

CNN不具有尺度，旋转，平移不变性，但是通过加上spatial transformer layer可以使之具有一定程度的上述性质。

spatial transformer其实就是对原始图像进行了一个仿射变换。输入的是原始图像，输出的是仿射变换的6个参数（也可以是4个）。

ST不只是可以用在卷积层之前，对原始图像进行处理，还可以用在每个filter之后，对feature map进行处理。

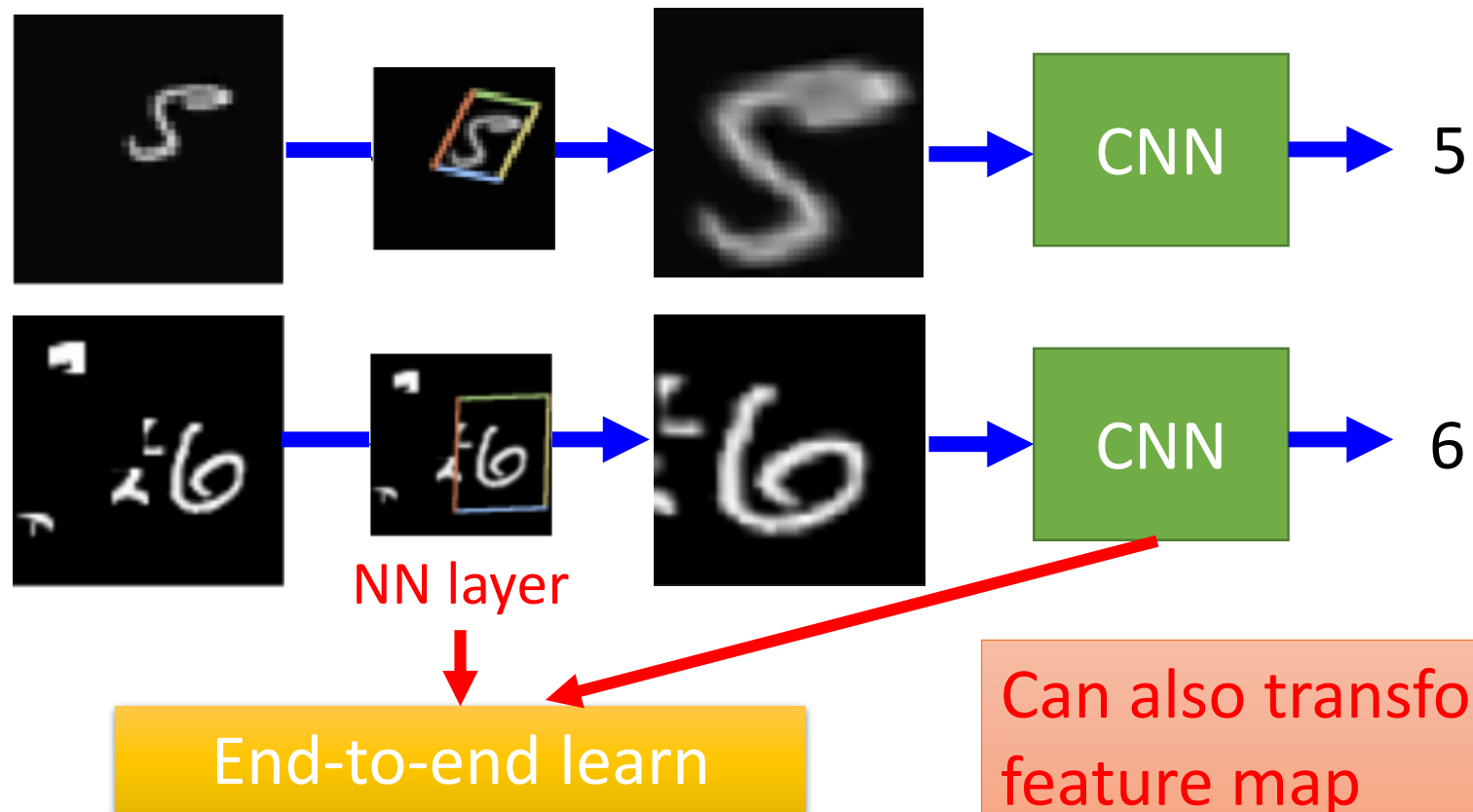
Spatial Transformer

Ref: Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu, "Spatial Transformer Networks", NIPS, 2015

ST不只是可以用在卷积层之前，对原始图像进行处理，还可以用在每个filter之后，对feature map进行处理。

Spatial Transformer Layer

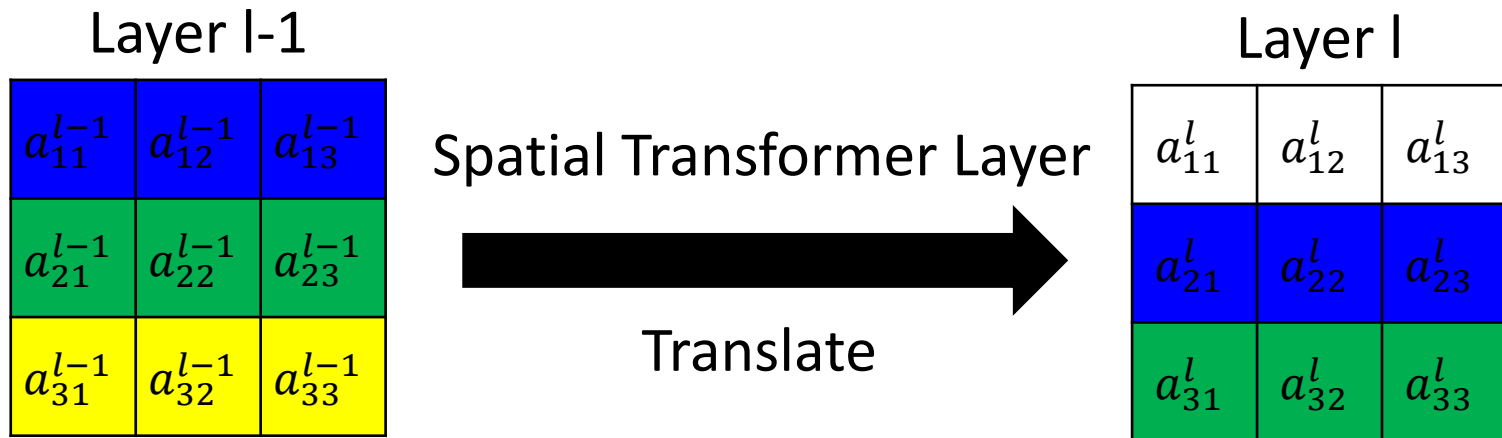
- CNN is not invariant to scaling and rotation



端到端指的就是输入的是原始数据，输出的是结果。而非端到端指的是原始数据要通过人工提取特征，再喂给网络，得到结果。

Spatial Transformer Layer

- How to transform an image/feature map



General layer:
$$a_{nm}^l = \sum_{i=1}^3 \sum_{j=1}^3 w_{nm,ij}^l a_{ij}^{l-1}$$

If we want translate as above:
$$a_{nm}^l = a_{(n-1)m}^{l-1}$$

$$w_{nm,ij}^l = 1 \quad \text{if } i = n - 1, j = m \quad w_{nm,ij}^l = 0 \quad \text{otherwise}$$

Spatial Transformer Layer

- How to transform an image/feature map

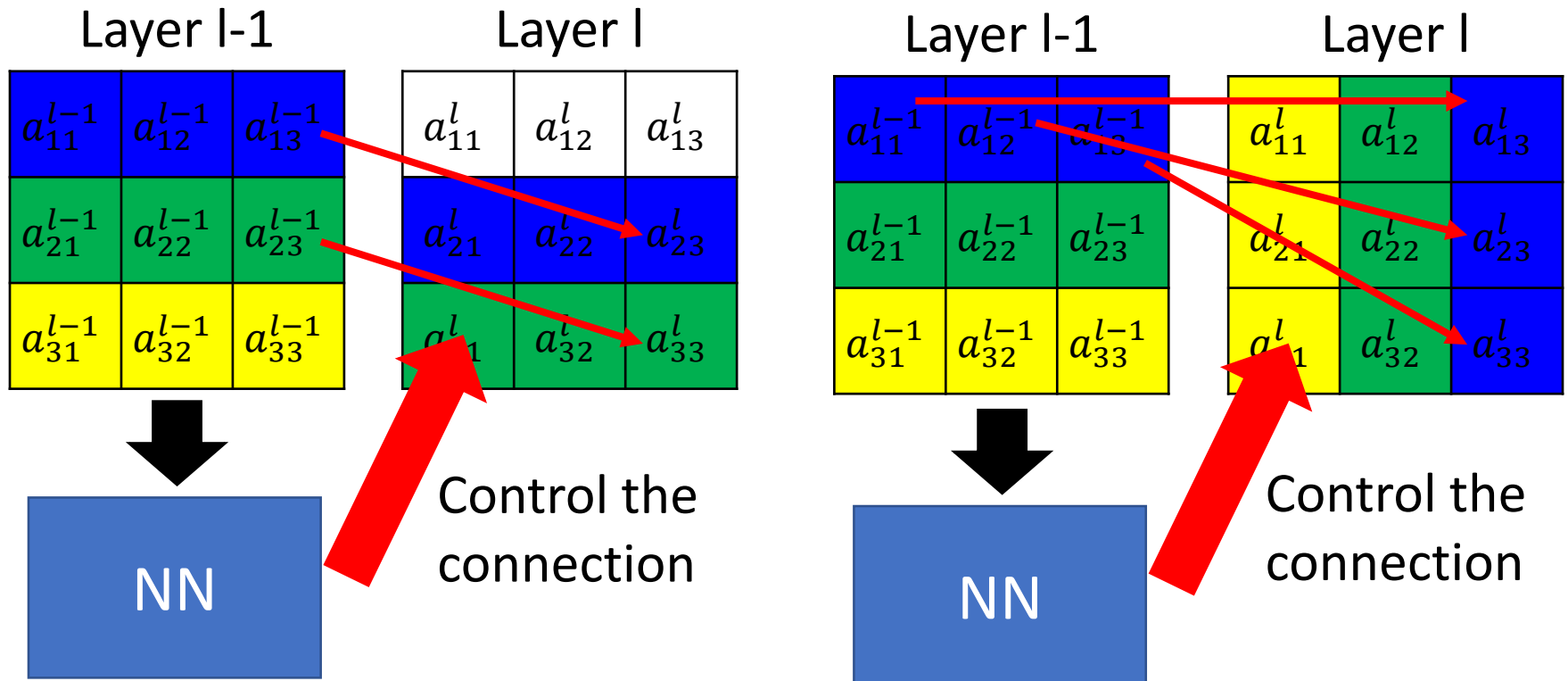


Image Transformation

Expansion, Compression, Translation

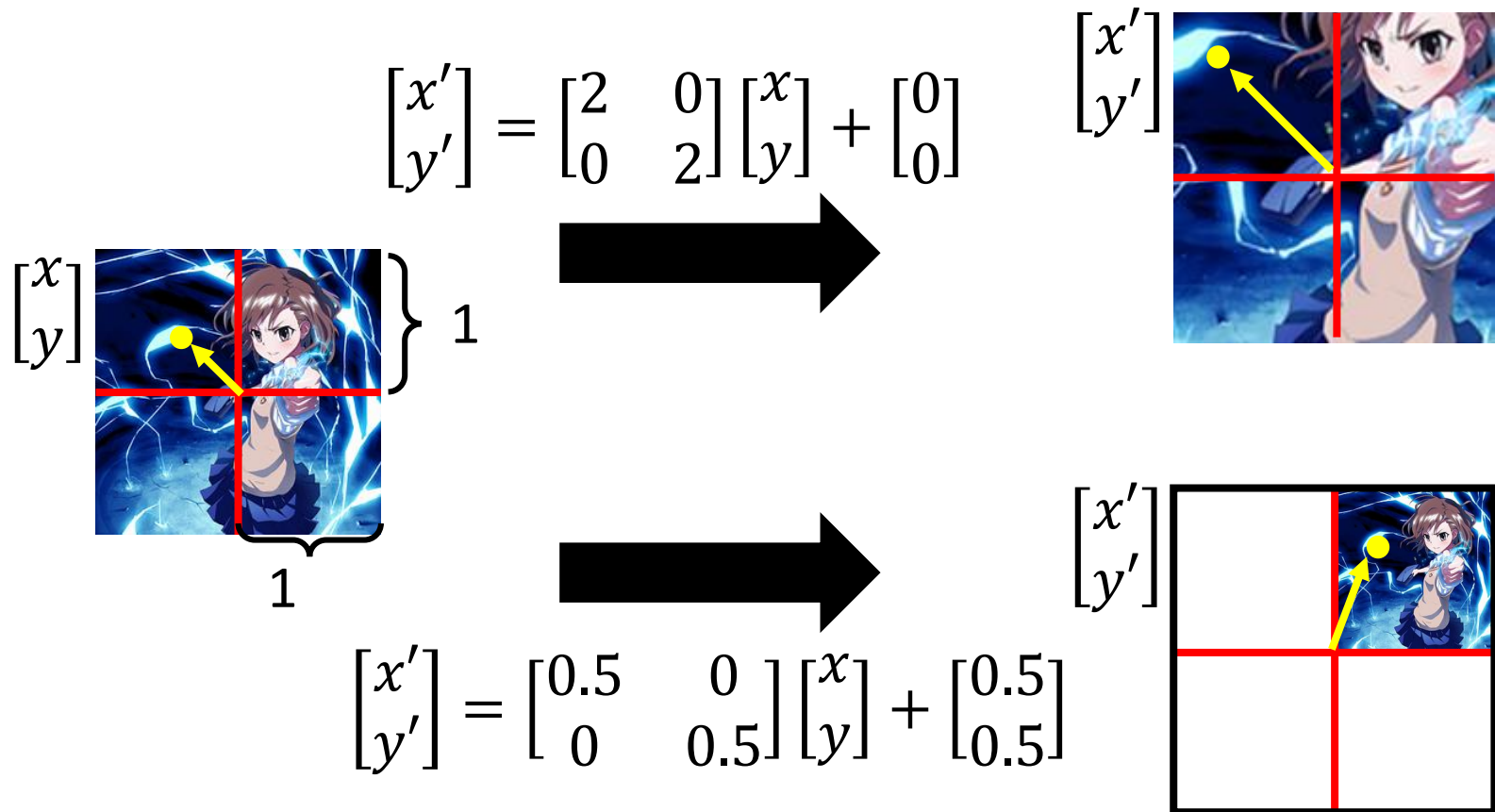
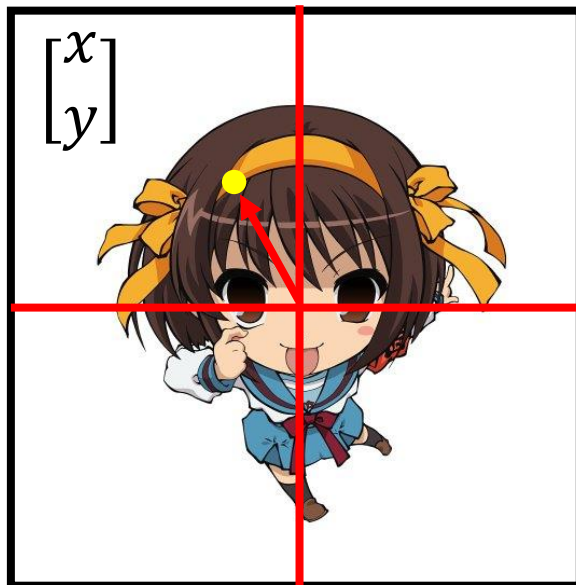


