Caltech

Image Captioning and Generation From Text

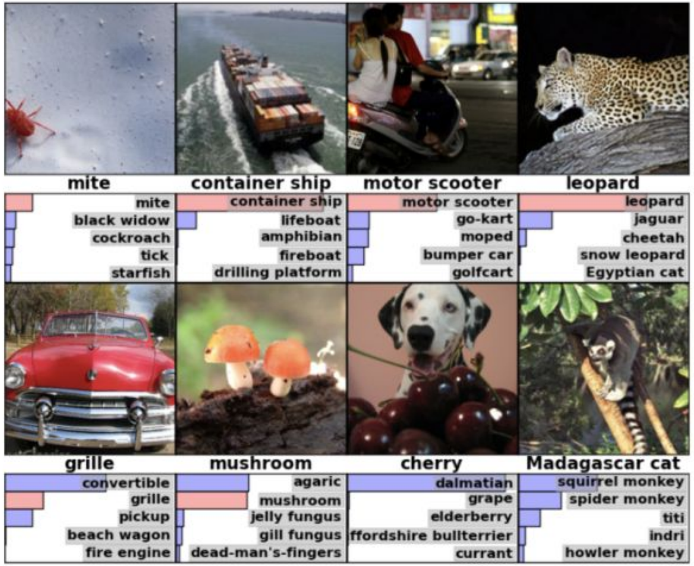
Presented by: Tony Zhang, Jonathan Kenny, and Jeremy Bernstein
Mentor: Stephan Zheng

CS159 Advanced Topics in Machine Learning: Structured Prediction

California Institute of Technology

Computer Vision Tasks

- Low-level: recognition
  - Object detection: specific, well-constrained conditions
  - Segmentation
  - Recognition: pre-specified learning object classes
- High-level: **scene understanding**
  - Contextual meanings
  - Object dependencies
- Datasets
  - ImageNet (14M)
  - Microsoft Common Objects in Context (2.5M)
  - CIFAR10/100 (60k)



CV Challenges

- Low-level: recognition
  - Most tasks are easy
  - Compared to humans
    - Strengths: classifying sub-classes
    - Weaknesses: small / distorted (e.g. through filters) objects
- High-level: **scene understanding**
  - Relative to humans: not comparable
  - Current solutions: use existing tools and combine together
  - Unsolved
- Metrics
  - Accuracy relatively meaningless (does not reflect key challenges)
  - Test set has been well exploited

Natural Language Processing Tasks

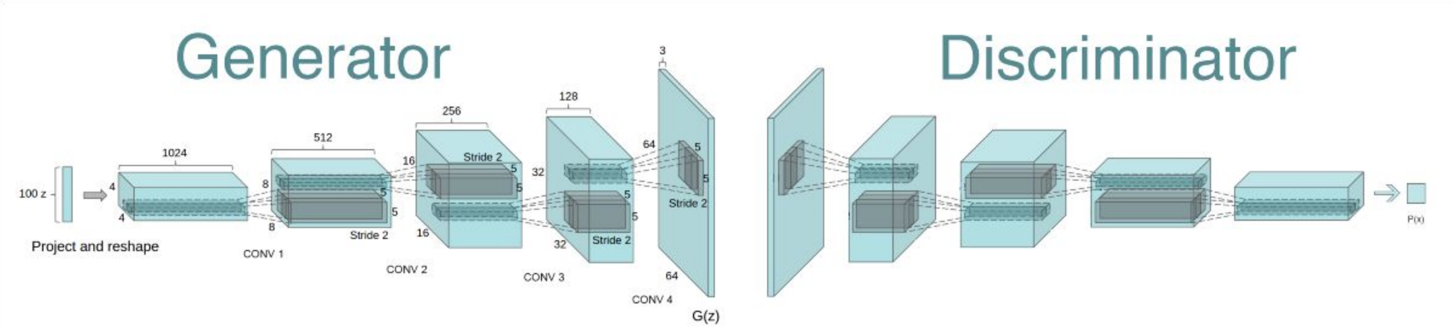
- Low-level: syntax
  - Part-of-speech tagging
  - Parsing (grammatical analysis)
- High-level: semantics, discourse, speech
  - Understanding
  - Generation
  - Tasks: translation, segmentation, speech recognition
- Datasets
  - Variety depending on context (e.g. sentiment analysis, classification, clustering)

NLP metrics for evaluation

- Automatic favored over manual evaluations
- Formative (mostly automatic) and summative (mostly manual)
- Intrinsic (evaluated based on system) and extrinsic (evaluated on task external to system)
- Component vs end-to-end
- Example: BLEU for translation (precision based on unigrams / bigram / trigram)
- **Challenge**: developing more human-like automatic metrics is critical
    - Requires better understanding of language structures itself
    - Current metric: correlation with human scores

Image Generation

- **Low-level**: generating similar digits or images with selective objects
- **High-level**: novel images with complex distributions scenery
  - With NLP: Caption -> NLP understanding -> generation
- Metrics
  - No good metrics for evaluation
  - A discriminator network (GAN)?
  - Need to develop better <u>understanding</u> of natural images’ properties



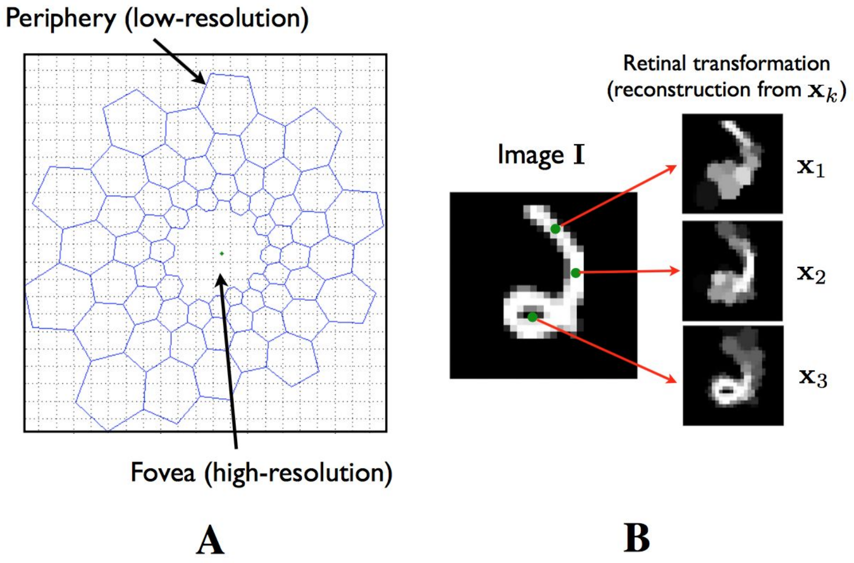
Attention Mechanisms

- Loosely ‘inspired’ by human attention (which we know almost nothing about)
- Advantages
  - Enhances complex long-range dependencies on top of LSTM
  - Allow better understanding of trained model
  - Allows network to refer back to input sequence, instead of forcing it to encode all information into one fixed-length vector
- Long (in recent deep learning literature) history
  - Learning to combine foveal glimpses with a third-order Boltzmann machine (Larochelle & Hinton, 2010)
  - Neural machine translation by jointly learning to align and translate (Bahdanau & Bengio, 2015)
  - Recent advances: applied to RNN for NLP & CV

Learning to combine foveal glimpses with a third-order Boltzmann machine

Larochelle & Hinton, 2010

- Boltzmann machine with third-order connections that learn how to accumulate information about a shape over several fixations
- The model uses a ‘retina’ that only has enough high resolution pixels to cover small area
- Must learn sequence of fixation
- Performance: comparable to existing models

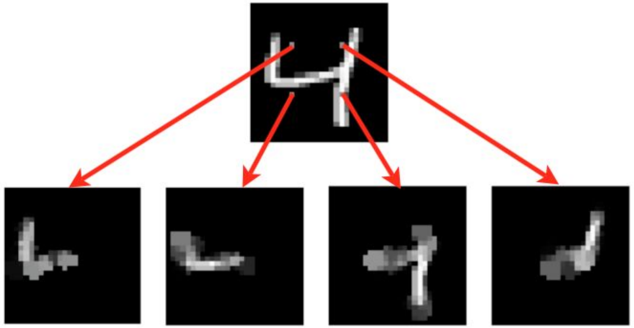


Learning to combine foveal glimpses with a third-order Boltzmann machine

Larochelle & Hinton, 2010

- Boltzmann machine with third-order connections that learn how to accumulate information about a shape over several fixations
- The model uses a ‘retina’ that only has enough high resolution pixels to cover small area
- Must learn sequence of fixation
- Performance: comparable to existing models

Experiment 1: MNIST with 4 fixations



Model	Error
NNet+RBM [22]	3.17% (± 0.15)
SVM [21]	3.03% (± 0.15)
Multi-fixation RBM (hybrid)	3.20% (± 0.15)
Multi-fixation RBM (hybrid-sequential)	2.76% (± 0.14)

Neural machine translation by jointly learning to align and translate

Bahdanau & Bengio, 2015

- Base: conventional RNN encoder-decoder
- Removes bottleneck on encoded vector length
- Model automatically soft-search for parts of source sentence relevant to predicting target word
- Must learn sequence of fixation
- Attention results make intuitive sense
- Performance: more robust to long input length, outperforms equivalent RNN

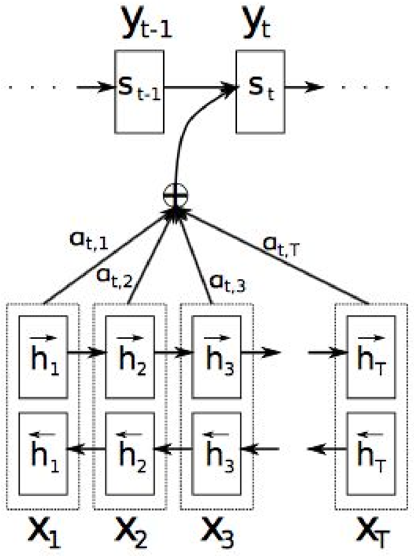


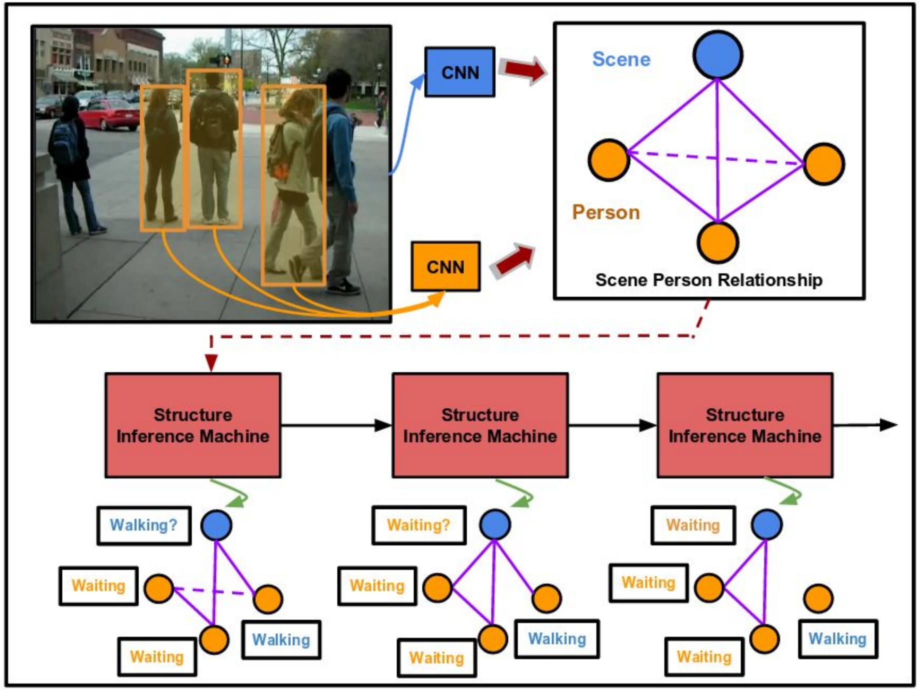
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu⋆ KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba† JIMMY@PSI.UTORONTO.CA
Ryan Kiros† RKIROS@CS.TORONTO.EDU
Kyunghyun Cho⋆ KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville⋆ AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov†∗ RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel†∗ ZEMEL@CS.TORONTO.EDU
Yoshua Bengio⋆∗ YOSHUA.BENGIO@UMONTREAL.CA
⋆ Université de Montréal, † University of Toronto, ∗ CIFAR

