

*CS 2750: Machine Learning*

# **Recurrent Neural Networks**

Prof. Adriana Kovashka  
University of Pittsburgh  
March 14, 2017

# One Motivation: Descriptive Text for Images



“It was an arresting face, pointed of chin, square of jaw. Her eyes were pale green without a touch of hazel, starred with bristly black lashes and slightly tilted at the ends. Above them, her thick black brows slanted upward, cutting a startling oblique line in her magnolia-white skin—that skin so prized by Southern women and so carefully guarded with bonnets, veils and mittens against hot Georgia suns”

Scarlett O'Hara described in *Gone with the Wind*

# Some pre-RNN good results



This is a picture of one sky, one road and one sheep. The gray sky is over the gray road. The gray sheep is by the gray road.



Here we see one road, one sky and one bicycle. The road is near the blue sky, and near the colorful bicycle. The colorful bicycle is within the blue sky.



This is a picture of two dogs. The first dog is near the second furry dog.



# Some pre-RNN bad results

## Missed detections:



Here we see one potted plant.



This is a picture of one dog.

## False detections:



There are one road and one cat.  
The furry road is in the furry cat.



This is a picture of one tree, one  
road and one person. The rusty  
tree is under the red road. The  
colorful person is near the rusty  
tree, and under the red road.

## Incorrect attributes:



This is a photograph of two sheep and one  
grass. The first black sheep is by the green  
grass, and by the second black sheep. The  
second black sheep is by the green grass.



This is a photograph of two horses and  
one grass. The first feathered horse is  
within the green grass, and by the  
second feathered horse. The second  
feathered horse is within the green  
grass

# Results with Recurrent Neural Networks



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



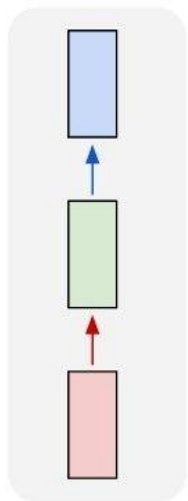
"two young girls are playing with lego toy."



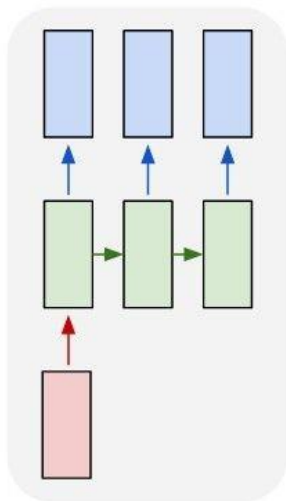
"boy is doing backflip on wakeboard."

# Recurrent Networks offer a lot of flexibility:

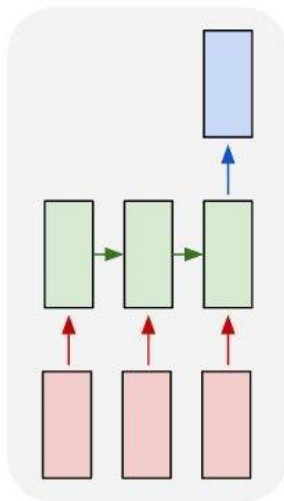
one to one



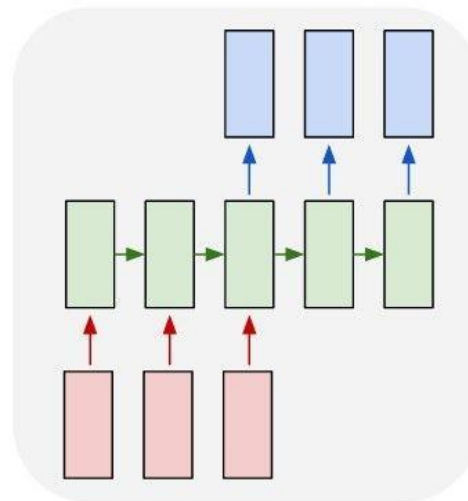
one to many



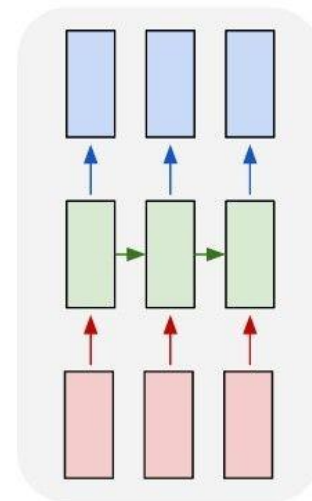
many to one



many to many



many to many

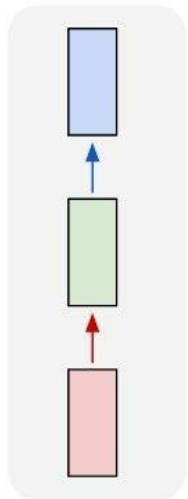


vanilla neural networks

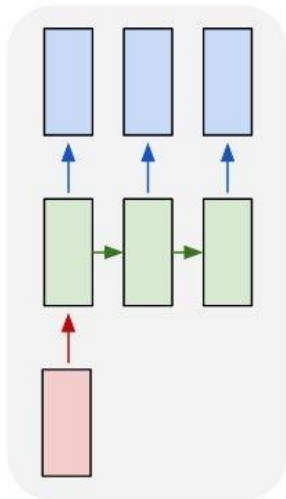


# Recurrent Networks offer a lot of flexibility:

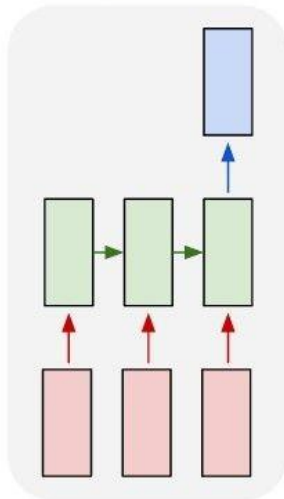
one to one



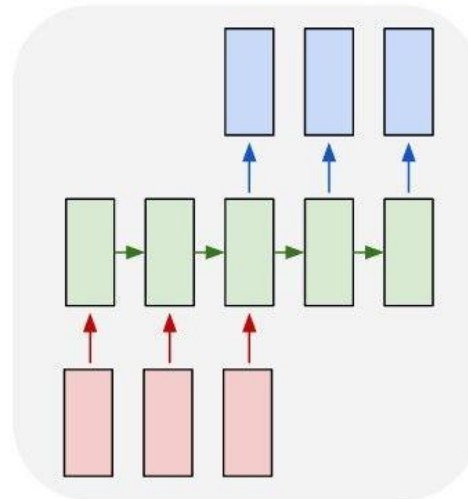
one to many



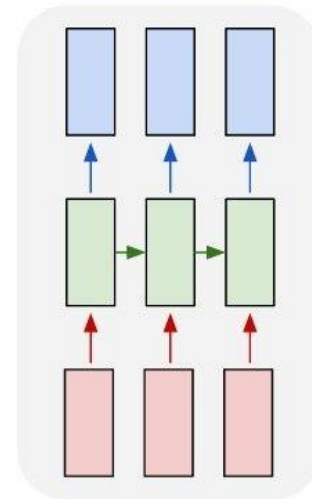
many to one



many to many



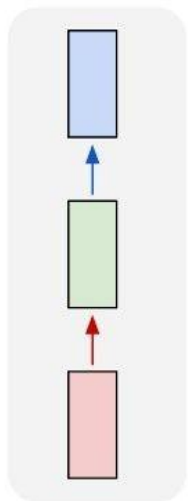
many to many



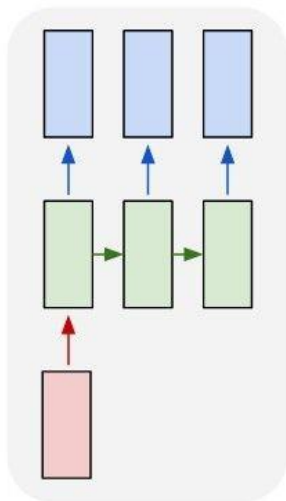
↖ e.g. **image captioning**  
image -> sequence of words

# Recurrent Networks offer a lot of flexibility:

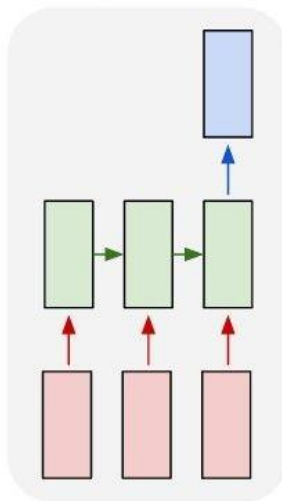
one to one



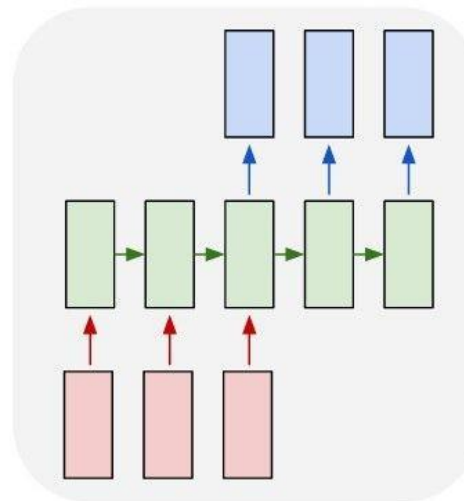
one to many



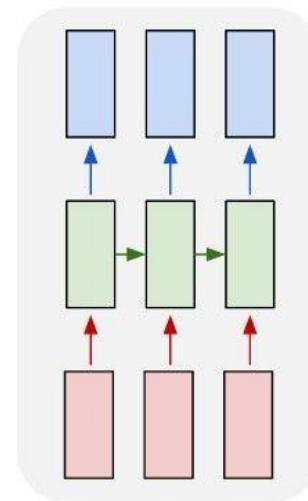
many to one



many to many



many to many

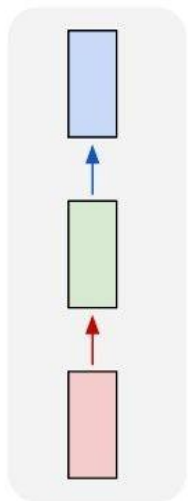


e.g. **sentiment classification**  
sequence of words -> sentiment

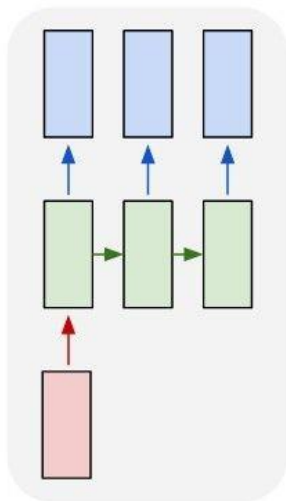


# Recurrent Networks offer a lot of flexibility:

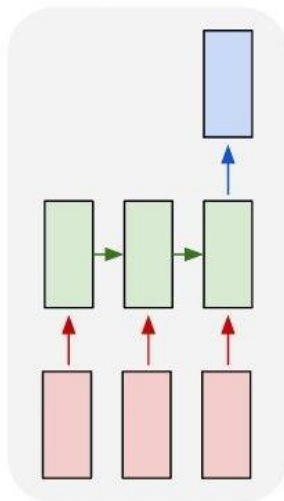
one to one



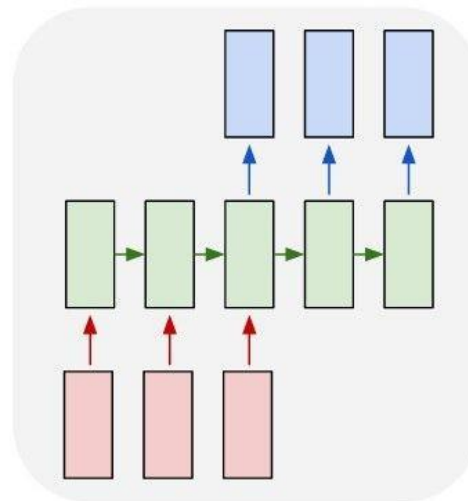
one to many



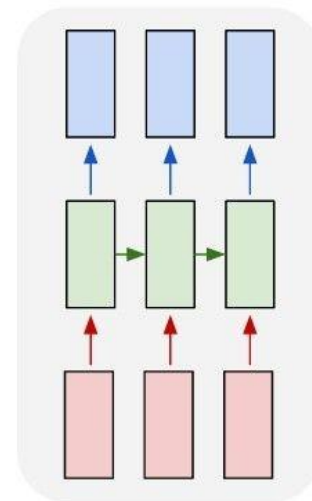
many to one



many to many



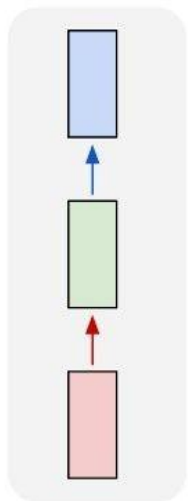
many to many



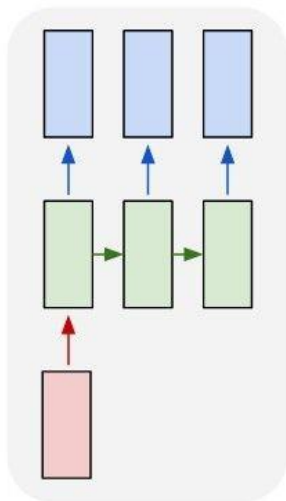
↖ e.g. **machine translation**  
seq of words -> seq of words

# Recurrent Networks offer a lot of flexibility:

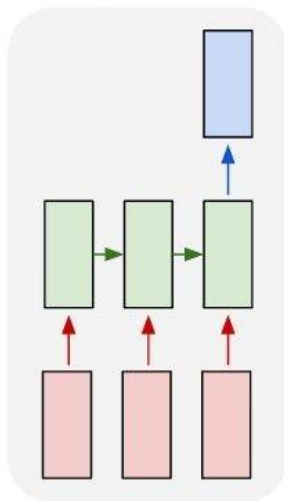
one to one



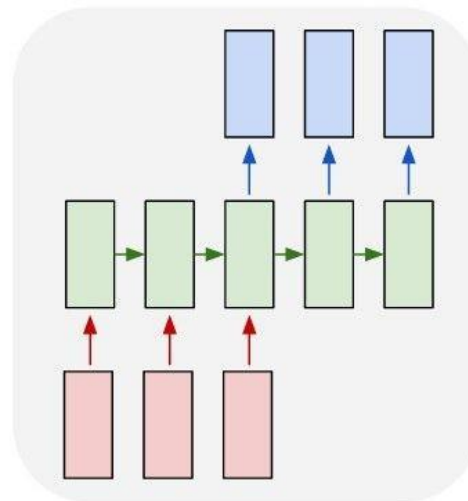
one to many



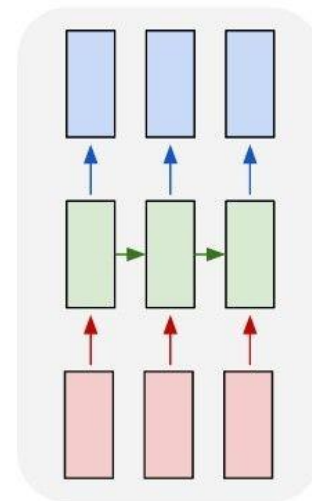
many to one



many to many



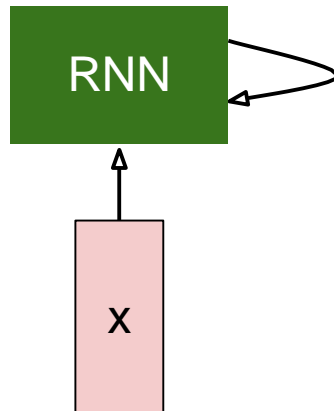
many to many



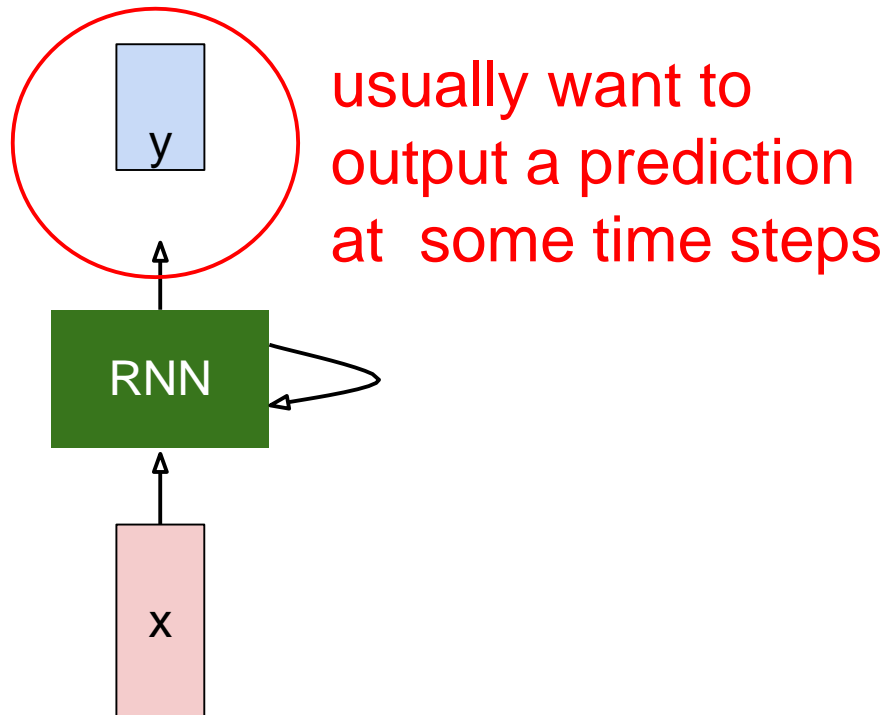
e.g. video classification on frame level



