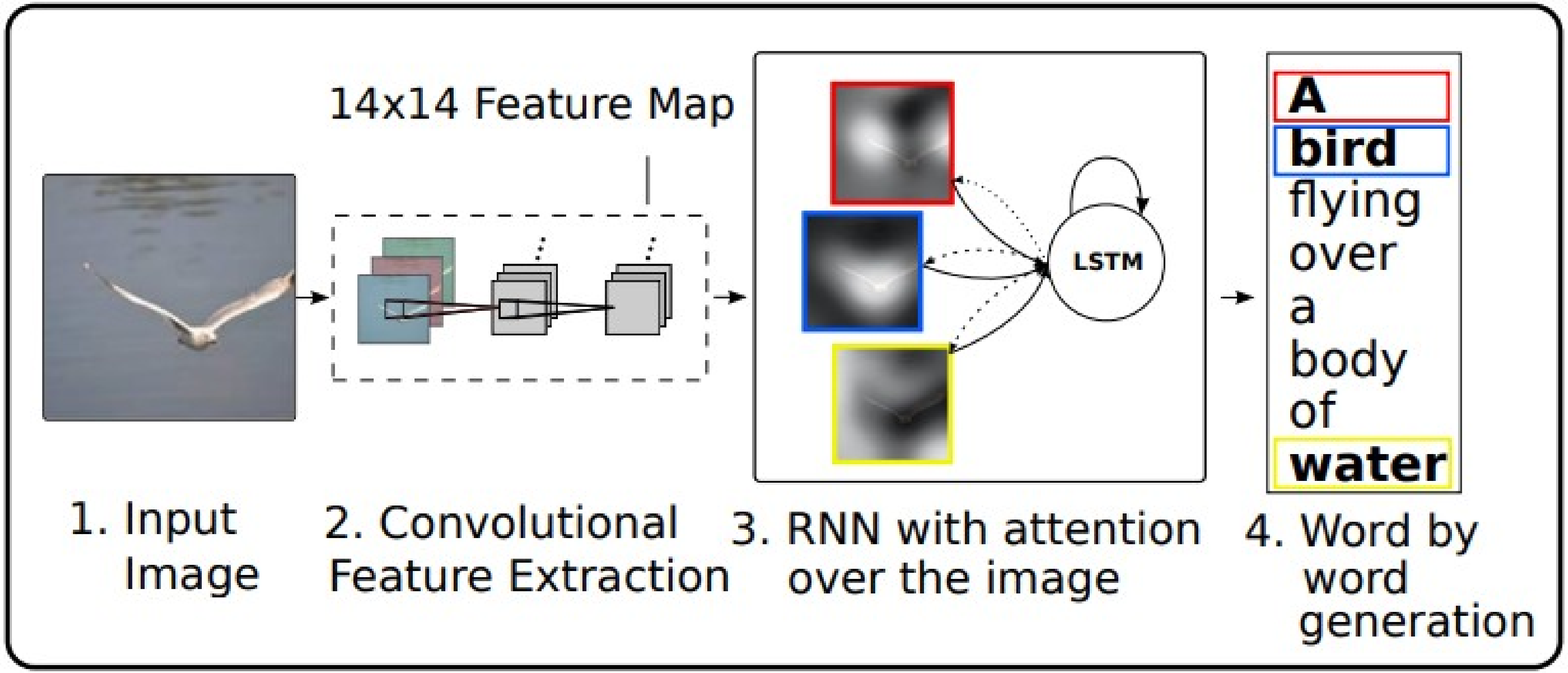
Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio

Overview

- Image captioning using high level VGG19 features
- Soft and Hard Attention
- Focus will be on the decoder

