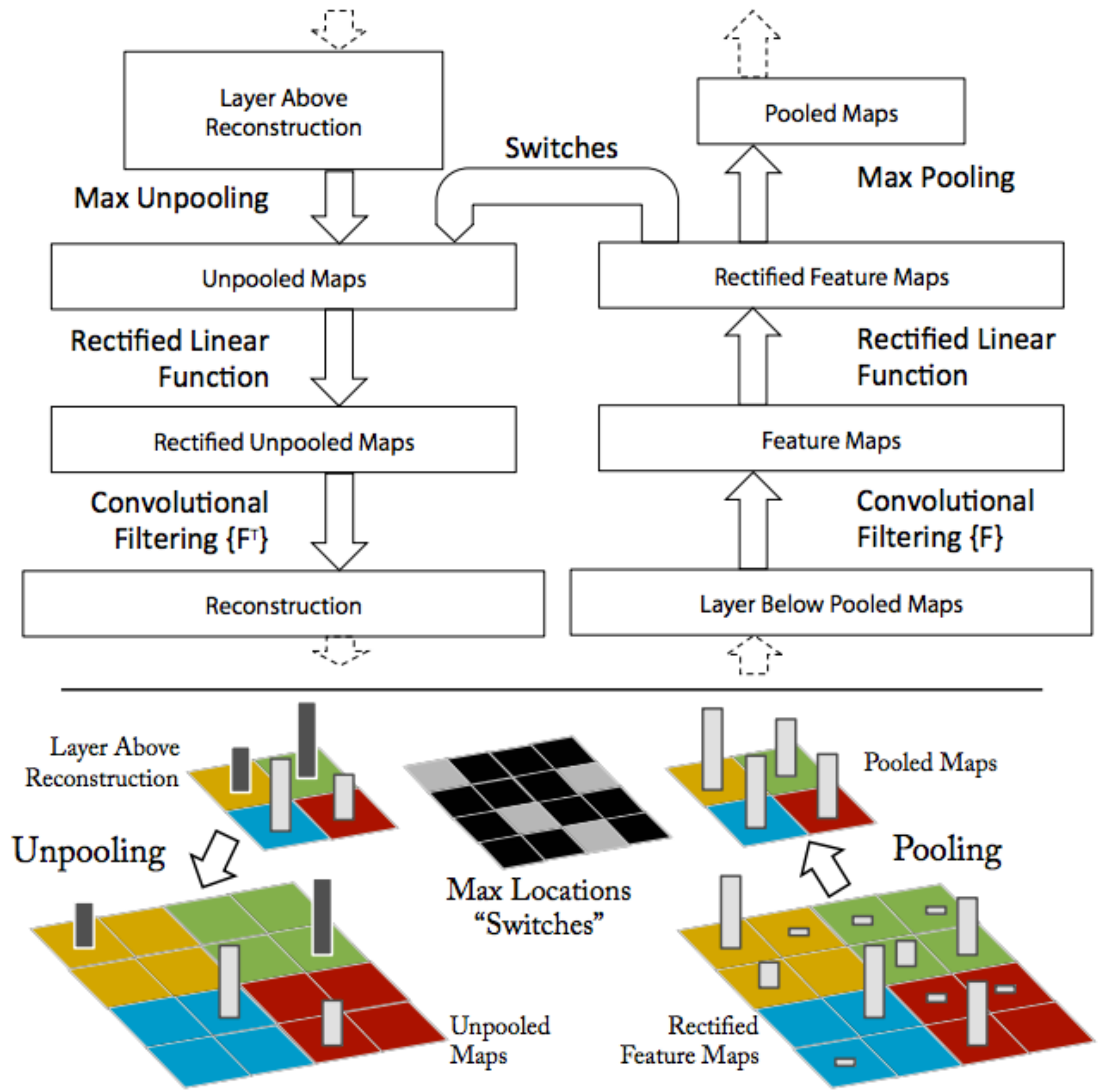
ELEC/COMP 576: Understanding and Visualizing Convnets & Introduction to Recurrent Neural Networks

Ankit B. Patel
Baylor College of Medicine (Neuroscience Dept.)
Rice University (ECE Dept.)

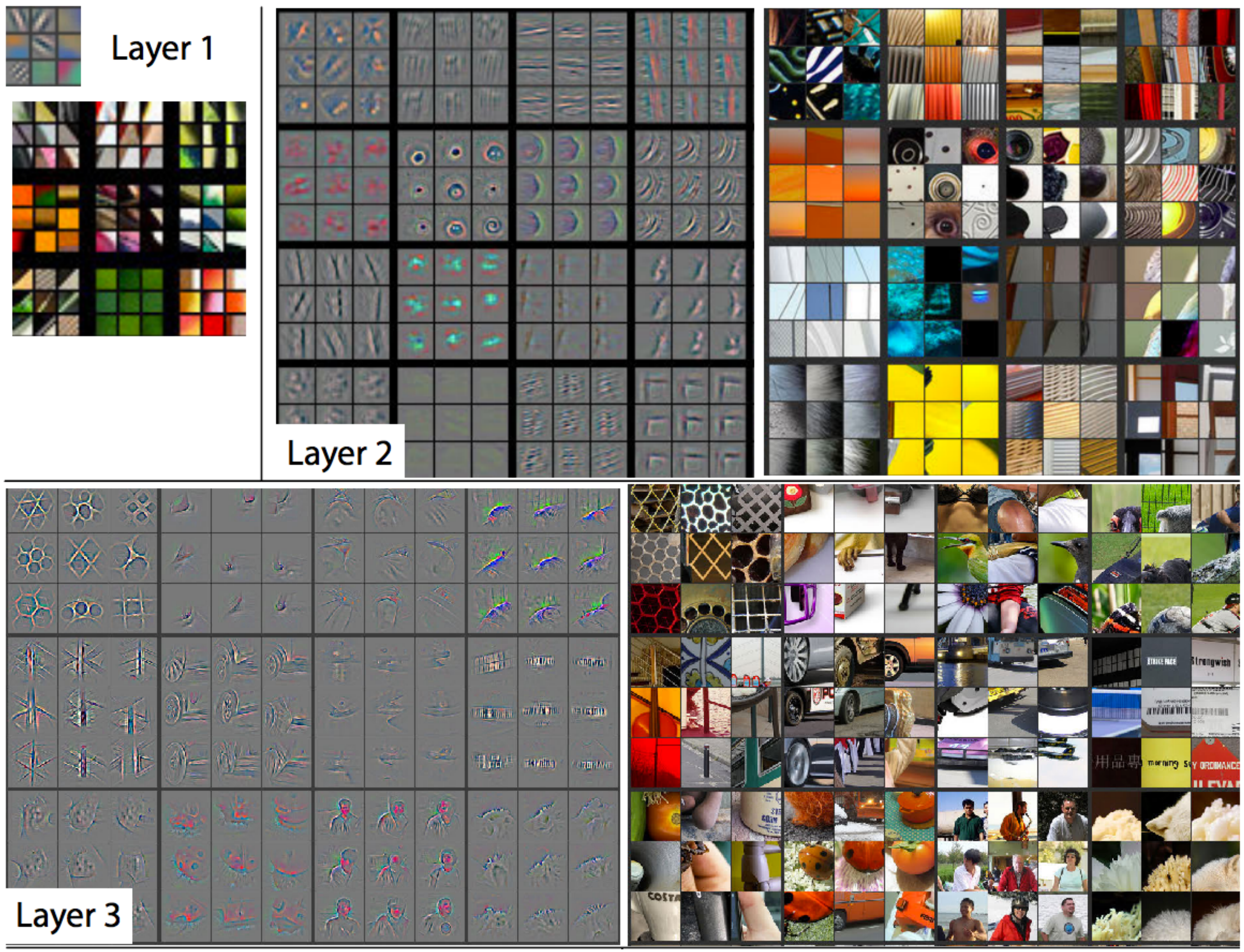
Understand & Visualizing Convnets

Deconvolutional Net



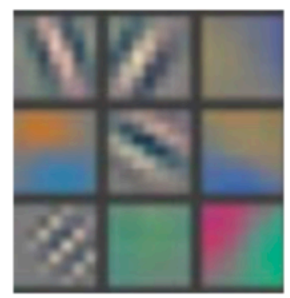
[Zeiler and Fergus]

Feature Visualization



[Zeiler and Fergus]

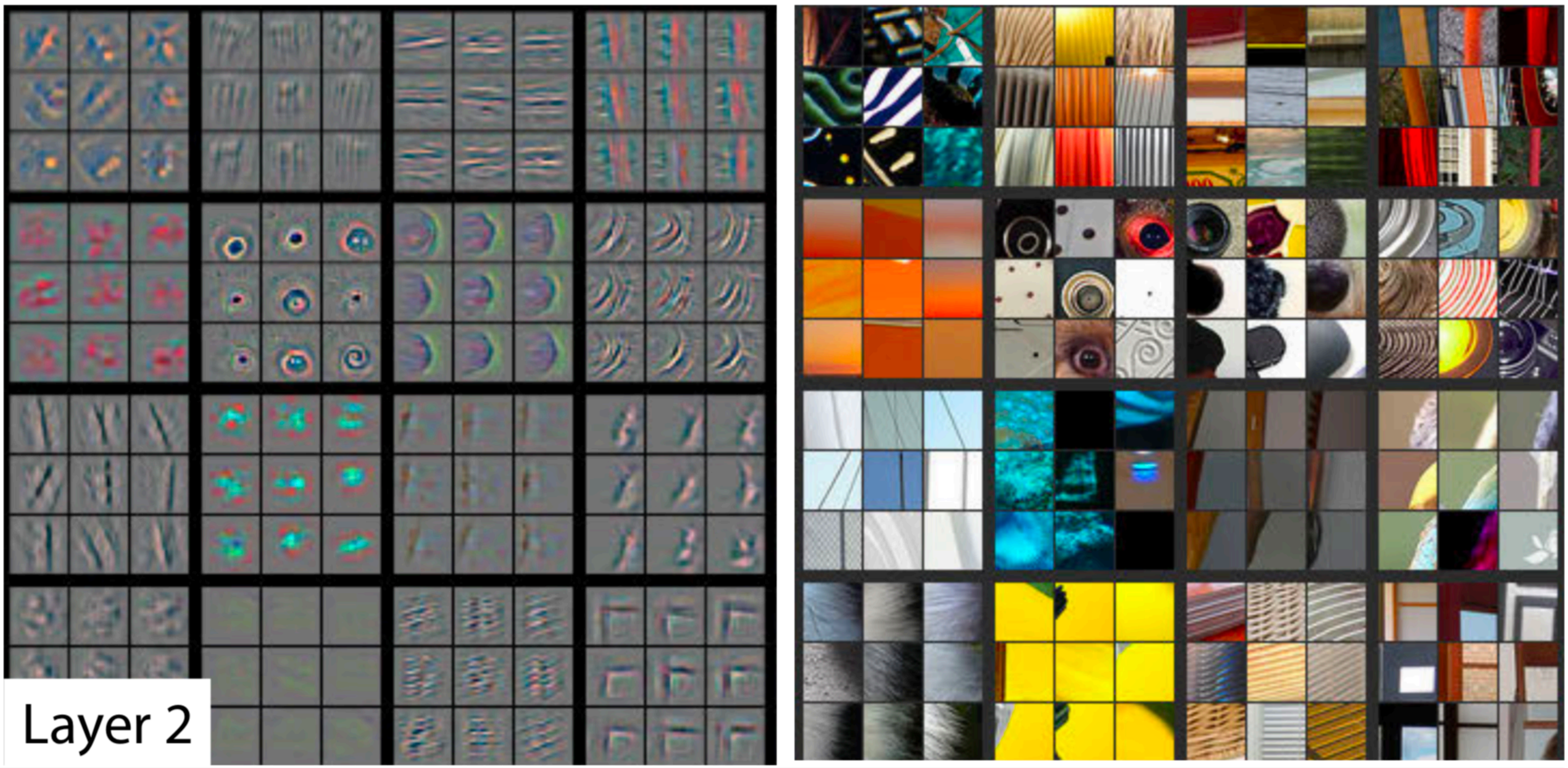
Feature Visualization



Layer 1

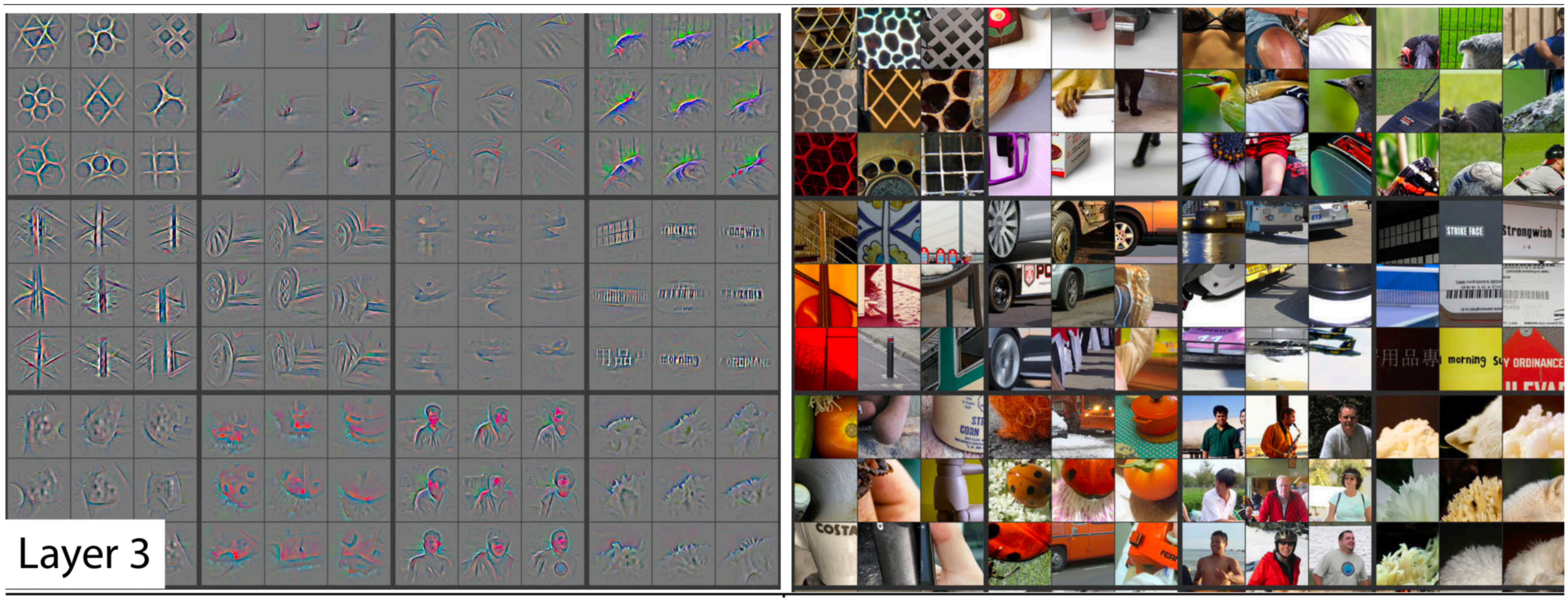
[Zeiler and Fergus]

Feature Visualization



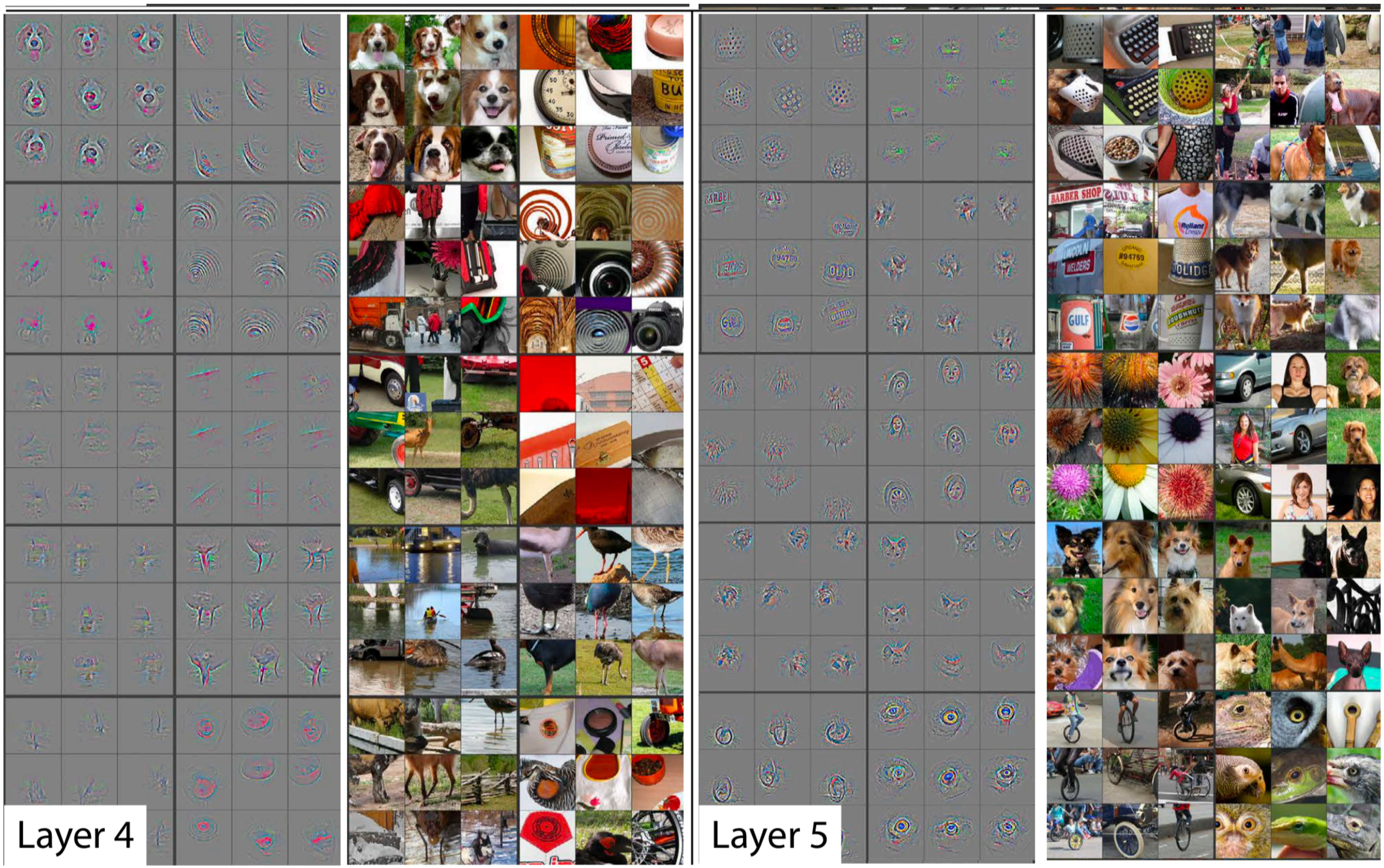
[Zeiler and Fergus]

Feature Visualization



[Zeiler and Fergus]

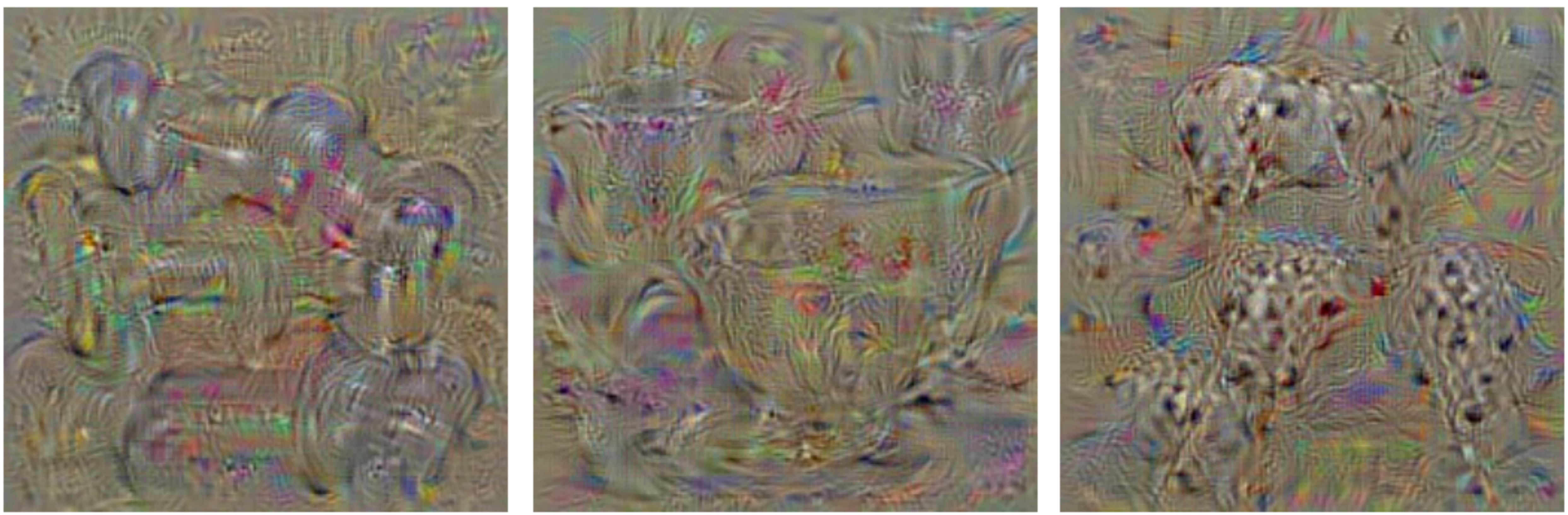
Feature Visualization



[Zeiler and Fergus]

