Reinforcement Learning-Based Non-Differentiable Optimization for Image Captioning

Mladen Nikolić

Machine Learning and Applications Group
Faculty of Mathematics
University of Belgrade

Overview

Overview

Image Captioning Basics

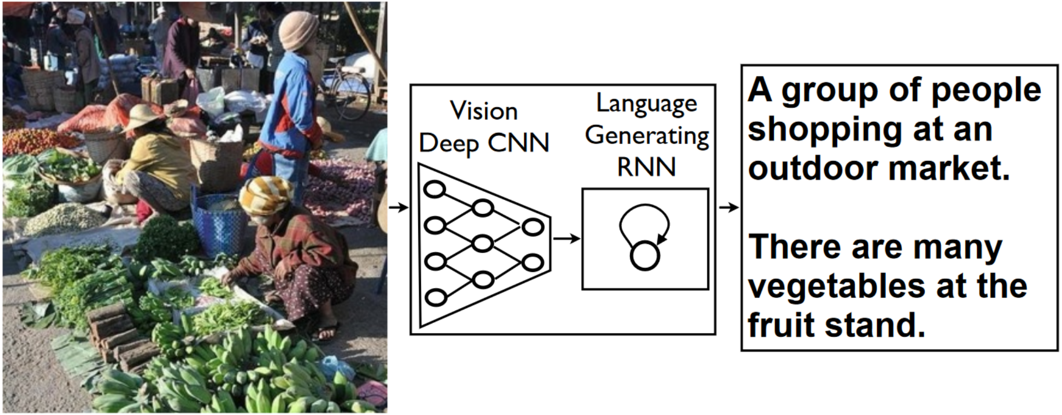
Policy Gradients

Policy Gradients for Image Captioning

Experimental Results

Image Captioning Problem

- For an image provide a caption describing it



COCO dataset

- Large-scale object detection, segmentation, and captioning dataset.
- $>$ 200K labeled images
- 5 captions per image

Maximal Likelihood Formulation

- Assume a probability distribution $p(\mathbf{y}|\mathbf{x},\theta)$ parametrized by learnable parameters θ
- Likelihood:

$$\prod_{i=1}^{N} p(\mathbf{y}^i|\mathbf{x}^i,\theta)$$

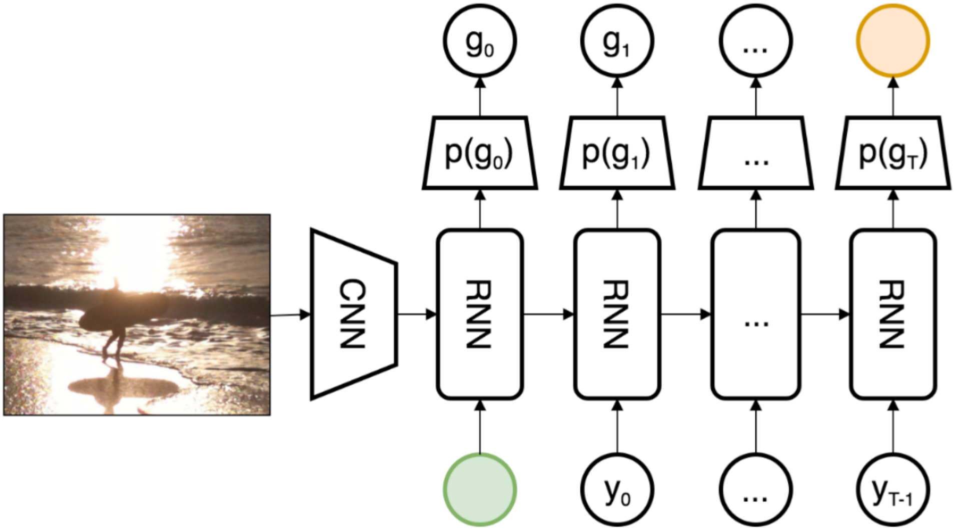
- Negative log likelihood loss:

$$\begin{aligned} L(\theta) &= -\sum_{i=1}^{N} \log p(\mathbf{y}^i|\mathbf{x}^i,\theta) \\ &= -\sum_{i=1}^{N}\sum_{j=1}^{N_i} \log p(y^i_j|y^i_{1:j-1},\mathbf{x}^i,\theta) \end{aligned}$$