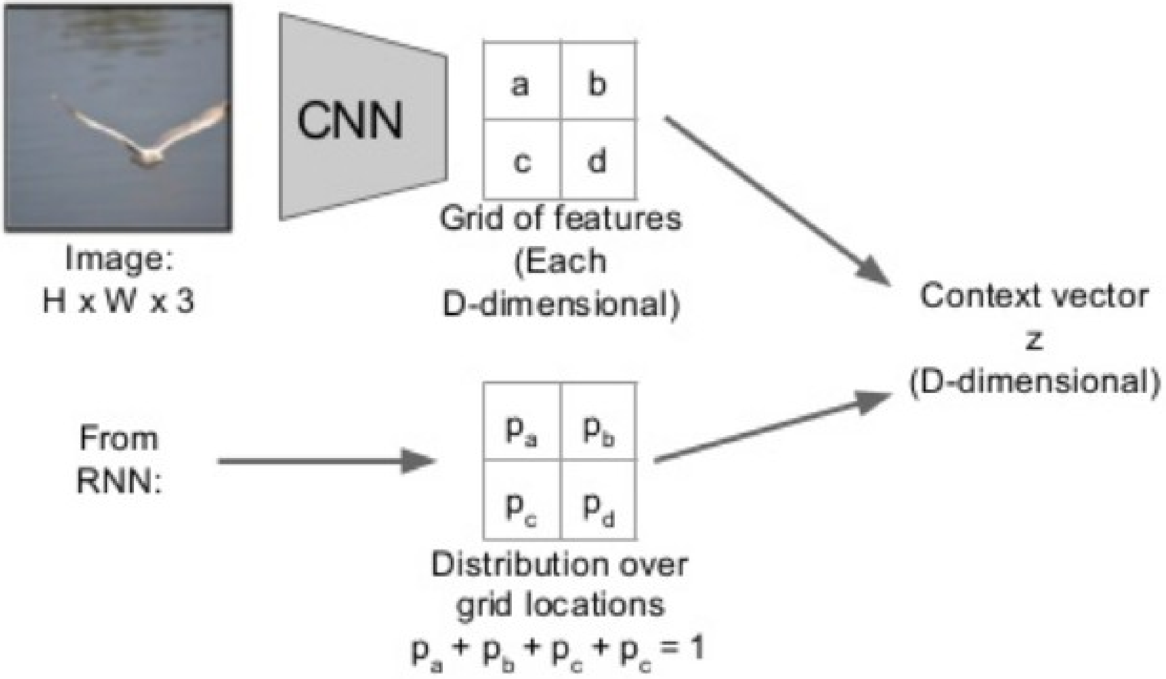
Architecture



Soft Attention

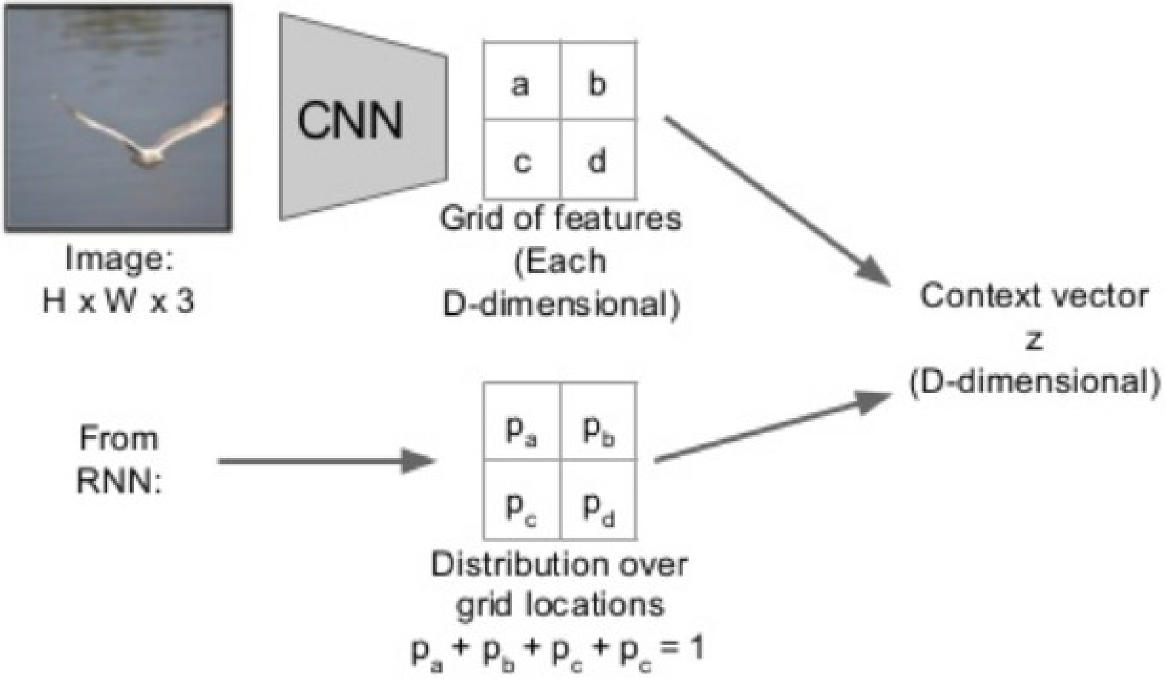
- Deterministic
- Based on Bahdanau’s et al., 2014 attention mechanism
- Trained using backpropagation

