Machine Intelligence & Deep Learning Workshop

Raymond Ptucha, Majid Rabbani, Mark Smith

The Kate Gleason COLLEGE OF ENGINEERING

Language & Vision

Raymond Ptucha
June 27-29, 2018
Rochester Institute of Technology
www.rit.edu/kgcoe/cqas/machinelearning

Fair Use Agreement

This agreement covers the use of all slides in this document, please read carefully.

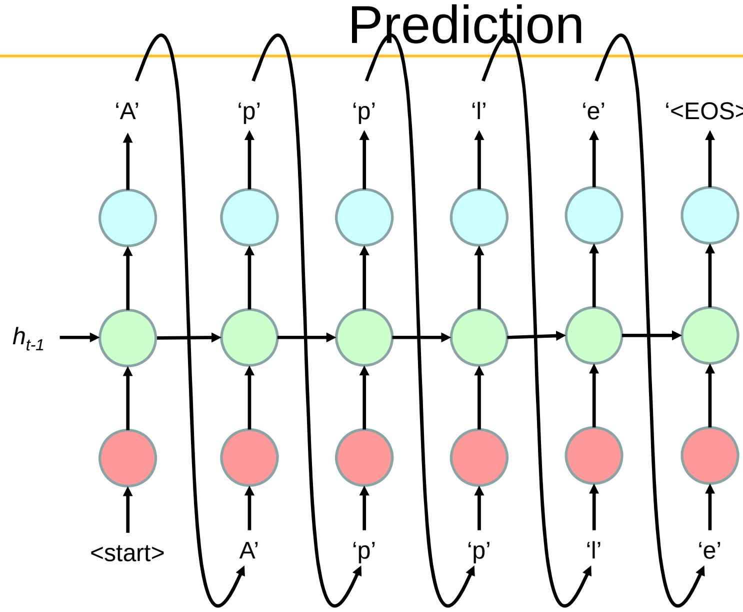
- You may freely use these slides for personal use, if:
  - My name (R. Ptucha) appears on each slide.
- You may freely use these slides externally, if:
  - You send me an email telling me the conference/venue/company name in advance, and which slides you wish to use.
  - You receive a positive confirmation email back from me.
  - My name (R. Ptucha) appears on each slide you use.

(c) Raymond Ptucha, rwpeec@rit.edu

Agenda

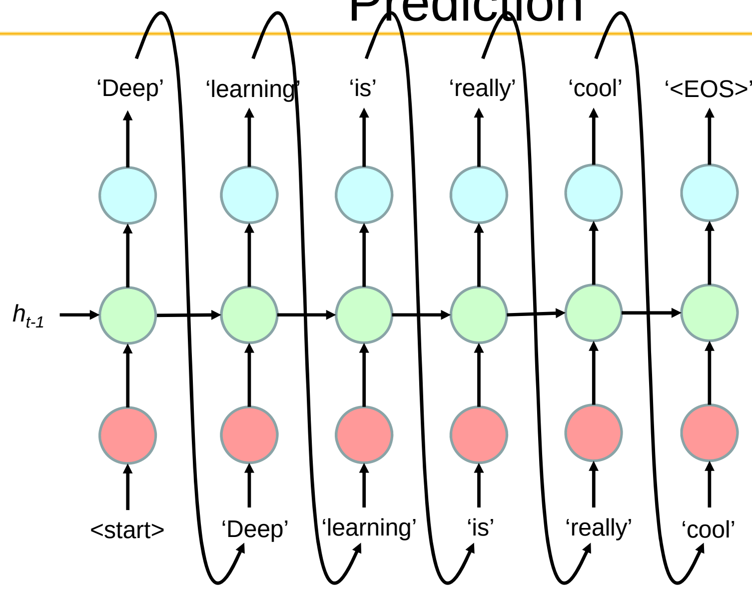
- Wed, June 27
  - 9-10:30am — Regression and Classification
  - 10:30-10:45pm — Break
  - 10:45-12:15pm — Boosting and SVM
  - 12:15-1:30pm — Lunch
  - 1:30-3:30pm — Neural Networks and Dimensionality Reduction
  - 3:30-5pm — Hands-on Python and Machine Learning
- Thur, June 28
  - 9-10:30am — Introduction to deep learning
  - 10:30-10:45pm — Break
  - 10:45-12:15pm — Convolutional Neural Networks
  - 12:15-1:30pm — Lunch
  - 1:30-3:30pm — Region and pixel-level convolutions
  - 3:30-5pm — Hands-on CNNs
- Fri, June 29
  - 9-10:30am — Recurrent neural networks
  - 10:30-10:45pm — Break
  - 10:45-12:15pm — **Language and Vision**
  - 12:15-1:30pm — Lunch
  - 1:30-3:30pm — Graph convolutional neural networks; Generative adversarial networks
  - 3:30-5pm — Hands-on regional CNNs, RNNs

Recurrent Networks for Character Prediction



For this to work, we need to represent characters as some latent vector numerical representation.

Recurrent Networks for Word Prediction



For this to work, we need to represent words as some latent vector numerical representation.

Word2vec

- In the simplest form, we can start with a one-hot encoded vector of all words, and then learn a model which converts to a lower dimensional representation.
- Word2vec, glove, and skip-gram are popular metrics which encode words to a latent vector representation (~300 dimensions).

- Now we have a way to represent images, characters, and words as vectors.

One-hot Word Representation

- One-hot character representations work great, but there are only 26-100 unique characters, meaning our character embedding only needs to be 26-100 dimensions.
- Words can be embedded identically to characters.
- If we have 10K unique words, our word embedding is 10K dimensions.

One-hot Word Representation

I want a glass of orange ______

- Lets say our vocabulary, V = [a, aaa, aaron, … , zulu], $V \in R^{10,000}$ in one-hot representation.
- Instead of predicting next character, we will predict next word.

$$
\begin{array}{cccccc}
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
(5391) & (9853) & (4914) & (7157) & (456) & (6257) \\
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} &
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} &
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} &
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} &
\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} &
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\end{array}
$$

- A key weakness with this embedding is that the Euclidean distance between any two words is the same.
- Further, for very large vocabularies, our embedded representation is of high dimension.

Inspired by Deeplearning.ai, course 5, week 2

Feature-based Representation

I want a glass of orange ______

- We want similar words to be closer together.
- If we could come up with various features, we can start to encode similarities from one word to the next
- Now, NLP algorithms can utilize similarity from word to word to try to predict the next word, ‘juice’ in this example.

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	+1	-.8	+.95	-.01	-.007
Royal	-.001	+.1	+.85	+.95	-.01	-.057
Age	+.01	-.15	+.75	+.85	+.03	-.06
⋮			⋮			
Food	+.11	+.05	+.05	+.02	+.92	+.96

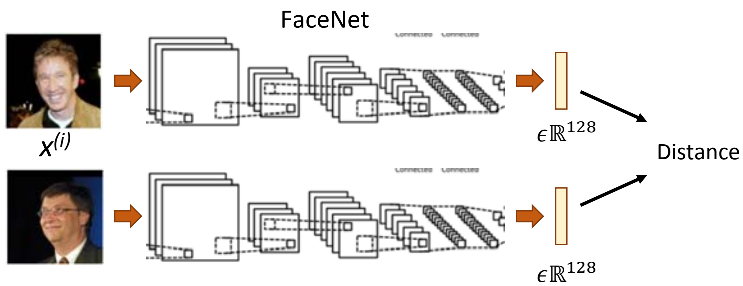
Feature-based Representation

I want a glass of orange juice

- Instead of hand crafted features, let the computer try to learn the features.
- By looking at statistics (what word comes before/after other words in sentences) from billions of samples; the computer can automatically learn this embedding.
- Once learned, we have an embedding, or a latent vector representation for words, **word2vec**.
- Although useful to the computer, it is not clear to humans what each dimension means.

Similar to Face Encoding

- These word embeddings are very similar to the face encoding or face embedding used in DeepFace [Taigman et al. 2014].
- One key difference is word embeddings work on a fixed vocabulary of words, where facial embeddings are meant to work on any unforeseen face.



Analogies

Embedding for man, e_{man}

Embedding for apple, e_{apple}

One-hot index 1:10k

Multiple features	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	+1	-.8	+.95	-.01	-.007
Royal	-.001	+.1	+.85	+.95	-.01	-.057
Age	+.01	-.15	+.75	+.85	+.03	-.06
⋮			⋮			
Food	+.11	+.05	+.05	+.02	+.92	+.96

man → woman as king → queen $\qquad e_{man} - e_{woman} \approx e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$e_{man} - e_{woman} \approx e_{king} - e_w$ Solve for e_w

Find closest word: $\max_{w} similarWord(e_w, e_{king} - e_{man} + e_{woman})$

Similarity Functions

$$\max_w similarWord(\underbrace{e_w}_{\text{Word1}}, \underbrace{e_{king} - e_{man} + e_{woman}}_{\text{Word2}})$$

- Word analogies generally get 30-75% accuracy using these methods.