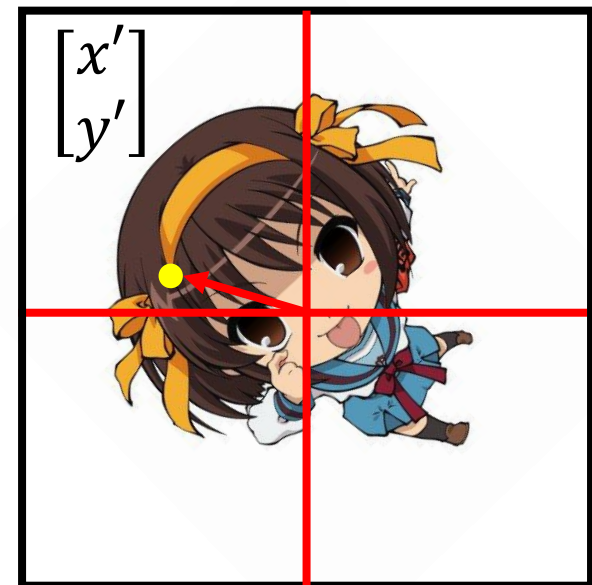
Image Transformation

- Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Rotate
 θ°

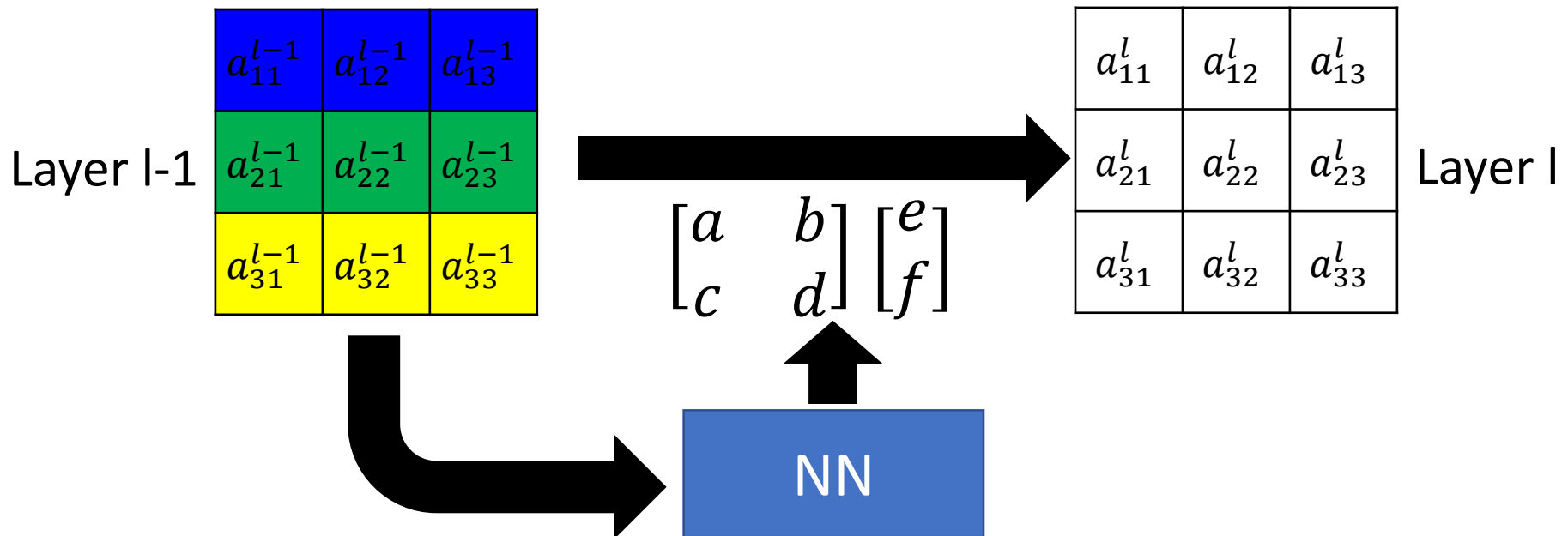


Spatial Transformer Layer

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

6 parameters to describe the affine transformation

Index of layer $l-1$ Index of layer l

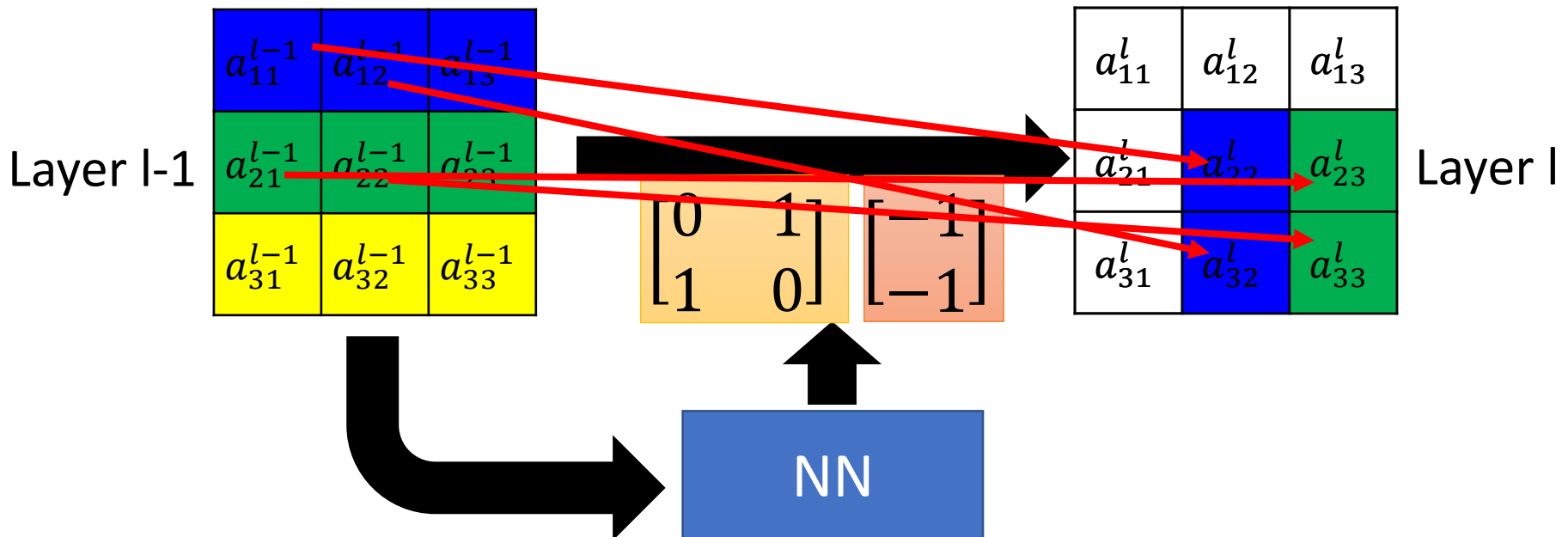


Spatial Transformer Layer

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

6 parameters to describe the affine transformation

Index of layer l-1 Index of layer l



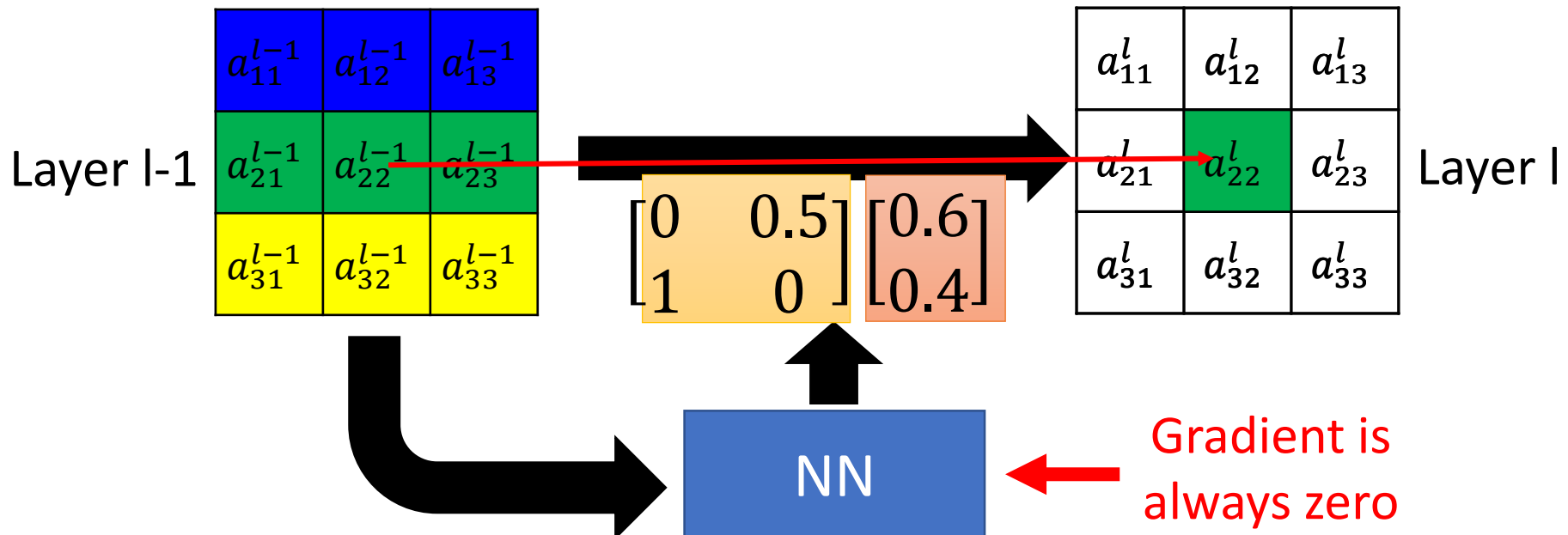
Spatial Transformer Layer

$$\begin{bmatrix} 1.6 \\ 2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0.5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$$

6 parameters to describe the affine transformation

Index of layer l-1 Index of layer l

What is the problem?



直接对小数结果进行四舍五入，那么如果对输入做一点变化，那么结果大概率保持不变，那么就无法进行GD。
所以采用插值法。

Interpolation

Now we can use gradient descent

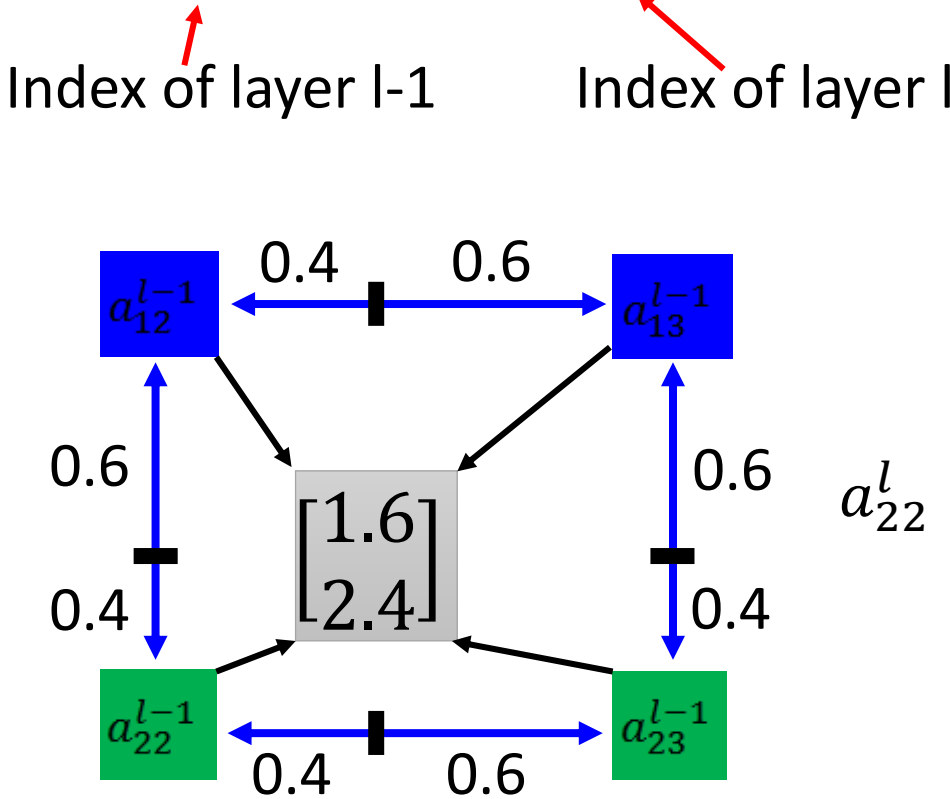
$$\begin{bmatrix} 1.6 \\ 2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0.5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$$

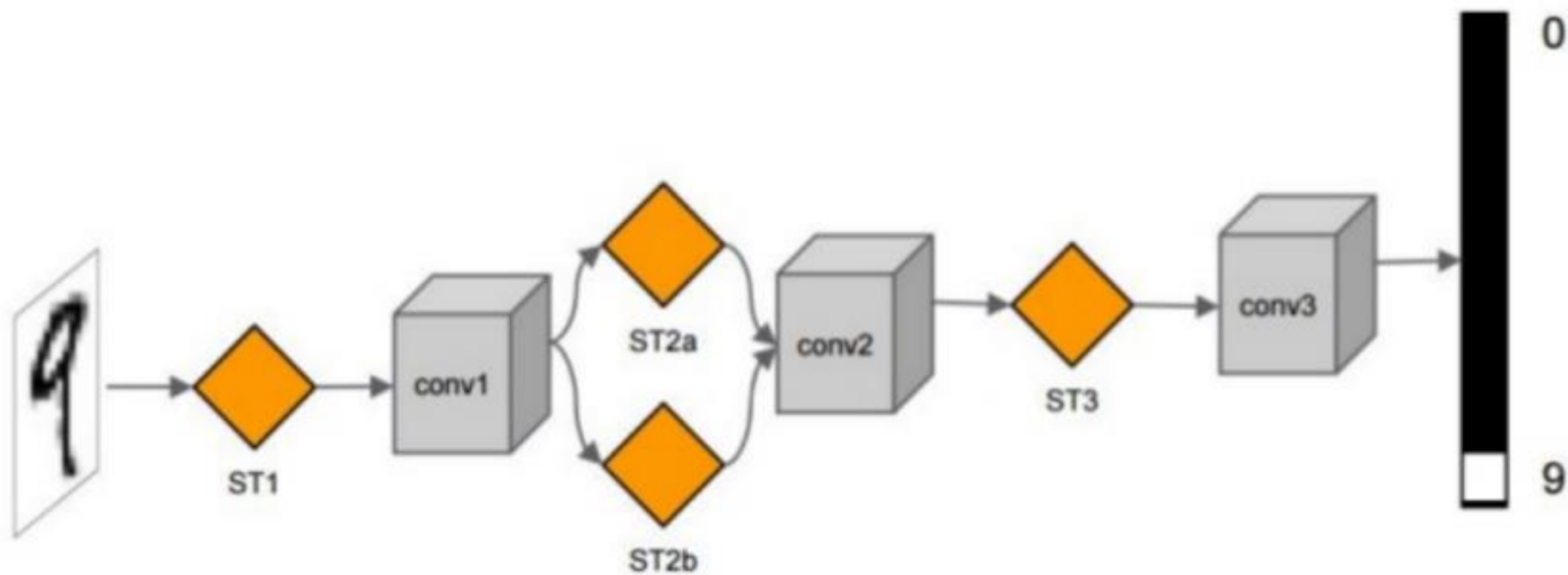
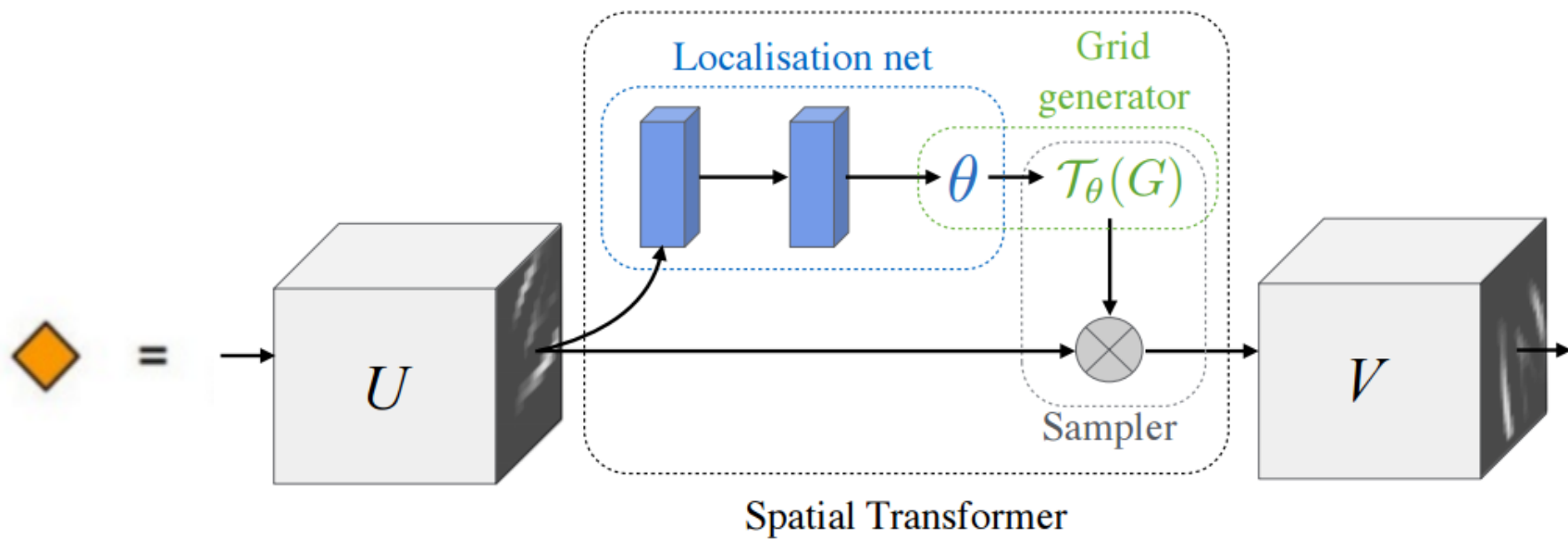
6 parameters to describe the affine transformation