Computer Vision and Scene Understanding

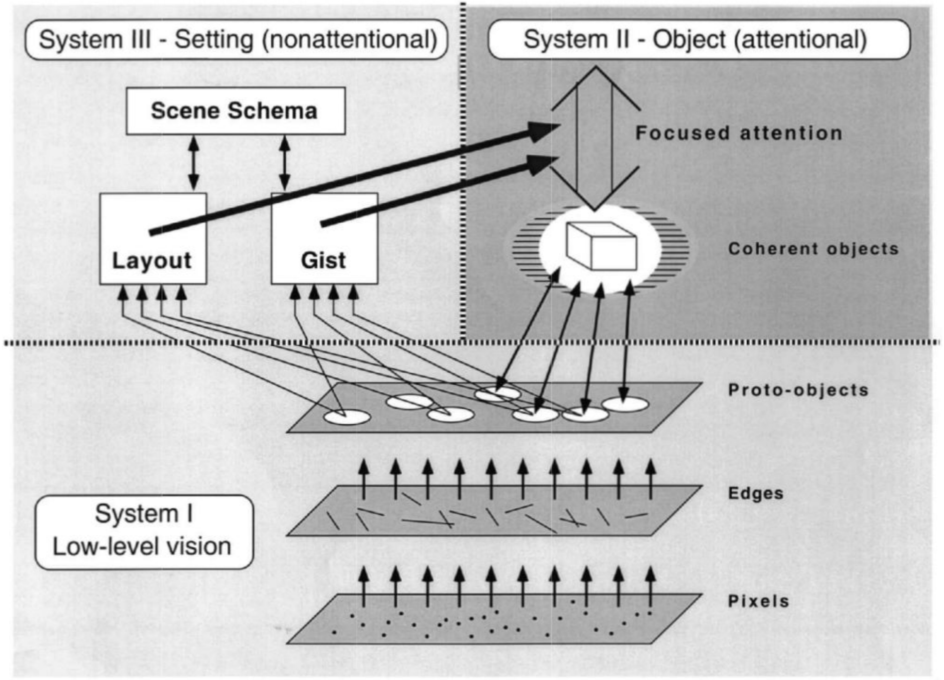
- Higher-order visual understanding requires the recognition of the individual objects in a scene, and the complicated relationships that may exist between them.
- This may involve the recognition and determination of image cues, spatial distance, object motion, and object properties.



Deng, Zhiwei, et al. "Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

Neuroscience-inspired Attention Mechanism

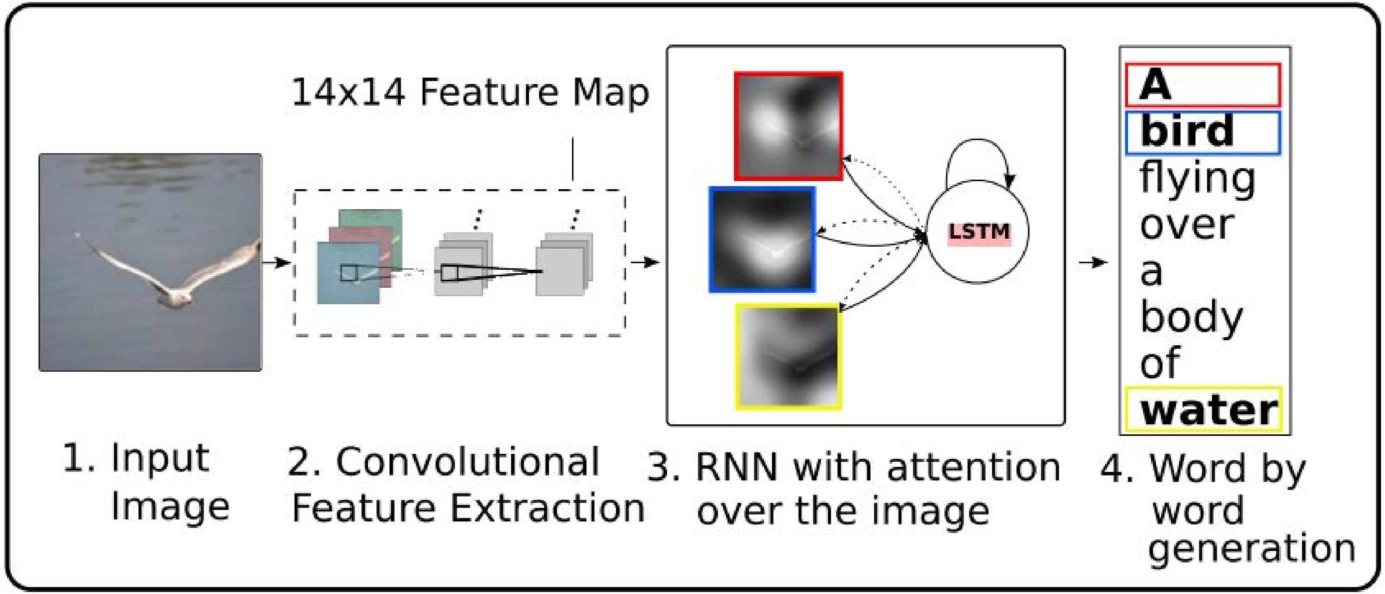
- Scene understanding in humans presents an environment that is very detailed, and where all objects are presented simultaneously.
- This is done by a lower-level “focused attention” mechanism that observes objects of interest one at a time.
- Systems in the higher-level visual pathway then aggregate these results and make it seem like all objects are presented at the same time.



Rensink, Ronald A. "The dynamic representation of scenes." *Visual cognition* 7.1-3 (2000): 17-42.

Caption Generation from Images

1. Determine what objects are in an image and which are important.
2. Determine relationships (both simple and complex) between objects.
3. Express the relationships in natural language.



Model Architecture: Encoder

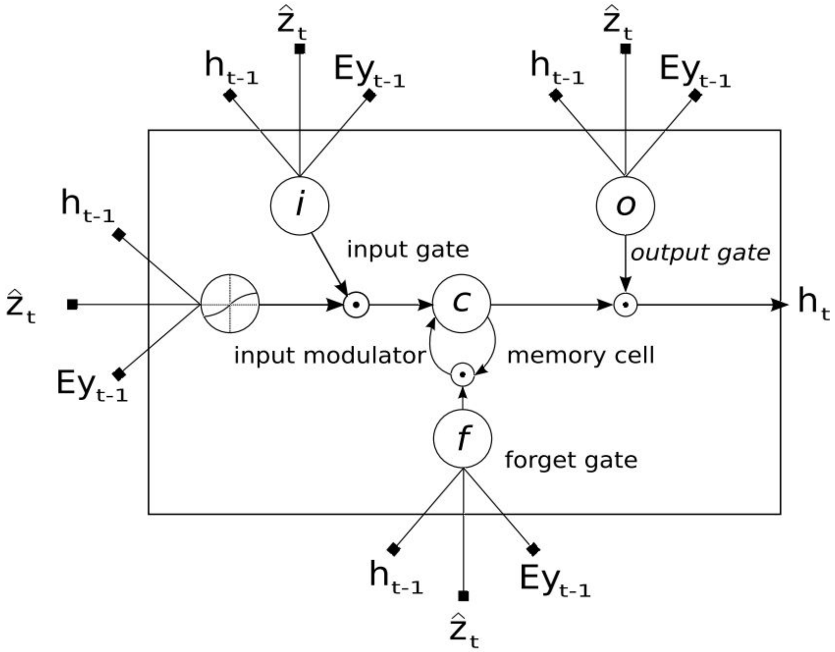
- Convolutional neural network, **Oxford VGGnet**, pre-trained on ImageNet.
- No additional fine-tuning of the CNN.
- Feature/annotation vectors for the decoder taken from the lower-level, layer 4, before the maxpool (14 * 14 * 512).
- Produces 14 * 14 = 196 annotation vectors, each with 512 dimensions.
- Lower-level features allow decoder to focus on parts of the image by selecting subsets of feature vectors.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

Model Architecture: Decoder

- Long short-term memory network, generates one caption word (**y**) per timestep.
- Depends on context vector ($\mathbf{z}_t$), previous hidden state ($\mathbf{h}_{t-1}$) and previously generated caption words ($\mathbf{y}_{t-1}$).
- **E** is a word embedding matrix based on a vocabulary of size K.
- The context vector ($\mathbf{z}_t$) is a determined from the 196 annotation vectors.



The Attention Model

- The context vector ($\mathbf{z}_t$) captures visual information associated with relevant locations in the input image.
- It is a dynamical representation that can change at each timestep.
- The context vector ($\mathbf{z}_t$) is constructed from the 196 annotation/feature vectors ($\mathbf{a}_i$, from the CNN) using an attention model (multilayer perceptron followed by special attention function ϕ).
- The attention model assigns a weight ($\boldsymbol{\alpha}_{ti}$) to each annotation vector ($\mathbf{a}_i$), based on the annotation vector and previous hidden state ($\mathbf{h}_{t-1}$). This relates how much focus to put on those annotation vectors when generating the next caption ($\mathbf{y}$). The context vector ($\mathbf{z}_t$) is calculated from the annotation vectors ($\mathbf{a}_i$) and weights ($\boldsymbol{\alpha}_{ti}$) using a special attention function ϕ.

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1}) \quad \alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}.$$

The Attention Model

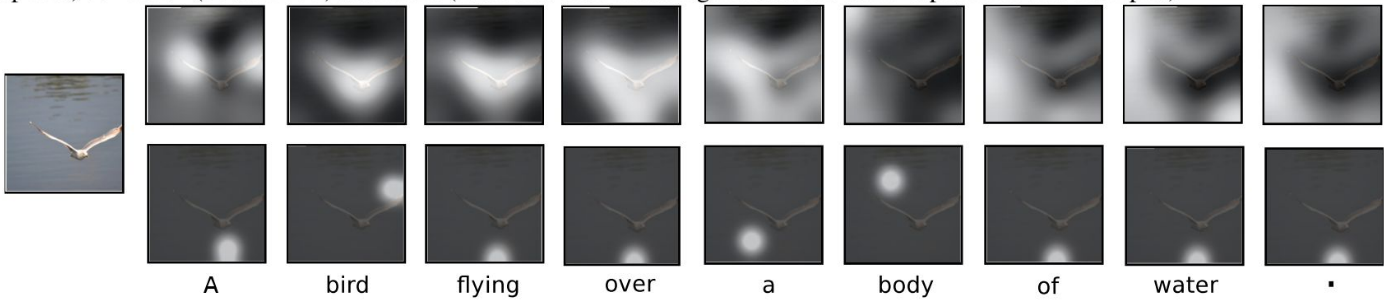
- The context vector ($\mathbf{z}_t$) captures visual information associated with relevant locations in the input image.
- It is a dynamical representation that can change at each timestep.
- The context vector ($\mathbf{z}_t$) is constructed from the 196 annotation/feature vectors ($\mathbf{a}_i$, from the CNN) using an attention model (multilayer perceptron followed by special attention function ϕ).
- The attention model assigns a weight (α_{ti}) to each annotation vector ($\mathbf{a}_i$), based on the annotation vector and previous hidden state ($\mathbf{h}_{t-1}$). This relates how much focus to put on those annotation vectors when generating the next caption ($\mathbf{y}$). The context vector ($\mathbf{z}_t$) is calculated from the annotation vectors ($\mathbf{a}_i$) and weights (α_{ti}) using a special attention function ϕ.