Implementing Soft Attention



Implementing Soft Attention

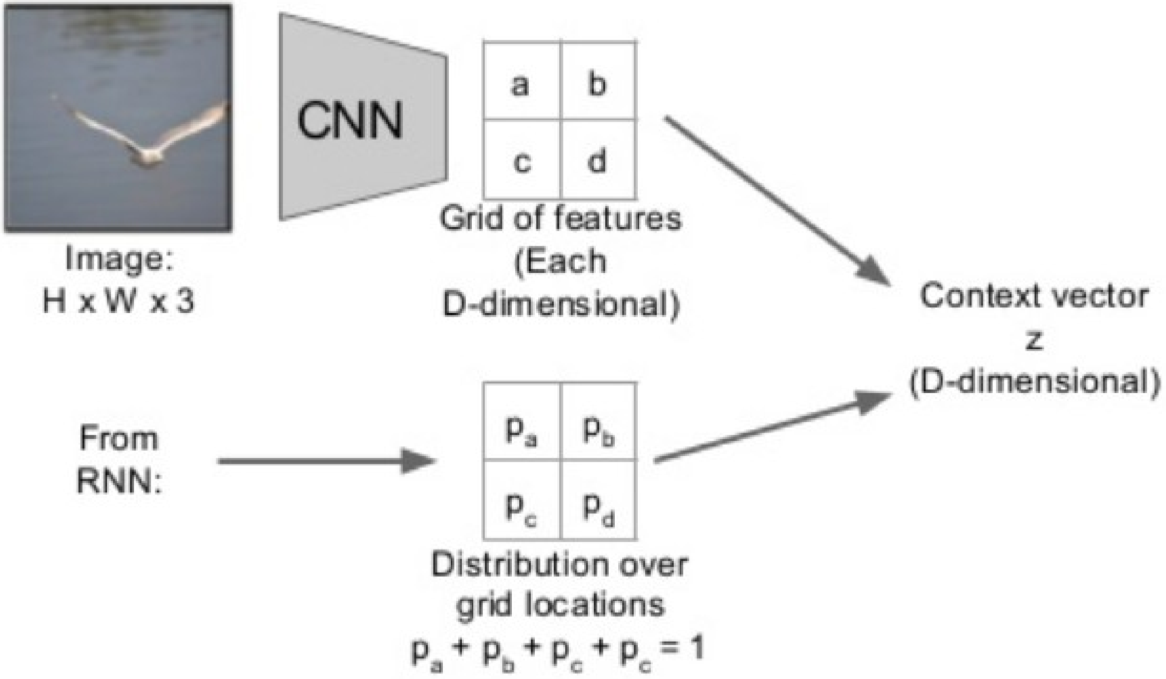
$$e_{ti} = f_{att}(G_i,\ h_{t-1})$$



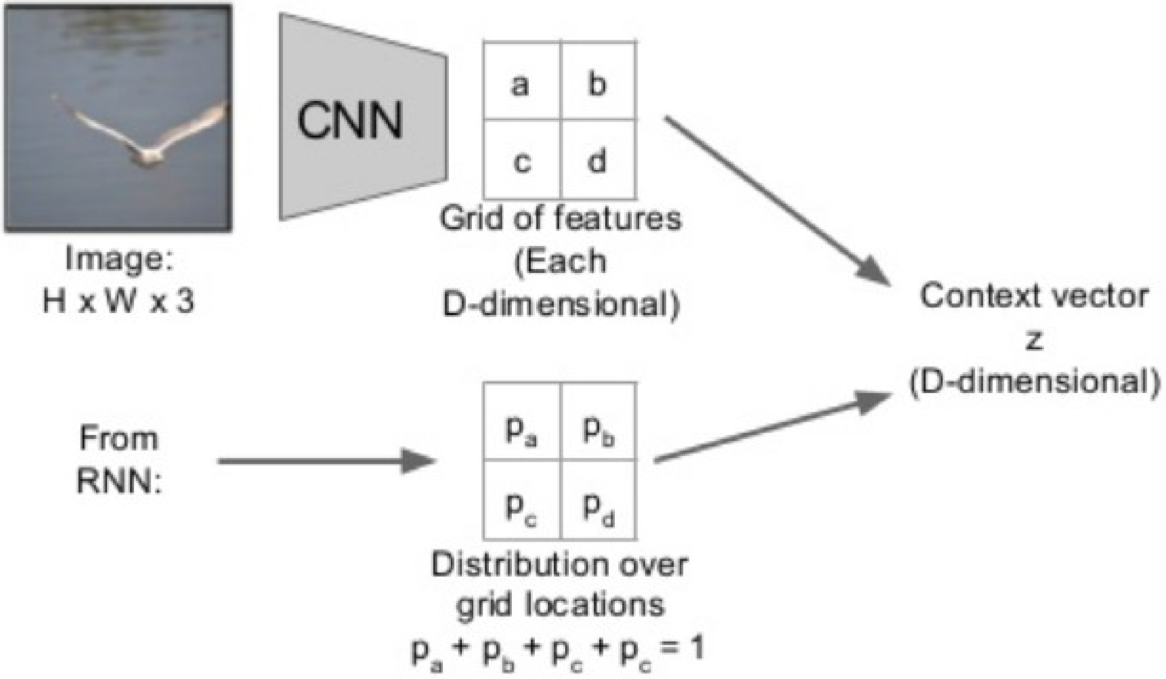
Implementing Soft Attention

$$e_{ti} = f_{att}(G_i,\ h_{t-1})$$

$$p_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})}$$



Implementing Soft Attention

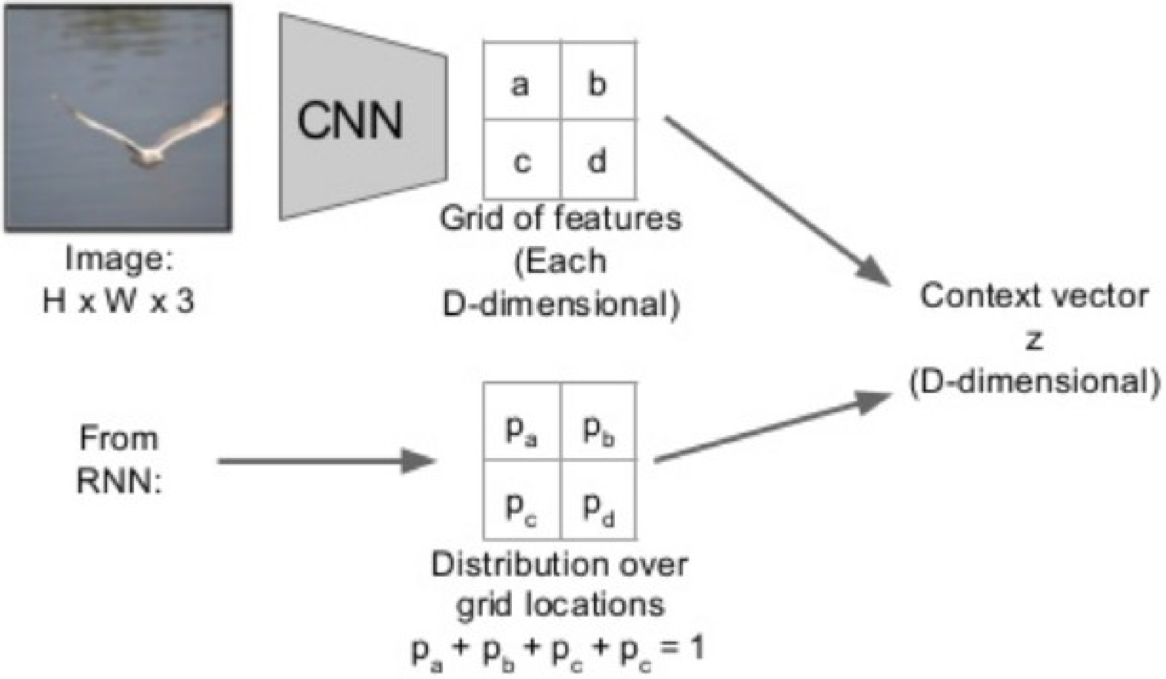


$$e_{ti} = f_{att}(G_i,\ h_{t-1})$$

$$p_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})}$$

$$z_t = \beta_t \sum_{j=1}^{L} p_{tj} G_j$$