Man:Woman as Boy:Girl
Ottawa:Canada as Nairobi:Kenya
Big: Bigger as Tall:Taller
Yen:Japan as Ruble:Russia

- Similarity can be measured in Euclidean distance.
- Similarity can be measured in cosine distance:
  - Similar words have small angles, cos(0)=1
  - Opposite words have opposite, 180º angles, cos(180)=-1

$$cosSimilarity(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

Embedding Matrix

- Lets say you have 10K words and your embedding is 300 dimensions.

$$w_e = W w_{one-hot}$$

Where:

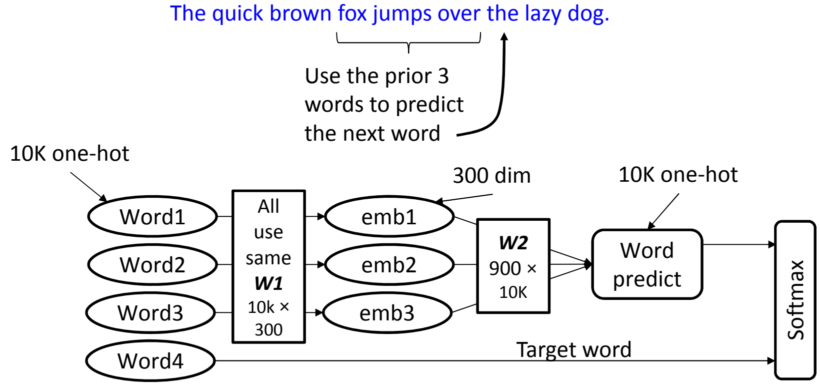
- $w_{one\text{-}hot}$ is the one-hot encoding of each word $\in \mathbb{R}^{10000}$
- W is the embedding matrix $\in \mathbb{R}^{300 \times 10000}$
- w_e is the embedded representation of each word $\in \mathbb{R}^{300}$

Where Does Embedding Matrix Come From?

Bengio et al. 2003, A Neural Probabilistic Language Model

- The embedding matrix is learned through backpropagation!

The quick brown fox jumps over the lazy dog.



- Word predict predicts target word, which is one of 10K words
- Learn ***W1***, ***W2*** using gradient descent, when done, ***W1*** is our embedding

 15

Other Forms of Learning Embedding Matrix

The quick brown fox jumps over the lazy dog.

- Use prior 4 to predict next word.

The quick brown fox jumps over the lazy dog.

- Use before and after to predict center word.

- Use prior 4 and after 4 to predict center word, …

 16

Skip grams

Mikolov et al., 2013, Efficient Estimation of Word Representations in Vector Space

- Start with any reference word, then pseudo randomly chose a nearby target word, say within +/- 5 or within +/- 10 words.
- Alternate model uses a bag of randomly selected neighboring words to predict a target word.
- If context words were chosen completely at random, very common words, often with little value would dominate the classifier.
  - As such, stop words are eliminated.
- Term Frequency-Inverse Document Frequency (TF-IDF) applied to words to increase importance of rare words.
  - TF-IDF makes rare words more important
  - https://en.wikipedia.org/wiki/TF-idf

Using Embedding Matrix

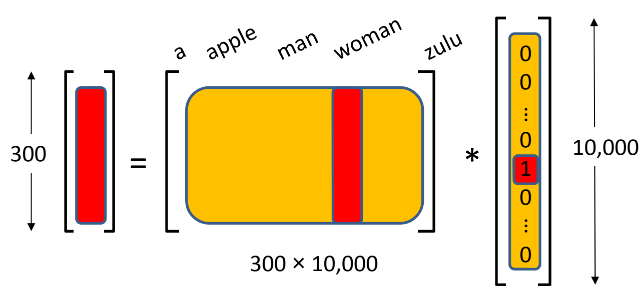
$$w_e = W w_{one-hot}$$

Where:

$w_{one\text{-}hot}$ is the one-hot encoding of each word $\in \mathbb{R}^{10000}$

W is the embedding matrix $\in \mathbb{R}^{300\times10000}$

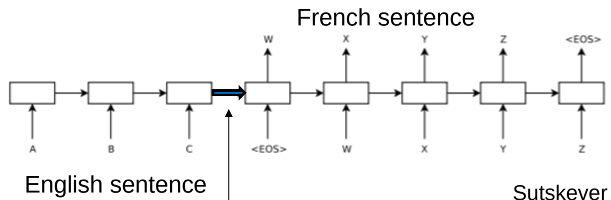
w_e is the embedded representation of each word $\in \mathbb{R}^{300}$



- But wait, only one value of one-hot vector is '1', the rest are zeros.
- Matrix multiply inefficient!
- Can use one-hot offset to extract column of matrix- no costly matrix multiply!

Sent2vec

- In the English to French translation, we have:



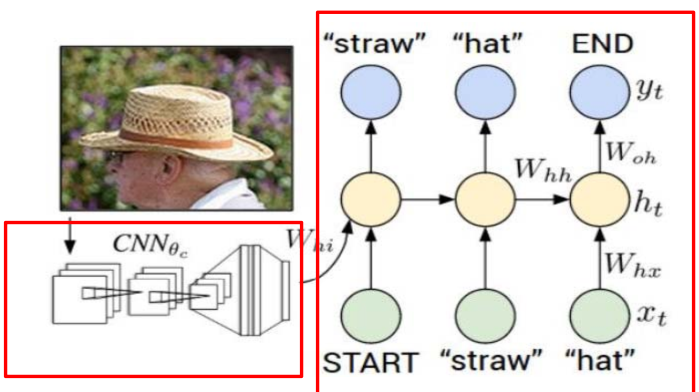
Sutskever et al., NIPS ‘14

…but wait, this point in the RNN is a representation (sent2vec) of all the words in the English sentence!

- Now we have a way to represent images, characters, words, and sentences as vectors…can extend to paragraphs and documents…

Image Captioning

RNN takes in a latent representation of an image, and generates a sequence.



CNN helps represent an image as a numeric value. (image2vec)

Karpathy & Li, CVPR'15

- We may have 50K words. Instead of one-hot encoding, we learn an embedding for each word.

<word1>

$y_t = f(W_{hy} h_t)$

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

<start>

- Glove embedding (300 long vector/word) is very popular.
- Alternately, can learn embedding- learn a matrix which goes from (50K) one-hot to 300, ie: $W_{ix} \in R^{50K \times 300}$
- Embedding and unembedding can be learned or inverses of one another.

Note: Word is sampled from distribution of word probabilities

<word1>

$y_t = f(W_{hy} h_t)$

h_{t-1} → $h_t = f(W_{xh} x_t + W_{hh} h_{t-1} + W_{vh} v)$

v

<start>

v could be FC6, FC7, conv5, conv4, …; or a combination of above

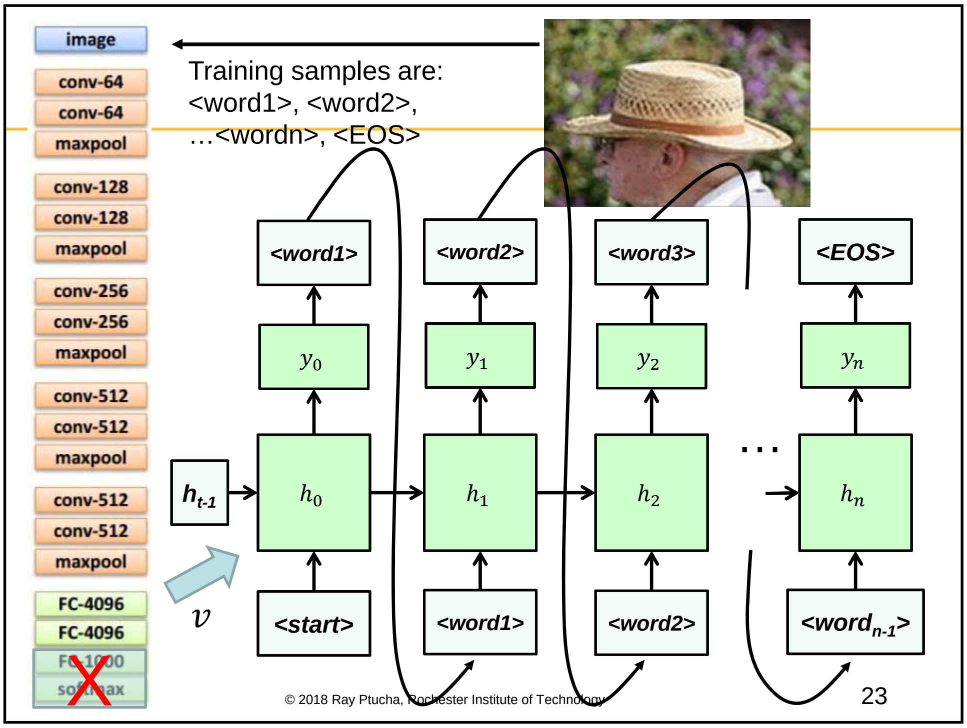


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
Training samples are:
<word1>, <word2>,
…<wordn>, <EOS>
<word1>
<word2>
<word3>
<EOS>
y_0
y_1
y_2
y_n
h_{t-1}
h_0
h_1
h_2
h_n
…
v
<start>
<word1>
<word2>
<word$_{n-1}$>
© 2018 Ray Ptucha, Rochester Institute of Technology
23

