Механизмы внимания для генерации описания изображения

Практикум на ЭВМ 2017/2018

Филимонов Владислав Аскольдович, студент 317 группы

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

10 мая 2018 г.

Постановка задачи генерации описания изображения

Для любой картинки нужно сопоставить описание y - последовательность one-hot-encoded слов из словаря.

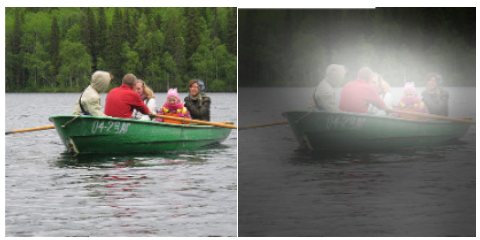
$$y = \{y_1, \ldots, y_C\},\ y_i \in R^K,\ K = \|Dictionary\|$$

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.

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

Решение с архитектурой encoder-decoder

Решение данной задачи может иметь следующую архитектуру:

1. Encoder для поиска "хорошего представления данных" (например, первые слои CNN).
2.
    1. Механизм внимания для поиска "важных" данных.
    2. Decoder, переводящий последовательность данных, выбранных механизмом внимания, в последовательность слов (например, LSTM).

Далее будет рассмотрено решение, описанное в [1]

Encoder

В качестве кодировщика была использована CNN, которая выделила L annotation векторов размерности D,

$$a = \{a_1, \ldots, a_L\}, \ a_i \in R^D$$

так, что каждый вектор соответствовал некоторой области исходной картинки. Например, если после сверточных слоев данные имели размер $14 \times 14 \times 512$, тогда $L = 196$, $D = 512$.

Decoder

В качестве декодера использовалась LSTM сеть, которая генерировала описание по слову за один шаг, основываясь на context векторе ($\hat{z}_t \in R^D$ сгенерировано с помощью механизма внимания), предыдущем hidden state ($h_{t-1} \in R^n$) и предыдущем сгенерированном слове ($Ey_{t-1},\ y \in R^K,\ E \in R^{m \times K}$).

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} T_{D+m+n,4n} \begin{pmatrix} Ey_{t-1} \\ h_{t-1} \\ \hat{z}_t \end{pmatrix}.$$

$T_{s,t} : R^s \to R^t$ - аффинное преобразование с обучаемыми параметрами.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t.$$

$$h_t = o_t \odot \tanh(c_t).$$

Decoder

Вычисление вероятности выходного слова при заданном LSTM state(h_t), context векторе и предыдущем слове:

$$p(y_t|a, y^{t-1}) \propto \exp(L_o(Ey_{t-1} + L_h h_t + L_z \hat{z}_t)),$$

где $L_o \in R^{K \times m}$, $L_h \in R^{m \times n}$, $L_z \in^{m \times D}$, $E \in R^{m \times K}$ - обучаемые параметры.

Механизмы внимания

Context вектор $\hat{z}_t$ по сути - представление релевантной части изображения. Механизм внимания ϕ генерирует $\hat{z}_t$, основываясь на множестве всех annotation векторов и их весов, вычисленных на основе некоторой функции f_{att} (в данном решении - f_{att} - MLP):

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

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

$$\hat{z}_t = \phi(\{a_i\}, \{\alpha_{ti}\})$$

Далее будет рассмотрено два механизма внимания:

- Stohastic “Hard” Attention
- Deterministic “Soft” Attention

Stohastic "Hard" Attention

Рассмотрим бинарный вектор $s_t \in \{0,1\}^L$ (location variable), который определяет, где нужно сконцентрировать внимание при генерации t-ого слова. Введем категориальное распределение (multinoulli distribution):

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

И рассмотрим context вектор:

$$\hat{z}_t = \sum_i s_{t,i} a_i.$$

Введем целевую функцию L_s, которая является нижней оценкой логорифма правдоподобия:

$$L_s = \sum_s p(s \mid a) \log p(y \mid s, a) \leq \log \sum_s p(s \mid a) p(y \mid s, a) = \log p(y \mid a)$$

Stohastic "Hard" Attention

Для обучения градиентными методами рассмотрим производную:

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

Так как, $\tilde{s_t} \sim \text{Multinoulli}_L(\{\alpha_{ti}\})$, то $\frac{\partial L_s}{\partial W}$ можно аппроксимировать методом Монте-Карло:

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

Существуют методы для снижения разброса этой оценки, описанные в [1]

Deterministic “Soft” Attention

В качестве context вектора в этом подходе используется матожидание context вектора из подхода Stohastic “Hard” Attention

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

При таком выборе context вектора вся модель - дифференцируемая и обучение происходит с помощью backpropagation.

Пример для сравнения двух подходов

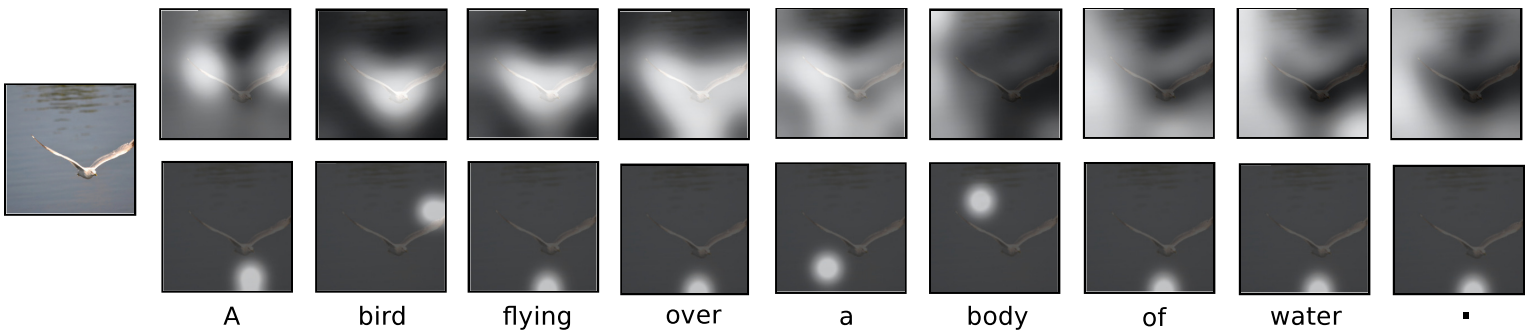


Рис.: Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

Примеры ошибок

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.

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

Список литературы

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv:1502.03044 ▸ link-to-article