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1}) \quad \alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}. \longrightarrow \hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right)$$

The Attention Model

- The context vector ($\mathbf{z}_t$) captures visual information associated with relevant locations in the input image.
- It is a dynamical representation that can change at each timestep.
- The context vector ($\mathbf{z}_t$) is constructed from the 196 annotation/feature vectors ($\mathbf{a}_i$, from the CNN) using an attention model (multilayer perceptron followed by special attention function ϕ).
- The attention model assigns a weight ($\boldsymbol{\alpha}_{ti}$) to each annotation vector ($\mathbf{a}_i$), based on the annotation vector and previous hidden state ($\mathbf{h}_{t-1}$). This relates how much focus to put on those annotation vectors when generating the next caption ($\mathbf{y}$). The context vector ($\mathbf{z}_t$) is calculated from the annotation vectors ($\mathbf{a}_i$) and weights ($\boldsymbol{\alpha}_{ti}$) using a special attention function ϕ.

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)



The Attention Function ϕ: "Soft" Deterministic

- Take the expectation of the context vector ($\mathbf{z}_t$) from the annotation vector ($\mathbf{a}_i$) and weights ($\boldsymbol{\alpha}_{ti}$), and use that as your context vector.

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$

- This suggests that the context vector ($\mathbf{z}_t$) given by your attention function ϕ is a soft attention weighted annotation vector, rather than any particular specific annotation vector.

$$\phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \sum_i^L \alpha_i \mathbf{a}_i$$

The Attention Function ϕ: "Soft" Deterministic

- Stochastic regularization is introduced using two methods:

1. By default, the annotation weights ($\boldsymbol{\alpha}_{ti}$) at each timestep sum over all *i* to 1.

$$\sum_i \alpha_{ti} = 1$$

Regularization can be introduced by having the weights over all timesteps *t* also approximately sum to 1.

$$\sum_t \alpha_{ti} \approx 1$$

This forces the attention model to pay more equal attention to every location of the image. This leads to improved metrics and more descriptive captions.

The Attention Function ϕ: "Soft" Deterministic

- Stochastic regularization is introduced using two methods:

2. The attention model also includes a scalar β, calculated from the softmax of the previous hidden state.

$$\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$$

The modified soft attention function is given by:

$$\phi\left(\{\mathbf{a}_i\},\{\alpha_i\}\right) = \beta \sum_i^L \alpha_i \mathbf{a}_i$$

This pushes the model to place attention on objects in the image. The model can then be trained using back-propagation and the following objective function:

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2$$

The Attention Function ϕ: "Hard" Stochastic

- Different from soft attention, here you select a single annotation vector (**a**$_i$) at each timestep using the selection variable **s**$_{t,i}$ (which is 1 at the **a**$_i$ of interest and zero for all of the others).

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i}\mathbf{a}_i$$

- Select **s**$_{t,i}$ by sampling from a multinoulli distribution characterized by the annotation weights $\boldsymbol{\alpha}_{ti}$.

$$\tilde{s}_t \sim \text{Multinoulli}_L(\{\alpha_i\})$$

The Attention Function ϕ: “Hard” Stochastic

- Update the model weights by optimizing a variational lower bound on the model output word probability, given the annotation vector (**p(y|a)**).

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \lambda_r (\log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) - b) \frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} + \lambda_e \frac{\partial H[\tilde{s}^n]}{\partial W} \right]$$

- Where λ_r and λ_e are hyperparameters set by cross-validation, H[**s**] is an entropy term based on the multinoulli samples to reduce gradient variance, and **b** is a moving average baseline over image minibatches to reduce gradient variance.

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log p(\mathbf{y} \mid \tilde{s}_k, \mathbf{a})$$

- Also 50% of the time just set the attention location $\mathbf{s}_{t,i}$ to the expected value of the multinoulli distribution.

Output Word Probability: Deep Output Layer

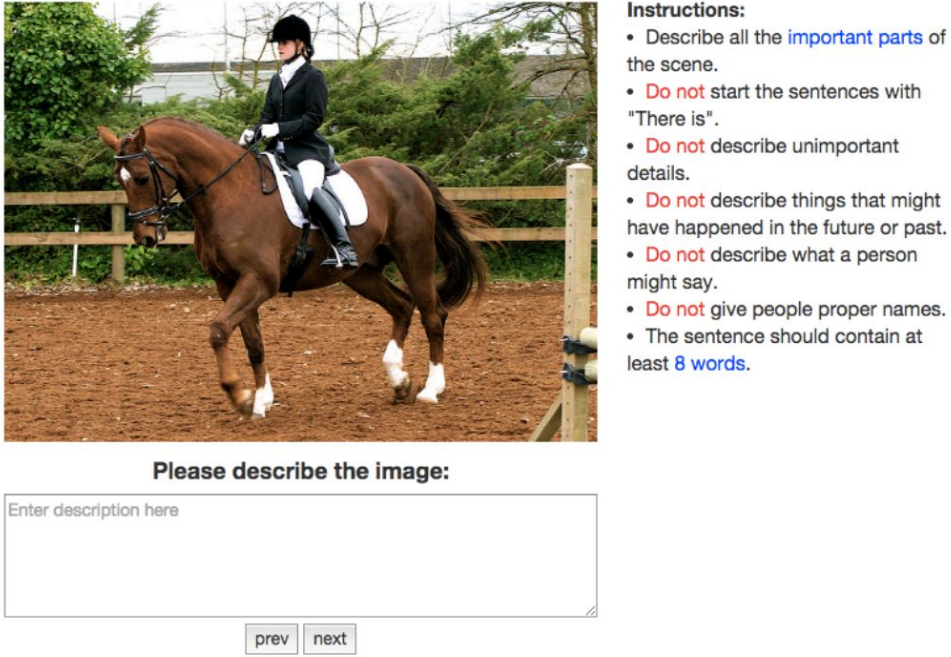
- Generate the output (caption, $\mathbf{y}_t$) probability based on the current context vector ($\mathbf{z}_t$), LSTM state ($\mathbf{h}_t$), and the previously generated caption word ($\mathbf{y}_{t-1}$).
- Do this using an output layer:

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t))$$

- Where **L** and **E** are trained weight matrices.

Training Dataset: Microsoft COCO

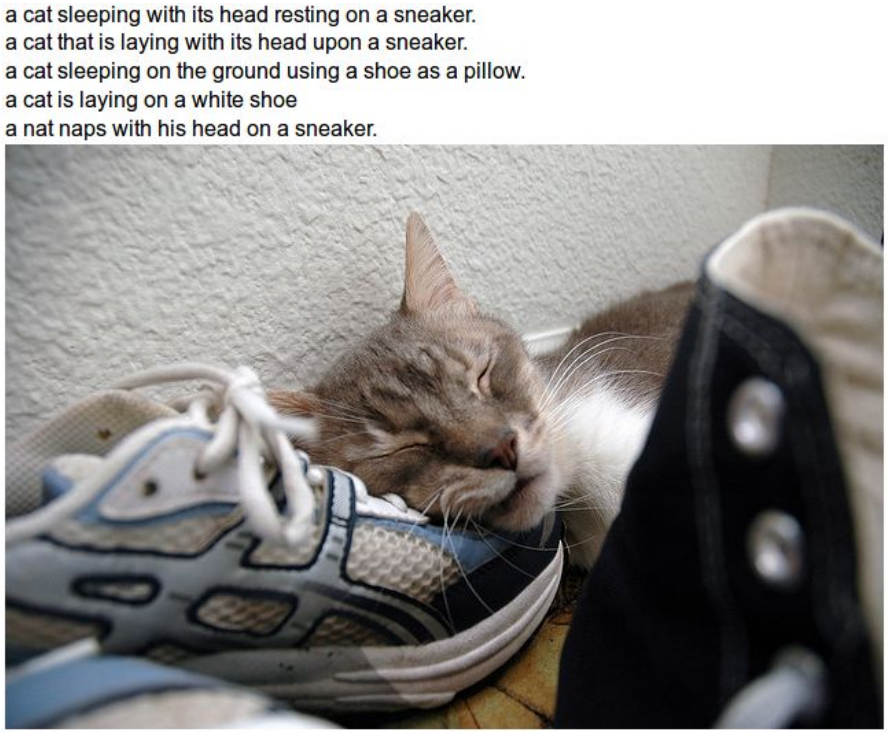
- Microsoft **COCO**: **C**ommon **O**bjects in **Co**ntext
- Images (from **Flickr**) with multiple objects in a naturalistic context.
- **82,783 images** (88% training, 6% validation, 6% testing), each with at least five human generated captions each (using Amazon Mechanical Turk).



Chen, Xinlei, et al. "Microsoft COCO captions: Data collection and evaluation server." *arXiv preprint arXiv:1504.00325* (2015).

Training Dataset: Microsoft COCO

- Microsoft **COCO**: **C**ommon **O**bjects in **Co**ntext
- Images (from **Flickr**) with multiple objects in a naturalistic context.
- **82,783 images** (88% training, 6% validation, 6% testing), each with at least five human generated captions each (using Amazon Mechanical Turk).



Chen, Xinlei, et al. "Microsoft COCO captions: Data collection and evaluation server." *arXiv preprint arXiv:1504.00325* (2015).

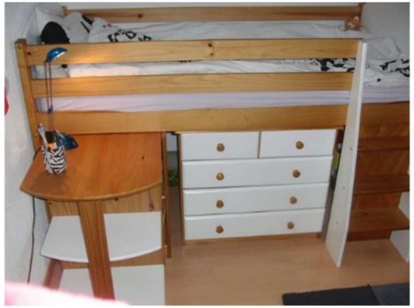
Training Dataset: Microsoft COCO

- Microsoft **COCO**: **C**ommon **O**bjects in **Co**ntext
- Images (from **Flickr**) with multiple objects in a naturalistic context.
- **82,783 images** (88% training, 6% validation, 6% testing), each with at least five human generated captions each (using Amazon Mechanical Turk).

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

A horse carrying a large load of hay and two people sitting on it.



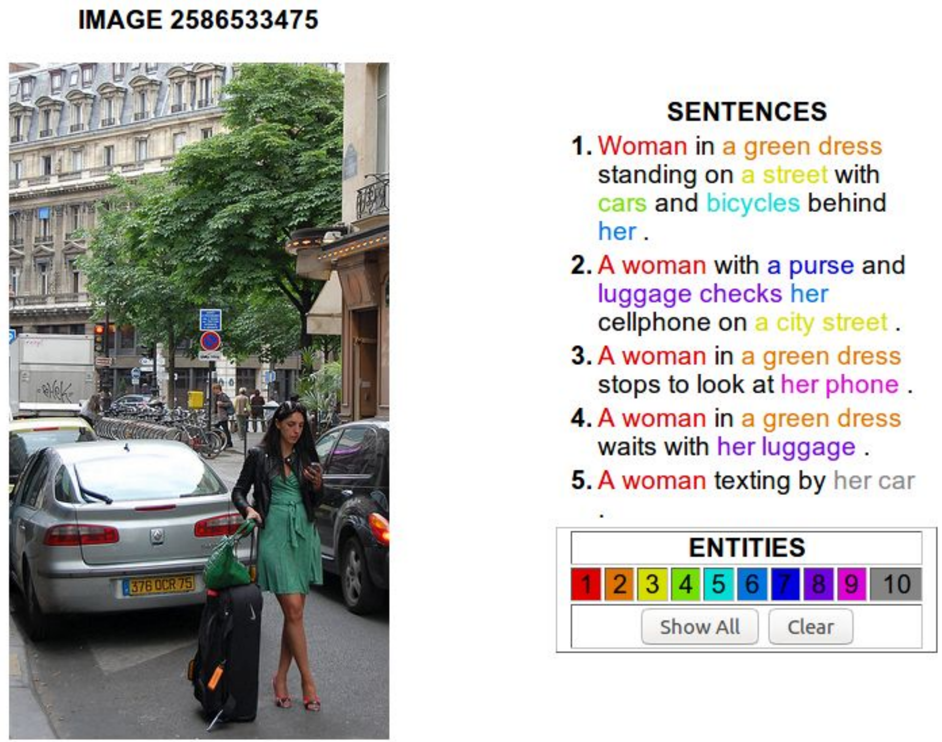
Bunk bed with a narrow shelf sitting underneath it.

