Summer Review 2
Hard Attention

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
Recurrent Models of Visual Attention
Multiple Object Recognition with Visual Attention

Reviewed by : Arshdeep Sekhon

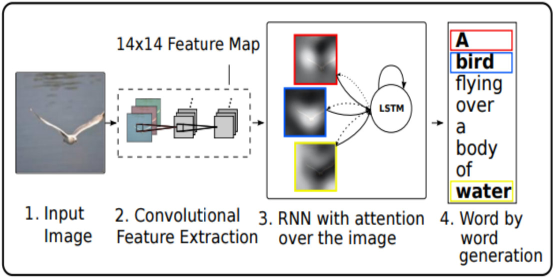
[1]Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

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

The Task

Image captioning with two types of attention: 'hard' and 'soft'

- Input: Image
- Use a CNN to extract images: $\{\boldsymbol{a_1}, \cdots, \boldsymbol{a_L}\}$ in $\boldsymbol{a_i} \in R^D$
- Output is $\{\boldsymbol{y_1}, \cdots, \boldsymbol{y_C}\}$ in $\boldsymbol{y_i} \in R^K$



Common framework

- $h_t = LSTM(\boldsymbol{\hat{z}_t}, Ey_{t-1}, h_{t-1})$
- $e_{ti} = f_{att}(\boldsymbol{a_i}, \boldsymbol{h_t})$
- $\alpha_t i = Softmax(e_{ti})$ over L image features
- $\hat{z}_t = \phi(\{\boldsymbol{a_i}\}, \{\alpha_i\})$
- Prediction: $p(\boldsymbol{y_t}|\boldsymbol{a}, \boldsymbol{y_{t-1}}) \propto exp(\boldsymbol{L_0}(\boldsymbol{Ey}_{t-1} + \boldsymbol{L_h h_t} + \boldsymbol{L_z \hat{z}_t}))$

Key Idea

$s_{t,i}$ is an indicator one-hot variable which is set to 1 if the i-th location (out of L) is the one used to extract visual features. By treating the attention locations as intermediate latent variables, we can assign a multinoulli distribution.

- $p(s_{t,i} = 1|s_{j<t}, \boldsymbol{a}) = \alpha_{t,i}$
- $\boldsymbol{\hat{z}_t} = \sum_{i=1} s_{t,i}\boldsymbol{a_i}$
- Minimize variational lower bound on the marginal log-likelihood $log(p(\boldsymbol{y}|\boldsymbol{a}))$

$$log(p(\boldsymbol{y}|\boldsymbol{a})) = log \sum_{s} p(s|\boldsymbol{a})p(\boldsymbol{y}|s, \boldsymbol{a}) \tag{1}$$

$$\geq \sum_{s} p(s|\boldsymbol{a}) log p(\boldsymbol{y}|s, \boldsymbol{a}) \tag{2}$$

Training Hard Attention

$$\frac{\partial L_s}{\partial W} = \sum_s p(s|\boldsymbol{a}) \left[\frac{\partial \log p(\boldsymbol{y}|s, \boldsymbol{a})}{\partial W} + \partial \log p(\boldsymbol{y}|s, \boldsymbol{a}) \frac{p(s|\boldsymbol{a})}{\partial W} \right] \tag{3}$$

- $$s_t \sim Multinoulli_L(\{\alpha_i\}) \tag{4}$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_s p(s^n|\boldsymbol{a}) \left[\frac{\partial \log p(\boldsymbol{y}|s^n, \boldsymbol{a})}{\partial W} + \log p(\boldsymbol{y}|s^n, \boldsymbol{a}) \frac{\partial p(s^n|\boldsymbol{a})}{\partial W} \right] \tag{5}$$

Soft Attention

- Take direct expectation of $\hat{z}_t$:

$$E_{p(s_t|a)}[\hat{\boldsymbol{z}}_{\boldsymbol{t}}] = \sum_{i=1}^{L} \alpha_{t,i} \boldsymbol{a}_{\boldsymbol{i}} \tag{6}$$

- a deterministic soft attention model
- Another variation: Doubly stochastic attention
- $\sum_t \alpha_{t,i} \approx 1$

RAM:Recurrent Models of Visual Attention

- Glimpse Sensor : $z_t = f_g(x_t, l_{t-1})$
- Internal State: $h_t = LSTM(h_{t-1}, z_t)$ of LSTM
- Actions: a_t, l_t: find next state or perform some action
- $l_t \sim p(|f_l(h_t; \theta_l))$
- The policy for the locations l was defined by a two-component Gaussian with a fixed variance. The location network outputs the mean of the location policy at time t
- $a_t \sim p(|f_a(h_t; \theta_a))$
- Reward: $R = \sum_{t=1}^{T} r_t$

Partially Observable Markov Decision Process

States not directly observable, learn stochastic policy $\pi((a_t, l_t)|s_{1:t}; \theta)$
$s_{1:t} = x_1, a_1, l_1, x_2, a_2, l_2, \ldots, x_t, a_{t-1}, l_{t-1}$

Training using REINFORCE algorithm

$$J(\theta) = E_{p_{s_{1:T};\theta}}\left[\sum_{t=1}^{T} r_t\right] = E_{p_{s_{1:T};\theta}} R \tag{7}$$

$$J(\theta) = E_{p_{s_{1:T};\theta}} \nabla_\theta \log \pi(u^t | s_{1:t}; \theta) R \tag{8}$$

$$J(\theta) = \frac{1}{M} \sum_{i=1}^{M} \nabla_\theta \log \pi(u_t^i | s_{1:t}^i; \theta) R^i \tag{9}$$