Activity Maximization (aka Saliency Maps)

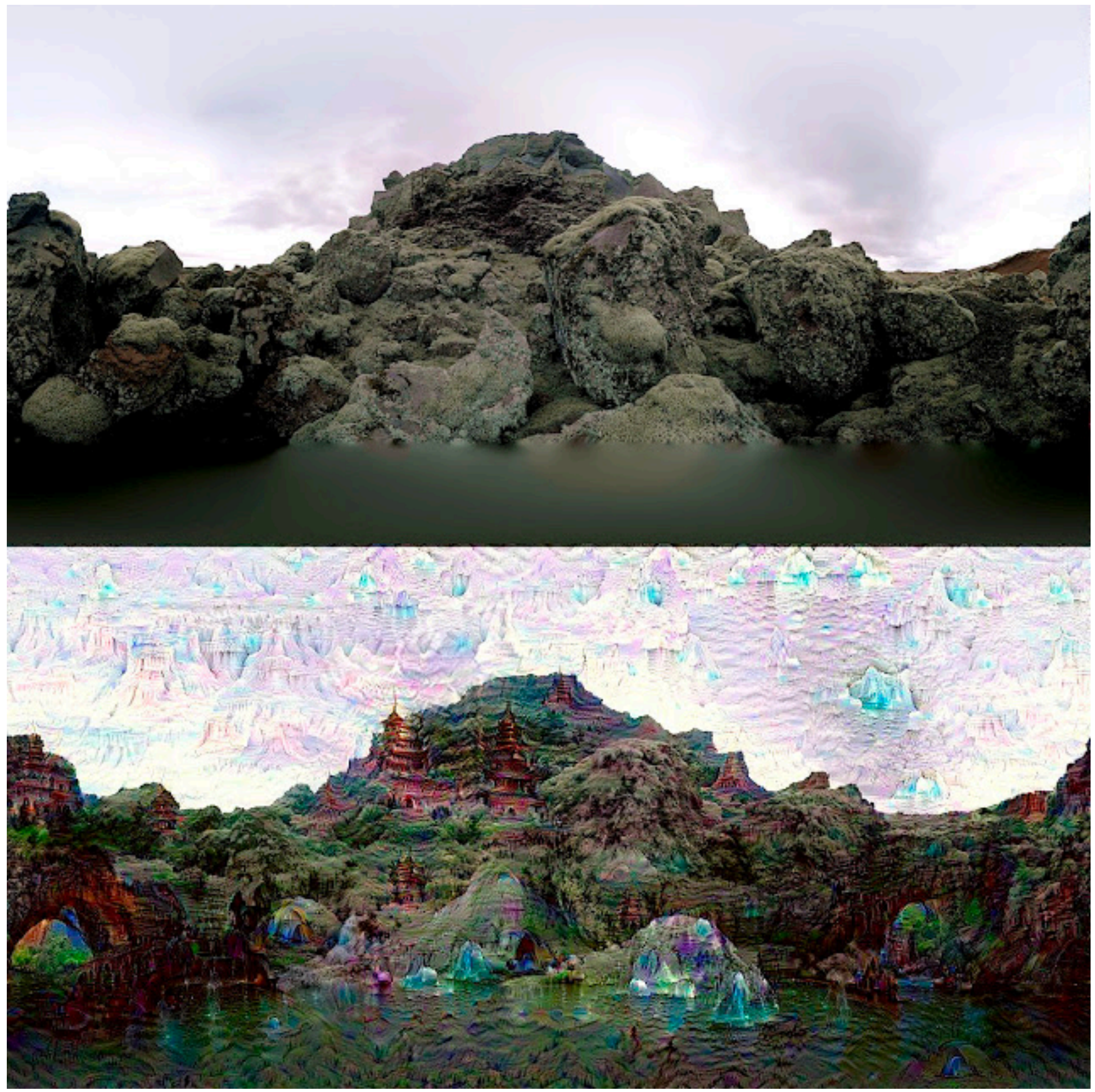
$$\arg\max_I S_c(I) - \lambda \|I\|_2^2, \qquad S_c(I) \approx w^T I + b,$$



dumbbell **cup** **dalmatian**

[Simonyan et al.]

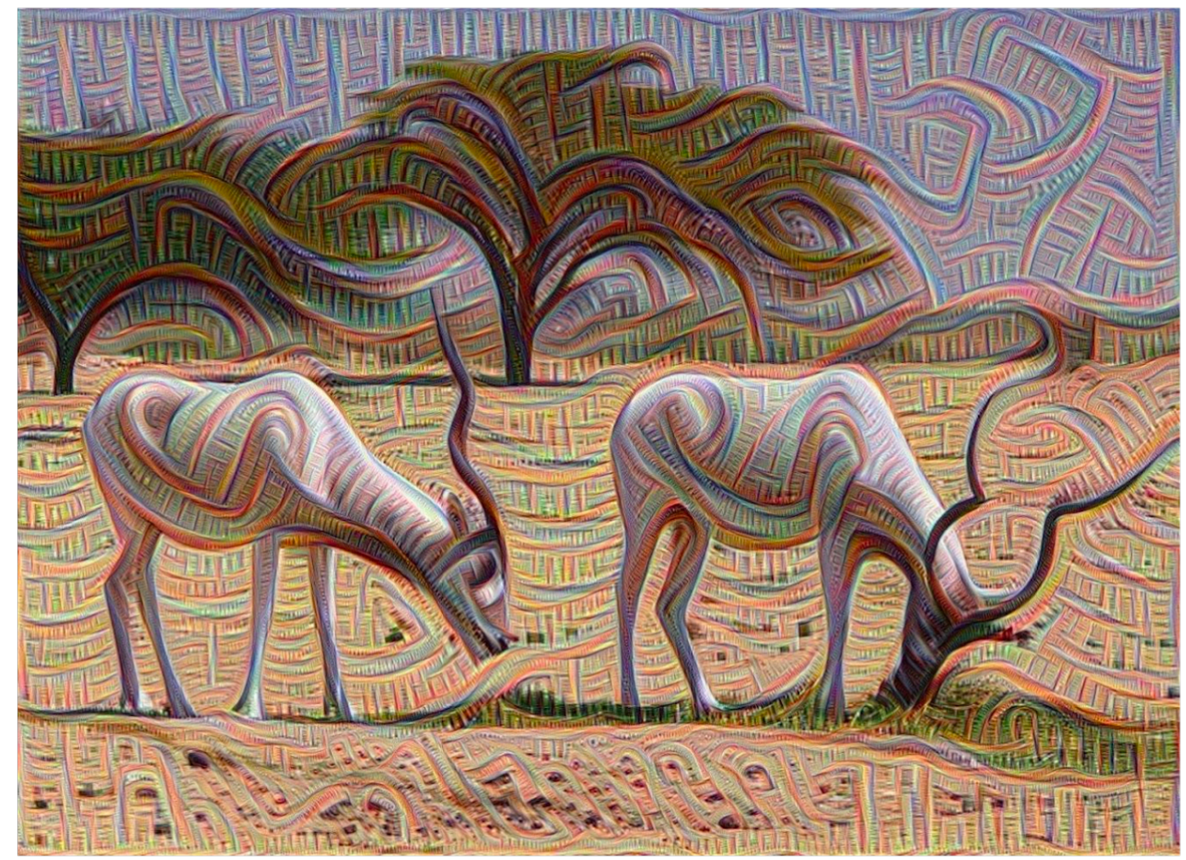
Deep Dream Visualization



Deep Dream Visualization

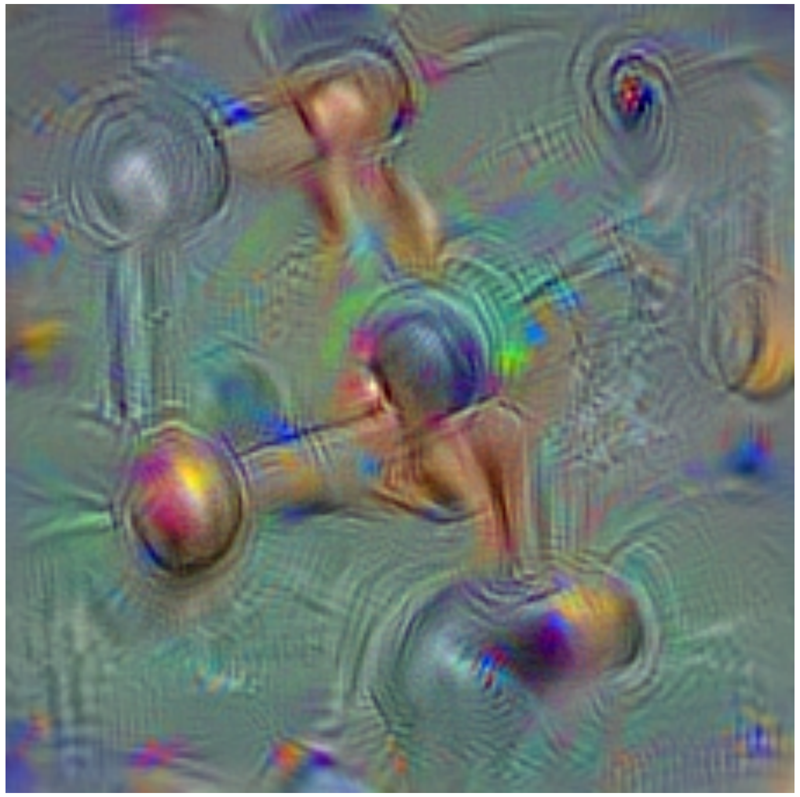
- To produce human viewable images, need to
  - Activity maximization (gradient ascent)
  - L2 regularization
  - Gaussian blur
  - Clipping
  - Multiple scales (octaves)
  - **Code:** https://github.com/google/deepdream/blob/master/dream.ipynb

Example Image

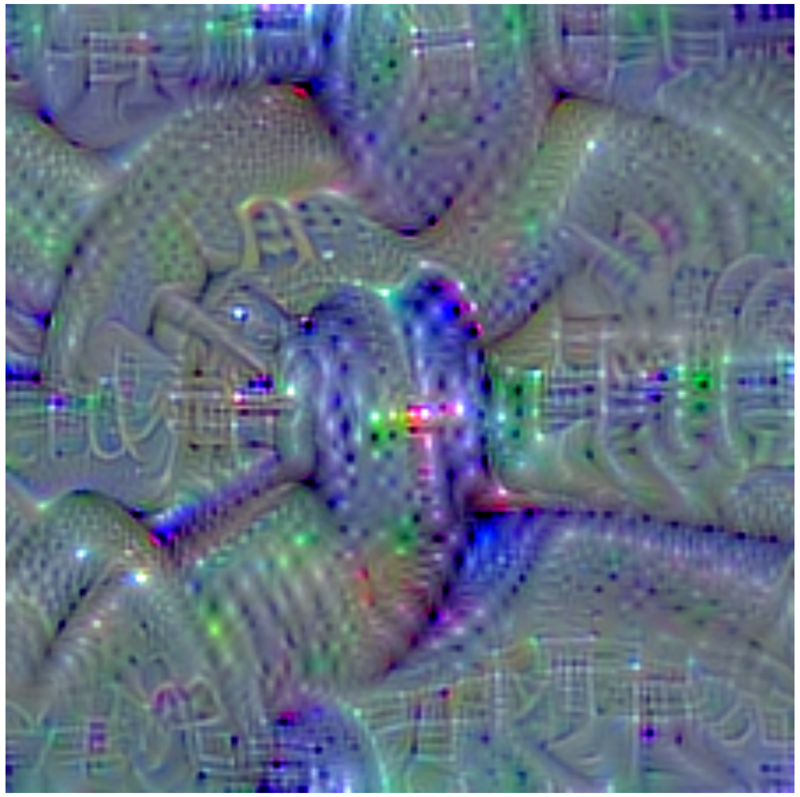


[Inceptionism Gallery]