a_{11}^l	a_{12}^l	a_{13}^l
a_{21}^l	a_{22}^l	a_{23}^l
a_{31}^l	a_{32}^l	a_{33}^l

Layer l

$$\begin{aligned} a_{22}^l &= (1 - 0.4) \times (1 - 0.4) \times a_{22}^{l-1} \\ &+ (1 - 0.6) \times (1 - 0.4) \times a_{12}^{l-1} \\ &+ (1 - 0.6) \times (1 - 0.6) \times a_{13}^{l-1} \\ &+ (1 - 0.4) \times (1 - 0.6) \times a_{23}^{l-1} \end{aligned}$$





1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

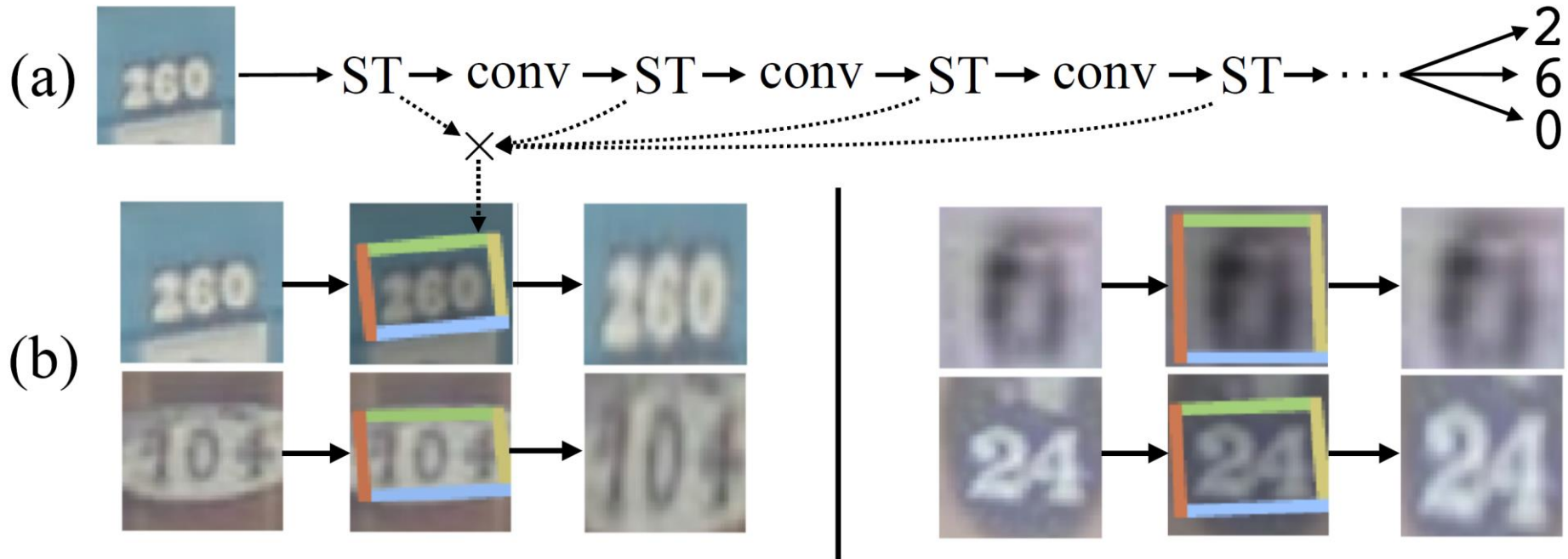
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Street View House Number

Model		Size	
		64px	128px
Maxout CNN [10]		4.0	-
CNN (ours)		4.0	5.6
DRAM* [1]		3.9	4.5
ST-CNN	Single	3.7	3.9
	Multi	3.6	3.9

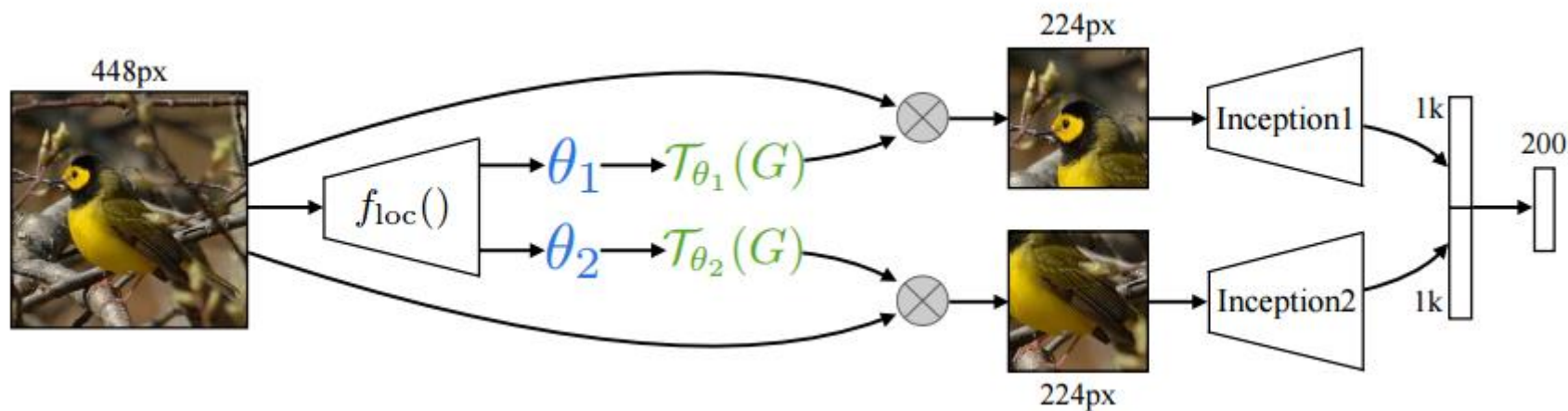
Single: one transformation layer

Multi: many transformation layer

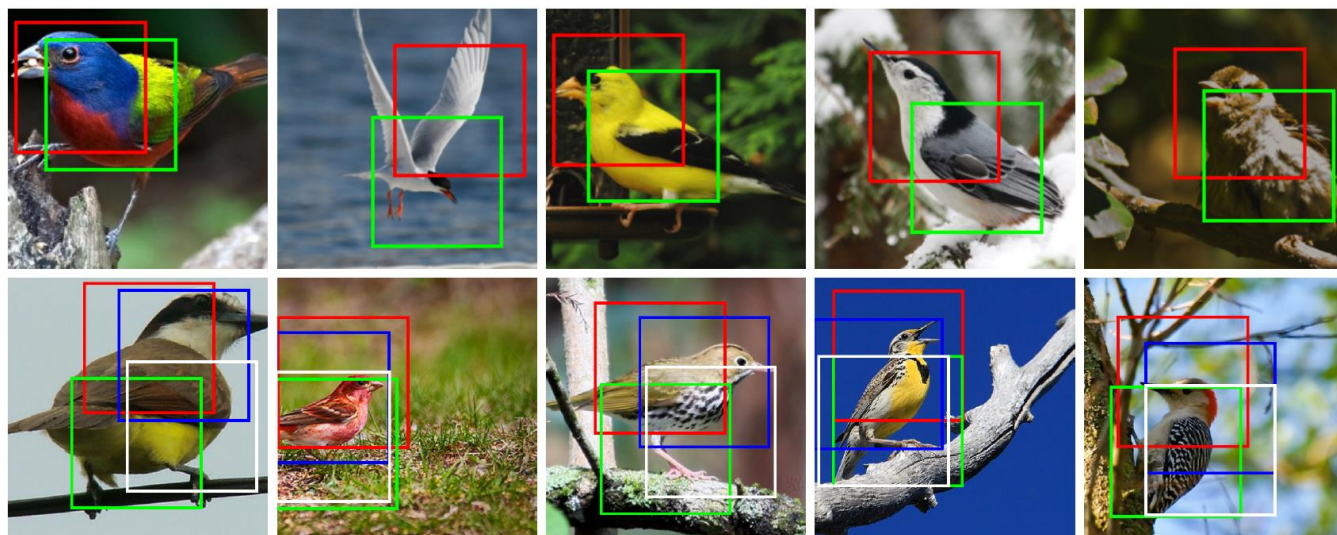


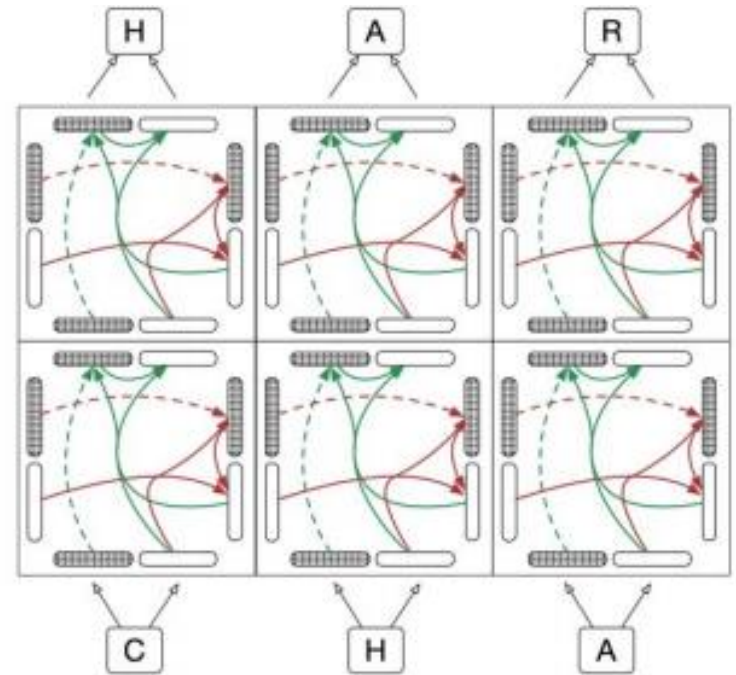
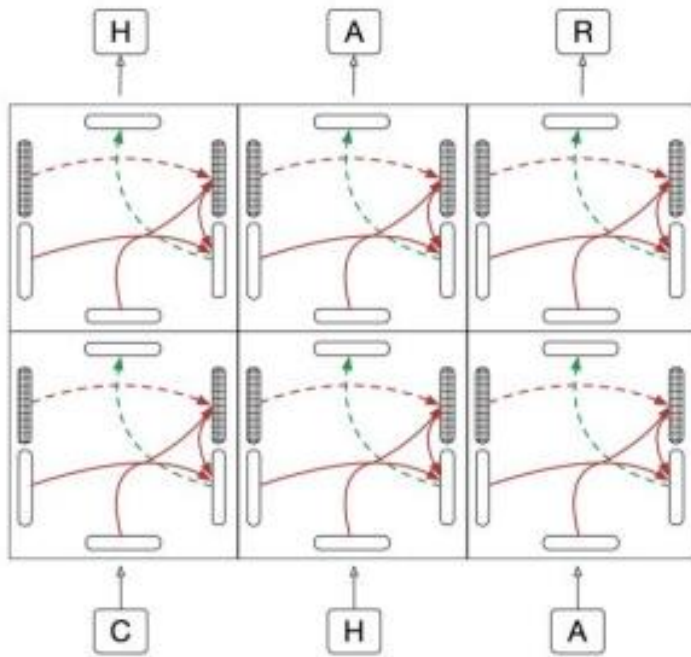
Bird Recognition

$$\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}$$



Model		
Cimpoi '15 [4]		66.7
Zhang '14 [30]		74.9
Branson '14 [2]		75.7
Lin '15 [20]		80.9
Simon '15 [24]		81.0
CNN (ours) 224px		82.3
2×ST-CNN 224px		83.1
2×ST-CNN 448px		83.9
4×ST-CNN 448px		84.1