Multiple Object Recognition with Visual Attention

- Task is Object Classification after G glimpses
- Same as above RAM (architecture) and Show, Attend and Tell (optimization)
- No prediction at each glimpse

$$\sum_{\ell} p(\ell|I, W) \log p(y|\ell, I, W) \tag{10}$$

$$\frac{\partial \mathcal{F}}{\partial W} = \sum_{l} p(l|I,W)\frac{\partial \log p(y|l,I,W)}{\partial W} + \sum_{l} \log p(y|l,I,W)\frac{\partial p(l|I,W)}{\partial W}$$
$$= \sum_{l} p(l|I,W)\left[\frac{\partial \log p(y|l,I,W)}{\partial W} + \log p(y|l,I,W)\frac{\partial \log p(l|I,W)}{\partial W}\right]$$

Figure: Optimization

Optimization

$$\tilde{l}^m \sim p(l_n | I, W) = \mathcal{N}(l_n; \hat{l}_n, \Sigma)$$

$$\frac{\partial \mathcal{F}}{\partial W} \approx \frac{1}{M} \sum_{m=1}^{M} \left[\frac{\partial \log p(y | \tilde{l}^m, I, W)}{\partial W} + \log p(y | \tilde{l}^m, I, W) \frac{\partial \log p(\tilde{l}^m | I, W)}{\partial W} \right]$$

Figure: Optimization

Table 1: Error rates on the MNIST pairs classification task.

Model	Test Err.
RAM Mnih et al. (2014)	9%
DRAM w/o context	7%
DRAM	**5%**

Table 2: Error rates on the MNIST two digit addition task.

Model	Test Err.
ConvNet 64-64-64-512	3.2%
DRAM	**2.5%**

Figure: RAM and DRAM results for object classification

Experiments

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

Figure: Image Captioning Results

References

- Mnih, Volodymyr, Nicolas Heess, and Alex Graves. "Recurrent models of visual attention." Advances in neural information processing systems. 2014.
- Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International Conference on Machine Learning. 2015.
- Ba, Jimmy, Volodymyr Mnih, and Koray Kavukcuoglu. "Multiple object recognition with visual attention." arXiv preprint arXiv:1412.7755 (2014).

Extra for REINFORCE Algorithm

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture14.pdf

REINFORCE algorithm

Expected reward: $$J(\theta) = \mathbb{E}_{\tau \sim p(\tau;\theta)}\left[r(\tau)\right] = \int_{\tau} r(\tau)p(\tau;\theta)\mathrm{d}\tau$$

Now let's differentiate this: $$\nabla_{\theta} J(\theta) = \int_{\tau} r(\tau)\nabla_{\theta} p(\tau;\theta)\mathrm{d}\tau$$

Intractable! Gradient of an expectation is problematic when p depends on θ

However, we can use a nice trick: $$\nabla_{\theta} p(\tau;\theta) = p(\tau;\theta)\frac{\nabla_{\theta} p(\tau;\theta)}{p(\tau;\theta)} = p(\tau;\theta)\nabla_{\theta}\log p(\tau;\theta)$$

If we inject this back:

$$\nabla_{\theta} J(\theta) = \int_{\tau} \left(r(\tau)\nabla_{\theta}\log p(\tau;\theta)\right) p(\tau;\theta)\mathrm{d}\tau$$
$$= \mathbb{E}_{\tau \sim p(\tau;\theta)}\left[r(\tau)\nabla_{\theta}\log p(\tau;\theta)\right]$$

Can estimate with Monte Carlo sampling

Lower bound

$$
\begin{aligned}
\ln p(x;\theta) &= \int q(z)\ln p(x;\theta)dz \\
&= \int q(z)\ln\left(\frac{p(x;\theta)p(z|x;\theta)}{p(z|x;\theta)}\right)dz \\
&= \int q(z)\ln\left(\frac{p(x,z;\theta)}{p(z|x;\theta)}\right)dz \\
&= \int q(z)\ln\left(\frac{p(x,z;\theta)q(z)}{p(z|x;\theta)q(z)}\right)dz \\
&= \int q(z)\ln\left(\frac{p(x,z;\theta)}{q(z)}\right)dz - \int q(z)\ln\left(\frac{p(z|x;\theta)}{q(z)}\right)dz \\
&= F(q,\theta) + KL(q||p)
\end{aligned}
$$

Lower bound

Jensen's Inequalityon the log probability of observations

$$\ln p(y) = \mathcal{L} + D_{KL},$$

where

$$\mathcal{L} = \sum_s q(s) \log \frac{p(y,s)}{q(s)}$$

and

$$D_{KL} = -\sum_s q(s) \log \frac{p(s|y)}{q(s)}.$$

So that using

$$p(y,s) = p(y|s)q(s),$$

we have

$$\begin{aligned} \ln p(y) &= \sum_s q(s) \log \frac{p(y,s)}{q(s)} + D_{KL} \\ &= \sum_s q(s) \log \frac{p(y|s)q(s)}{q(s)} + D_{KL} \\ &= \sum_s q(s) \log p(y|s) + D_{KL} \\ &= L_s + D_{KL} \end{aligned}$$

Now since $D_{KL} \geq 0$ we have $L_s \leq \log p(y)$ which is the sense in which it is a "lower bound" on the log probability. To complete the conversion to their notation just add the additional conditional dependence on a.

Figure: