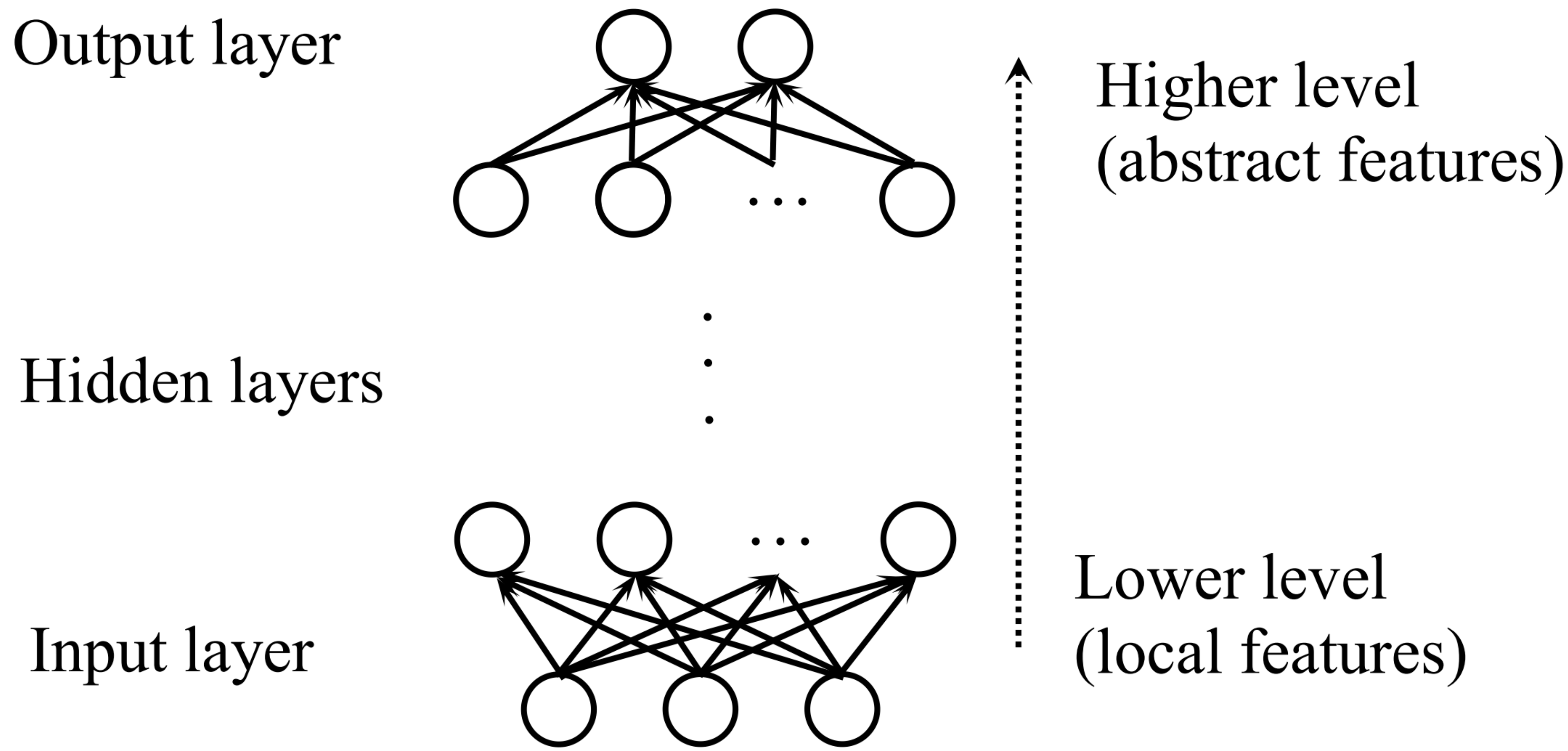
Convolutional Neural Networks & Recurrent Neural Networks

Lili Mou
lmou@ualberta.ca
lili-mou.github.io
UNIVERSITY OF ALBERTA

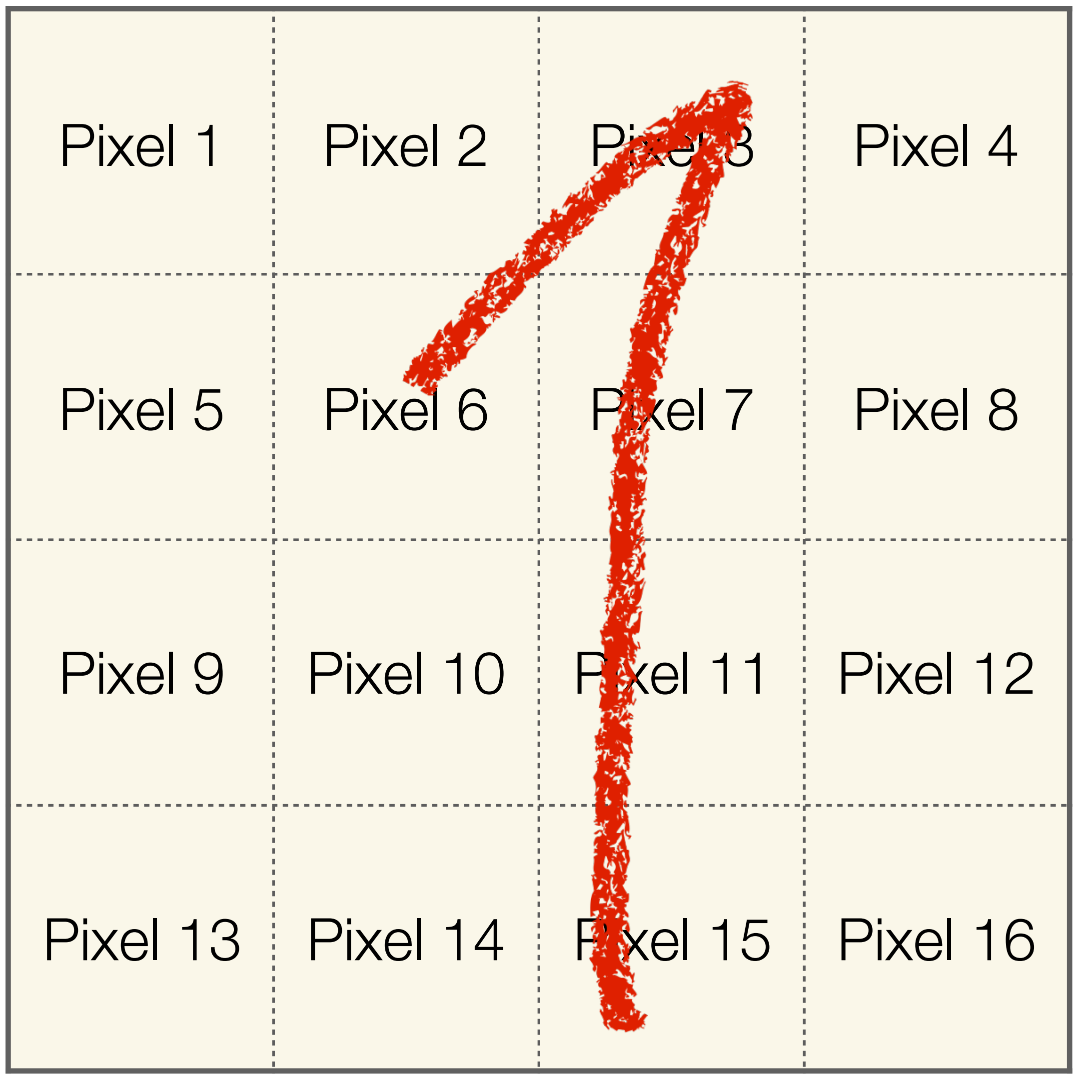
A Generic NN



Input

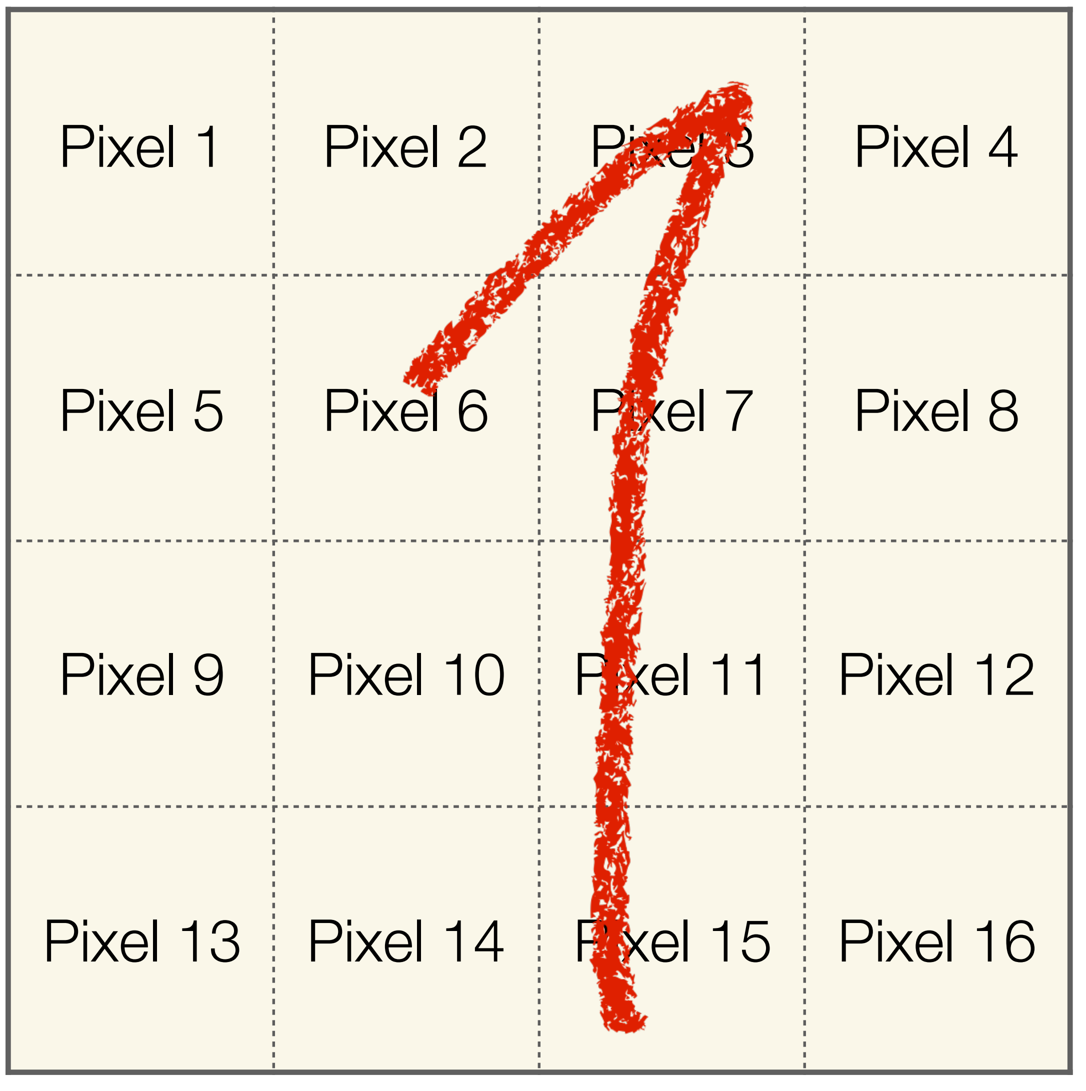
- Image of $n \times n$ pixels: n^2-dimensional features
- Text: embeddings (how about varying-length sentences?)

A Generic NN



$$
\begin{array}{l}
\text{Pixel 1} \\ \text{Pixel 2} \\ \text{Pixel 3} \\ \text{Pixel 4} \\ \text{Pixel 5} \\ \text{Pixel 6} \\ \text{Pixel 7} \\ \text{Pixel 8} \\ \text{Pixel 9} \\ \text{Pixel 10} \\ \text{Pixel 11} \\ \text{Pixel 12} \\ \text{Pixel 13} \\ \text{Pixel 14} \\ \text{Pixel 15} \\ \text{Pixel 16}
\end{array}
\begin{bmatrix}
\\ \\ 1 \\ \\ \\ 1 \\ 1 \\ \\ \\ \\ 1 \\ \\ \\ \\ 1 \\ \\
\end{bmatrix}
$$

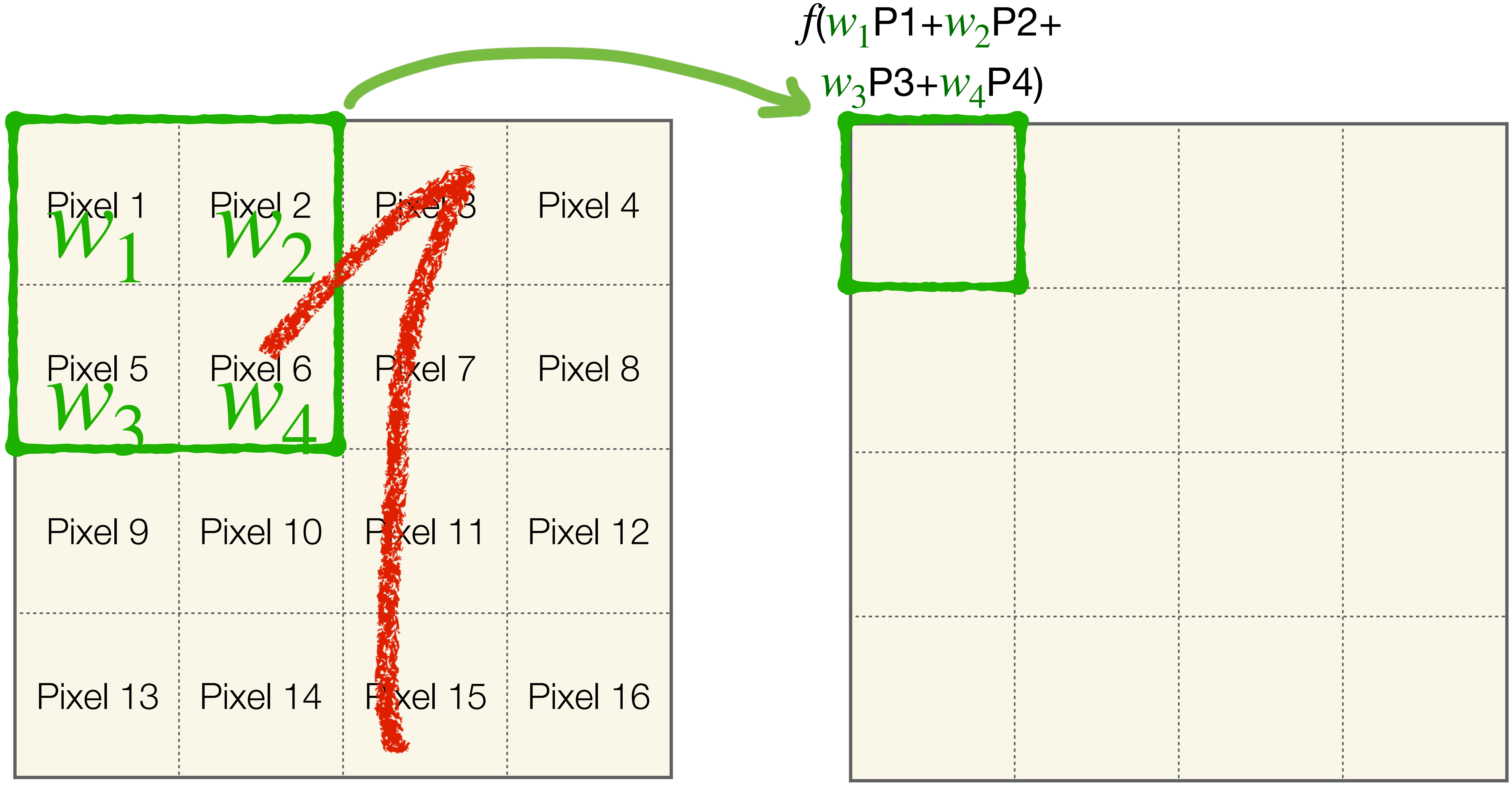
Intuition



Pixel 3 is most similar to Pixels 2, 4, and 7.

