LAB SEMINAR

20160804 Lee KyooChul

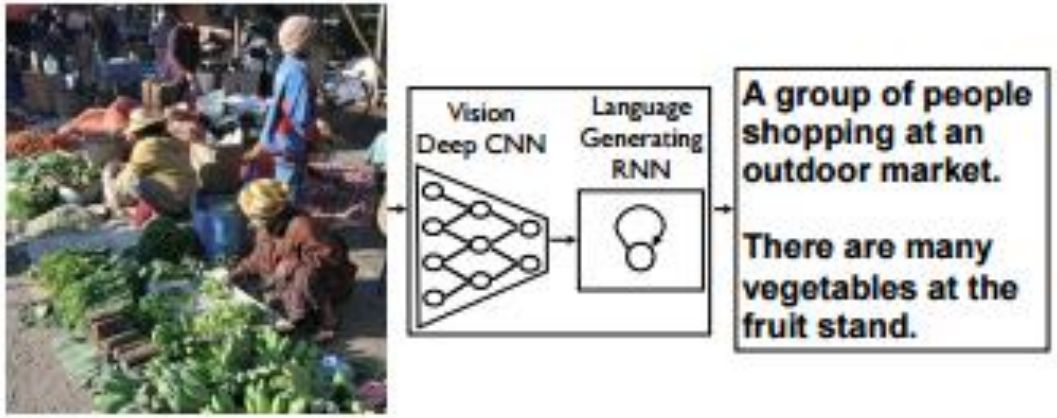
Topic

Topic

Image Caption

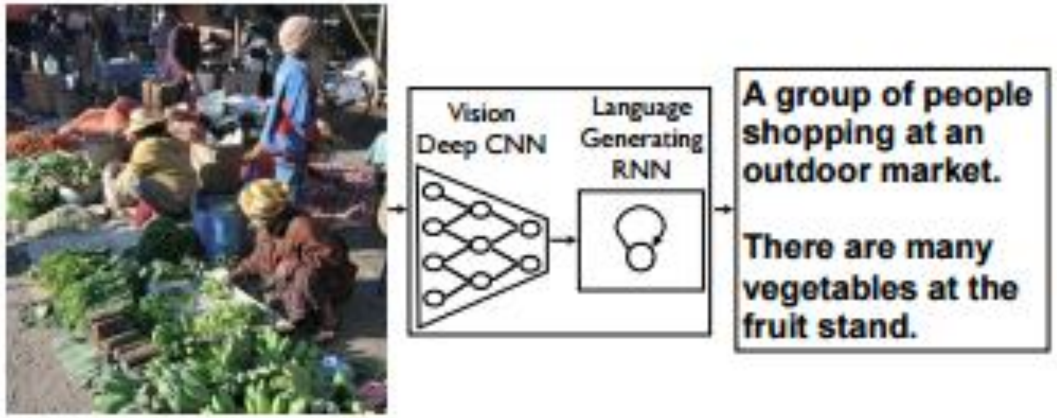
Topic

Image Caption



Topic

Image Caption



- Dealing with two most advanced tasks in the deep learning area(vision & language)
- Picking up the salient information from the dense environment
  → similar environment with vehicle perception

Paper Introduction

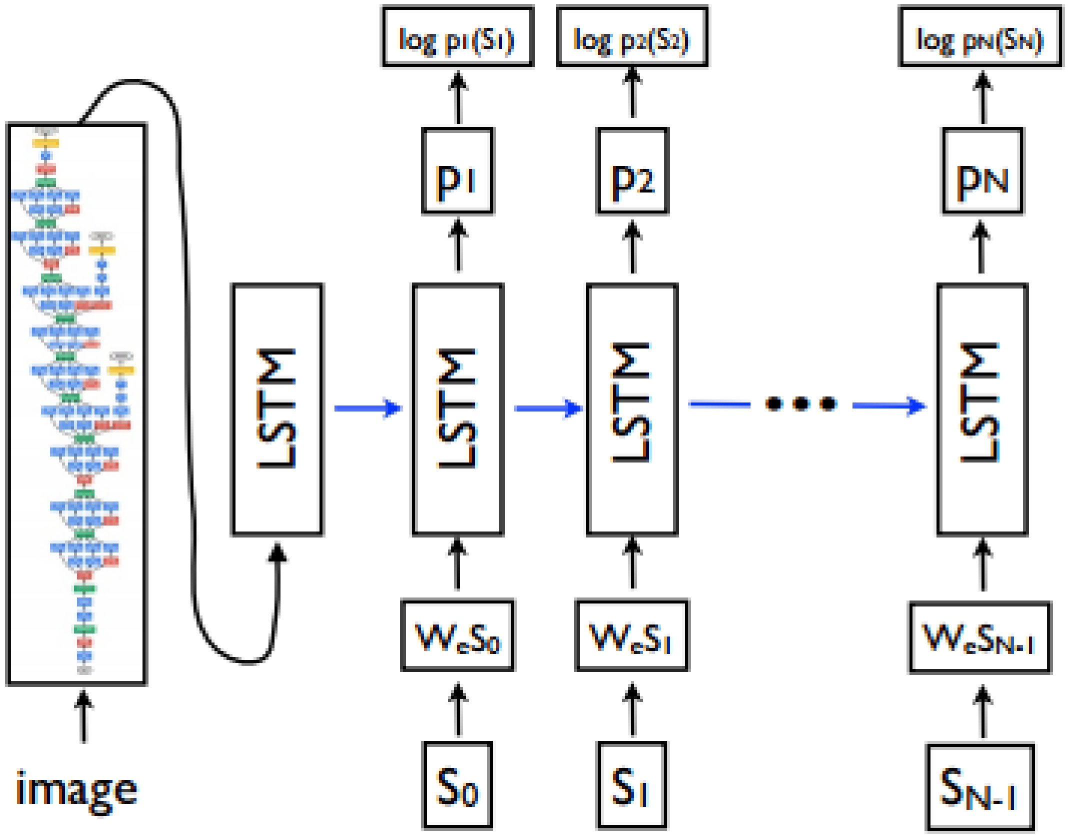
- Show and Tell : A neural Image Caption Generator (NIC)
  - ✓ 2015 CVPR
  - ✓ Oriol Vinyals et al. (Google)

- Show, Attend and Tell : Neural Image Caption Generation with Visual Attention (SAT)
  - ✓ 2015 ICML
  - ✓ Kelvin Xu et al. (Kyunghyun Cho, Yoshua Bengio)

- 1) Model 2) Training 3) Evaluation

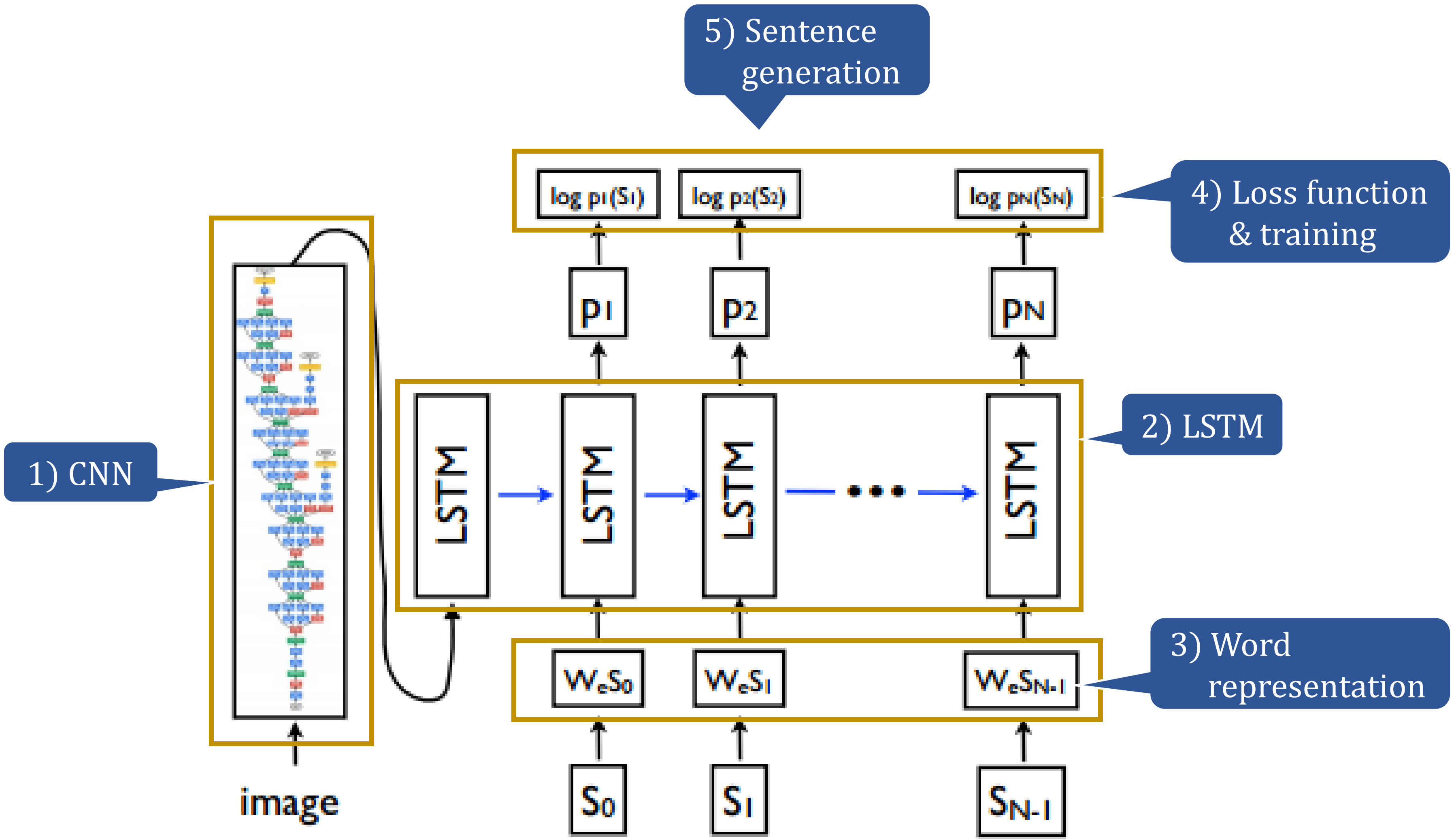
NIC - Model

NIC - Model



log p1(S1)
log p2(S2)
log pN(SN)
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
WeS0
WeS1
WeSN-1
S0
S1
SN-1
image

NIC - Model



5) Sentence generation
4) Loss function & training
2) LSTM
1) CNN
3) Word representation
log p1(S1)
log p2(S2)
log pN(SN)
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
WeS0
WeS1
WeSN-1
S0
S1
SN-1
image

NIC – Convolutional Encoder

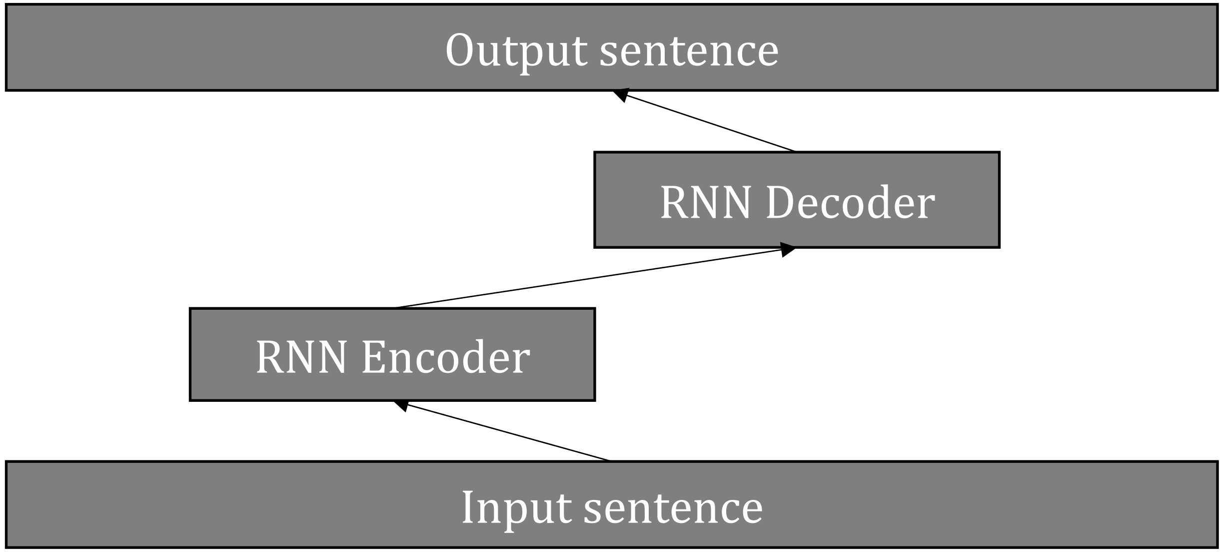
- Encoder-Decoder framework from Machine Translation