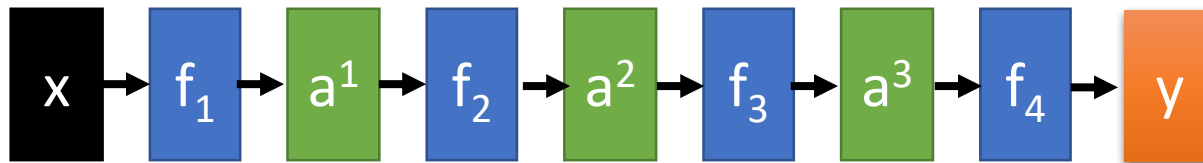




Highway Network & Grid LSTM

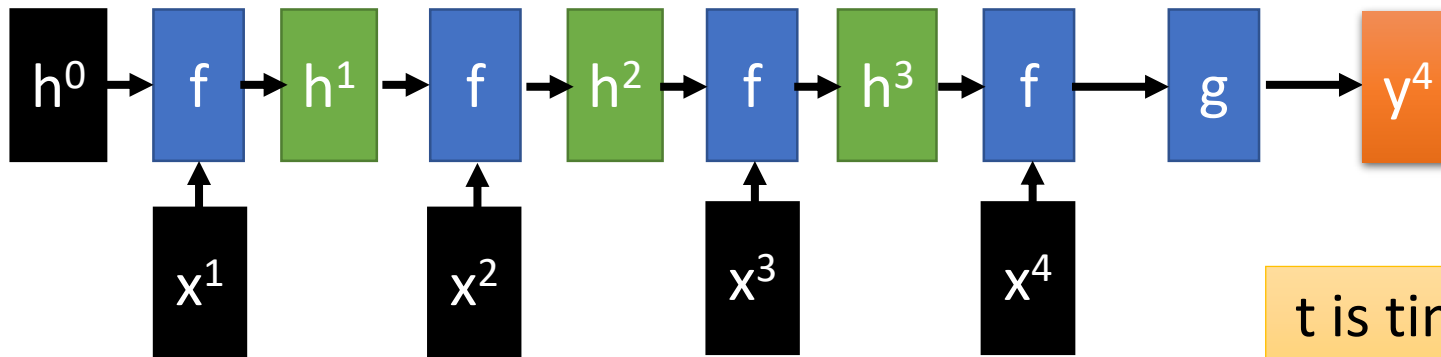
Feedforward v.s. Recurrent

1. Feedforward network does not have input at each step
2. Feedforward network has different parameters for each layer



$$a^t = f_l(a^{t-1}) = \sigma(W^t a^{t-1} + b^t)$$

t is layer



$$a^t = f(a^{t-1}, x^t) = \sigma(W^h a^{t-1} + W^i x^t + b^i)$$

t is time step

Applying gated structure in feedforward network

GRU \rightarrow Highway Network

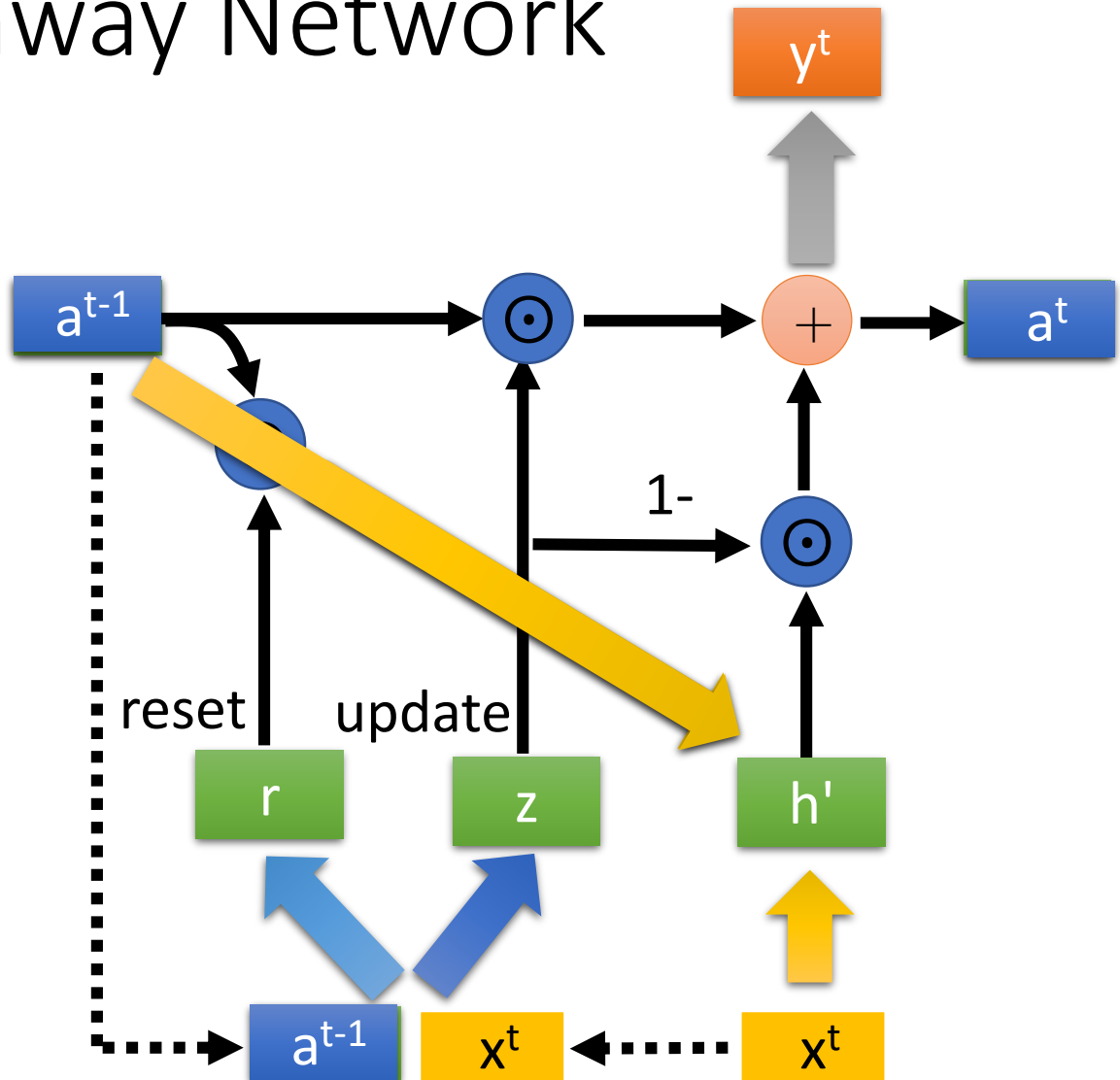
No input x^t at each step

No output y^t at each step

a^{t-1} is the output of the $(t-1)$ -th layer

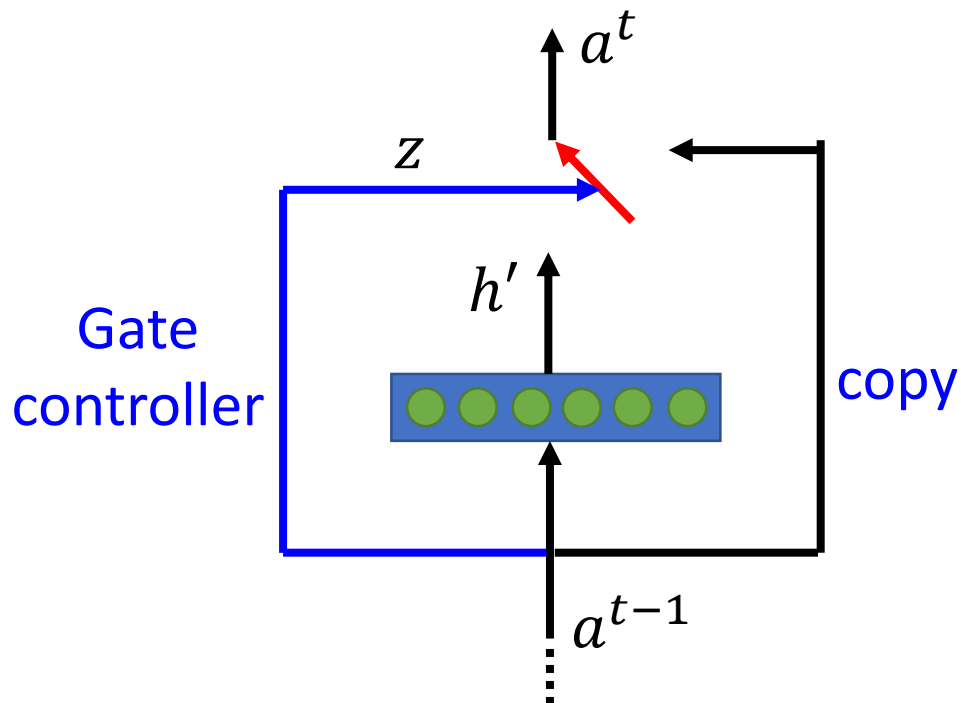
a^t is the output of the t -th layer

No reset gate



Highway Network

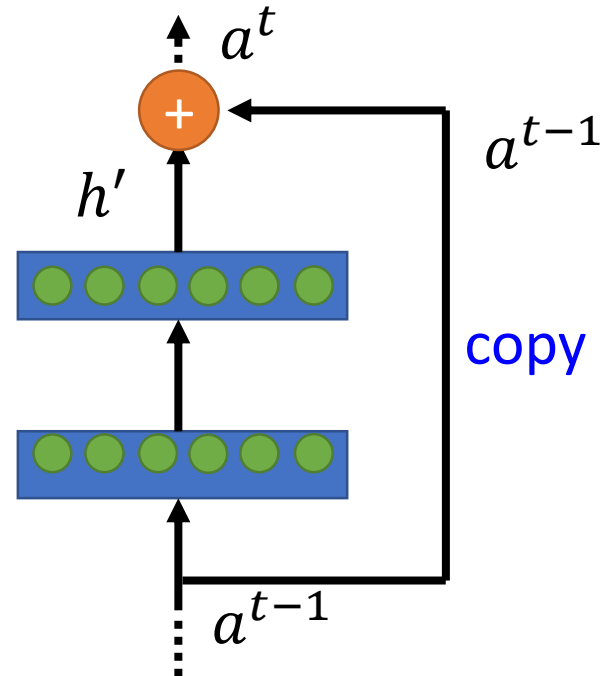
- **Highway Network**



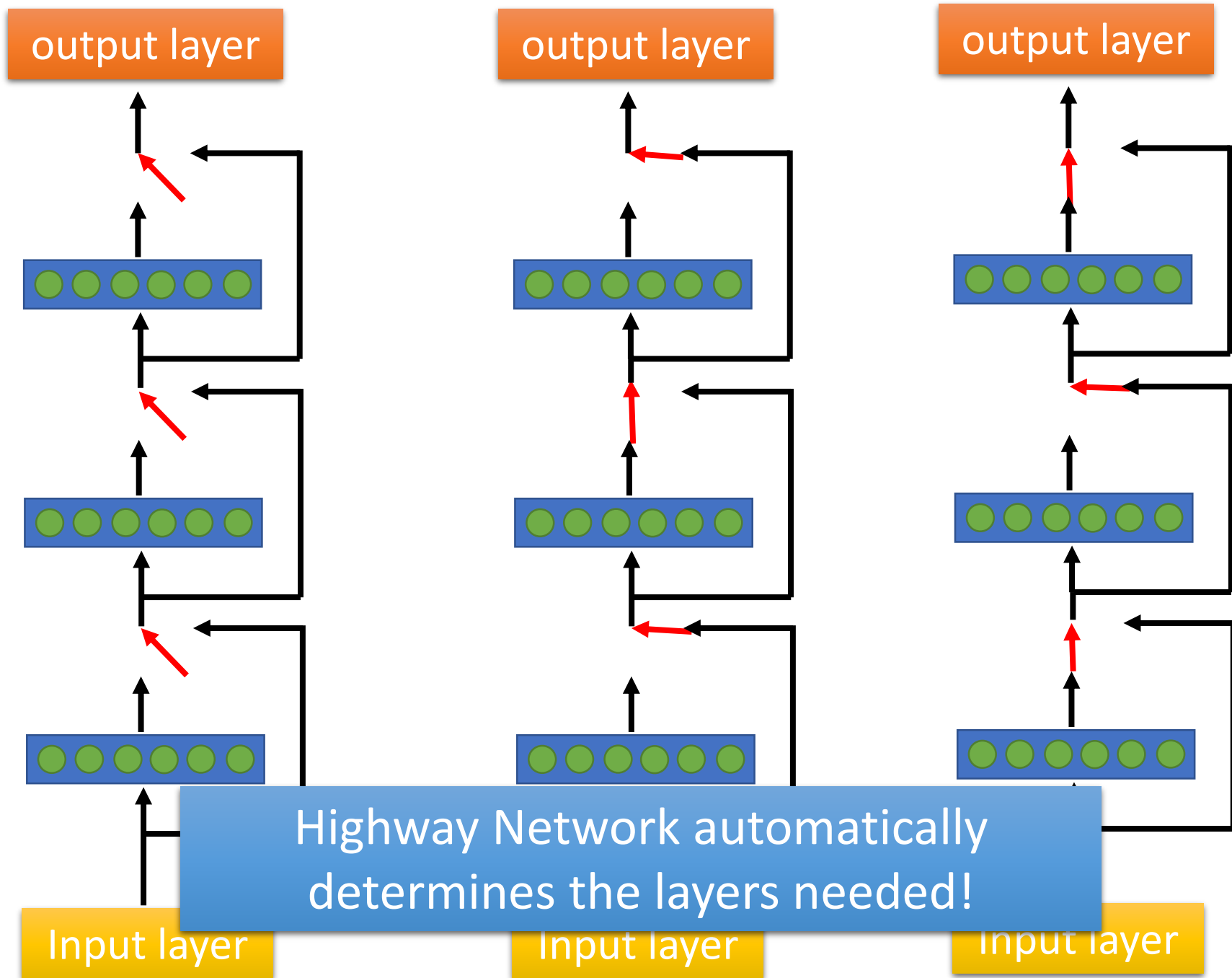
Training Very Deep Networks
<https://arxiv.org/pdf/1507.06228v2.pdf>

$$h' = \sigma(Wa^{t-1})$$
$$z = \sigma(W'a^{t-1})$$
$$a^t = z \odot a^{t-1} + (1 - z) \odot h$$

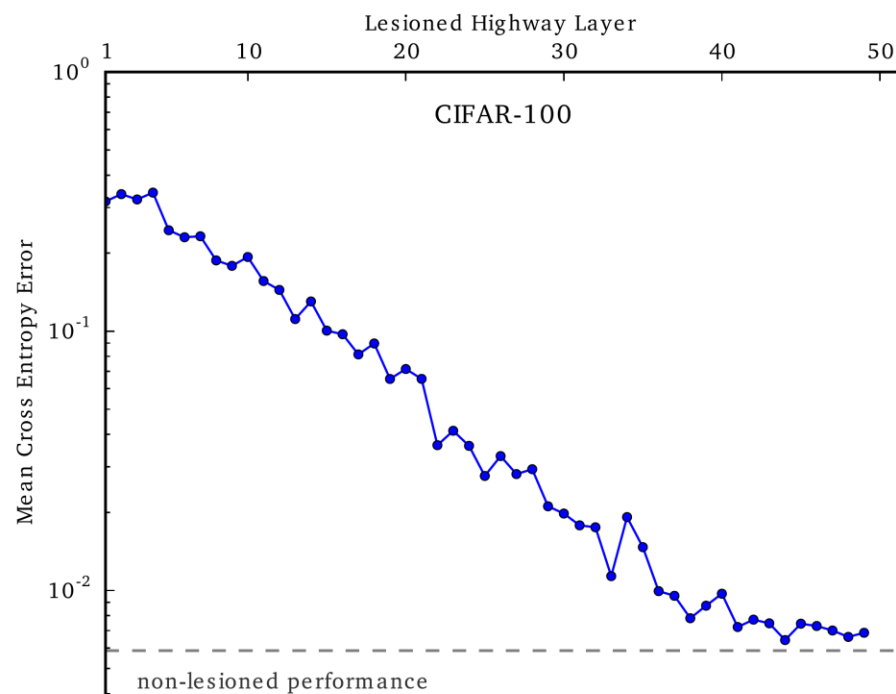
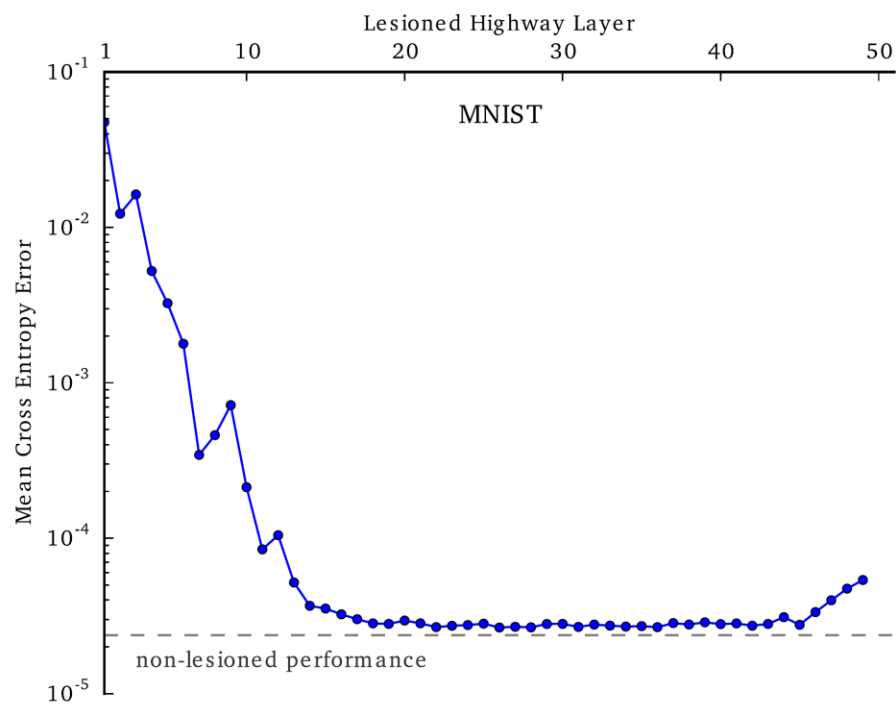
- **Residual Network**



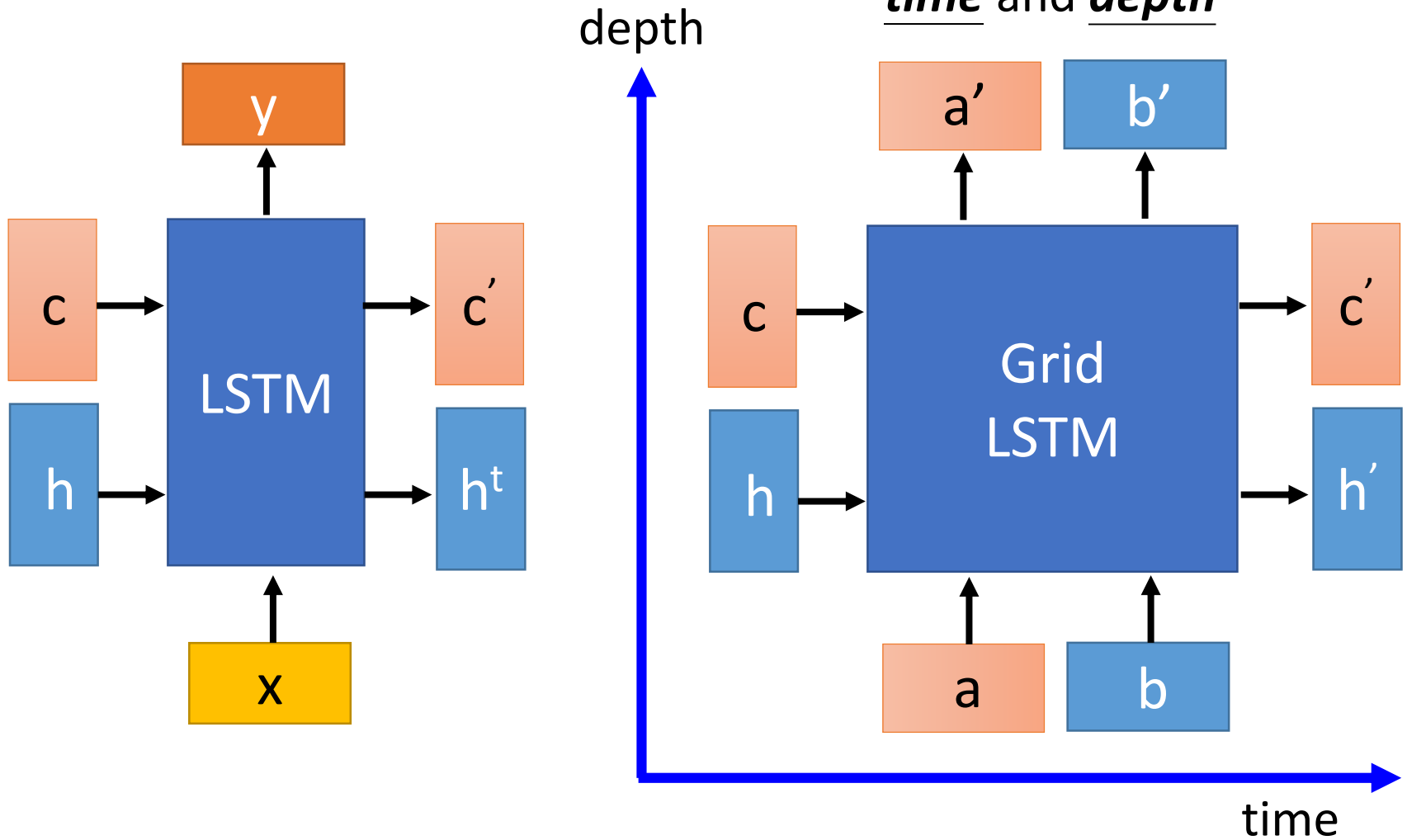
Deep Residual Learning for Image Recognition
<http://arxiv.org/abs/1512.03385>

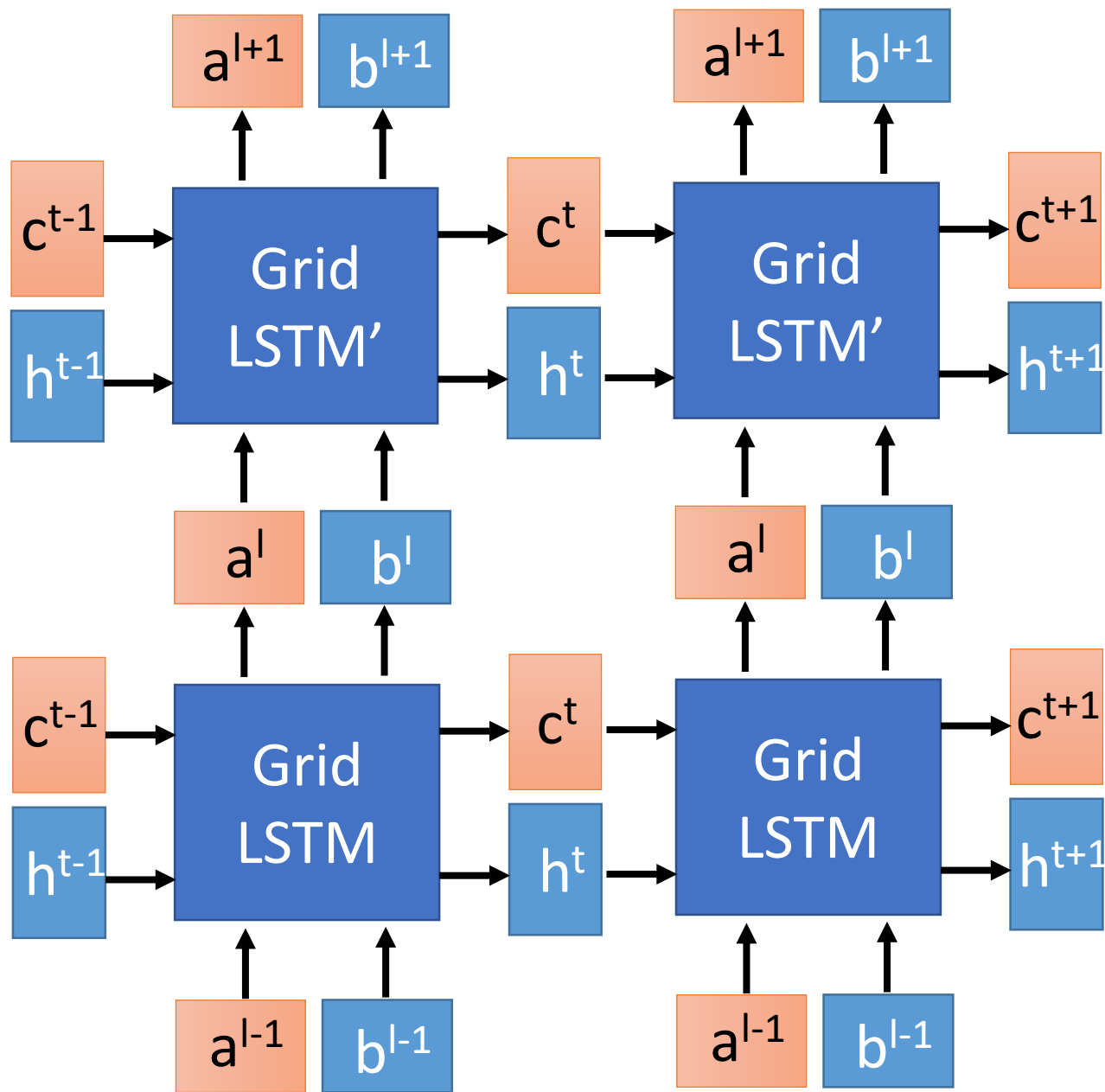


Highway Network

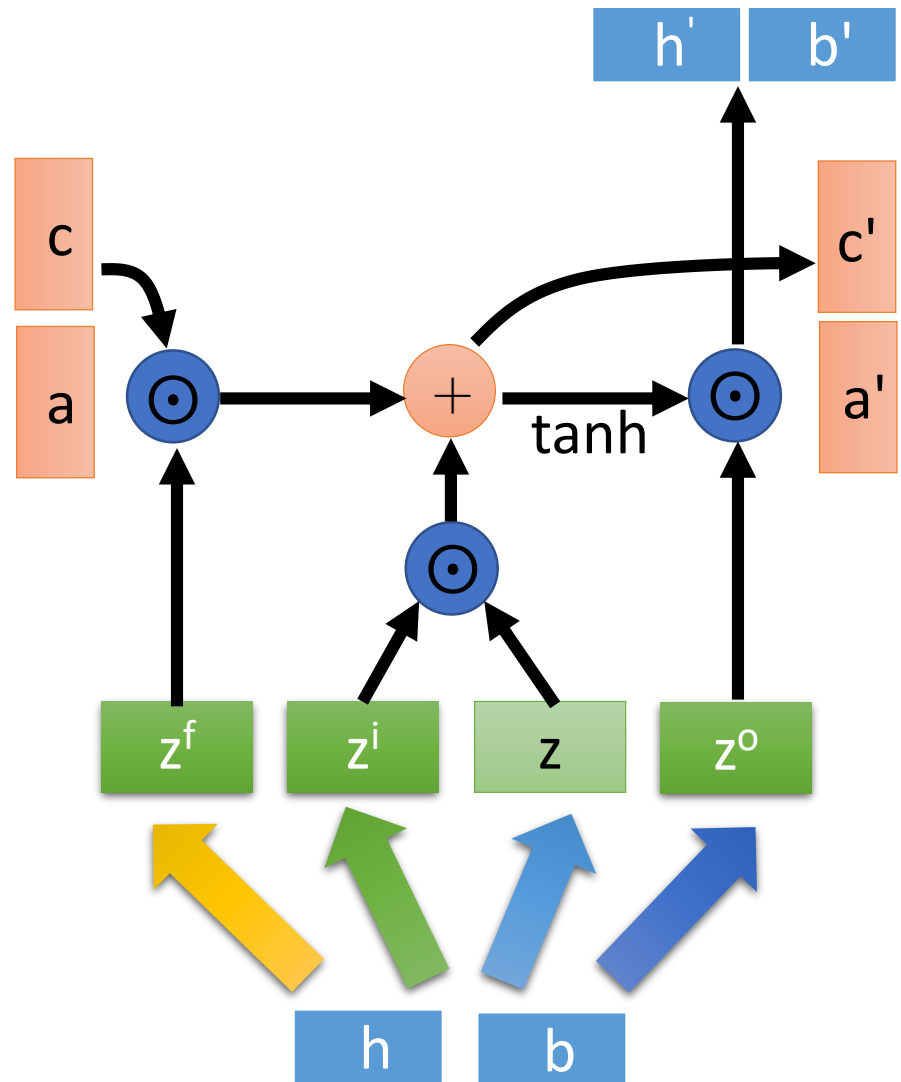
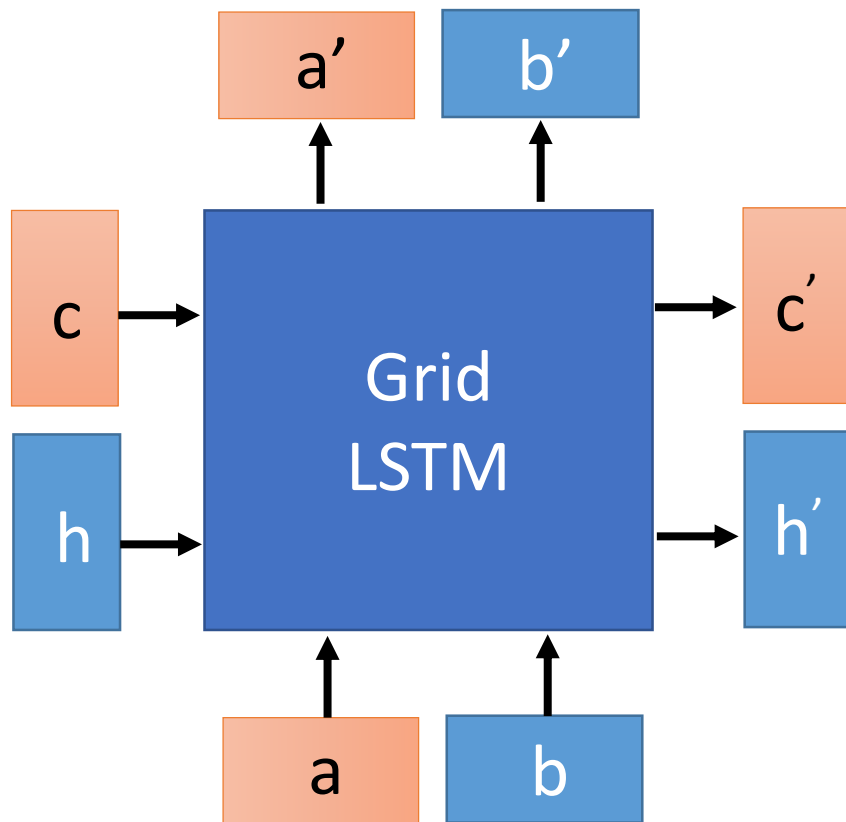


Grid LSTM

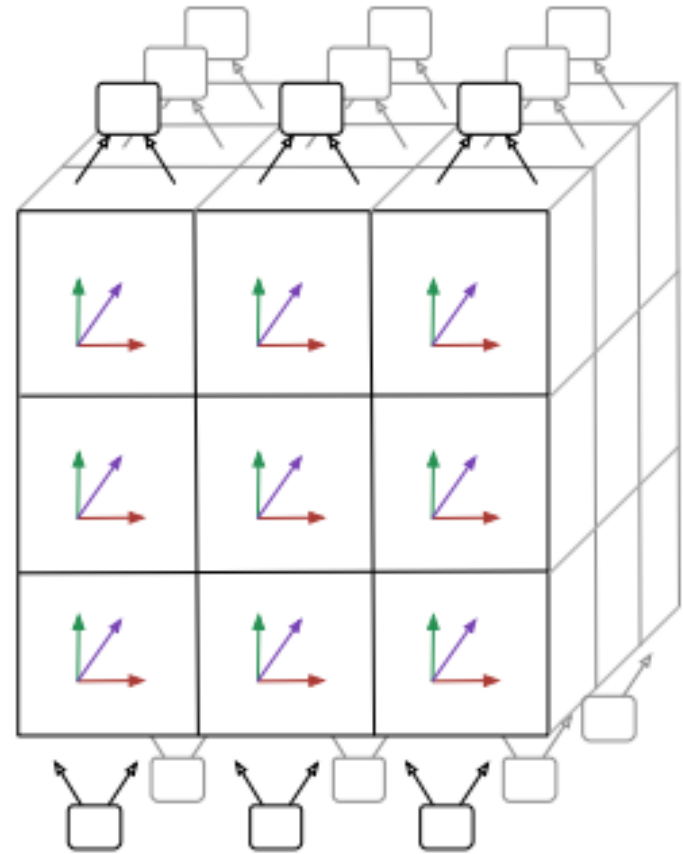
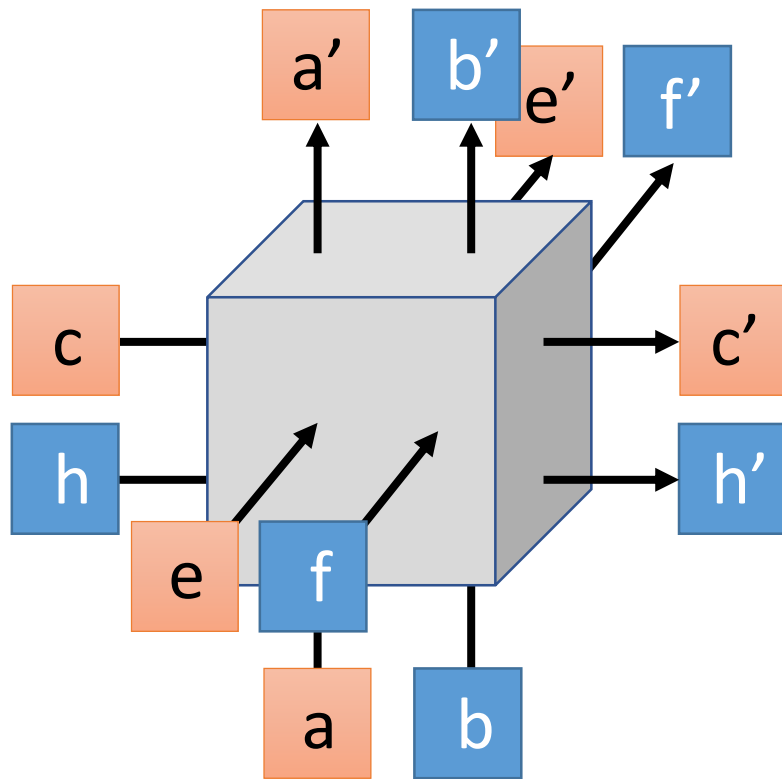