A layer-wise fully connected NN cannot capture such prior knowledge

Convolutional Neural Networks

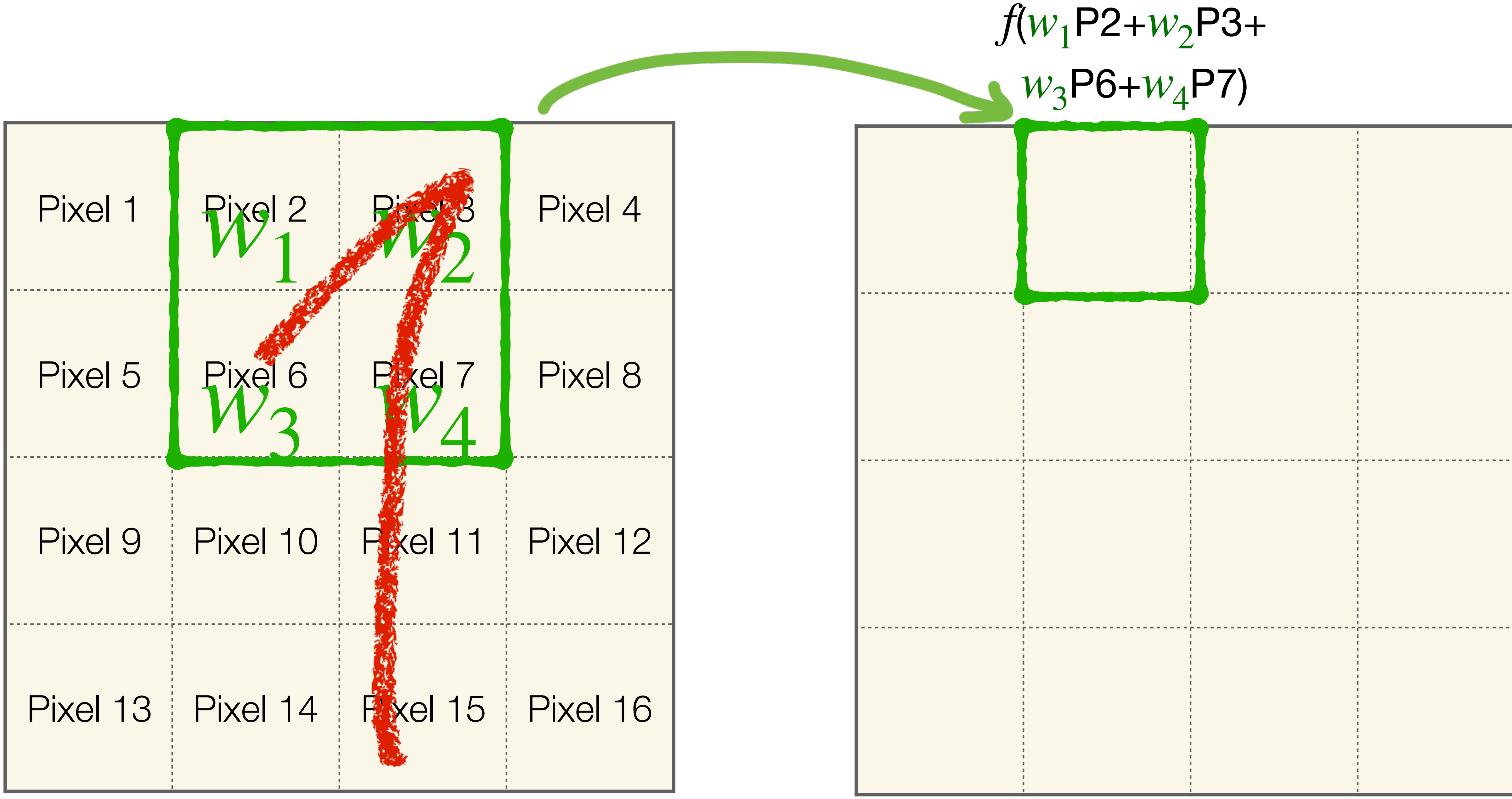


Local feature detector (kernel, filter, window)

- Working with a local “neighborhood”

UNIVERSITY OF ALBERTA

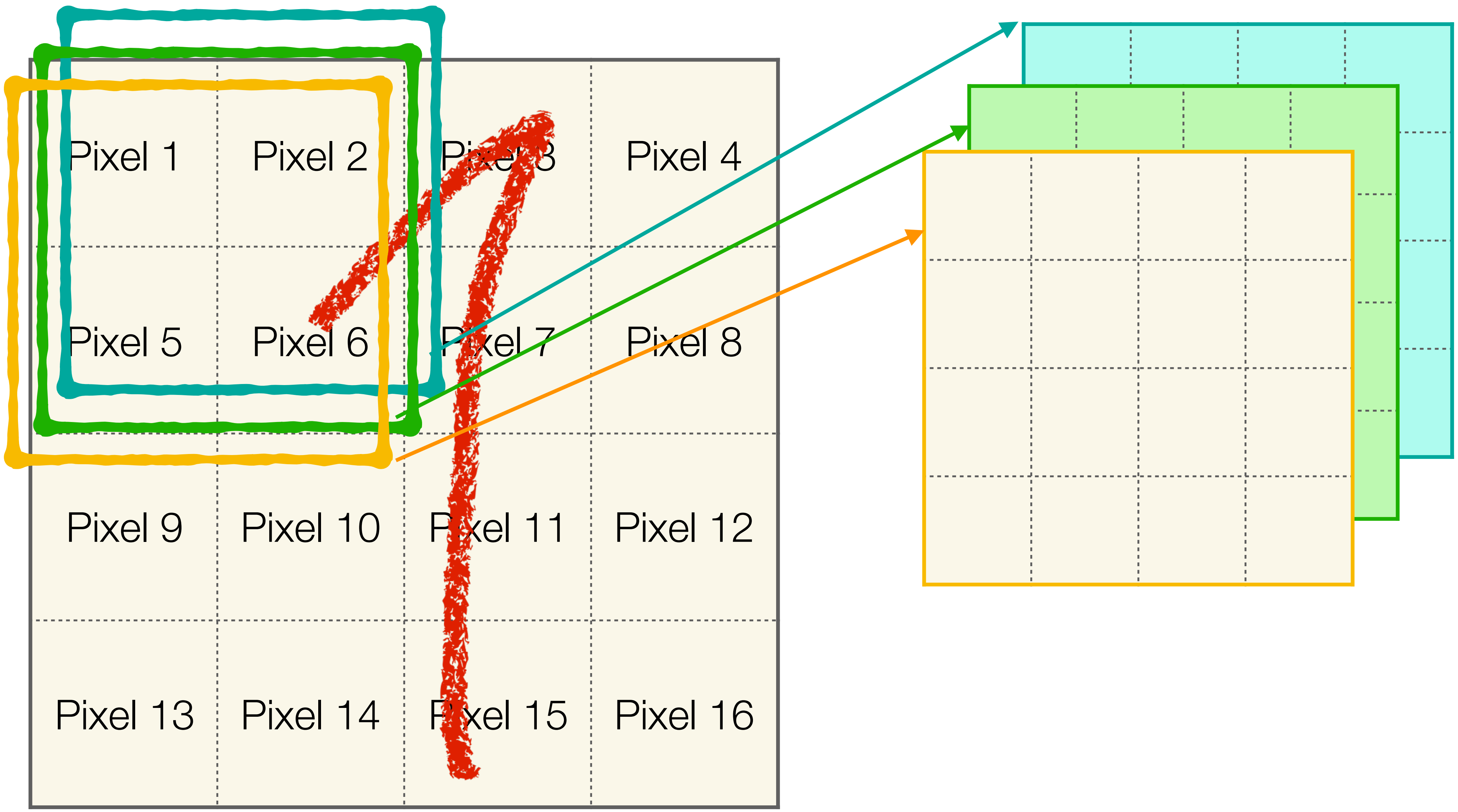
Convolutional Neural Networks



Local feature detector (kernel, filter, window)

- Working with a local "neighborhood"
- Moving over the entire image

Multiple Filters



Local feature detector (kernel, filter, window)

- Working with a local “neighborhood”
- Moving over the entire image

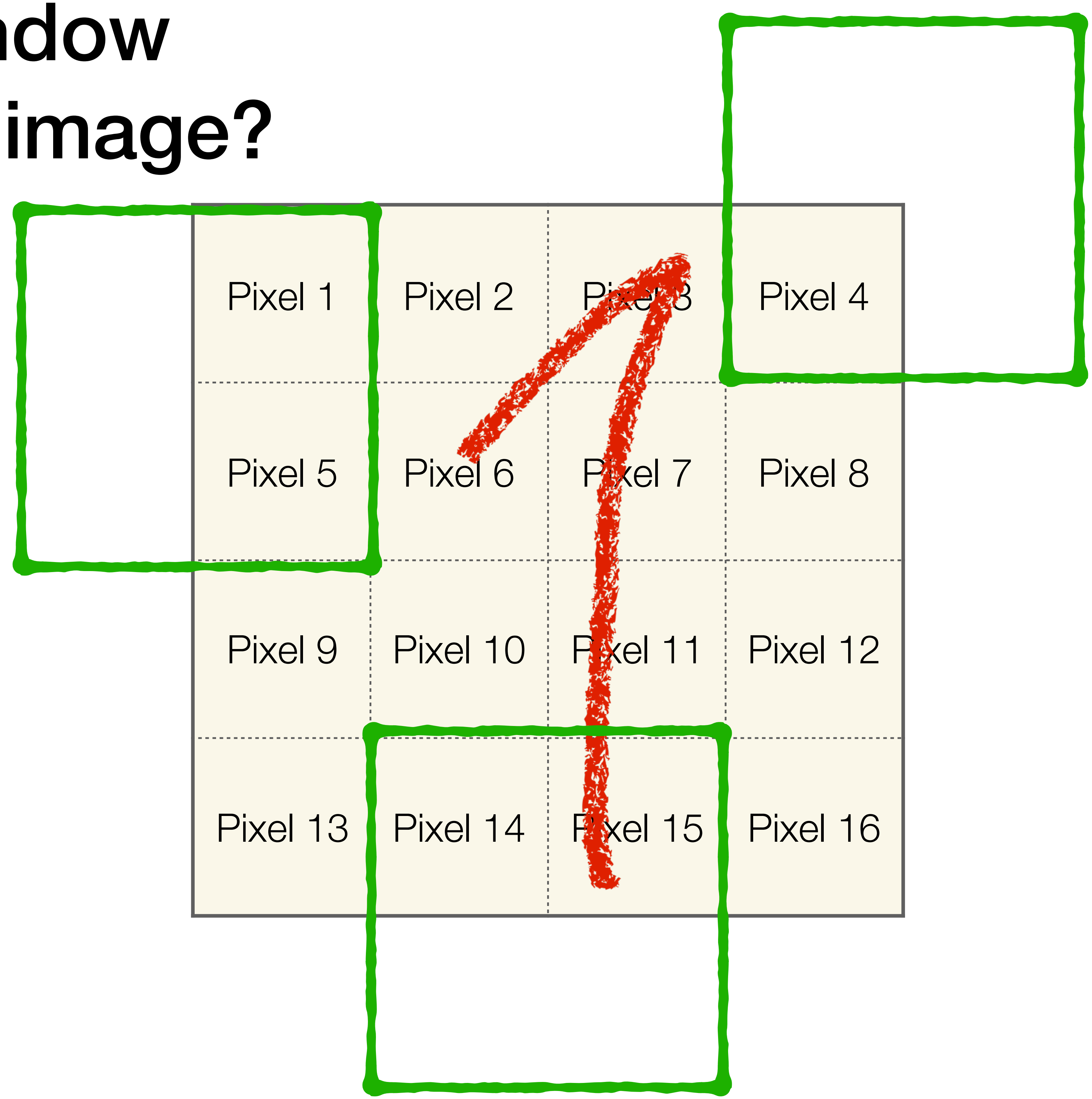
What if the window moves out of the image?

Padding:

- No padding (shrinking)
- Zero-padding
- "Same" padding
- etc.