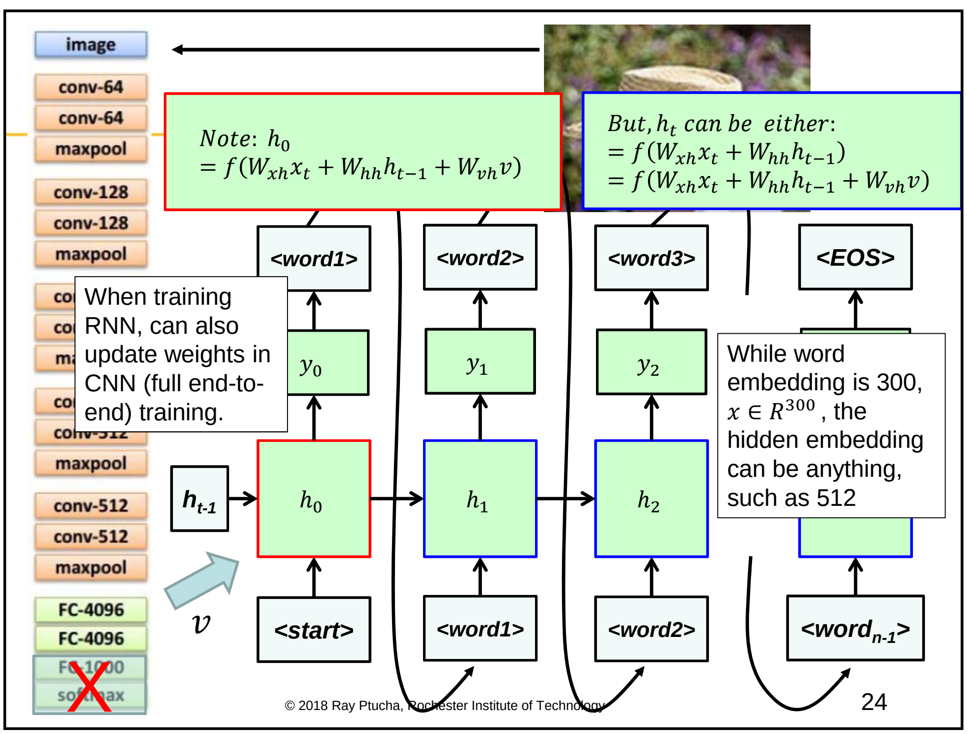


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
Note: $h_0 = f(W_{xh}x_t + W_{hh}h_{t-1} + W_{vh}v)$
But, h_t can be either:
$= f(W_{xh}x_t + W_{hh}h_{t-1})$
$= f(W_{xh}x_t + W_{hh}h_{t-1} + W_{vh}v)$
<word1>
<word2>
<word3>
<EOS>
When training RNN, can also update weights in CNN (full end-to-end) training.
y_0
y_1
y_2
While word embedding is 300, $x \in R^{300}$, the hidden embedding can be anything, such as 512
h_{t-1}
h_0
h_1
h_2
v
<start>
<word1>
<word2>
<word$_{n-1}$>
© 2018 Ray Ptucha, Rochester Institute of Technology
24

Caption Prediction Example

Input Image

2D Plot of fc8 Feature Vector

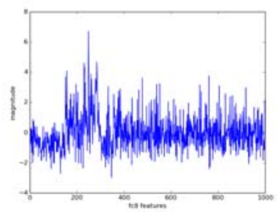
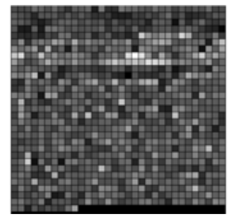


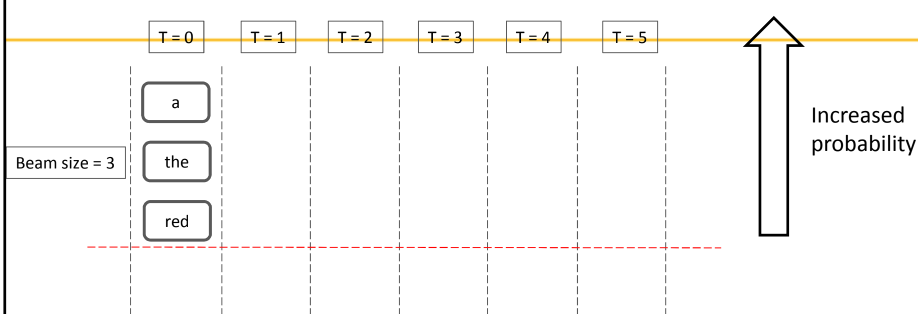
Image of fc8 Feature Vector



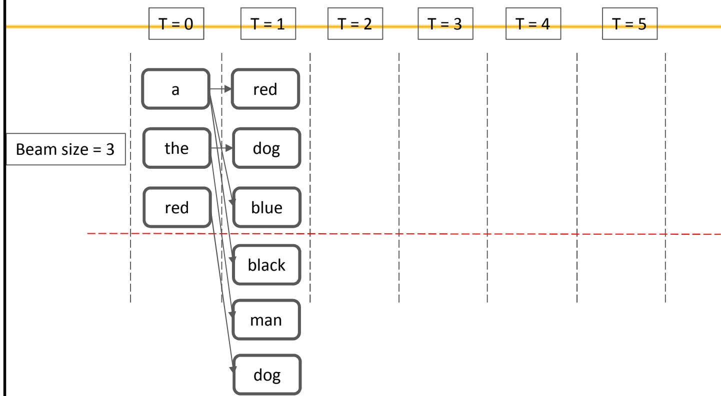
Predicted Sentence

Beam width	Caption
1	A black and white cat sitting on a bed.
2	A black and white cat laying on a bed.
3	A black and white cat laying on top of a bed.
4	A black and white cat laying on a bed with a blanket.