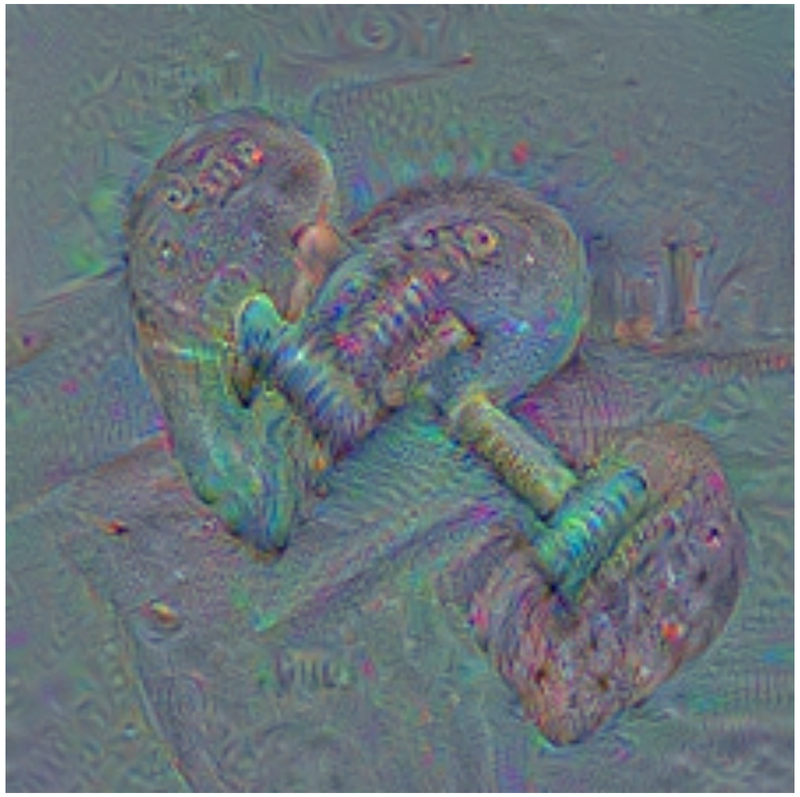
Dumbbell Deep Dream



AlexNet

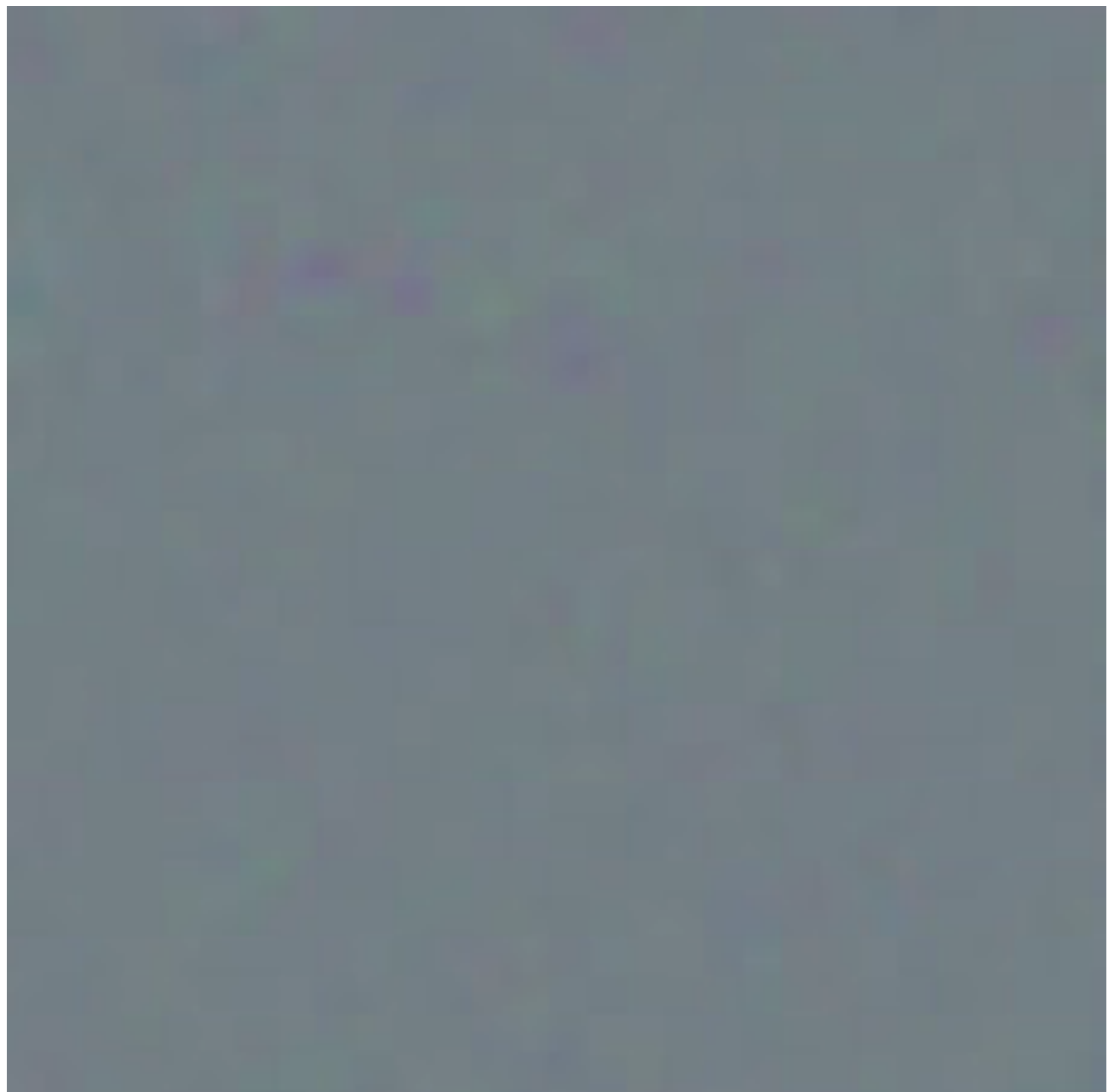


VGGNet



GoogleNet

Deep Dream Video

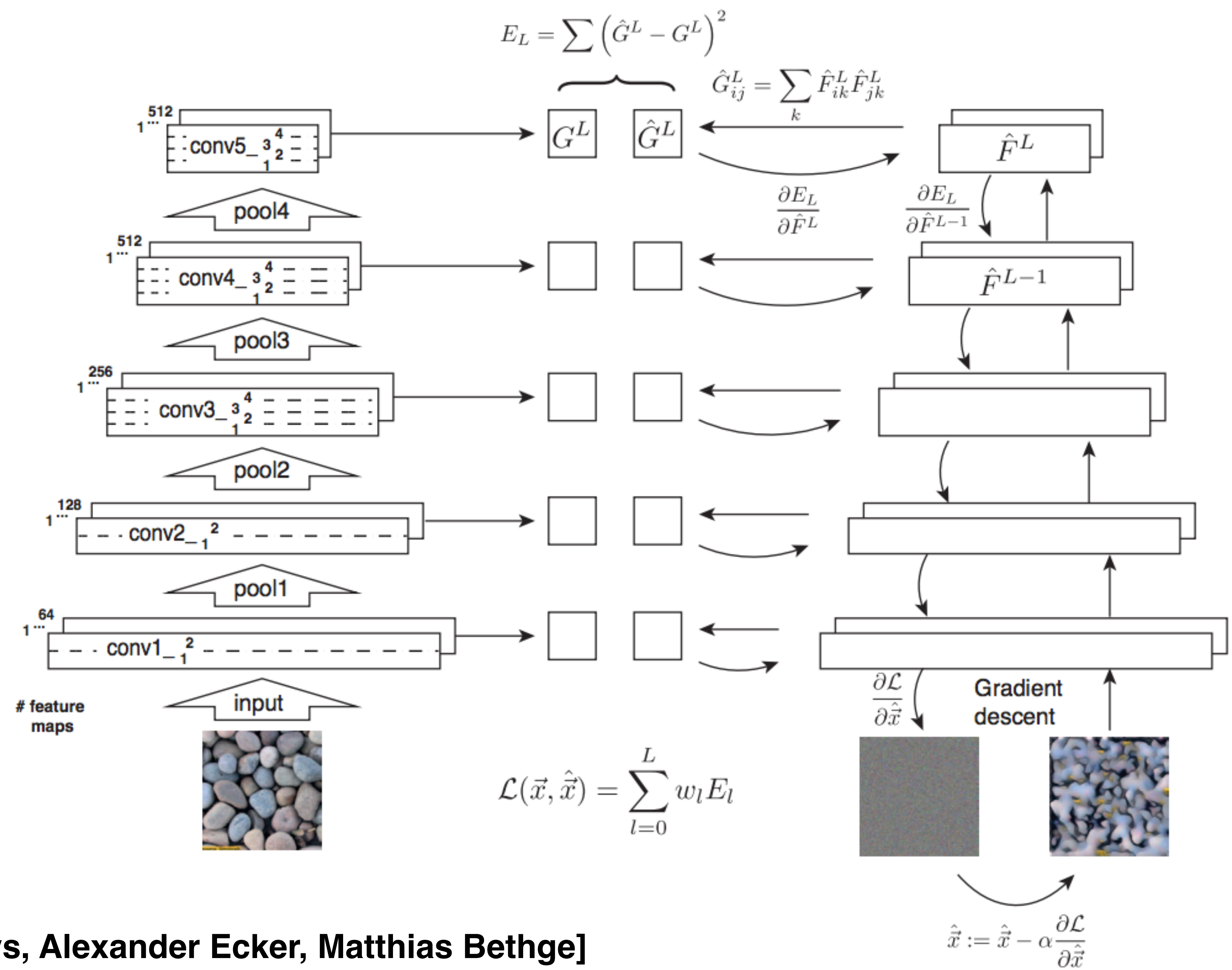


Class: goldfish, Carassius auratus

Infinite Zoom-In on Deep Dream

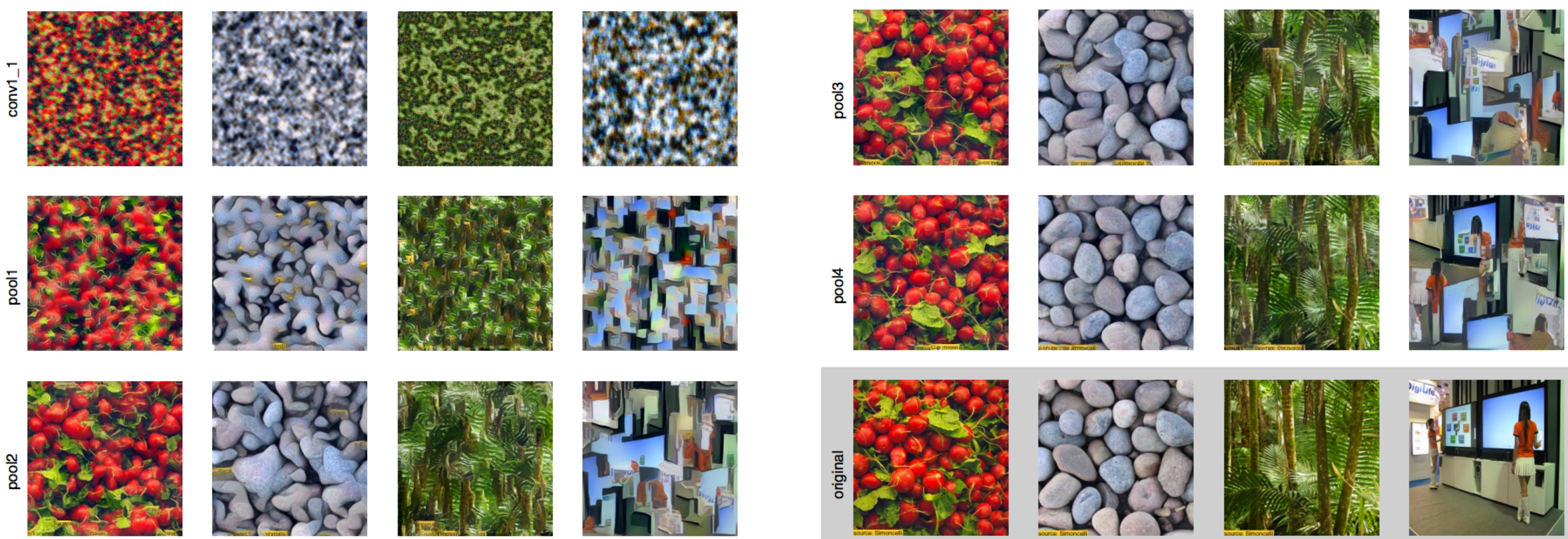
https://www.youtube.com/watch?v=SCE-QeDfXtA

Texture Synthesis



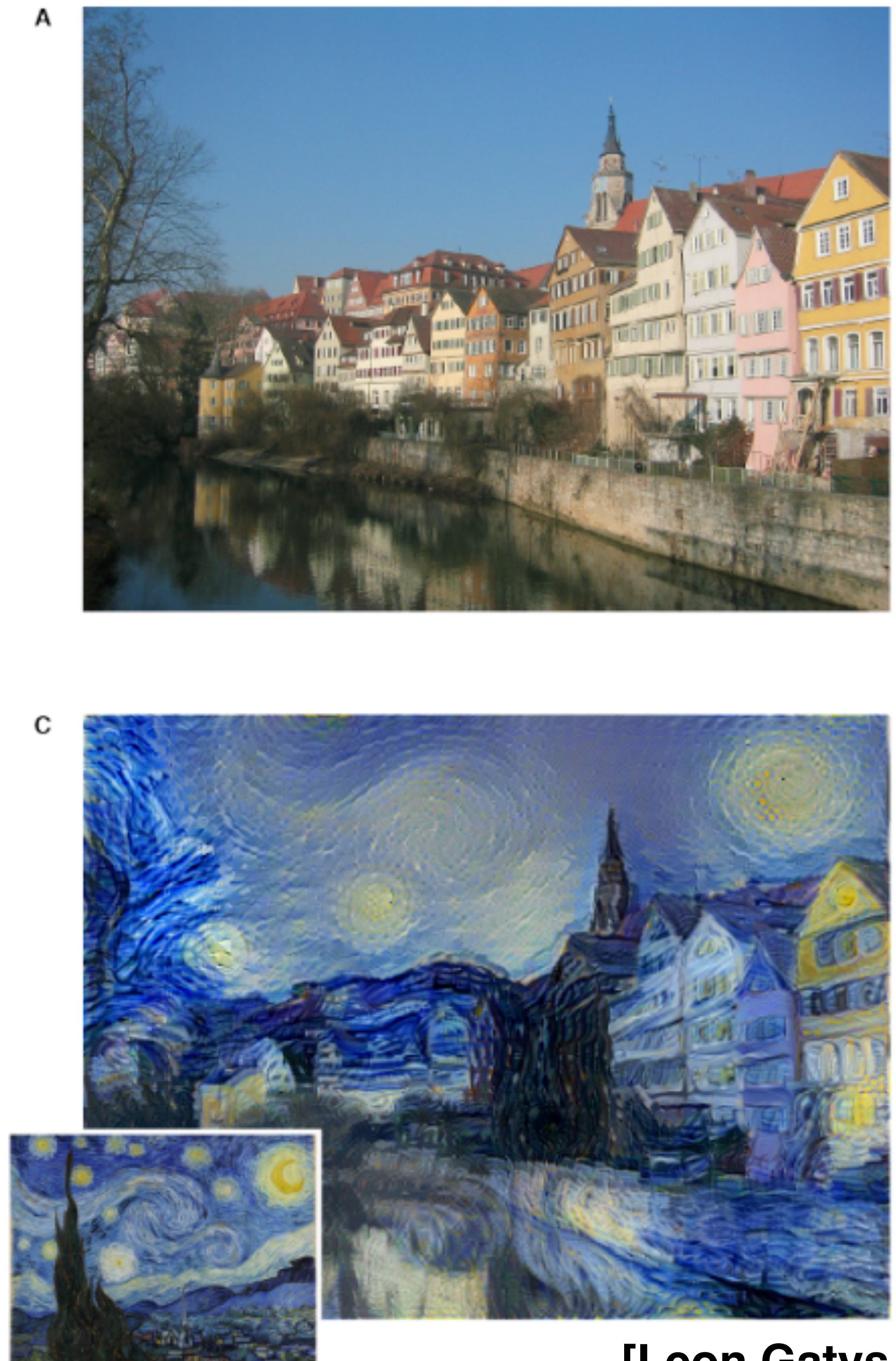
[Leon Gatys, Alexander Ecker, Matthias Bethge]

Generated Textures



[Leon Gatys, Alexander Ecker, Matthias Bethge]

DeepStyle Examples



[Leon Gatys, Alexander Ecker, Matthias Bethge]

DeepStyle: Combining Style + Content from Distinct Images

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2 .$$

The derivative of this loss with respect to the activations in layer l equals

$$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 . \end{cases}$$

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G^l_{ij} - A^l_{ij}\right)^2$$

and the total loss is

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left(\left(F^l\right)^{\mathrm{T}} \left(G^l - A^l\right)\right)_{ji} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 . \end{cases}$$

[Leon Gatys, Alexander Ecker, Matthias Bethge]

Introduction to Recurrent Neural Networks

What Are Recurrent Neural Networks?

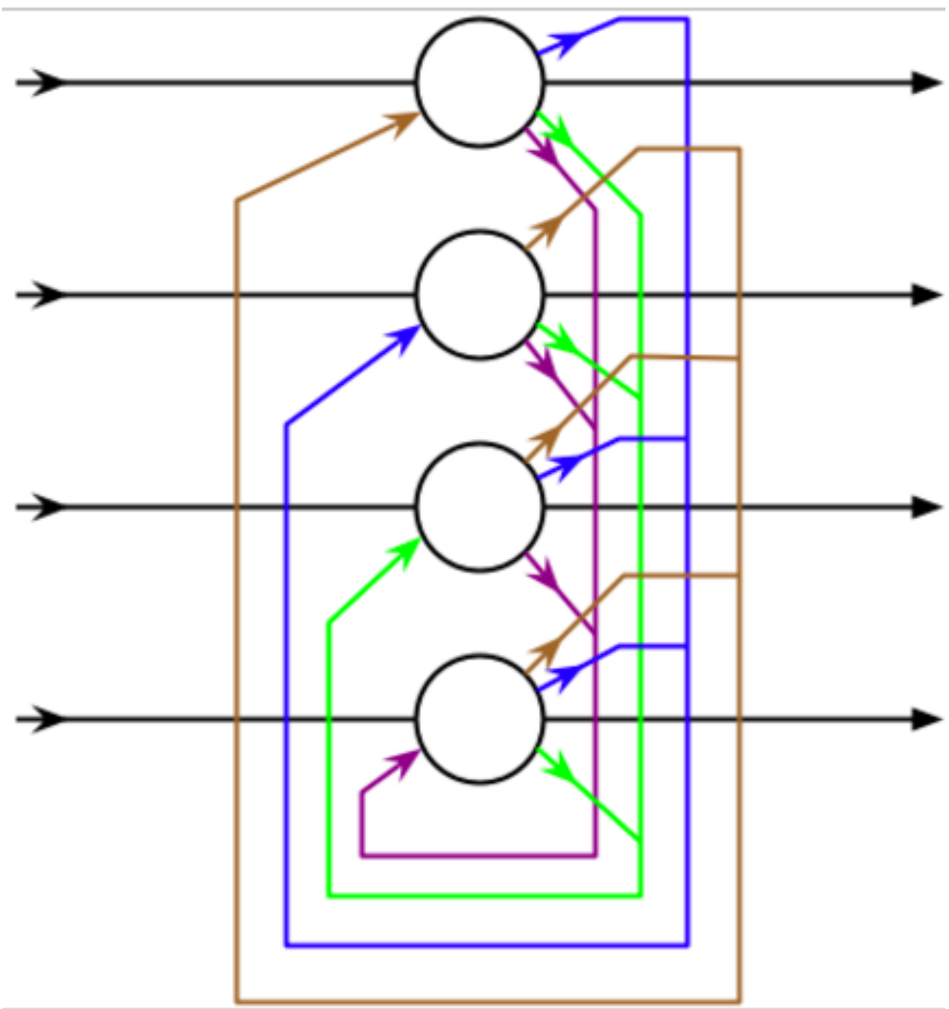
- Recurrent Neural Networks (RNNs) are networks that have feedback
    - Output is feed back to the input
    - Sequence processing
- Ideal for time-series data or sequential data

History of RNNs

Important RNN Architectures

- Hopfield Network
- Jordan and Elman Networks
- Echo State Networks
- Long Short Term Memory (LSTM)
- Bi-Directional RNN
- Gated Recurrent Unit (GRU)
- Neural Turing Machine

Hopfield Network



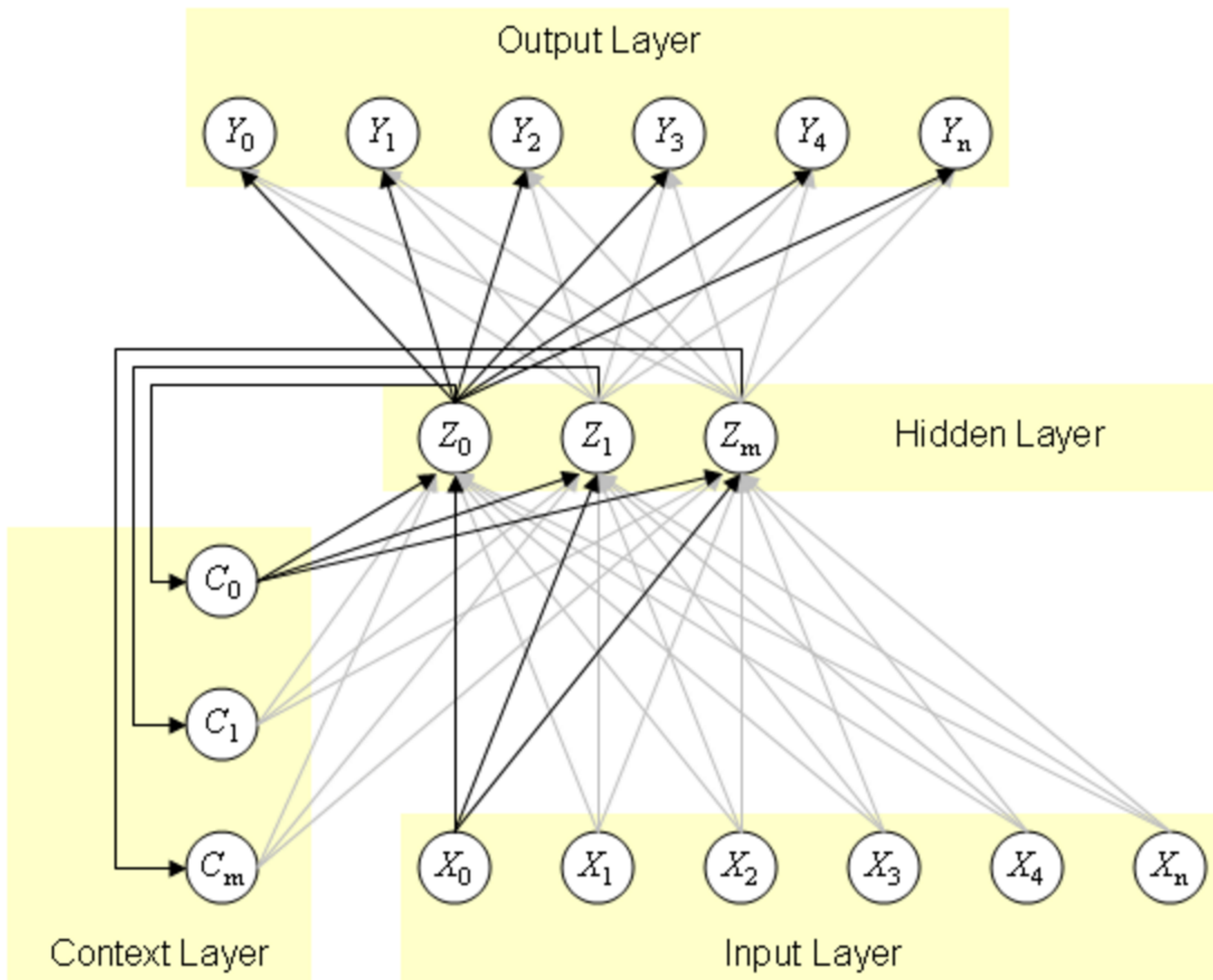
$$s_i \leftarrow \begin{cases} +1 & \text{if } \sum_j w_{ij} s_j \geq \theta_i, \\ -1 & \text{otherwise.} \end{cases}$$

where:

- w_{ij} is the strength of the connection weight from unit j to unit i (the weight of the connection).
- s_j is the state of unit j.
- θ_i is the threshold of unit i.

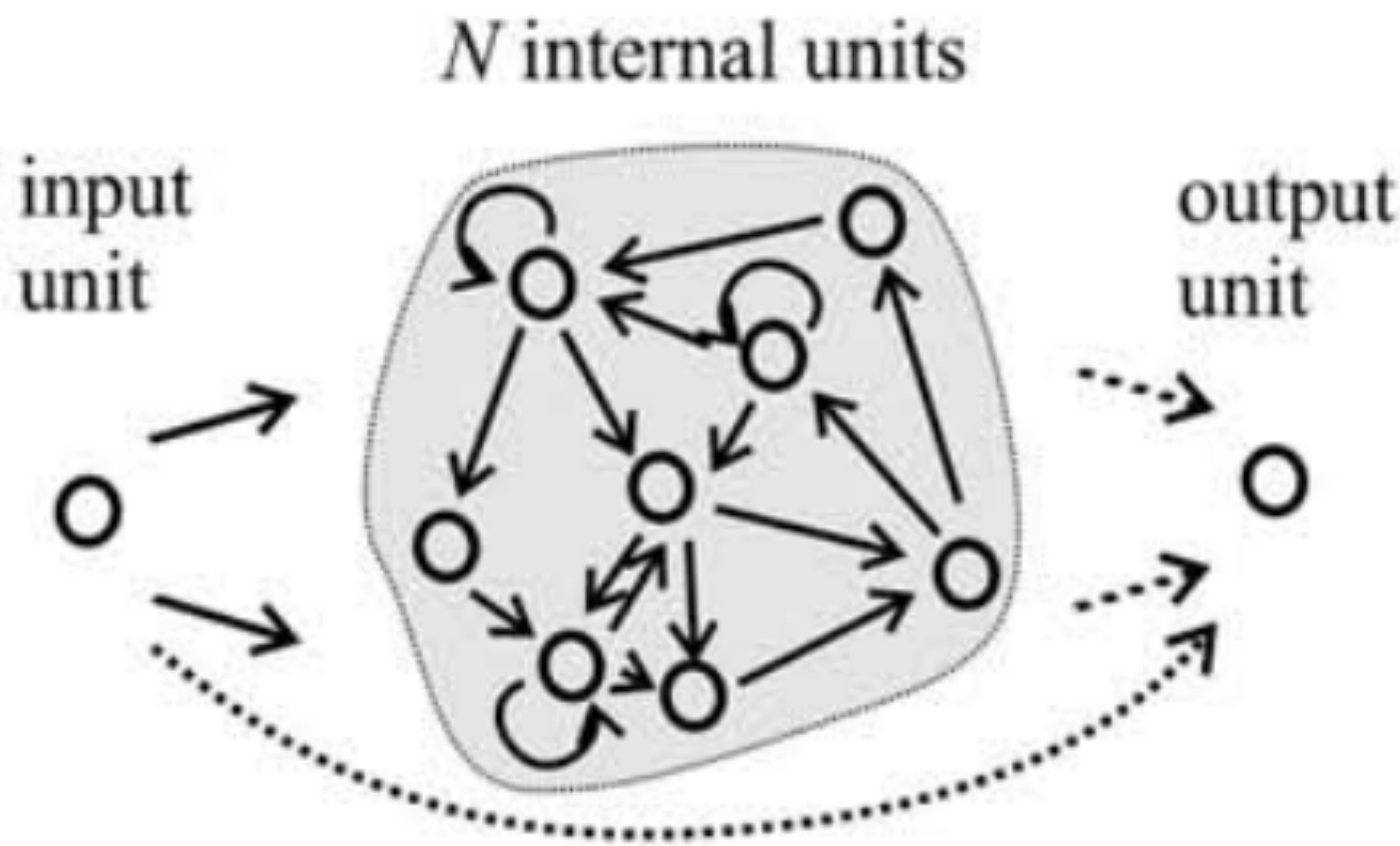
[Wikipedia]

Elman Networks



[John McCullock]

Echo State Networks



[Herbert Jaeger]