NIC – Convolutional Encoder

- Encoder-Decoder framework from Machine Translation
  - ✓ Grammar based methods, Semantic based methods, Statistical methods etc.
  - ✓ Feedforward language modeling (Bengio et al., 2003)
  - ✓ RNN encoder-decoder framework (Cho et al., 2014)

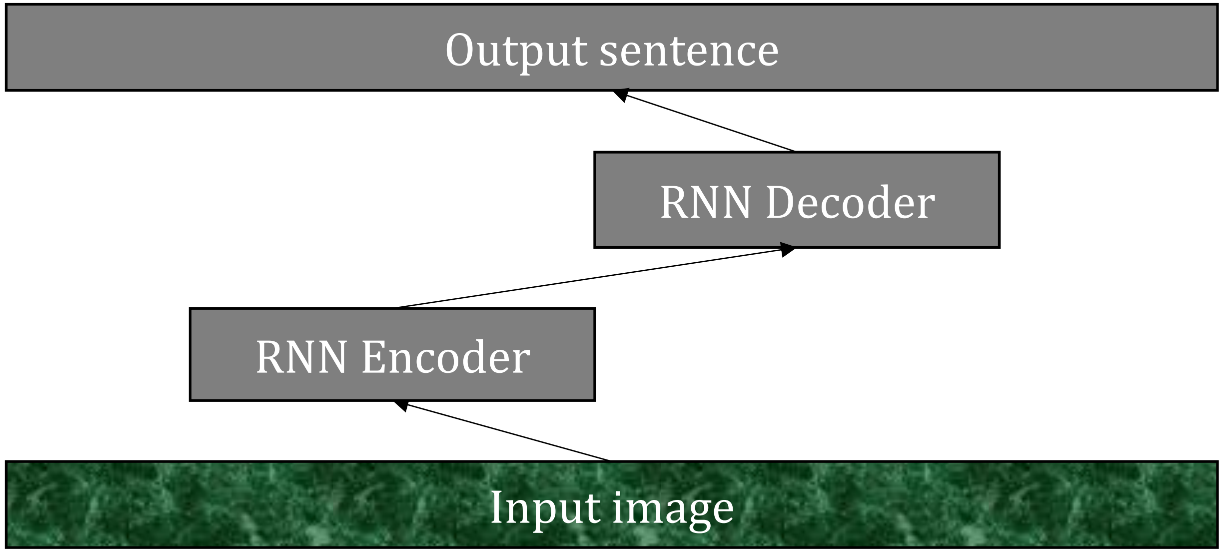
NIC – Convolutional Encoder

- Encoder-Decoder framework from Machine Translation
  - ✓ Grammar based methods, Semantic based methods, Statistical methods etc.
  - ✓ Feedforward language modeling (Bengio et al., 2003)
  - ✓ RNN encoder-decoder framework (Cho et al., 2014)



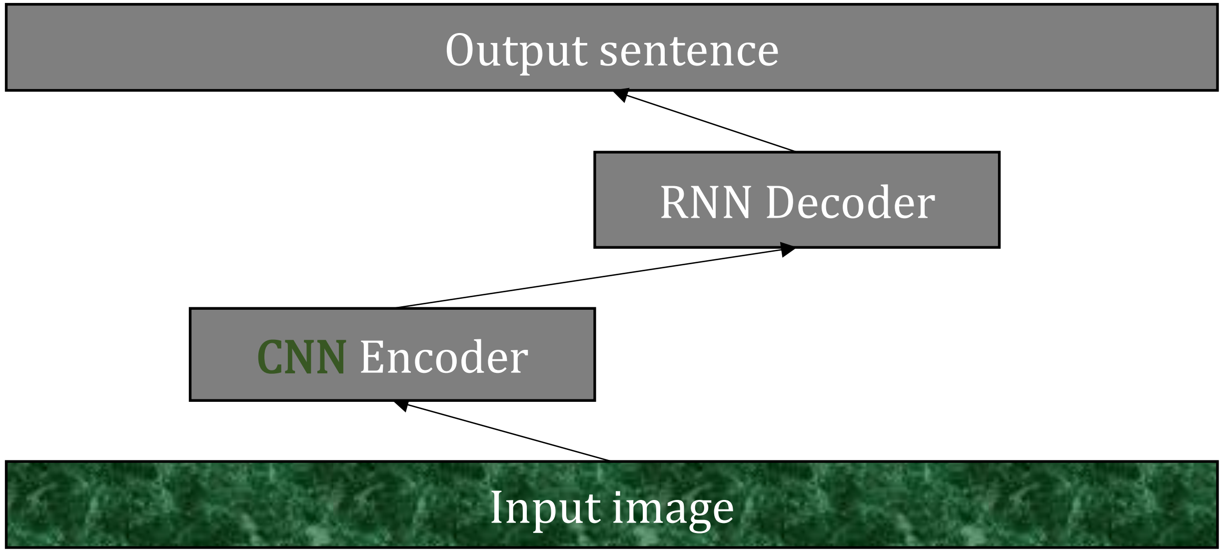
NIC – Convolutional Encoder

- Encoder-Decoder framework from Machine Translation



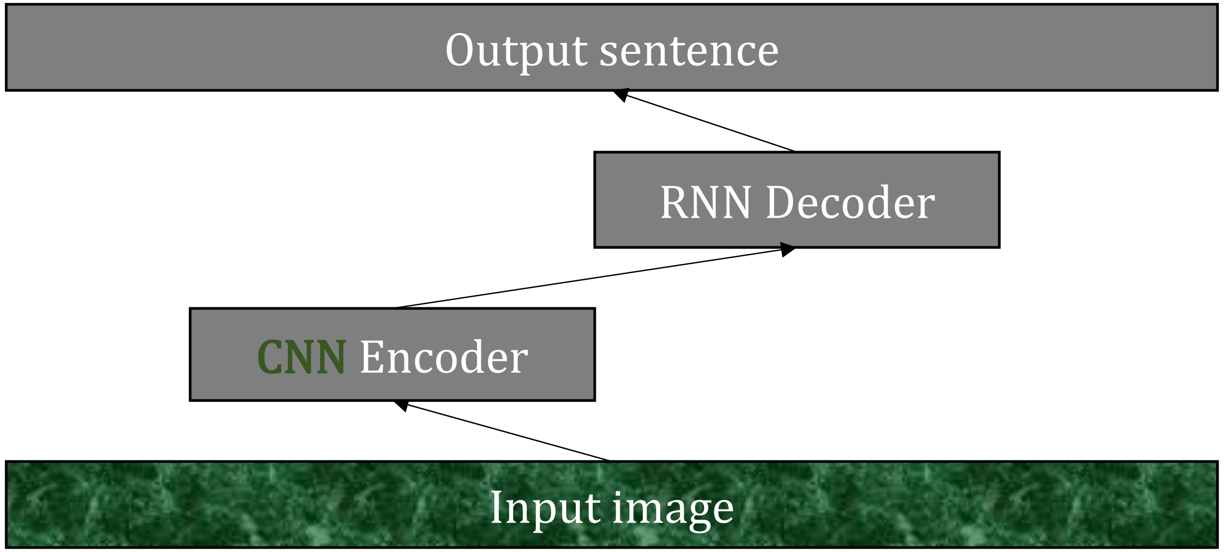
NIC – Convolutional Encoder

- Encoder-Decoder framework from Machine Translation



NIC – Convolutional Encoder

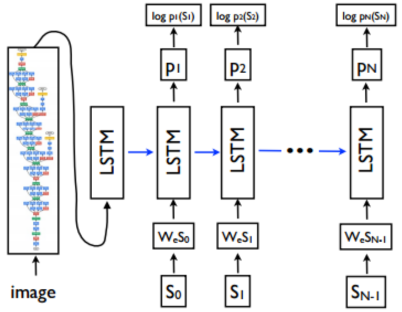
- Encoder-Decoder framework from Machine Translation
  - ✓ Use GooleNet[1] (the winner of the classification competition of ILSVRC 2014)
  - ✓ Pre-trained with ImageNet data
  - ✓ Apply as the input only at the beginning

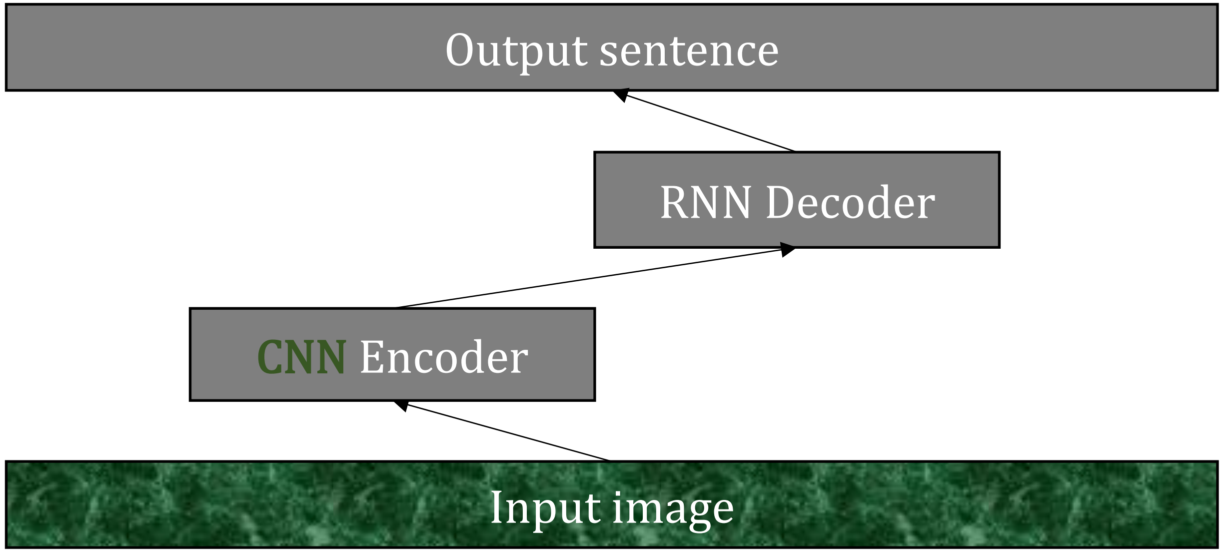


[1] Szegedy et al., Going Deeper with Convolutions. CVPR, 2015

NIC – Convolutional Encoder

- Encoder-Decoder framework from Machine Translation
  - ✓ Use GooleNet[1] (the winner of the classification competition of ILSVRC 2014)
  - ✓ Pre-trained with ImageNet data
  - ✓ Apply as the input only at the beginning