Maximal Likelihood Formulation Issue

- In training, since the labels are known $\log p(y_j^i | y_{1:j-1}^i, \mathbf{x}^i, \theta)$ can be computed if $p(\cdot|\cdot, \theta)$ is defined
- While doing prediction, true labels are unavailable and model predictions are fed instead

Show and Tell Architecture



Maximal Likelihood Formulation Issue

- This leads to accumulation of errors
- There are methods to mitigate this problem, but with their own problems
- Also, MLE criterion need not correlate with human judgement too well
- Alternative path - optimize different metrics

Image Captioning Metrics

- BLEU
- METEOR
- ROUGE
- CIDEr
- SPICE

BLEU

- Assumes output sentence of length N and 1 or more reference sentences are provided.
- A word of the output sentence has a maximal number m of occurrences among all reference sentences and occurence count of n in the output sentence
- Its score is $\min(n, m)/N$
- Such scores are averaged over all sentences and all words in them to obtain BLEU score
- The score is between 0 and 1
- BLEU-N is a generalization to word N-grams

METEOR

- Performs alignment of output sentence with one of the reference sentences by matching words so that any word is matched to at most one word from another sentence
- Largest alignment is selected and if there are several, then the one with least matching crosses
- Precision and recall for the alignment are computed and F mean is computed as $10PR/(R+9P)$
- *Penalty* is computed as 0.5 times cubed number of chunks consisting of adjacent matched words divided by the number of matched words
- Score is computed as $F \cdot (1 - \mathit{Penalty})$
- The best score for all reference sentences is reported

ROUGE-N

- Compute precision and recall of N-grams in output sentence and in reference sentence and then compute F measure
- Larger N puts more emphasis on word order

CIDEr

- Compute TF-IDF weights for each N-gram appearing in all sentences related to all images.
- Represent each sentence by a vector of TF-IDF values of its N-grams
- CIDEr$_n$ for a candidate sentence and a set of reference sentences is an average of cosines between candidate sentence representation and representations of reference sentences
- CIDEr is an average of CIDEr$_n$ for $n = 1, \ldots, 4$

SPICE

- Correlation of all previous metrics to human judgement has been disputed
- Sentences are parsed and their parse trees are converted to scene graphs
- Graphs for output and reference sentences are compared
- Details are out of scope
- Drastically higher correlation with human judgement than previous metrics
- Great, but how to optimize such metrics??

Overview

The Goal

- Policy $\pi_\theta(a|s)$ is a parametrized distribution (can be a neural network) over actions, given a state
- Environment state transition probability is $p(s_{t+1}|s_t, a_t)$
- Probability of a trajectory τ under policy π_θ:

$$p_\theta(\tau) = p_\theta(s_0, a_0, \ldots, s_T, a_T) = p(s_0) \prod_{t=0}^{T} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t)$$

- Reward for trajectory τ:

$$r(\tau) = \sum_{i=0}^{T} r(s_t, a_t)$$

- The expected reward:

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau)]$$

- The goal is to maximize the expected reward with respect to θ

Gradient of The Expected Reward

- Gradient can't pass through expectation, but:

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau)] \\
&= \nabla_\theta \int r(\tau) p_\theta(\tau) d\tau \\
&= \int r(\tau) \nabla_\theta p_\theta(\tau) d\tau \\
&= \int r(\tau) \frac{\nabla_\theta p_\theta(\tau)}{p_\theta(\tau)} p_\theta(\tau) d\tau \\
&= \int r(\tau) [\nabla_\theta \log p_\theta(\tau)] p_\theta(\tau) d\tau \\
&= \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau) \nabla_\theta \log p_\theta(\tau)]
\end{aligned}
$$

Gradient of The Expected Reward

- Consider $\nabla_\theta \log p_\theta(\tau)$:

$$\nabla_\theta \left[\log p(s_0) + \sum_{t=0}^{T} (\log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)) \right] = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)$$

- Which yields:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[r(\tau) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

REINFORCE algorithm

- Basic form:
  1. Sample $\tau \sim p_\theta(\tau)$ and observe reward $r(\tau)$
  2. $\theta \leftarrow \theta + \alpha \left[r(\tau) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$
- Can be augmented by minibatches and stuff

Comparison with MLE

- Imagine a supervised scenario in which best actions were provided as labels

$$\nabla_\theta J_{PG}(\theta) \approx r(\tau) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \qquad \nabla_\theta J_{MLE}(\theta) \approx \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^\star|s_t)$$

- In supervised regime we would fit the policy directly to the best actions by exploiting gradients for those actions
- In RL scenario we are weighting gradients for sampled actions by reward those actions yielded
- This is a weaker supervision and the variance of stochastic approximation is big, so expect slow convergence
- However, the reward can be an arbitrary function which allows optimization of wider range of functions!

