients from the output layer back through the hidden layers.)

$\frac{\partial L}{\partial x} + \frac{\partial f_3}{\partial -} - - -$