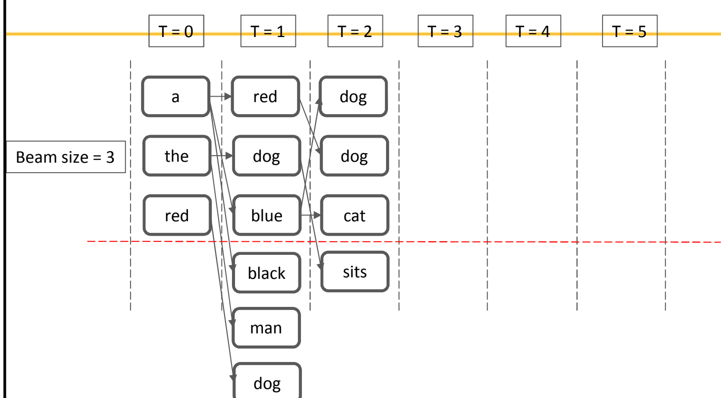
Beam Size Example



Only keep top three sentences at each time step

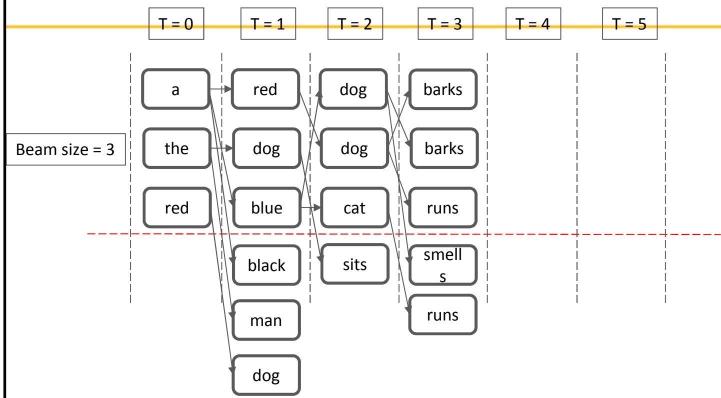


Beam Size Example



Only keep growing top three sentences at each time step

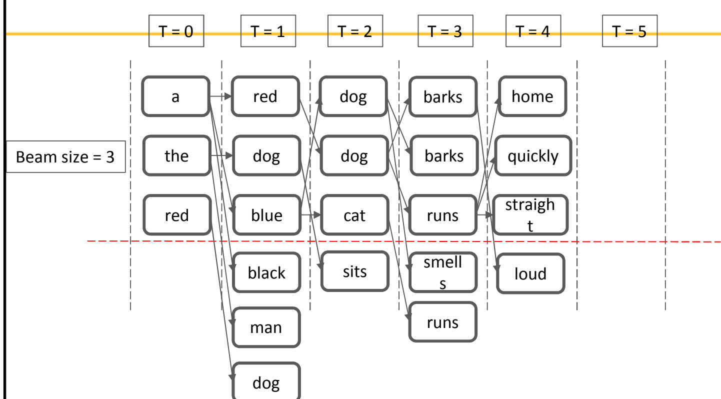
Beam Size Example



Only keep growing top three sentences at each time step

29

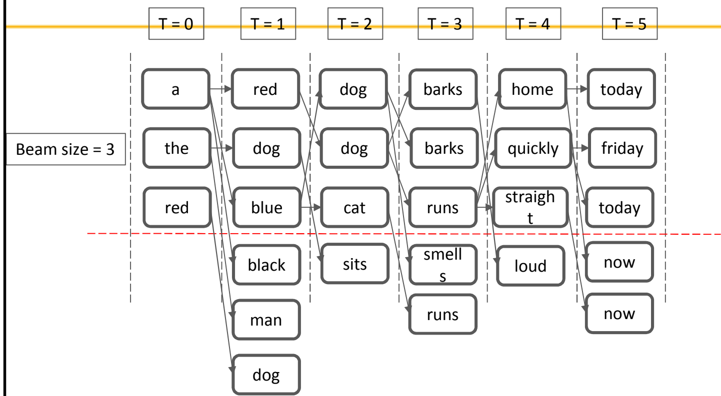
Beam Size Example



Only keep growing top three sentences at each time step

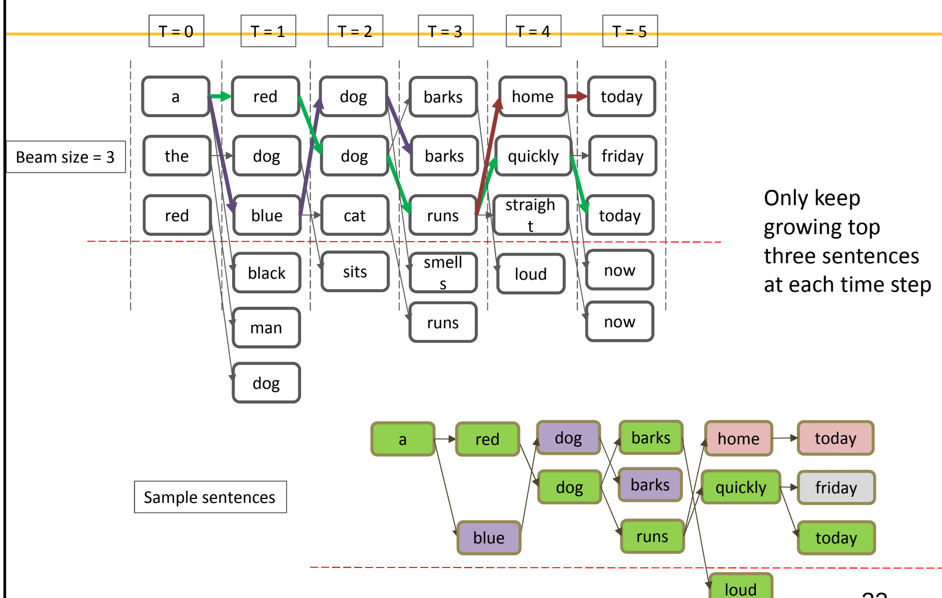
30

Beam Size Example



Only keep growing top three sentences at each time step

Beam Size Example



Only keep growing top three sentences at each time step

Data for Captioning

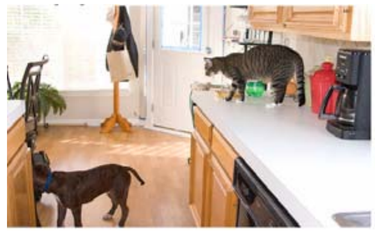
- Flickr8K
  - 8,000 images, from Flickr website, each with five captions
  - http://nlp.cs.illinois.edu/HockenmaierGroup/8k-pictures.html
- Flickr30K
  - 31,783 images, from Flickr website, each with five captions
  - http://shannon.cs.illinois.edu/DenotationGraph/
- MSCOCO
  - 80,000 training images, each with five captions
  - http://mscoco.org/

Captioning Datasets

Amazon mechanical turkers do all labeling
https://www.mturk.com/mturk/welcome

a man riding a bike on a dirt path through a forest.
bicyclist raises his fist as he rides on desert dirt trail.
this dirt bike rider is smiling and raising his fist in triumph.
a man riding a bicycle while pumping his fist in the air.
a mountain biker pumps his fist in celebration.

MSCOCO Dataset



- A pile of wooden boxes filled with fruits and vegetables.
- An assortment of fruit in buckets for sale in a shop.
- An outdoor fruit stand with various types of fruits for sale.
- A display of crates of fruit on a city street.
- There are many crates with fruit and vegetables.

- A cat stands on a counter while a dog stands on the floor.
- A cat on the kitchen counter is looking down at a dog.
- A cat is looking at a dog rummage in the garbage.
- A cat on the counter and a dog on the ground in the kitchen.
- A cat stalking a dog on the kitchen floor.

How to Tell How Well Models are Working?
BLEU & METEOR Scores

- Popular metrics : BLEU (BiLingual Evaluation Understudy) and METEOR (Metric for Evaluation of Translation with Explicit Order)
- BLEU (B1,B2,B3,B4) uses *n*-gram (*n*=1,2,3,4) comparisons between sentences to evaluate the precision and penalizes sentences shorter than reference sentence
- Precision is the sum of ratio of *n*-gram matches and total *n*-grams
- Longer *n*-grams account for the fluency and have higher correlation with human judgement
- METEOR is similar to BLEU except that it considers paraphrased sentences and synonymous words and phrases as matches

Sentence Comparison

Prediction: Picture of a man
Ground truth: Portrait of a man

unigrams: {picture, of , a, man} {portrait, of, a, man}
unigram matches: 3 Precision (B-1): ¾

bigrams: {picture of, of a, a man} {portrait of, of a, a man}
bigram matches: 2 Precision(B-2): 2/3

trigrams: {picture of a, of a man} {portrait of a, of a man}
trigram matches: 1 B-3: ½

4-grams: {picture of a man} {portrait of a man}
4-gram matches: 0 B-4: 0

BLEU Calculation

- For image I_i, evaluate generated caption c_i, given a set of ground truth captions $S_i=\{s_{i1}, \ldots, s_{im}\}$.
- The number of times an *n*-gram, w_k occurs in a sentence s_{ij} or caption c_i is $h_k(s_{ij})$ or $h_k(c_i)$.

Sum over all generated captions

Sum over all *n*-grams in both c_i and s_{ij}.

Lesser of two args

\# times k^{th} *n*-gram occurs in c_i.

Max # times *n*-gram occurs in each of the *m* reference sentences, s_{ij}.

$$CP_n(C,S) = \frac{\sum_i \sum_k \min(h_k(c_i), \max_{j \in m} h_k(s_{ij}))}{\sum_i \sum_k h_k(c_i)}$$

Single precision value calculated for each *n*-gram

Turns into percentage of *n*-grams in each generated sentence which occur in one of the reference sentence.

BLEU Calculation

- CP_n is a precision score and favors shorter generated captions and/or longer ground truth captions.
- A brevity penalty is used, where l_C / l_S is the length of prediction / ground truth caption.
- When multiple ground truth captions used, the closest length, l_{Si} to l_C is used.

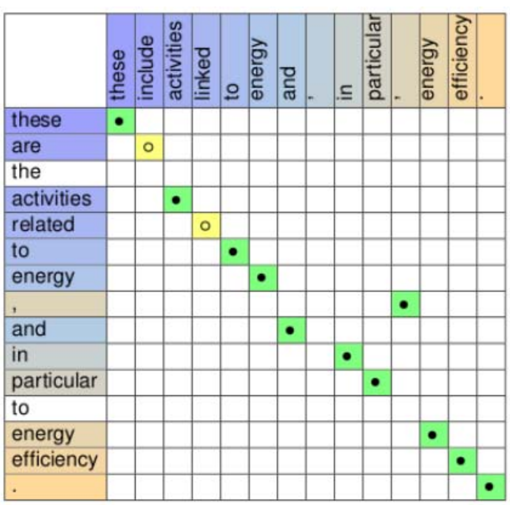
$$b(C,S) = \begin{cases} 1 & \text{if } l_C > l_S \\ e^{1-l_S/l_C} & \text{if } l_C \leq l_S \end{cases},$$

- To combine all individual CP_n scores into a single BLEU score, use geometric mean:

$$BLEU_N(C,S) = b(C,S)\exp\left(\sum_{n=1}^{N} w_n \log CP_n(C,S)\right)$$

Note: for BLEU, w_n is typically a constant such as 1

Sample METEOR Output



METEOR uses exact alignment matching of tokens. However, tokens use WordNet [Miller '95] synonyms, stemmed tokens*, and paraphrases.