Grid LSTM

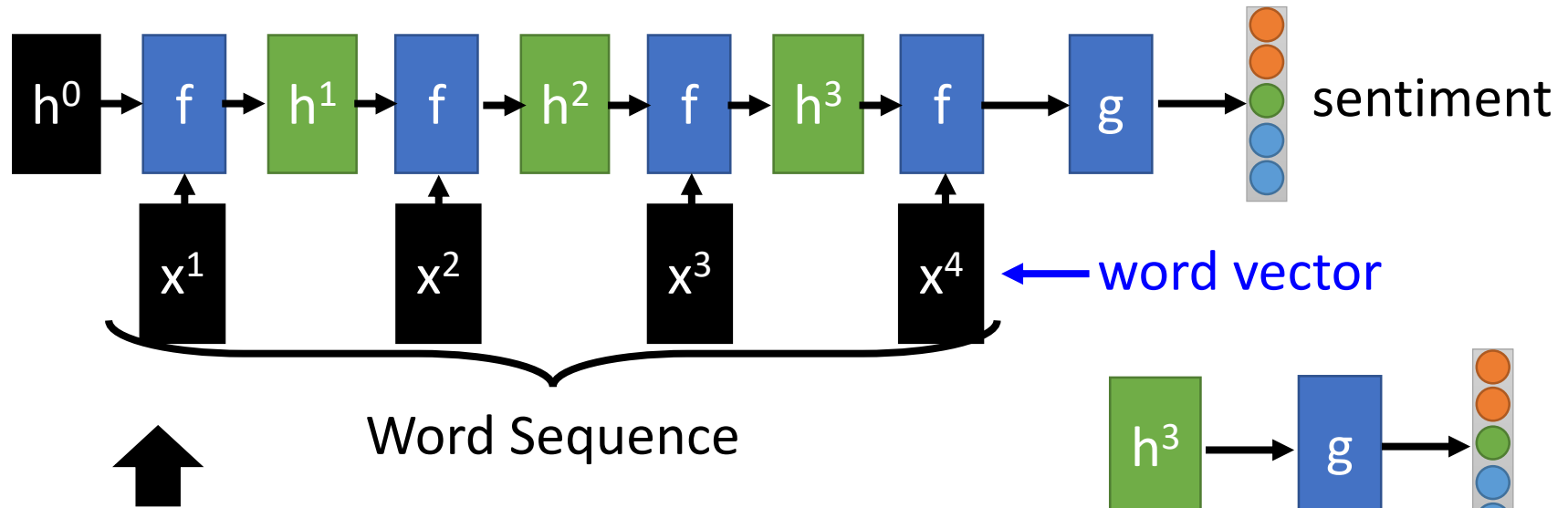


3D Grid LSTM



Recursive Structure

Application: Sentiment Analysis

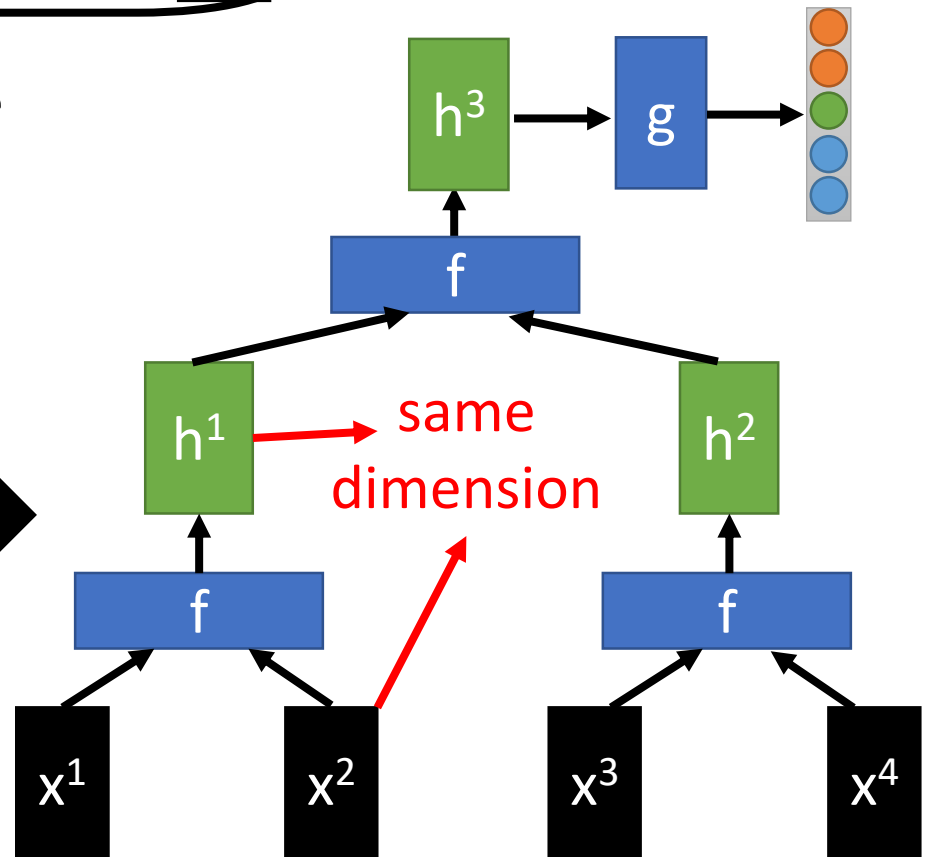
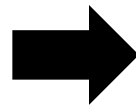


Recurrent Structure

Special case of recursive structure

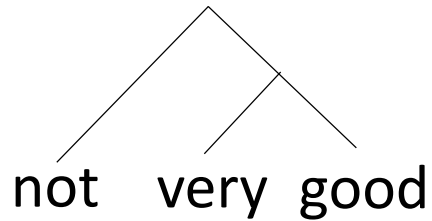
Recursive Structure

How to stack function f is already determined



Recursive Model

syntactic structure



How to do it is out
of the scope

word sequence:

not

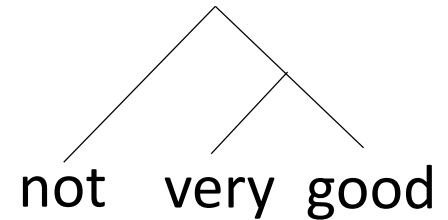
very

good

Recursive Model

By composing the two meaning, what should the meaning be.

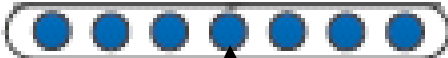
syntactic structure




Dimension of word vector = $|Z|$

Input: $2 \times |Z|$, output: $|Z|$

Meaning of "very good"


 $V(\text{"very good"})$

f

 $V(\text{"not"})$

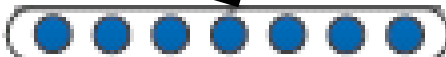
$V(\text{"not"})$

not

 $V(\text{"very"})$

$V(\text{"very"})$

very

 $V(\text{"good"})$

$V(\text{"good"})$

good

Recursive Model

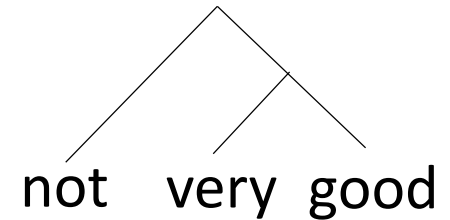
$$V(w_A w_B) \neq V(w_A) + V(w_B)$$

“not”: neutral

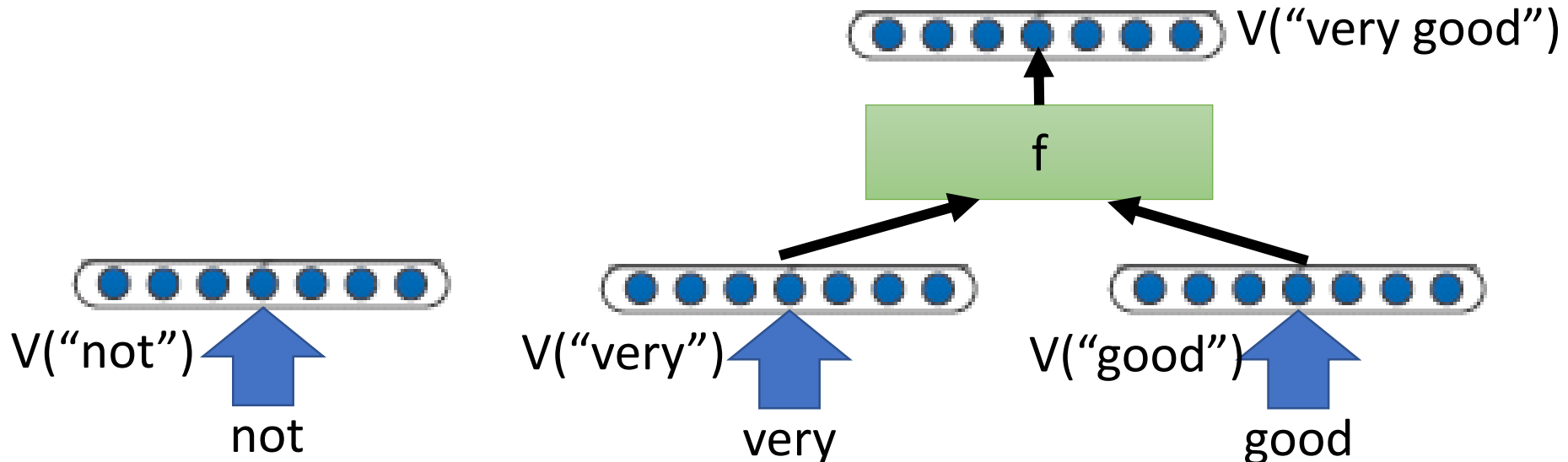
“good”: positive

“not good”: negative

syntactic structure



Meaning of “very good”