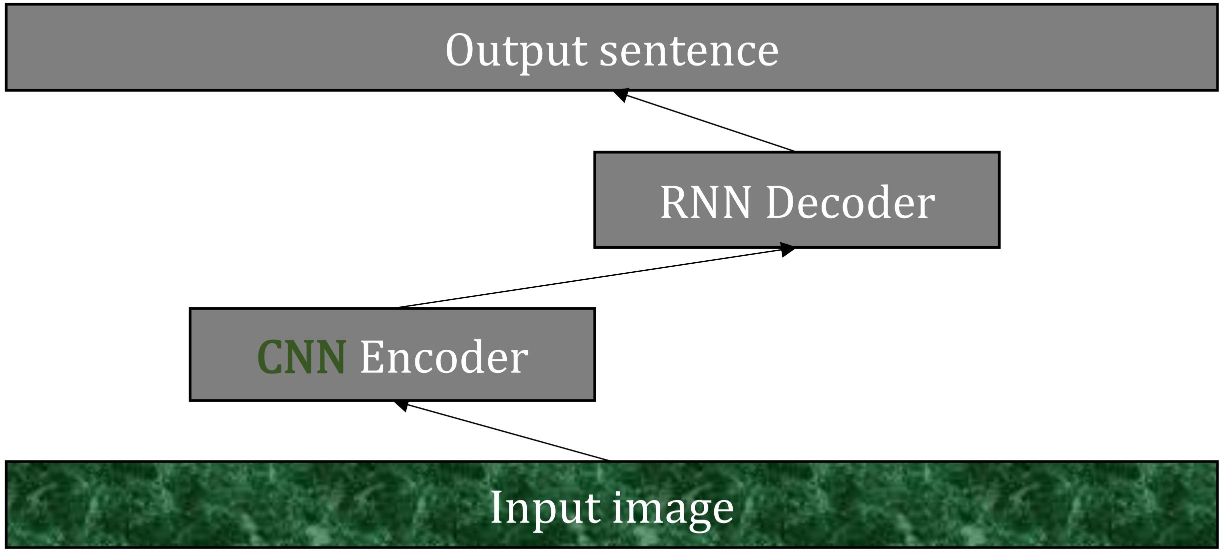


[1] Szegedy et al., Going Deeper with Convolutions. CVPR, 2015

NIC – Convolutional Encoder

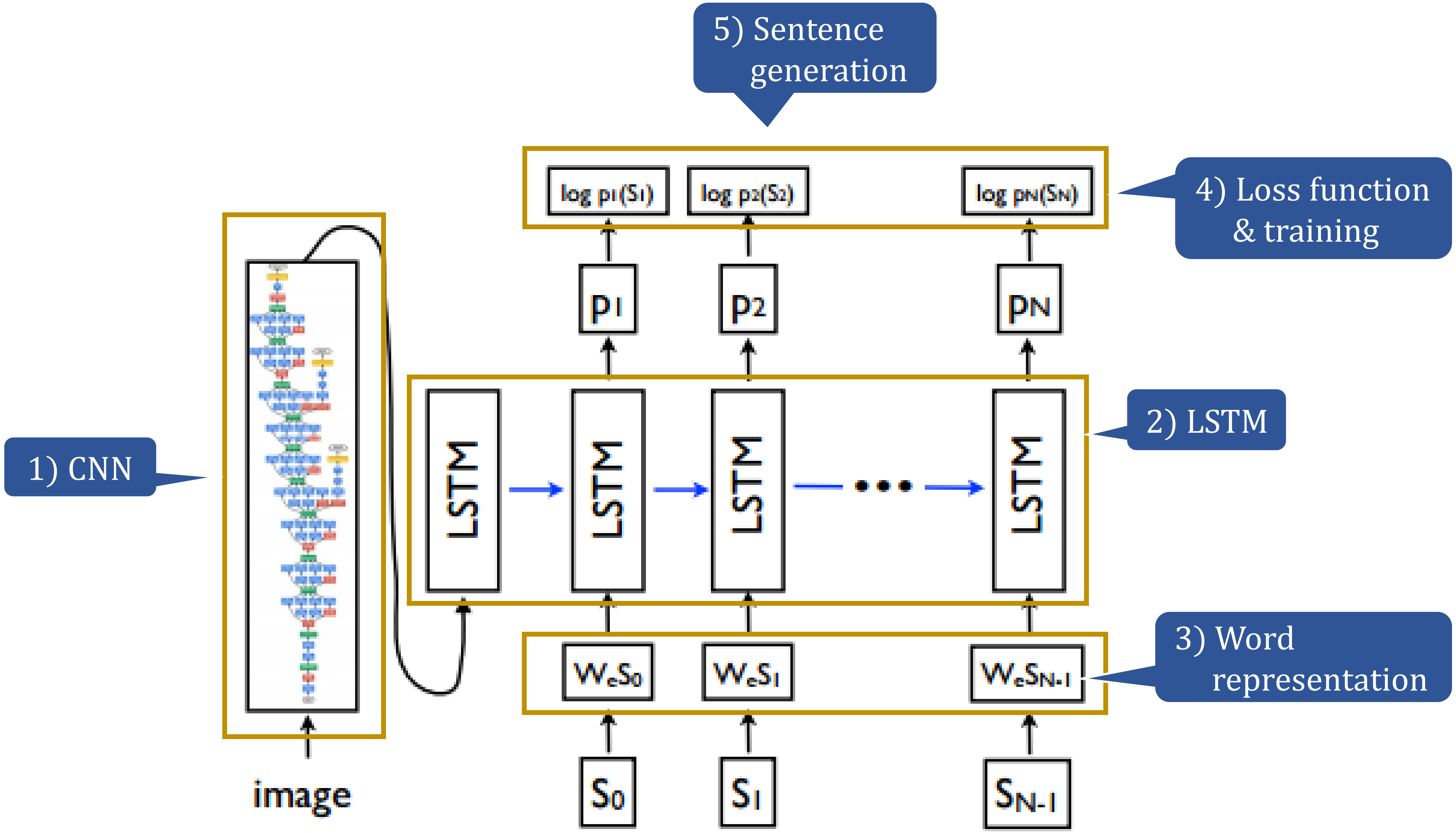
- Encoder-Decoder framework from Machine Translation
  - ✓ Use GooleNet[1] (the winner of the classification competition of ILSVRC 2014)
  - ✓ Pre-trained with ImageNet data
  - ✓ Apply as the input only at the beginning

To avoid overfitting



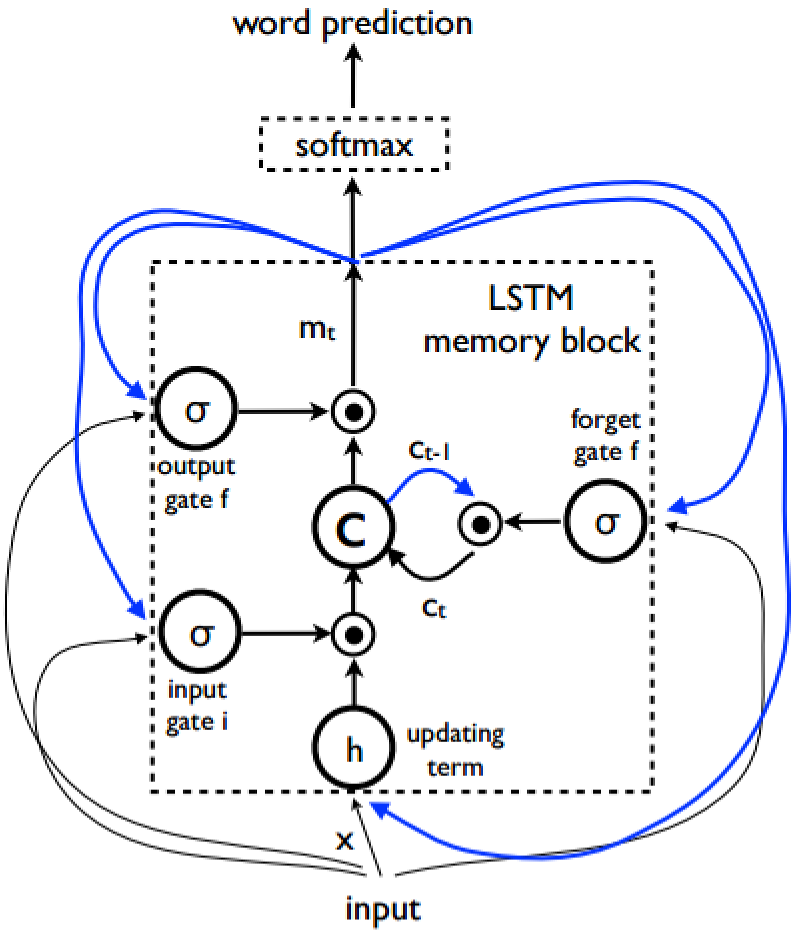
[1] Szegedy et al., Going Deeper with Convolutions. CVPR, 2015

NIC - Model



5) Sentence generation
4) Loss function & training
2) LSTM
1) CNN
3) Word representation
log p1(S1)
log p2(S2)
log pN(SN)
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
WeS0
WeS1
WeSN-1
S0
S1
SN-1
image

NIC – LSTM Decoder



- The goal of LSTM : Remembering the previous state better
- 3 gates

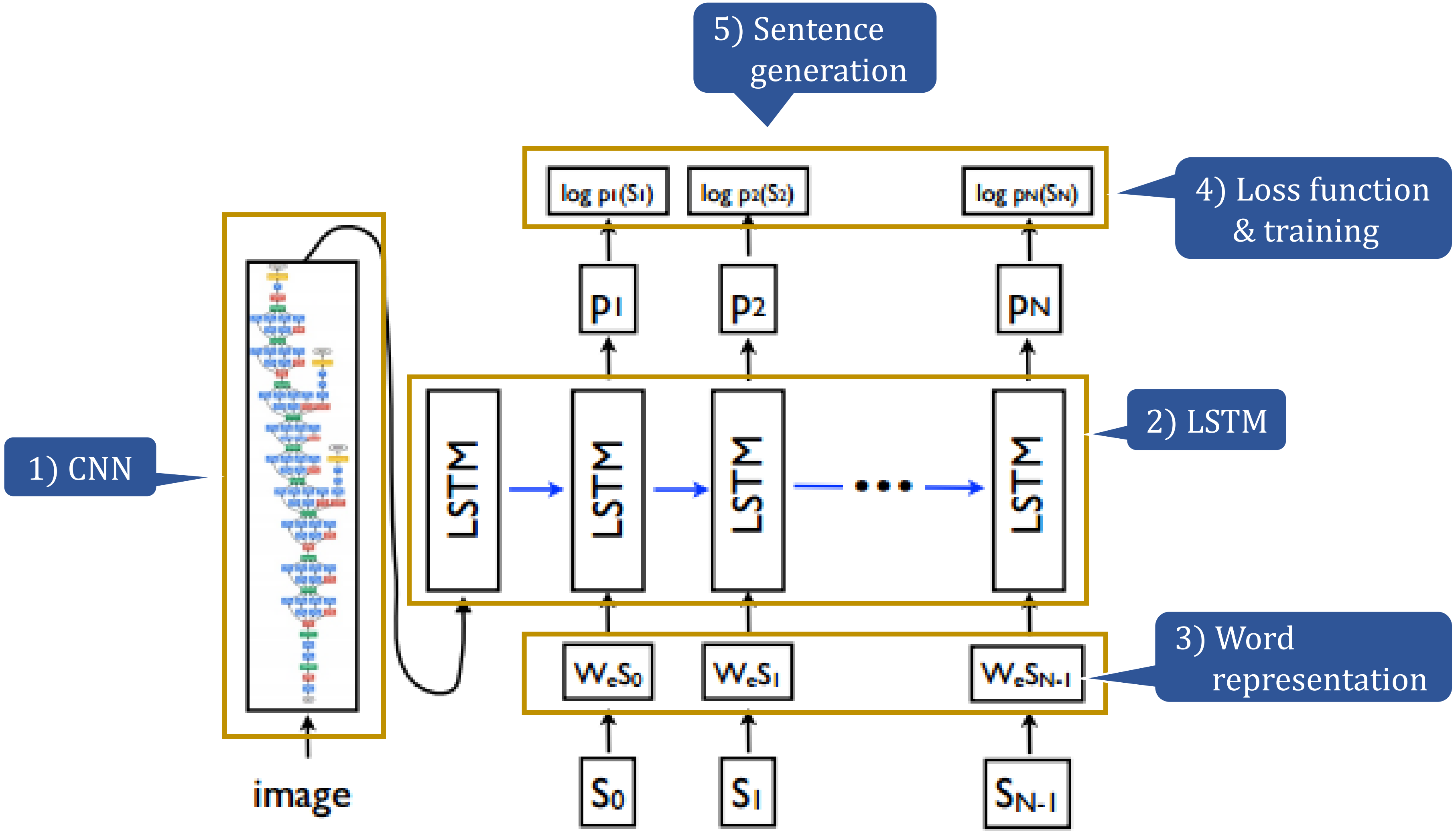
$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1})
\end{aligned}
$$

