Advanced Computer Vision

Практический курс

Савельева Юлия Олеговна, 3 семестр, 15.10.2020

Text Recognition

Градиент целевой функции

Реализовать вычисление градиента целевой функции ->

absum[t] = $$\sum_{s=1}^{|\mathbf{l}'|} \frac{\alpha_t(s)\beta_t(s)}{y^t_{\mathbf{l}'_s}}$$

grad[t] = $$\sum_{s \in lab(\mathbf{z},k)} \hat{\alpha}_t(s)\hat{\beta}_t(s)$$

out[t][i] = $$y^t_k$$

Algorithm 2: CTC Loss gradient computation

Data: $out_{m \times n}$ (result of softmax), where m=$\bar{W}/4$, n=$|\hat{A}|$,
l (label encoded by alphabet),
bl=0 (blank index),
a, b (alpha and beta from paper)

begin

- $L = 2 \times len(l) + 1$
- $T = m$
- $grad = zeros(T, n)$
- $ab = a * b$
- **for** $s := 1$ **to** L **do**
  - **if** s ***mod*** $2 = 0$ **then**
    - **for** $t := 1$ **to** T **do**
      - $grad^{bl}_t += ab^s_t$
      - $ab^s_t = \frac{ab^s_t}{out^{bl}_t}$
  - **else**
    - **for** $t := 1$ **to** T **do**
      - $i = \frac{s-1}{2}$
      - $grad^i_t += ab^s_t$
      - $ab^s_t = \frac{ab^s_t}{out^i_t}$
- $absum = zeros(T)$
- **for** $t := 1$ **to** T **do**
  - $absum_t = \sum_{s=1}^{L} ab^s_t$
- **for** $t := 1$ **to** T **do**
  - **for** $i := 1$ **to** n **do**
    - $grad^i_t = out^i_t - \frac{grad^i_t}{out^i_t * absum_t}$
- **return** grad

Text Recognition

Декодирование последовательности

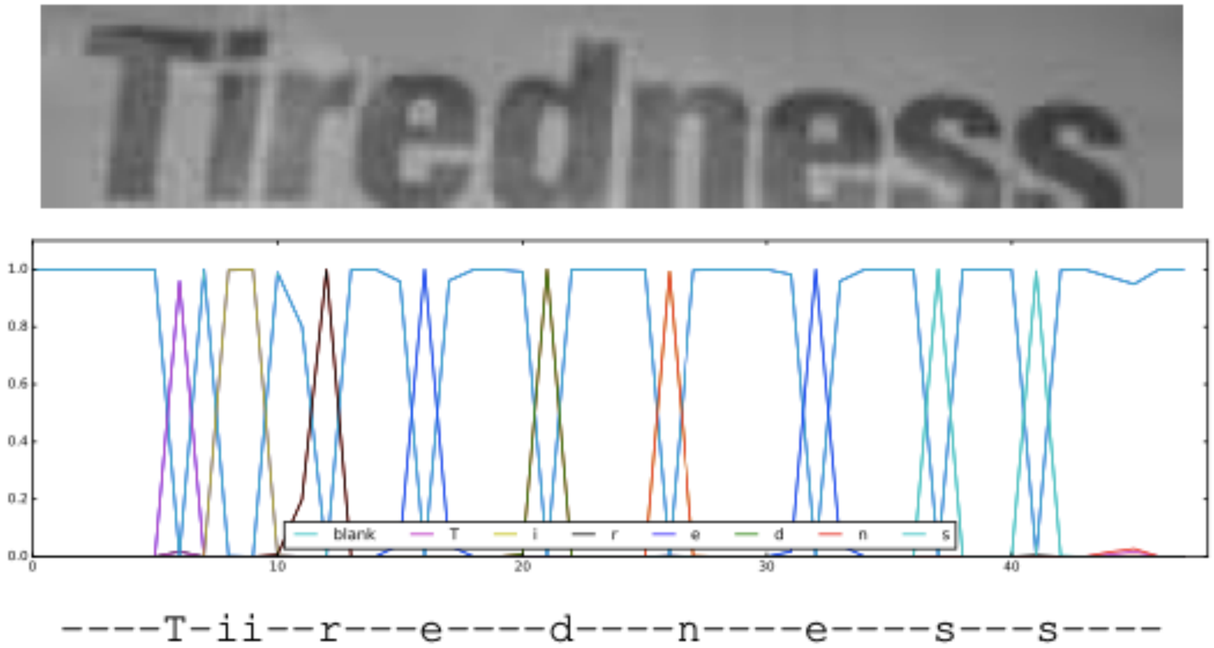
Реализовать декодирование:

1) На каждом отсчете t брать argmax(out[t]) и получить, например:

- - -MM- -A-ШШШ- -AA- -

2) Убрать из последовательности повторяющиеся буквы

3) Убрать пробелы



Deep TextSpotter: An End-to-End Trainable Scene Text Localization and Recognition Framework

Text Recognition

Подсчет точности

Реализовать:

1) Чтобы сравнить gt слово и decoded слово, используем расстояние Левенштейна: https://tirinox.ru/levenstein-python/

2) Полученное целое число делим на dist = max(len(gt), len(decoded)). Получаем 1.0, если последовательности совсем непохожи, и 0.0, если идентичны

3) Для каждой пары (gt, decoded) валидационной выборки считаем acc = 1 - dist

4) Все значения acc, полученные на 3 шаге, суммируем и делим на количество пар

P.S. Код подсчета расстояния Левенштейна можно взять из интернета :)

Дедлайн 05.11.2020 00:00