Recursive Model

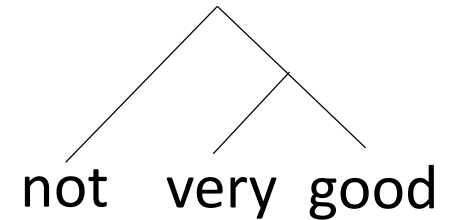
$$V(w_A w_B) \neq V(w_A) + V(w_B)$$

“棒”: positive

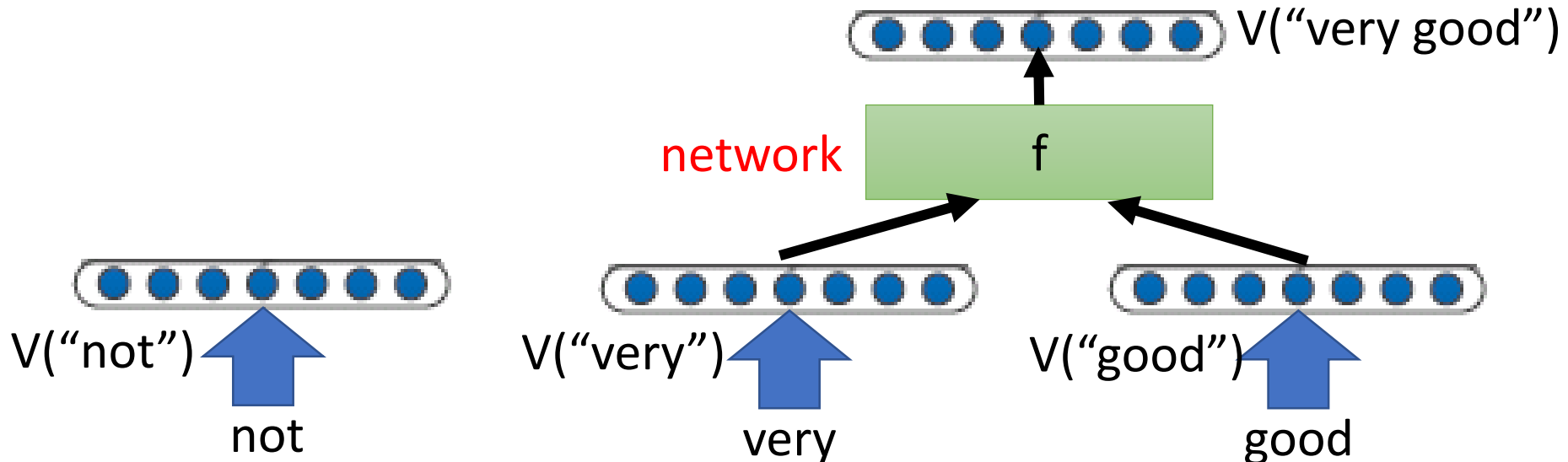
“好棒”: positive

“好棒棒”: negative

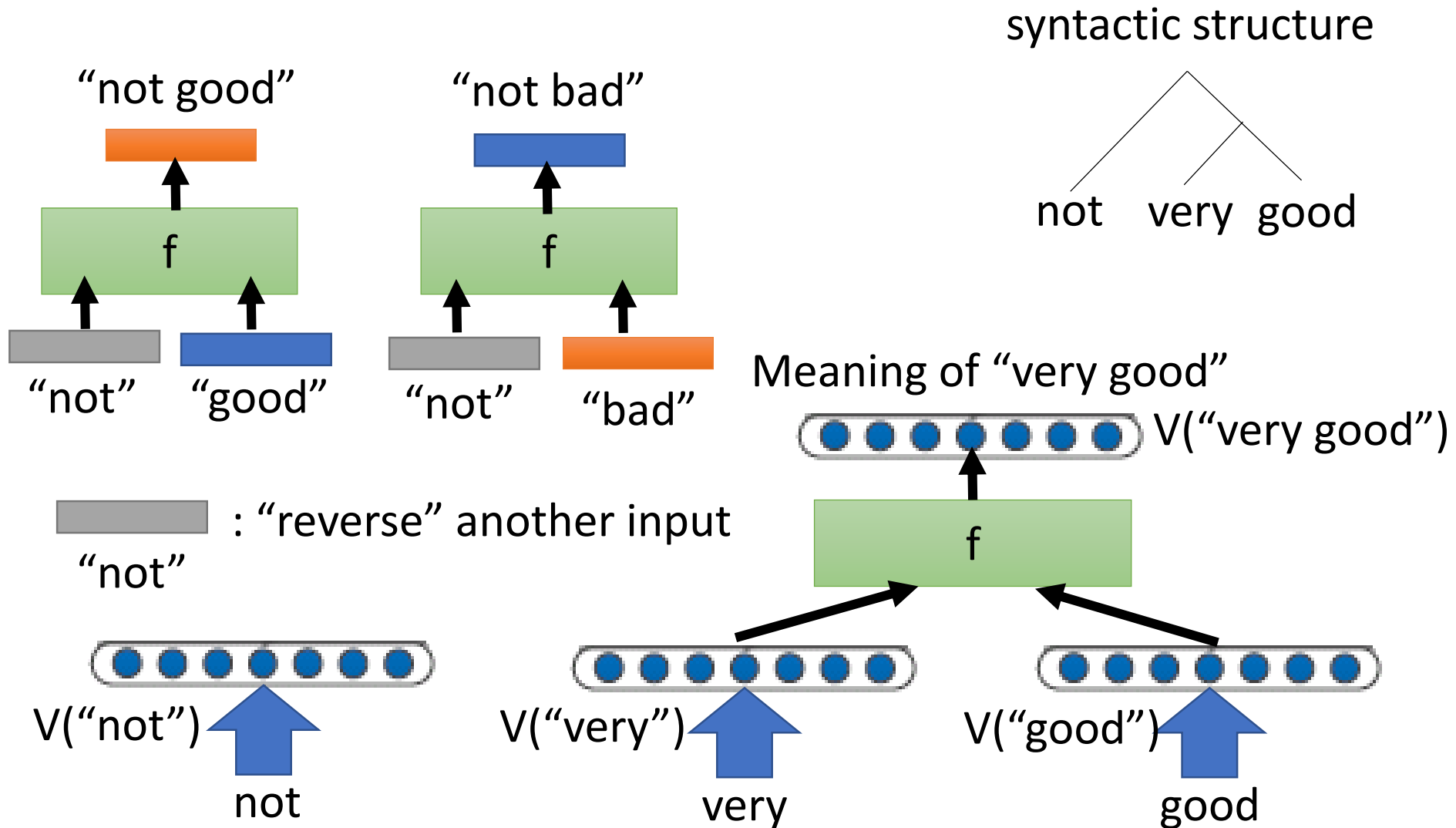
syntactic structure



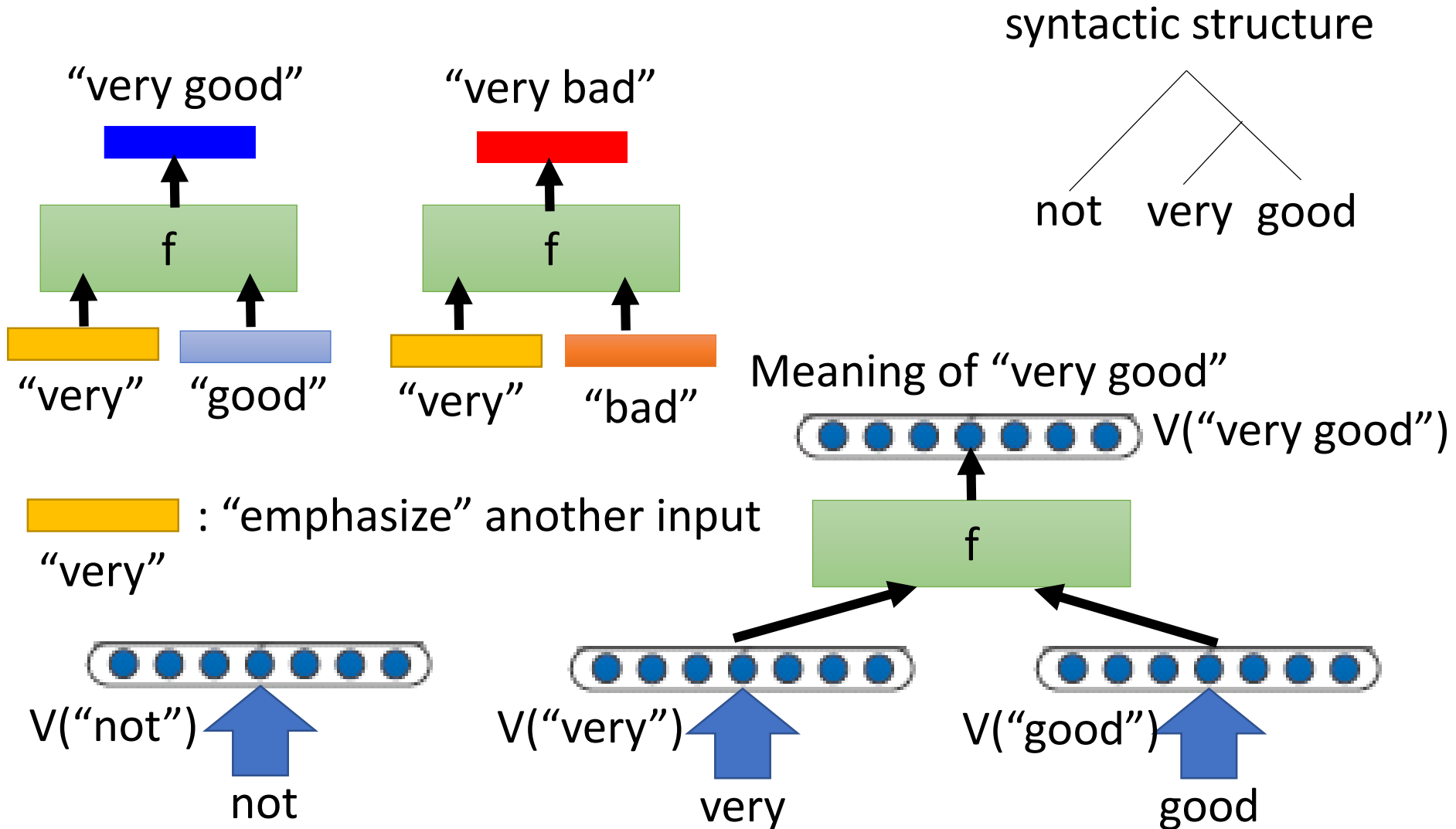
Meaning of “very good”

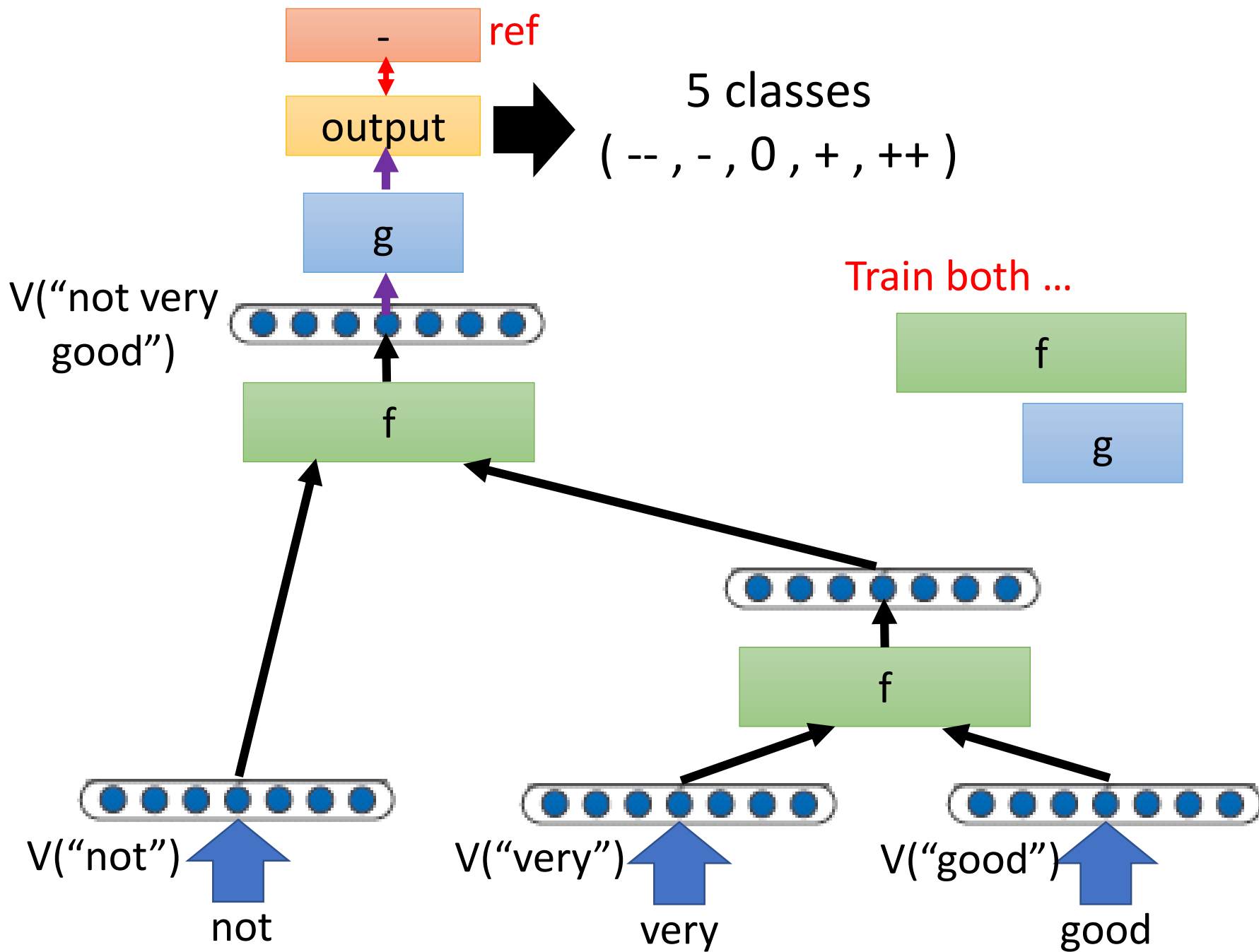


Recursive Model



Recursive Model

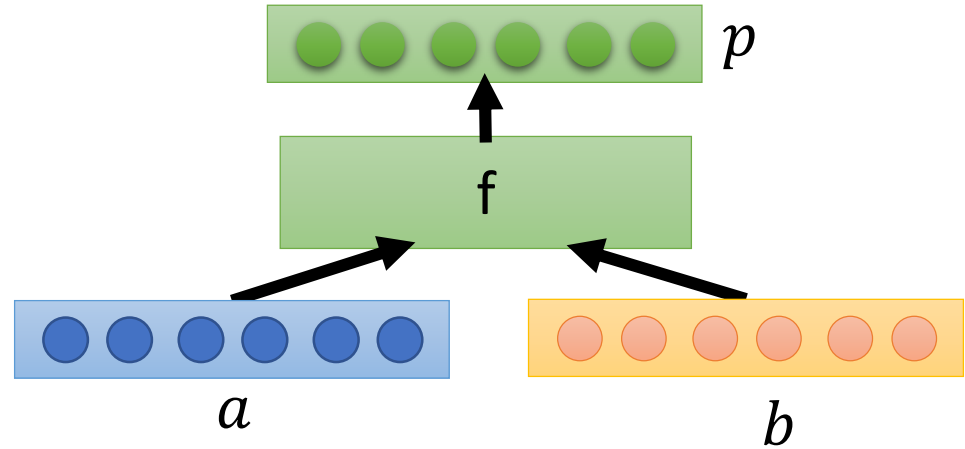




Recursive Neural Tensor Network

$$\begin{bmatrix} \bullet \\ \bullet \end{bmatrix} = \sigma \left(\begin{bmatrix} W \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \end{bmatrix} \right)$$

Little interaction between
a and b



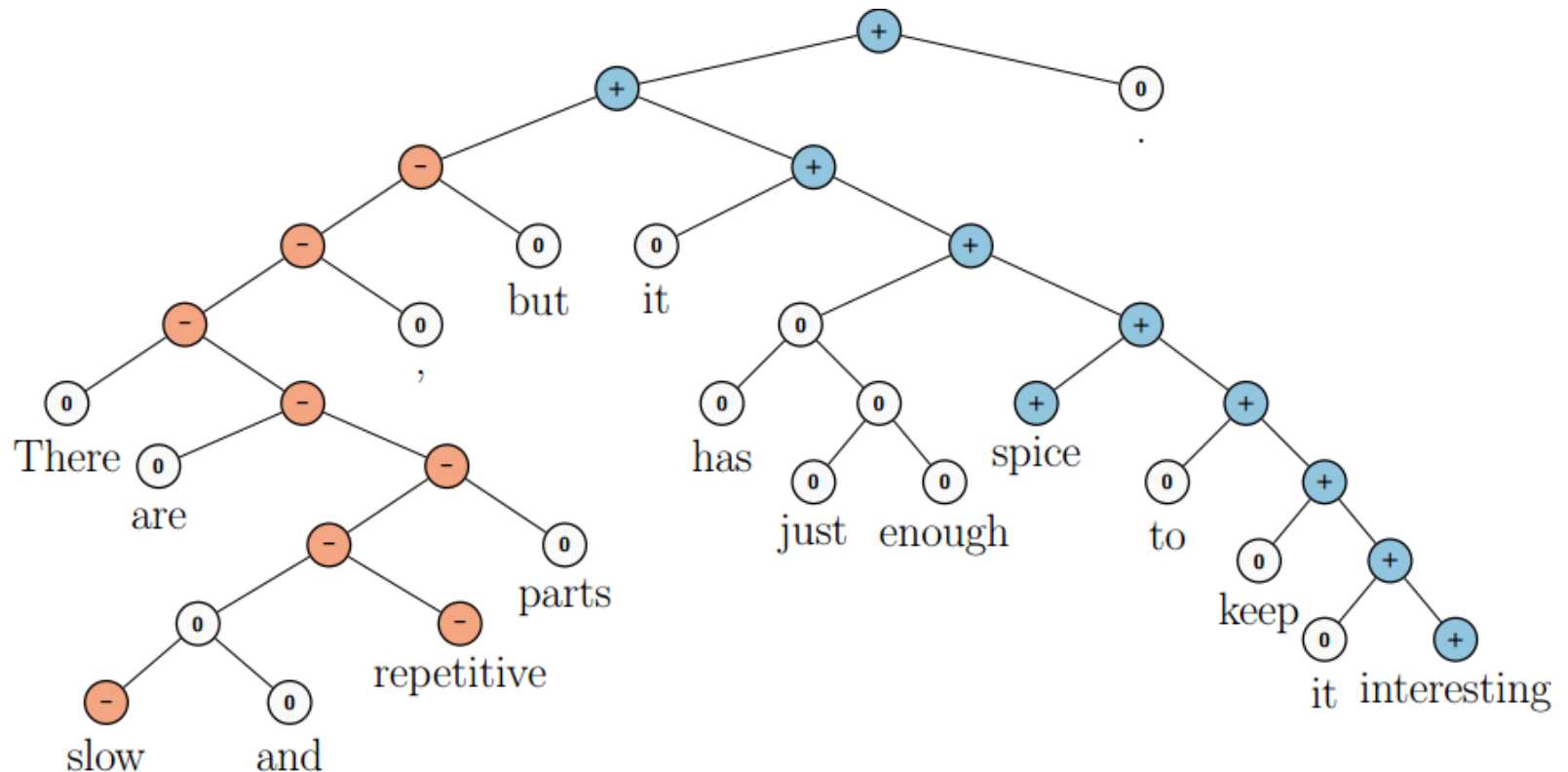
$$\begin{bmatrix} \bullet \\ \bullet \end{bmatrix} = \sigma \left(\begin{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet \end{bmatrix} \\ \begin{bmatrix} \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \end{bmatrix} \end{bmatrix} \begin{bmatrix} W \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \end{bmatrix} \right) + \sum_{i,j} W_{ij} x_i x_j \begin{bmatrix} \bullet \\ \bullet \end{bmatrix}$$

Diagram illustrating a more complex neural network layer. The input vectors a (blue) and b (orange) are fed into a function block f . The output of f is a vector p (green). The function f is defined as:

$$f(a, b) = \sigma \left(\begin{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet \end{bmatrix} \\ \begin{bmatrix} \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \end{bmatrix} \end{bmatrix} \begin{bmatrix} W \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \end{bmatrix} \right) + \sum_{i,j} W_{ij} x_i x_j \begin{bmatrix} \bullet \\ \bullet \end{bmatrix}$$