# Recurrent Neural Network



# Recurrent Neural Network





# Recurrent Neural Network

We can process a sequence of vectors  $\mathbf{x}$  by applying a recurrence formula at every time step:

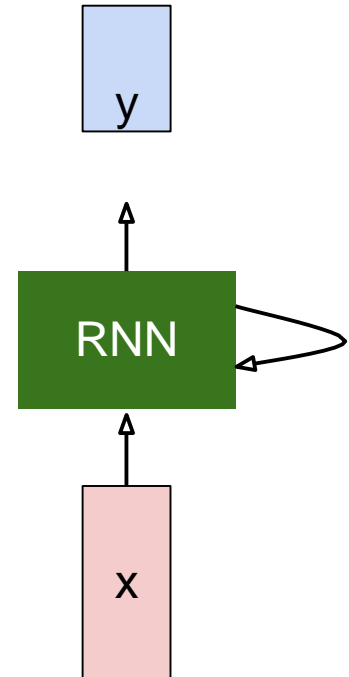
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function with parameters  $W$

old state

input vector at some time step

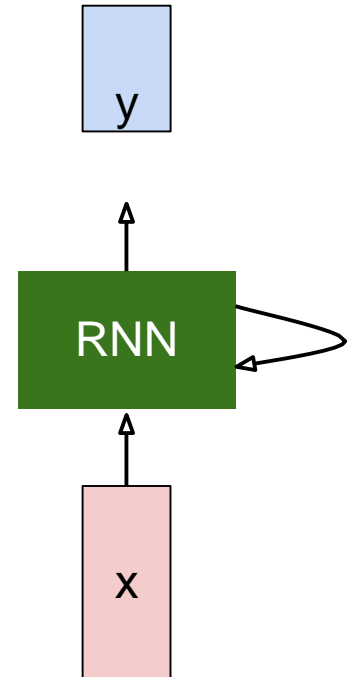


# Recurrent Neural Network

We can process a sequence of vectors  $\mathbf{x}$  by applying a recurrence formula at every time step:

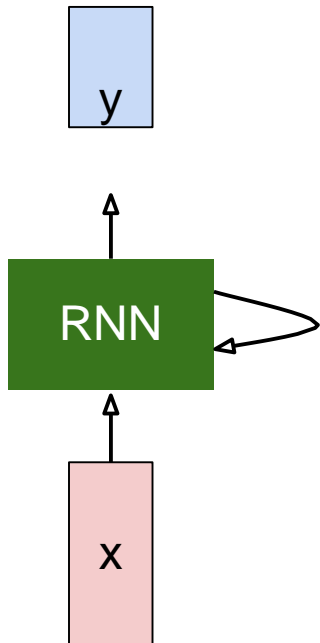
$$h_t = f_W(h_{t-1}, x_t)$$

Notice: the same function and the same set of parameters are used at every time step.



# (Vanilla) Recurrent Neural Network

The state consists of a single “*hidden*” vector  $h$ :



$$h_t = f_W(h_{t-1}, x_t)$$



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

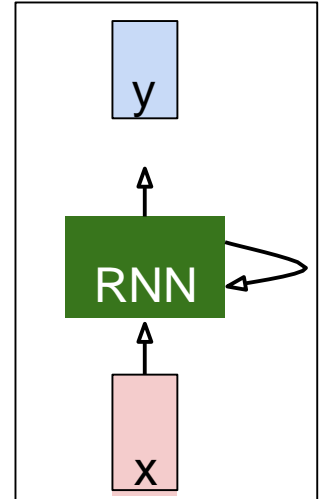
$$y_t = W_{hy}h_t$$

# Example

## Character-level language model example

Vocabulary:  
[h,e,l,o]

Example training  
sequence:  
**“hello”**



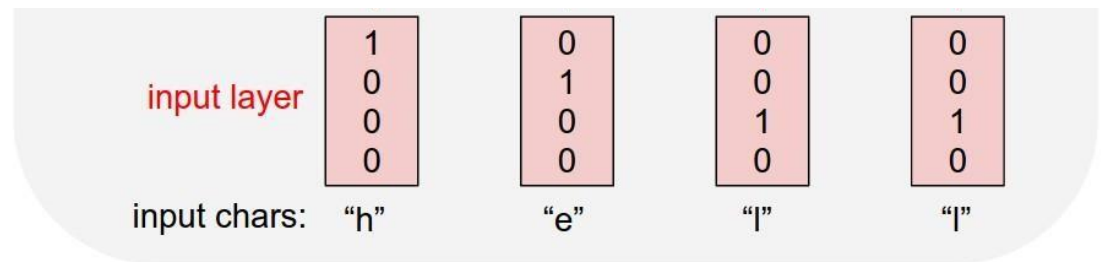


# Example

## Character-level language model example

Vocabulary:  
[h,e,l,o]

Example training  
sequence:  
“hello”



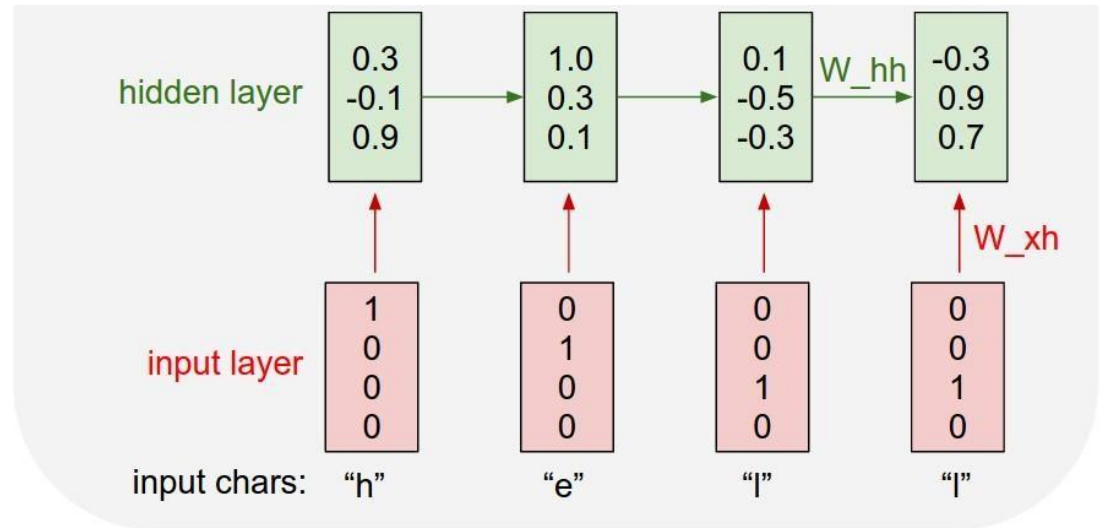
# Example

## Character-level language model example

Vocabulary:  
[h,e,l,o]

Example training sequence:  
“hello”

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

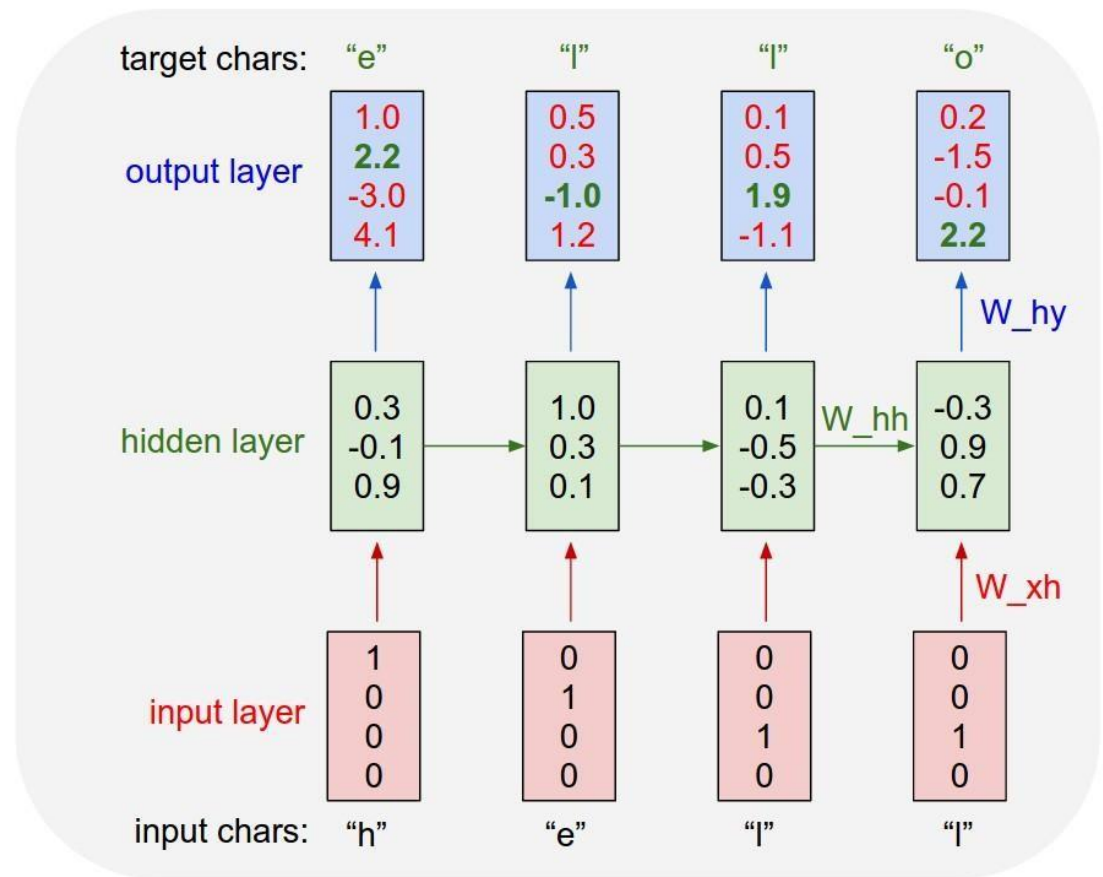


# Example

## Character-level language model example

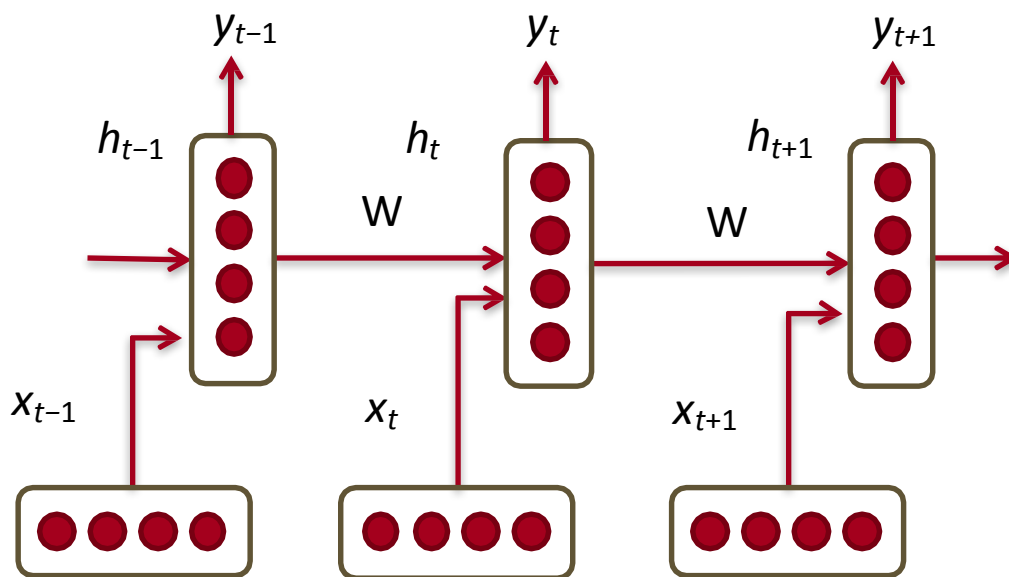
Vocabulary:  
[h,e,l,o]

Example training sequence:  
“hello”



# Recurrent Neural Networks

- RNNs tie the weights at each time step
- Condition the neural network on all previous words





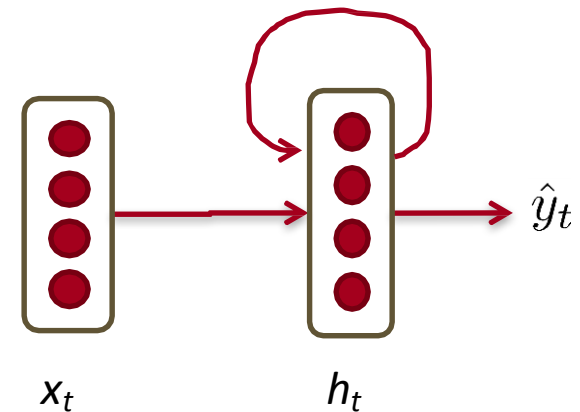
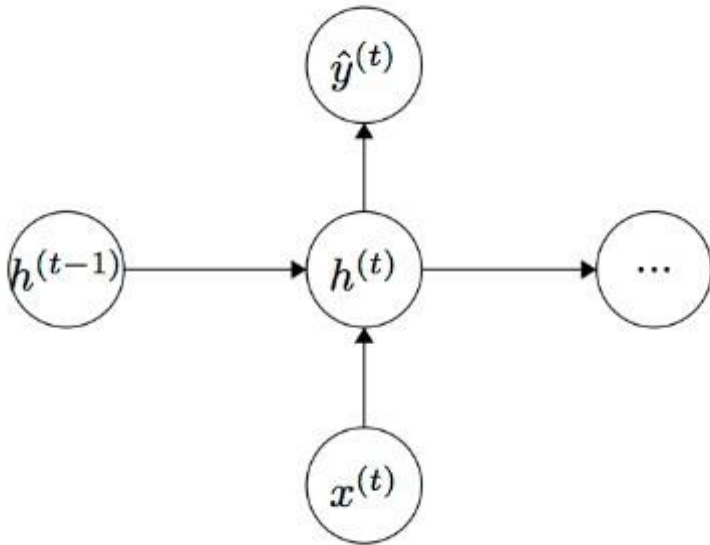
# Recurrent Neural Network language model

Given list of word **vectors**:  $x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_T$

At a single time step: 
$$h_t = \sigma \left( W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]} \right)$$

$$\hat{y}_t = \text{softmax} \left( W^{(S)} h_t \right)$$

$$\hat{P}(x_{t+1} = v_j \mid x_t, \dots, x_1) = \hat{y}_{t,j}$$



# Recurrent Neural Network language model

Given list of word **vectors**:  $x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_T$

At a single time step:  $h_t = \sigma \left( W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]} \right)$

$$\hat{y}_t = \text{softmax} \left( W^{(S)} h_t \right)$$

$$\hat{P}(x_{t+1} = v_j \mid x_t, \dots, x_1) = \hat{y}_{t,j}$$

We use the same set of  $W$  weights at all time steps!