Implementing Soft Attention



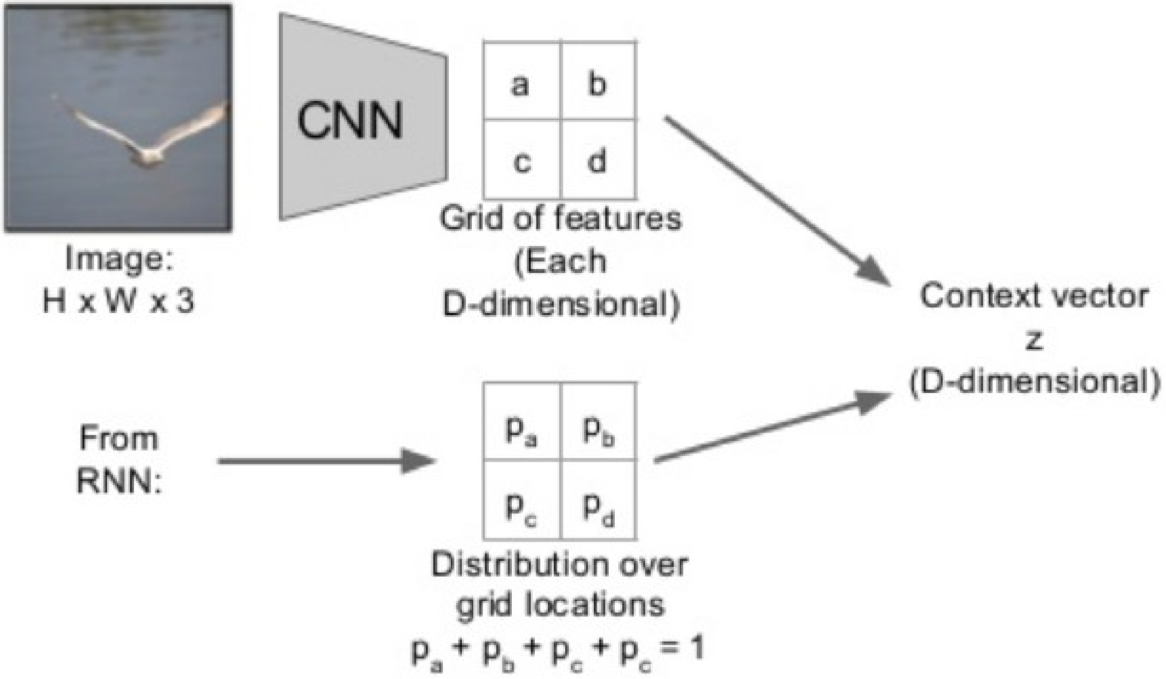
$$e_{ti} = f_{att}(G_i,\ h_{t-1})$$

$$p_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})}$$

$$z_t = \beta_t \sum_{j=1}^{L} p_{tj} G_j$$

$$\beta_t = \sigma(f_\beta(h_{t-1}))$$

Implementing Soft Attention



$$e_{ti} = f_{att}(G_i,\ h_{t-1})$$

$$p_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})}$$

$$z_t = \beta_t \sum_{j=1}^{L} p_{tj} G_j$$

$$\beta_t = \sigma(f_\beta(h_{t-1}))$$

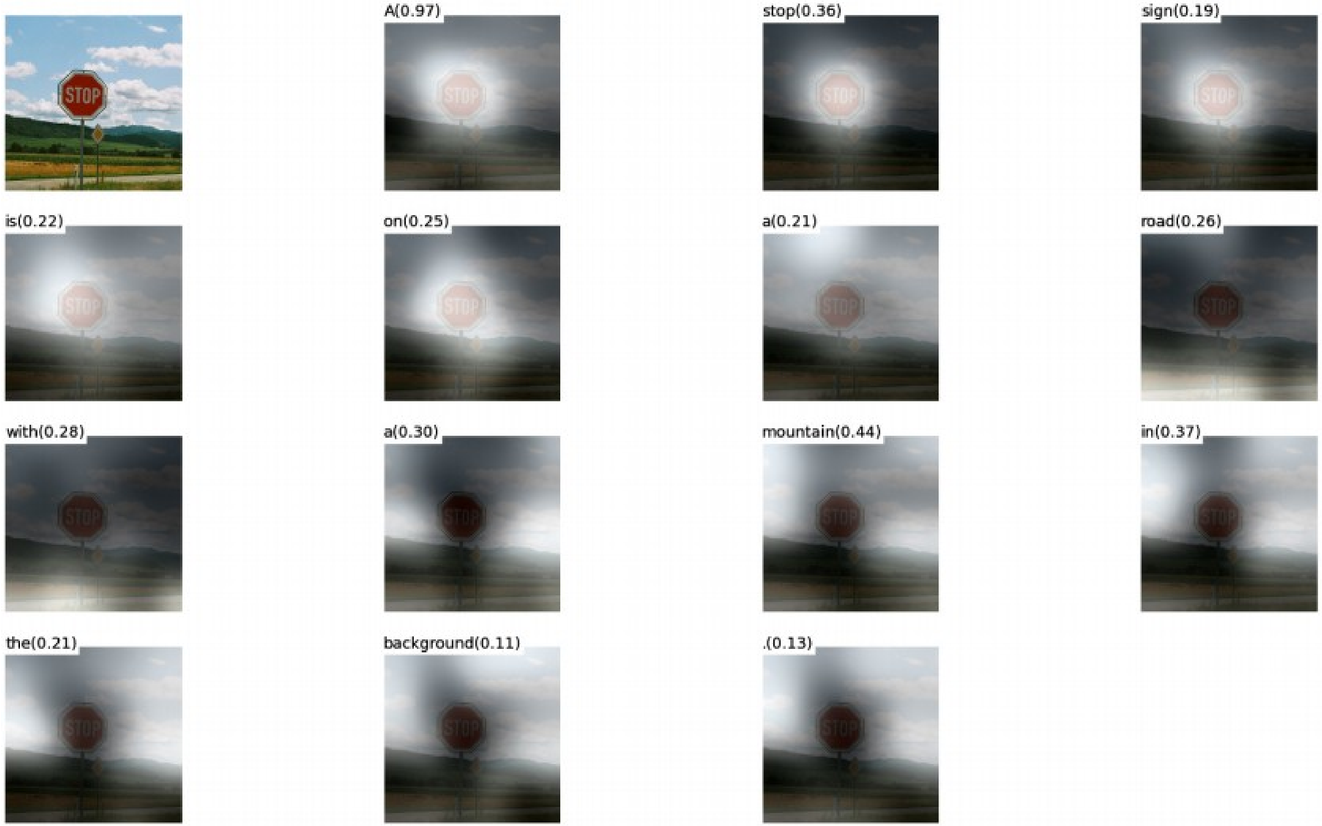