P:	0.897
R:	0.907
Frag:	0.514
Score:	0.440

*the words "fishing", "fished", and "fisher" are all stemmed to the same root word, "fish".

BLEU, Meteor, CIDEr, ROUGE

Consensus using TF-IDF

More for text summarization

- If you use the Microsoft COCO caption evaluation tool, you get all four which evaluate how close your generated caption is to the ground truth captions.
  - https://github.com/tylin/coco-caption ← Has detail calculation of each metric.
- **BLEU**: K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics, 2002.
- **Meteor**: S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, volume 29, pages 65–72, 2005.
- **CIDEr**: R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4566–4575, 2015.
- **ROUGE**: C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: Proceedings of the ACL-04 workshop, volume 8. Barcelona, Spain, 2004.

Human Generated Caption vs. Ground Truth Captions

METEOR generally believed to correlate best with how a human would rate captioning performance.

Since generated caption needs to match only one or more of ground truth captions, having 40 captions per image gives better results.

5 captions per image

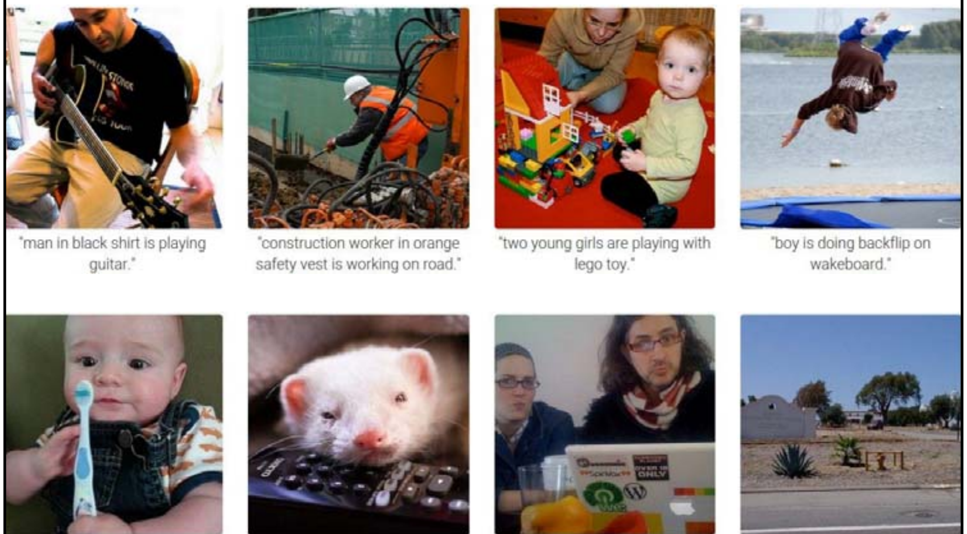
40 captions per image

Metric Name	MS COCO c5	MS COCO c40
BLEU 1	0.663	0.880
BLEU 2	0.469	0.744
BLEU 3	0.321	0.603
BLEU 4	0.217	0.471
METEOR	0.252	0.335
ROUGE$_L$	0.484	0.626
CIDEr-D	0.854	0.910

Much easier to match 1-gram vs. 4-gram

METEOR hardest, and CIDEr hardest to get high scores.

Chen et al. 2015



"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."

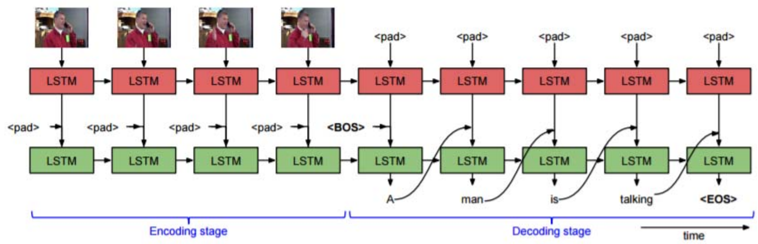
Karpathy'15

Video Data for Captioning

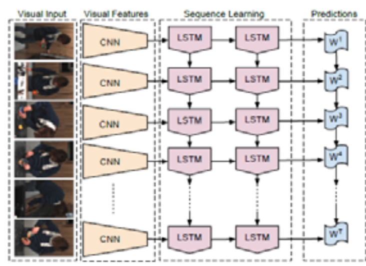
MSVD: Microsoft Video Description Dataset
MSR-VTT: Microsoft Research -Video to Text
M-VAD: Movie description dataset M-VAD

	MSVD	MSR-VTT	M-VAD
#sentences	80,827	200,000	54,997
#sent. per video	∼42	20	∼1-2
vocab. size	9,729	24,282	16,307
avg. length	10.2s	14.8s	5.8s
#train video	1,200	6,513	36,921
#val. video	100	497	4,651
#test video	670	2,990	4,951

Image and Language Applications

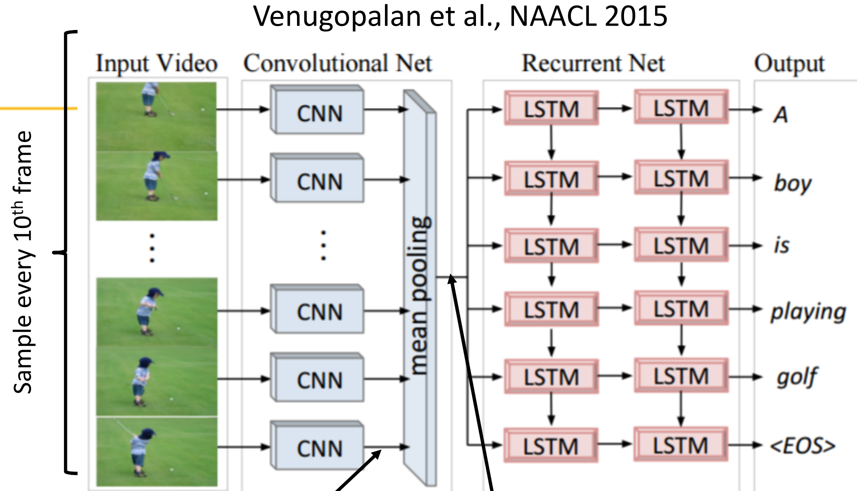


Venugopalan, 2015



Donahue, et al., 2015

Venugopalan et al., NAACL 2015



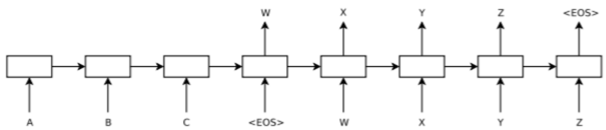
AlexNet 4K FC7

Mean pooling over f frames $V = \frac{1}{f}\sum_{i=1}^{f}[v_1^i, v_2^i, \cdots, v_{4096}^i]$

Pre-train on alternate caption datasets, fine tune to your dataset

The Sequence to Sequence Learning with Neural Networks paper

- In 2014, Sutskever, Vinyals, and Le (from Google) showed that a simple encoder-decoder framework was just as good as sophisticated Statistical Machine Translation (SMT) systems, and almost as good as SMT systems paired with nnets.



Sutskever et al., NIPS '14

The Sequence to Sequence Learning with Neural Networks paper- Performance on EMT'14 English to French test set (ntst14):

12M sentences consisting of 348M French words and 304M English words

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**

- Used separate LSTM for encoding and decoding
- Used 4-layer LSTM,1000 units
- Reversing words in input sentence help!

English to French: So for example, instead of mapping the sentence "Hello my friend" to the sentence "Bonjour mon ami", map "friend my Hello" to "Bonjour mon ami".

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [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

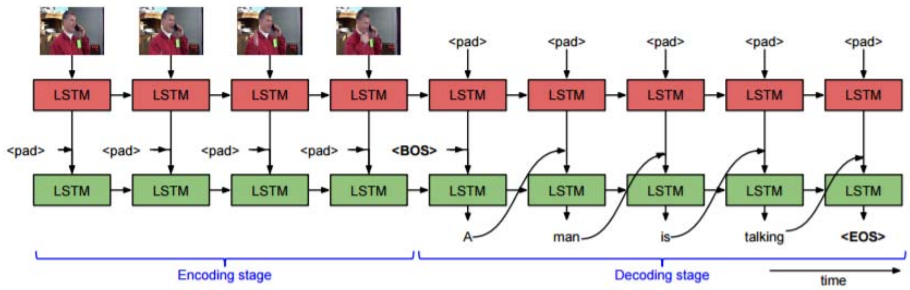
Note: within 0.5 BLEU score!

Note: state-of-the art [9] is a finely tuned phrase based SMT specifically for the ntst14 test set.