1 memory cell $\quad c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1})$

1 hidden state $\quad m_t = o_t \odot c_t$

1 output $\quad p_{t+1} = \text{Softmax}(m_t)$

NIC - Model


5) Sentence generation
4) Loss function & training
2) LSTM
1) CNN
3) Word representation
log p1(S1)
log p2(S2)
log pN(SN)
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
WeS0
WeS1
WeSN-1
S0
S1
SN-1
image

NIC – Word embedding

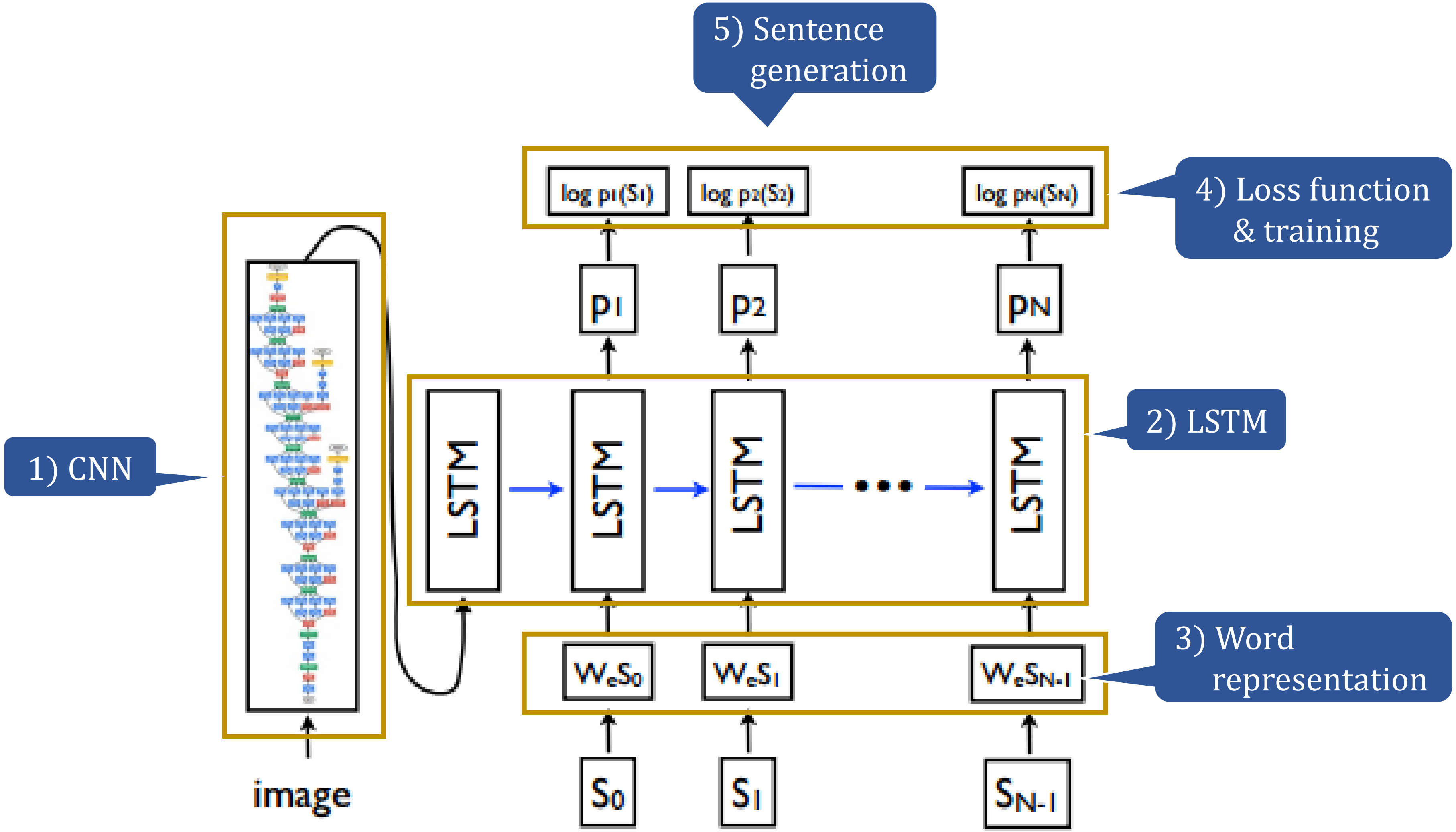
- A word represented as the one-hot vector

0	1	0	0	0	0	➔ 'Computer'
0	0	0	1	0	0	➔ 'Vehicle'

⋮

- Embedding matrix W_e
  - ✓ Mapping the words to the same space with the image
  - ✓ Trainable parameters

NIC - Model


5) Sentence generation
4) Loss function & training
2) LSTM
1) CNN
3) Word representation
log p1(S1)
log p2(S2)
log pN(SN)
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
WeS0
WeS1
WeSN-1
S0
S1
SN-1
image

NIC – Loss function & Training

- $\theta^* = argmax_\theta \sum_{(I,S)} \log p(S|I;\theta)$

I : input image
S : correct transcription of I
θ : parameters of the model

- $\log p(S|I) = \sum_{t=0}^{N} log\, p(S_t|I, S_0, \dots, S_{t-1})$

S $=\{S_0, S_1, \dots, S_N\}$

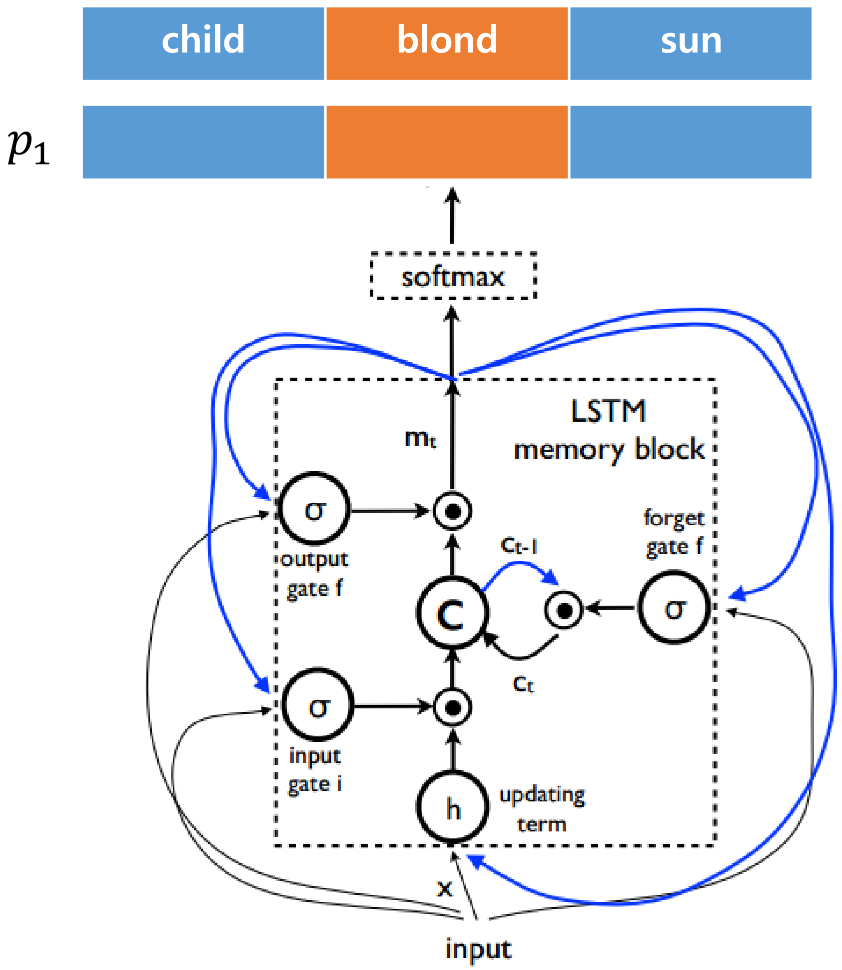
NIC – Loss function & Training

- How the dataset looks

[Flickr30k dataset example]

NIC – Loss function & Training

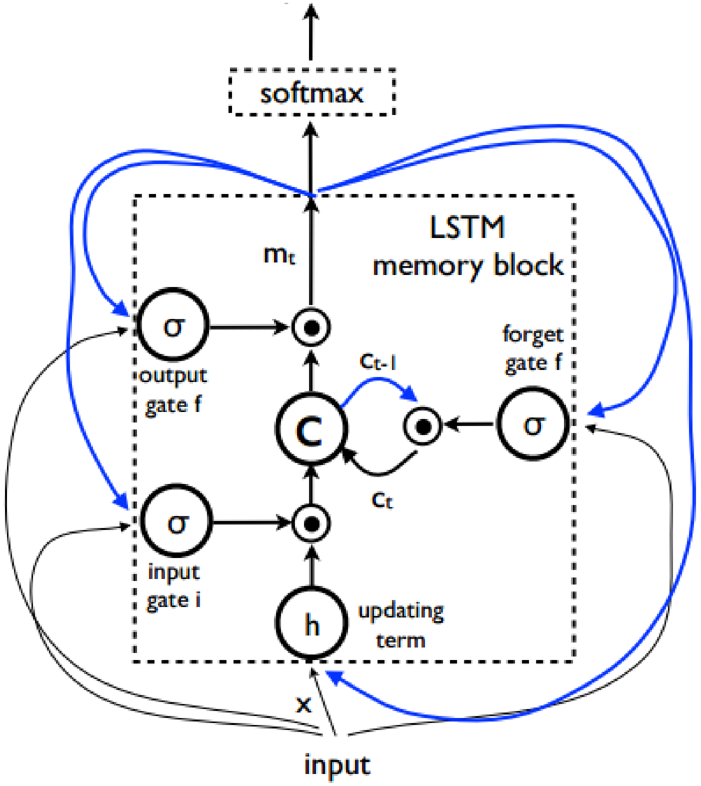
- How the dataset looks



NIC – Loss function & Training

- How the dataset looks

	child	blond	sun
p_1	0.3	0.6	0.1



NIC – Loss function & Training

- Loss function $L(I, S) = -\sum_{t=0}^{N} \log p_t(S_t)$

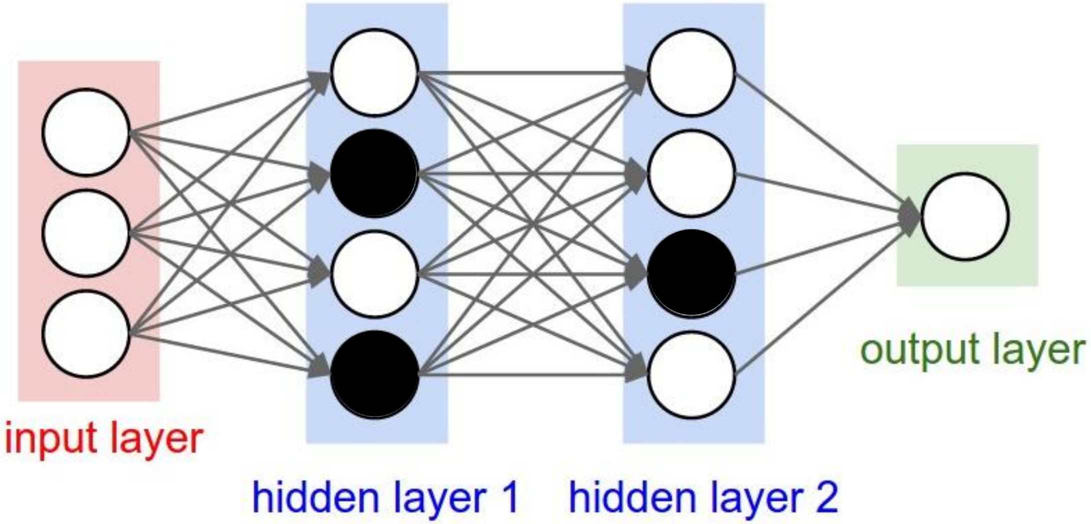
- End-to-End training with stochastic gradient descent