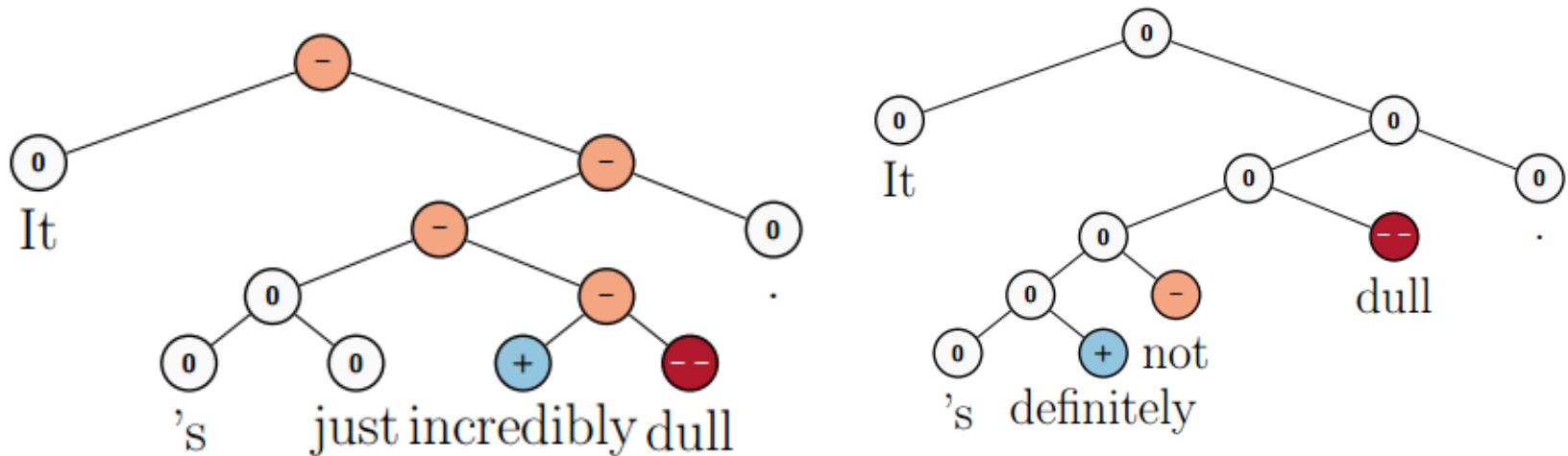
Experiments

5-class sentiment classification (-- , - , 0 , + , ++)



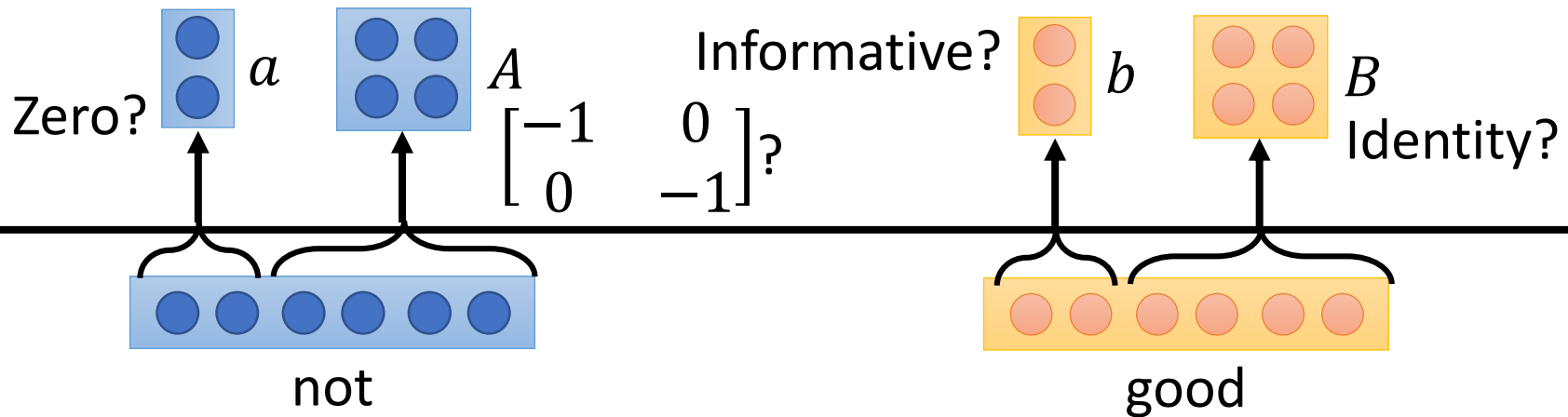
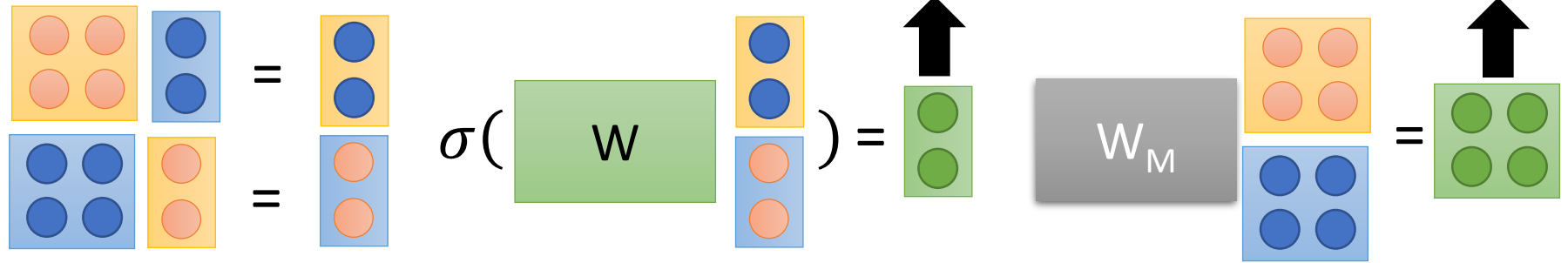
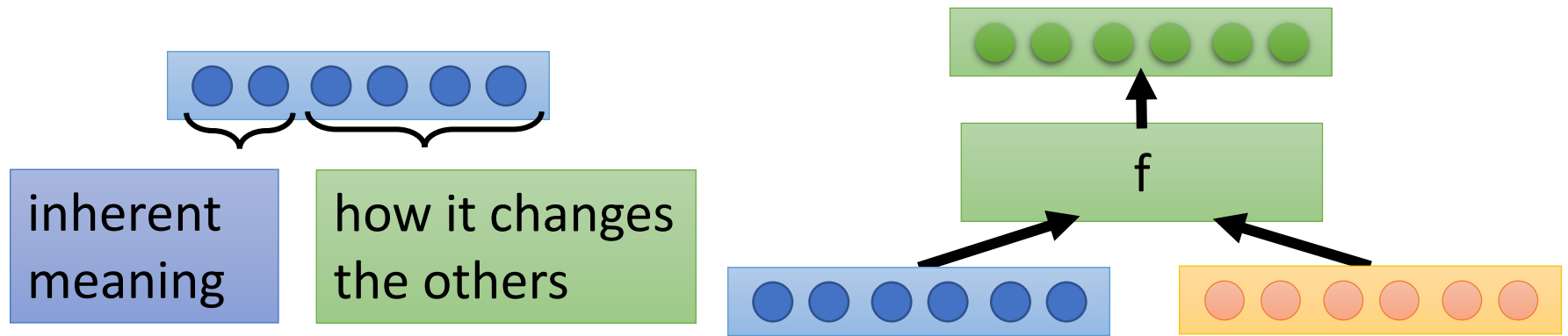
Demo: <http://nlp.stanford.edu:8080/sentiment/rntnDemo.html>

Experiments



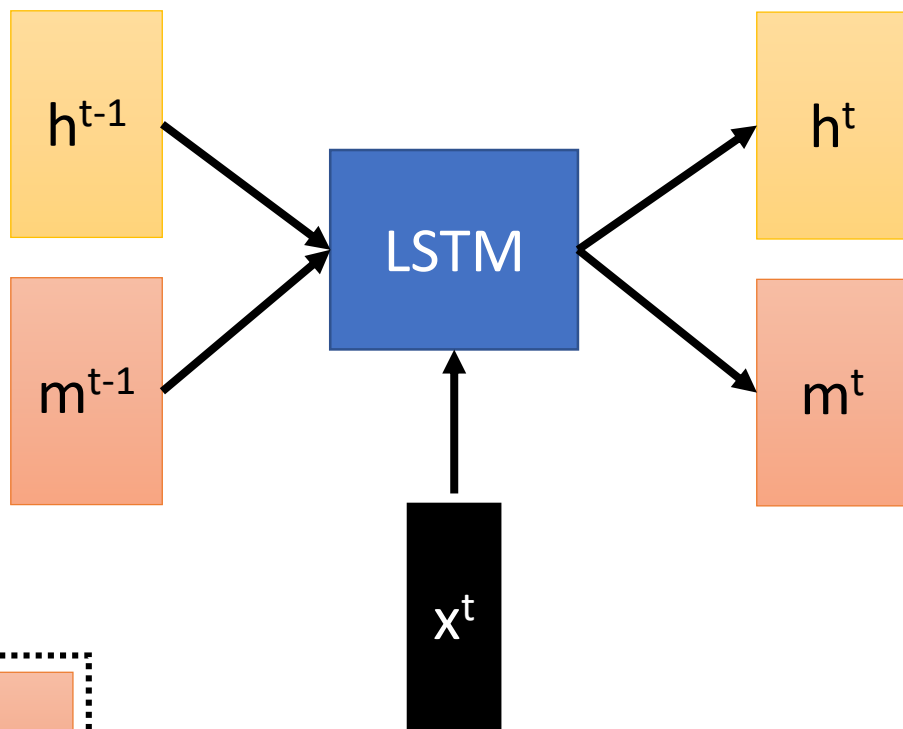
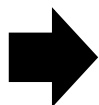
Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631. 2013.

Matrix-Vector Recursive Network

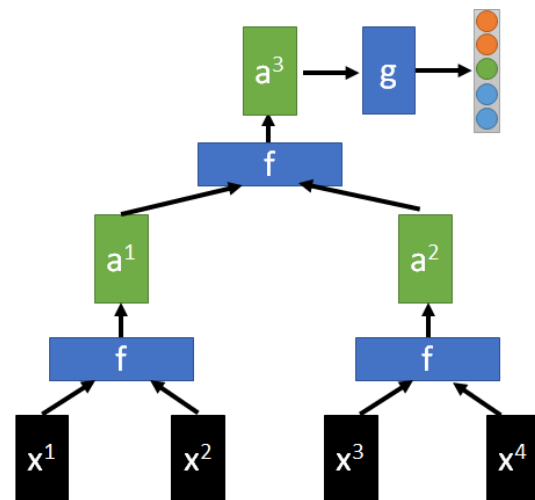
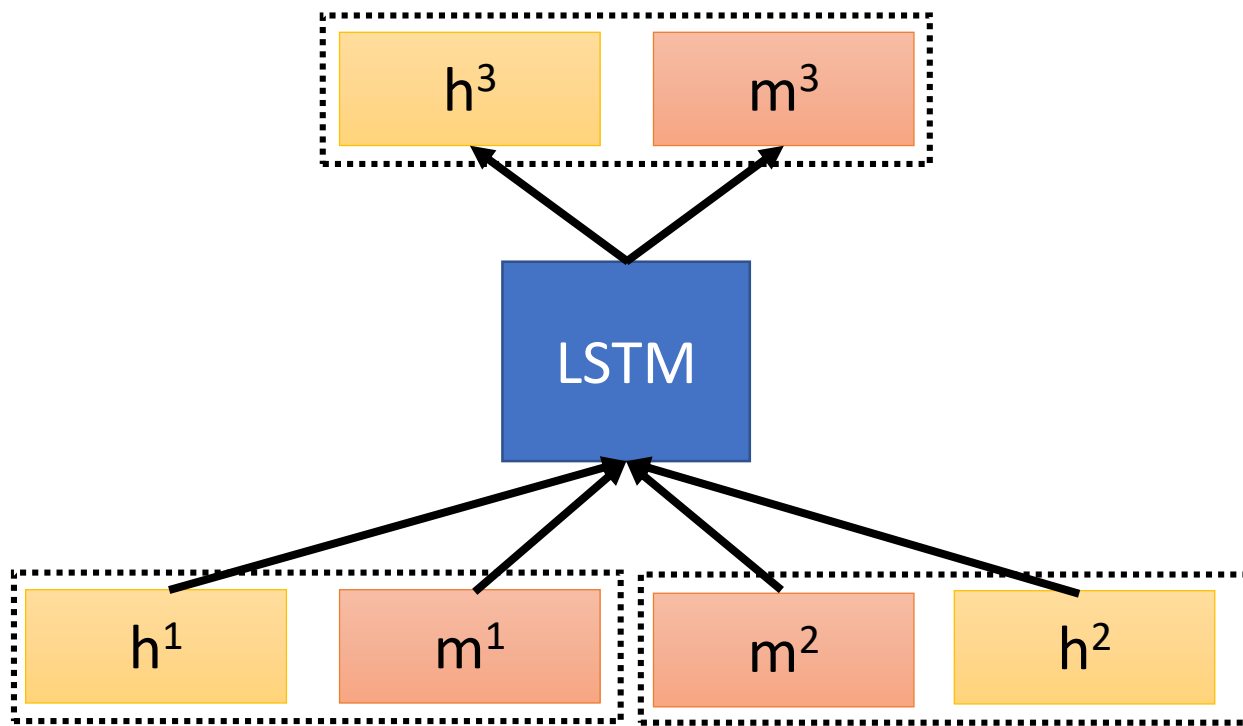


Tree LSTM

Typical LSTM



Tree LSTM



More Applications

- Sentence relatedness

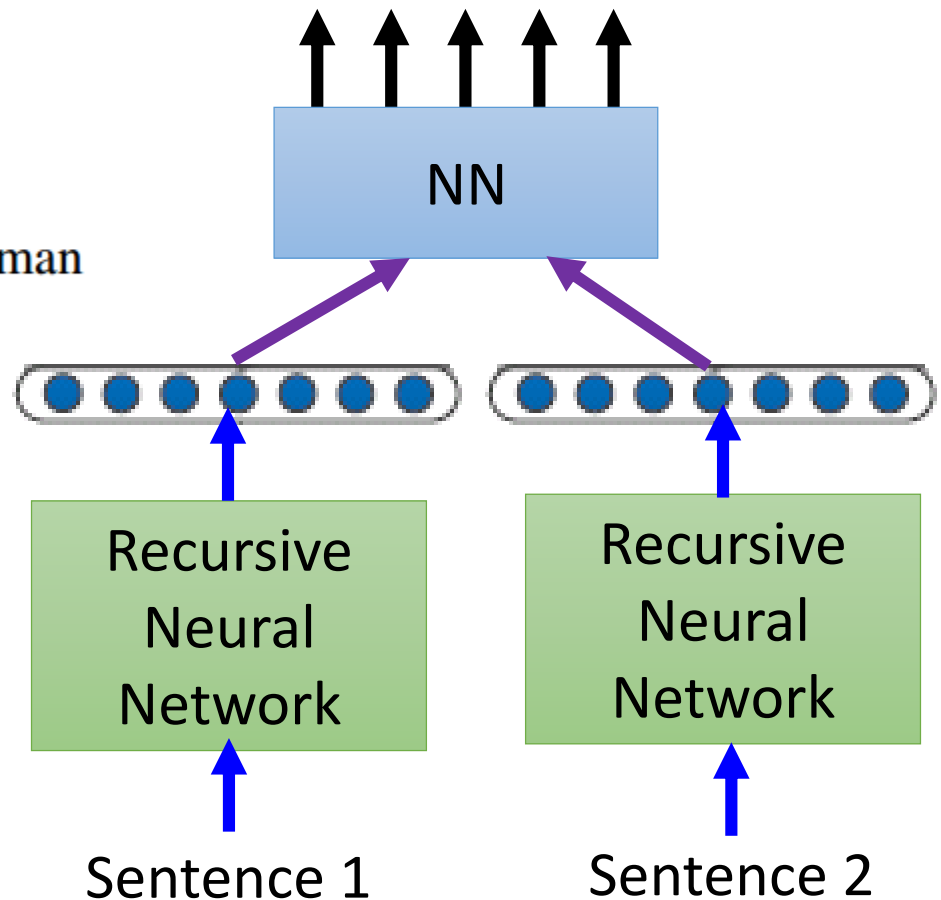
a woman is slicing potatoes

4.82 a woman is cutting potatoes

4.70 potatoes are being sliced by a woman

4.39 tofu is being sliced by a woman

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." *arXiv preprint arXiv:1503.00075* (2015).



Batch Normalization

Experimental Results