$h_0 \in \mathbb{R}^{D_h}$  is some initialization vector for the hidden layer at time step 0

$x_{[t]}$  is the column vector at time step  $t$

$$W^{(hh)} \in \mathbb{R}^{D_h \times D_h} \quad W^{(hx)} \in \mathbb{R}^{D_h \times d} \quad W^{(S)} \in \mathbb{R}^{|V| \times D_h}$$

# Recurrent Neural Network language model

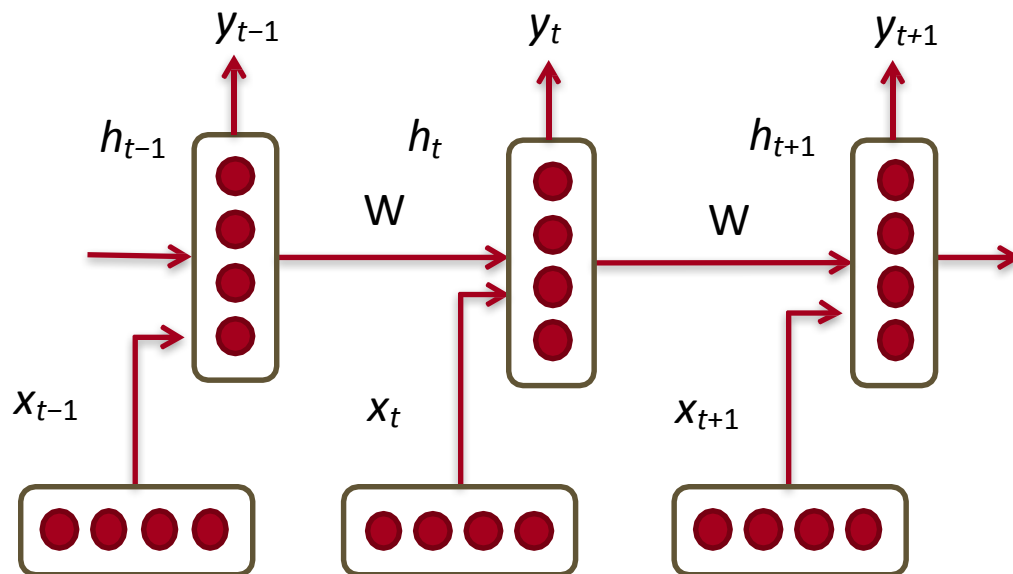
$\hat{y} \in \mathbb{R}^{|V|}$  is a probability distribution over the vocabulary

Same cross entropy loss function but predicting words instead of classes, over dataset of size (number of words)  $T$

$$J = -\frac{1}{T} \sum_{t=1}^T \sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

# Training RNNs is hard

- Multiply the same matrix at each time step during forward prop



- Ideally inputs from many time steps ago can modify output  $y$



# The vanishing gradient problem

- Similar but simpler RNN formulation:

$$\begin{aligned}h_t &= W f(h_{t-1}) + W^{(hx)} x_{[t]} \\ \hat{y}_t &= W^{(S)} f(h_t)\end{aligned}$$

- Total error is the sum of each error at time steps  $t$

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W}$$

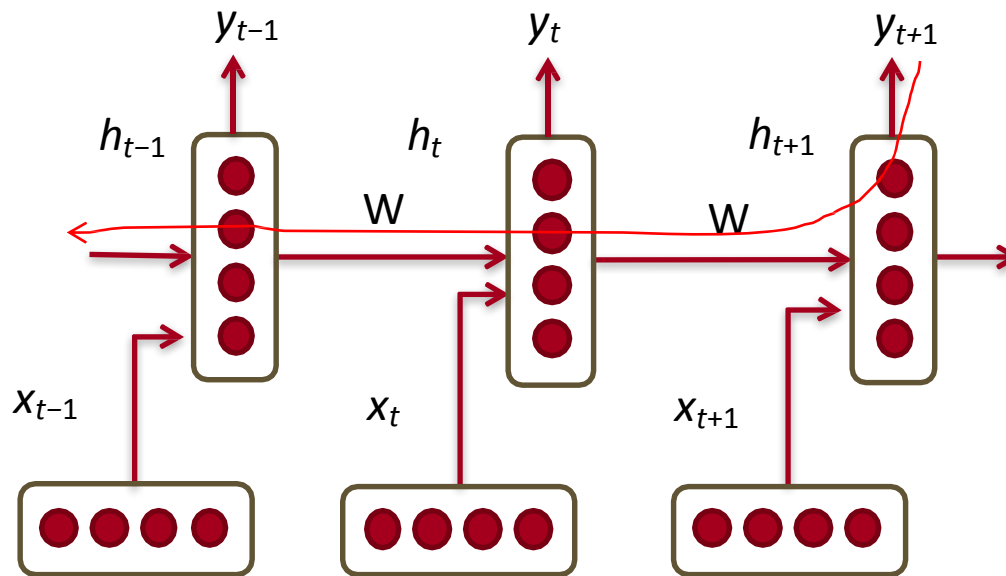
- Chain rule application:

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^t \frac{\partial h_j}{\partial h_{j-1}}$$

# Why is the vanishing gradient a problem?

- The error at a time step ideally can tell a previous time step from many steps away to change during backprop



# The vanishing gradient problem for language models

- In the case of language modeling or question answering words from time steps far away are not taken into consideration when training to predict the next word
- Example:

Jane walked into the room. John walked in too. It was late in the day. Jane said hi to \_\_\_\_\_

# Gated Recurrent Units (GRUs)

- More complex hidden unit computation in recurrence!
- Introduced by Cho et al. 2014
- Main ideas:
  - keep around memories to capture long distance dependencies
  - allow error messages to flow at different strengths depending on the inputs

# Gated Recurrent Units (GRUs)

- Standard RNN computes hidden layer at next time step directly:

$$h_t = f \left( W^{(hh)} h_{t-1} + W^{(hx)} x_t \right)$$

- GRU first computes an update **gate** (another layer) based on current input word vector and hidden state

$$z_t = \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} \right)$$

- Compute reset gate similarly but with different weights

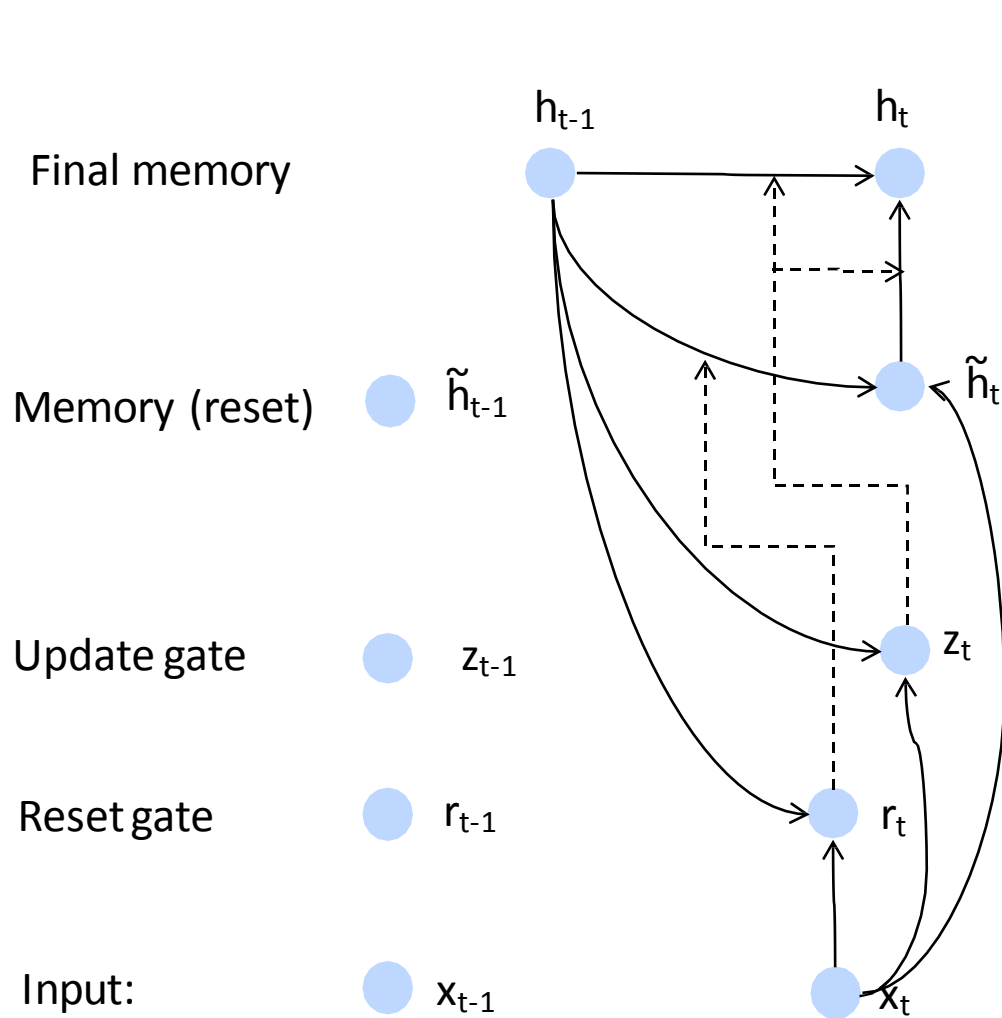
$$r_t = \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} \right)$$

# Gated Recurrent Units (GRUs)

- Update gate  $z_t = \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} \right)$
- Reset gate  $r_t = \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} \right)$
- New memory content:  $\tilde{h}_t = \tanh (W x_t + r_t \circ U h_{t-1})$   
If reset gate unit is  $\sim 0$ , then this ignores previous memory and only stores the new word information
- Final memory at time step combines current and previous time steps:  $h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$



# Gated Recurrent Units (GRUs)



$$z_t = \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} \right)$$

$$r_t = \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} \right)$$

$$\tilde{h}_t = \tanh \left( W x_t + r_t \circ U h_{t-1} \right)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

# Gated Recurrent Units (GRUs)

- If reset is close to 0, ignore previous hidden state: Allows model to drop information that is irrelevant in the future

$$\begin{aligned}z_t &= \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} \right) \\r_t &= \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} \right) \\\tilde{h}_t &= \tanh (W x_t + r_t \circ U h_{t-1}) \\h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t\end{aligned}$$

- Update gate  $z$  controls how much of past state should matter now.
  - If  $z$  close to 1, then we can copy information in that unit through many time steps! **Less vanishing gradient!**
- Units with short-term dependencies often have reset gates very active

# Gated Recurrent Units (GRUs)

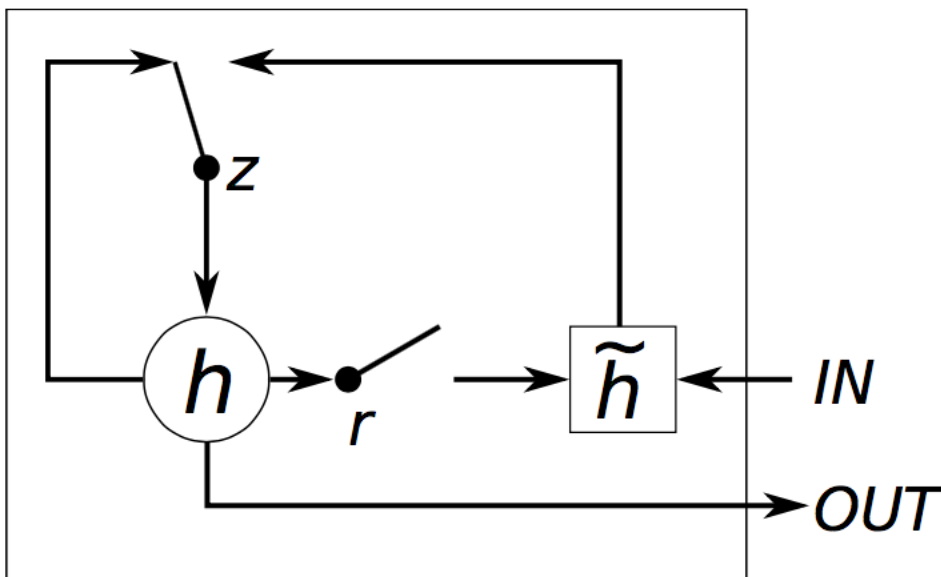
- Units with long term dependencies have active update gates  $z$
- Illustration:

$$z_t = \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} \right)$$

$$r_t = \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} \right)$$

$$\tilde{h}_t = \tanh (W x_t + r_t \circ U h_{t-1})$$

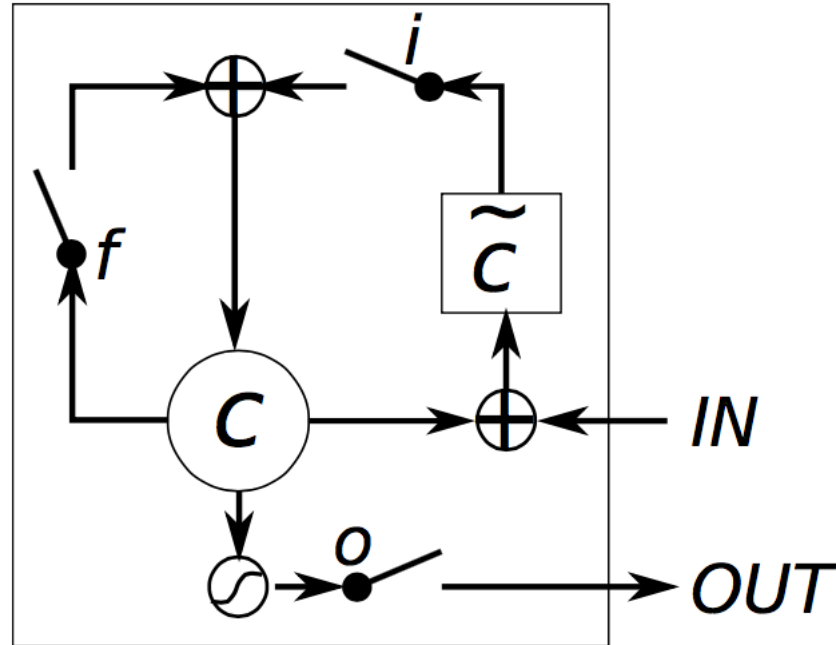
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$



# Long-short-term-memories (LSTMs)

- Proposed by Hochreiter and Schmidhuber in 1997
- We can make the units even more complex
- Allow each time step to modify
  - Input gate (current cell matters)  $i_t = \sigma \left( W^{(i)} x_t + U^{(i)} h_{t-1} \right)$
  - Forget (gate 0, forget past)  $f_t = \sigma \left( W^{(f)} x_t + U^{(f)} h_{t-1} \right)$
  - Output (how much cell is exposed)  $o_t = \sigma \left( W^{(o)} x_t + U^{(o)} h_{t-1} \right)$
  - New memory cell  $\tilde{c}_t = \tanh \left( W^{(c)} x_t + U^{(c)} h_{t-1} \right)$
- Final memory cell:  $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$
- Final hidden state:  $h_t = o_t \circ \tanh(c_t)$

# Long-short-term-memories (LSTMs)



Intuition: memory cells can keep information intact, unless inputs makes them forget it or overwrite it with new input

Cell can decide to output this information or just store it

# LSTMs are currently very hip!

- En vogue default model for most sequence labeling tasks
- Very powerful, especially when stacked and made even deeper (each hidden layer is already computed by a deep internal network)
- Most useful if you have lots and lots of data

# Deep LSTMs compared to traditional systems

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	34.81

Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT'14 result [9]	37.0
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	36.5
Oracle Rescoring of the Baseline 1000-best lists	~45

Sequence to Sequence Learning by Sutskever et al. 2014

# Generating poetry with RNNs

## Sonnet 116 – Let me not ...

*by William Shakespeare*

Let me not to the marriage of true minds  
Admit impediments. Love is not love  
Which alters when it alteration finds,  
Or bends with the remover to remove:  
O no! it is an ever-fixed mark  
That looks on tempests and is never shaken;  
It is the star to every wandering bark,  
Whose worth's unknown, although his height be taken.  
Love's not Time's fool, though rosy lips and cheeks  
Within his bending sickle's compass come:  
Love alters not with his brief hours and weeks,  
But bears it out even to the edge of doom.  
If this be error and upon me proved,  
I never writ, nor no man ever loved.



# Generating poetry with RNNs

at first:

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

↓ train more

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

↓ train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of  
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort  
how, and Gogition is so overelical and ofter.

↓ train more

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

More info: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

# Generating poetry with RNNs

PANDARUS:

Alas, I think he shall be come approached and the day  
When little strain would be attain'd into being never fed,  
And who is but a chain and subjects of his death,  
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,  
Breaking and strongly should be buried, when I perish  
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and  
my fair nudes begun out of the fact, to be conveyed,  
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it.

VIOLA:

Why, Salisbury must find his flesh and thought  
That which I am not apt, not a man and in fire,  
To show the reining of the raven and the wars  
To grace my hand reproach within, and not a fair are hand,  
That Caesar and my goodly father's world;  
When I was heaven of presence and our fleets,  
We spare with hours, but cut thy council I am great,  
Murdered and by thy master's ready there  
My power to give thee but so much as hell:  
Some service in the noble bondman here,  
Would show him to her wine.

KING LEAR:

O, if you were a feeble sight, the courtesy of your law,  
Your sight and several breath, will wear the gods  
With his heads, and my hands are wonder'd at the deeds,  
So drop upon your lordship's head, and your opinion  
Shall be against your honour.

# Generating textbooks with RNNs











open source textbook on algebraic geometry



The Stacks Project

[home](#) [about](#) [tags explained](#) [tag lookup](#) [browse](#) [search](#) [bibliography](#) [recent comments](#) [blog](#) [add slogans](#)

**Browse chapters**

Part	Chapter	online	TeX source	view pdf
Preliminaries				
	1. Introduction	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	2. Conventions	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	3. Set Theory	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	4. Categories	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	5. Topology	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	6. Sheaves on Spaces	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	7. Sites and Sheaves	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	8. Stacks	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	9. Fields	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 
	10. Commutative Algebra	<a href="#">online</a>	<a href="#">tex</a> 	<a href="#">pdf</a> 

**Parts**

- [Preliminaries](#)
- [Schemes](#)
- [Topics in Scheme Theory](#)
- [Algebraic Spaces](#)
- [Topics in Geometry](#)
- [Deformation Theory](#)
- [Algebraic Stacks](#)
- [Miscellany](#)

**Statistics**

The Stacks project now consists of

- 455910 lines of code
- 14221 tags (56 inactive tags)
- 2366 sections

Latex source



# Generating textbooks with RNNs

For  $\bigoplus_{n=1,\dots,m}$  where  $\mathcal{L}_{m*} = 0$ , hence we can find a closed subset  $\mathcal{H}$  in  $\mathcal{H}$  and any sets  $\mathcal{F}$  on  $X$ ,  $U$  is a closed immersion of  $S$ , then  $U \rightarrow T$  is a separated algebraic space.

*Proof.* Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by  $\coprod Z \times_U U \rightarrow V$ . Consider the maps  $M$  along the set of points  $\text{Sch}_{fppf}$  and  $U \rightarrow U$  is the fibre category of  $S$  in  $U$  in Section, ?? and the fact that any  $U$  affine, see Morphisms, Lemma ??. Hence we obtain a scheme  $S$  and any open subset  $W \subset U$  in  $\text{Sh}(G)$  such that  $\text{Spec}(R') \rightarrow S$  is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that  $f_i$  is of finite presentation over  $S$ . We claim that  $\mathcal{O}_{X,x}$  is a scheme where  $x, x', s'' \in S'$  such that  $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}_{X',x''}$  is separated. By Algebra, Lemma ?? we can define a map of complexes  $\text{GL}_{S'}(x'/S'')$  and we win.  $\square$

To prove study we see that  $\mathcal{F}|_U$  is a covering of  $\mathcal{X}'$ , and  $T_i$  is an object of  $\mathcal{F}_{X/S}$  for  $i > 0$  and  $\mathcal{F}_p$  exists and let  $\mathcal{F}_i$  be a presheaf of  $\mathcal{O}_X$ -modules on  $\mathcal{C}$  as a  $\mathcal{F}$ -module. In particular  $\mathcal{F} = U/\mathcal{F}$  we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{opp}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \mapsto (U, \text{Spec}(A))$$

is an open subset of  $X$ . Thus  $U$  is affine. This is a continuous map of  $X$  is the inverse, the groupoid scheme  $S$ .

*Proof.* See discussion of sheaves of sets.  $\square$

The result for prove any open covering follows from the less of Example ??. It may replace  $S$  by  $X_{spaces, \acute{e}tale}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{Zar}$ , see Descent, Lemma ??. Namely, by Lemma ?? we see that  $R$  is geometrically regular over  $S$ .

**Lemma 0.1.** Assume (3) and (3) by the construction in the description.

Suppose  $X = \lim |X|$  (by the formal open covering  $X$  and a single map  $\text{Proj}_X(\mathcal{A}) = \text{Spec}(B)$  over  $U$  compatible with the complex

$$\text{Set}(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X, \mathcal{O}_X}).$$

When in this case of to show that  $\mathcal{Q} \rightarrow \mathcal{C}_{Z/X}$  is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If  $T$  is surjective we may assume that  $T$  is connected with residue fields of  $S$ . Moreover there exists a closed subspace  $Z \subset X$  of  $X$  where  $U$  in  $X'$  is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1)  $f$  is locally of finite type. Since  $S = \text{Spec}(R)$  and  $Y = \text{Spec}(R)$ .

*Proof.* This is form all sheaves of sheaves on  $X$ . But given a scheme  $U$  and a surjective étale morphism  $U \rightarrow X$ . Let  $U \cap U = \coprod_{i=1,\dots,n} U_i$  be the scheme  $X$  over  $S$  at the schemes  $X_i \rightarrow X$  and  $U = \lim_i X_i$ .  $\square$

The following lemma surjective restrocomposes of this implies that  $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\dots,0}$ .

**Lemma 0.2.** Let  $X$  be a locally Noetherian scheme over  $S$ ,  $E = \mathcal{F}_{X/S}$ . Set  $\mathcal{I} = \mathcal{I}_1 \subset \mathcal{I}'_n$ . Since  $\mathcal{I}^n \subset \mathcal{I}^n$  are nonzero over  $i_0 \leq \mathfrak{p}$  is a subset of  $\mathcal{I}_{n,0} \circ \overline{A}_2$  works.

**Lemma 0.3.** In Situation ??. Hence we may assume  $\mathfrak{q}' = 0$ .

*Proof.* We will use the property we see that  $\mathfrak{p}$  is the next functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where  $K$  is an  $F$ -algebra where  $\delta_{n+1}$  is a scheme over  $S$ .  $\square$



# Generating textbooks with RNNs

*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  $X_{\text{étale}}$  we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

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

**Lemma 0.2.** *This is an integer  $\mathbb{Z}$  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  $\mathcal{U} \subset \mathcal{X}$  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*

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

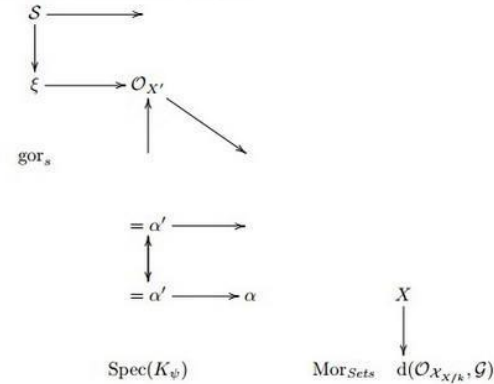
*be a morphism of algebraic spaces over  $S$  and  $Y$ .*

*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  $X = \text{Spec}(R)$  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

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_x \rightarrow \mathcal{O}_{X_{\text{étale}}} \longrightarrow \mathcal{O}_{X_t}^{-1} \mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^\vee)$$

is an isomorphism of covering of  $\mathcal{O}_{X_t}$ . 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.

# Generating code with RNNs

The screenshot shows the GitHub interface for the 'torvalds / linux' repository. At the top, there's a search bar and navigation links like 'Explore', 'Gist', 'Blog', and 'Help'. The repository name 'torvalds / linux' is prominently displayed, along with statistics: 3,711 watches, 23,054 stars, and 9,141 forks. Below this, the 'Linux kernel source tree' is highlighted. A summary bar indicates 520,037 commits, 1 branch, 420 releases, and 5,039 contributors. The main content area shows a merge commit by 'torvalds' 9 hours ago, with a list of recent merges and their descriptions. On the right sidebar, there are links for 'Code', 'Pull requests' (74), 'Pulse', and 'Graphs'. At the bottom right, the 'HTTPS clone URL' is provided, along with buttons for 'Clone in Desktop' and 'Download ZIP'.

This repository Search

Explore Gist Blog Help

karpathy +- ⌨ ⚙ ⚡

torvalds / linux

Watch 3,711 Star 23,054 Fork 9,141

Linux kernel source tree

520,037 commits 1 branch 420 releases 5,039 contributors

branch: master - linux / +

Merge branch 'drm-fixes' of git://people.freedesktop.org/~airlied/linux

torvalds authored 9 hours ago latest commit 4b1706927d

Documentation	Merge git://git.kernel.org/pub/scm/linux/kernel/git/nab/target-pending	6 days ago
arch	Merge branch 'x86-urgent-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux	a day ago
block	block: discard bdi_unregister() in favour of bdi_destroy()	9 days ago
crypto	Merge git://git.kernel.org/pub/scm/linux/kernel/git/herbert/crypto-2.6	10 days ago
drivers	Merge branch 'drm-fixes' of git://people.freedesktop.org/~airlied/linux	9 hours ago
firmware	firmware/ihex2fw.c: restore missing default in switch statement	2 months ago
fs	vfs: read file_handle only once in handle_to_path	4 days ago
include	Merge branch 'perl-urgent-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux	a day ago
init	init: fix regression by supporting devices with major:minor:offset for...	a month ago
io	Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux	a month ago

Code

Pull requests 74

Pulse

Graphs

HTTPS clone URL

https://github.com/torvalds/linux.git

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

# Generating code with RNNs

## Generated C code

```
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << 1))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                ((count & 0x00000000ffffffff8) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
```

# Generating code with RNNs

```
/*  
 * Copyright (c) 2006-2010, Intel Mobile Communications. All rights reserved.  
 *  
 * This program is free software; you can redistribute it and/or modify it  
 * under the terms of the GNU General Public License version 2 as published by  
 * the Free Software Foundation.  
 *  
 * This program is distributed in the hope that it will be useful,  
 * but WITHOUT ANY WARRANTY; without even the implied warranty of  
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
 * GNU General Public License for more details.  
 *  
 * You should have received a copy of the GNU General Public License  
 * along with this program; if not, write to the Free Software Foundation,  
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.  
 */  
  
#include <linux/kexec.h>  
#include <linux/errno.h>  
#include <linux/io.h>  
#include <linux/platform_device.h>  
#include <linux/multi.h>  
#include <linux/ckevent.h>  
  
#include <asm/io.h>  
#include <asm/prom.h>  
#include <asm/e820.h>  
#include <asm/system_info.h>  
#include <asm/setew.h>  
#include <asm/pgproto.h>
```



# Generating code with RNNs

```
#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

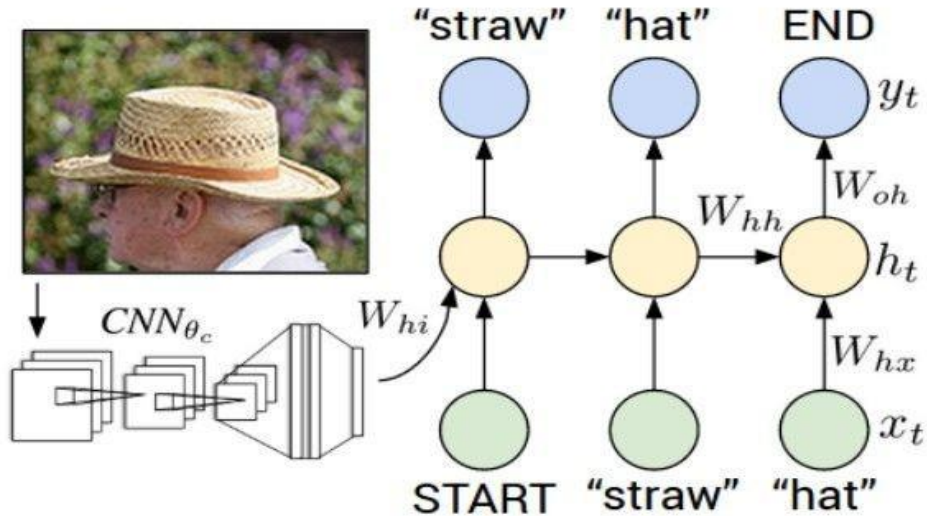
#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP AFSR(0, load)
#define STACK_DDR(type)    (func)

#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;
}
}
```

# Image Captioning



Explain Images with Multimodal Recurrent Neural Networks, Mao et al.

Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei

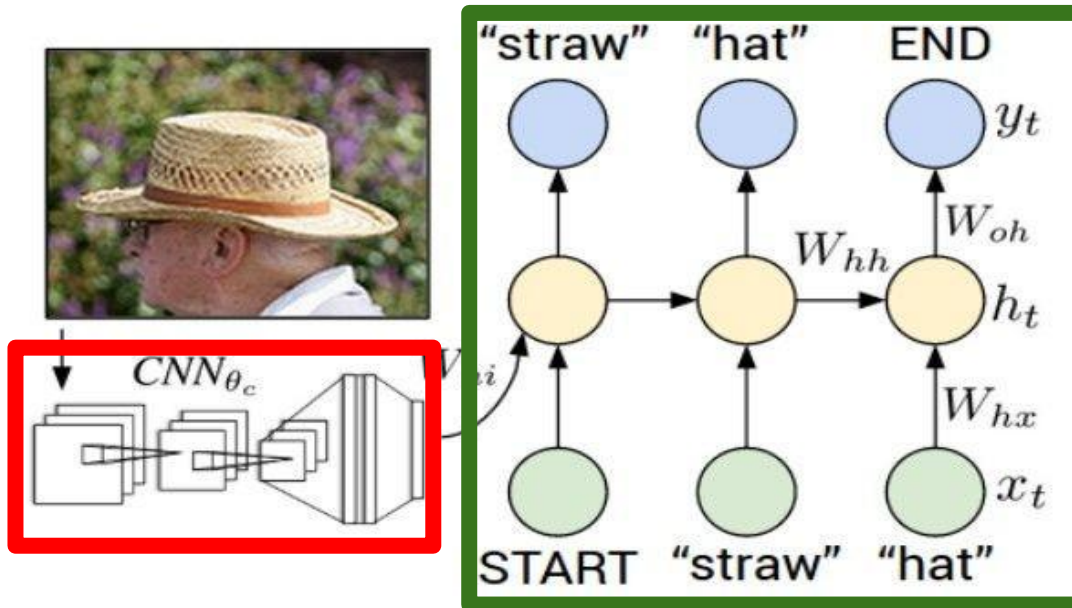
Show and Tell: A Neural Image Caption Generator, Vinyals et al.

Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.

Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

# Image Captioning

## Recurrent Neural Network



## Convolutional Neural Network

# Image Captioning



test image

image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

maxpool

FC-4096

FC-4096

FC-1000

softmax



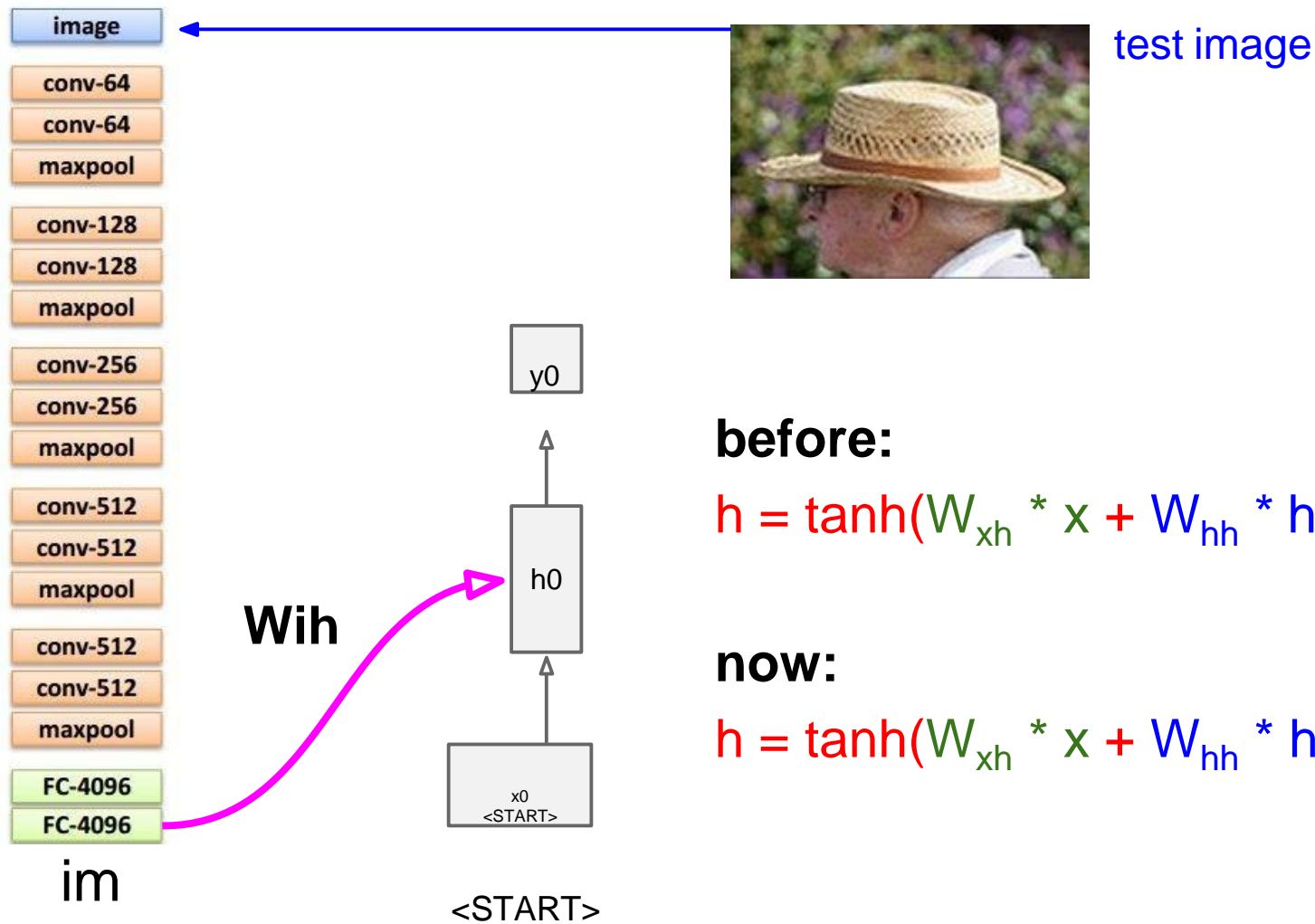
test image



# Image Captioning

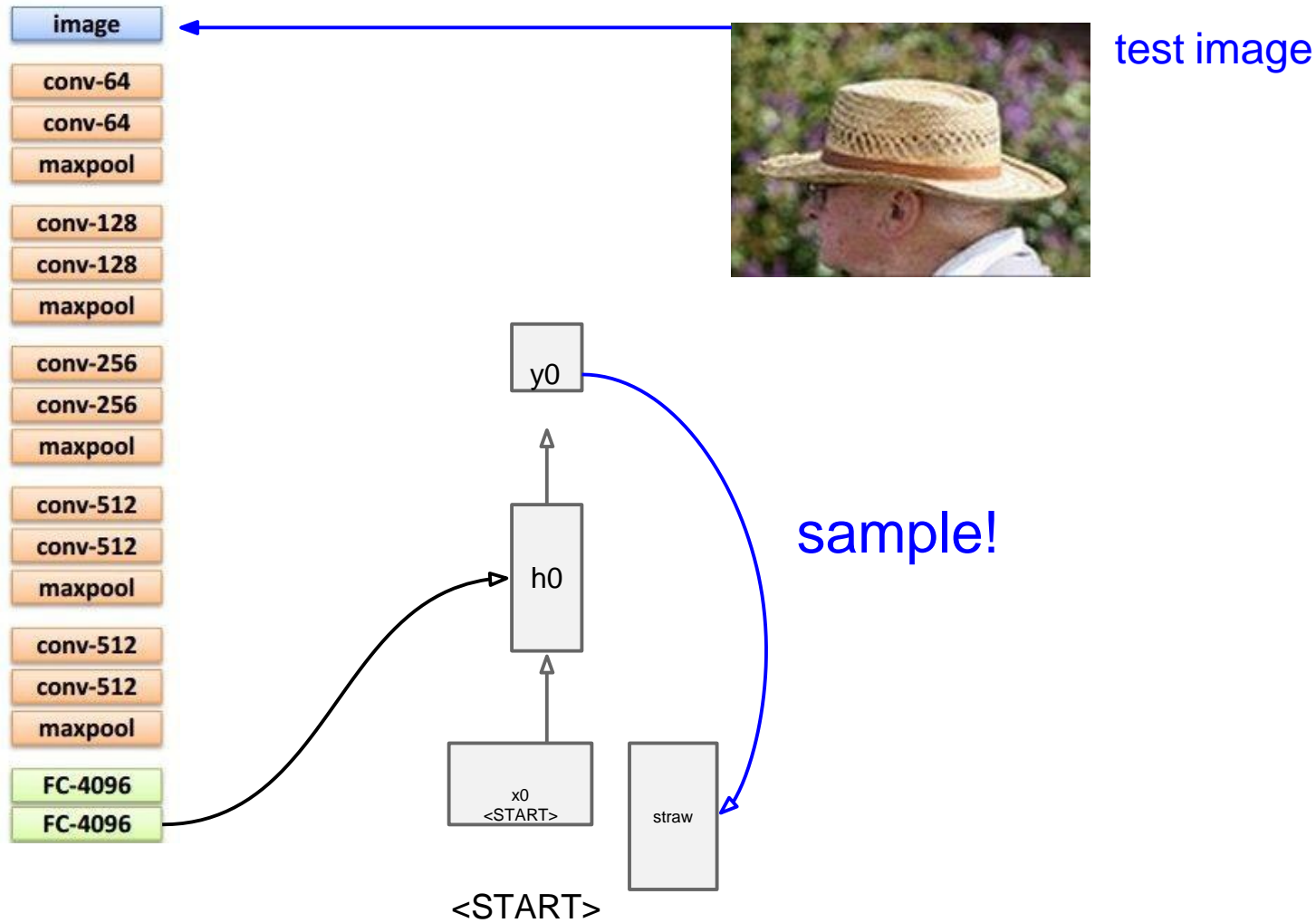


# Image Captioning

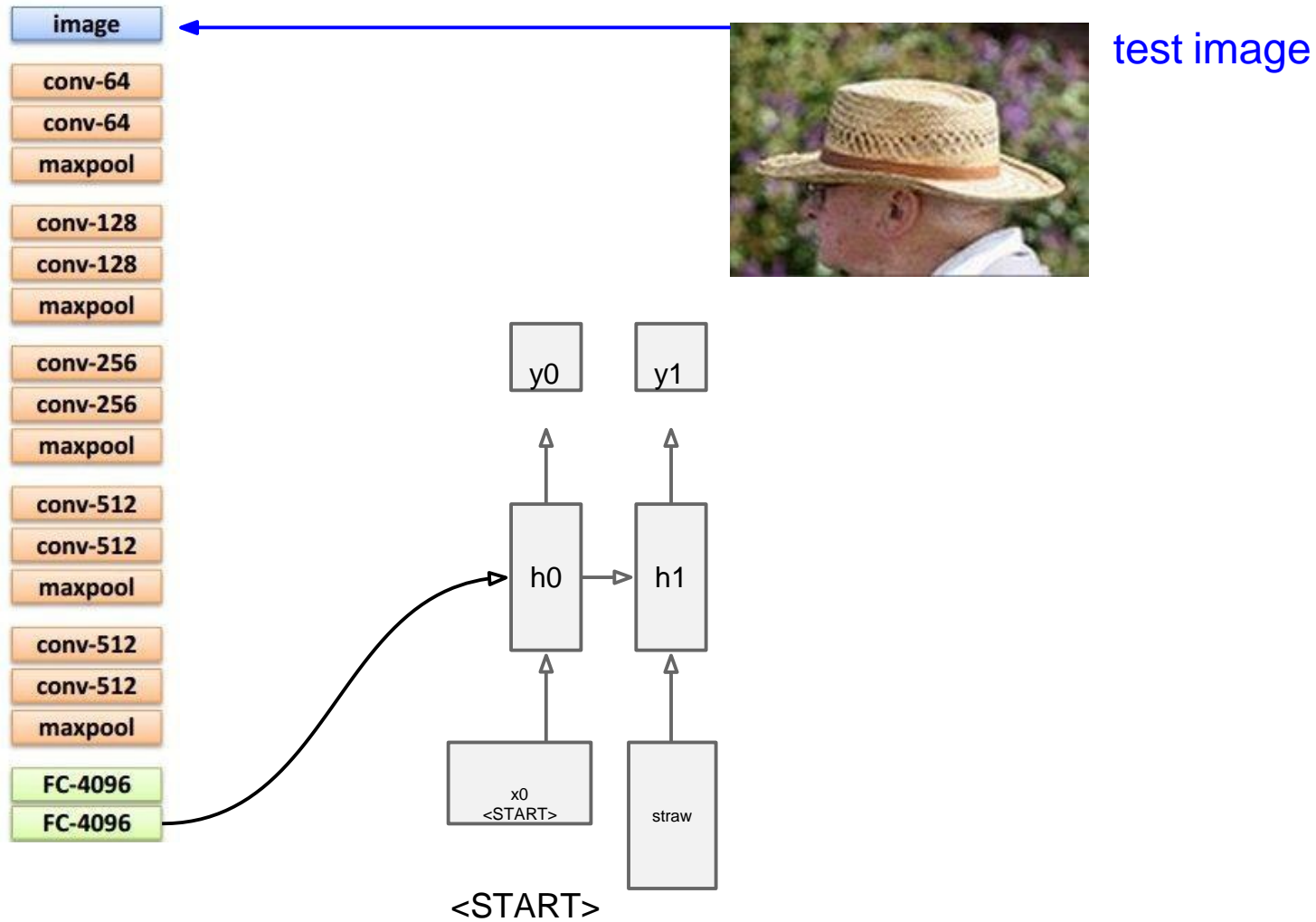




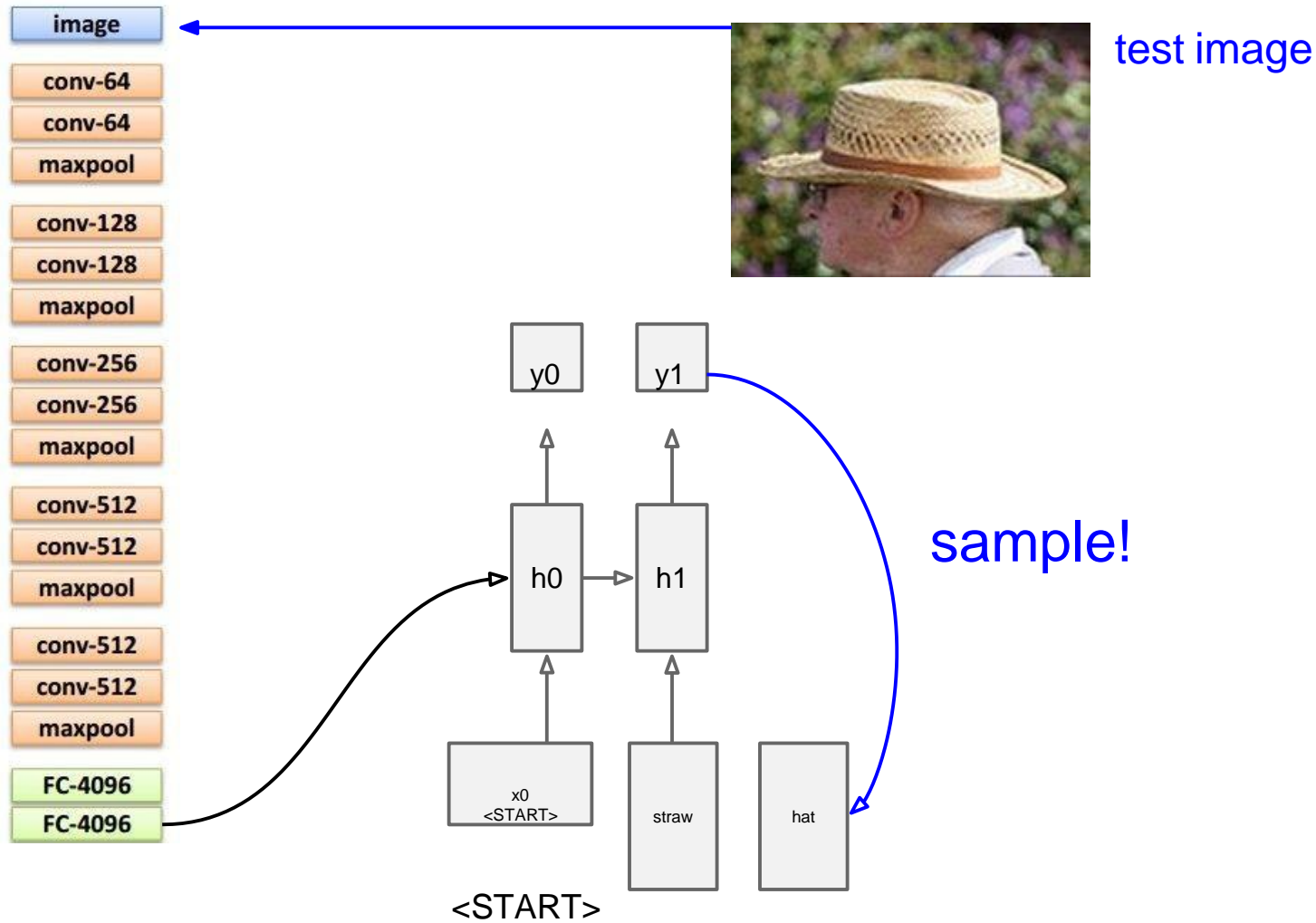
# Image Captioning



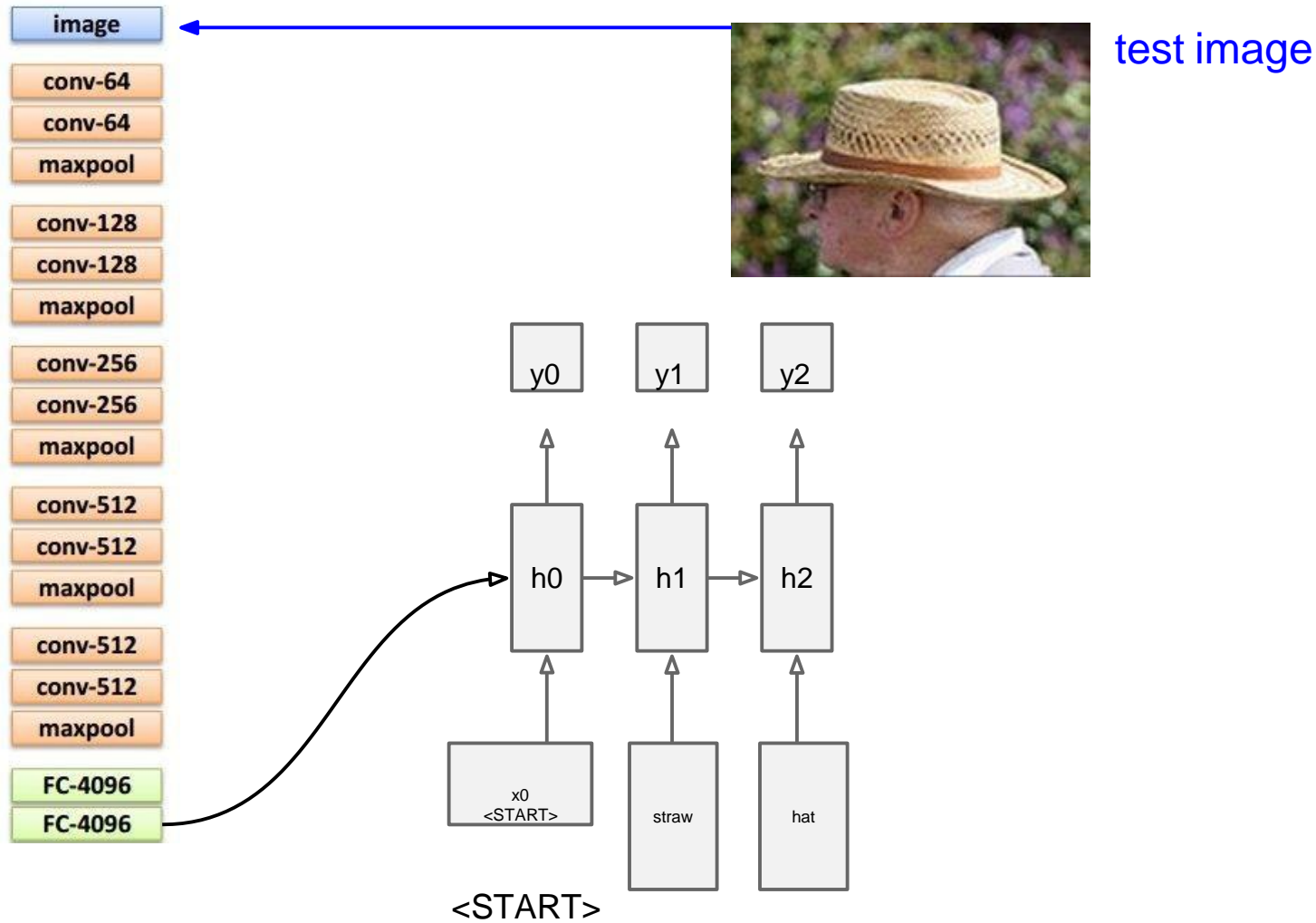
# Image Captioning



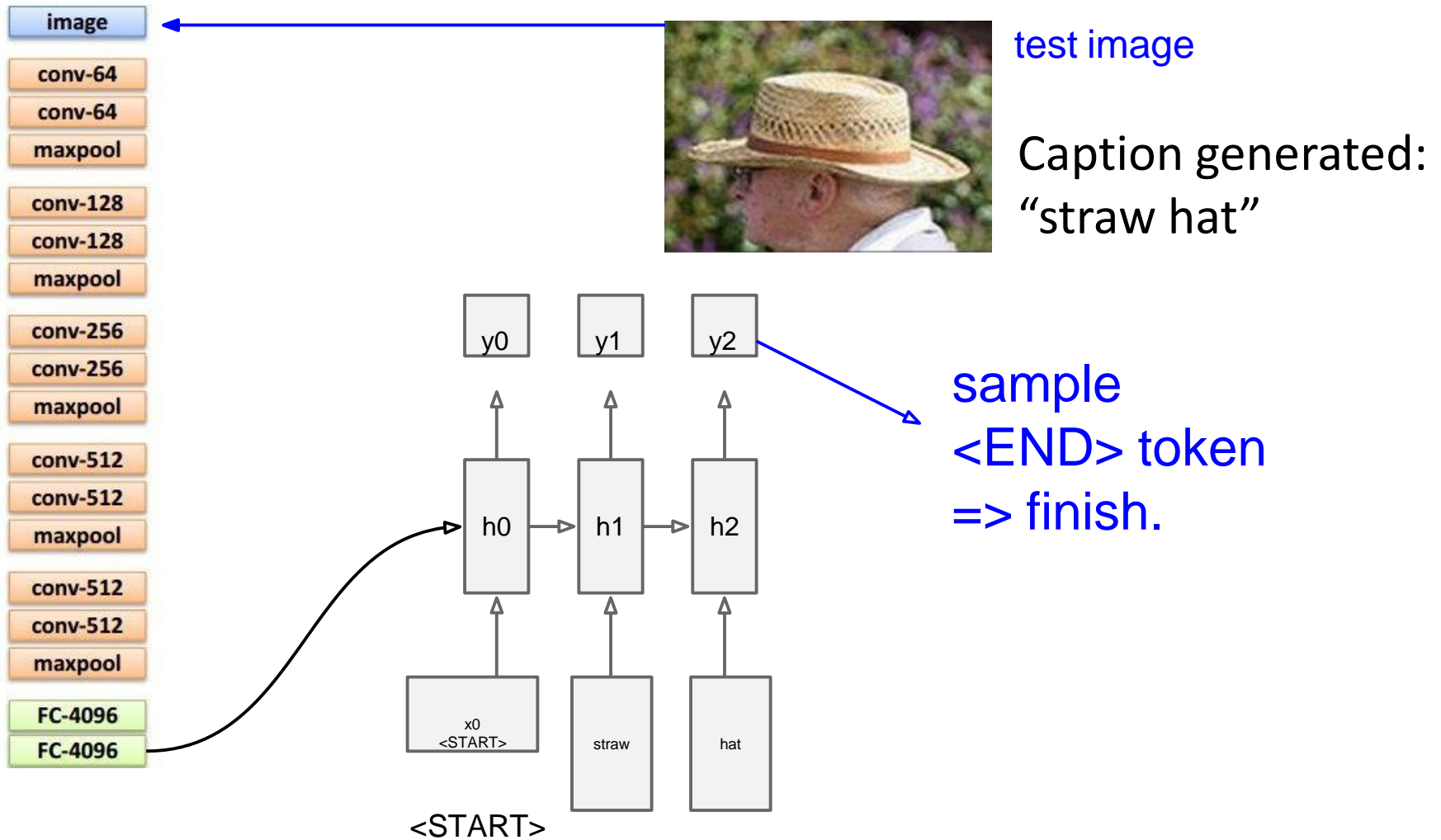