Expected Grad-Log-Prob Lemma

$$
\begin{aligned}
\mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \log p_\theta(\tau) \right] &= \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) d\tau \\
&= \int p_\theta(\tau) \frac{\nabla_\theta p_\theta(\tau)}{p_\theta(\tau)} d\tau \\
&= \int \nabla_\theta p_\theta(\tau) d\tau \\
&= \nabla_\theta \int p_\theta(\tau) d\tau \\
&= \nabla_\theta 1 \\
&= 0
\end{aligned}
$$

Variance Reduction Through Baselines

- It holds:

$$\begin{aligned}
\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[(r(\tau)-b)\nabla_\theta \log p_\theta(\tau)\right] &= \nabla_\theta J(\theta) - \mathbb{E}_{\tau\sim p_\theta(\tau)}\left[b\nabla_\theta \log p_\theta(\tau)\right] \\
&= \nabla_\theta J(\theta) - b\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\nabla_\theta \log p_\theta(\tau)\right] \\
&= \nabla_\theta J(\theta)
\end{aligned}$$

- Subtracting any constant from the reward changes nothing. So, why do it?
- It can reduce the variance of the stochastic approximation if properly chosen!
- Variance minimization with respect to baseline can be either analytical or numerical
- This conclusion can be easily generalized to nonconstant baselines as long as they do not depend on actions

Alternative Gradient Estimate

- It holds:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau)] \\
&= \nabla_\theta V_\theta(s_0) \\
&= \nabla_\theta \sum_\tau \pi_\theta(\tau|s_0) R(\tau) \\
&= \sum_\tau \nabla_\theta \pi_\theta(\tau|s_0) R(\tau)
\end{aligned}$$

- Consider the derivative of the policy

$$\begin{aligned}
\nabla_\theta \pi_\theta(\tau|s_0) &= \nabla_\theta \prod_{t=0}^{T} \pi_\theta(a_t|s_t) \\
&= \sum_{t=0}^{T} \pi_\theta(\tau_{0:t-1}|s_0) \nabla_\theta \pi_\theta(a_t|s_t) \pi_\theta(\tau_{t+1:T}|s_t + 1)
\end{aligned}$$

Alternative Gradient Estimate

- By substituting and changing the order of summation:

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{t=0}^{T} \sum_{\tau_{0:t-1}} \pi_\theta(\tau_{0:t-1}|s_0) \sum_{a_t} \nabla_\theta \pi_\theta(a_t|s_t) \sum_{\tau_{t+1:T}} \pi_\theta(\tau_{t+1:T}|s_{t+1}) R(\tau) \\
&= \sum_{t=0}^{T} \sum_{\tau_{0:t-1}} \pi_\theta(\tau_{0:t-1}|s_0) \sum_{a_t} \pi_\theta(a_t|s_t) \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \sum_{\tau_{t+1:T}} \pi_\theta(\tau_{t+1:T}|s_{t+1}) R(\tau) \\
&= \sum_{t=0}^{T} \sum_{\tau_{0:t-1}} \pi_\theta(\tau_{0:t-1}|s_0) \sum_{a_t} \pi_\theta(a_t|s_t) \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{\tau_{t+1:T}} \pi_\theta(\tau_{t+1:T}|s_{t+1}) R(\tau) \\
&= \sum_{t=0}^{T} \mathbb{E}_{\tau_{0:t-1}}[\mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t)) \mathbb{E}_{\tau_{t+1:T}}[R(\tau)]]] \\
&= \sum_{t=0}^{T} \mathbb{E}_{\tau_{0:t-1}}[\mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t)) \mathbb{E}_{\tau_{t+1:T}}[R(\tau_{0:t-1}) + R(\tau_{t:T})]]]
\end{aligned}
$$