Chen, Xinlei, et al. "Microsoft COCO captions: Data collection and evaluation server." *arXiv preprint arXiv:1504.00325* (2015).

Training Dataset: Flickr8k and Flickr30k

- 8,000 and 30,000 images
- More images (from **Flickr**) with multiple objects in a naturalistic context.
- 1,000 testing, 1,000 validation, and the rest training.

Young, Peter, et al. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions." *Transactions of the Association for Computational Linguistics* 2 (2014): 67-78.



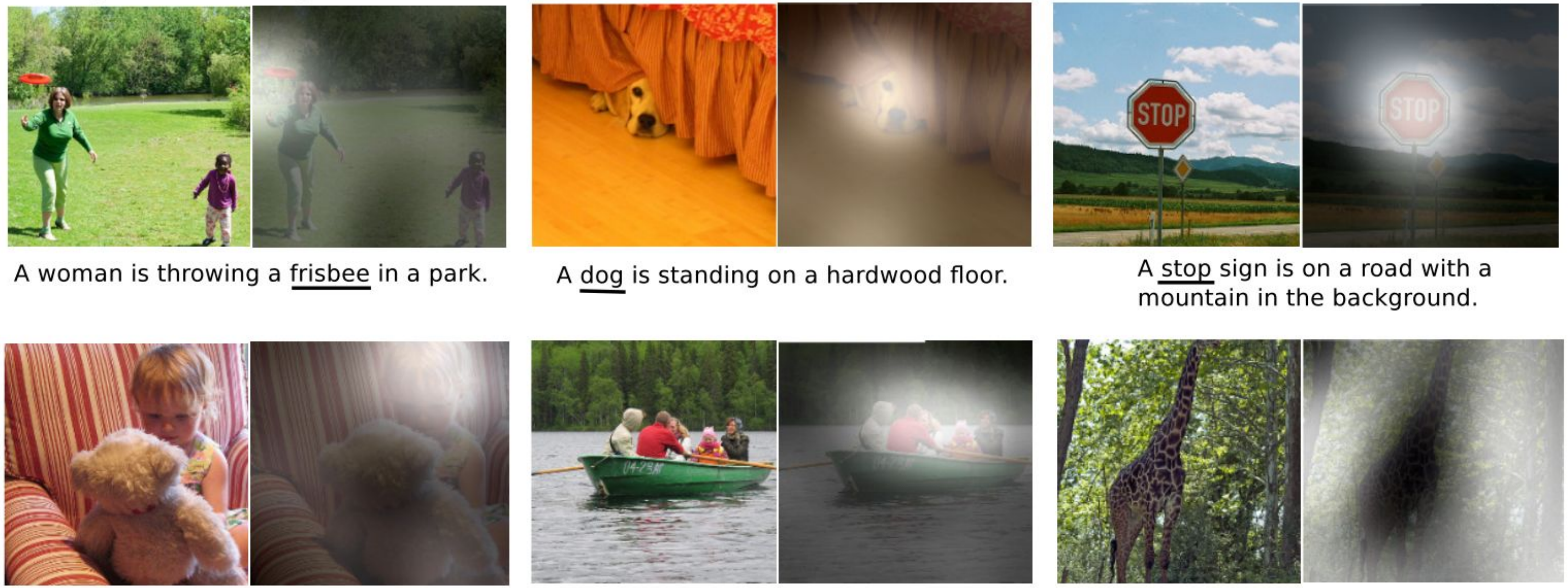
Results: Caption Generation

Table 1. BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ∘ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, a indicates using AlexNet

Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

Results: Caption Generation

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

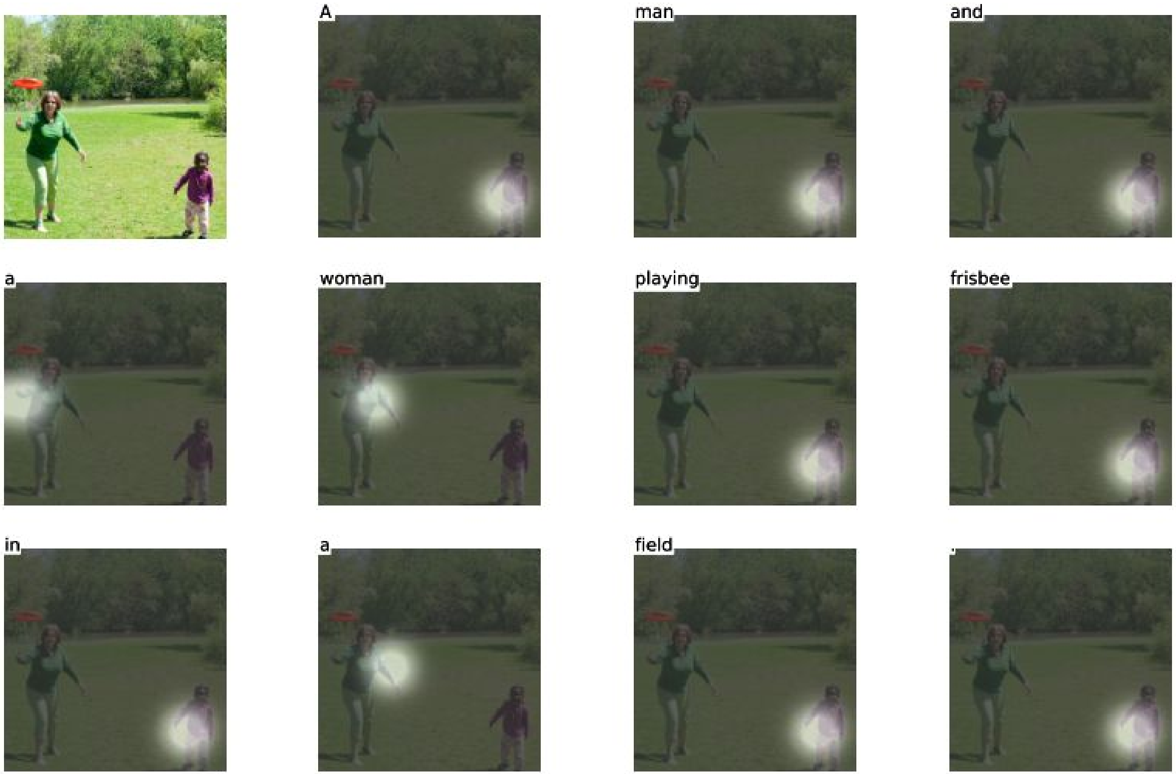
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Results: Attention Mechanism (Soft)

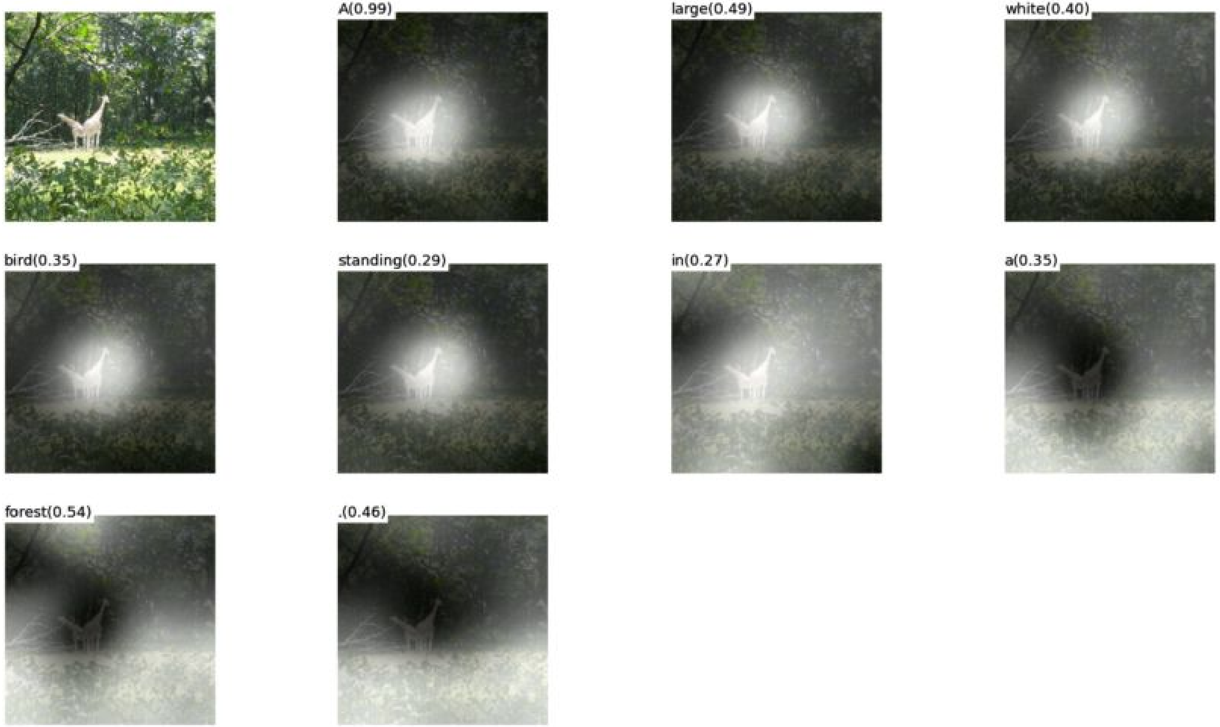
(b) A woman is throwing a frisbee in a park.

Results: Attention Mechanism (Hard)



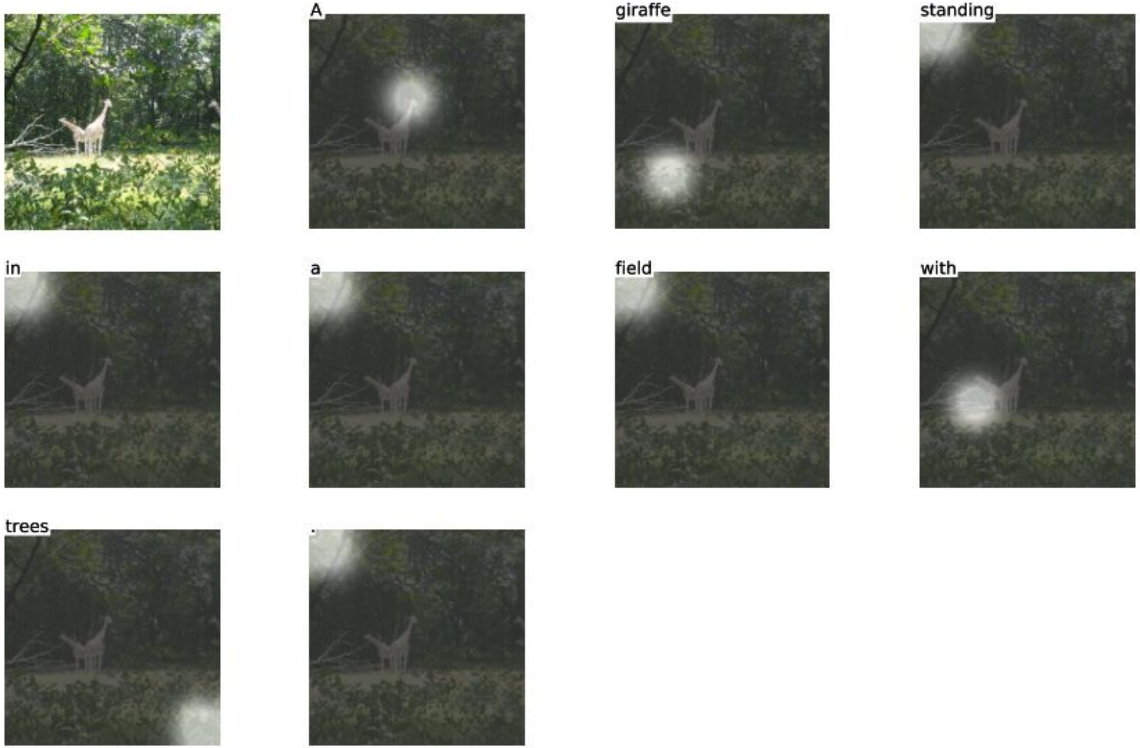
(a) A man and a woman playing frisbee in a field.