Definition of RNNs

RNN Formulation

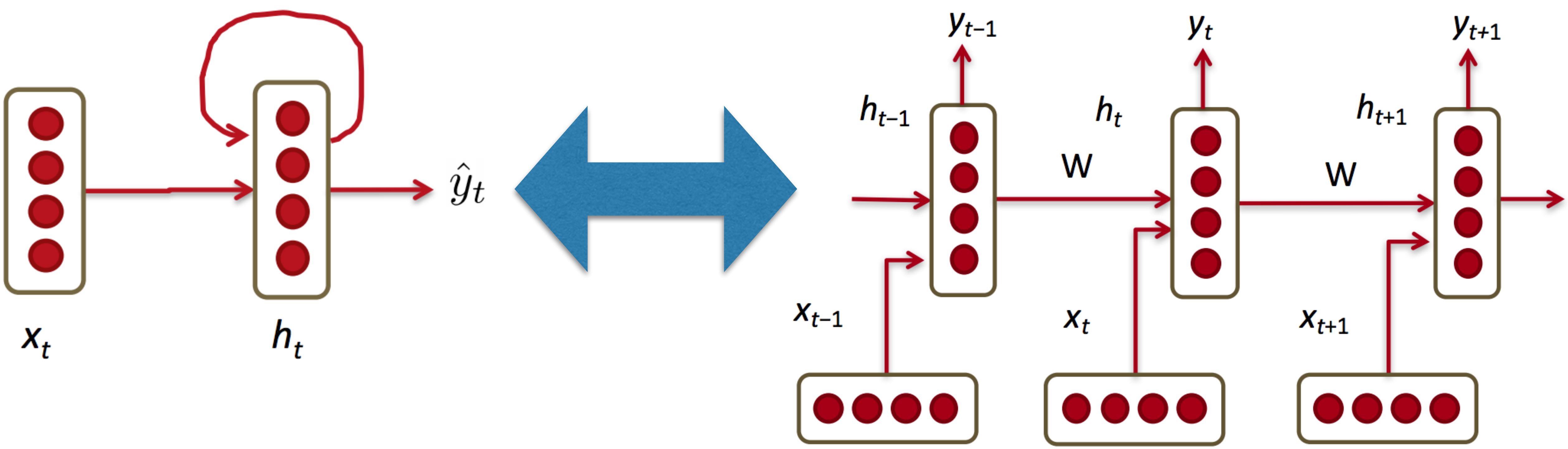
$$x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$$

$$\begin{aligned} h_t &= \sigma\left(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]}\right) \\ \hat{y}_t &= \mathrm{softmax}\left(W^{(S)} h_t\right) \end{aligned}$$

[Richard Socher]

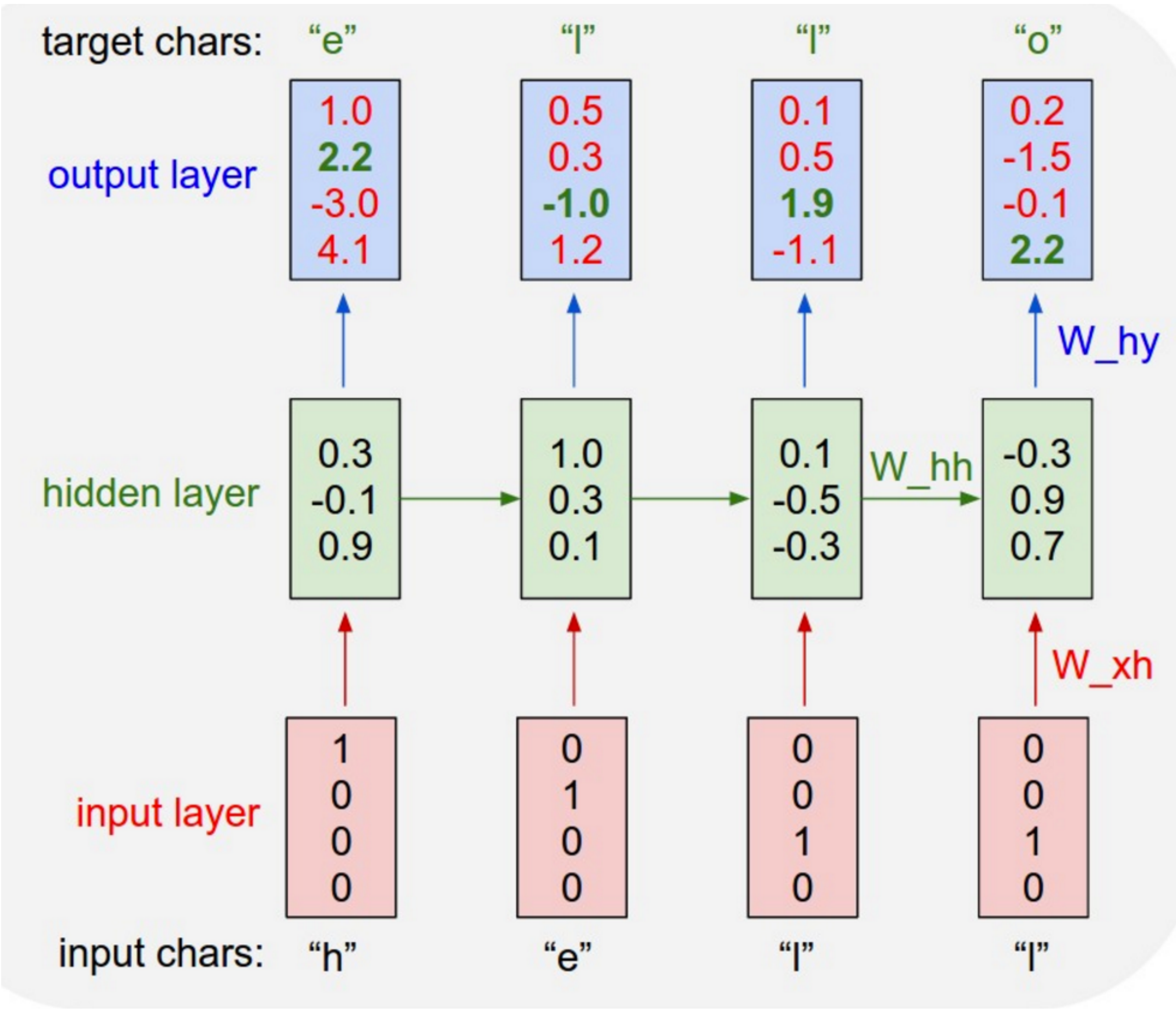
RNN Diagram

Unrolled into FF NN



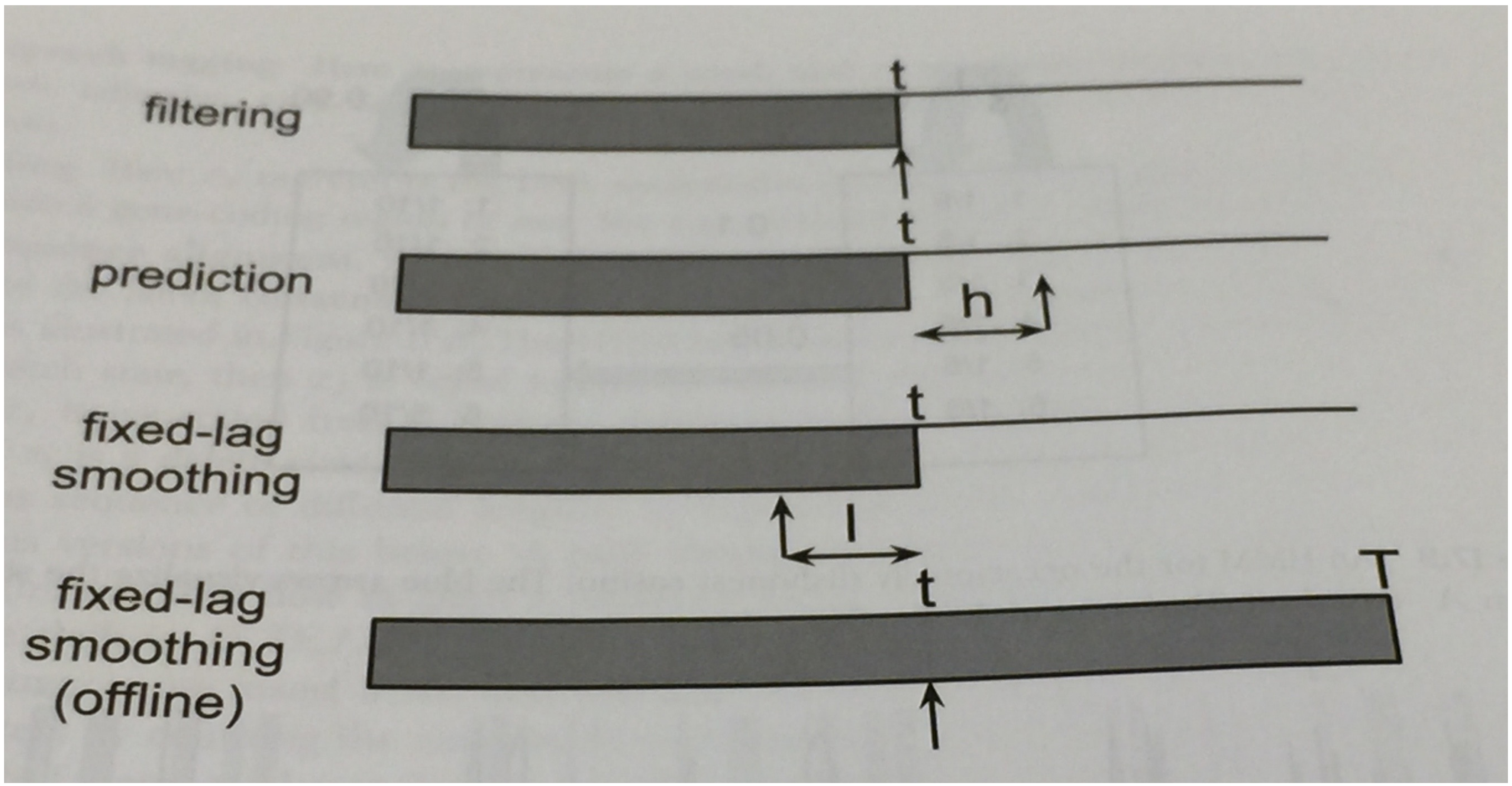
[Richard Socher]

RNN Example



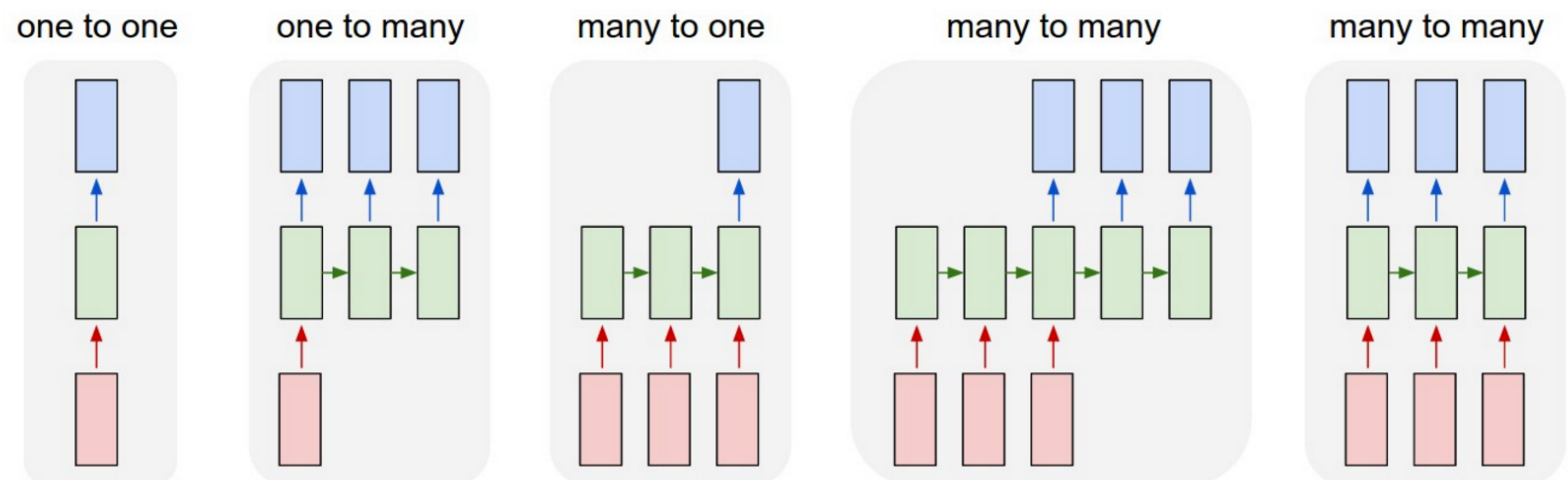
[Andrej Karpathy]

Different Inference Tasks —> Different RNN Architectures



[Kevin Murphy]

Different Structures for Filtering/Prediction Tasks



[Andrej Karpathy]

Object Recognition	Image Captioning	Action Recognition	Machine Translation	Object Tracking

Universal Expressive Power Results

The ***Universal Approximation Theorem*** tells us that:

> Any non-linear dynamical system can be approximated to any accuracy by a recurrent neural network, with no restrictions on the compactness of the state space, provided that the network has enough sigmoidal hidden units.

This underlies the computational power of recurrent neural networks.

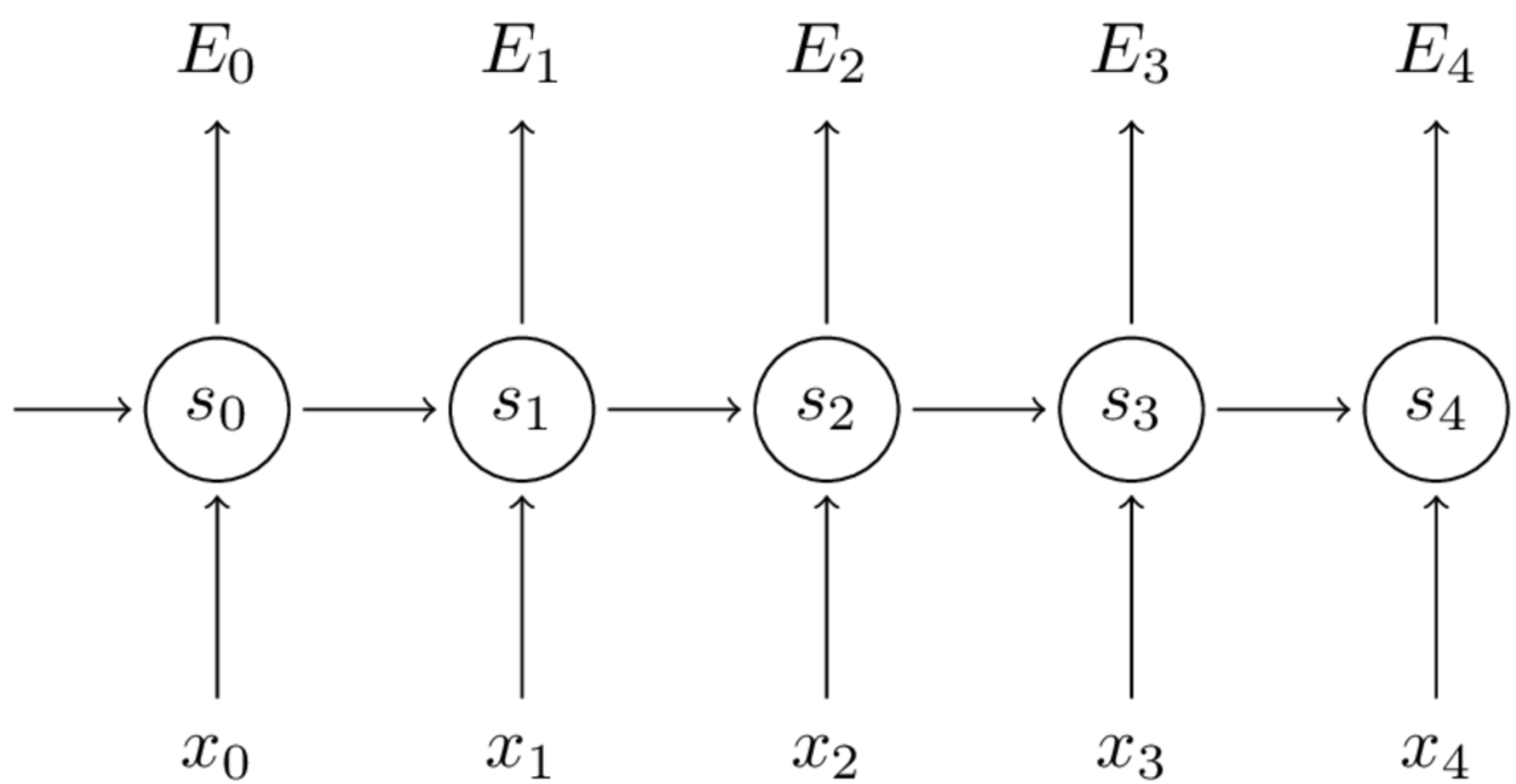
[John Bullinaria]

Training RNNs

Training an RNN

- Use back propagation through time (BPTT)

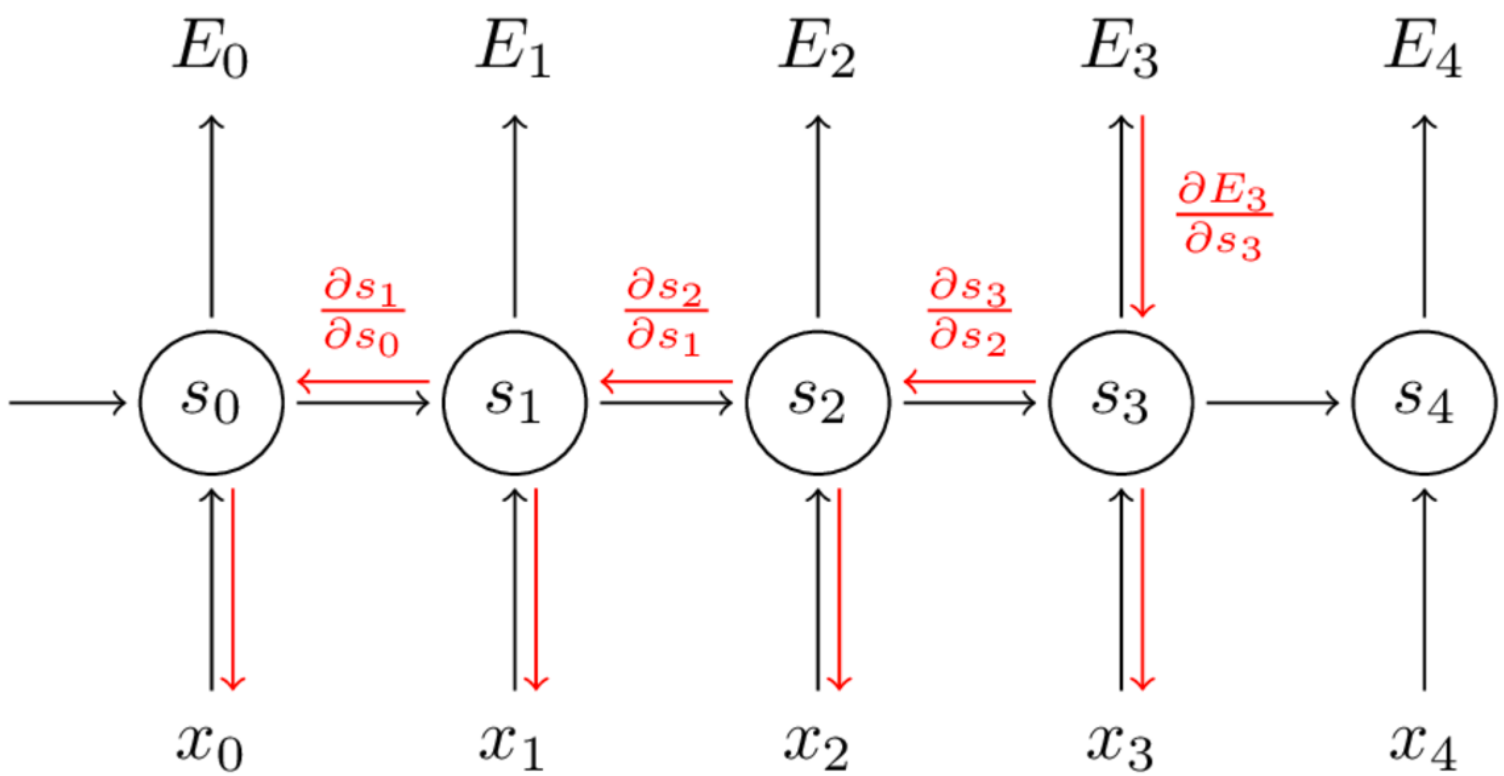
$$\begin{aligned} E_t(y_t, \hat{y}_t) &= -y_t \log \hat{y}_t \\ E(y, \hat{y}) &= \sum_t E_t(y_t, \hat{y}_t) \\ &= -\sum_t y_t \log \hat{y}_t \end{aligned}$$



[Denny Britz]