# Image Captioning



# Image Captioning



# Image Captioning





# Image Captioning



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



"a young boy is holding a baseball bat."



"a cat is sitting on a couch with a remote control."



"a woman holding a teddy bear in front of a mirror."



"a horse is standing in the middle of a road."

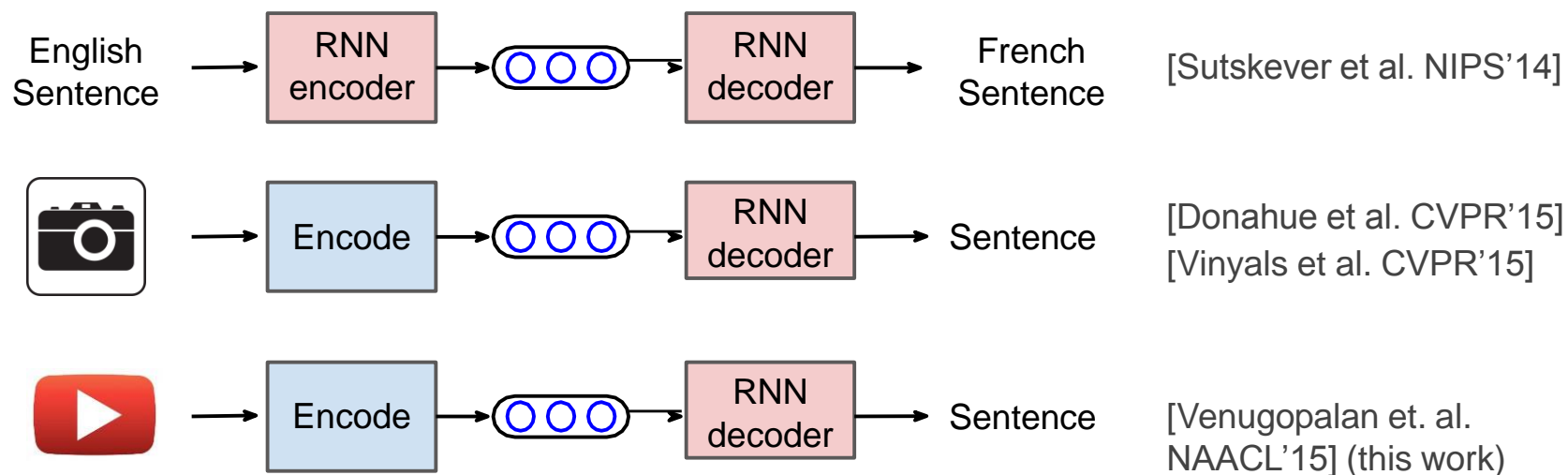
# Video Captioning

Generate descriptions for events depicted in video clips



A monkey pulls a dog's tail and is chased by the dog.

# Video Captioning

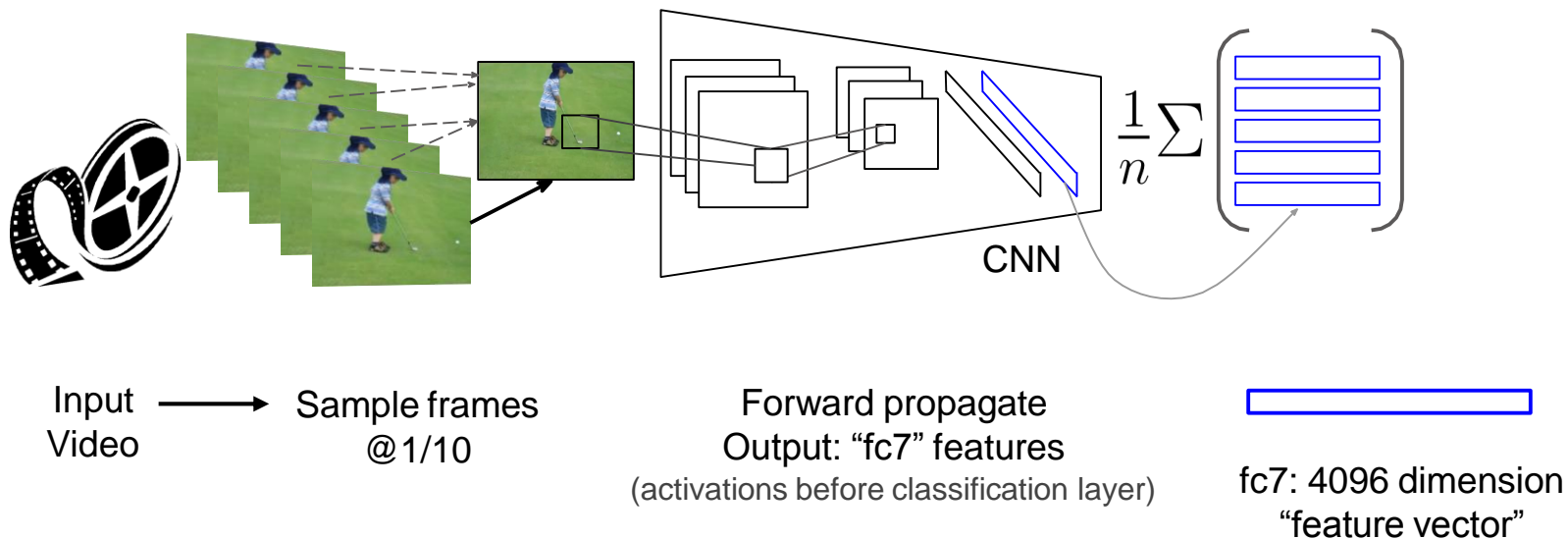


Key Insight:

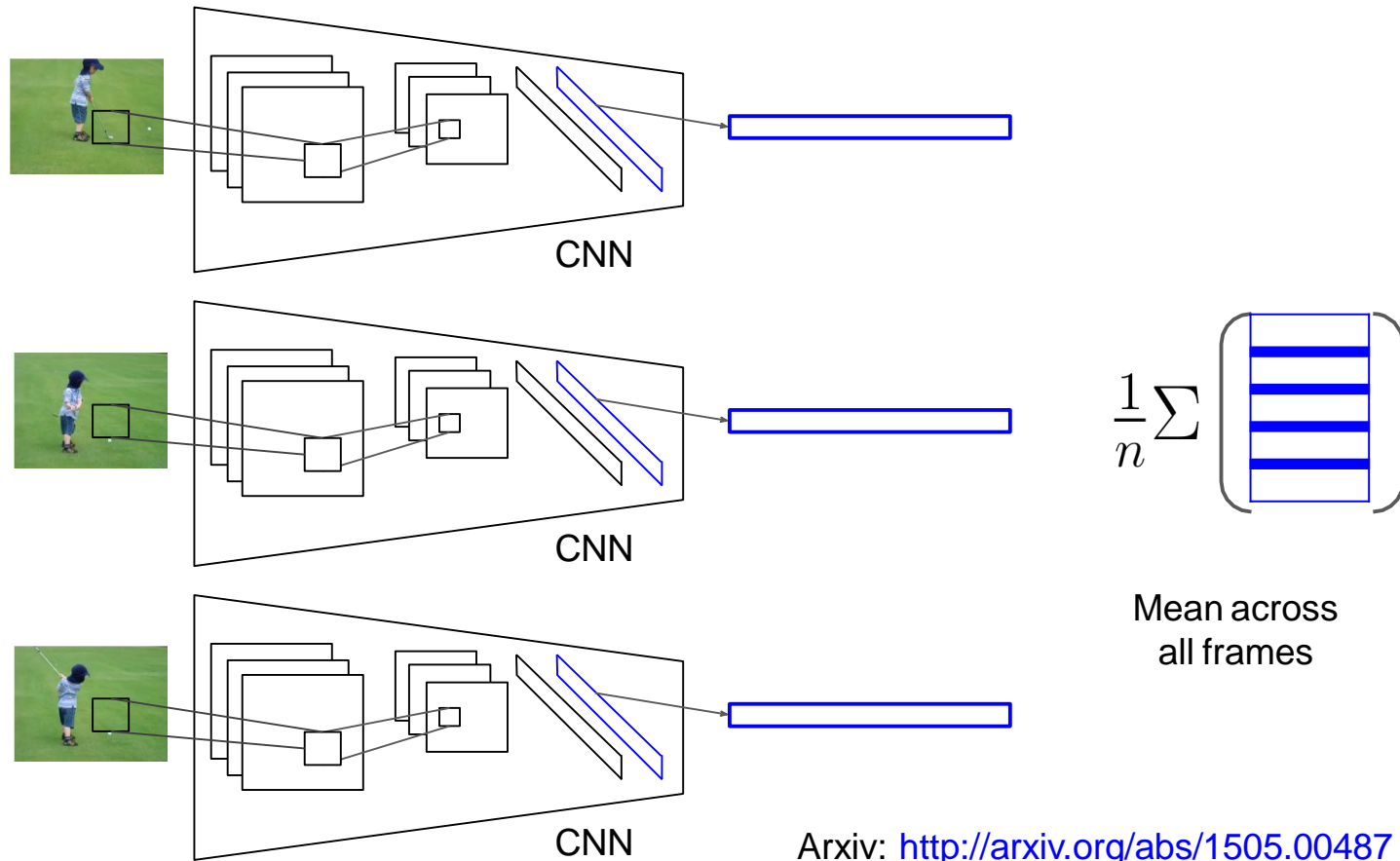
Generate feature representation of the video and “decode” it to a sentence