Results: Attention Mechanism (Soft)



(b) A large white bird standing in a forest.

Results: Attention Mechanism (Hard)

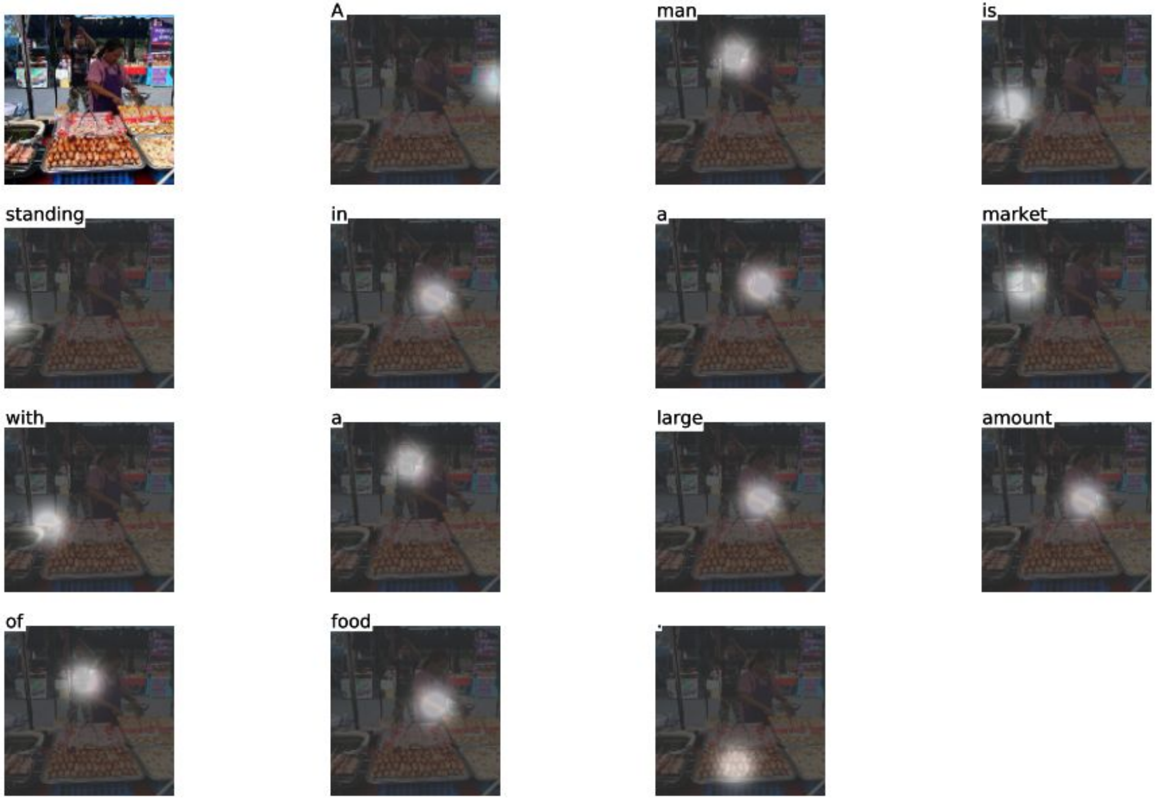


(a) A giraffe standing in the field with trees.

Results: Attention Mechanism (Soft)

(b) A woman is sitting at a table with a large pizza.

Results: Attention Mechanism (Hard)



(a) A man is standing in a market with a large amount of food.

Results: Improper Captions

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

A man is talking on his cell phone while another man watches.

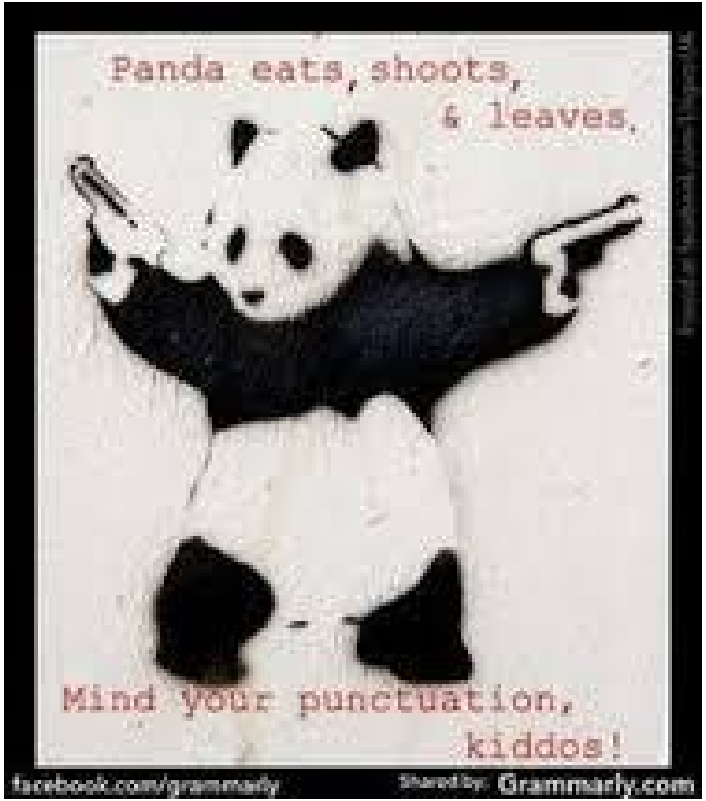
GENERATING IMAGES FROM CAPTIONS WITH ATTENTION

Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba & Ruslan Salakhutdinov
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
{emansim,eparisotto,rsalakhu}@cs.toronto.edu, jimmy@psi.utoronto.ca

Caption → Image

The cat sat on the mat →

Some subtleties



How should we model this problem?

How should we read in the caption?

What if we ignore the sequential structure?

I.e. just feed it into a multilayer perceptron?

Bad idea since hard to process captions of different length, and better to hard code in the sequential nature of text

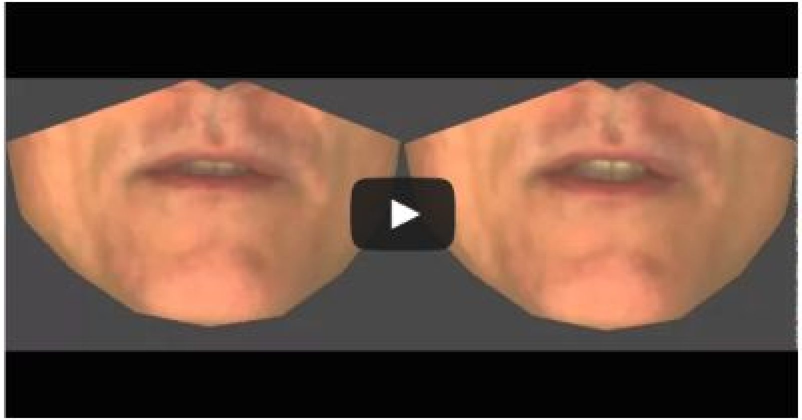
Similar problem?

In “A Decision Tree Framework for Spatiotemporal Sequence Prediction”

Input speech: “P R E D I C T I O N”

		Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
(a)	**x**	Token	-	p	p	r	ih	ih	d	d	ih	ih	ih	ih	k	k	sh	sh	sh	sh	uh	uh	n	-

-------------->



“A Decision Tree Framework for Spatiotemporal Sequence Prediction”

Converting phonetics → lip motion

This problem has a lot of temporal locality in the input -- output relation.

This motivates the SLIDING WINDOW approach

This resembles our problem if we imagine sequentially generating the image

But we don’t have the same temporal locality

For example:

The cat sitting on the bright red mat **was very fat.**

Therefore a sliding window model is not the most obvious choice here (although probably it could be made to work…)

The authors went with LSTM to read in the input

But the idea of sequentially generating the output image is not a bad one.

It can hard code the idea that natural images are built up compositionally:

BACKGROUND + CAT + MAT =

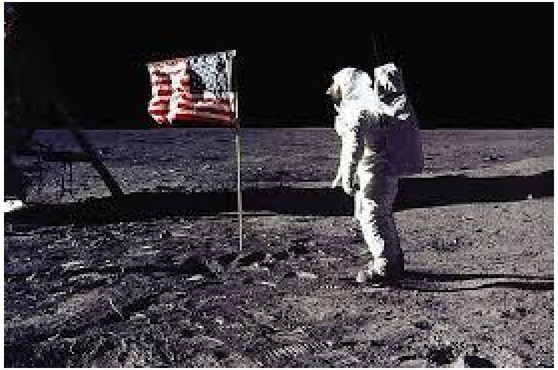
Maybe building output compositionally can help to do this:

Train on:

Man on wire: Moon in sky:

Generalise to:

Man on moon?



We may want to draw the output image sequentially

But not in the order it is fed in, and maybe paying atte
time, and ignoring other words

E.g.

Step 1:

The student, whilst thinking of contrived examples, w

Step 2:

The student, whilst **thinking** of contrived examples, w

If we’re paying attention to regions of the input caption

Why not generate image sequentially

and pay attention to regions of the canvas too?

The model

Image at t-1

Compute image residual + region of attention

Image at t

v(The) v(cat) v(sat) v(on) v(the) v(mat)

Attention

v(The) v(cat) v(sat) v(on) v(the) v(mat)

Bi-directional LSTM

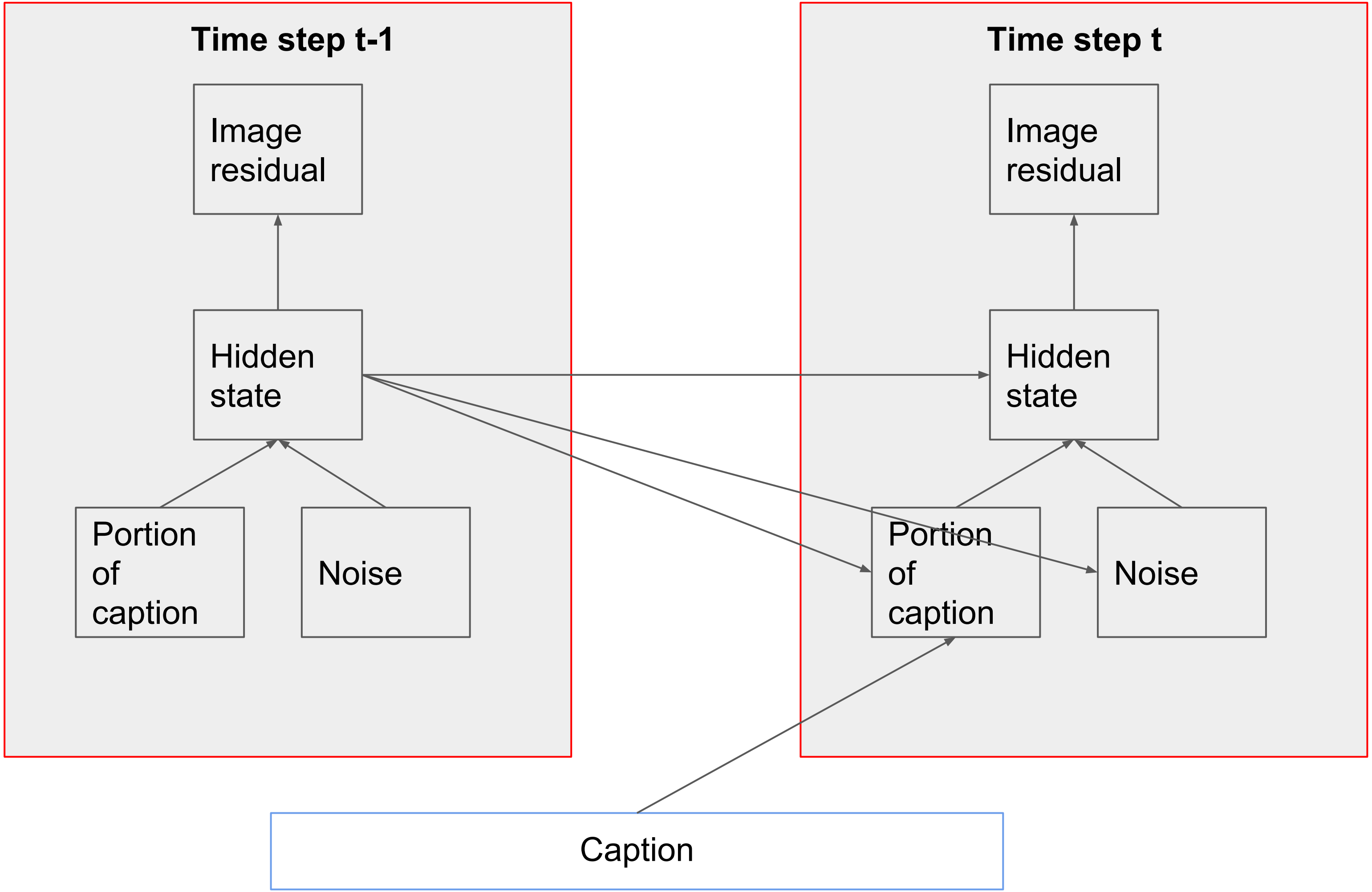
The cat sat on the mat

Our model has several components

But they are all neural network modules

→ differentiable end-to-end

→ train by SGD


Time step t-1
Image residual
Hidden state
Portion of caption
Noise
Time step t
Image residual
Hidden state
Portion of caption
Noise
Caption

Comments on the model

It has a lot of redundancy:

E.g.