Back Propagation through Time

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}.$$



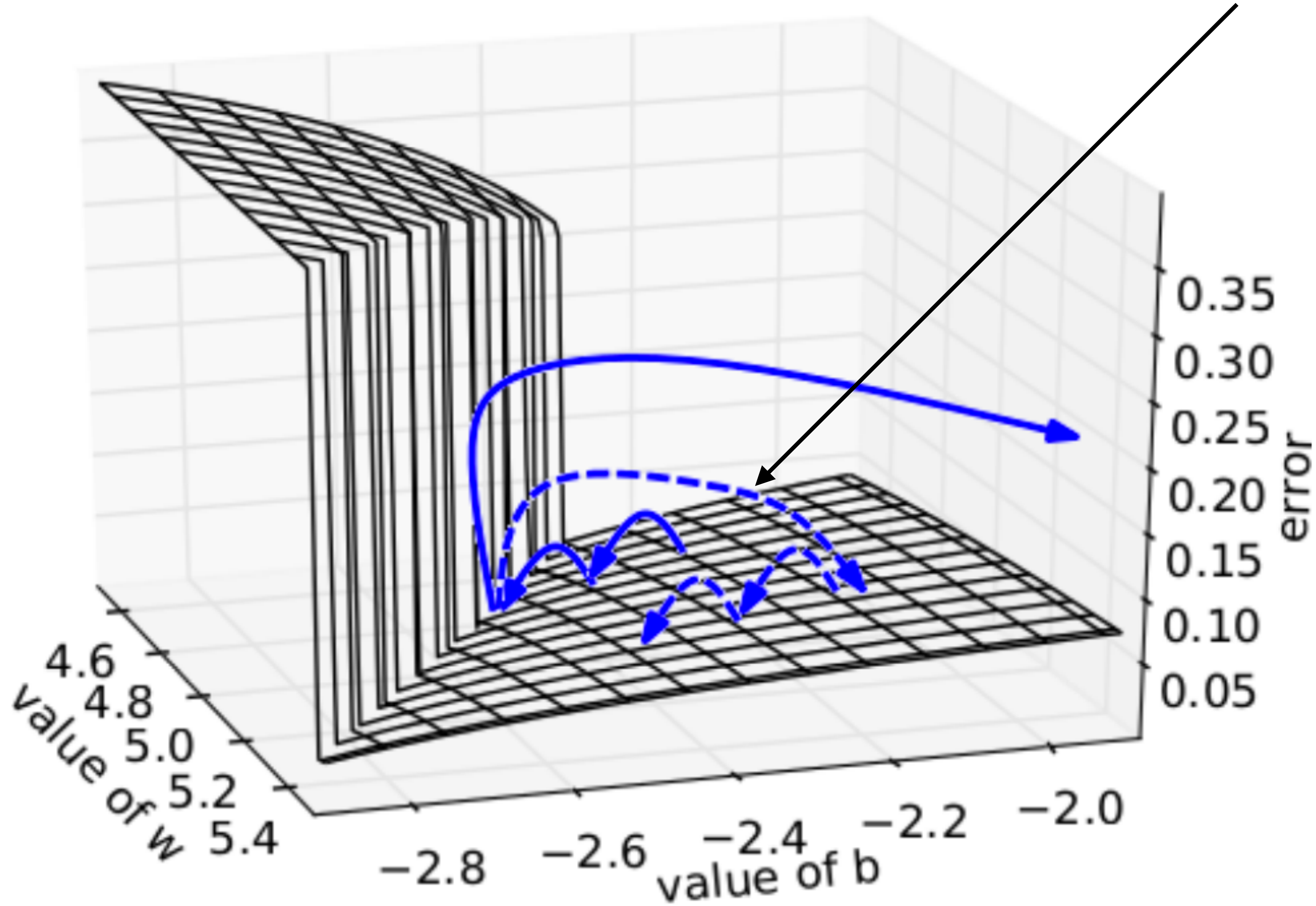
[Denny Britz]

RNN Training Issues

- Exploding/Vanishing gradients
- Exploding/Vanishing activations

Exploding Gradients

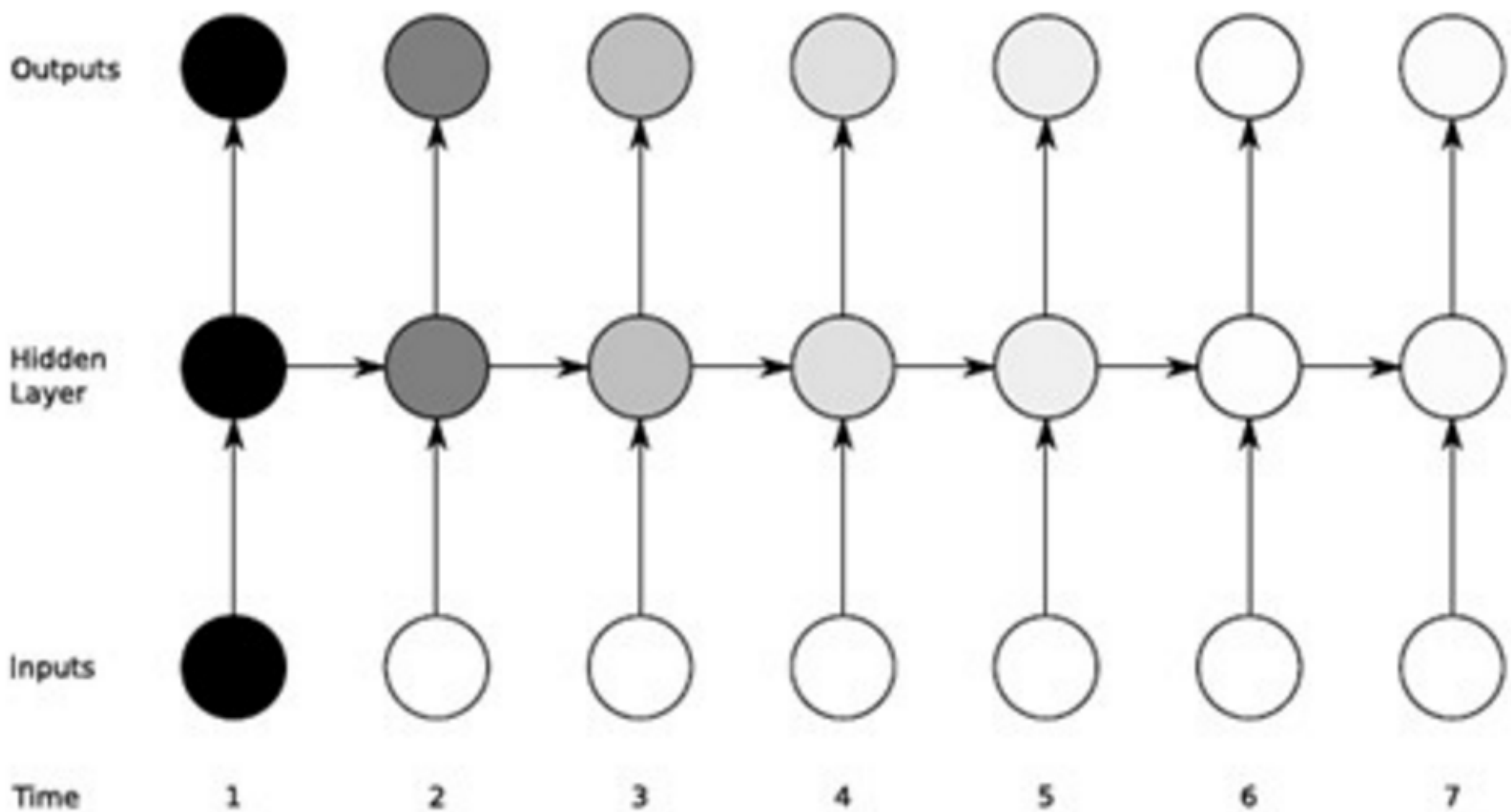
Solution: Gradient Clipping



[Richard Socher]

http://www.jmlr.org/proceedings/papers/v28/pascanu13.pdf

Vanishing Gradients/ Activations



[Hochreiter, Schmidhuber]

Why Training is Unstable

$$x^{(l)} = W^{(l-1)}y^{(l-1)} + b^{(l-1)}$$
$$y^{(l)} = f(x^{(l)})$$

Let the activation function $f(x) = \alpha x + \beta$,

$$Var\left(y^{(l)}\right) = \alpha^2 n_{l-1}\sigma_{l-1}^2\left(Var\left(y^{(l-1)}\right) + \beta^2 I_{n_l}\right).$$

$$Var\left(\frac{\partial cost}{\partial y^{(l-1)}}\right) = \alpha^2 n_l \sigma_{l-1}^2 Var\left(\frac{\partial cost}{\partial y^{(l)}}\right).$$

Variance of activations/gradients grows multiplicatively

[Xu, Huang, Li]

Interesting Question

- Are there modifications to an RNN such that it can combat these activations/gradient problems?

RNNs with Longer Term Memory

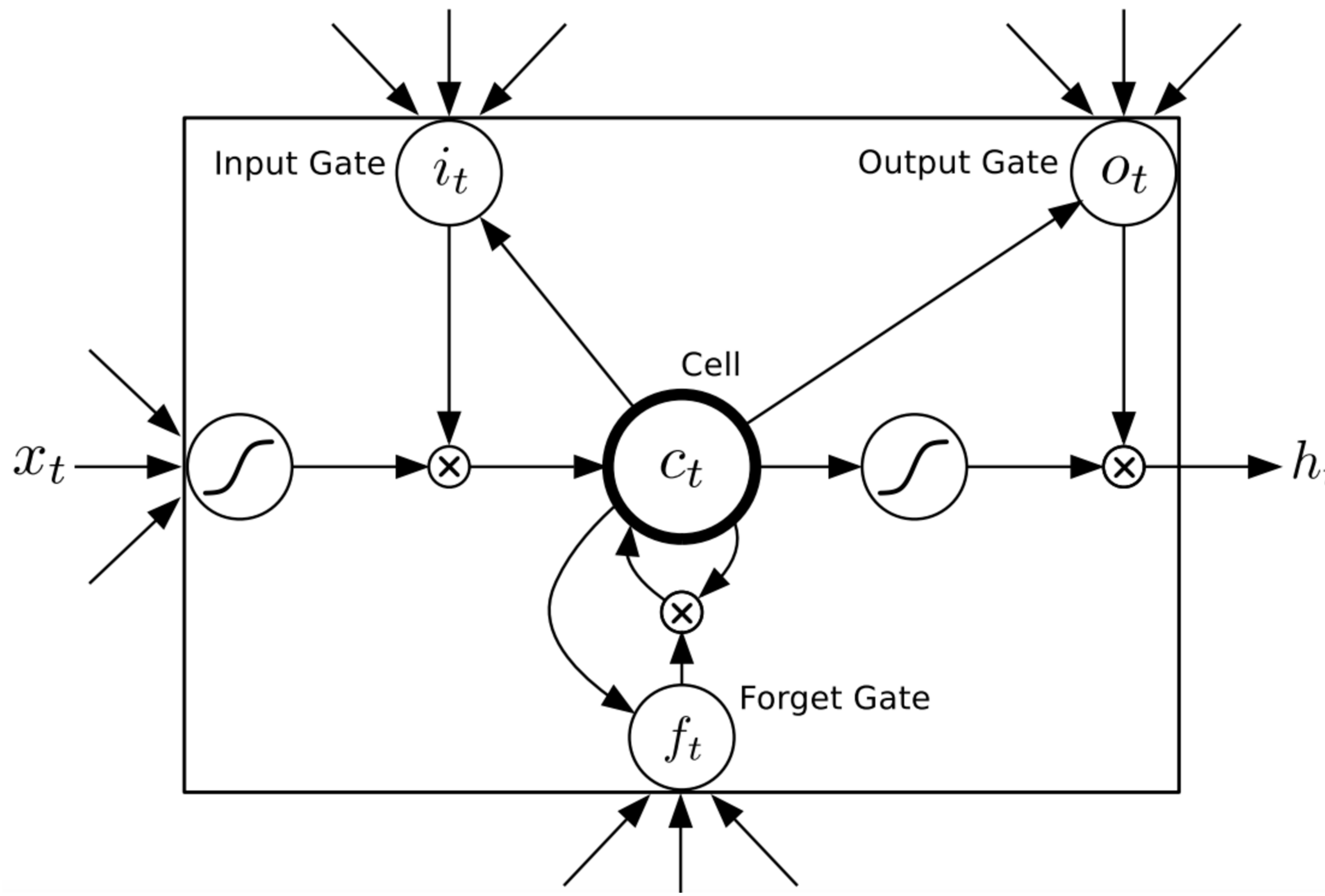
Motivation

- The need to remember certain events for arbitrarily long periods of time (Non-Markovian)

- The need to forget certain events

Long Short Term Memory

- 3 gates
  - Input
  - Forget
  - Output



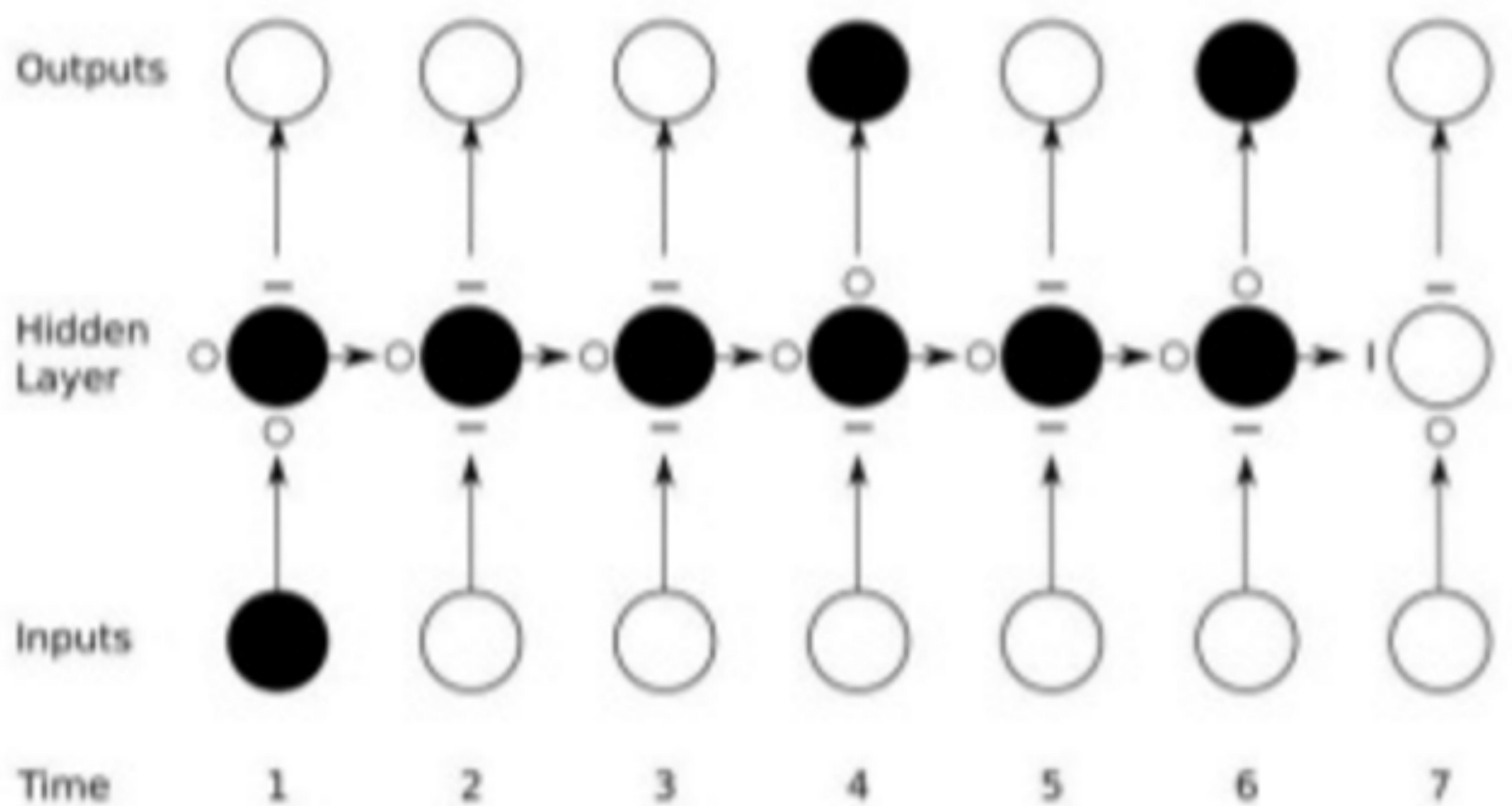
[Zygmunt Z.]

LSTM Formulation

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$
$$y_t = W_{ho}h_t + b_o$$

[Alex Graves, Navdeep Jaitly]

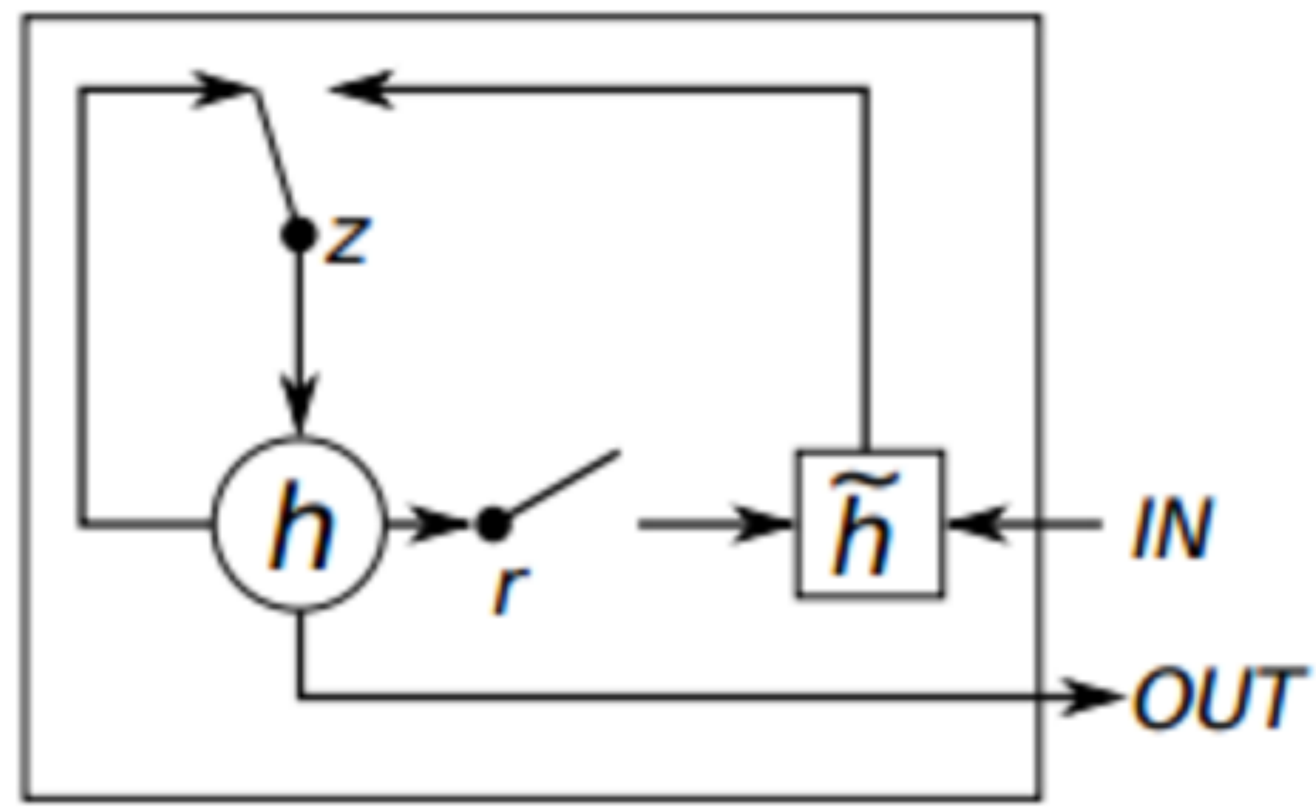
Preserving Gradients



[Hochreiter, Schmidhuber]

Gated Recurrent Unit

- 2 gates
  - Reset
    - Combine new input with previous memory
  - Update
    - How long the previous memory should stay



[Zygmunt Z.]