Sutskever et al., NIPS '14

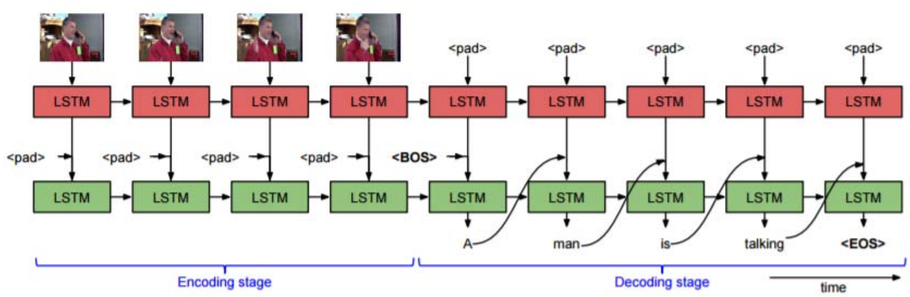
Video Captioning

SV2T, Venugopalan, 2015



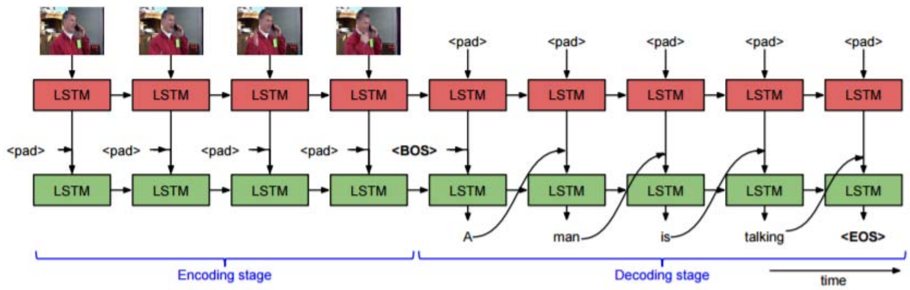
- Single LSTM for both encode and decode state.
- LSTM identical to what we talked about in class.
- Two layer LSTM, 1000 hidden units each

Back to S2VT (Venugopalan ICCV'15)



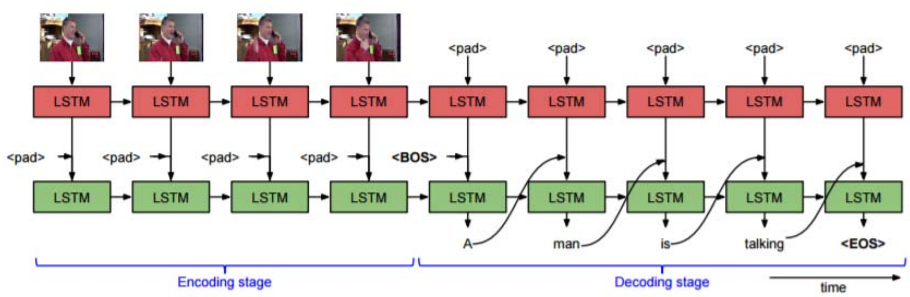
- First LSTM learns video concepts
- Second LSTM concentrates on language details.

Back to S2VT (Venugopalan ICCV'15)



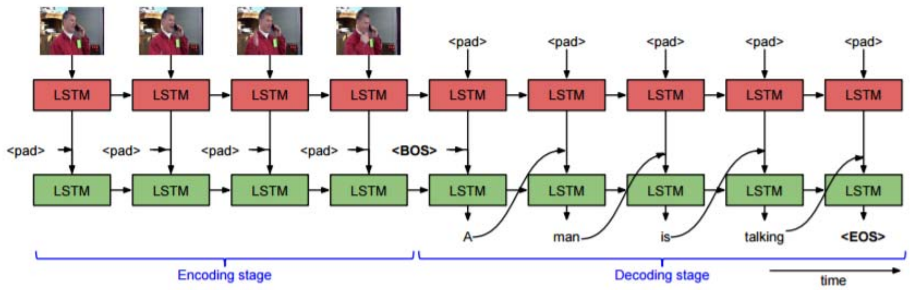
- Embedding layers used to convert both images and words to 500 dimensions each.
  - Remove last FC layer and insert 500 dim FC layer for AlexNet and VGGNet- jointly learn new weights with LSTM weights during backprop.
  - One hot encode words get converted to 500 via a *DictLength*×500 matrix

Back to S2VT (Venugopalan ICCV'15)



- Optical flow maps also tried.
  - Center Δx and Δy around 128.
  - Scale so flow values fall between 0:255
  - Calculate magnitude and use as 3rd channel
  - Pretrain using flow CNN from UCF101 video dataset

Back to S2VT (Venugopalan ICCV'15)



- RGB and flow merged with late fusion
  - At each timestep RGB and flow makes work prediction
  - Use weighted sum of both for final word probability

Three Video Datasets

	MSVD	MPII-MD	MVAD
#-sentences	80,827	68,375	56,634
#-tokens	567,874	679,157	568,408
vocab	12,594	21,700	18,092
#-videos	1,970	68,337	46,009
avg. length	10.2s	3.9s	6.2s
#-sents per video	≈41	1	1-2

MSVD Dataset

Traditional NLP

Mean pool are all around Venugopalan NAACL'15

[43] is the Yao Spatial-temporal attention, ICCV'15

Model	METEOR	
FGM [36]	23.9	(1)
Mean pool		
- AlexNet [39]	26.9	(2)
- VGG	27.7	(3)
- AlexNet COCO pre-trained [39]	29.1	(4)
- GoogleNet [43]	28.7	(5)
Temporal attention		
- GoogleNet [43]	29.0	(6)
- GoogleNet + 3D-CNN [43]	29.6	(7)
S2VT (ours)		
- Flow (AlexNet)	24.3	(8)
- RGB (AlexNet)	27.9	(9)
- RGB (VGG) random frame order	28.2	(10)
- RGB (VGG)	29.2	(11)
- RGB (VGG) + Flow (AlexNet)	29.8	(12)

SMT is a statistical machine translation (traditional NLP features)

In place of an encoding stage, Visual-Labels uses a variety of visual features such as object detectors and scene classifiers

MPII-MD Dataset

Approach	METEOR
SMT (best variant) [28]	5.6
Visual-Labels [27]	7.0
Mean pool (VGG)	6.7
S2VT: RGB (VGG), ours	7.1

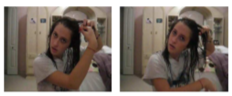
MSM-VAD Dataset

Approach	METEOR
Visual-Labels [27]	6.3
Temporal att. (GoogleNet+3D-CNN) [43] [4]	4.3
Mean pool (VGG)	6.1
S2VT: RGB (VGG), ours	6.7

Correct descriptions.

S2VT: A man is doing stunts on his bike.

S2VT: A herd of zebras are walking in a field.



S2VT: A young woman is doing her hair.

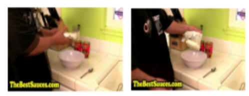
S2VT: A man is shooting a gun at a target.

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

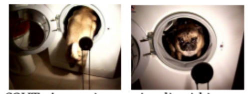
S2VT: A man is cutting a piece of a pair of a paper.

S2VT: A cat is trying to get a small board.



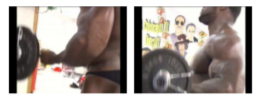
S2VT: A man is spreading butter on a tortilla.

Irrelevant descriptions.



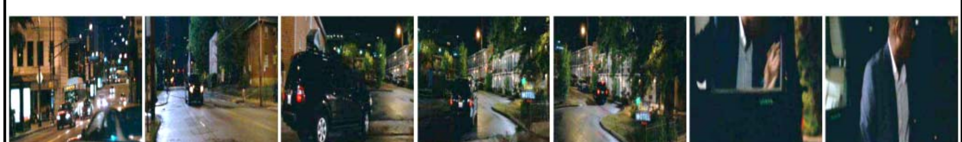
S2VT: A man is pouring liquid in a pan.

S2VT: A polar bear is walking on a hill.



S2VT: A man is doing a pencil.

S2VT: A black clip to walking through a path.



(1) (2) (3) (4) (5) (6a) (6b)

Temporal Attention (GNet+3D-conv$_{att}$): (1) At night , SOMEONE and SOMEONE step into the parking lot.
(2) Now the van drives away.
(3) They drive away.
(4) They drive off.
(5) They drive off.
(6) At the end of the street , SOMEONE sits with his eyes closed.

S2VT (Ours): (1) Now, the van pulls out a window and a tall brick facade of tall trees . a figure stands at a curb.
(2) Someone drives off the passenger car and drives off.
(3) They drive off the street.
(4) They drive off a suburban road and parks in a dirt neighborhood.
(5) They drive off a suburban road and parks on a street.
(6) Someone sits in the doorway and stares at her with a furrowed brow.