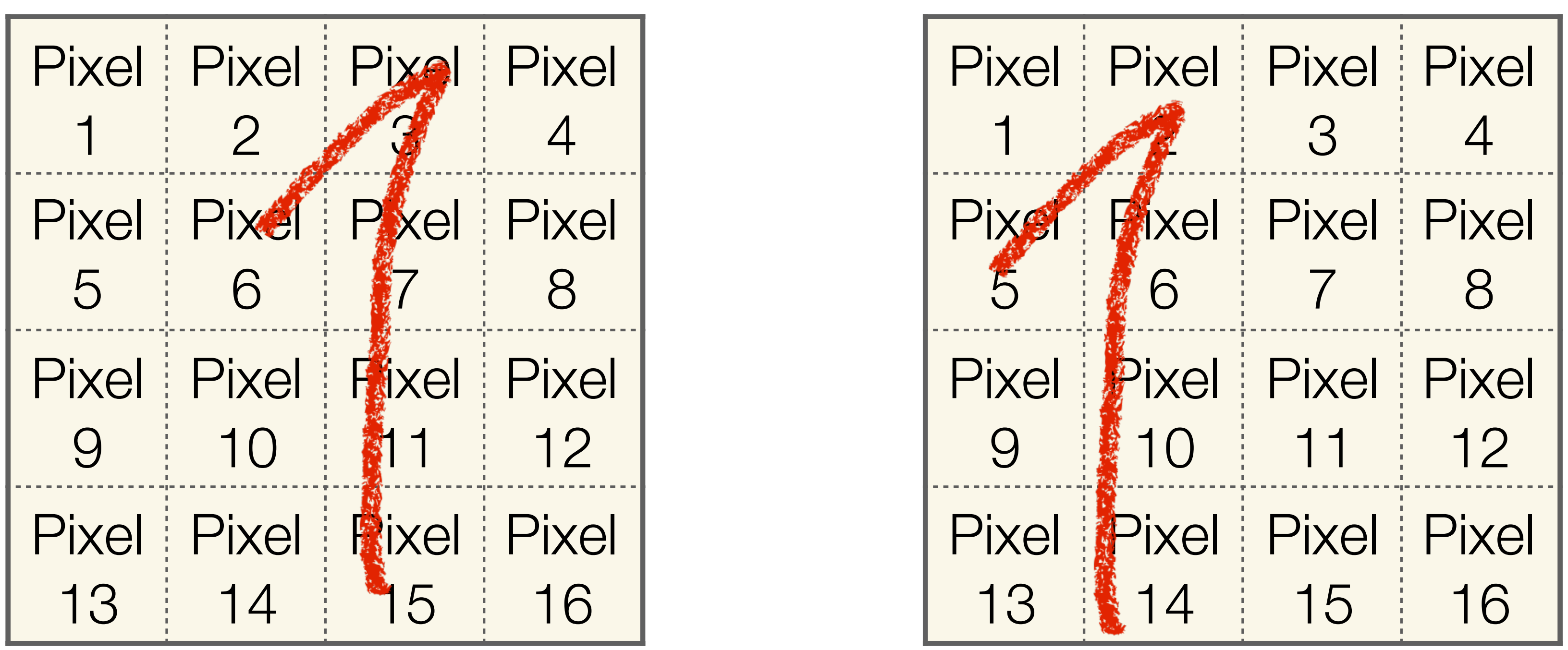
Other hyperparameters:

- Window size
- Stride



Intuition of CNN

- CNN defines spatial neighbourhoods
  - Our prior knowledge
- The feature of a window is spatial invariant
  - The features of a convolutional layer is spatial equivariant



Convolution in Signal Processing

- 1-D convolution [continuous]

If $f, g : \mathbb{R} \to \mathbb{R}$, then $f * g : \mathbb{R} \to \mathbb{R}$

$$(f * g)(t) \stackrel{def}{=} \int_{-\infty}^{\infty} f(x)g(t-x)dx$$

- 2-D convolution [continuous]

If $f, g : \mathbb{R}^2 \to \mathbb{R}$, then $f ** g : \mathbb{R}^2 \to \mathbb{R}$

$$(f * g)(t_1, t_2) \stackrel{def}{=} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, x_2)g(t_1 - x_1, t_2 - x_2)dx_1dx_2$$

Convolution in Signal Processing

- Continuous [2D]

If $f, g : \mathbb{R}^2 \to \mathbb{R}$, then $f ** g : \mathbb{R}^2 \to \mathbb{R}$

$$(f * g)(t_1, t_2) \stackrel{def}{=} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2) g(t_1 - x_1, t_2 - x_2) dx_1 dx_2$$

- Discrete [2D]

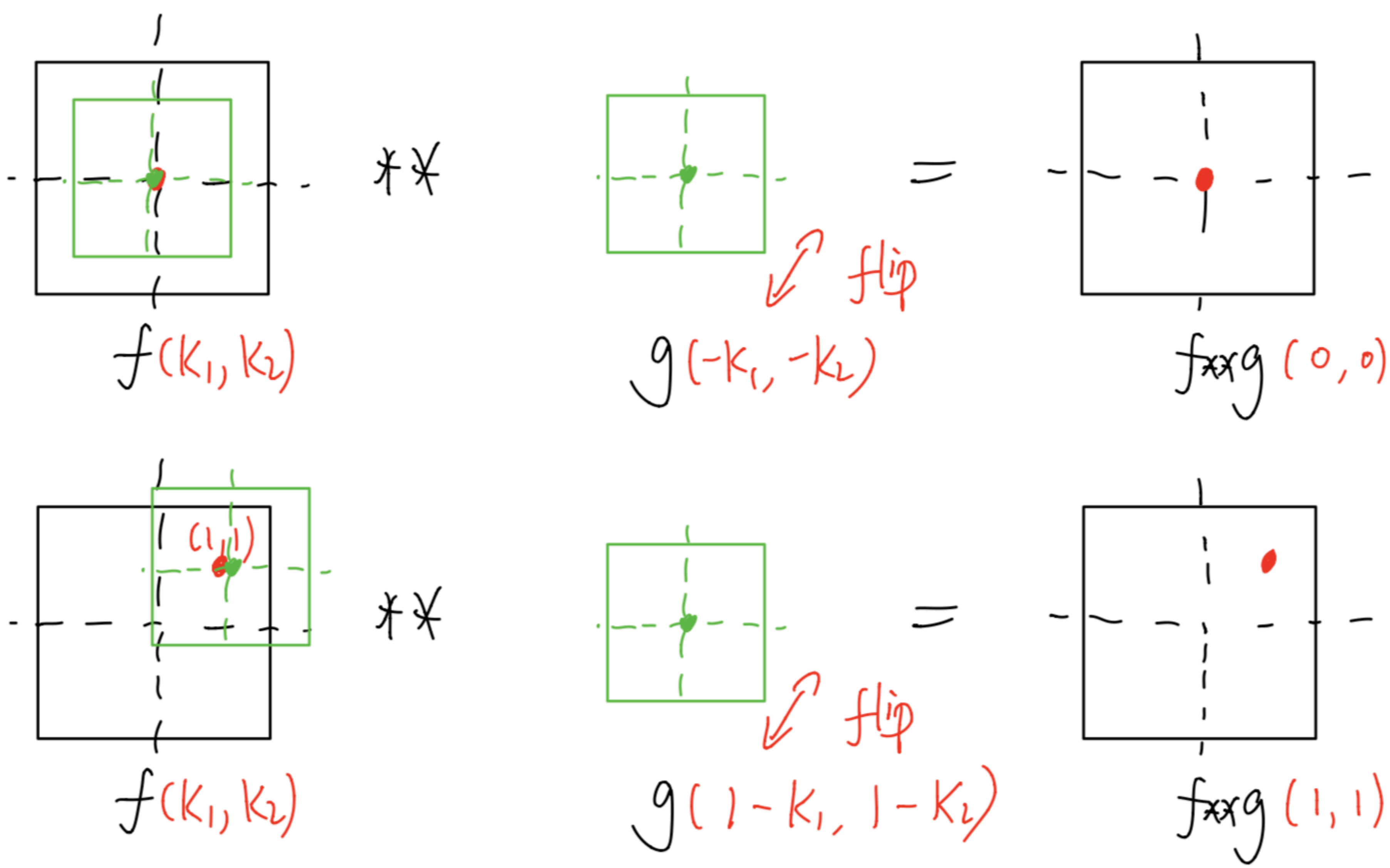
If $f, g : \mathbb{Z}^2 \to \mathbb{R}$, then $f ** g : \mathbb{Z}^2 \to \mathbb{R}$

$$(f * g)(n_1, n_2) \stackrel{def}{=} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2) g(n_1 - k_1, n_2 - k_2)$$

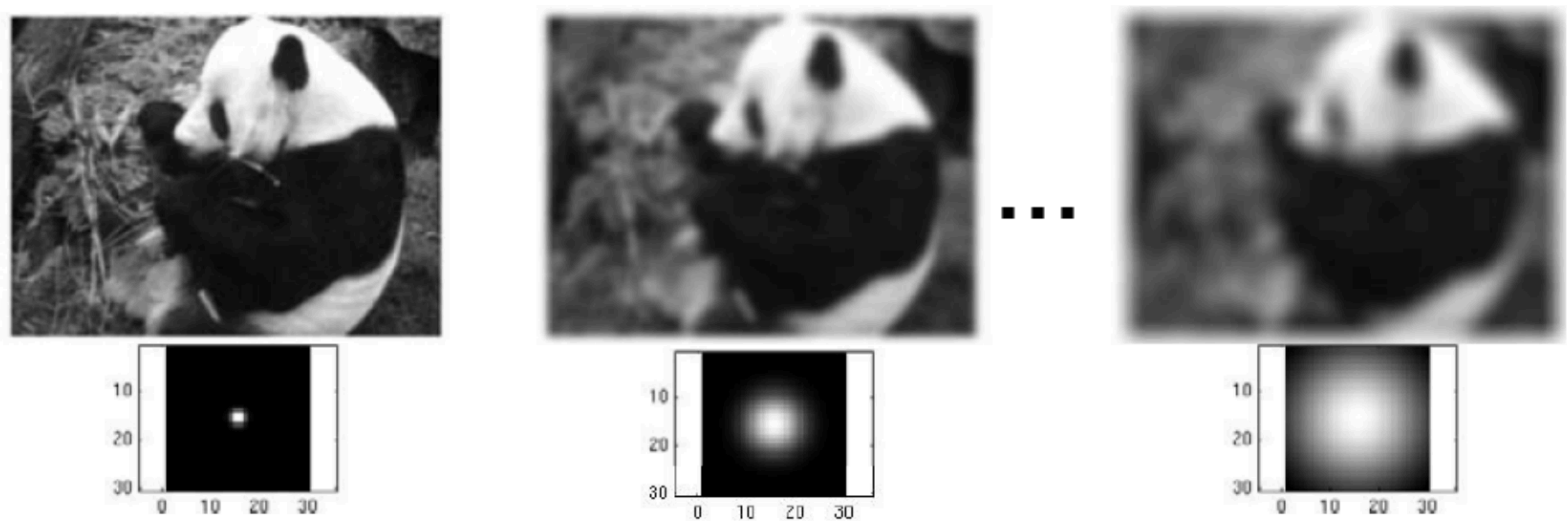
Covolution (2D discrete)

$f, g : \mathbb{Z}^2 \to \mathbb{R} \qquad f ** g : \mathbb{Z}^2 \to \mathbb{R}$

$$(f ** g)(n_1, n_2) \stackrel{\text{def}}{=} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1, k_2)\, g(n_1 - k_1, n_2 - k_2)$$



Low-Pass Filter



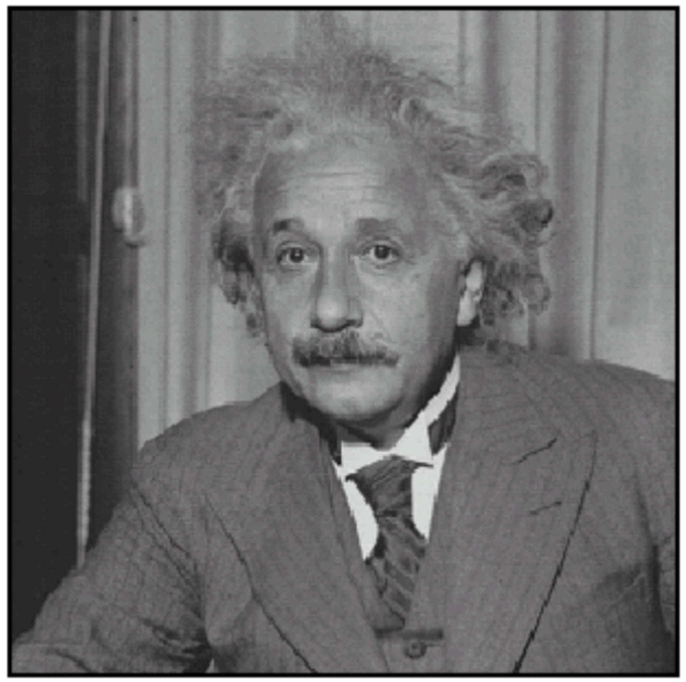

```
for sigma=1:3:10
   h = fspecial('gaussian', fsize, sigma);
   out = imfilter(im, h);
   imshow(out);
   pause;
end
```

[Source: K. Grauman]

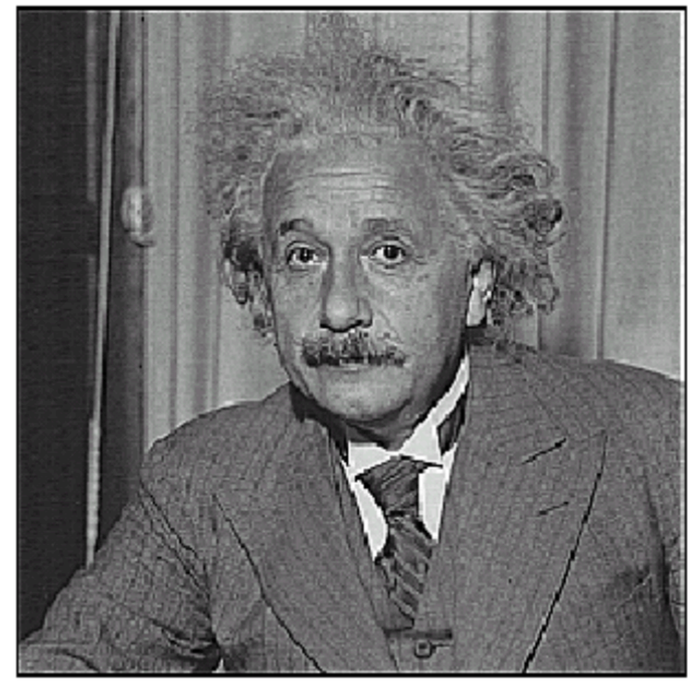
[Slides source: https://www.cs.toronto.edu/~urtasun/courses/CV/lecture02.pdf]

UNIVERSITY OF ALBERTA

Higher-Pass Filter



before



after

[Source: D. Lowe]

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)$$

[Slides source: https://www.cs.toronto.edu/~urtasun/courses/CV/lecture02.pdf]

UNIVERSITY OF ALBERTA

Signal Processing vs. NN

- Convolution in signal processing
  - Flipping (given by def’n), Frobenius inner-product, Sliding

$\langle\ ,\ \rangle_F$

with non-linear f,
kernel is learnable

unnecessary

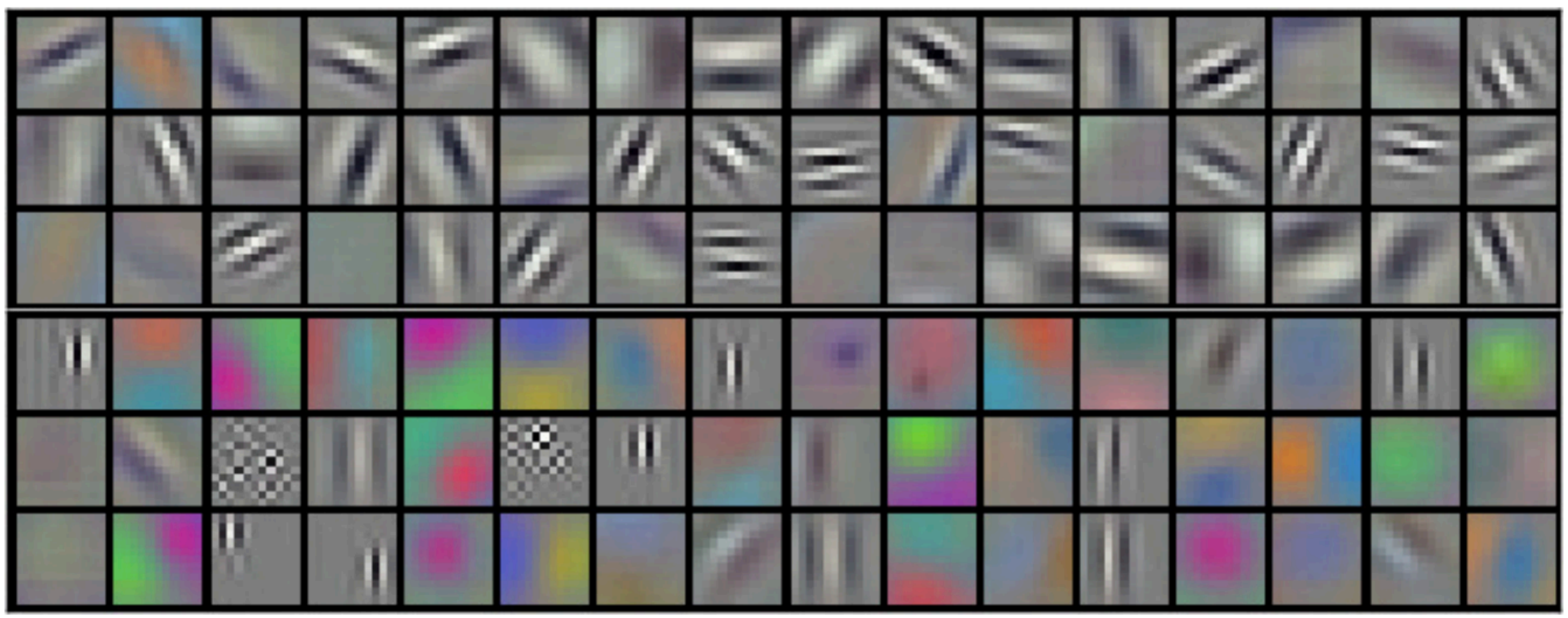
the same

- Convolution in neural networks

Digression: A Few Tips of Learning Math

- Math is important
- No mathematician knows every theorem
  - Lifetime of a mathematician is finite
  - The number of theorems is obviously infinite
- A way to better learn math
  - Get high-level pictures
  - Derive every equation
  - Remember results (and derivations)

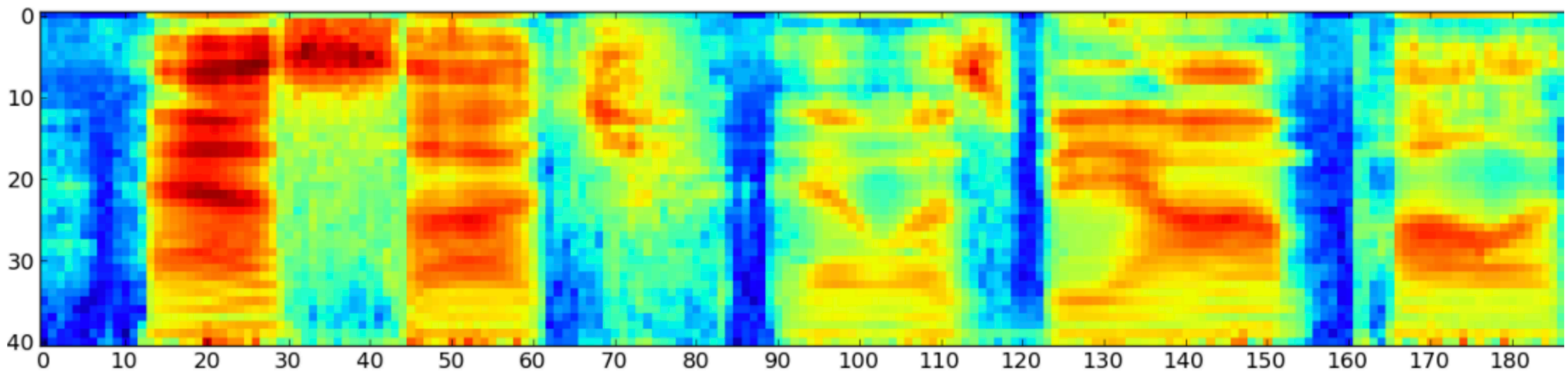
Learned CNN Features



Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

CNN for Speech Processing

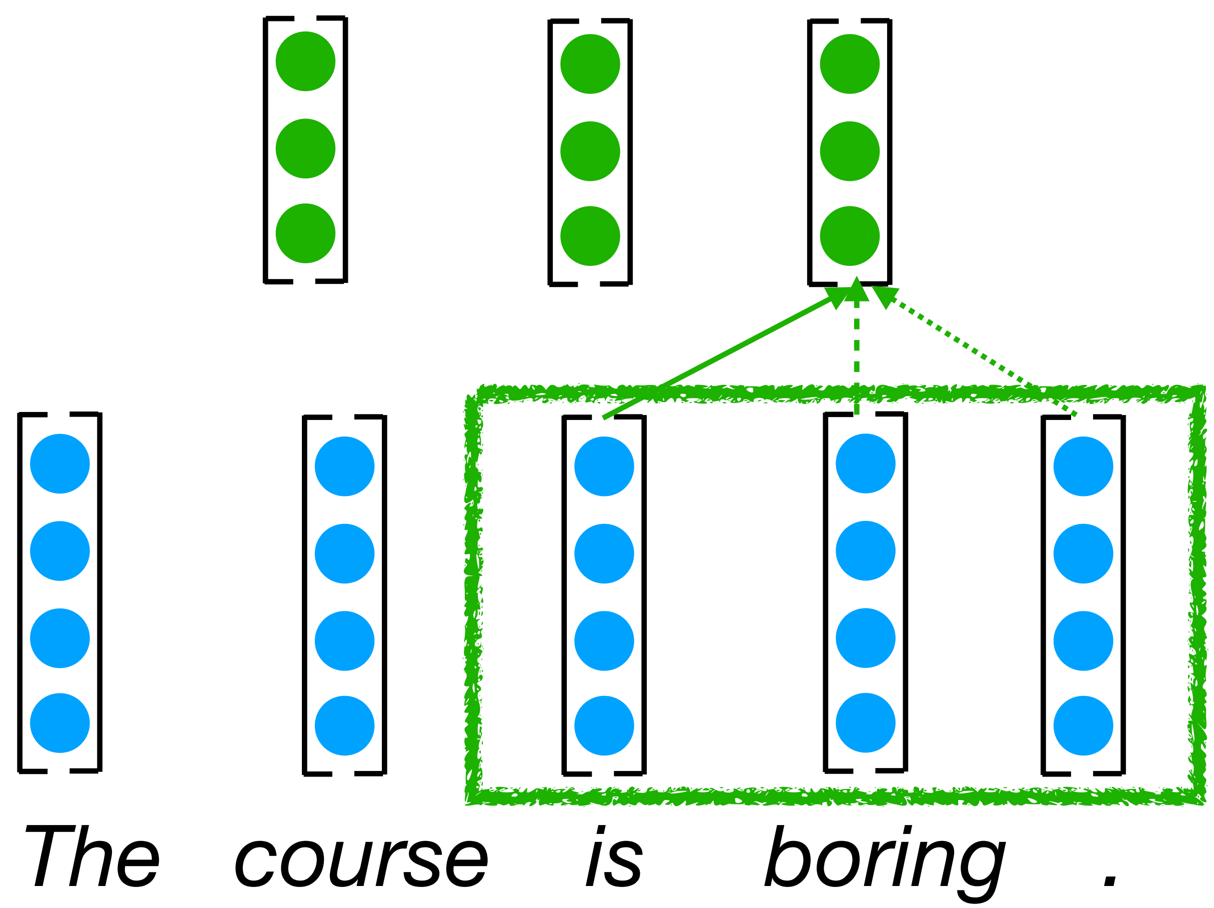
- Convolutional window in the frequency domain



[Graves+, ICASSP'13]

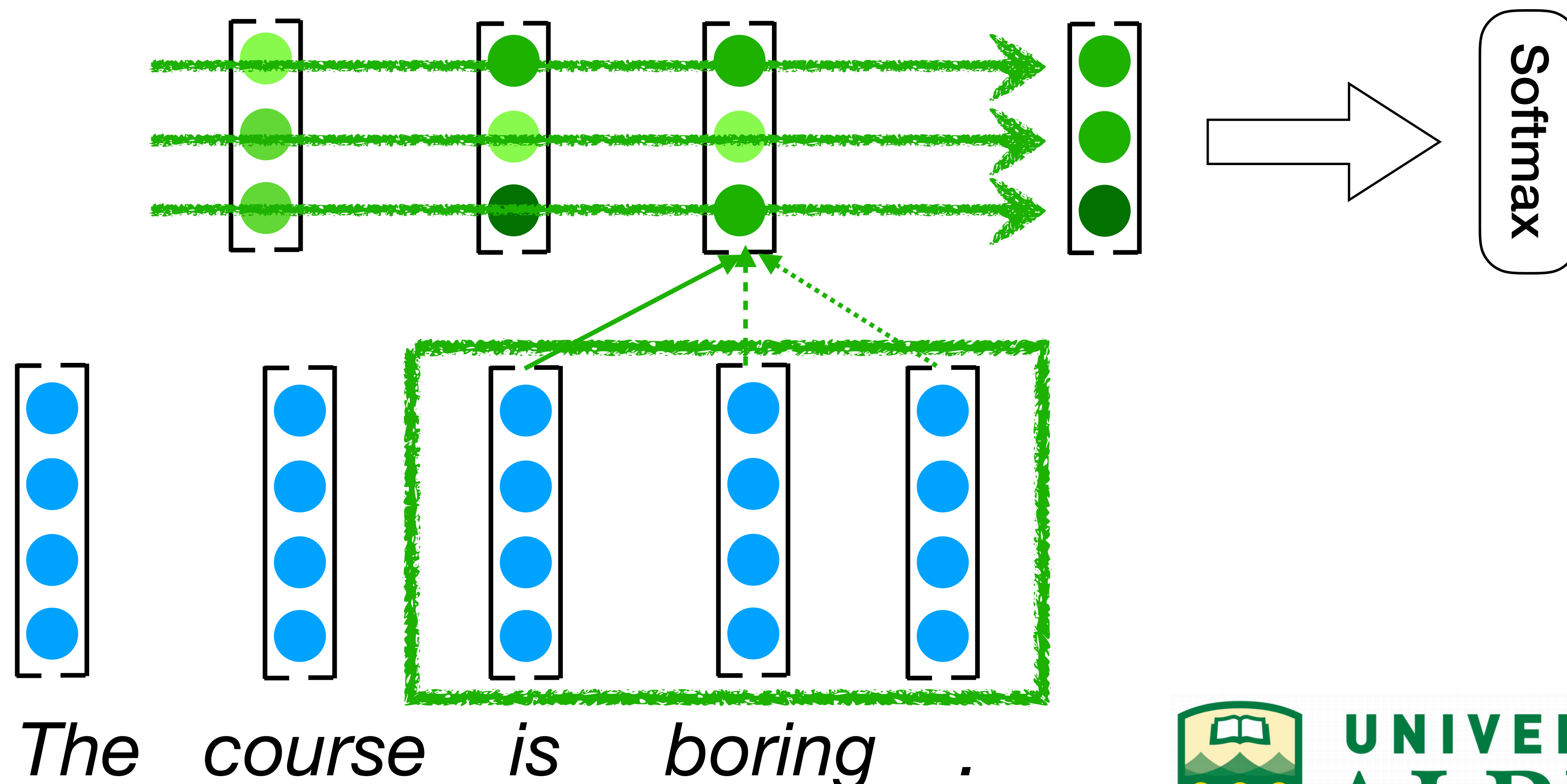
CNN for Text Processing

- Next problem:
  - Length varies among sentences

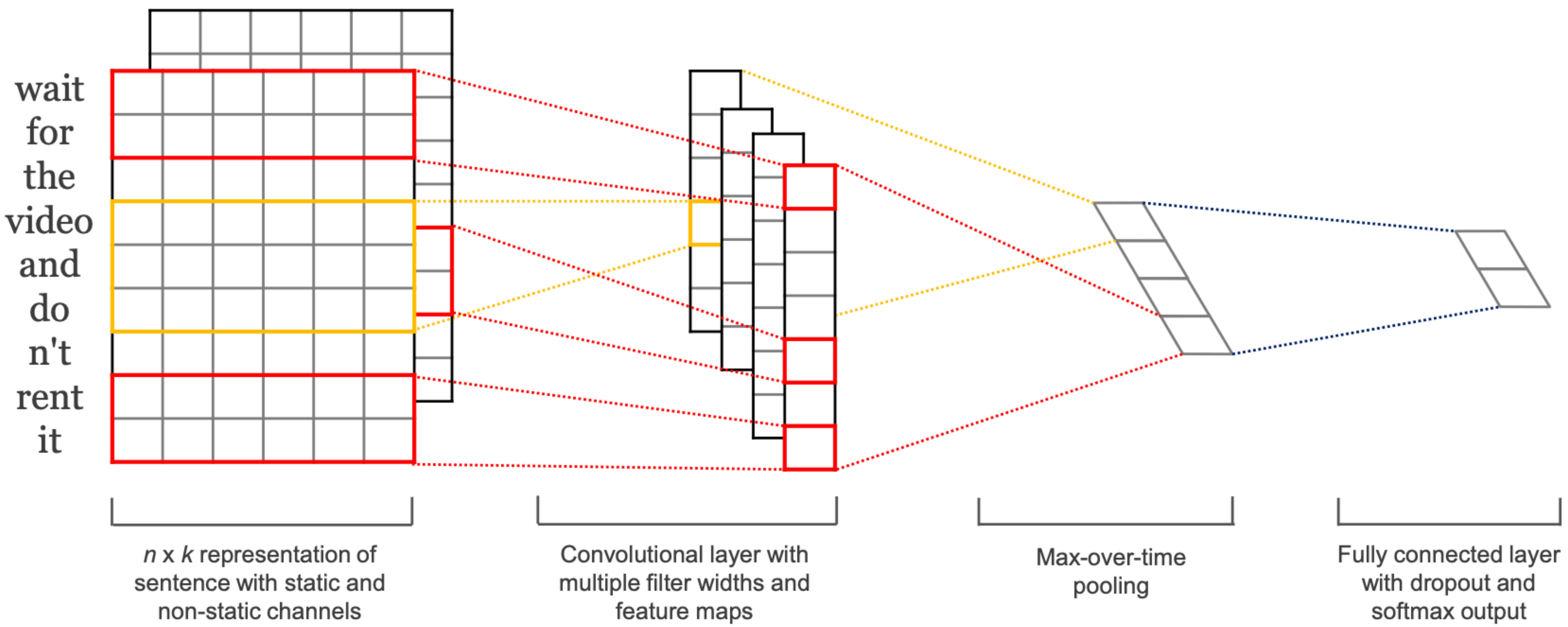


Pooling

- Max pooling: Takes the maximum value in each dimension
  - Intuition: To what extent is a feature satisfied most?
- Sum pooling, average pooling, etc.