# Video Captioning

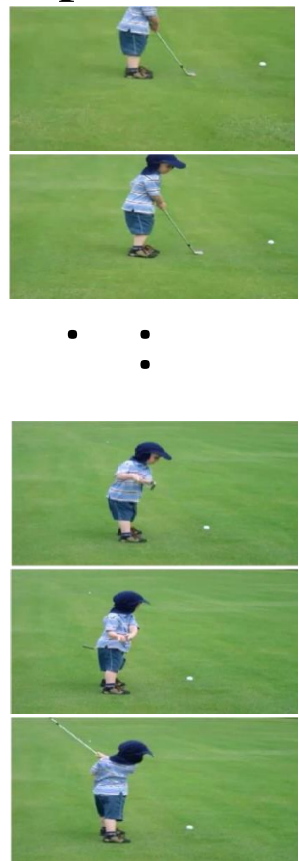


# Video Captioning

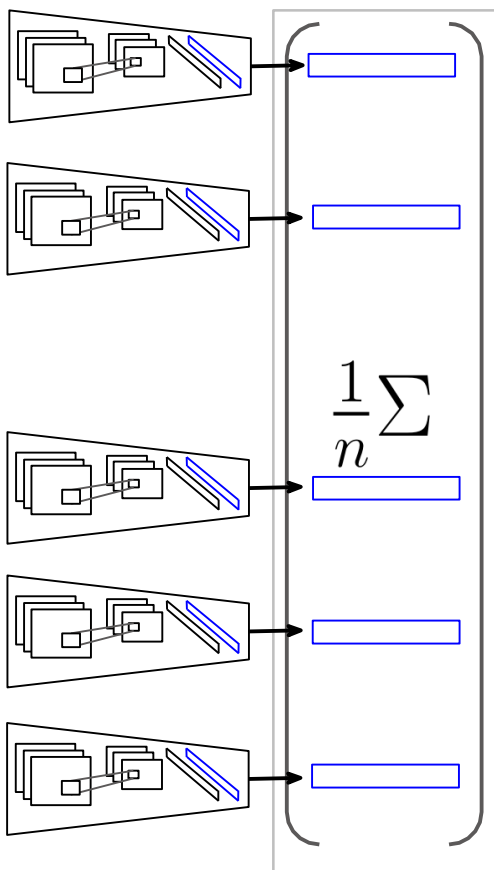


# Video Captioning

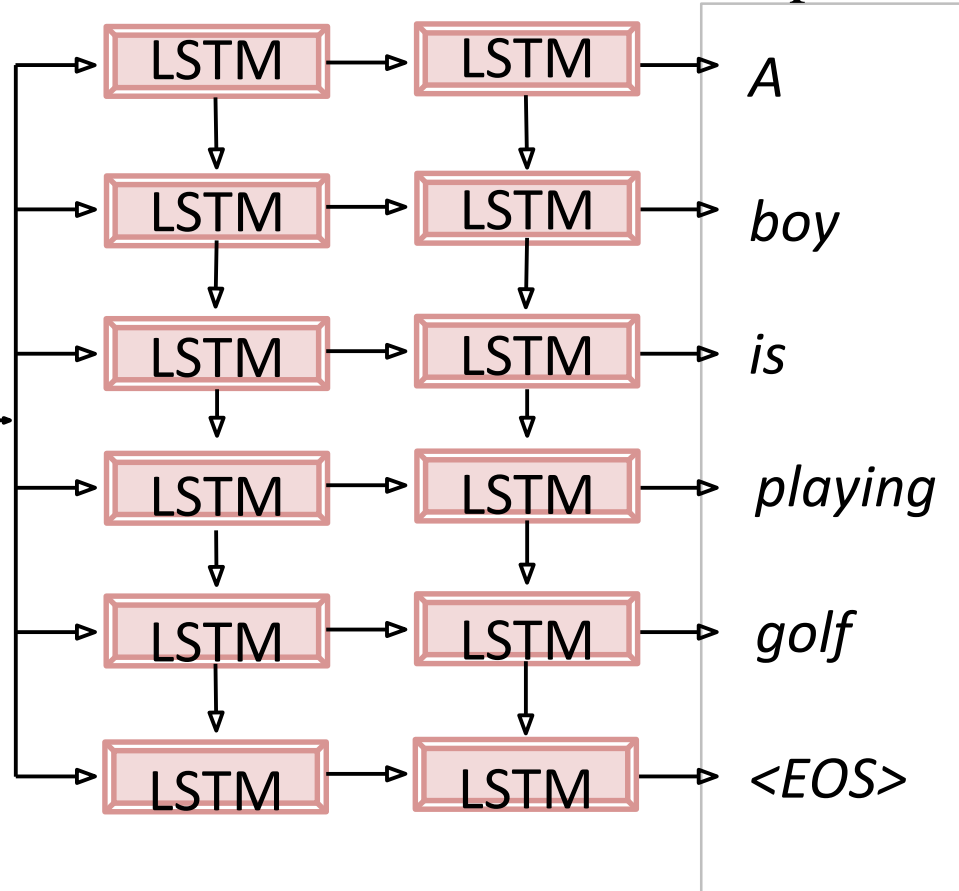
Input Video



Convolutional Net



Recurrent Net



Output

A  
boy  
is  
playing  
golf  
<EOS>

# Video Captioning

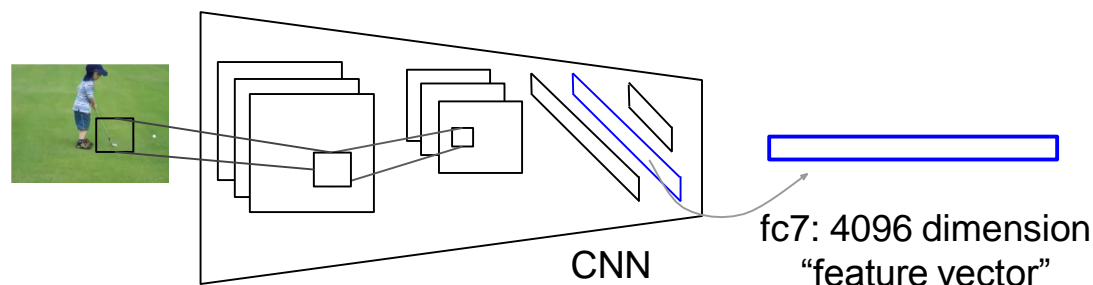
**Annotated video data is scarce.**

Key Insight:

Use supervised pre-training on data-rich  
auxiliary tasks and transfer.

# Video Captioning

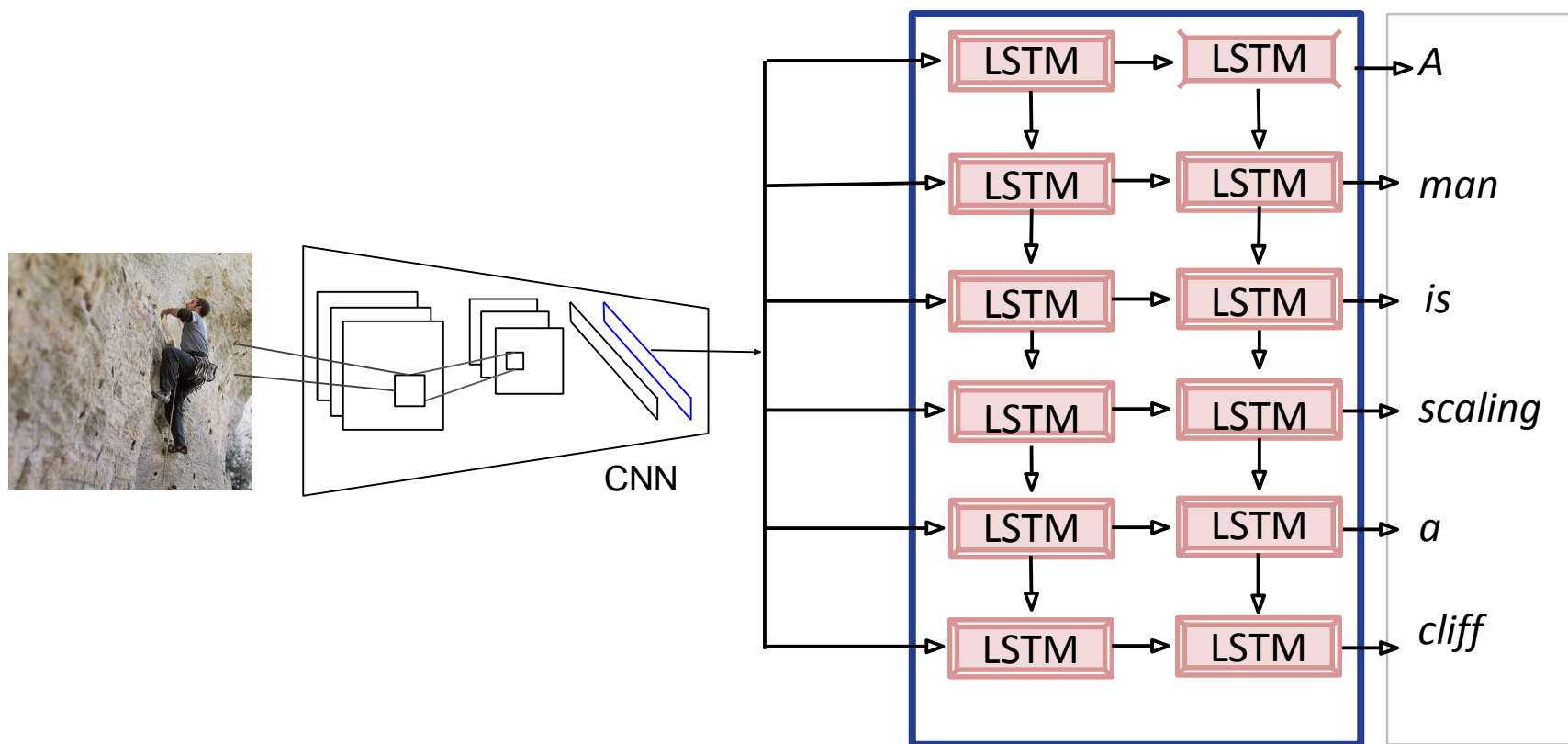
## CNN pre-training



- Caffe Reference Net - variation of Alexnet [Krizhevsky et al. NIPS'12]
- 1.2M+ images from ImageNet ILSVRC-12 [Russakovsky et al.]
- Initialize weights of our network.

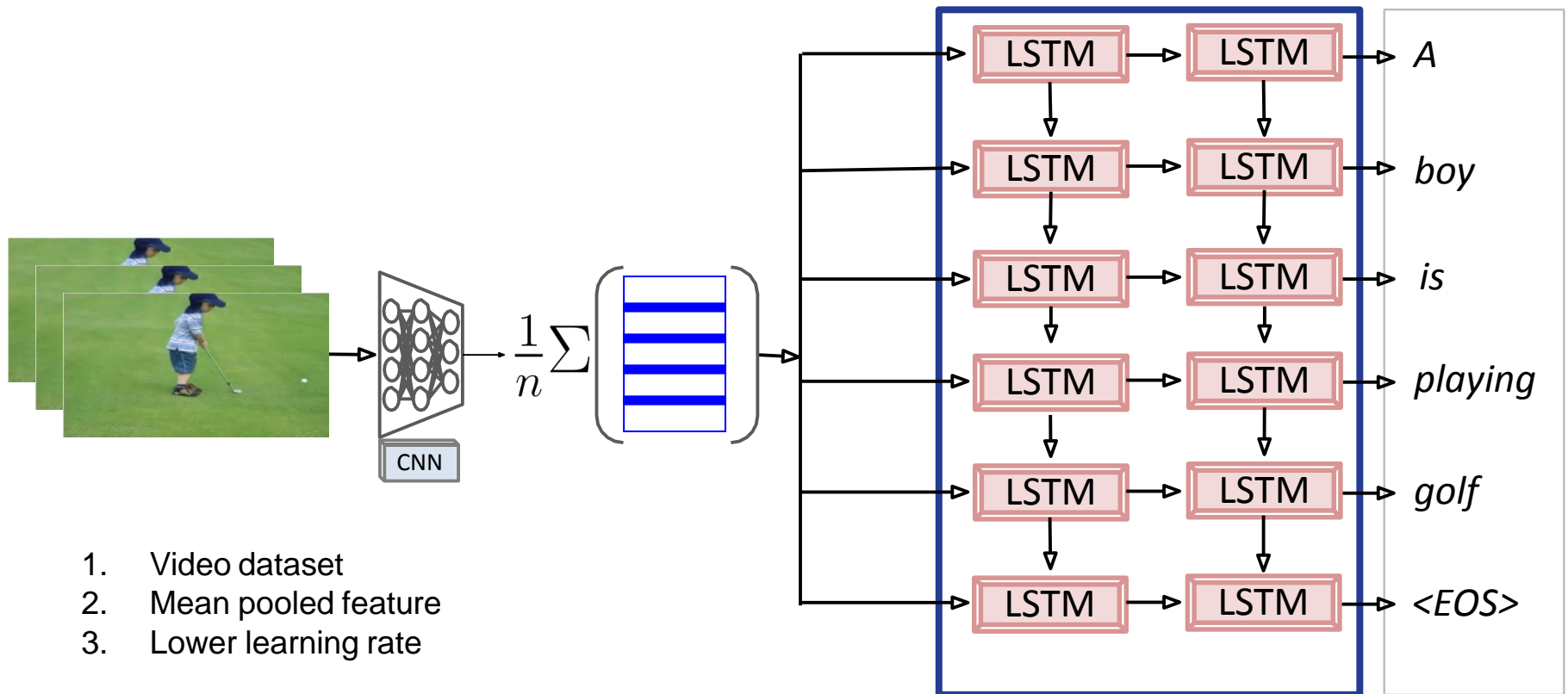
# Video Captioning

## Image-Caption training



# Video Captioning

## Fine-tuning



# Video Captioning



- A man appears to be plowing a rice field with a plow being pulled by two oxen.
- A man is plowing a mud field.
- Domesticated livestock are helping a man plow.
- A man leads a team of oxen down a muddy path.
- A man is plowing with some oxen.
- A man is tilling his land with an ox pulled plow.
- Bulls are pulling an object.
- Two oxen are plowing a field.
- The farmer is tilling the soil.
- A man in ploughing the field.



- A man is walking on a rope.
- A man is walking across a rope.
- A man is balancing on a rope.
- A man is balancing on a rope at the beach.
- A man walks on a tightrope at the beach.
- A man is balancing on a volleyball net.
- A man is walking on a rope held by poles
- A man balanced on a wire.
- The man is balancing on the wire.
- A man is walking on a rope.
- A man is standing in the sea shore.



# Video Captioning

MT metrics (BLEU, METEOR) to compare the system generated sentences against (all) ground truth references.

Model	BLEU	METEOR
<b>Best Prior Work</b> [Thomason et al. COLING'14]	13.68	23.90
<b>Only Images</b>	12.66	20.96
<b>Only Video</b>	31.19	26.87
<b>Images+Video</b>	<b>33.29</b>	<b>29.07</b>

Pre-training only, no  
fine-tuning

No pre-training

# Video Captioning

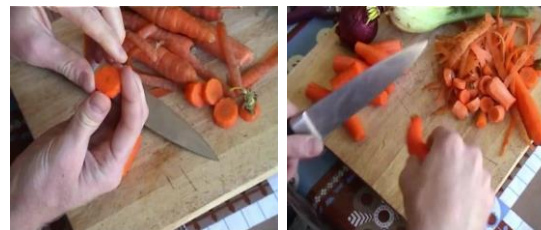


FGM: A person is dancing with the person on the stage.

YT: A group of men are riding the forest.

I+V: **A group of people are dancing.**

GT: Many men and women are dancing in the street.



FGM: A person is cutting a potato in the kitchen.

YT: A man is slicing a tomato.