DVS: (1) Now , at night , our view glides over a highway its lanes glittering from the lights of traffic below.
(2) Someone's suv cruises down a quiet road.
(3) Then turn into a parking lot .
(4) A neon palm tree glows on a sign that reads oasis motel.
(5) Someone parks his suv in front of some rooms.
(6) He climbs out with his briefcase , sweeping his cautious gaze around the area.

This is Yao Spatial-temporal attention, ICCV'15

This is S2VT

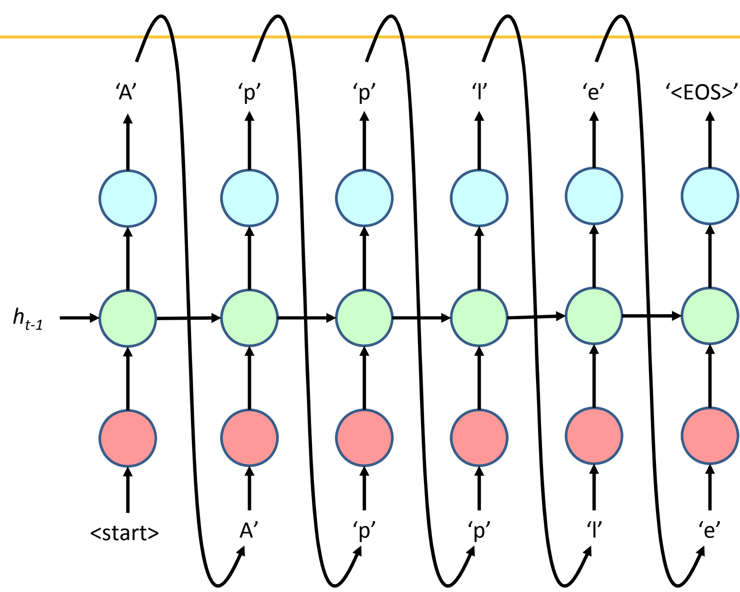
This is ground truth

CNN as Vector Representation

- Fully connected layers are excellent descriptors of the input image!
- For example, you can pass images through a pre-trained CNN, then take the output from a FC layer as input to a SVM classifier. (image2vec)

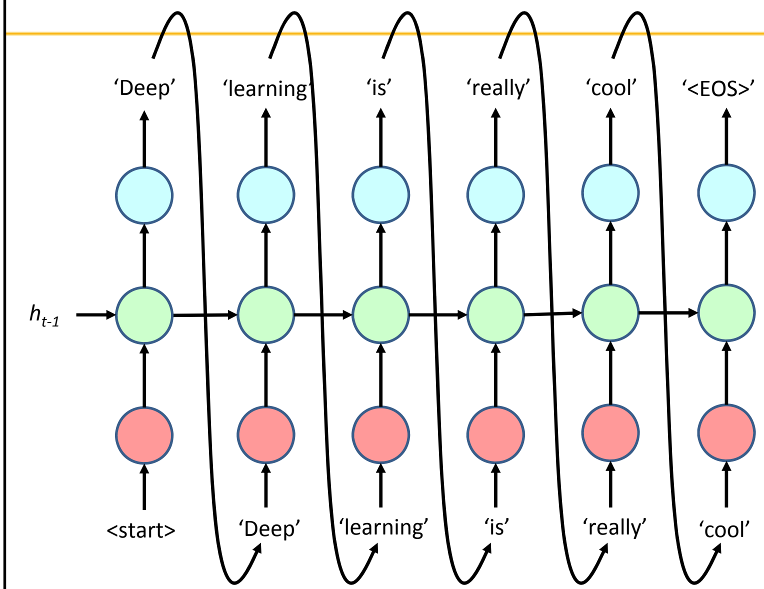
- Images in this vector space generally have the property that similar images are close in this latent representation.

Recurrent Networks for Character Prediction



For this to work, we need to represent characters as some latent vector numerical representation.

Recurrent Networks for Word Prediction



For this to work, we need to represent words as some latent vector numerical representation.

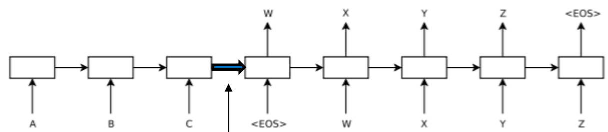
Word2vec

- In the simplest form, we can start with a one-hot encoded vector of all words, and then learn a model which converts to a lower dimensional representation.
- Word2vec, glove, and skip-gram are popular metrics which encode words to a latent vector representation (~300 dimensions).
- Now we have a way to represent images, characters, and words as vectors.

Sent2vec

- In the English to French translation, we have:

French sentence



English sentence

Sutskever et al., NIPS ‘14

…but wait, this point in the RNN is a representation (sent2vec) of all the words in the English sentence!

- Now we have a way to represent images, characters, words, and sentences as vectors…can extend to paragraphs and documents…

What about Video2vec????

https://arxiv.org/pdf/1412.0767.pdf

Learning Spatiotemporal Features with 3D Convolutional Networks

Du Tran[1,2], Lubomir Bourdev[1], Rob Fergus[1], Lorenzo Torresani[2], Manohar Paluri[1]
[1]Facebook AI Research, [2]Dartmouth College
{dutran,lorenzo}@cs.dartmouth.edu {lubomir,robfergus,mano}@fb.com

Abstract

We propose a simple, yet effective approach for spatiotemporal feature learning using deep 3-dimensional convolutional networks (3D ConvNets) trained on a large scale supervised video dataset. Our findings are three-fold: 1) 3D ConvNets are more suitable for spatiotemporal feature learning compared to 2D ConvNets; 2) A homogeneous architecture with small 3 × 3 × 3 convolution kernels in all layers is among the best performing architectures for 3D ConvNets; and 3) Our learned features, namely C3D (Convolutional 3D), with a simple linear classifier outperform state-of-the-art methods on 4 different benchmarks and are comparable with current best methods on the other 2 benchmarks. In addition, the features are compact: achieving

ing, and retrieving tasks much more scalable; (iii) it needs to be **efficient** to compute, as thousands of videos are expected to be processed every minute in real world systems; and (iv) it must be **simple** to implement. Instead of using complicated feature encoding methods and classifiers, a good descriptor should work well even with a simple model (e.g. linear classifier).

Inspired by the deep learning breakthroughs in the image domain [24] where rapid progress has been made in the past few years in feature learning, various pre-trained convolutional network (ConvNet) models [16] are made available for extracting image features. These features are the activations of the network's last few fully-connected layers which perform well on transfer learning tasks [47, 48]. However, such image based deep features are not directly suitable for

C3D

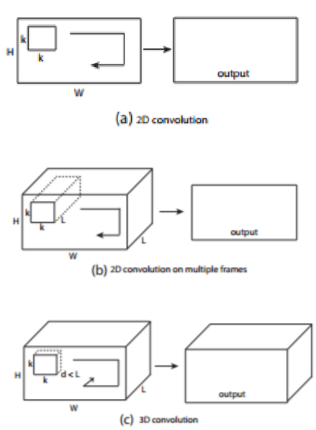
Tran et al. "Learning Spatiotemporal Features with 3D Convolutional Networks", ICCV 2015.

- Rather than learn a single vector (e.g. FC7), introduced a spatio-temporal video feature representation using deep 3D ConvNets.
- Not the first to propose 3D ConvNets, but first to exploit deep nets with large supervised datasets.
- Models appearance and motion.
- Showed that:
  - 3D ConvNets are better than 2D ConvNets
  - Simple architecture with 3×3×3 filters works very well
  - Learned features are then passed into simple linear classifier to give state-of-the-art results

2D and 3D Convolution

(will still work with *c* channels and *f* frames)

(Similar phenomenon for pooling)



Tran et al., 2015

- 2D conv on a 2D image results in 2D image
- 2D conv on a 3D volume results in 2D image
  - Because filter depth matches volume depth.
- 3D conv on a 3D volume results in 3D volume
  - Preserves spatio-temporal information.

input

3×16×128×171

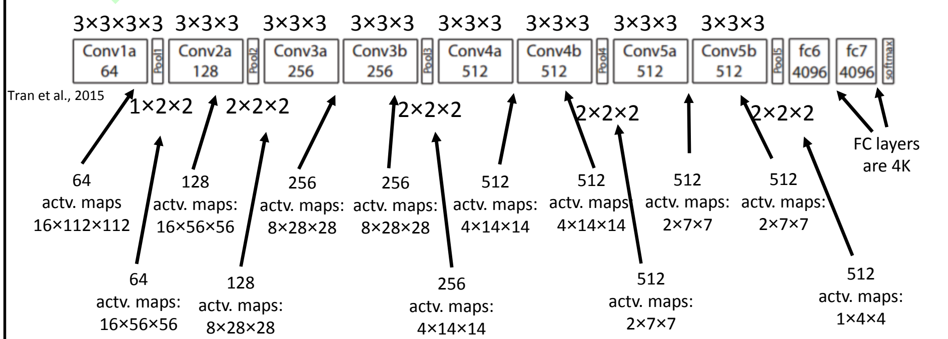
augmentation

3×16×112×112

Full C3D Architecture (Tran et al. ICCV'15)