- Techniques for avoiding overfitting

  - ✓ Pre-trained CNN weights (rest of the parameters are initialized randomly)
  - ✓ Dropout & Ensembling the model
  - ✓ Limitation – insufficient number of data (10 times less than the ImageNet)

NIC – Loss function & Training

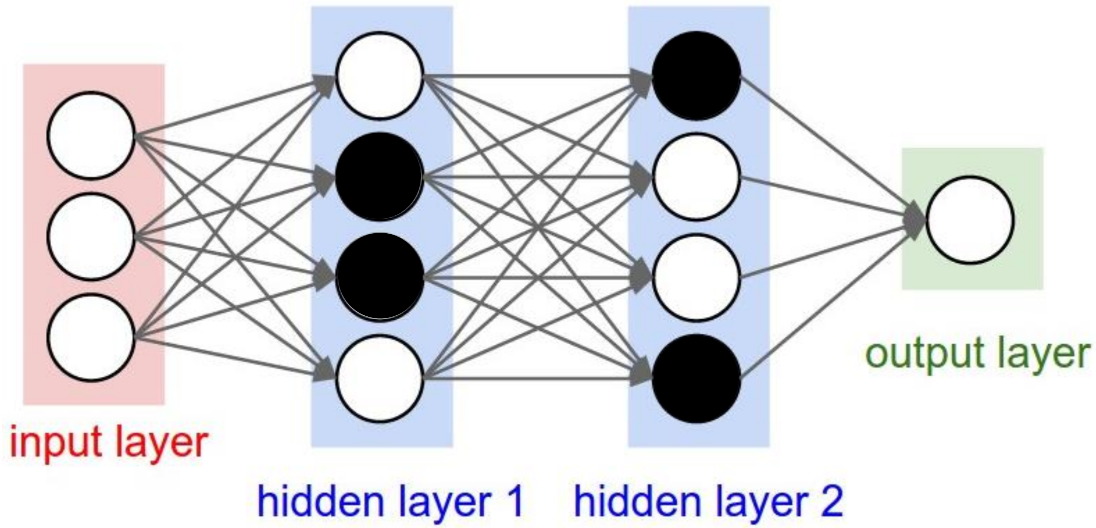
- Dropout



- Ensembling

NIC – Loss function & Training

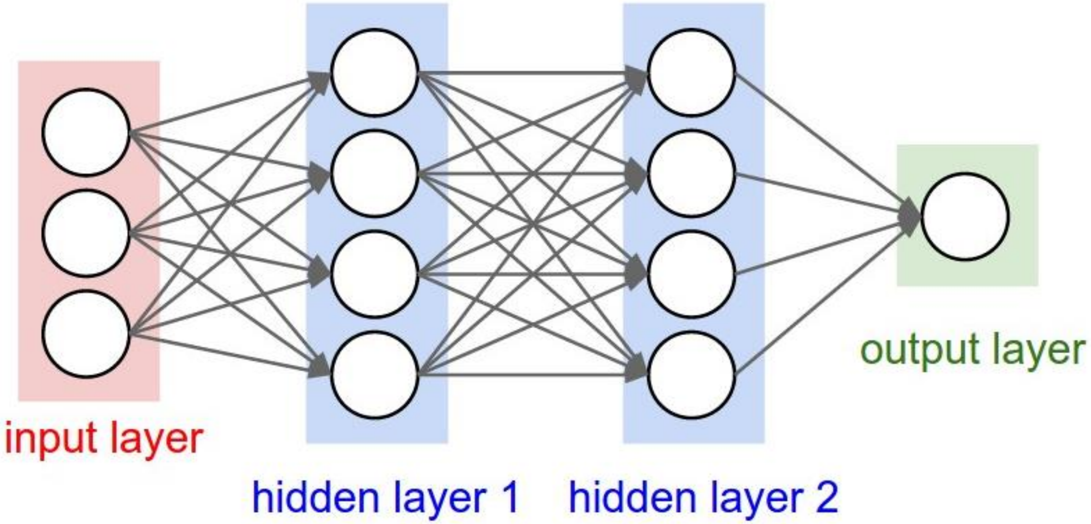
- Dropout



- Ensembling

NIC – Loss function & Training

- Dropout



- Ensembling

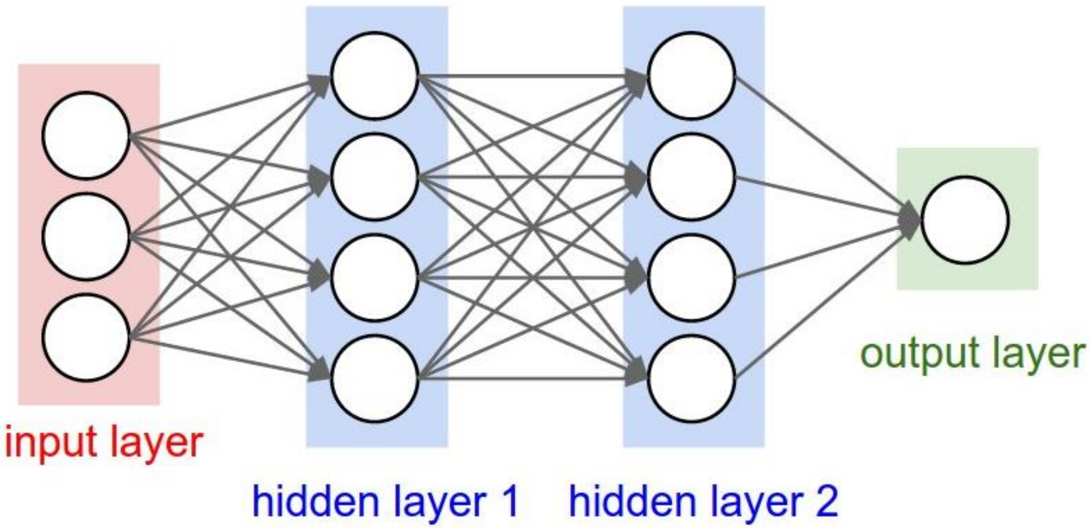
Iteration 2

Iteration 1

Initial weight

NIC – Loss function & Training

- Dropout



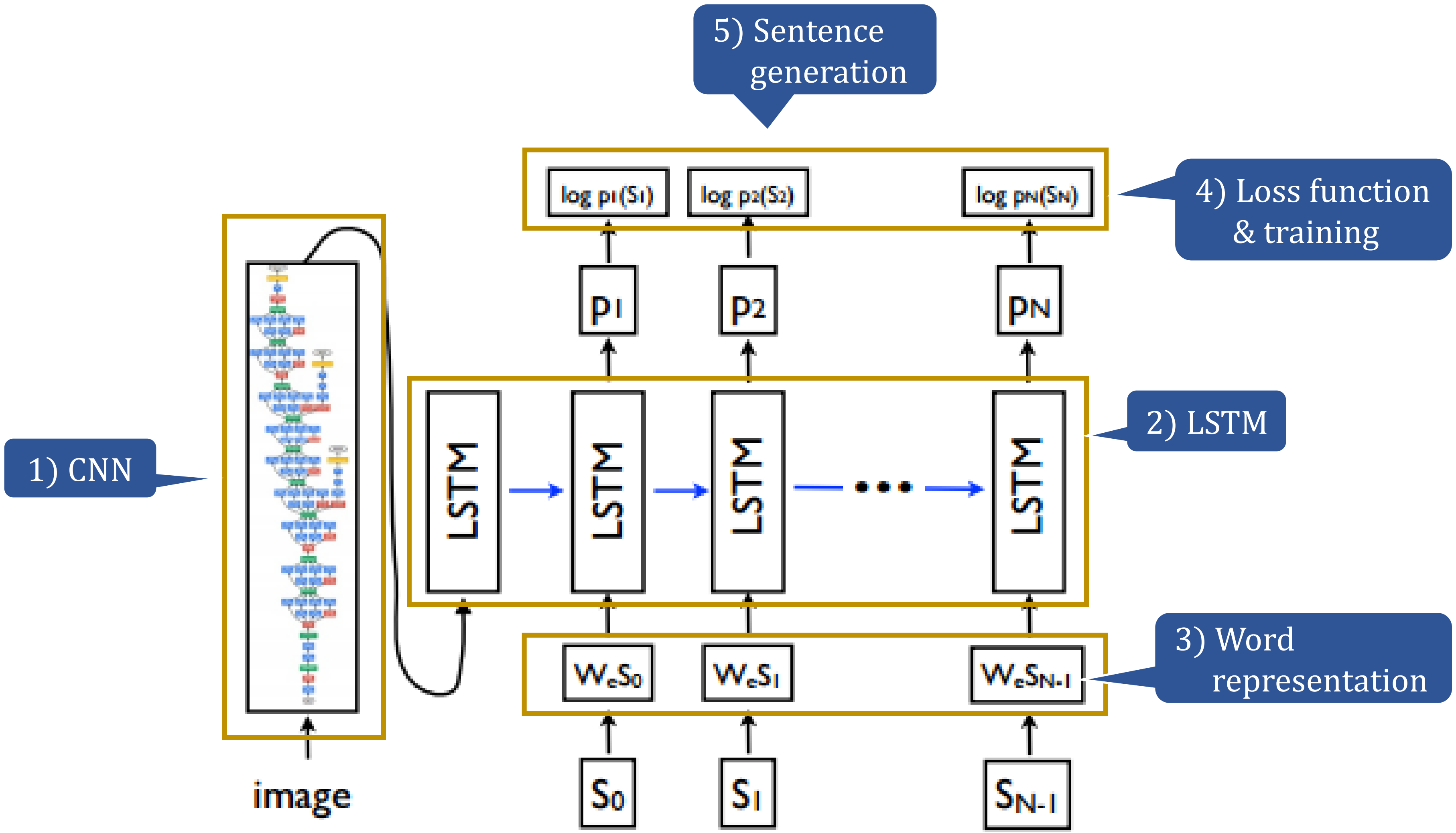
- Ensembling

Iteration 1	Iteration 2
Initial weight	Initial weight

NIC – Loss function & Training

- Loss function $L(I,S) = -\sum_{t=0}^{N} log\, p_t(S_t)$
- End-to-End training with stochastic gradient descent
- Techniques for avoiding overfitting
  - ✓ Pre-trained CNN weights (rest of the parameters are initialized randomly)
  - ✓ Dropout & Ensembling the model
  - ✓ Limitation – insufficient number of data (10 times less than the ImageNet)