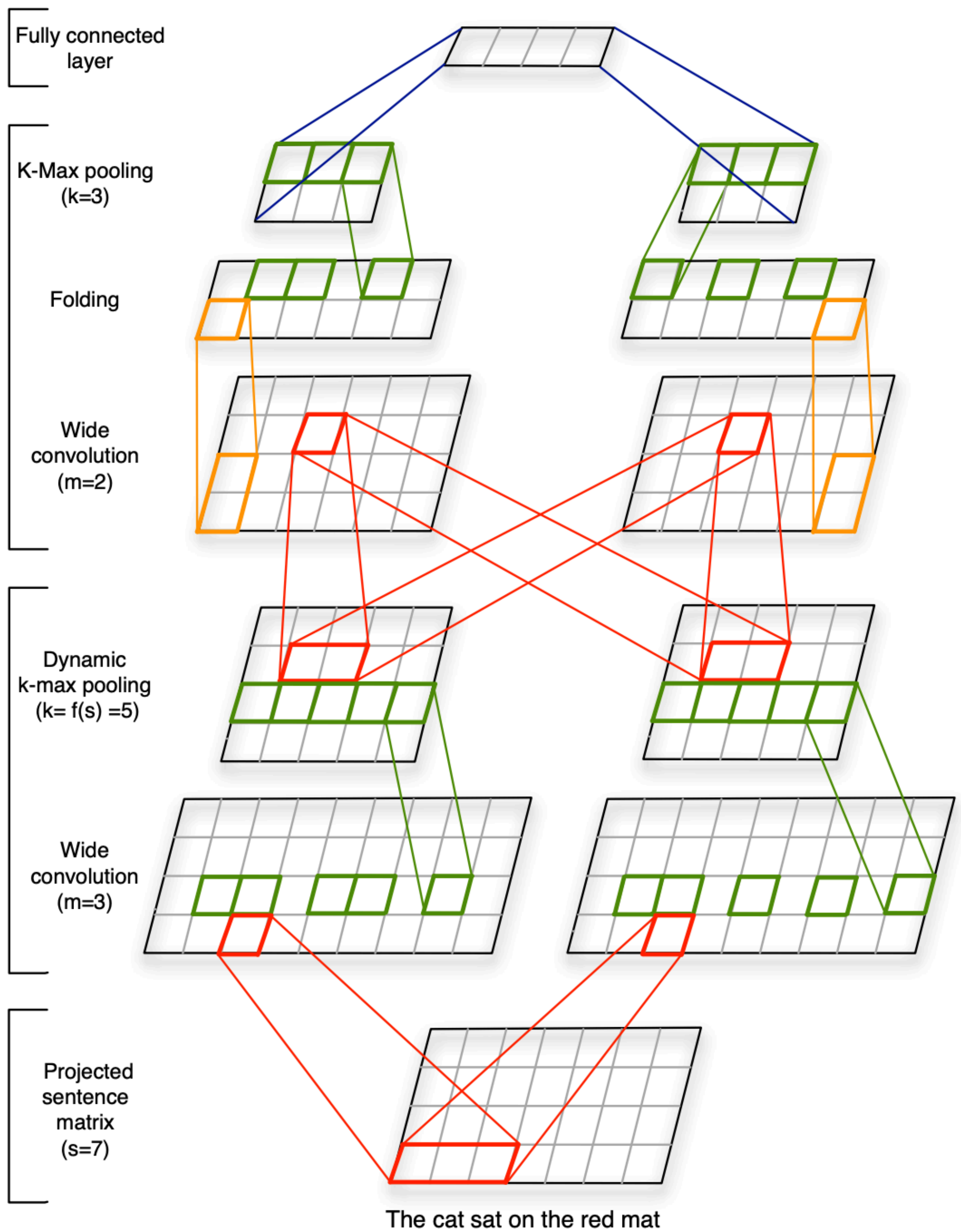
Example



Yoon Kim, Convolutional Neural Networks for Sentence Classification. In *EMNLP* 2014.

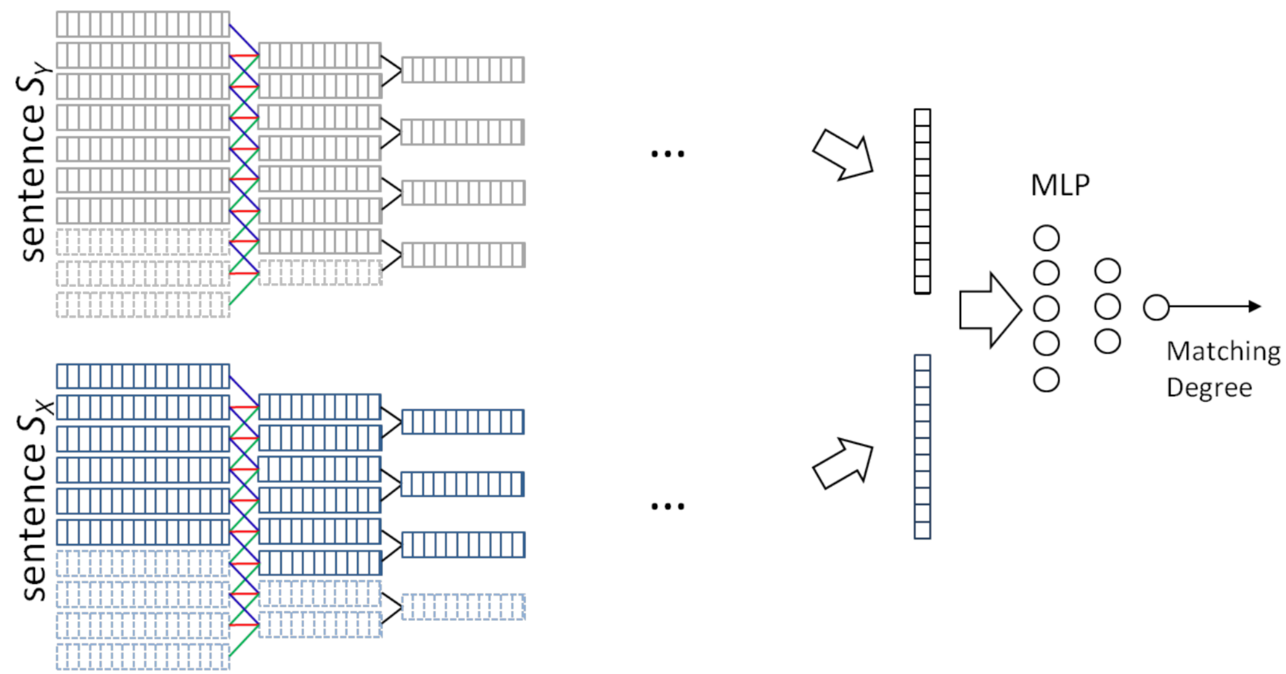
UNIVERSITY OF ALBERTA

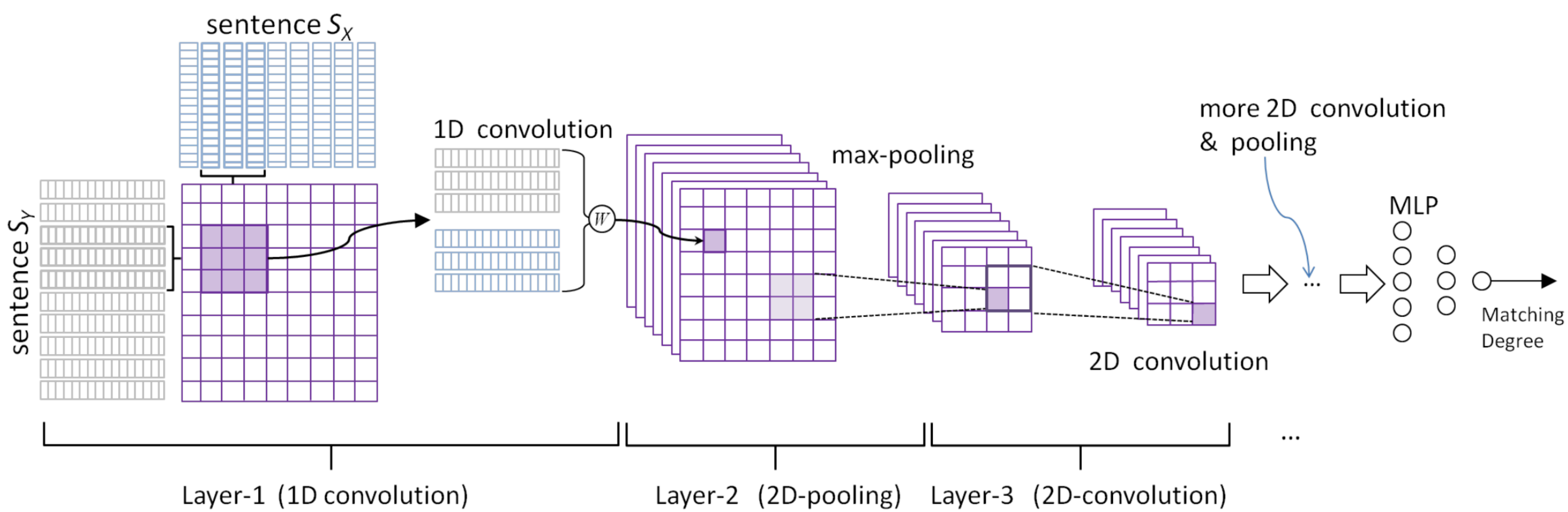
Example



Kalchbrenner et al, A Convolutional Neural Network for Modelling Sentences. In *ACL* 2014.

Example: Sentence Matching





Hu et al., Convolutional neural network architectures for matching natural language sentences. In *NIPS* 2014.

Problems with CNNs

- Direct interaction only within a window
- Too much information loss with max pooling

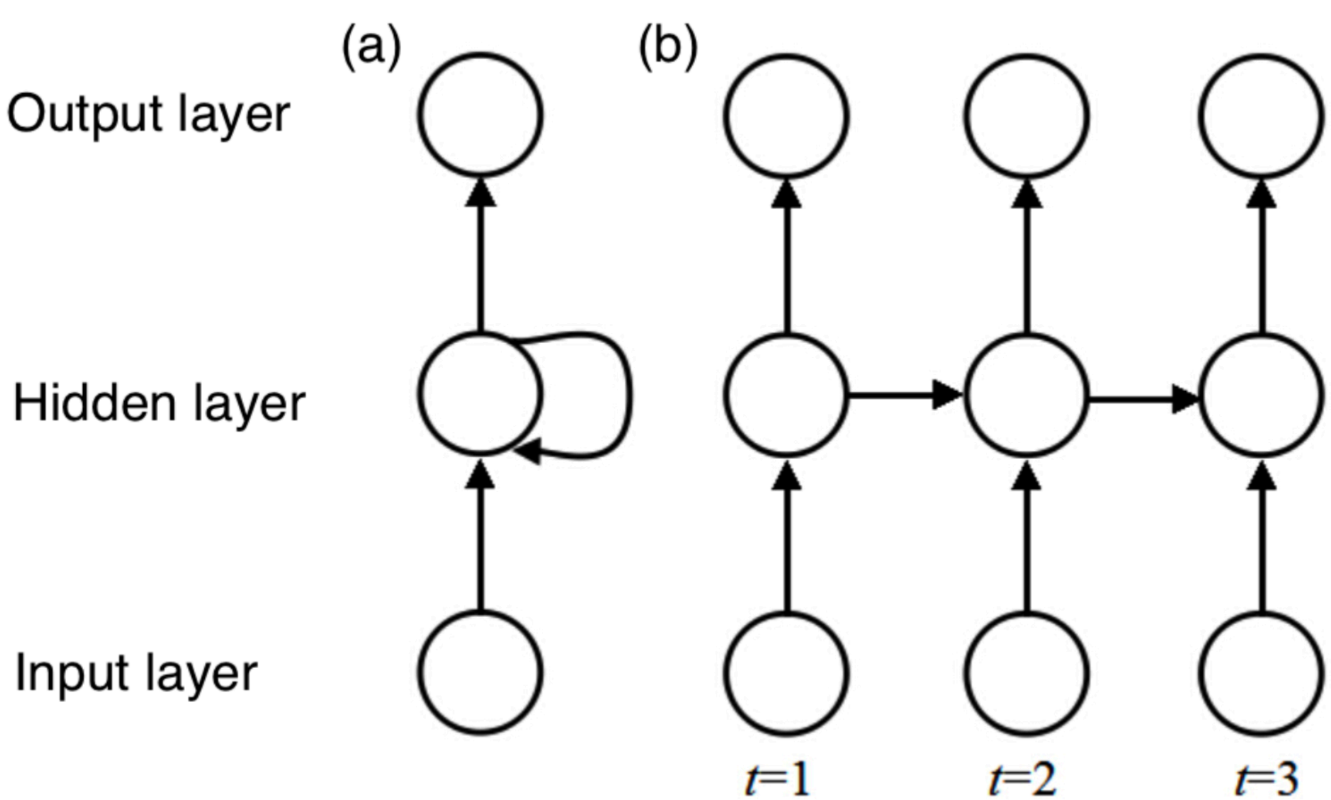
Want: Global dependency modelling for potentially infinitely long sequences