Split video into 16-frame clips with 8-frame overlap

All convolution uses zero pad and stride=1



 67

input

3×16×128×171

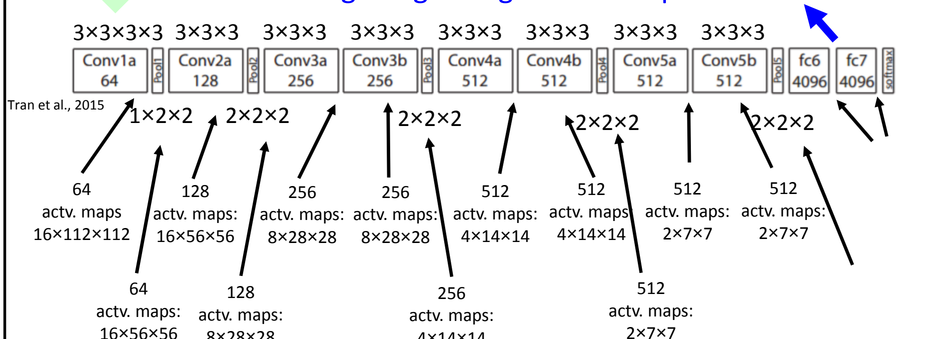
augmentation

3×16×112×112

Full C3D Architecture (Tran et al. ICCV'15)

Split video into 8-frame overlap 16-frame clips

When used as a video2vec feature descriptor, take output from all fc6 layers of all clips, and average to get single 4K descriptor of video.



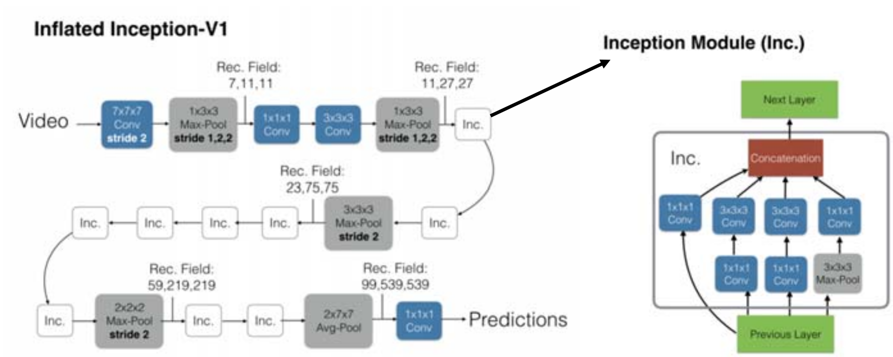
Pu et al., 2017:

The dimensions for features extracted from *pool2*, *pool3*, *pool4* and *pool5* are $28 \times 28 \times N/2 \times 128$, $14 \times 14 \times N/4 \times 256$, $7 \times 7 \times N/8 \times 512$ and $4 \times 4 \times N/16 \times 512$, respectively. N is the number of frames of input video. After the convolutional transformation, the dimensions will be all $4 \times 4 \times N/16 \times 512$.

 68

Inflated Inception v1 for Video (I3D)

Filters and Pooling Increased from 2D to 3D



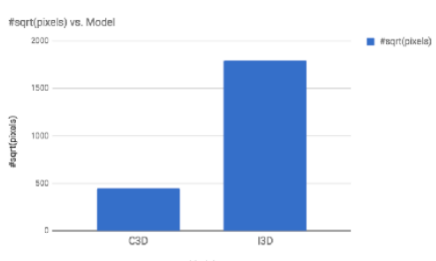
Same fellow who introduced VGG and used optical flow for video

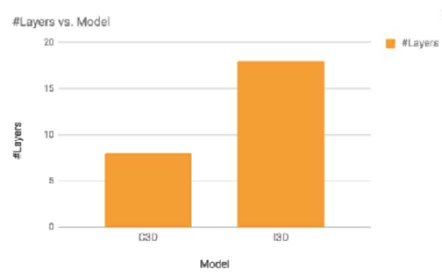
Quo Vadis Action Recognition: a New Model and the Kinetics Dataset. Carreira and Zisserman, CVPR 2017, http://openaccess.thecvf.com/content_cvpr_2017/papers/Carreira_Quo_Vadis_Action_CVPR_2017_paper.pdf

Inflated Inception v1 for Video (I3D)

C3D:
- 8 convolutional layers
- 79M parameters
- Inputs: 112x112, 16-frame clips

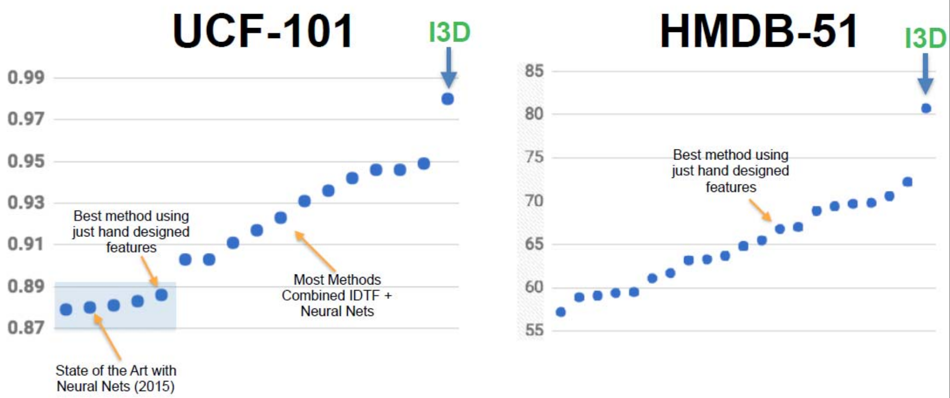
I3D:
- 18 convolutional layers
- 12M parameters
- Inputs: 224x224, 64-frame clips





Quo Vadis Action Recognition: a New Model and the Kinetics Dataset. Carreira and Zisserman, CVPR 2017

Inflated Inception v1 for Video (I3D)



Quo Vadis Action Recognition: a New Model and the Kinetics Dataset. Carreira and Zisserman, CVPR 2017

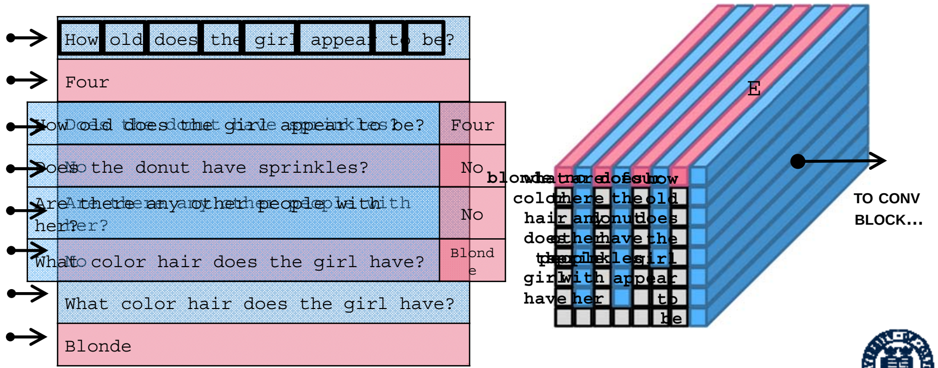
Inflated Inception v1 for Video (I3D)

Architecture	UCF-101				HMDB-51			
	Original	Fixed	Full-FT	Δ	Original	Fixed	Full-FT	Δ
(a) LSTM	81.0	81.6	82.1	-6%	36.0	46.6	46.4	-16.7%
(b) 3D-ConvNet	51.6	76.0	79.9	**-58.5%**	24.3	47.5	49.4	**-33.1%**
(c) Two-Stream	91.2	90.3	91.5	-3.4%	58.3	64.0	58.7	-13.7%
(d) 3D-Fused	89.3	88.5	90.1	-7.5%	56.8	59.0	61.4	-10.6%
(e) Two-Stream I3D	**93.4**	**95.7**	**96.5**	-47.0%	**66.4**	**74.3**	**75.9**	-28.3%

Note: I3D always started from an ImageNet model.
Original: train on UCF-101/HMDB-51
Fixed: train on miniKinetics, tune last layer on UCF-101/HMDB-51
Full-FT: train on miniKinetics, fine tune the entire network on UCF-101/HMDB-51
3D fused does fuses the cone net outputs from flow and RGB before the FC layers

Quo Vadis Action Recognition: a New Model and the Kinetics Dataset. Carreira and Zisserman, CVPR 2017

‘COLouring’ dialogue with convolutions



Convolutions for language: Kalchbrenner et al. (2014), Pham et al. (2016), Bai, et al. (2018) Slide credit: Daniela Massiceti, University of Oxford

Thank you!!

Ray Ptucha
rwpeec@rit.edu

https://www.rit.edu/mil