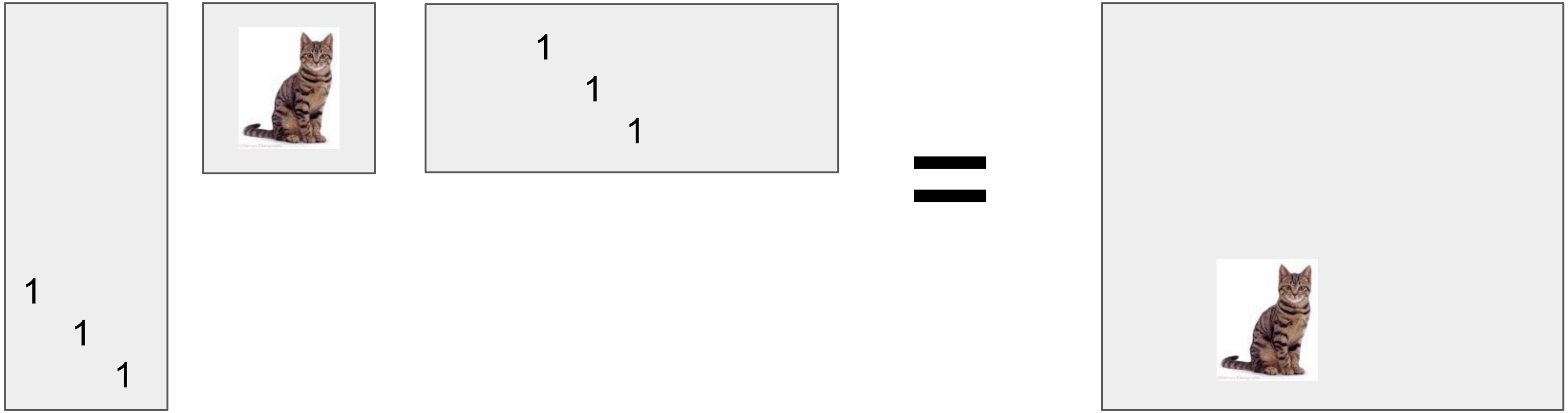
1. The previous hidden state feeds into a lot of different components… Why?
   …..It probably improved test performance

2. Both the bidirectional LSTM input and the attention mechanism on the input enable to link words which are far apart. Are both really necessary?

Essentially the model is very flexible.

The drawing mechanism

The drawing mechanism also uses attention…



Now must discuss learning

Feed in caption + ground truth image

Maximise lower bound on log likelihood, conditioned on caption

$$\mathcal{L} = \sum_{Z} Q(Z \mid \mathbf{x}, \mathbf{y}) \log P(\mathbf{x} \mid \mathbf{y}, Z) - \mathrm{D}_{\mathrm{KL}} \left(Q(Z \mid \mathbf{x}, \mathbf{y}) \,\|\, P(Z \mid \mathbf{y}) \right) \leq \log P(\mathbf{x} \mid \mathbf{y}). \tag{9}$$

Amounts to a reconstruction loss on the attended to image region, plus a regularisation to ensure a noisy latent code (prevents memorisation of the training set)

The loss can be reduced by altering the attended to region, altering the drawing mechanism, altering the attention applied to caption, altering the caption LSTM...

Results

A stop sign is flying in blue skies.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.

Figure 1: Examples of generated images based on captions that describe novel scene compositions that are highly unlikely to occur in real life. The captions describe a common object doing unusual things or set in a strange location.

We can compose concepts not seen in training set

A yellow school bus parked in a parking lot.

A red school bus parked in a parking lot.

A green school bus parked in a parking lot.

A blue school bus parked in a parking lot.

The decadent chocolate desert is on the table.

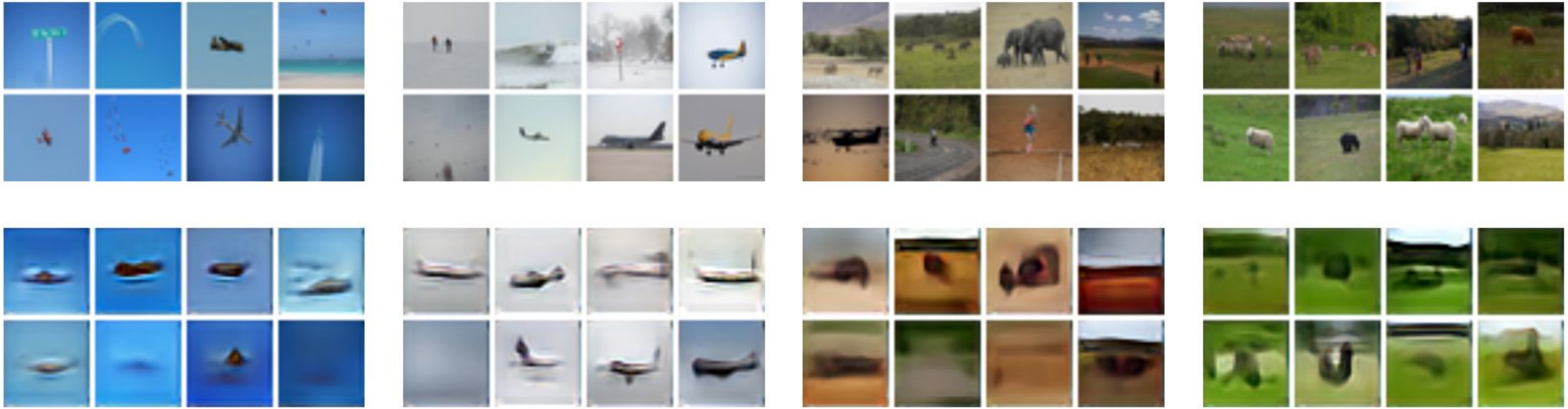
A bowl of bananas is on the table.

A vintage photo of a cat.

A vintage photo of a dog.

Figure 3: **Top**: Examples of changing the color while keeping the caption fixed. **Bottom**: Examples of changing the object while keeping the caption fixed. The shown images are the probabilities $\boldsymbol{\sigma}(c_T)$. Best viewed in colour.

We can change the color of a bus…………..But can’t turn a dog into a cat



A very large commercial plane flying in blue skies.

A very large commercial plane flying in rainy skies.

A herd of elephants walking across a dry grass field.

A herd of elephants walking across a green grass field.

Figure 4: **Bottom**: Examples of changing the background while keeping the caption fixed. **Top**: The respective nearest training images based on pixel-wise L2 distance. The nearest images from the training set also indicate that the model was not simply copying the patterns it observed during the learning phase.

Again, we can compose. And not just reproducing training set

Interpretability

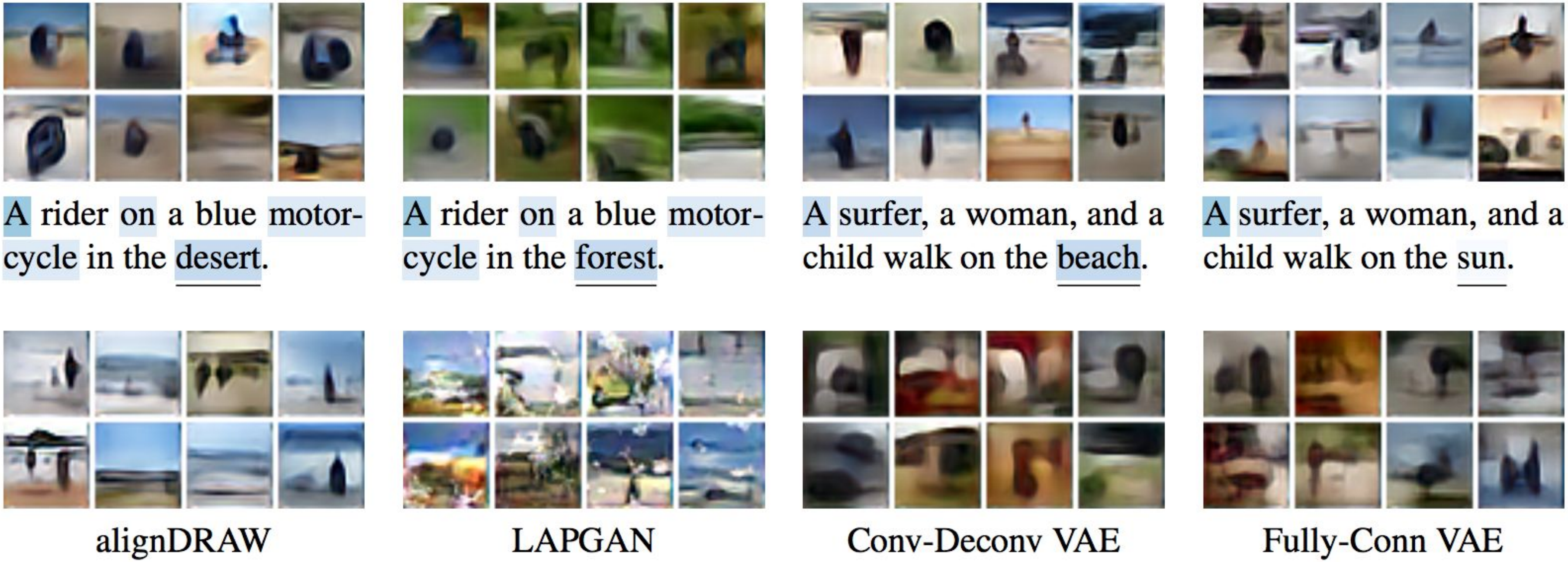
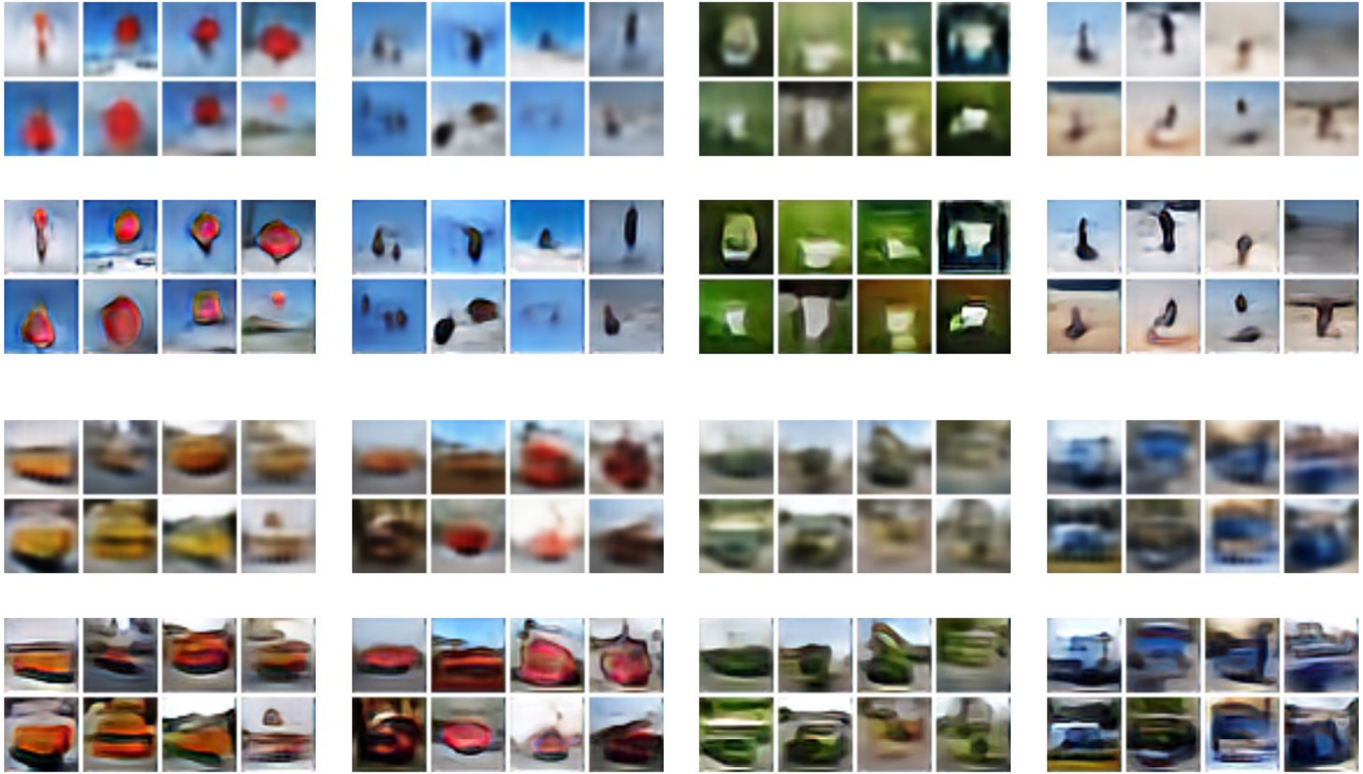