Recurrent Neural Networks

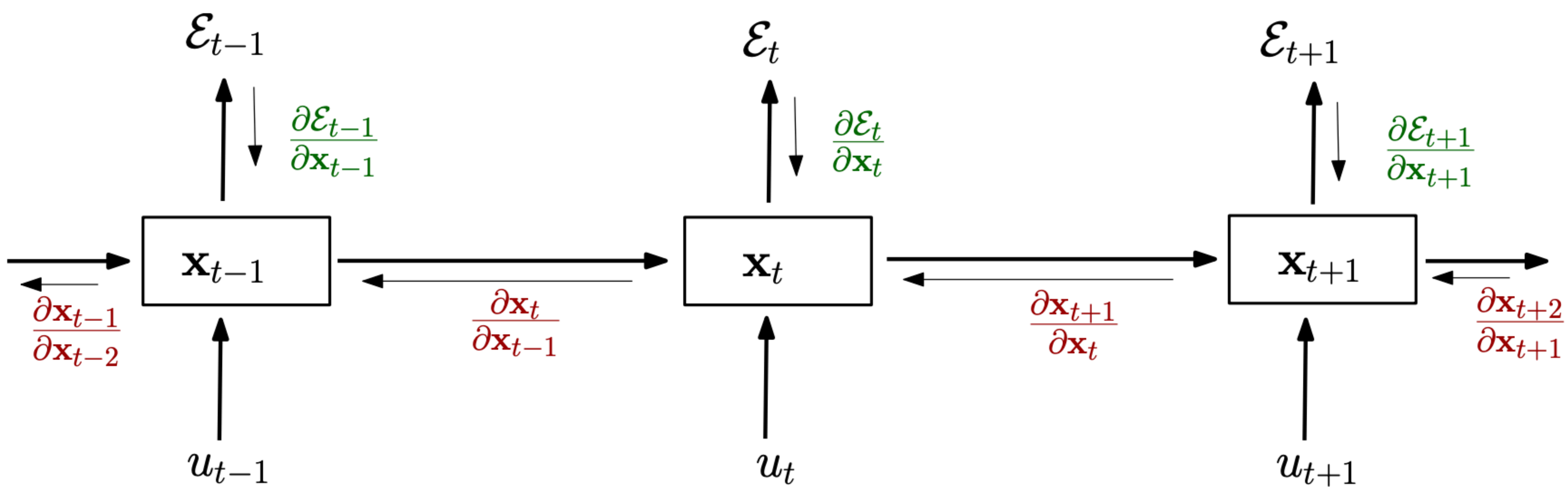
- Vanilla RNN

$$\boldsymbol{h}^{(t)} = f(W_h \boldsymbol{h}^{(t-1)} + W_x \boldsymbol{x}^{(t)} + \boldsymbol{b})$$

- Learning: Backpropagation through time (BPTT)
  - Just a fancy terminology
- Gradient vanishing or explosion
  - BP is a linear system
  - FP is a non-linear system (potentially chaotic)



Gradient Vanishing or Explosion



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta}$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \boxed{\prod_{t \ge i > k} \mathbf{W}_{rec}^T} diag(\sigma'(\mathbf{x}_{i-1}))$$

$$\| \quad \| \le \prod \| \quad \| \; \| \quad \|$$

L_2-norm of a matrix is the max eigenvalue

Pascanu et al., On the difficulty of training recurrent neural networks, *ICML* 2013.

UNIVERSITY OF ALBERTA

What's wrong?

- BP doesn't give exact gradient for RNN? [NO.]
- Numerical underflow/overflow causes gradient vanishing and explosion? [NO.]
- RNNs do not have enough model capacity? [Partially.]
- Exact gradient isn't what we want? [Yes.]

LSTM & GRU

Long short term memory: Keeps a cell $\boldsymbol{c}_t$ and hidden state $\boldsymbol{h}_t$

Input gate $\boldsymbol{i}_t = \sigma(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i)$

Forget gate $\boldsymbol{f}_t = \sigma(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f)$

Output gate $\boldsymbol{o}_t = \sigma(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o)$

Cell content $\boldsymbol{g}_t = \tanh(W_g \boldsymbol{x}_t + U_g \boldsymbol{h}_{t-1} + \boldsymbol{b}_g)$

Cell $\boldsymbol{c}_t = \boldsymbol{i}_t \circ \boldsymbol{g}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1}$

Hidden state $\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t)$

Gated recurrent unit:

Simplied version

$$\boldsymbol{r} = \sigma(W_r \boldsymbol{h}_{t-1})$$

$$\boldsymbol{z} = \sigma(W_z \boldsymbol{h}_{t-1})$$

$$\tilde{\boldsymbol{h}} = \tanh(W_x \boldsymbol{h}_{t-1} + W_g(\boldsymbol{r} \odot \boldsymbol{h}_{t-1}))$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z} \odot \tilde{\boldsymbol{h}}_t$$

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, *9*(8), pp.1735-1780.

UNIVERSITY OF ALBERTA

Intuition of LSTM/GRU

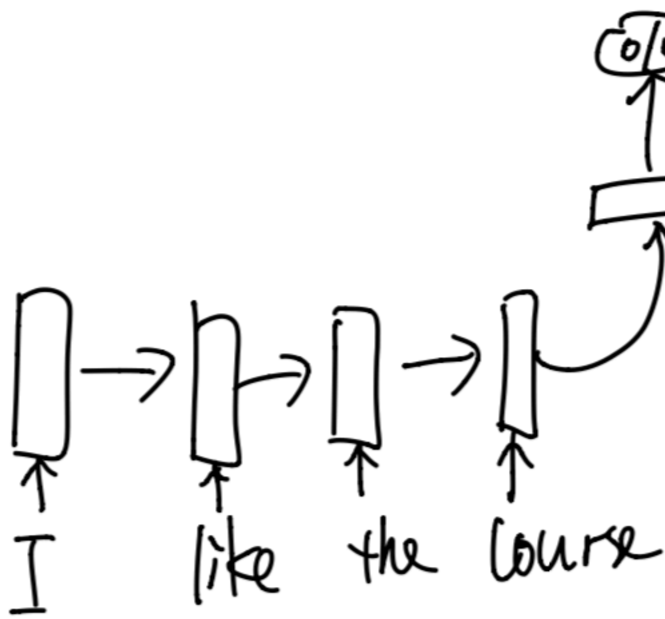
- **No Free-Lunch Theorem**: If someone's intuition can potentially explain everything, then such intuition is not useful.
- If we have too much belated intuition, our intuition is overfitting to experiments.

RNN Usages

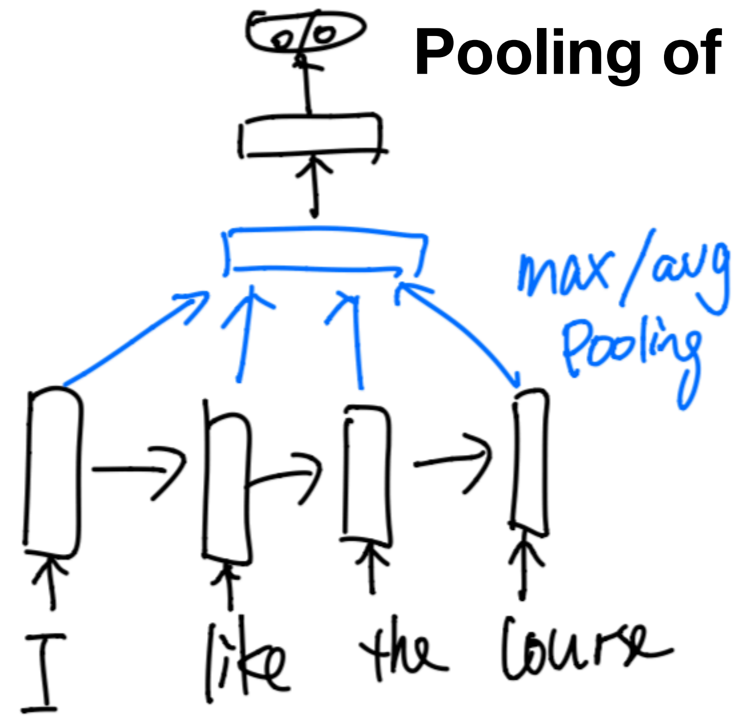
- Single output (e.g., classification)
- Sequential output (one-one corresponding to input tokens)
- Sequential output (no correspondence)

Single Output

Using the last hidden state

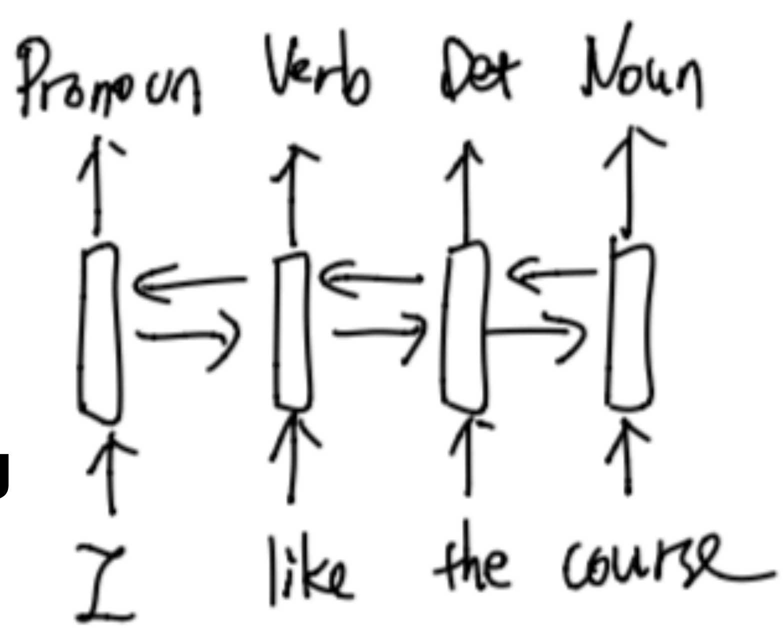


Pooling of all hidden states

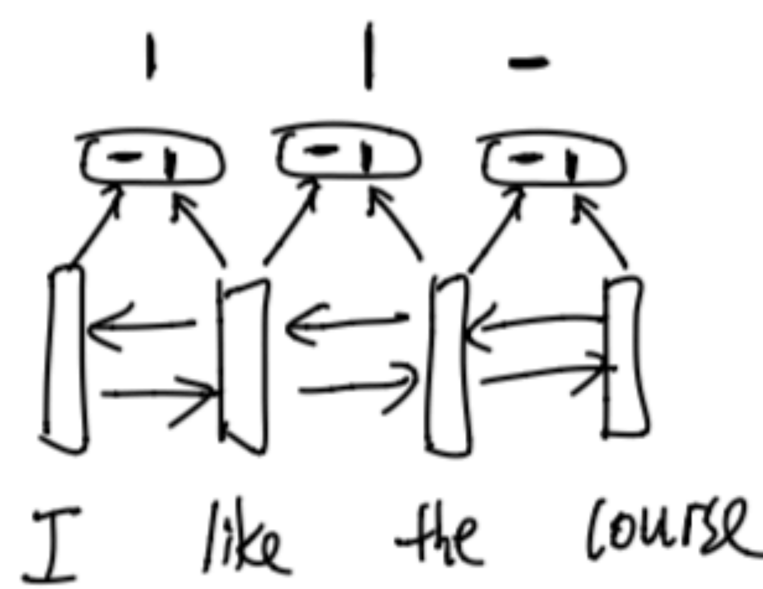


Sequential Output

(one-one corresponding to input tokens)



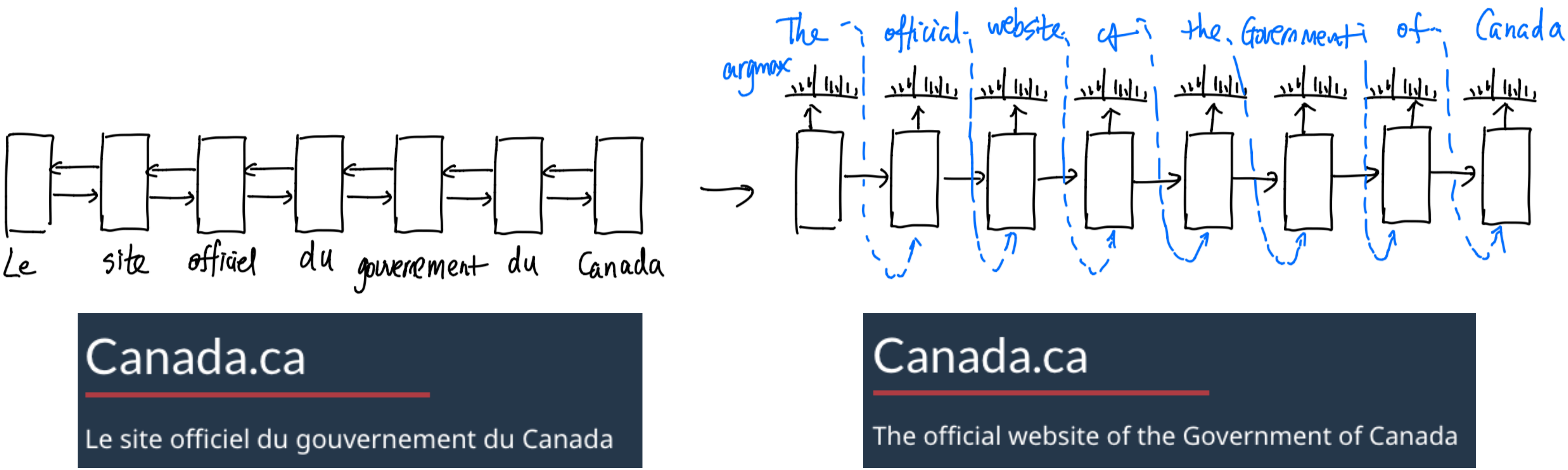
POS tagging



Chunking

Sequential Output

(No correspondence to input tokens)



[Source: canda.ca]

- **Questions:**

  Why do we feed back the generated words?

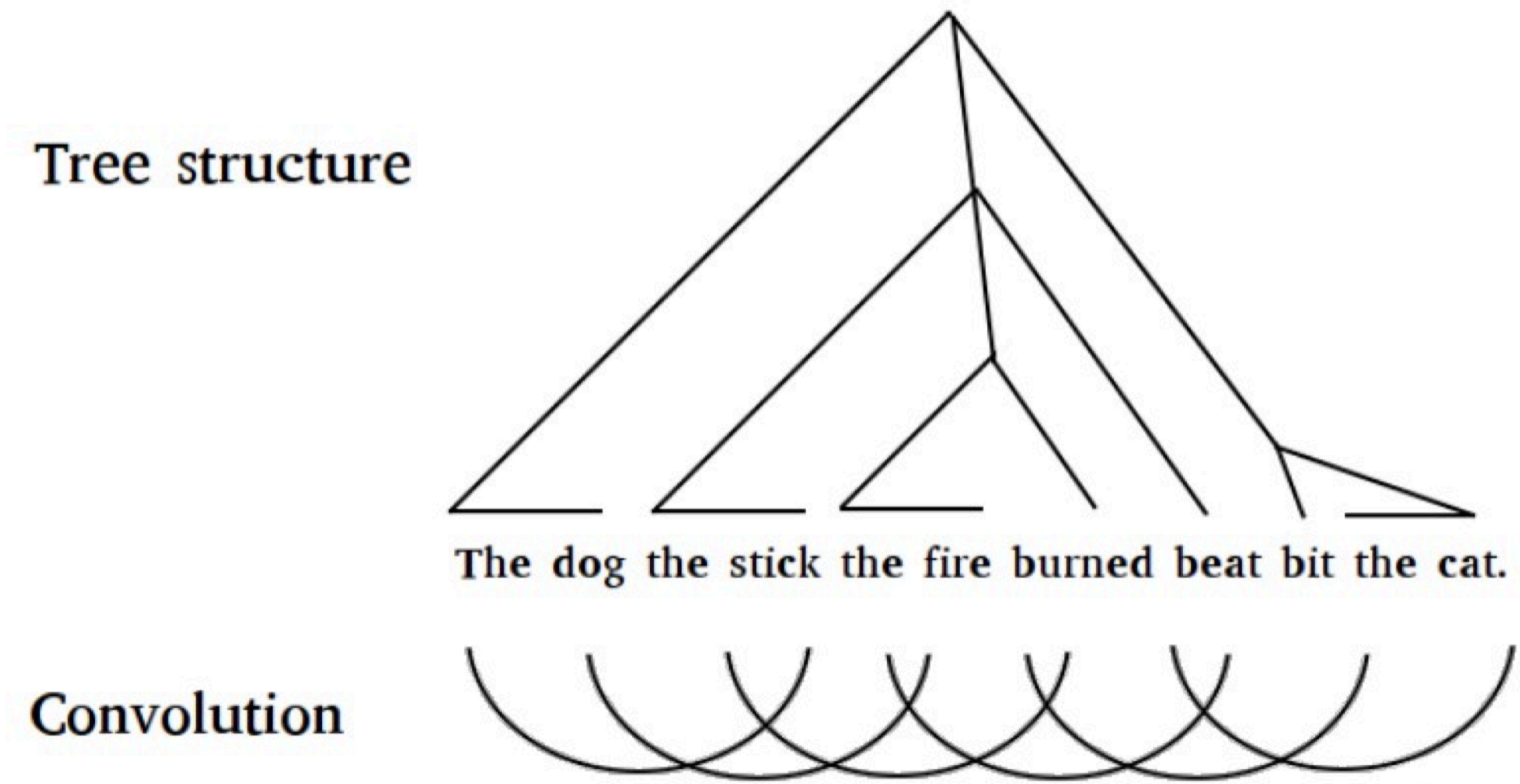
  How can we train Seq2Seq models?

  How do we do inference?

Prior Knowledge

- CNN
  - Spatial neighborhood
  - Sliding window
- RNN
  - Ordered information
  - Sequential processing

Why parse trees may be important?



Pinker, Steven. The Language Instinct: The New Science of Language and Mind. Penguin UK, 1995.

Recursive Propagation

- Perception-like interaction

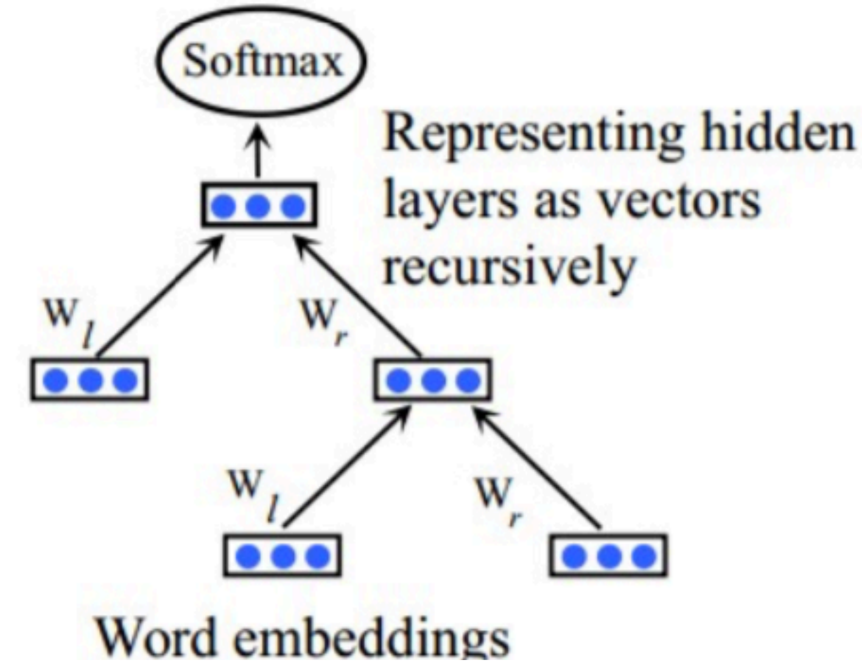
$$p = f(W[c_1\ c_2]^T)$$

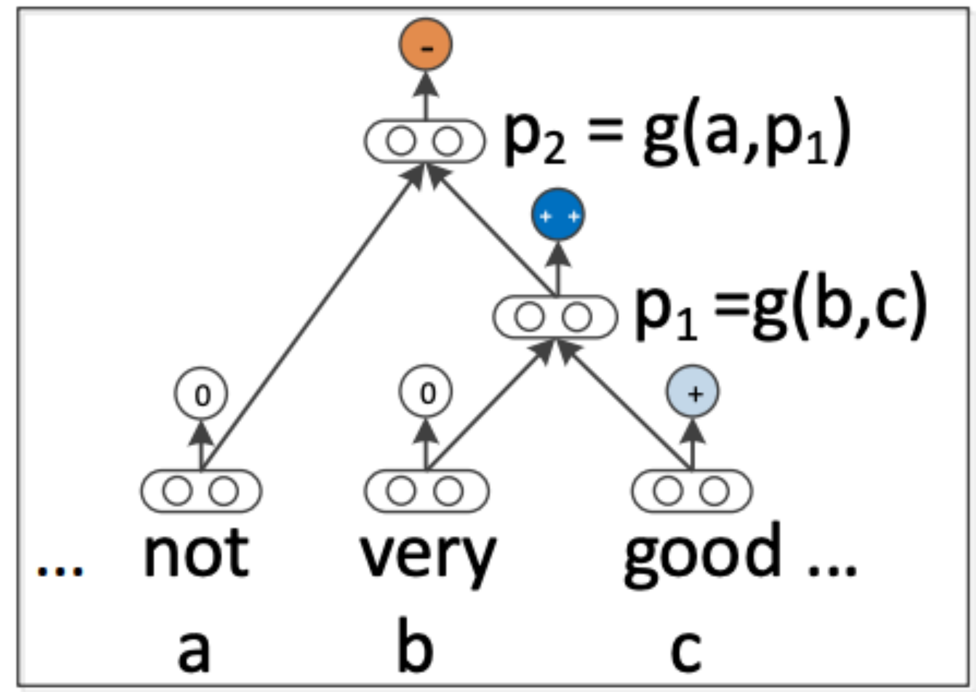
- Matrix-vector interaction

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right)$$

- Tensor interaction

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]}\begin{bmatrix} b \\ c \end{bmatrix} + W\begin{bmatrix} b \\ c \end{bmatrix}\right)$$





Socher R, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. EMNLP, 2011

Socher, R, et al. "Semantic compositionality through recursive matrix-vector spaces." EMNLP-CoNLL, 2012.

Socher, R, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP, 2013.

UNIVERSITY OF ALBERTA

Even More Interaction

- LSTM interaction

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$
$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right),$$
$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right),$$
$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right),$$
$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right),$$
$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$
$$h_j = o_j \odot \tanh(c_j),$$

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." ACL, 2015

Zhu, Xiaodan, Parinaz Sobihani, and Hongyu Guo. "Long short-term memory over recursive structures." ICML, 2015.

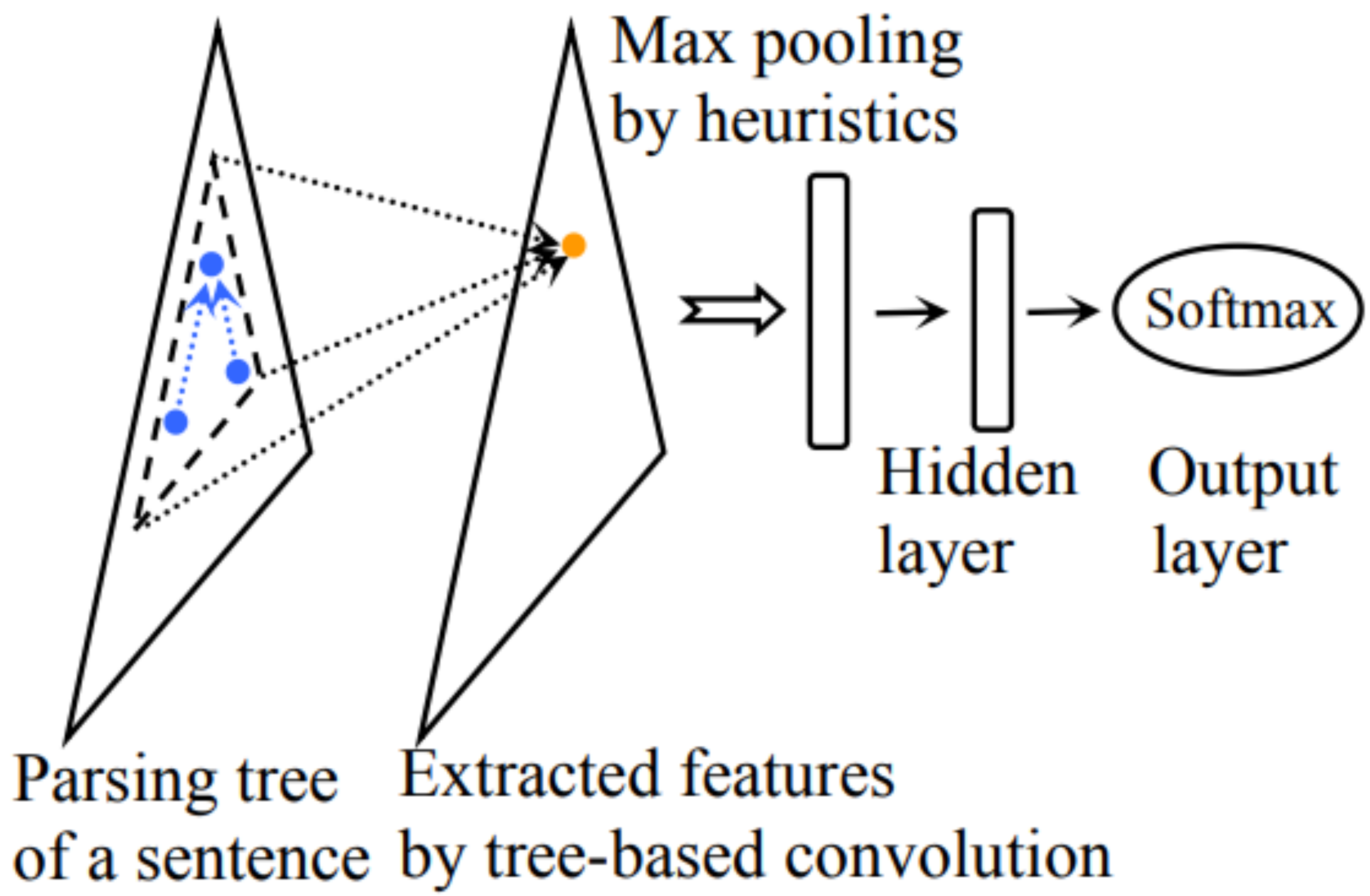
Le, Phong, and Willem Zuidema. "Compositional distributional semantics with long short term memory." arXiv:1503.02510 (2015).