NIC - Model



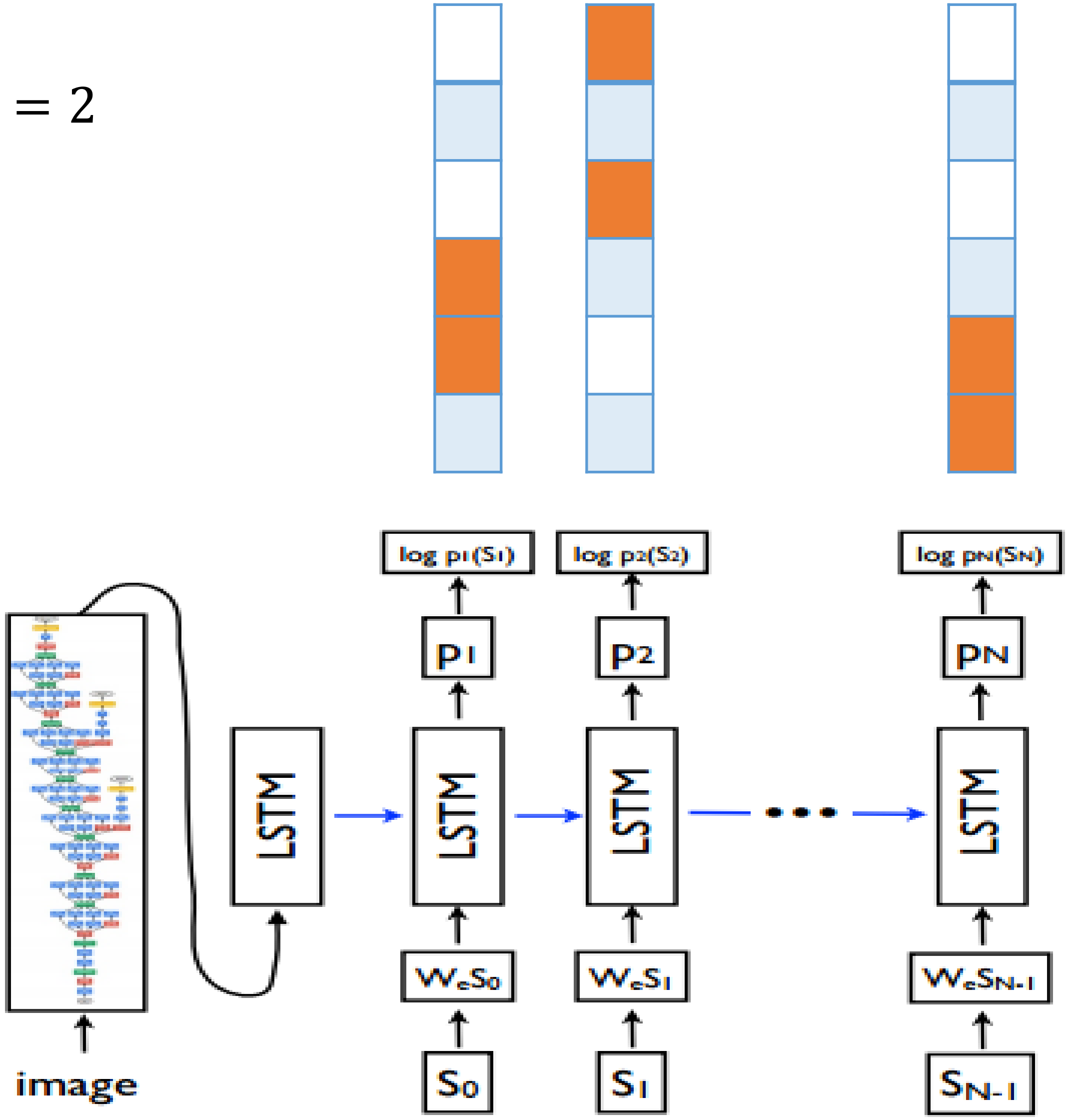
5) Sentence generation
log p1(S1)
log p2(S2)
log pN(SN)
4) Loss function & training
p1
p2
pN
LSTM
LSTM
LSTM
LSTM
2) LSTM
1) CNN
WeS0
WeS1
WeSN-1
3) Word representation
S0
S1
SN-1
image

NIC – Sentence Generation

- Sampling
  - ✓ Sample the n-th word according to p_n
- BeamSearch
  - ✓ Consider only the k best word for each time step
  - ✓ $S = argmax_{S'} p(S'|I)$

NIC – Sentence Generation

- BeamSearch $k = 2$



$$S = argmax_{S'} p(S'|I)$$

NIC – Sentence Generation

- Sampling
  - ✓ Sample the first word according to p_1
- BeamSearch
  - ✓ Consider only the k best sentence for each time step
  - ✓ $S = argmax_{S'} p(S'|I)$
  - ✓ Used in this model with $k = 20$
  - ✓ It is experimentally found that if $k = 1$(greedy search), performance degrades

NIC - Evaluation

- BLEU(Bi-Lingual Evaluation Understudy) score

NIC - Evaluation

- BLEU(Bi-Lingual Evaluation Understudy) score
  - ✓ BLEU-1 score example

Candidate : *It is a guide to action which ensures that the military always obey the commands the party.*

Reference 1 : *It is a guide to action that ensures that the military will forever heed Party commands.*

Reference 2 : *It is the guiding principle which guarantees the military forces always being under the command of the Party.*

Reference 3 : *It is the practical guide for the army always to heed directions of the party*

NIC - Evaluation

- BLEU(Bi-Lingual Evaluation Understudy) score
  - ✓ BLEU-1 score example

Candidate : *It is a guide to action which ensures that the military always obey the commands the party.*

Reference 1 : *It is a guide to action that ensures that the military will forever heed Party commands.*

Reference 2 : *It is the guiding principle which guarantees the military forces always being under the command of the Party.*

Reference 3 : *It is the practical guide for the army always to heed directions of the party*

NIC - Evaluation

- BLEU(Bi-Lingual Evaluation Understudy) score
  - ✓ BLEU-1 score example

Candidate : *It is a guide to action which ensures that the military always obey the commands the party.*

BLEU-1 score : 17

Reference 1 : *It is a guide to action that ensures that the military will forever heed Party commands.*

Reference 2 : *It is the guiding principle which guarantees the military forces always being under the command of the Party.*

Reference 3 : *It is the practical guide for the army always to heed directions of the party*

NIC - Evaluation

- BLEU(Bi-Lingual Evaluation Understudy) score
- METEOR, CIDER …

NIC - Evaluation

- Result

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

[BLEU-1 score]   *SOTA : State-Of-The-Art

Metric	BLEU-4	METEOR	CIDER
NIC	**27.7**	**23.7**	**85.5**
Random	4.6	9.0	5.1
Nearest Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

[Scores on the MSCOCO]

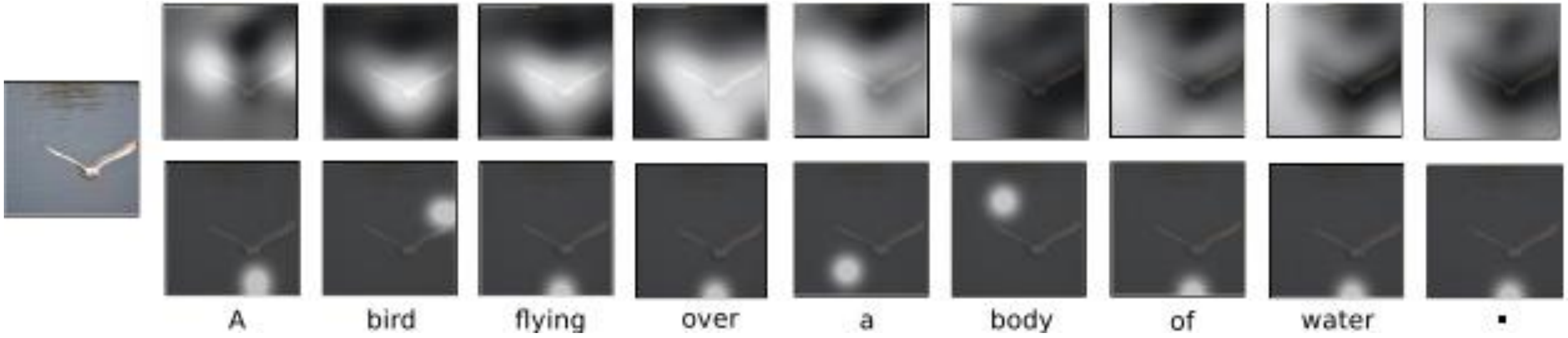
Show, Attend and Tell

Show, Attend and Tell

Show, Attend and Tell

“Soft” attention

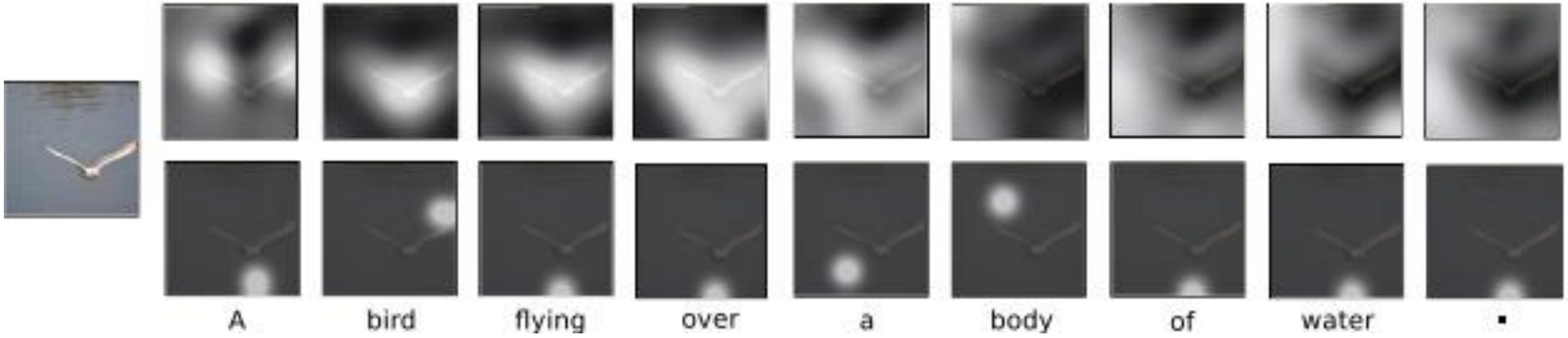
“Hard” attention



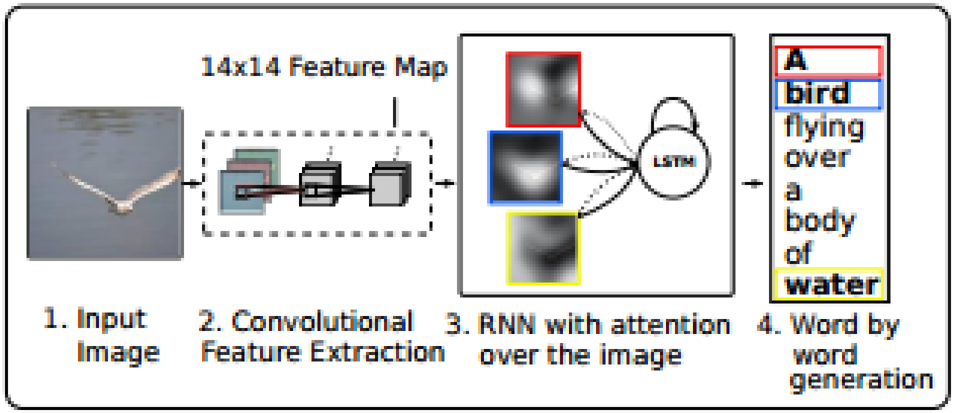
Show, Attend and Tell

“Soft” attention

“Hard” attention



- Model



- LSTM Decoder

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

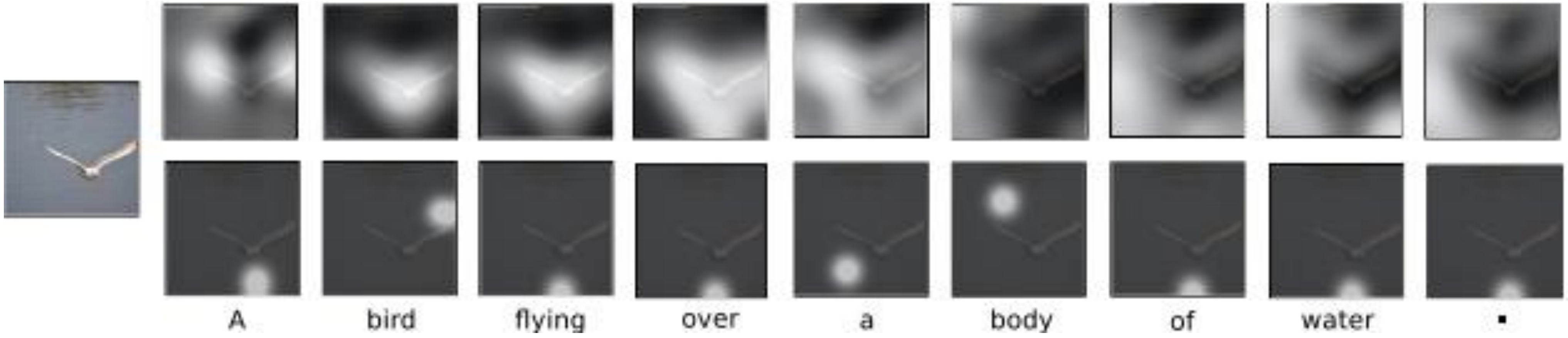
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