$(z_t,\ E_{y_{t-1}},\ h_{t-1})$ used as input

Doubly Stochastic Attention

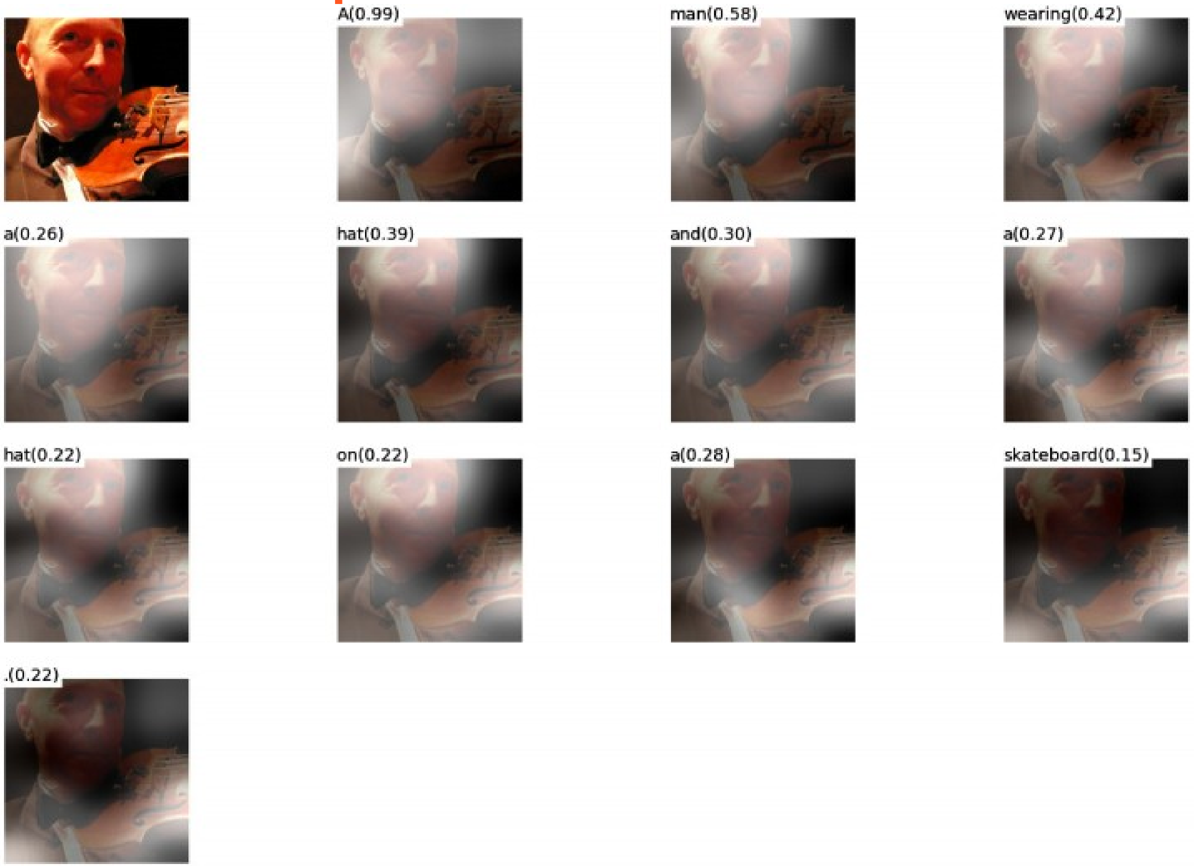
- To encourage the model to look at various parts of the image

$$L_d = -log(P(y|x)) + \lambda \sum_{i=1}^{L}(1 - \sum_{t=1}^{C} p_{ti})^2$$

Positive Example



Negative Example



Hard Attention

- Stochastic
- Assign a multinoulli distribution to the attention weights **p** and view the weighted input **z** as a random variable
- Gradient estimated using Monte Carlo
- Trained using the REINFORCE learning rule