Figure 5: **Top**: Examples of most attended words while changing the background in the caption. **Bottom**: Four different models displaying results from sampling caption *A group of people walk on a beach with surf boards*.

APPENDIX C: EFFECT OF SHARPENING IMAGES.

Some examples of generated images before (top row) and after (bottom row) sharpening images using an adversarial network trained on residuals of a Laplacian pyramid conditioned on the skipthought vectors of the captions.



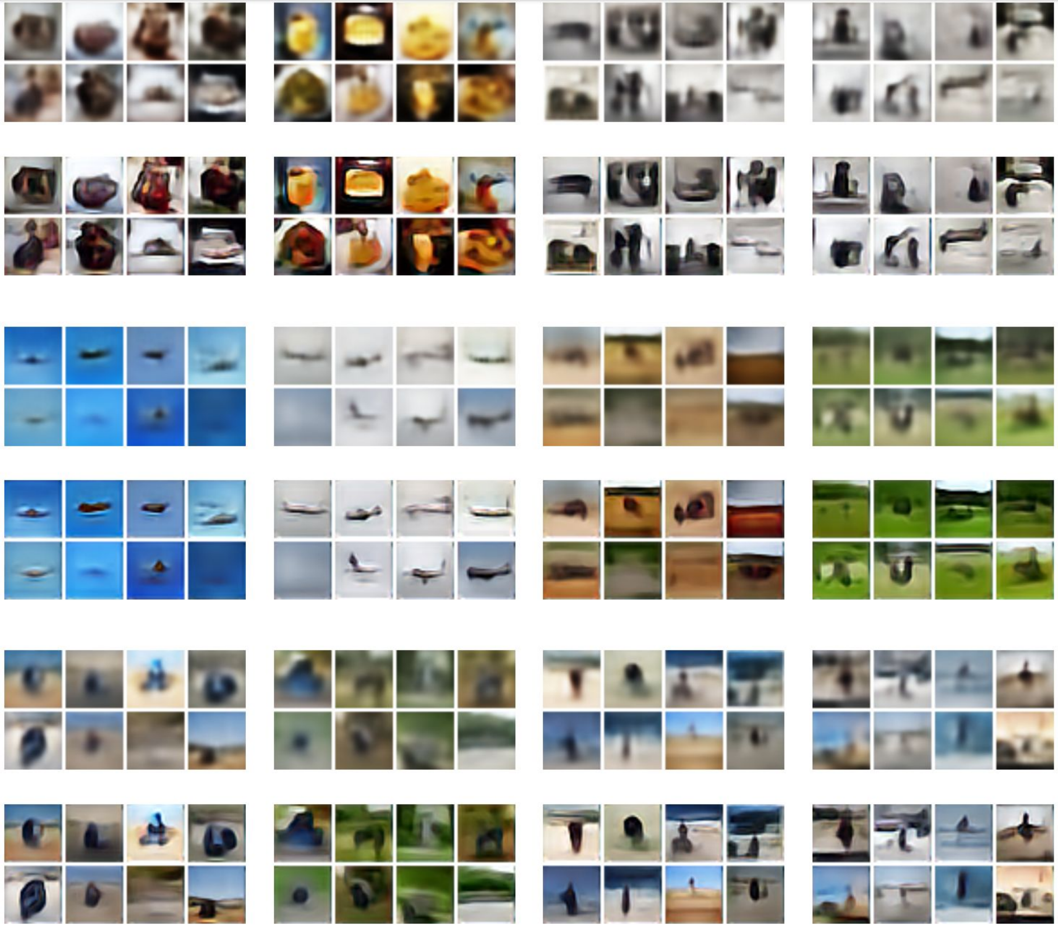


Figure 7: Effect of sharpening images.

Problems

1. No universally agreed upon way to evaluate generative models
2. No link between words attended to and patch drawn at a particular time step
3. Sharpening the images using a GAN at the end is dodgy

There’s still a long way to go:

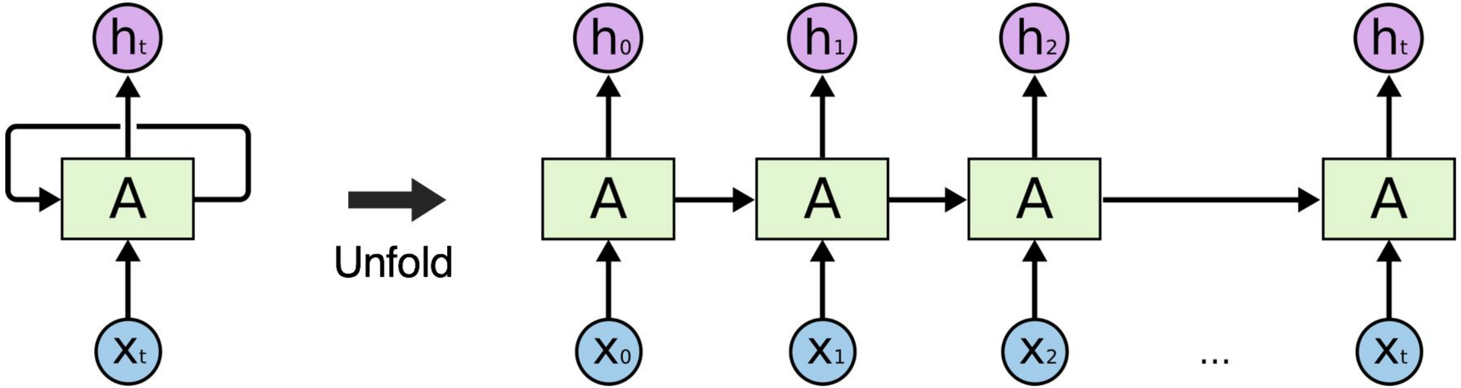
State of the art:

A red school bus parked in a parking lot.

Thank you

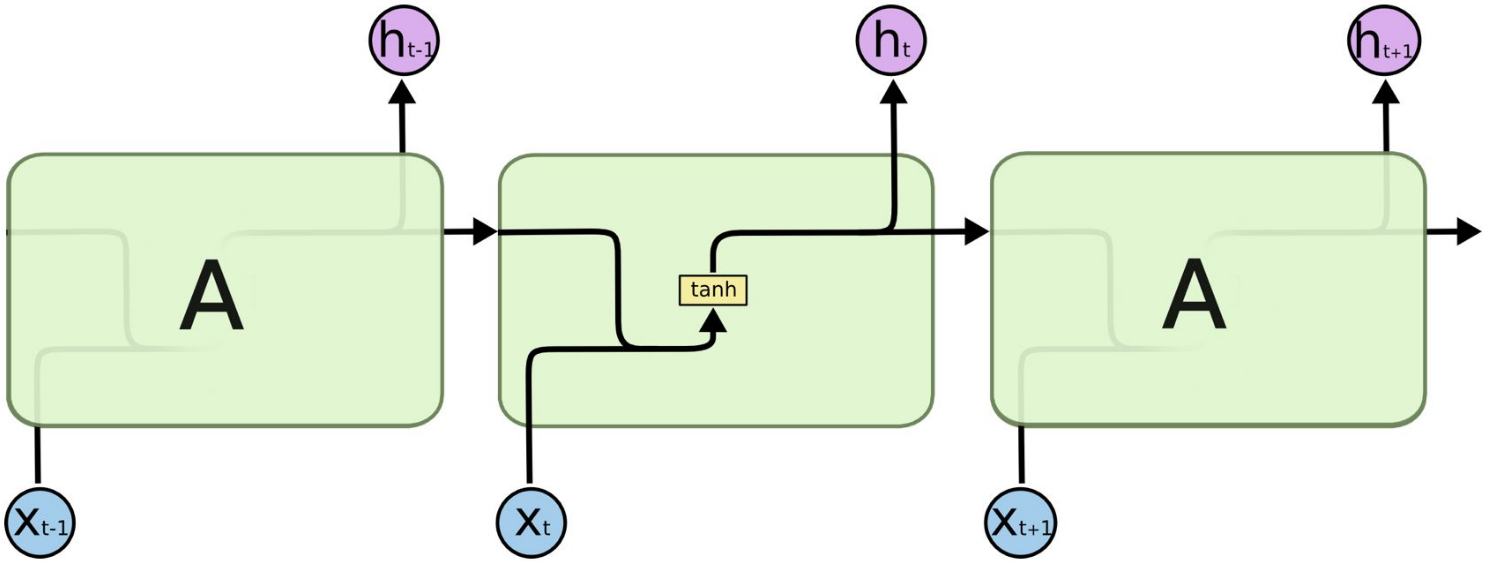
END OF PRESENTATION

Standard RNN



(Olah, 2015)