GRU Formulation

$$z = \sigma(x_t U^z + s_{t-1} W^z)$$
$$r = \sigma(x_t U^r + s_{t-1} W^r)$$
$$h = tanh(x_t U^h + (s_{t-1} \circ r) W^h)$$
$$s_t = (1 - z) \circ h + z \circ s_{t-1}$$

[Danny Britz]

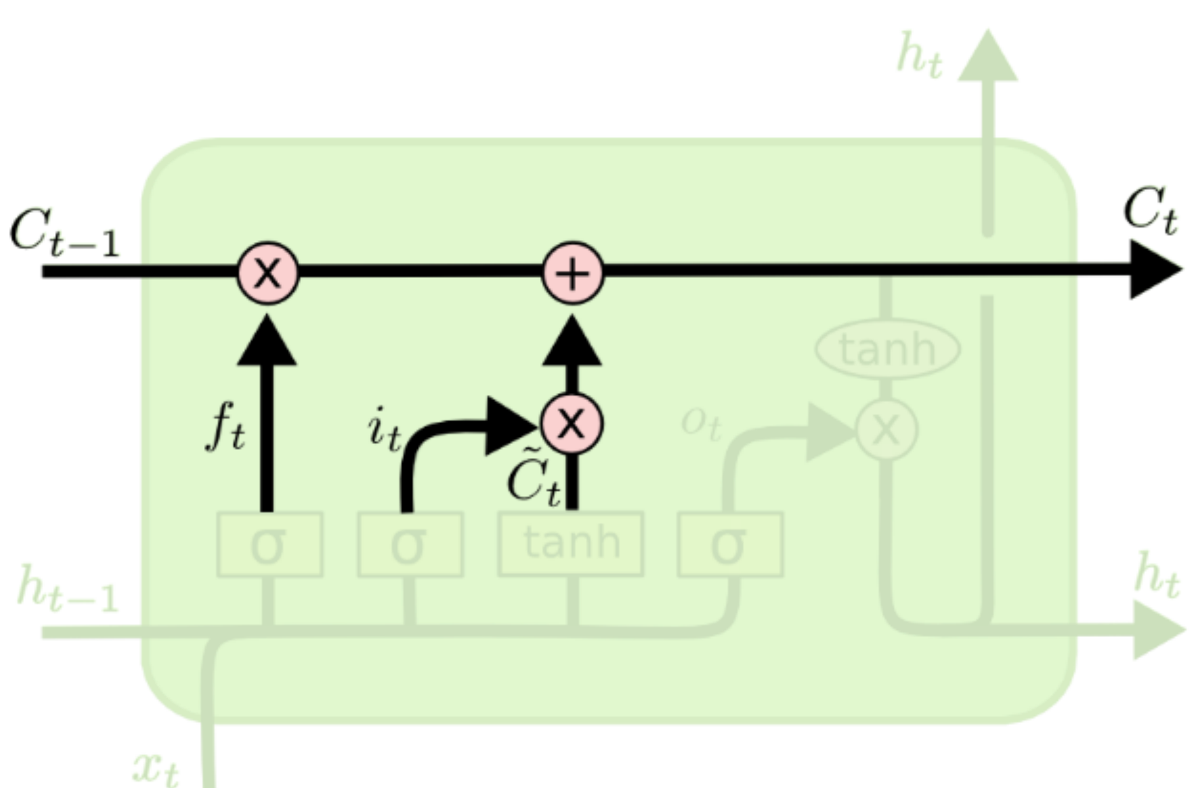
LSTM & GRU Benefits

- Remember for longer temporal durations
  - RNN has issues for remembering longer durations
- Able to have feedback flow at different strengths depending on inputs

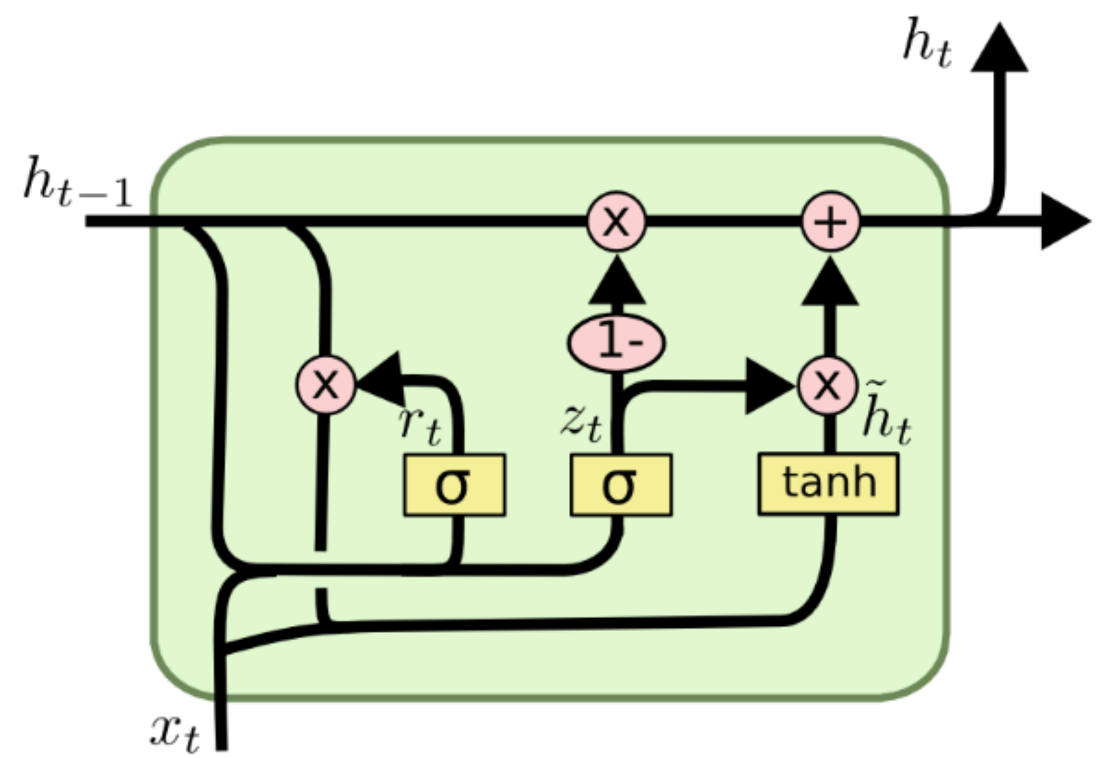
Differences between LSTM & GRU

- GRU has two gates, while LSTM has three gates
- GRU does not have internal memory
- GRU does not use a second nonlinearity for computing the output

Visual Difference of LSTM & GRU



LSTM



GRU

[Chris Olah]

LSTM vs GRU Results

			tanh	GRU	LSTM
Music Datasets	Nottingham	train	3.22	2.79	3.08
		test	**3.13**	3.23	3.20
	JSB Chorales	train	8.82	6.94	8.15
		test	9.10	**8.54**	8.67
	MuseData	train	5.64	5.06	5.18
		test	6.23	**5.99**	6.23
	Piano-midi	train	5.64	4.93	6.49
		test	9.03	**8.82**	9.03
Ubisoft Datasets	Ubisoft dataset A	train	6.29	2.31	1.44
		test	6.44	3.59	**2.70**
	Ubisoft dataset B	train	7.61	0.38	0.80
		test	7.62	**0.88**	1.26

[Chung, Gulcehre, Cho, Bengio]

Other Methods for Stabilizing RNN Training

Why Training is Unstable

$$x^{(l)} = W^{(l-1)} y^{(l-1)} + b^{(l-1)}$$
$$y^{(l)} = f(x^{(l)})$$

Let the activation function $f(x) = \alpha x + \beta$,

$$Var\left(y^{(l)}\right) = \alpha^2 n_{l-1} \sigma_{l-1}^2 \left(Var\left(y^{(l-1)}\right) + \beta^2 I_{n_l}\right).$$

$$Var\left(\frac{\partial cost}{\partial y^{(l-1)}}\right) = \alpha^2 n_l \sigma_{l-1}^2 Var\left(\frac{\partial cost}{\partial y^{(l)}}\right).$$

Variance of activations/gradients grows multiplicatively

[Xu, Huang, Li]

Stabilizing Activations & Gradients

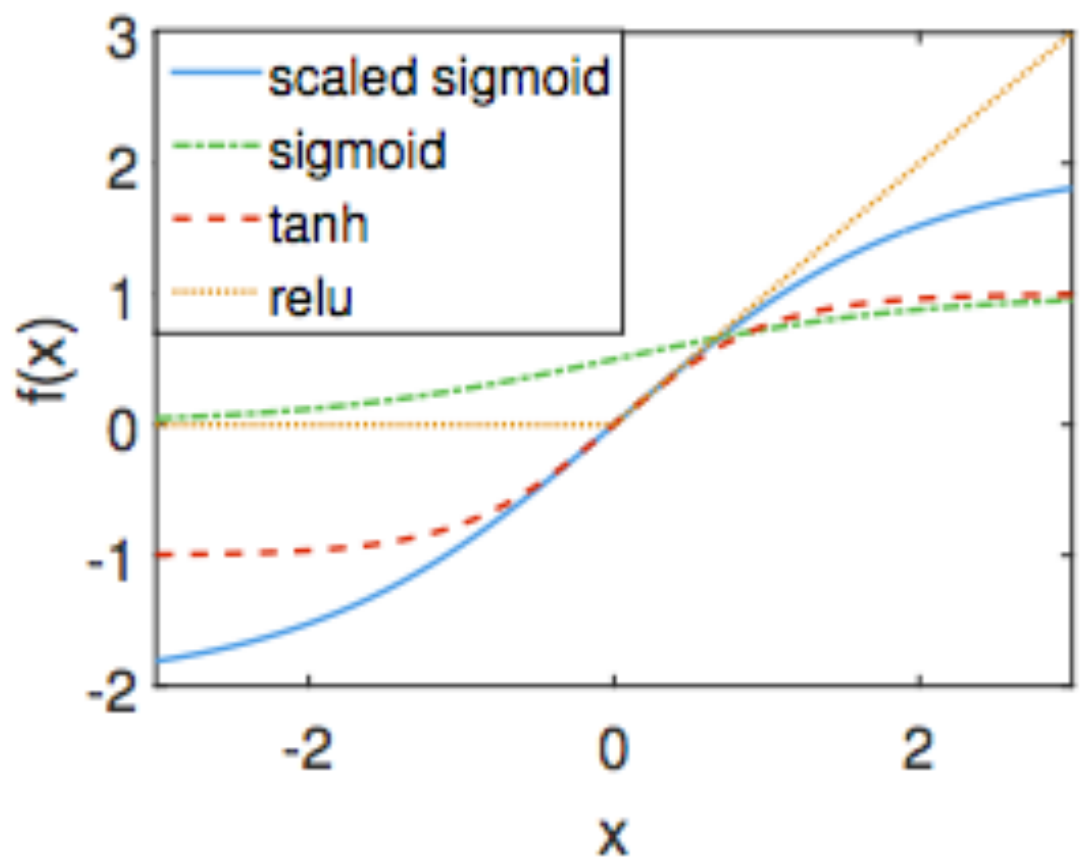
$$\mathrm{Var}\left(y^{(l)}\right) = \mathrm{Var}\left(y^{(l-1)}\right) \quad \text{and} \quad \mathrm{Var}\left(\frac{\partial \mathrm{cost}}{\partial y^{(l)}}\right) = \mathrm{Var}\left(\frac{\partial \mathrm{cost}}{\partial y^{(l-1)}}\right);$$

$$n_l \sigma_{l-1}^2 \approxeq 1 \quad \text{and} \quad n_{l-1} \sigma_{l-1}^2 \approxeq 1;$$

We want $\alpha = 1 \quad \text{and} \quad \beta = 0.$

[Xu, Huang, Li]

Taylor Expansions of Different Activation Functions



$$\mathrm{sigmoid}(x) = \frac{1}{2} + \frac{x}{4} - \frac{x^3}{48} + O(x^5)$$

$$\tanh(x) = 0 + x - \frac{x^3}{3} + O(x^5)$$

$$\mathrm{relu}(x) = 0 + x \quad \text{for } x \geq 0.$$

[Xu, Huang, Li]

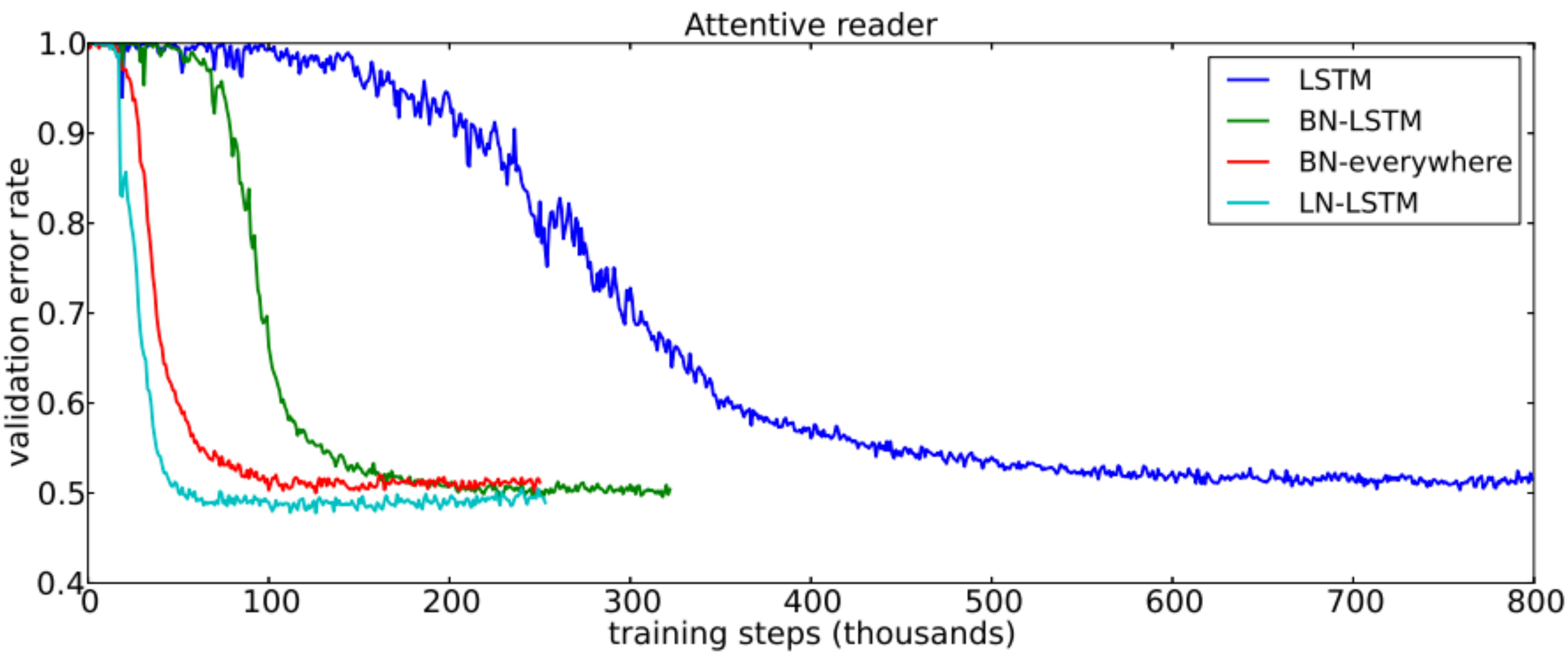
Layer Normalization

- Similar to batch normalization
  - Apply it to RNNs to stabilize the hidden state dynamics

$$\mathbf{h}^t = f\left[\frac{\mathbf{g}}{\sigma^t} \odot \left(\mathbf{a}^t - \mu^t\right) + \mathbf{b}\right] \qquad \mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t \qquad \sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^t - \mu^t\right)^2}$$

[Ba, Kiros, Hinton]

Layer Normalization Results

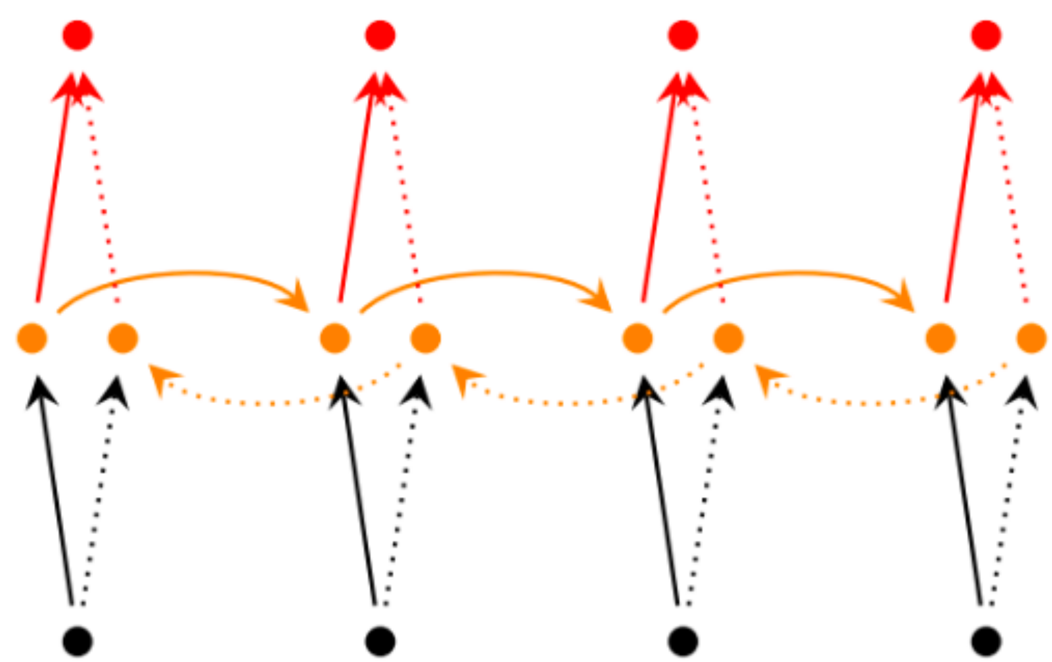


[Ba, Kiros, Hinton]

Variants of RNNs

Bidirectional RNNs

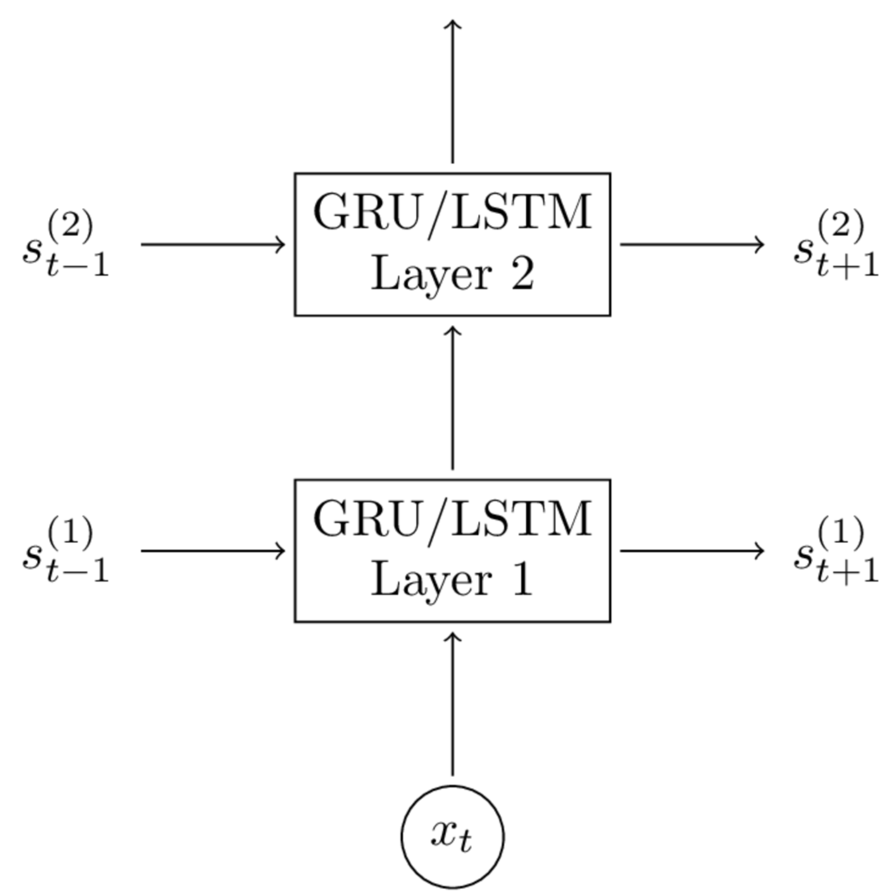
- The output at time *t* does not depend on previous time steps but also the future
    - Two RNNs stacked on top of each other



[Danny Britz]

Deep RNNs

- Stack them on top of each other
  - The output of the previous RNN is the input to the next one



[Danny Britz]

The Power of RNNs: Understanding and Visualizing

The Effectiveness of an RNN

```
#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP  AFSR(0, load)
#define STACK_DDR(type)     (func)

#define SWAP_ALLOCATE(nr)     (e)
#define emulate_sigs()  arch_get_unaligned_child()
#define access_rw(TST)  asm volatile("movd %%esp, %0, %3" : : "r" (0));   \
  if (__type & DO_READ)

static void stat_PC_SEC __read_mostly offsetof(struct seq_argsqueue, \
          pC>[1]);

static void
os_prefix(unsigned long sys)
{
#ifdef CONFIG_PREEMPT
  PUT_PARAM_RAID(2, sel) = get_state_state();
  set_pid_sum((unsigned long)state, current_state_str(),
           (unsigned long)-1->lr_full; low;
}
```

[Andrej Karpathy]

The Effectiveness of an RNN

```
Naturalism and decision for the majority of Arab countries' capitalide was grounded
by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated
with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal
in the [[Protestant Immineners]], which could be said to be directly in Cantonese
Communication, which followed a ceremony and set inspired prison, training. The
emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom
of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known
in western [[Scotland]], near Italy to the conquest of India with the conflict.
Copyright was the succession of independence in the slop of Syrian influence that
was a famous German movement based on a more popular servicious, non-doctrinal
and sexual power post. Many governments recognize the military housing of the
[[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]],
that is sympathetic to be to the [[Punjab Resolution]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.htm Official economics Adjoint for the Nazism, Montgomery
was swear to advance to the resources for those Socialism's rule,
was starting to signing a major tripad of aid exile.]]
```

[Andrej Karpathy]

The Effectiveness of an RNN

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset X$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

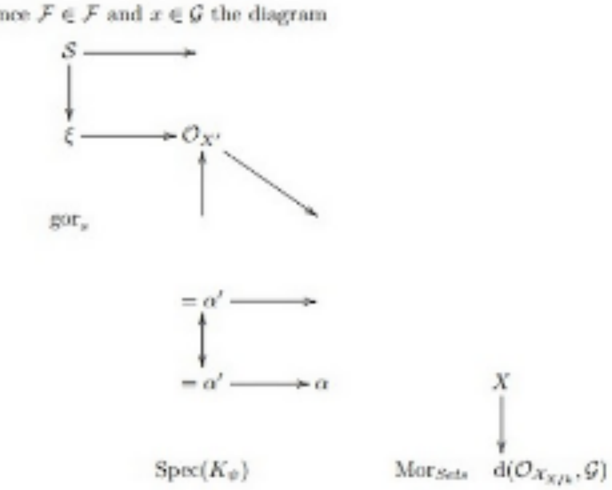
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\bar{x}} \ {-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

[Andrej Karpathy]

The Effectiveness of an RNN

Trained on *War & Peace*

Iteration: 100

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

Iteration: 300

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

Iteration: 2000

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

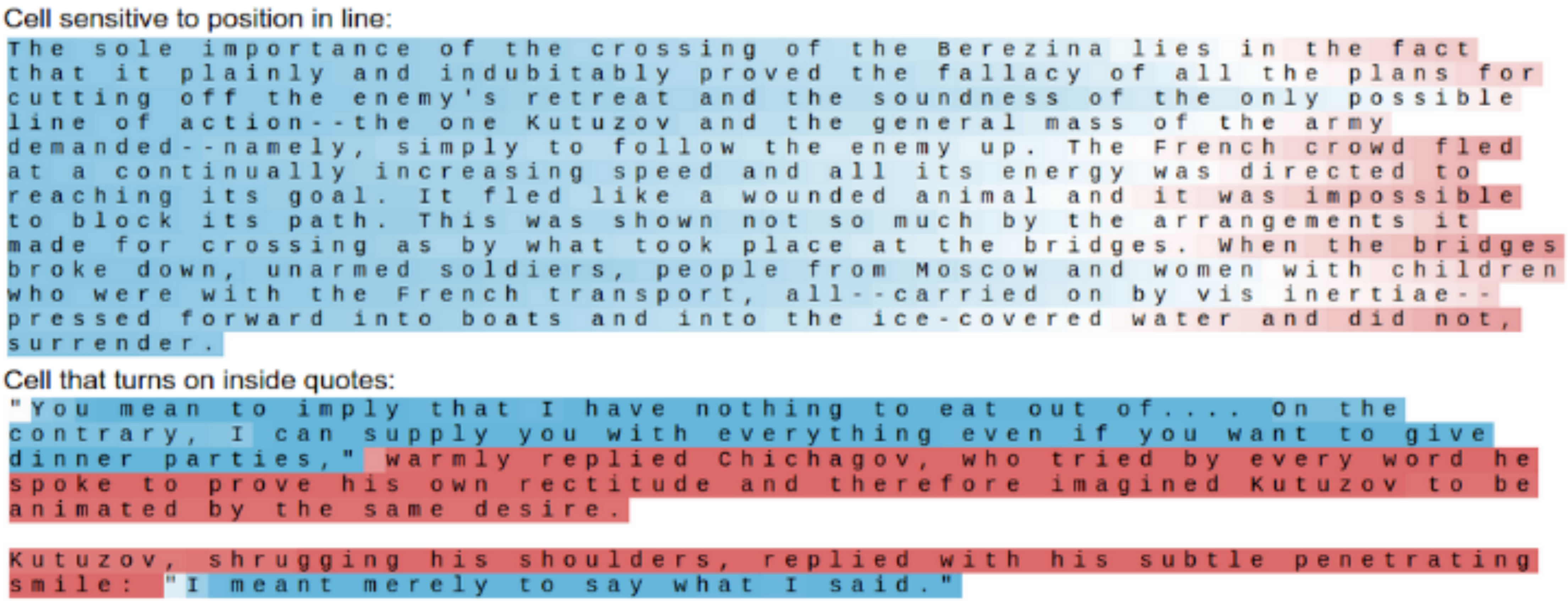
[Andrej Karpathy]

Visualize the Neurons of an RNN

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
   siginfo_t *info)
{
 int sig = next_signal(pending, mask);
 if (sig) {
  if (current->notifier) {
   if (sigismember(current->notifier_mask, sig)) {
    if (!(current->notifier)(current->notifier_data)) {
     clear_thread_flag(TIF_SIGPENDING);
     return 0;
    }
   }
  }
  collect_signal(sig, pending, info);
 }
 return sig;
}
```

[Andrej Karpathy]

Visualize the Neurons of an RNN



[Andrej Karpathy]

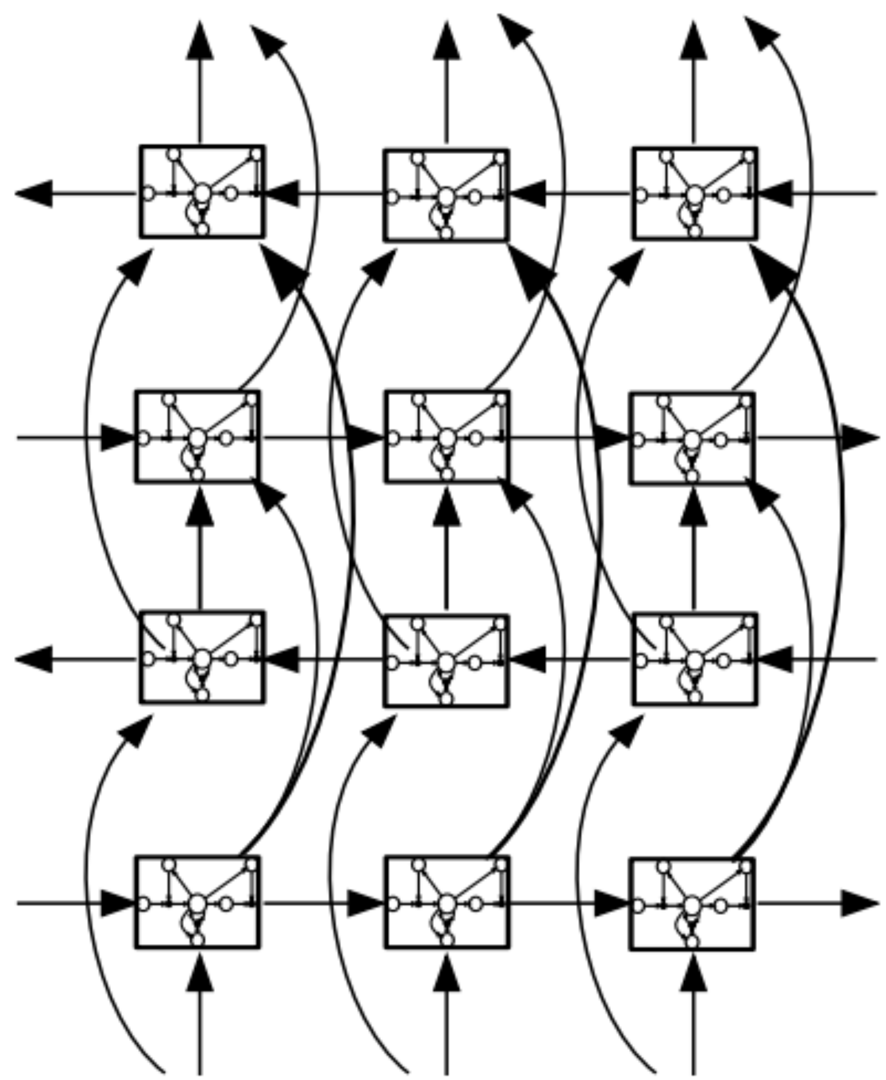
Applications

RNN Applications

- Speech Recognition
- Natural Language Processing
- Action Recognition
- Machine Translation
- Many more to come

Speech Recognition

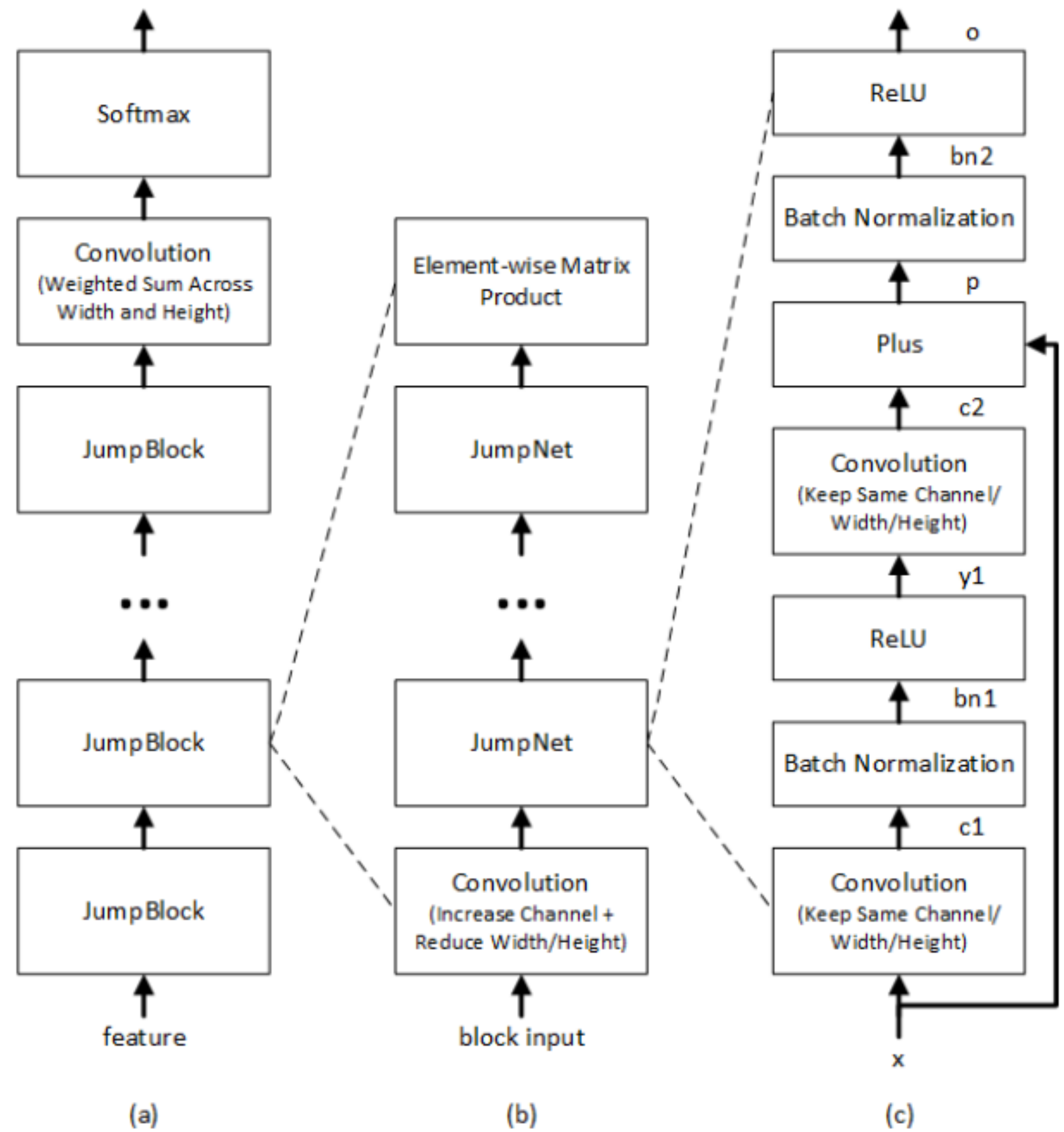
- Deep Bidirectional LSTM



[Alex Graves, Navdeep Jaitly, Abdel-rahman Mohamed]

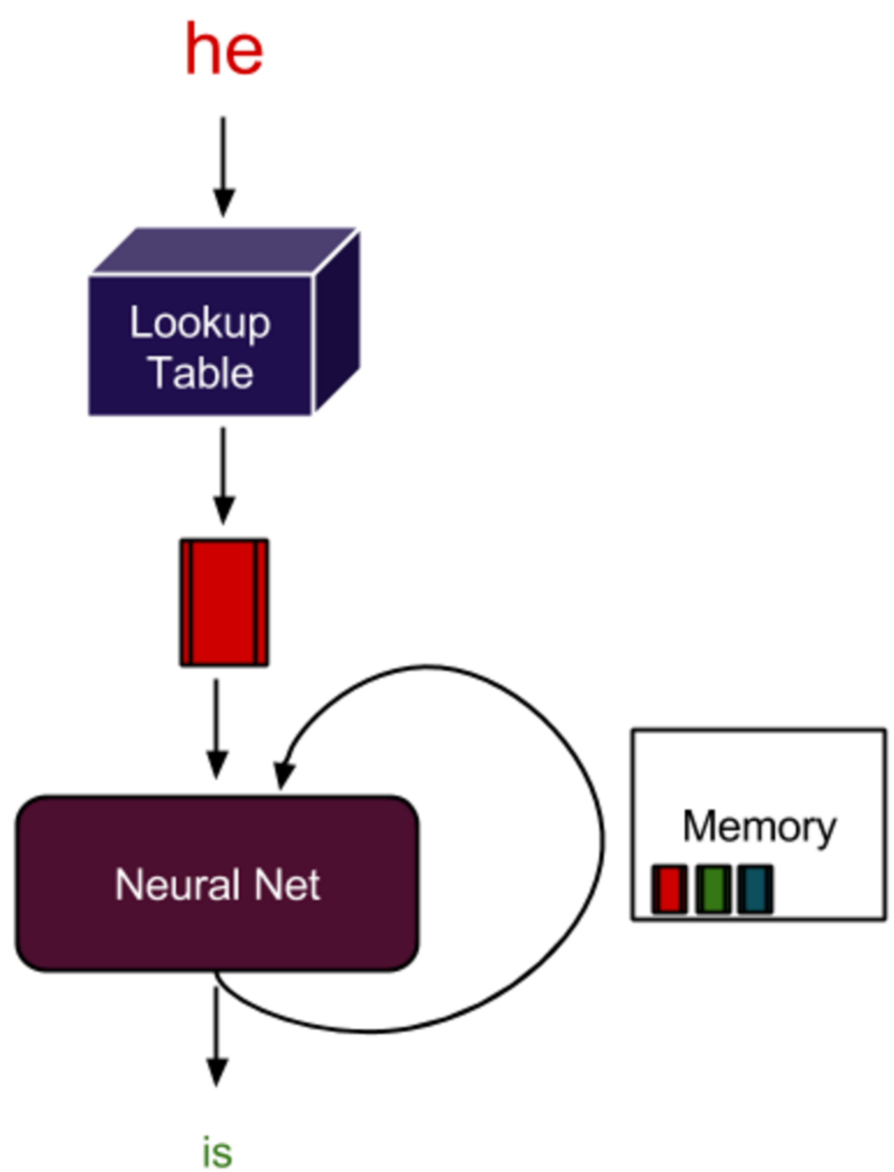
Conversational Speech Recognition

- Achieving human parity



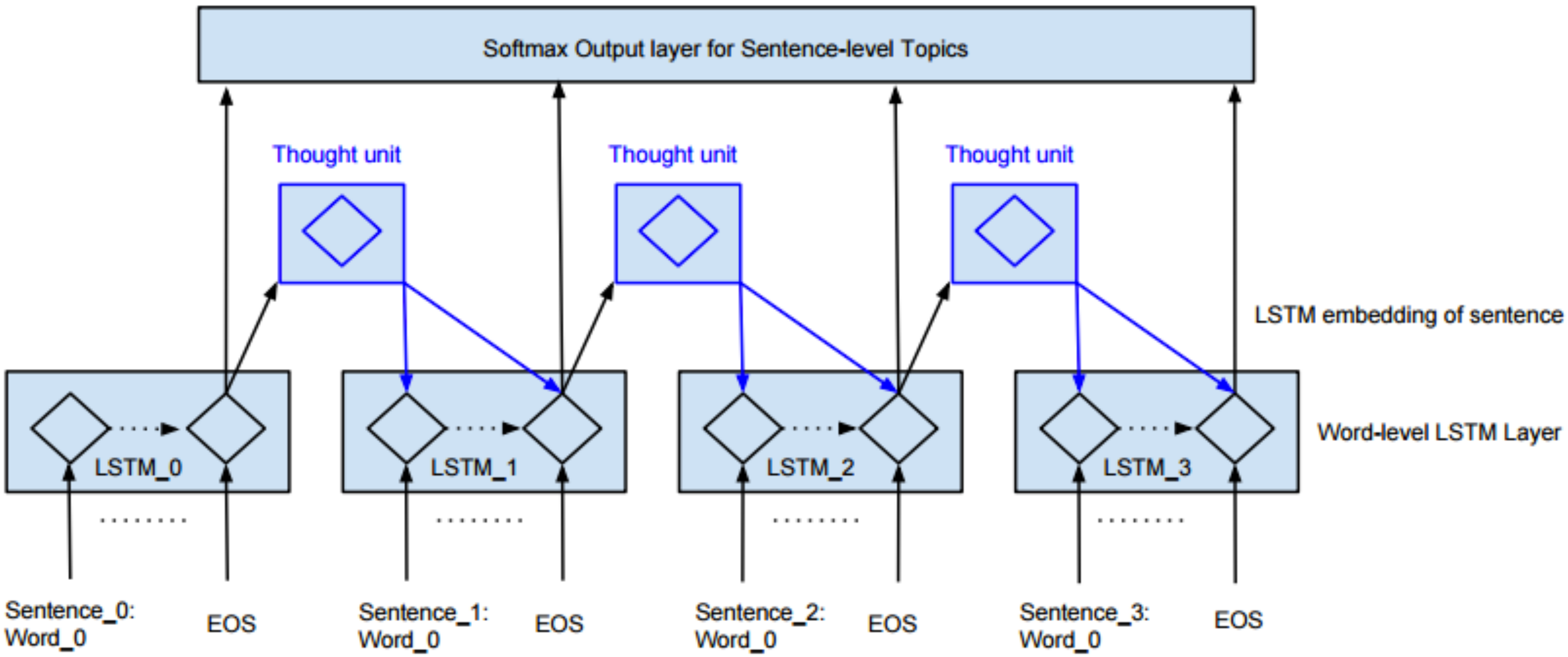
[Xiong et al.]

Natural Language Processing



[Soumith Chantala]

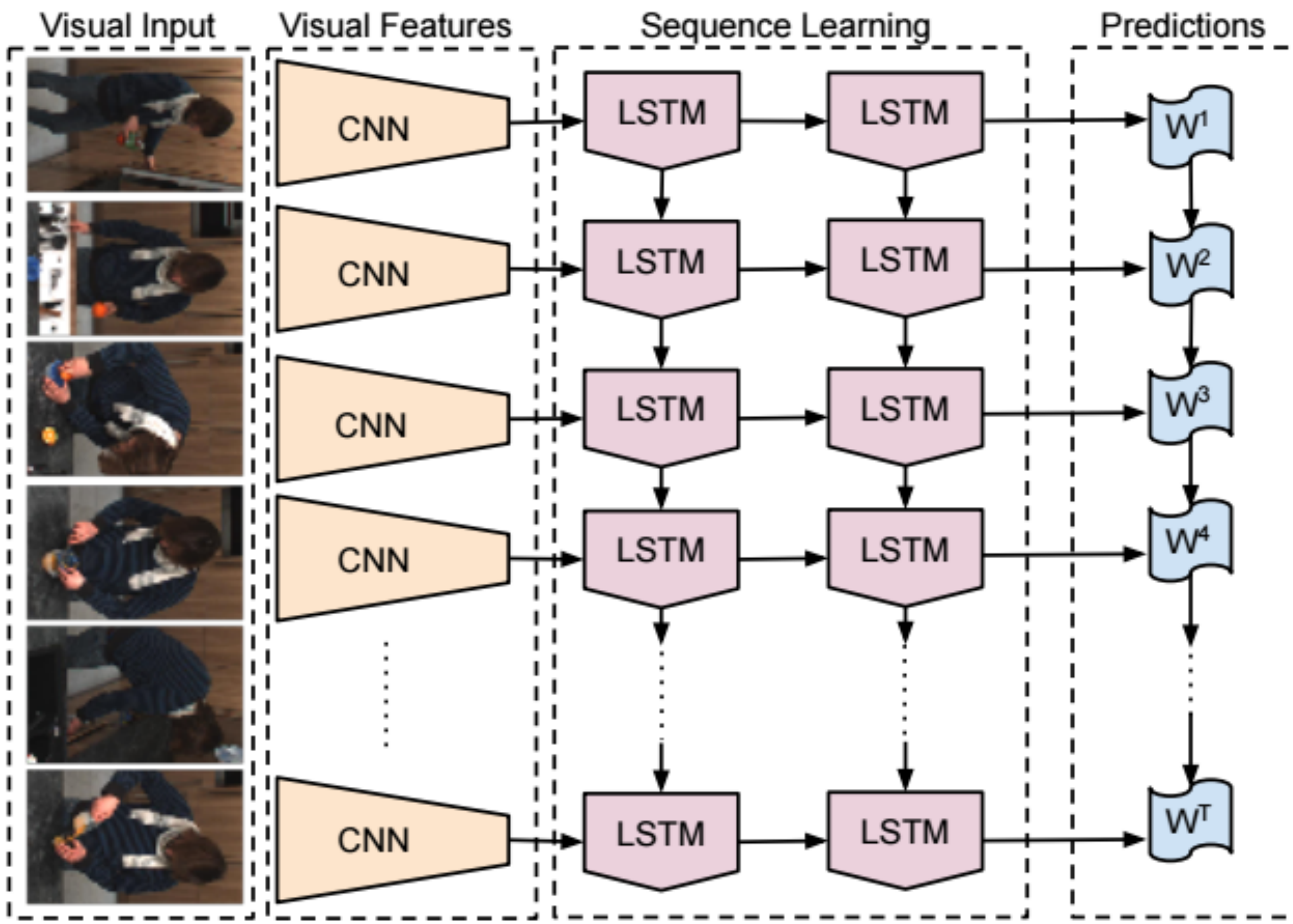
Contextual LSTM for NLP Tasks



[Ghost et al.]

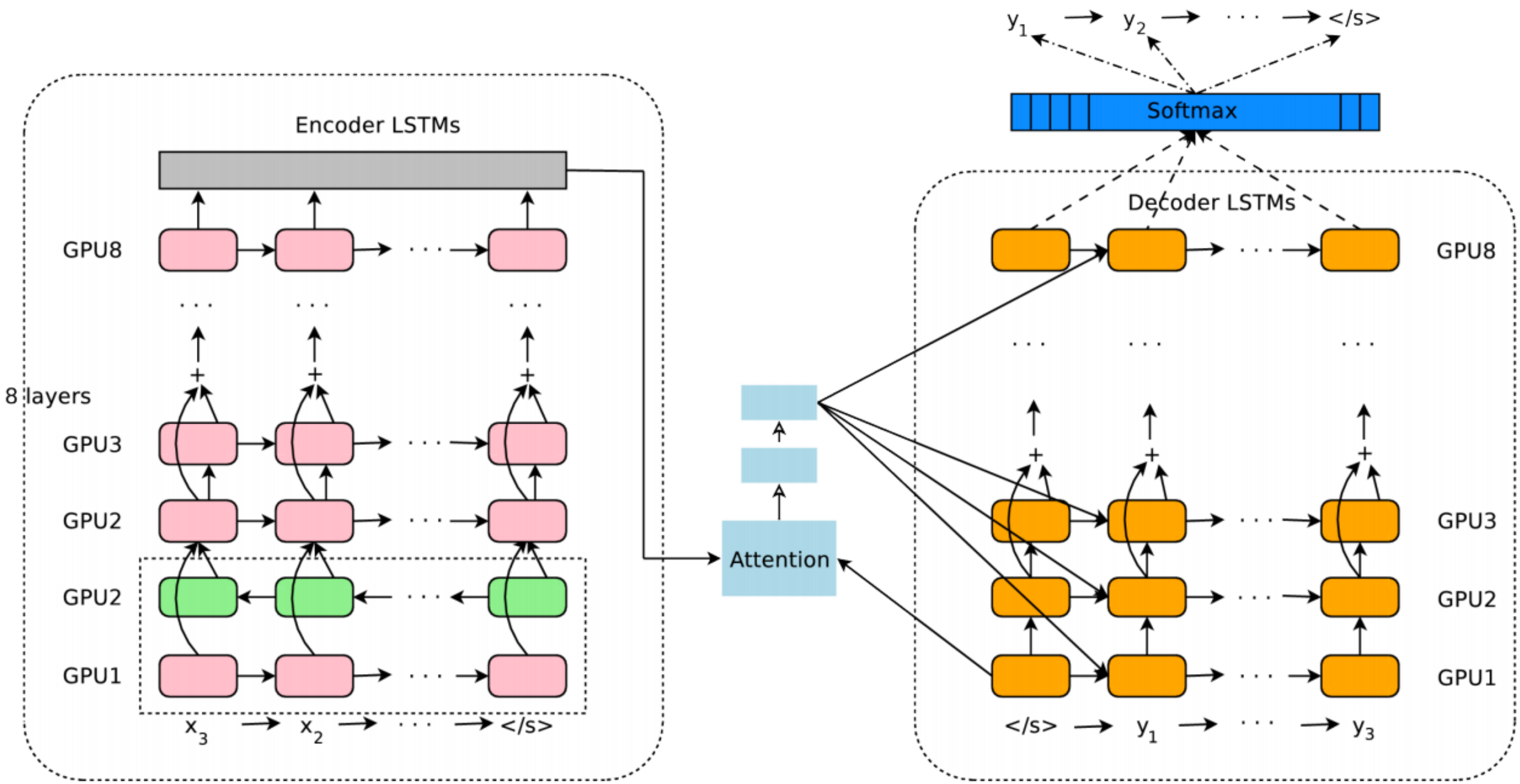
Action Recognition

- Long-term Recurrent Convnet



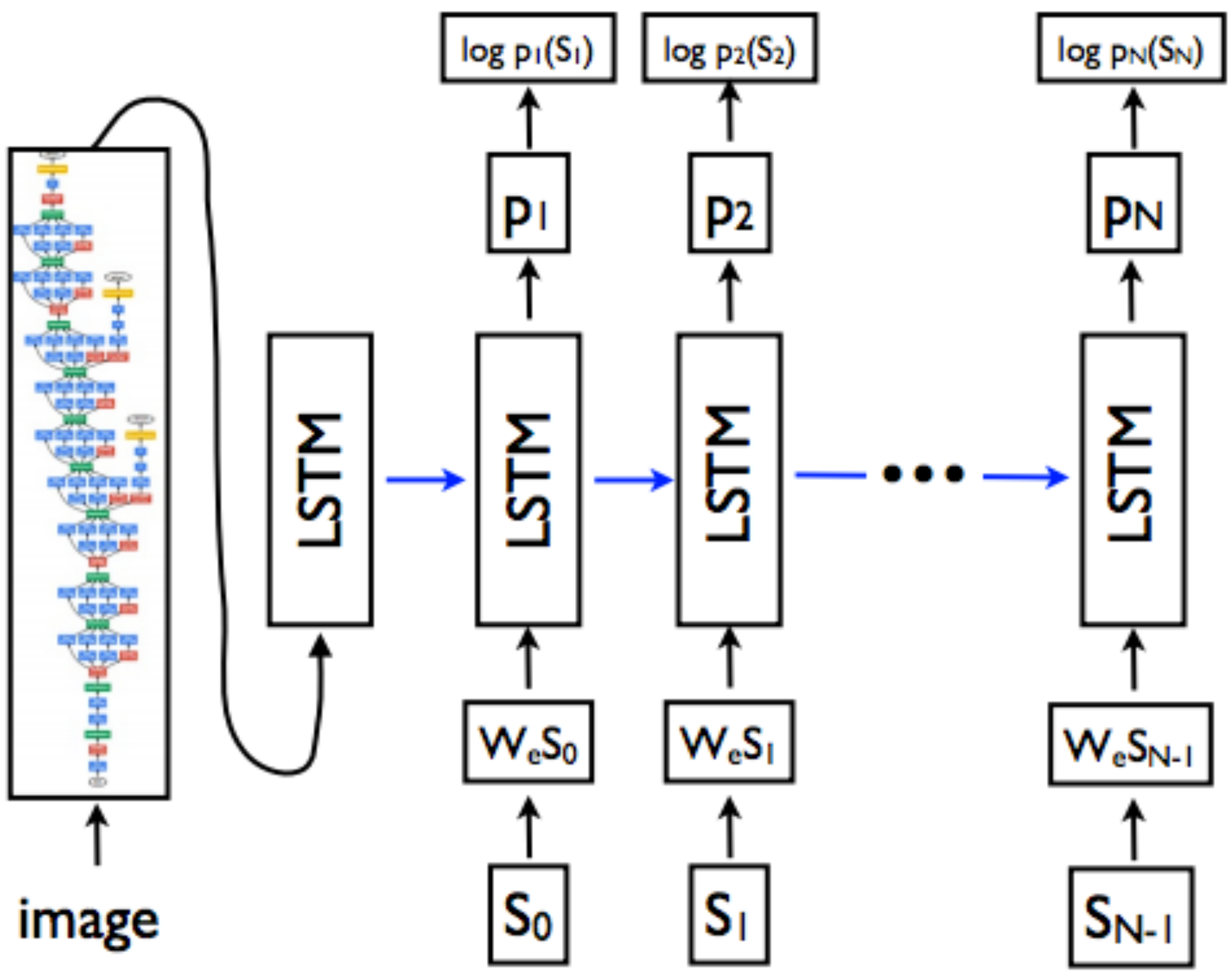
[Donahue et al.]

Google's Neural Machine Translation System



[Yonghui Wu et al.]

Image Captioning

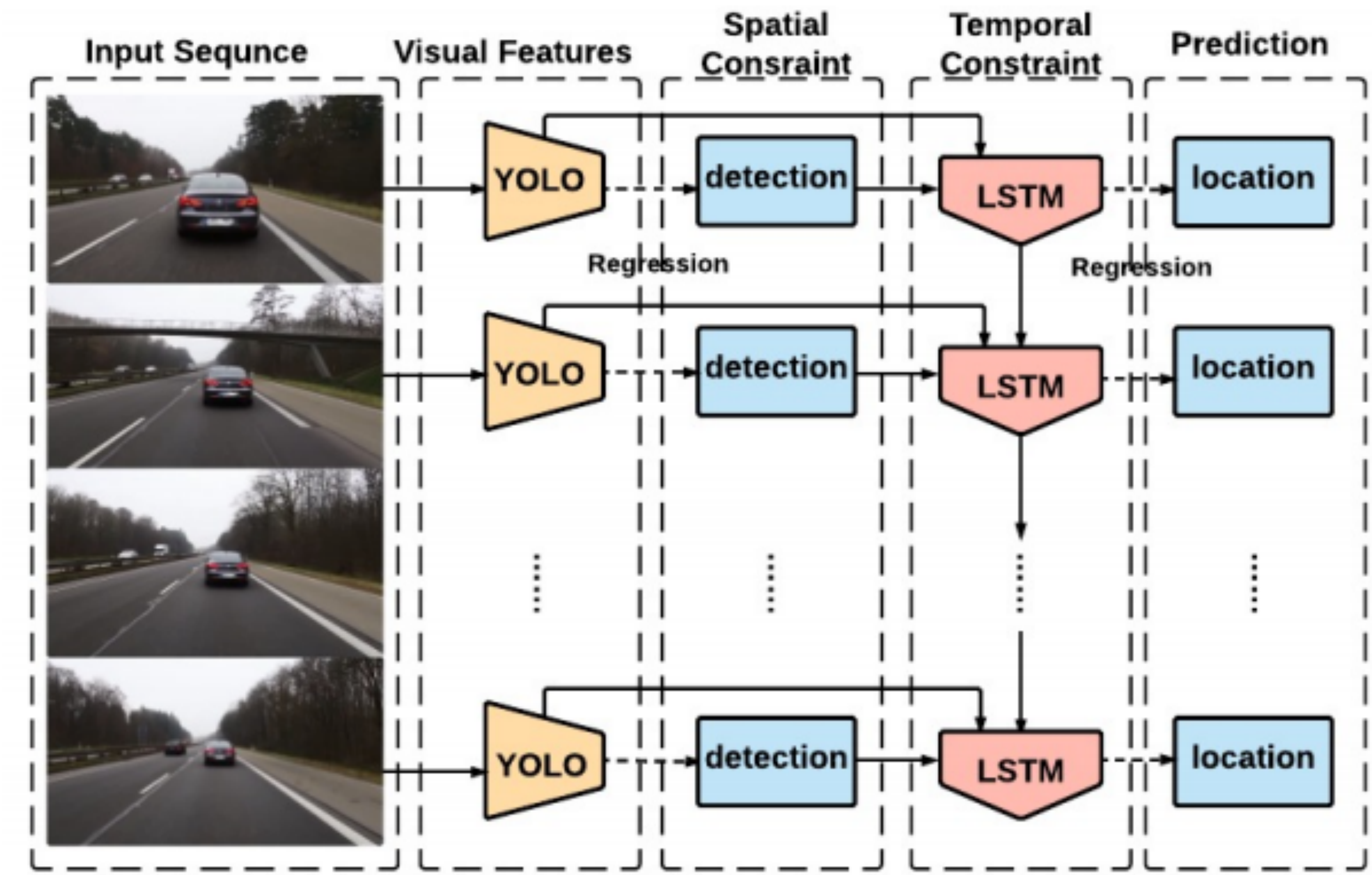


[Vinyals, Toshev, Bengio, Erhan]

Image Captioning

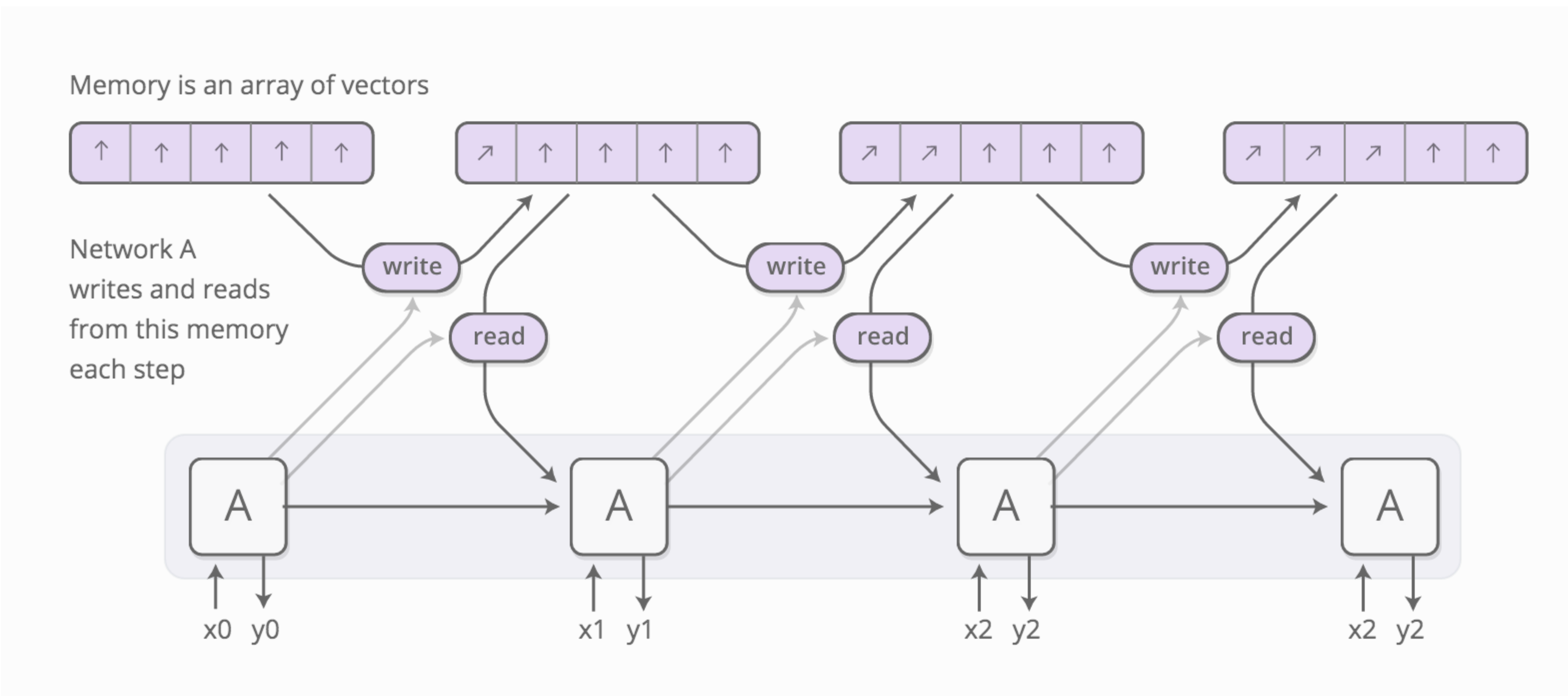
[Vinyals, Toshev, Bengio, Erhan]

Object Tracking



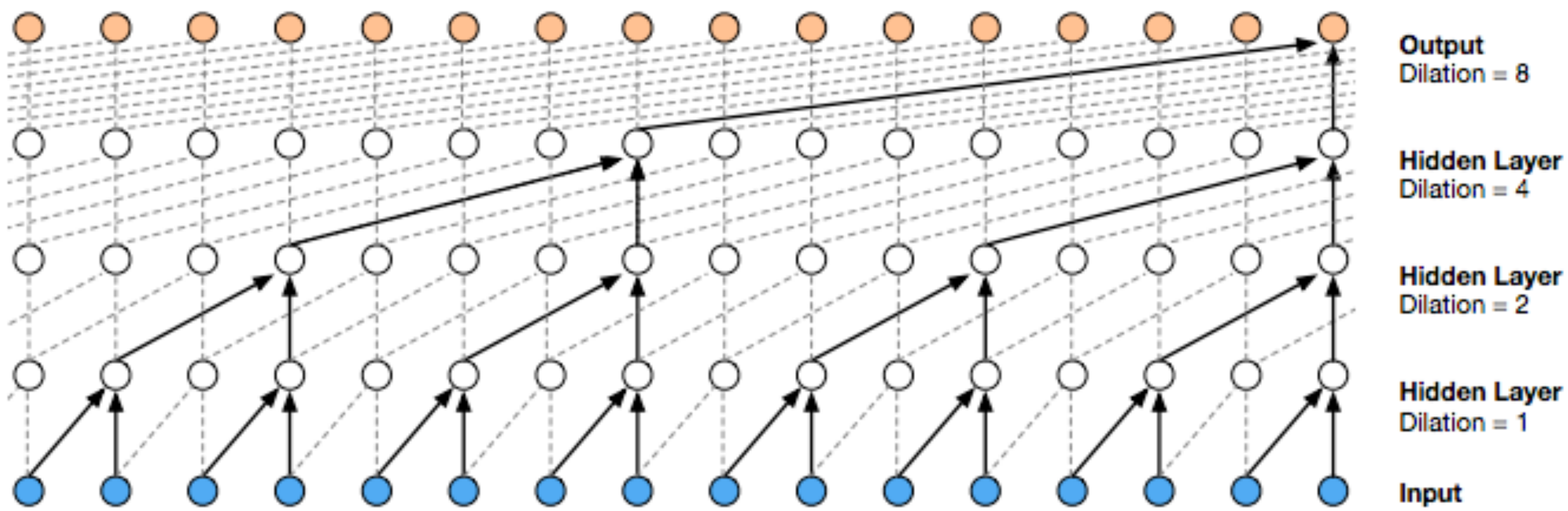
[Ning, Zhang, Huang, He, Wang]

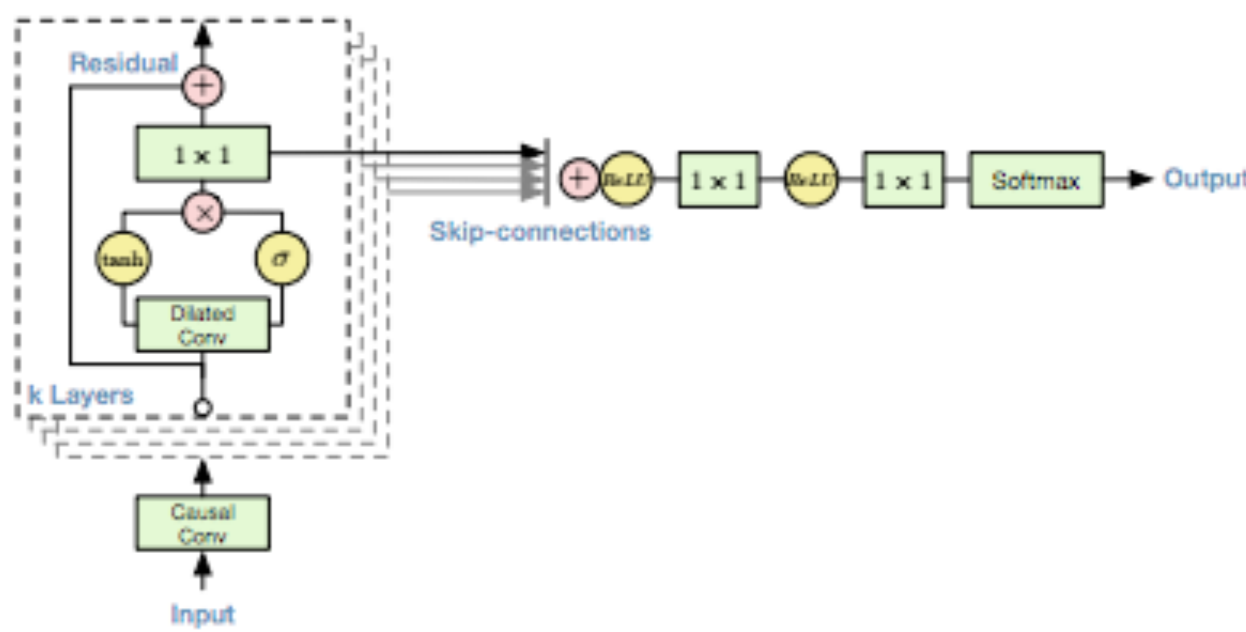
Neural Turing Machines



[Chris Olah]

WaveNet





[van den Oord et al.]

DoomBot

- Doom Competition
  - Facebook won 1st place (F1)
  - https://www.youtube.com/watch?v=94EPSjQH38Y

ODE2RNN: Parameter Estimation for Systems of Ordinary Differential Equations