Implementing Hard Attention

Implementing Hard Attention

- We would like to maximize

$$\log p(y|G)$$

Implementing Hard Attention

- We would like to maximize

$$\log p(y|G)$$

$$= \log \sum_{s} p(s|G)p(y|s, G) \qquad \text{where } p(s_{t,i} = 1|s_{j<t}, G) = p_{ti}$$

Implementing Hard Attention

- We would like to maximize

$$\log p(y|G)$$

$$= \log \sum_s p(s|G) p(y|s, G) \qquad \text{where } p(s_{t,i} = 1 | s_{j<t}, G) = p_{ti}$$

$$\geq \sum_s p(s|G) \log p(y|s, G) = L_s$$

Implementing Hard Attention

- We would like to maximize

$$\log p(y|G)$$

$$= \log \sum_s p(s|G)p(y|s,G) \qquad \text{where } p(s_{t,i} = 1|s_{j<t}, G) = p_{ti}$$

$$\geq \sum_s p(s|G) \log p(y|s,G) = L_s$$

$$\frac{\partial L_s}{\partial W} = \sum_s \frac{\partial p(s|G)}{\partial W} \log p(y|s,G) + p(s|G) \frac{\partial \log p(y|s,G)}{\partial W}$$

Implementing Hard Attention

$$\frac{\partial L_s}{\partial W} = \sum_s \frac{\partial p(s|G)}{\partial W} log\, p(y|s, G) \;+\; p(s|G)\, \frac{\partial log\, p(y|s,G)}{\partial W}$$

Implementing Hard Attention

$$\frac{\partial L_s}{\partial W} = \sum_s \frac{\partial p(s|G)}{\partial W} \log p(y|s, G) + p(s|G) \frac{\partial \log p(y|s,G)}{\partial W}$$

$$\frac{\partial \log p(s|G)}{\partial W} = \frac{1}{p(s|G)} \frac{\partial p(s|G)}{\partial W}$$

$$\frac{\partial p(s|G)}{\partial W} = p(s|G) \frac{\partial \log p(s|G)}{\partial W}$$

Implementing Hard Attention

$$\frac{\partial L_s}{\partial W} = \sum_s \frac{\partial p(s|G)}{\partial W} \log p(y|s, G) \ + \ p(s|G) \ \frac{\partial \log p(y|s,G)}{\partial W}$$

$$\frac{\partial \log p(s|G)}{\partial W} = \frac{1}{p(s|G)} \frac{\partial p(s|G)}{\partial W}$$

$$\frac{\partial p(s|G)}{\partial W} = p(s|G) \frac{\partial \log p(s|G)}{\partial W}$$

$$\frac{\partial L_s}{\partial W} = \sum_s p(s|G) \frac{\partial \log p(s|G)}{\partial W} \log p(y|s, G) \ + \ p(s|G) \ \frac{\partial \log p(y|s,G)}{\partial W}$$

Implementing Hard Attention

$$\frac{\partial L_S}{\partial W} = \sum_s \frac{\partial p(s|G)}{\partial W} log\, p(y|s, G) + p(s|G) \frac{\partial log\, p(y|s,G)}{\partial W}$$

$$\frac{\partial log\, p(s|G)}{\partial W} = \frac{1}{p(s|G)} \frac{\partial p(s|G)}{\partial W}$$

$$\frac{\partial p(s|G)}{\partial W} = p(s|G) \frac{\partial log\, p(s|G)}{\partial W}$$

$$\frac{\partial L_S}{\partial W} = \sum_s p(s|G) \frac{\partial log\, p(s|G)}{\partial W} log\, p(y|s, G) + p(s|G) \frac{\partial log\, p(y|s,G)}{\partial W}$$

$$\frac{\partial L_S}{\partial W} = \sum_s p(s|G) [\frac{\partial log\, p(s|G)}{\partial W} log\, p(y|s, G) + \frac{\partial log\, p(y|s,G)}{\partial W}]$$

Implementing Hard Attention

- This means that Monte Carlo Sampling can be performed!

Implementing Hard Attention

- This means that Monte Carlo Sampling can be performed!

$$s_t' \sim Multinoulli(\{p_i\})$$

$$\frac{\partial L_S}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\frac{\partial log\, p(s'^n|G)}{\partial W} log\, p(y|s'^n, G) + \frac{\partial log\, p(y|s'^n, G)}{\partial W}]$$

Implementing Hard Attention

- This means that Monte Carlo Sampling can be performed!

$$s'_t \sim Multinoulli(\{p_i\})$$

$$\frac{\partial L_S}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\frac{\partial log\ p(s'^n|G)}{\partial W} log\ p(y|s'^n, G) + \frac{\partial log\ p(y|s'^n, G)}{\partial W}]$$

- The issue is that the variance in this estimate is too high

Implementing Hard Attention

- This means that Monte Carlo Sampling can be performed!

$$s'_t \sim Multinoulli(\{p_i\})$$

$$\frac{\partial L_S}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\frac{\partial log\, p(s'^n|G)}{\partial W} log\, p(y|s'^n, G) \; + \; \frac{\partial log\, p(y|s'^n, G)}{\partial W}]$$