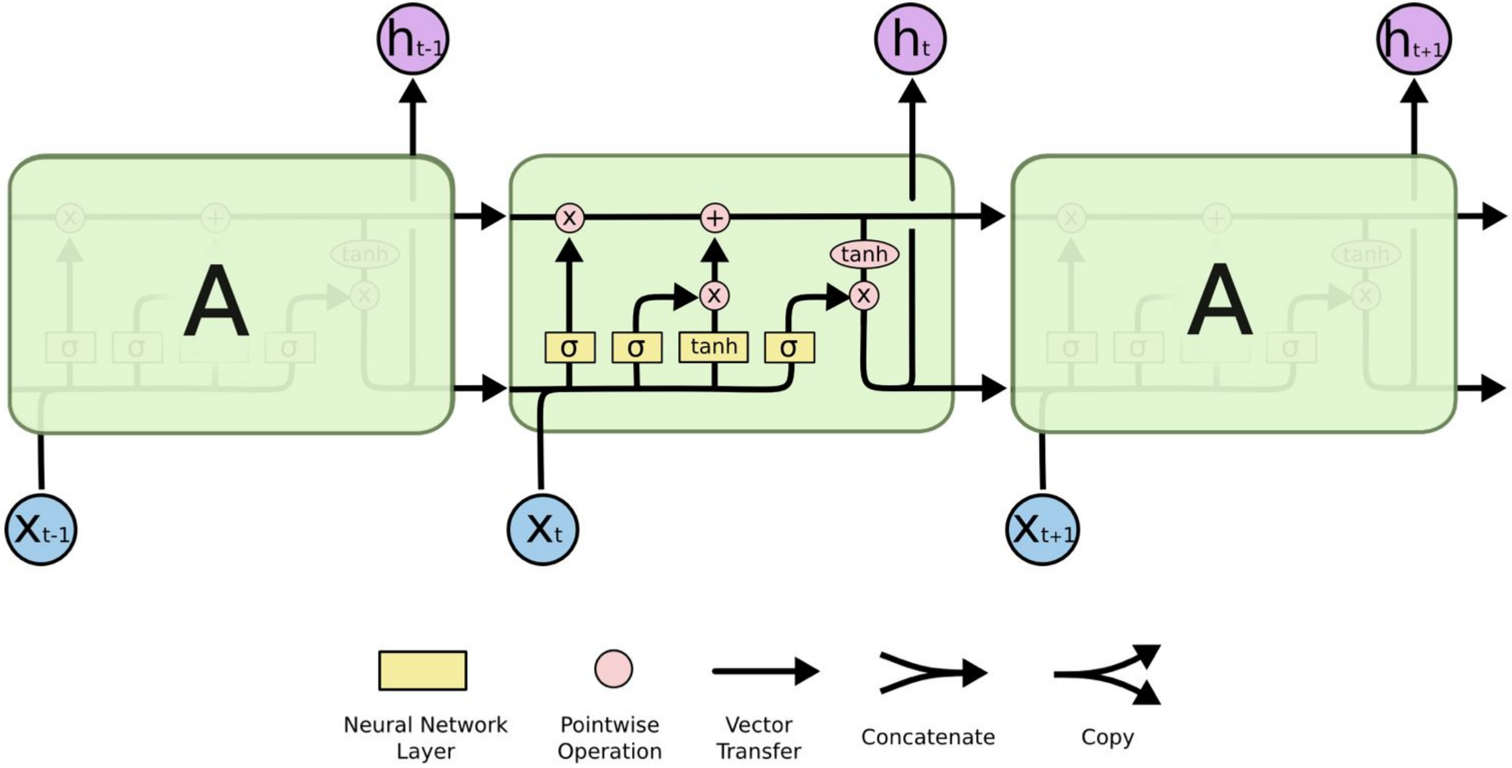
Standard RNN



Vanishing and exploding gradient problems (Bengio et al, 1994; Pascanu et al, 2013)

(Olah, 2015)

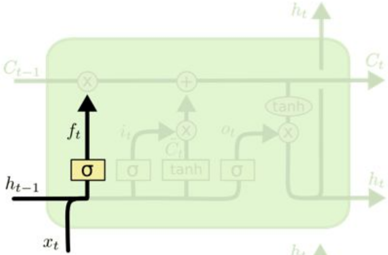
LSTM



(Olah, 2015)

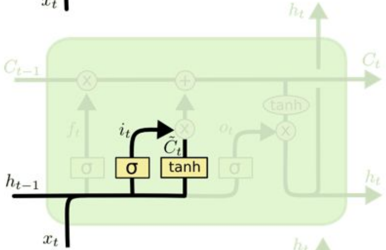
LSTM Cell

Forget gate



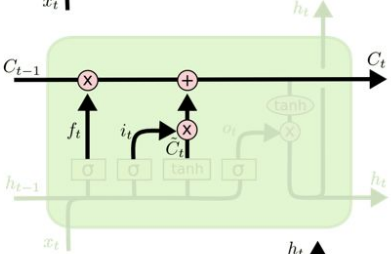
$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

Input layer



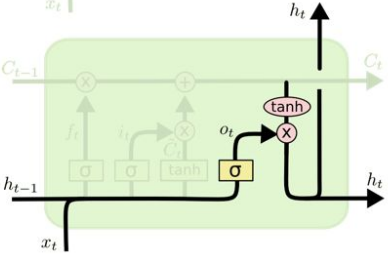
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

Cell state update



$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_h(W_c x_t + U_c h_{t-1} + b_c)$$

Output
(filtered cell state)



$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$h_t = o_t \circ \sigma_h(c_t)$$

$c_0 = 0$ and $h_0 = 0$

Activations

- σ_g: sigmoid
- σ_h: tanh

$\boldsymbol{f, i, c, o, h}$ = d x 1

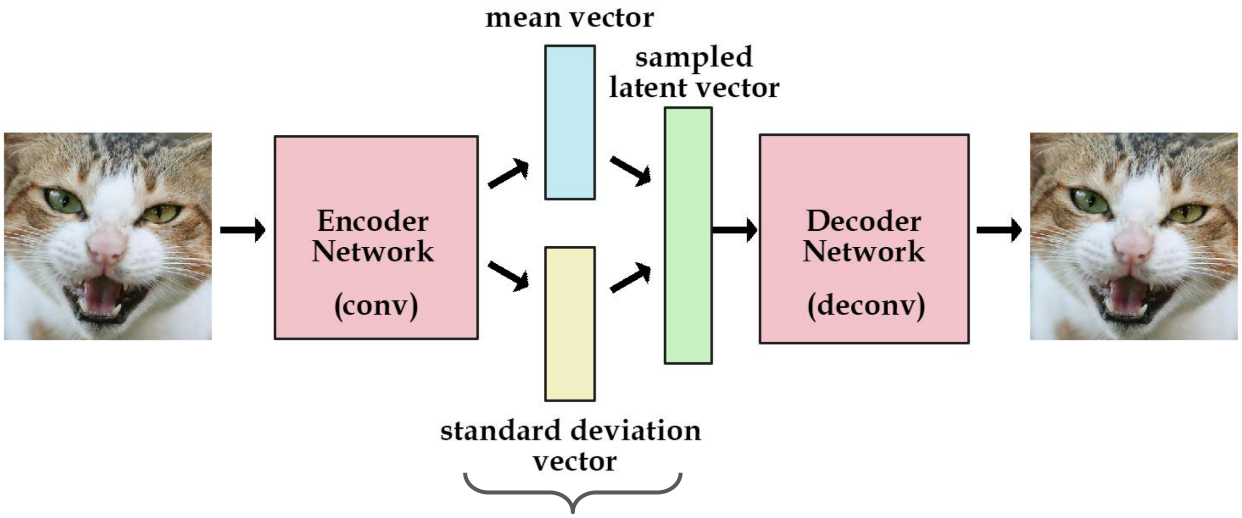
$\boldsymbol{x}$ = n x 1

$\boldsymbol{W}$ = d x n

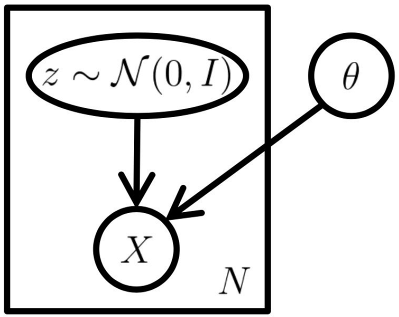
$\boldsymbol{U}$ = d x d

(Olah, 2015)

Variational Autoencoders (VAE)

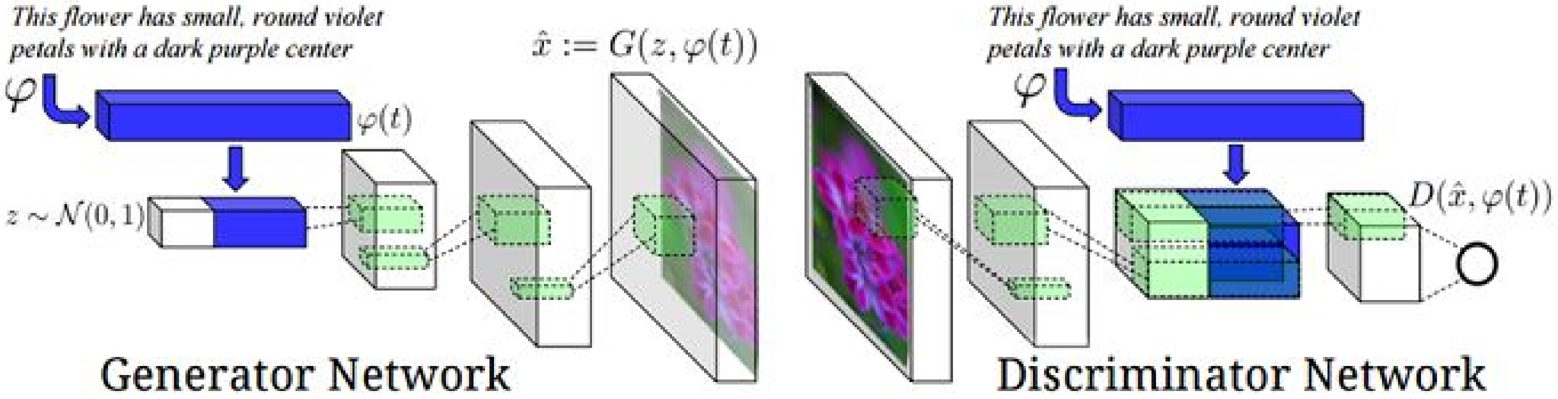


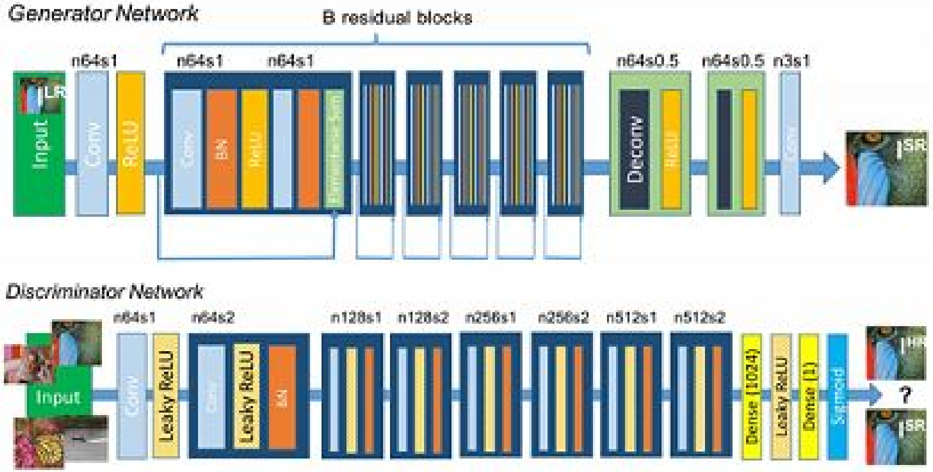
Reparameterization
$Q(z|x)$



Minimizing 2 losses: generative and latent loss

Generative adversarial networks





Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Metrics: BLEU (2002)

- Automatic, quick, language-invariant, machine translation evaluation measure - **bi**lingual **e**valuation **u**nderstudy (**BLEU**)
- Compare machine translation to professional human translation.
- Compare *n*-gram matches without regard to position, the more matches, the better the machine translation.
- Clip the *n*-gram precision and modify so short sentences aren't favored.

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

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

$$BLEU_N(C,S) = b(C,S) \exp\left(\sum_{n=1}^{N} w_n \log CP_n(C,S)\right), \quad (3)$$

Chen, Xinlei, et al. "Microsoft COCO captions: Data collection and evaluation server." *arXiv preprint arXiv:1504.00325* (2015).

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002.

Metrics: METEOR (2014)

$$Pen = \gamma \left(\frac{ch}{m}\right)^{\theta}$$

$$F_{mean} = \frac{P_m R_m}{\alpha P_m + (1-\alpha) R_m}$$

$$P_m = \frac{|m|}{\sum_k h_k(c_i)}$$

$$R_m = \frac{|m|}{\sum_k h_k(s_{ij})}$$

$$METEOR = (1 - Pen) F_{mean}$$

Lavie, Michael Denkowski Alon. "Meteor universal: Language specific translation evaluation for any target language." *ACL 2014* (2014): 376.