Alternative Gradient Estimate

- Since $R(\tau_{0:t-1})$ is independent of a_t and $\tau_{t+1:T}$ it holds:

$$\begin{aligned}\mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t))\mathbb{E}_{\tau_{t+1:T}}[R(\tau_{0:t-1})]] &= R(\tau_{0:t-1})\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t|s_t)] \\ &= 0\end{aligned}$$

- where the last equation is due to the expected grad-log-prob lemma

Alternative Gradient Estimate

- Also consider:

$$\mathbb{E}_{\tau_{t+1:T}}[R(\tau_{t:T})] = r_t + \mathbb{E}_{\tau_{t+1:T}}[R(\tau_{t+1:T})] = Q_\theta(s_t, a_t)$$

- Therefore

$$\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{t=0}^{T} \mathbb{E}_{\tau_{0:t-1}}[\mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t))\mathbb{E}_{\tau_{t+1:T}}[R(\tau_{0:t-1}) + R(\tau_{t:T})]]] \\
&= \sum_{t=0}^{T} \mathbb{E}_{\tau_{0:t-1}}[\mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t))Q(s_t, a_t)]] \\
&= \mathbb{E}_\tau\left[\sum_{t=0}^{T} \mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t))Q(s_t, a_t)]\right] \\
&= \mathbb{E}_\tau\left[\sum_{t=0}^{T} \sum_{a_t} \nabla_\theta \pi_\theta(a_t|s_t) Q(s_t, a_t)\right]
\end{aligned}$$

What's the difference

- Compare stochastic approximations

$$\underbrace{\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{t=0}^{T} r(s_t, a_t)}_{\text{PG}} \qquad \underbrace{\sum_{t=0}^{T} \mathbb{E}_{a_t}[(\nabla_\theta \log \pi_\theta(a_t|s_t))Q(s_t, a_t)]}_{\text{alternative PG}}$$

- Second estimate weights gradients only by future rewards, which is known to reduce the variance of stochastic estimate (this can be applied to the standard algorithm, too)
- In the second estimate estimation is taken over a_t instead of using an action from a single trajectory, which reduces the variance
- Q includes an expectation instead of reward along a single trajectory, with the same effect
- Second estimate is more expensive

Overview

Why Use RL?

- Image captioning is a supervised task, so RL is not a natural approach to it
- Still, MLE has its issues and other metrics are non-differentiable
- REINFORCE can be used in such scenarios even for supervised learning if the task is properly posed as a RL problem

Image Captioning as a RL Problem

- Agent: caption generator
- Episode: caption generation
- State: caption generated so far
- Action: a word to add to the caption
- Reward: value of the metric at the end of the episode and 0 for other steps

MIXER

- Vanilla REINFORCE approach does not work – too much exploration is needed and there is too much variance
- MIXER is an approach which first applied policy gradients to image captioning
- It mixes MLE and REINFORCE objectives
- If T is the length of the sequence, it minimizes MLE loss for first t words and maximizes BLEU-4 loss for the rest of the sequence
- BLEU-4 part relies on REINFORCE for optimization
- It starts with $t = T$ and decreases it to 0 according to some carefully selected schedule
- The approach is not robust and needs careful tuning, so it is not easy to use

Proposed Approach

- Policy: $\pi_\theta(w_t | w_{1:t-1}, \mathbf{x})$
- Stochastic approximation of the gradient:

$$\nabla_\theta J(\theta) \approx \sum_{t=1}^{T} \sum_{w_t} \nabla_\theta \pi_\theta(w_t | w_{1:t-1}) Q(w_{1:t-1}, w_t)$$

- Reward: $R(w_{1:T} | \mathbf{x}, \mathbf{y})$ given at the end
- Reward is sparse, so Monte Carlo rollouts are used for intermediate rewards:

$$Q_\theta(w_{1:t-1}, w_t) \approx \frac{1}{K} \sum_{k=1}^{K} R(w_{0:t-1}; w_t; w_{t+1:T}^k | \mathbf{x}, \mathbf{y})$$

Variance Reduction

- Variance reduction:

$$\nabla_\theta J(\theta) \approx \sum_{t=1}^{T} \sum_{w_t} \nabla_\theta \pi_\theta(w_t | w_{1:t-1})(Q(w_{1:t-1}, w_t) - B_\phi(w_{1:t-1}))$$