- The issue is that the variance in this estimate is too high

$$b_k = \; 0.9\, x\, b_{k-1} \; + \; 0.1\, x\, log\, p(y|s_k', G)$$

Where **k** corresponds to the mini-batch number

Implementing Hard Attention

- This means that Monte Carlo Sampling can be performed!

$$s_t' \sim Multinoulli(\{p_i\})$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\frac{\partial log\, p(s'^n|G)}{\partial W} log\, p(y|s'^n, G) \; + \; \frac{\partial log\, p(y|s'^n, G)}{\partial W}]$$

- The issue is that the variance in this estimate is too high

$$b_k = \; 0.9\, x\, b_{k-1} \; + \; 0.1\, x\, log\, p(y|s_k', G)$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\lambda_r (log\, p(y|s'^n, G) - b) \frac{\partial log\, p(s'^n|G)}{\partial W} \; + \; \frac{\partial log\, p(y|s'^n, G)}{\partial W}]$$

Implementing Hard Attention

- The authors further reduce the variance by adding an entropy term to the attention weights

Implementing Hard Attention

- The authors further reduce the variance by adding an entropy term to the attention weights

$$\frac{\partial L_S}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\lambda_r (\log p(y|s^n, G) - b) \frac{\partial \log p(s^n|G)}{\partial W} + \frac{\partial \log p(y|s^n, G)}{\partial W} + \lambda_e \frac{\partial H[s^n]}{\partial W}]$$

Implementing Hard Attention

- The authors further reduce the variance by adding an entropy term to the attention weights

$$\frac{\partial L_S}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}[\lambda_r(\log p(y|s'^n, G) - b)\frac{\partial \log p(s'^n|G)}{\partial W} + \frac{\partial \log p(y|s'^n,G)}{\partial W} + \lambda_e\frac{\partial H[s'^n]}{\partial W}]$$

Where $H[s'^n]$ is simply $\sum_{j=1}^{L} p_j \log(p_j)$

Implementing Hard Attention

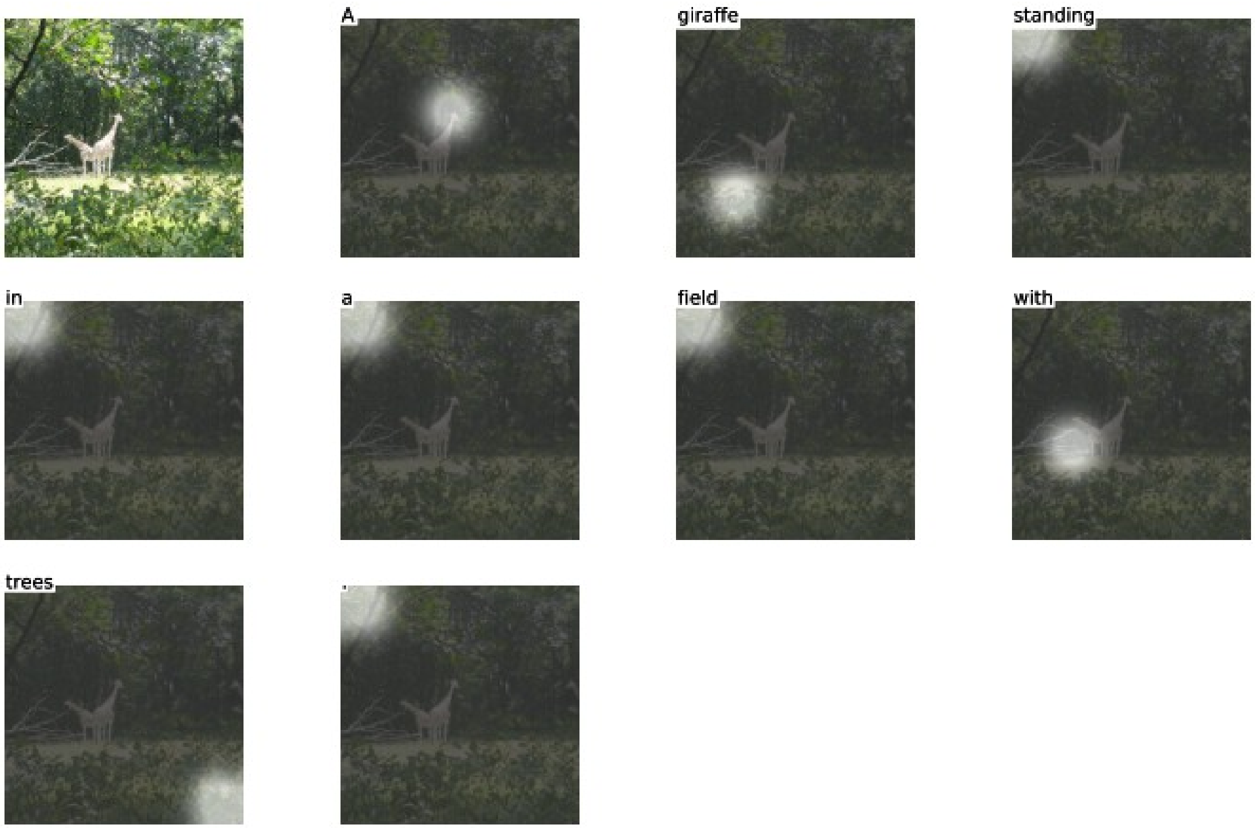
- The authors further reduce the variance by adding an entropy term to the attention weights