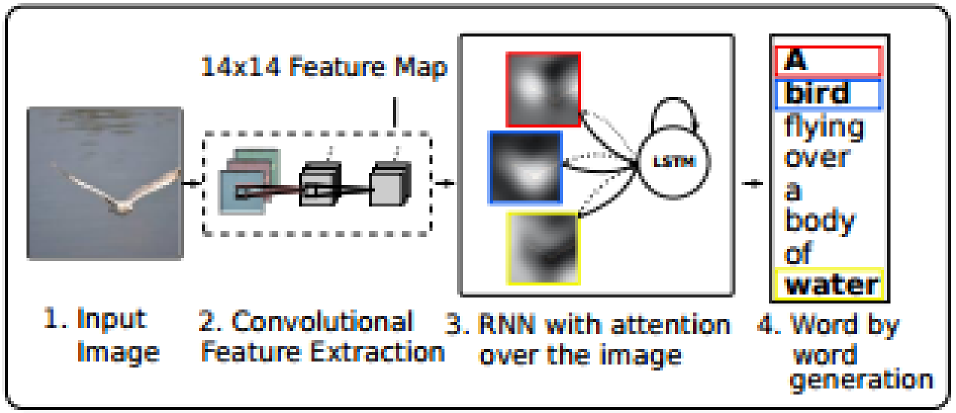
Show, Attend and Tell

"Soft" attention

"Hard" attention



- Model



- LSTM Decoder

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

Context vector

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

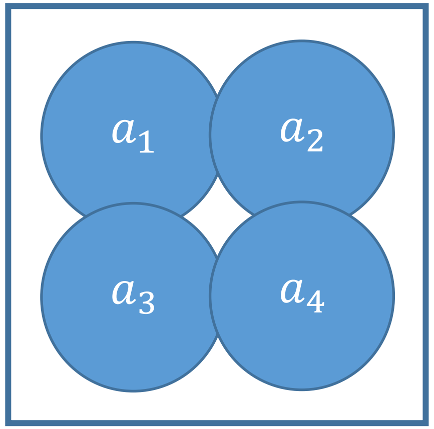
SAT – Attention model

- Context Vector

  ✓ a_i : annotation vector
  the feature vector of the image extracted via the CNN

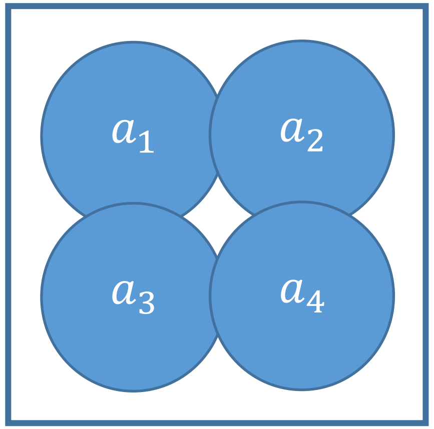
  ✓ $\alpha_i = \frac{\exp(e_{tk})}{\sum_{k=1}^{L} \exp(e_{tk})}$, where $e_{ti} = f_{att}(a_i, h_{t-1})$
  : the probability that location i is the right place to focus to produce the next word

  ✓ $\hat{z}_t = \phi(\{a_i\}, \{\alpha_i\})$



SAT – Attention model

- Context Vector
  - ✓ a_i : annotation vector
    the feature vector of the image extracted via the CNN
  - ✓ $\alpha_i = \frac{\exp(e_{tk})}{\sum_{k=1}^{L}\exp(e_{tk})}$, where $e_{ti} = f_{att}(a_i, h_{t-1})$
    : the probability that location i is the right place to focus to produce the next word
  - ✓ $\hat{z}_t = \phi(\{a_i\}, \{\alpha_i\})$



- Word inference
  - ✓ $p(y_t|a, y_{1:t-1}) \propto \exp(L_o(Ey_{t-1} + L_h h_t + L_z \hat{z}_t))$

SAT – Stochastic "Hard" Attention

- Attention model

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i}$$

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

1) A location variable
2) One-hot encoded
3) A latent variable
4) Assigned with a multinoulli distribution parametrized by $\{\alpha_i\}$

- Loss function → Maximizing the lower bound of the marginal log-likelihood $\log p(y|a)$

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

$$= \log \sum_s p(s|a) p(y|s,a)$$

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

SAT – Stochastic “Hard” Attention

- Derivative

$$\frac{\partial L_s}{\partial W} = \sum_s p(s|a)\left[\frac{\partial logp(y|s,a)}{\partial W} + logp(y|s,a)\frac{\partial \log p(s|a)}{\partial W}\right]$$

- Monte Carlo based sampling approximation

$$\tilde{s}_t \sim Multinoulli_L(\{\alpha_i\})$$

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

SAT – Stochastic "Hard" Attention

- Reinforcement learning

Game	SAT
Action in the game	Choosing s_t
Game score	$p(y\|a)$
Reward	± 1
Objective function	L_s
Parameter update	$W^* = W + \alpha \frac{\partial L_s}{\partial W}$
State	image

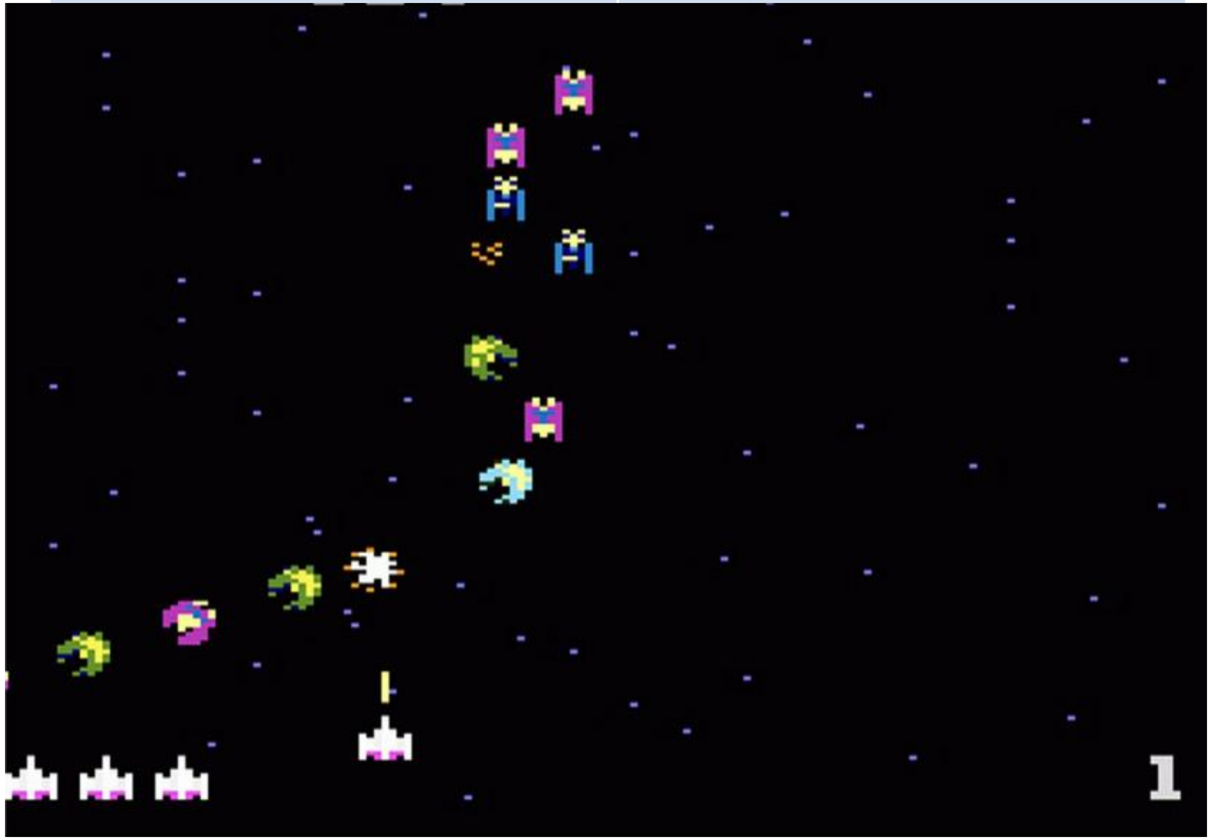
$$\tilde{s}_t \sim Multinoulli_L(\{\alpha_i\})$$

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

SAT – Stochastic “Hard” Attention

- Reinforcement learning

Game	SAT



$$\tilde{s}_t \sim Multinoulli_L(\{\alpha_i\})$$

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

SAT – Stochastic “Hard” Attention

- Reinforcement learning

Game	SAT
Action in the game	Choosing s_t
Game score	$p(y\|a)$
Reward	± 1
Objective function	L_s
Parameter update	$W^* = W + \alpha \frac{\partial L_s}{\partial W}$
State	image

$$\tilde{s}_t \sim Multinoulli_L(\{\alpha_i\})$$

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

SAT – Deterministic “Soft” Attention

$$E_{p(s_t|a)}[\hat{z}_t] = \sum_z zP(Z = z)$$

$$= \sum_z \left(\sum_{j=1}^{L} s_{t,j} a_j \right) P(s_{t,i} = 1|a)$$

$$= \sum_s a_i \alpha_{t,i} = \sum_{i=1}^{L} \alpha_{t,i} a_i \stackrel{\text{def}}{=} \phi(\{a_i\}, \{\alpha_i\})$$

SAT – Deterministic "Soft" Attention

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

$$E_{p(s_t|a)}[\hat{z}_t] = \sum_z zP(Z=z)$$

$$= \sum_z \left(\sum_{j=1}^{L} s_{t,j} a_j\right) P(s_{t,i} = 1|a)$$

$$= \sum_s a_i \alpha_{t,i} = \sum_{i=1}^{L} \alpha_{t,i} a_i \stackrel{\text{def}}{=} \phi(\{a_i\},\{\alpha_i\})$$

From the LSTM equation, $h_t = o_t \odot \tanh(c_t)$

$\tanh(x) \approx x - \frac{1}{3}x^3 \ldots$

$E_{p(s_t|a)}[h_t]$ → forward propagation with $E_{p(s_t|a)}[\hat{z}_t]$

SAT – Deterministic "Soft" Attention

$$E_{p(s_t|a)}[\hat{z}_t] = \sum_z zP(Z = z)$$

$$= \sum_z \left(\sum_{j=1}^{L} s_{t,j}a_j\right) P(s_{t,i} = 1|a)$$

$$= \sum_s a_i \alpha_{t,i} = \sum_{i=1}^{L} \alpha_{t,i} a_i \stackrel{\text{def}}{=} \phi(\{a_i\},\{\alpha_i\})$$

From the LSTM equation, $h_t = o_t \odot \tanh(c_t)$

$\tanh(x) \approx x - \frac{1}{3}x^3 \ldots$

$E_{p(s_t|a)}[h_t]$ → forward propagation with $E_{p(s_t|a)}[\hat{z}_t]$

Let $n_t = L_o(Ey_{t-1} + L_h h_t + L_z \hat{z}_t)$

Considering NWGM for the softmax k-th word prediction,

$$NWGM[p(y_t = k|a)] = \frac{\prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}} = \frac{\exp(E_{p(s_t|a)}[n_{t,k}])}{\sum_j \exp(E_{p(s_t|a)}[n_{t,k}])}$$

$$\approx E[p(y_t = k|a)]$$ (Baldi & Sadowski, 2014)

➔ The expectation of the outputs is computed by simple forward propagation with $E[\hat{z}_t]$

➔ End-to-end learning using standard backpropagation becomes possible

SAT – Training

- Pre-trained Oxford VGGnet[2]
- Dropout & Early stopping on BLEU score (to avoid overfitting)
- 3 days on the NVIDIA Titan Black GPU

[2] Simonyan, K and Zisserman, A. Very deep convolutional networks for large-scale image recognition. CoRR, 2014

SAT - Evaluation

Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k Human : 63	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 Human : 66	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