Structure

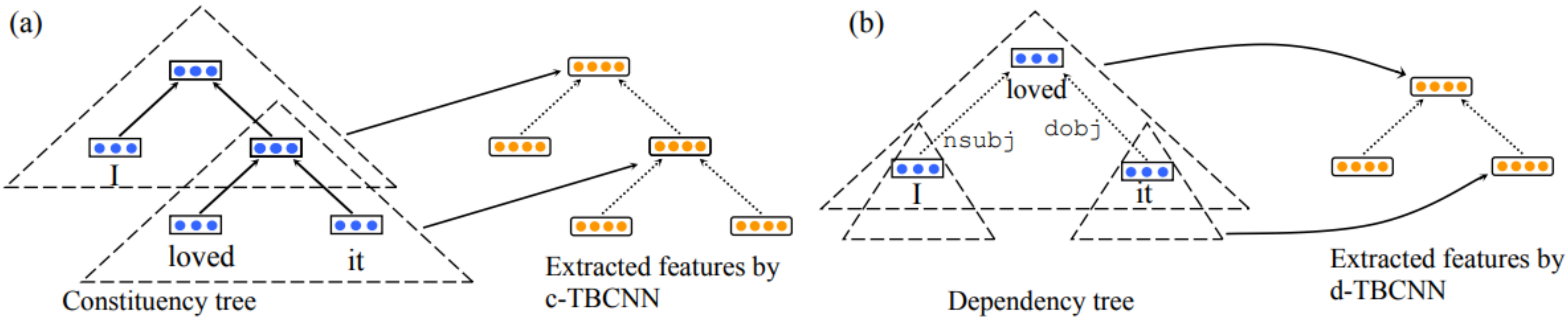
Information aggregation

	Sequence	**Tree**
Sliding window	CNN	RNN
Interative	?	TreeRNN

Tree-Based Convolution



Tree-Based Convolution

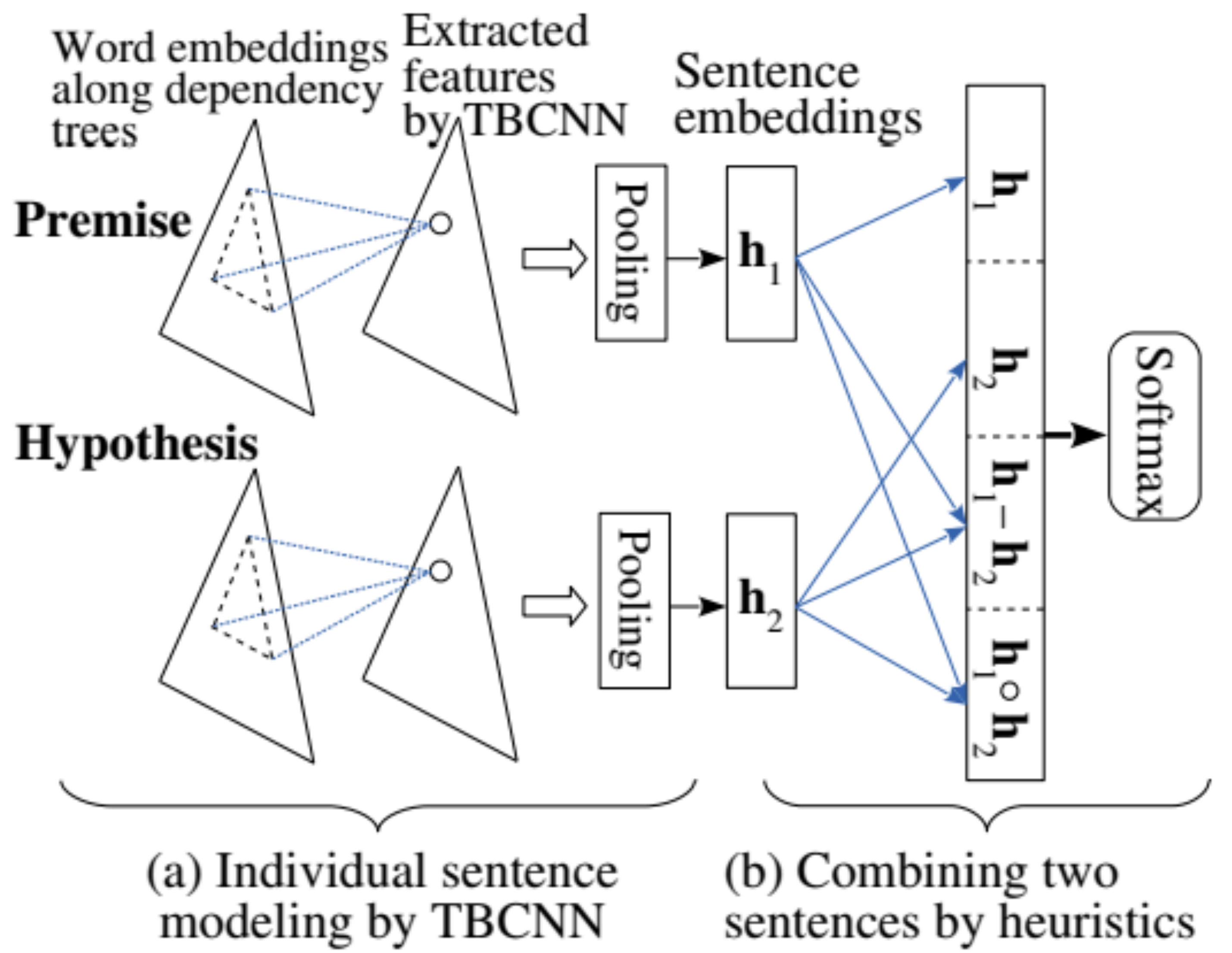


Constituency tree **Dependency tree**

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, Zhi Jin. Discriminative neural sentence modeling by tree-based convolution. In *EMNLP*, 2015.

Tree-Based Convolution

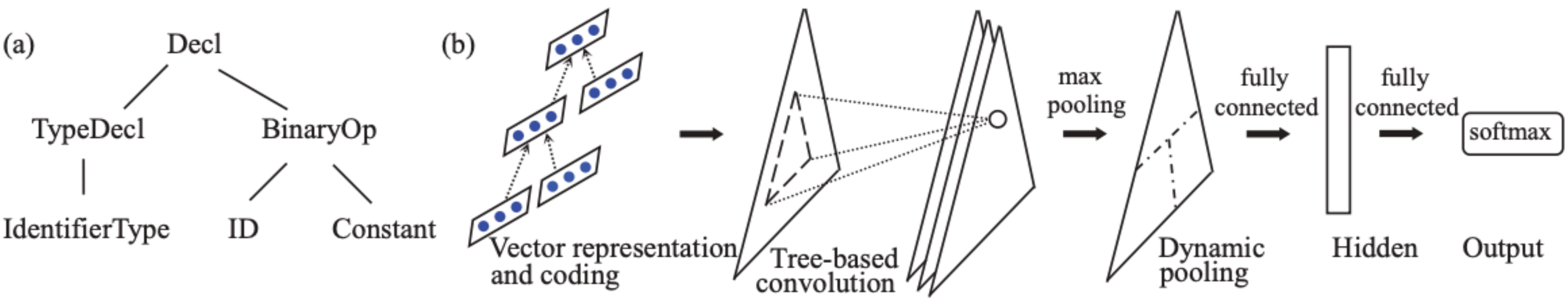
Sentence Matching



Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *ACL*, 2016.

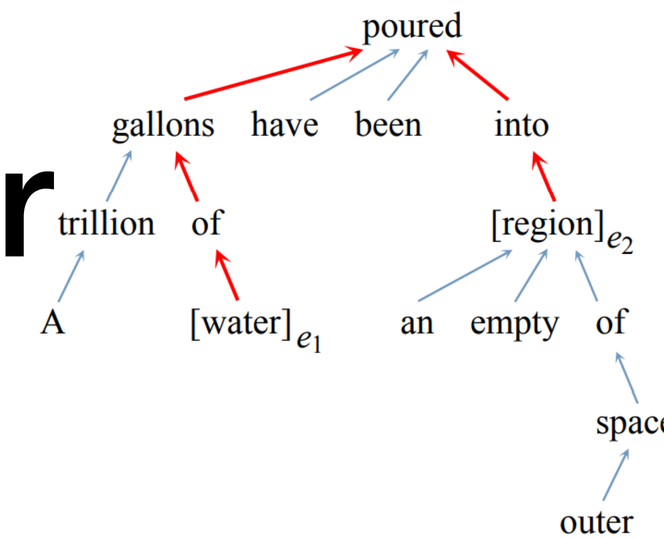
Tree-Based Convolution

Programming language: Abstract Syntax Tree



Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin. Convolutional neural networks over tree structures for programming language processing. In *AAAI*, 2016.

Task-Specific Prior

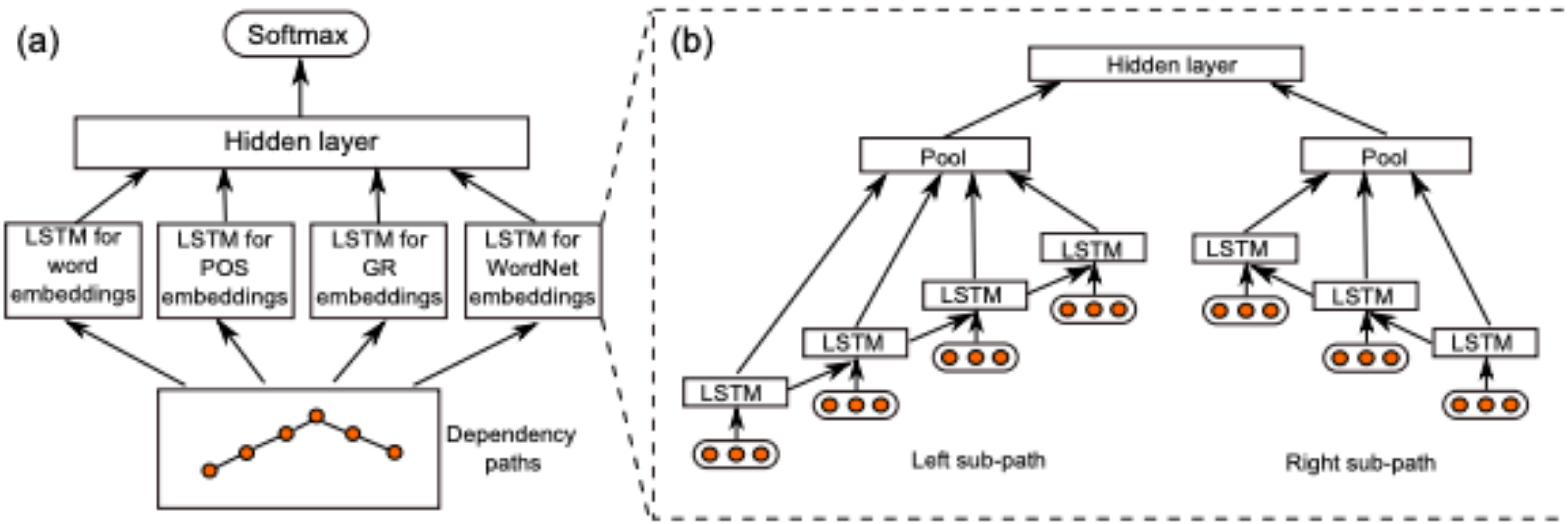


Relation Classification

Input:

A trillion gallons of **water** have been poured into an empty region of outer **space**

Target: Entity_Destination



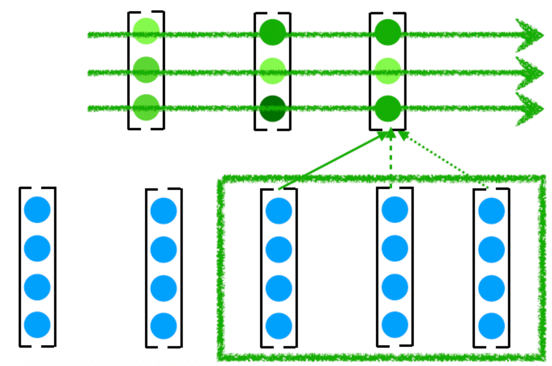
Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*, 2015.

CMPUT 651 (Fall 2019)

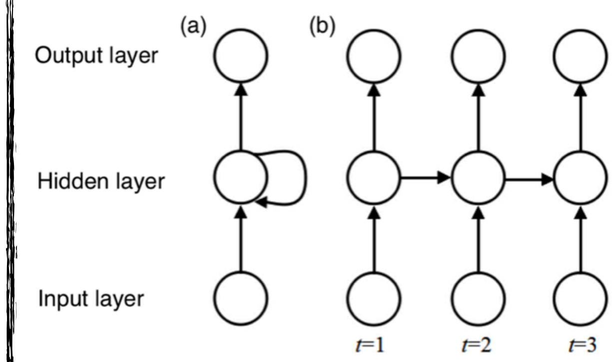
Mindmap

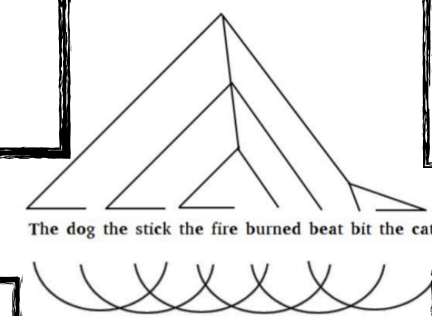
Generic DNN

CNN

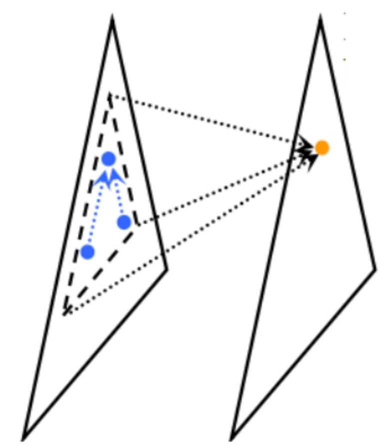


RNN

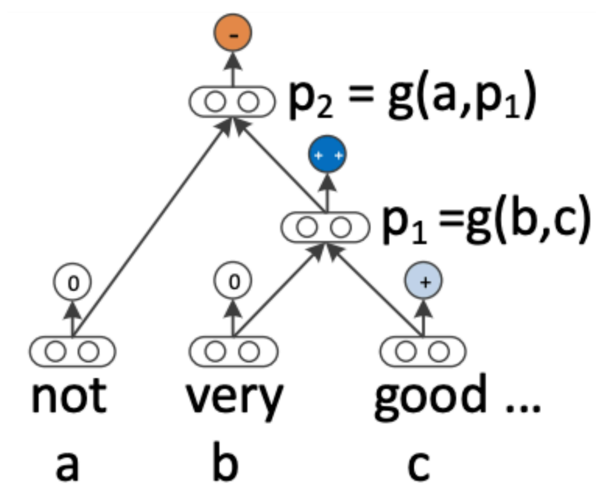




TBCNN



TreeRNN

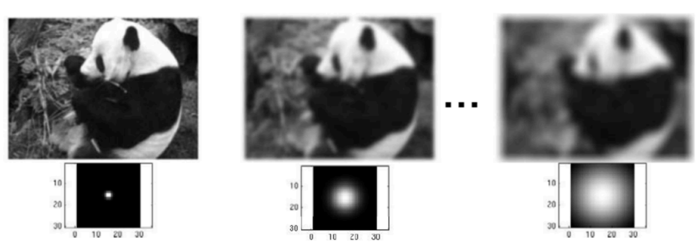


Task-specific architectures

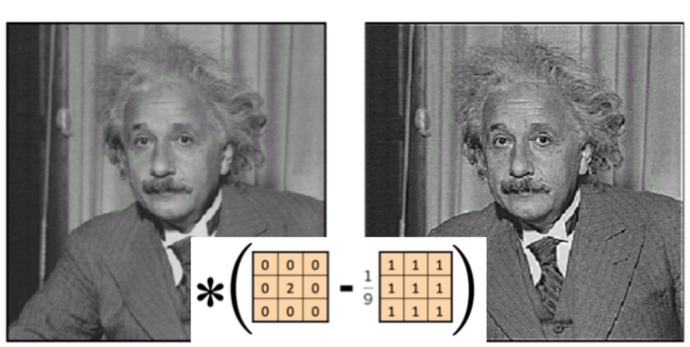
Convolution in signal processing

$\langle \, , \, \rangle_F$

Low-pass filter



High-pass filter



Suggested Reading

- A textbook (or course) on Digital Signal Processing
- Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, *9*(8), pp.1735-1780.
- Pascanu et al., 2013. On the difficulty of training recurrent neural networks. In *ICML*.

More References

- Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.
- Yoon Kim, Convolutional Neural Networks for Sentence Classification. In *EMNLP* 2014.
- Kalchbrenner et al, A Convolutional Neural Network for Modelling Sentences. In *ACL* 2014.
- Hu et al., Convolutional neural network architectures for matching natural language sentences. In *NIPS* 2014.
- Sutskever, I., Vinyals, O. and Le, Q.V. Sequence to sequence learning with neural networks. In *NIPS* 2014.
- Socher, R., Pennington, J., Huang, E.H., Ng, A.Y. and Manning, C.D., July. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP* 2011.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A. and Potts, C., October. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.
- Zhu, X., Sobihani, P. and Guo, H, June. Long short-term memory over recursive structures. In *ICML*, 2015.
- Le, P. and Zuidema, W., 2015. Compositional distributional semantics with long short term memory. arXiv preprint arXiv:1503.02510.
- Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, Zhi Jin. Discriminative neural sentence modeling by tree-based convolution. In *EMNLP*, 2015.
- Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin. Convolutional neural networks over tree structures for programming language processing. In *AAAI*, 2016.
- Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *ACL*, 2016.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In EMNLP, 2015.