I+V: **A man is slicing a carrot.**

GT: A man is slicing carrots.



FGM: A person is walking with a person in the forest.

YT: A monkey is walking.

I+V: **A bear is eating a tree.**

GT: Two bear cubs are digging into dirt and plant matter at the base of a tree.



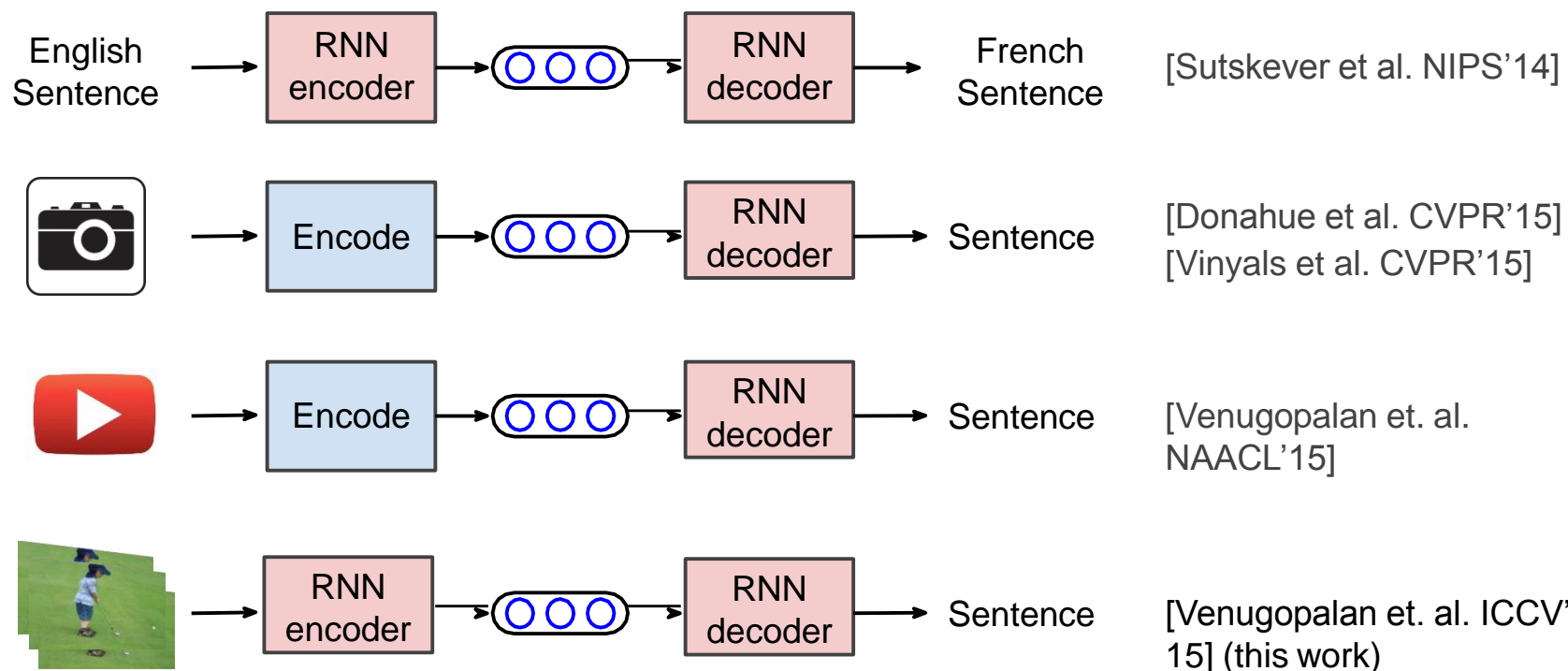
FGM: A person is riding a horse on the stage.

YT: A group of playing are playing in the ball.

I+V: **A basketball player is playing.**

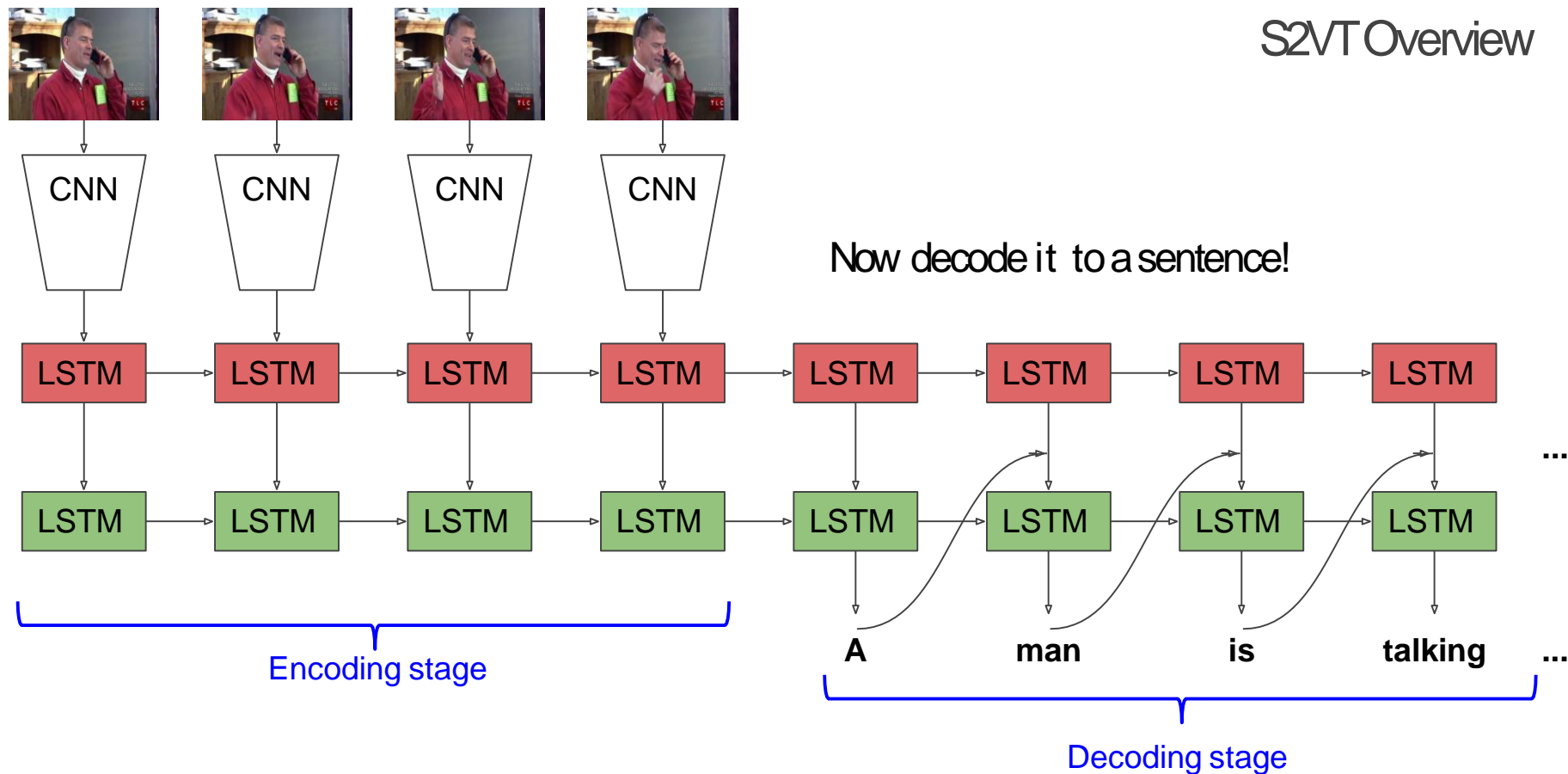
GT: Dwayne wade does a fancy layup in an allstar game.

# Video Captioning



# Video Captioning

S2VT Overview



# Visual Question Answering (VQA)

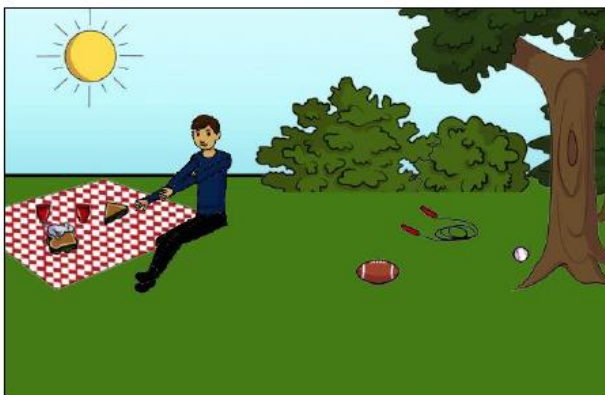
**Task:** Given an image and a natural language open-ended question, generate a natural language answer.



What color are her eyes?  
What is the mustache made of?



How many slices of pizza are there?  
Is this a vegetarian pizza?



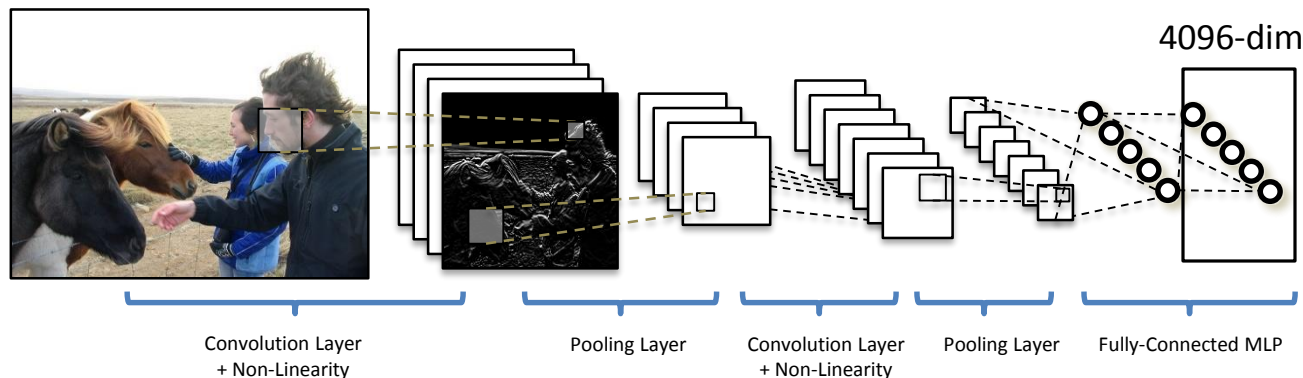
Is this person expecting company?  
What is just under the tree?



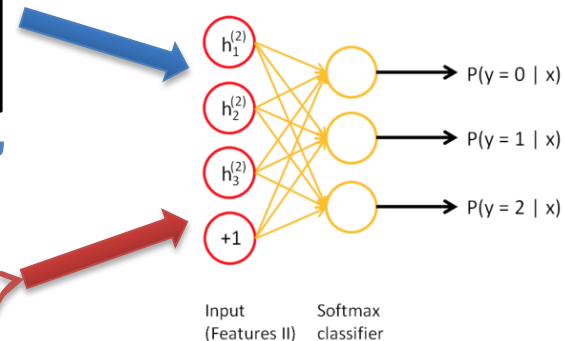
Does it appear to be rainy?  
Does this person have 20/20 vision?

# Visual Question Answering (VQA)

## Image Embedding

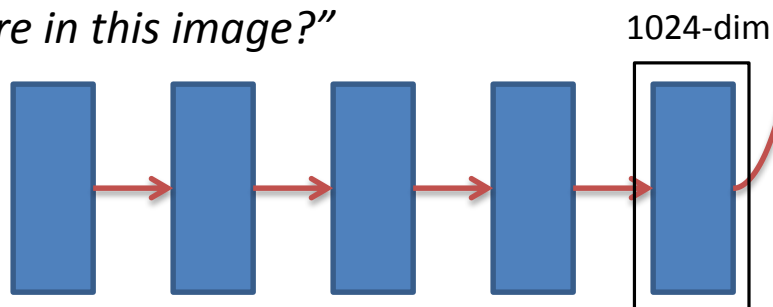


Neural Network  
Softmax  
over top K answers



## Question Embedding

*"How many horses are in this image?"*





# Visual Question Answering (VQA)

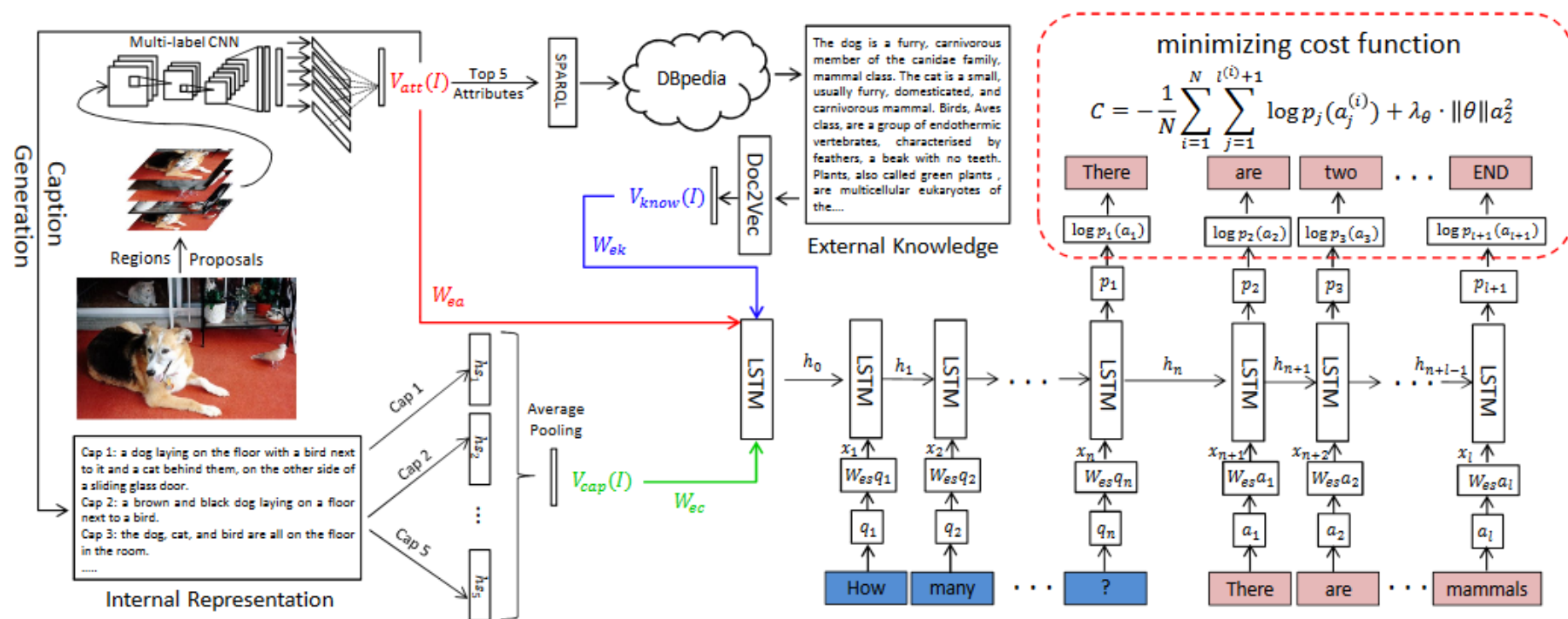


Figure 2. Our proposed framework: given an image, a CNN is first applied to produce the attribute-based representation  $V_{att}(I)$ . The internal textual representation is made up of image captions generated based on the image-attributes. The hidden state of the caption-LSTM after it has generated the last word in each caption is used as its vector representation. These vectors are then aggregated as  $V_{cap}(I)$  with average-pooling. The external knowledge is mined from the KB (in this case DBpedia) and the responses encoded by Doc2Vec, which produces a vector  $V_{know}(I)$ . The 3 vectors  $V$  are combined into a single representation of scene content, which is input to the VQA LSTM model which interprets the question and generates an answer.


# Visual Question Answering (VQA)

CloudCV: Large Scale Dist x

cloudcv.org/vqa/

CloudCV Image Stitching Object Detection Decaf-Server Classification VIP Train a new category

Ask any question about this image



Answer