- Baseline is a neural network trained to minimize the loss

$$L(\phi) = \sum_t \mathbb{E}_{s_t} \mathbb{E}_{w_t} (Q(s_t, w_t) - B_\phi(s_t))^2$$

where for s_t the hidden state of the generator is used, but gradients from L are not propagated to the generator

Rewards

- BCMR:

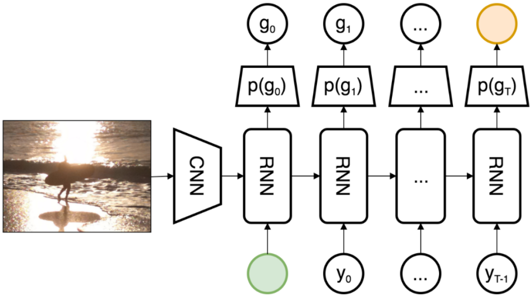
  $5.0BLEU-1+0.5BLEU-2+1.0BLEU-3+1.0BLEU-4+1.0CIDEr+5.0METEOR+2.0ROUGE$

- SPICE
- Combination of SPICE and CIDEr

Training

- Actions are words, which makes a huge action space
- First the generator is trained using MLE to help warm start
- Then it is trained by policy gradients

Architecture



- 512-dimensional word embeddings
- Inception-V3 CNN encoder pretrained on ImageNet
- RNN decoder is one-layer LSTM with 512 units
- In test time RNN decoder gets its previous output as its input

Overview

Image Captioning Basics

Policy Gradients

Policy Gradients for Image Captioning

Experimental Results

Experimental Setup

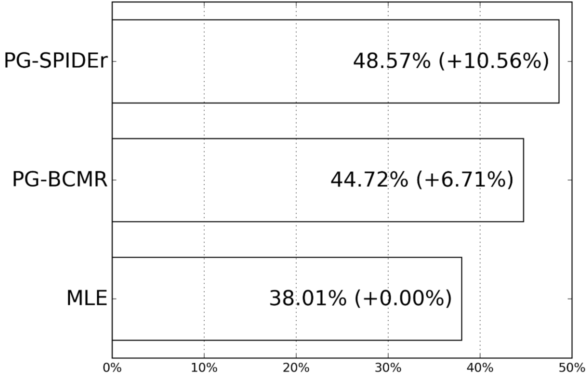
- COCO dataset
- $120,553$ training and $1,665$ validation images
- At least 5 captions per image
- Vocabulary size of $8,855$ words

Experimental Results

Submissions	CIDEr-D	Meteor	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
MSM@MSRA [28]	0.984	0.256	0.542	0.739	0.575	0.436	0.330
Review Net [27]	0.965	0.256	0.533	0.720	0.550	0.414	0.313
ATT [29]	0.943	0.250	0.535	0.731	0.565	0.424	0.316
Google [22]	0.943	0.254	0.530	0.713	0.542	0.407	0.309
Berkeley LRCN [7]	0.921	0.247	0.528	0.718	0.548	0.409	0.306
MLE	0.947	0.251	0.531	0.724	0.552	0.405	0.294
PG-BLEU-4	0.966	0.249	0.550	0.737	0.587	**0.455**	**0.346**
PG-CIDEr	0.995	0.249	0.548	0.737	0.581	0.442	0.333
MIXER-BCMR	0.924	0.245	0.532	0.729	0.559	0.415	0.306
MIXER-BCMR-A	0.991	**0.258**	0.545	0.747	0.579	0.431	0.317
PG-BCMR	**1.013**	**0.257**	**0.55**	**0.754**	**0.591**	0.445	0.332
PG-SPIDEr	1.000	0.251	0.544	0.743	0.578	0.433	0.322

Human Evaluation

- Evaluation at crowdsourcing platform
- 87% ground truth captions evaluated as not bad



Conclusions

- We can perform gradient based optimization of non-differentiable losses via policy gradient algorithms
- We repay that in convergence rate and stability of optimization process, which decrease
- There is a variety of tricks to improve performance, but it is not easy

References

- S. Liu, Z. Zhu, N. Ye, S. Guadarrama, K. Murphy, Improved Image Captioning via Policy Gradient Optimization of SPIDEr
- M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence Level Training with Recurrent Neural Networks

THANKS!