SAT - Evaluation

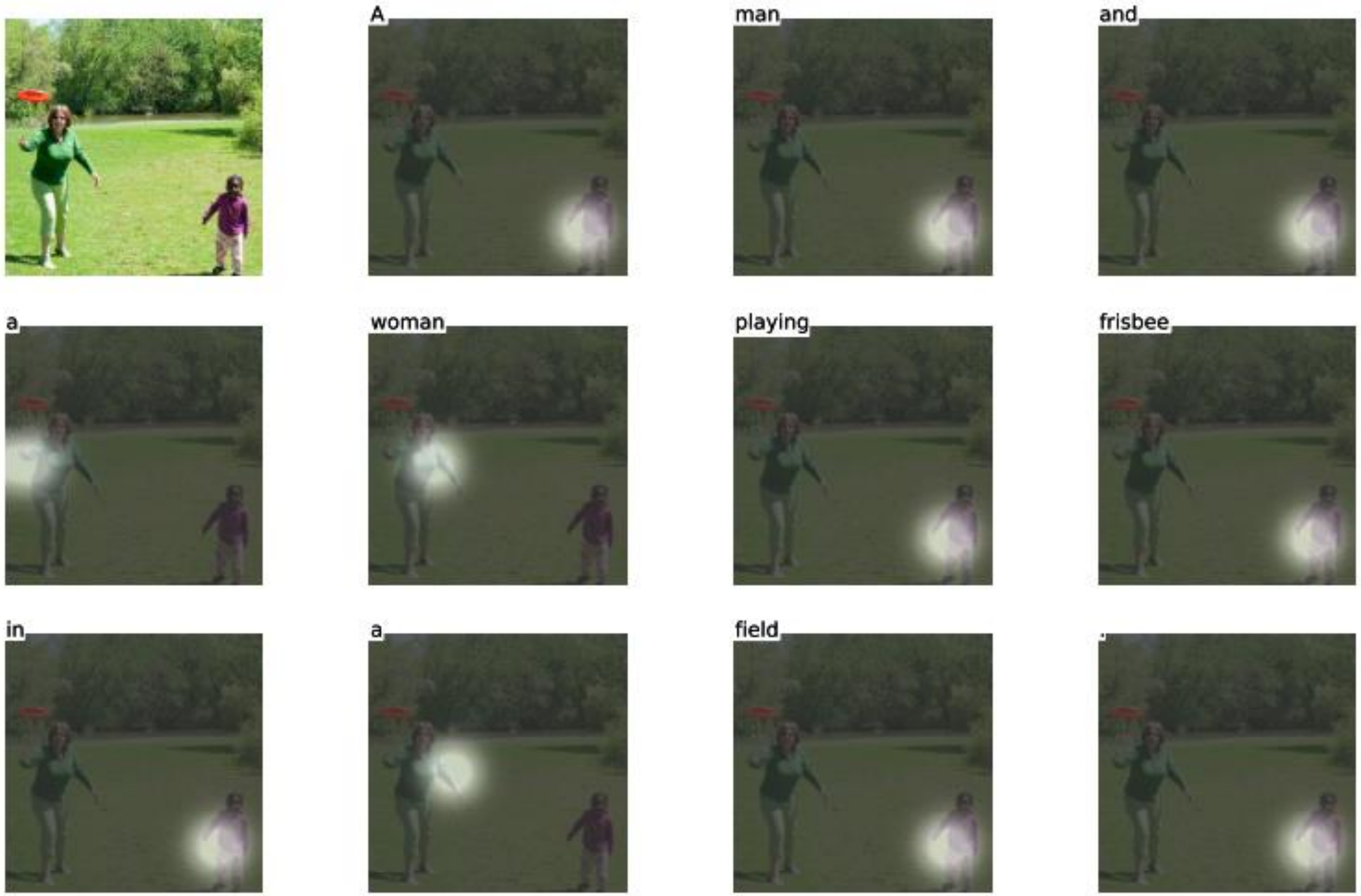
Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k Human : 63	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k Human : 66	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

SAT - Evaluation

Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k Human : 63	Google NIC(Vinyals et al., 2014)[†Σ]	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)[∘]	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 Human : 66	Google NIC[†∘Σ]	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)[†a]	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)[∘]	64.2	45.1	30.4	20.3	—
	Google NIC[†∘Σ]	66.6	46.1	32.9	24.6	—
	Log Bilinear[∘]	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

SAT - Evaluation

- Hard attention



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

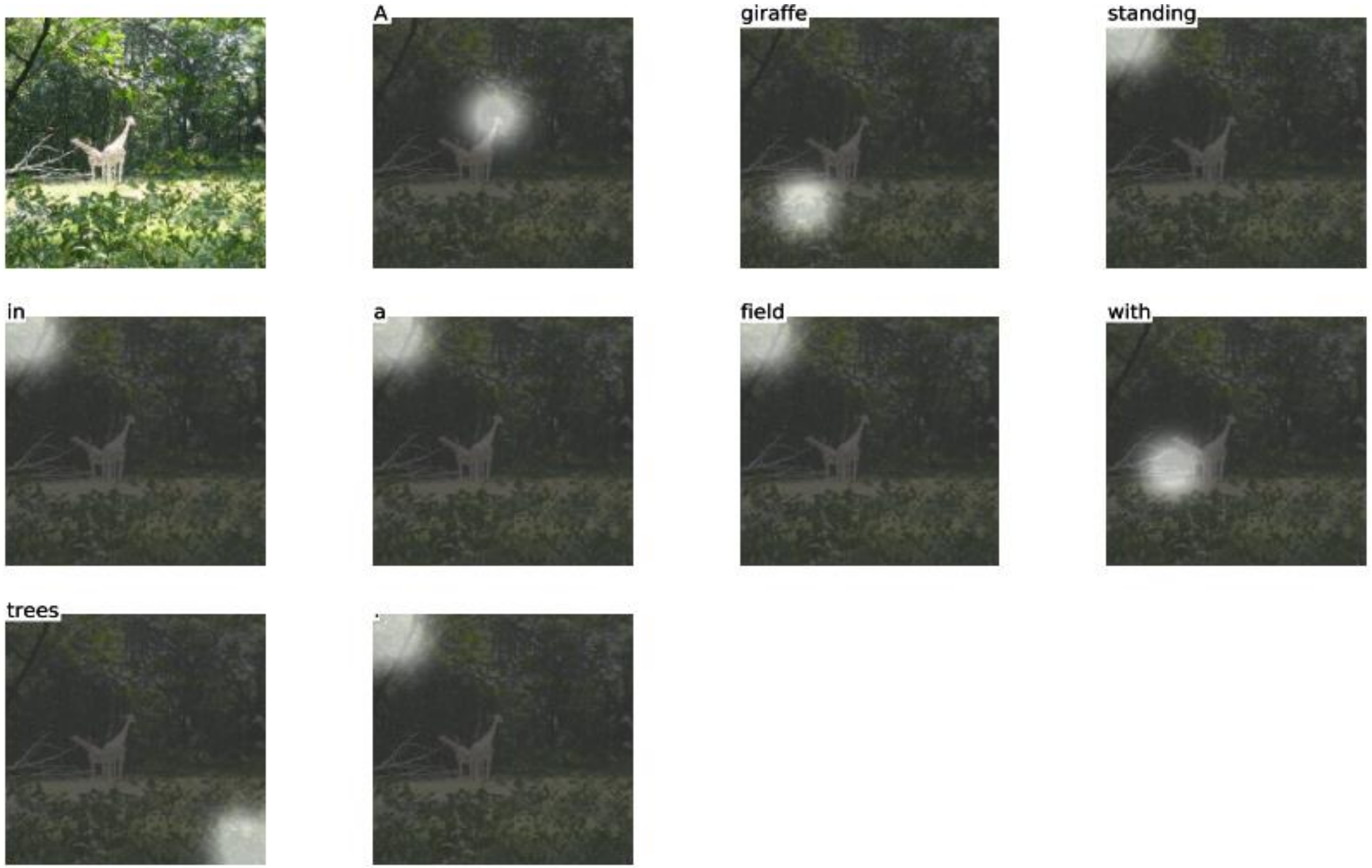
SAT - Evaluation

- Soft attention

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

SAT - Evaluation

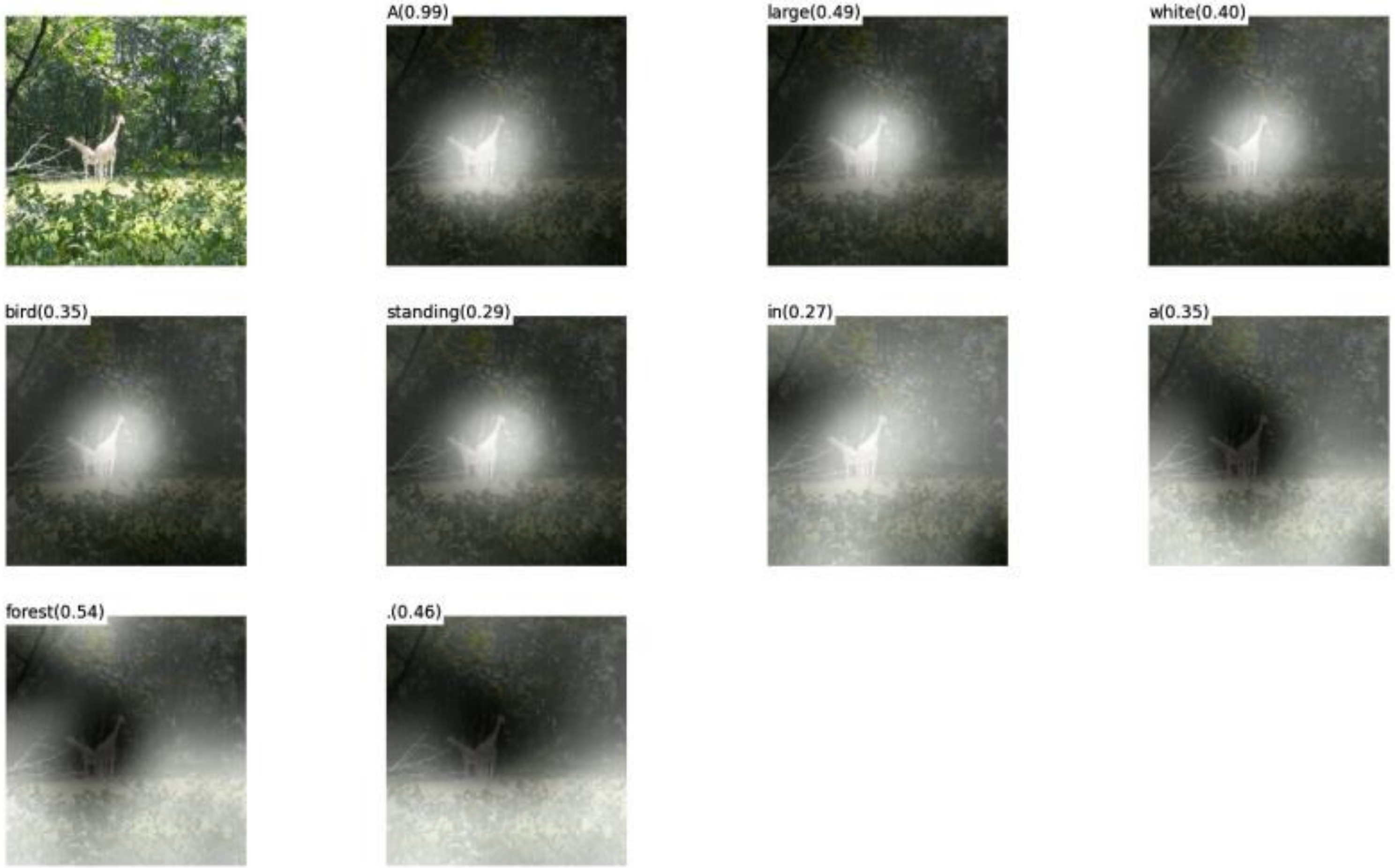
- Hard attention



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

SAT - Evaluation

- Soft attention



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

Conclusion

- NIC
  - ✓ Great improvement with Encoder-Decoder framework
  - ✓ End-to-End trainable system
- SAT
  - ✓ Attention to the salient part of the image for caption generation
  - ✓ Stochastic “Hard” attention & Deterministic “Soft” attention
- Some notes
  - ✓ Encoder-Decoder framework is effective
  - ✓ CNN with pre-trained weights is better for generalization
  - ✓ The amount of the good data is essential for the performance