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} [\lambda_r (\log p(y|s'^n, G) - b) \frac{\partial \log p(s'^n|G)}{\partial W} + \frac{\partial \log p(y|s'^n, G)}{\partial W} + \lambda_e \frac{\partial H[s'^n]}{\partial W}]$$

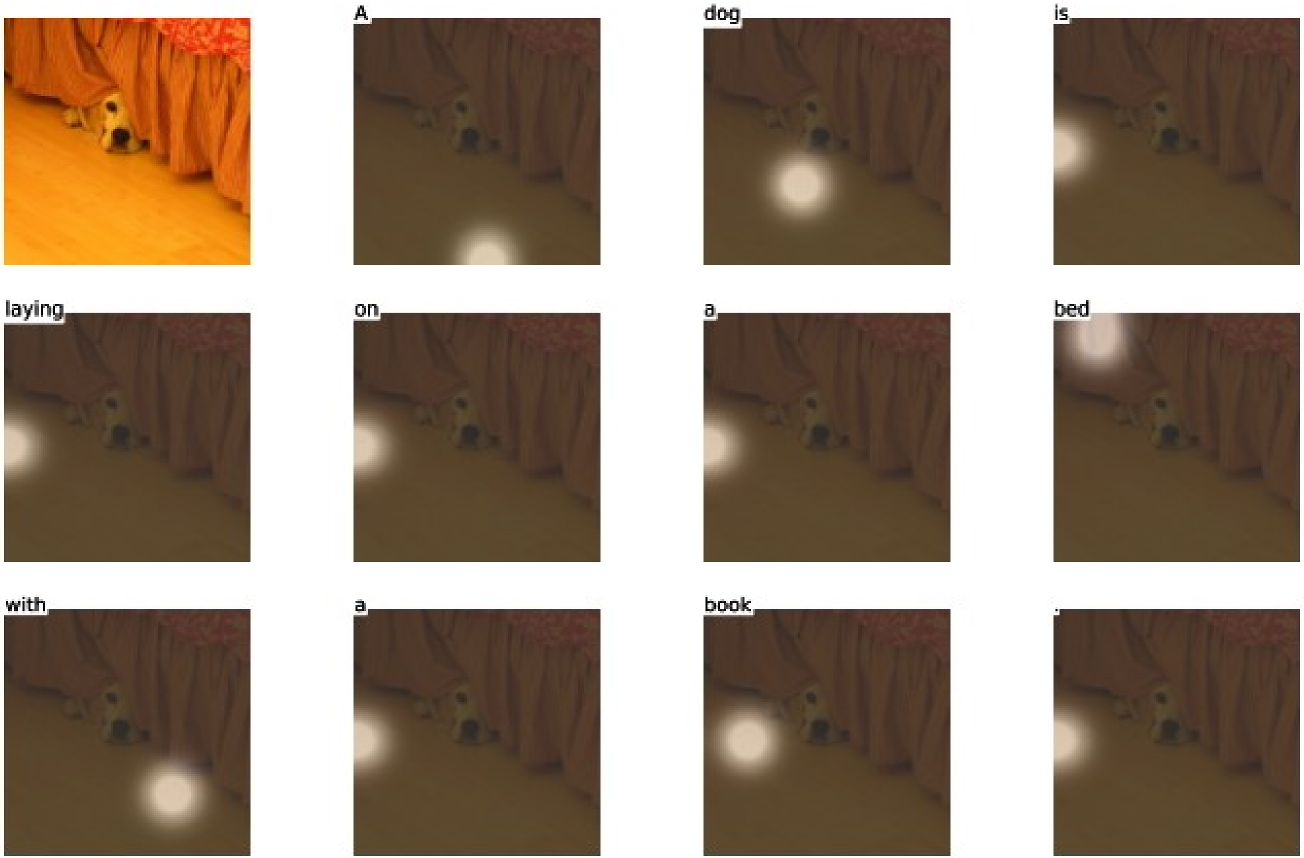
Where $H[s'^n]$ is simply $\sum_{j=1}^{L} p_j \log(p_j)$

- With 0.5 probability, **s** is set to its soft attention value

Positive Example



Negative Example



Training Procedure

- VGG19 is used as the encoder
- 14x14x512 feature map from the 5th conv layer is used
- Evaluated on Flickr8k, Flickr30k, and COCO

Results

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				METEOR
		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

Conclusions

- Hard attention seems to outperform soft attention
- It is not clear whether the improvement was driven due to a better encoder
- Lack of ablation studies
- Interesting approach nonetheless

Thank you!

References

1. Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International Conference on Machine Learning. 2015.
2. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
3. https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-attention-models-upc-2016
4. http://cs231n.stanford.edu/
5. https://github.com/kelvinxu/arctic-captions/blob/master/capgen.py

Discussion Points (Kayvan Tirdad)

- Very strong work and journal paper, cited over 2300 times!
- First version appeared in ICML 2015 then the final version appeared in PMLR 2016
- Author’s original code is in Theano but there are different implementations using tensorflow available on the web, check it out
- This work is mainly based on “Karpathy, Andrej and Li, Fei-Fei. Deep visual-semantic alignments for generating image descriptions. CVPR 2015.” , but that work used an object detection approach instead of attention
- There is some criticism against BLEU so the authors used METEOR as another metric. Soft and hard attention outperform the other approaches. Interestingly, soft attention obtains a better result with METEOR. However, the difference in real-world application is negligible
- Authors didn’t provide a clue which attention mechanism is superior, and why?
- Length of the caption is a tricky issue while training, due to number of times that the LSTM should be run. To remedy this, the authors used a dictionary for storing captions of equal length