

Recurrent Mapping Model (7)

RMM 모델 구현 및 실험

Geonil Yun,

rjsdlf45@gmail.com

2021. 12. 28.



Overview

- **Hypernym Discovery**
 - Preprocessing
 - RMM sample output
 - List-wise Loss function output
- **Scene Text Recognition**
 - Tensorflow Lite
 - PGNet – A, E

Hypernym Discovery

- Preprocessing

- Implementation

- Use original corpus of UMBC, training with **window size 10 (100GB train pairs)**

“ The DeLorme PN-20 represents a new breed of GPS device. Do you... ”

↓ sentence tokenizing

“ The DeLorme PN-20 represents a new breed of GPS devices. ”
“Do you know your family members?”
...

↓ word tokenizing

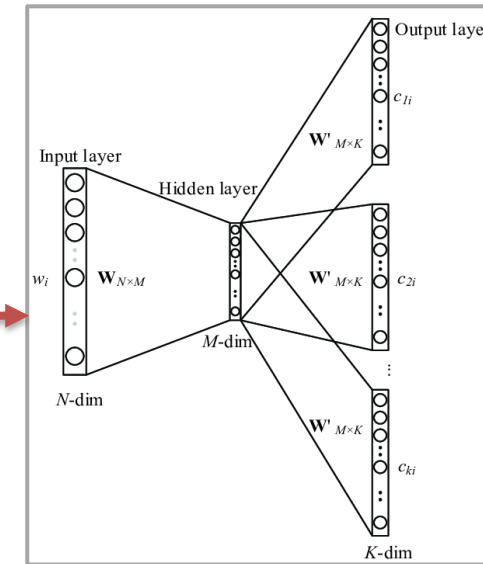
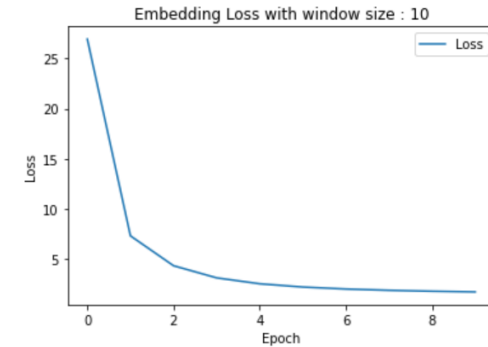
“ The”, “DeLorme” “PN-20” “represents”, ...



vocabulary

⊕ combination words (single word to trigram)
convert to lower case

training

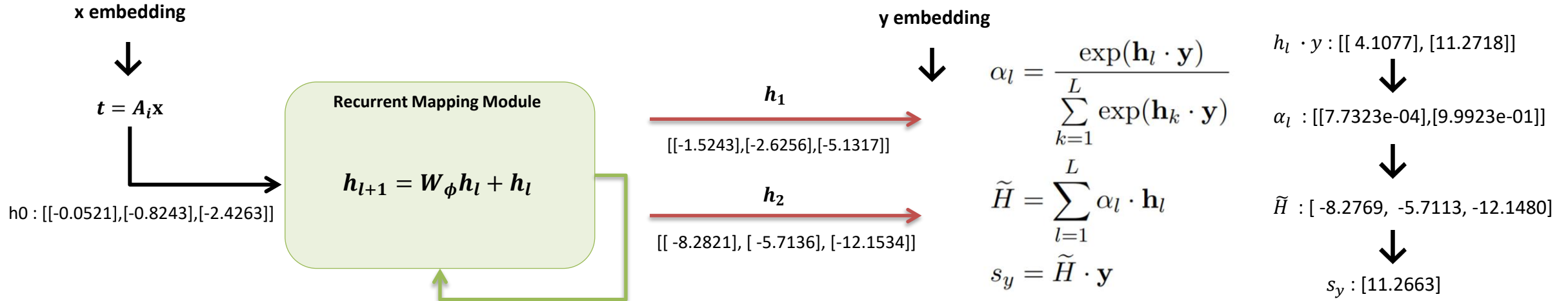


Skip gram model

Hypernym Discovery

- Recurrent Mapping Model

$x : [[1.5262], [-0.5106], [1.3392]]$
 $y : [[-0.3219], [0.0785], [-0.7450]]$
 $A : [[1.2791, -0.1770, -1.5641], [0.2235, -0.5736, -1.0889], [-0.2071, 1.0943, -1.1585]]$
 $W : [[2.6849, -0.1727, 0.6077], [-0.4655, -0.0721, 0.7769], [1.2052, -0.4586, 1.2449]]$

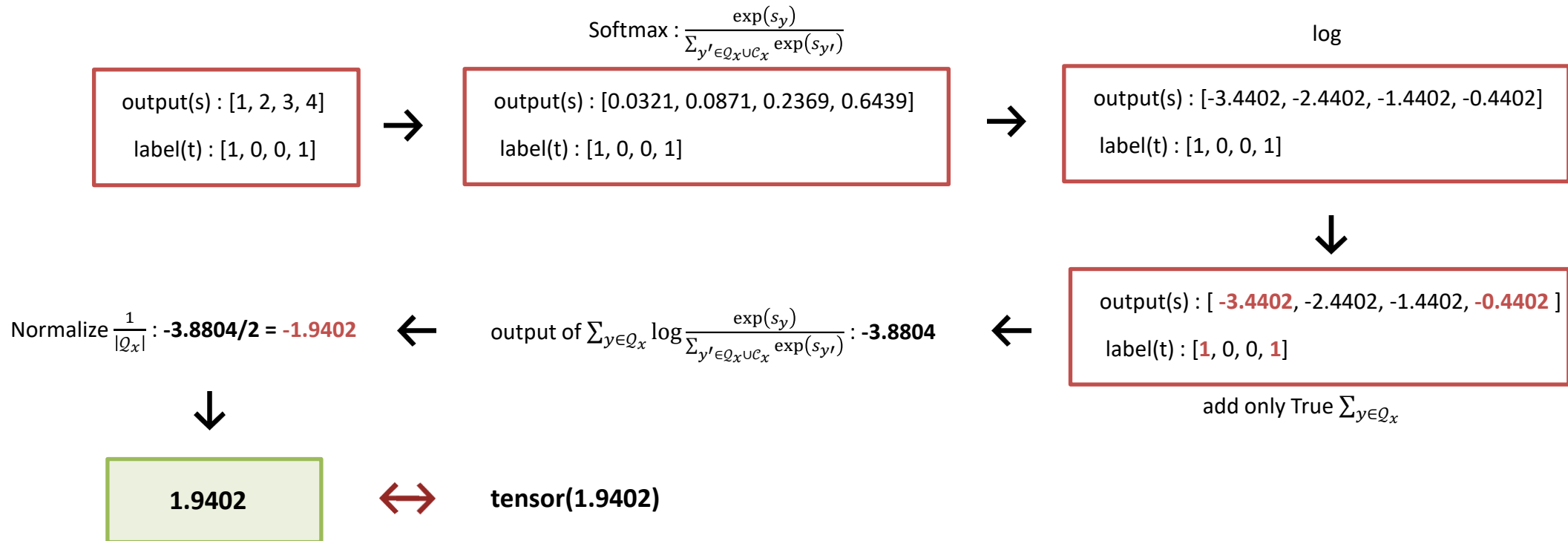


$x \in \mathbb{R}^{d \times 1}$ query embedding
 $y \in \mathbb{R}^{d \times 1}$ candidate embedding
 $A_0 \in \mathbb{R}^{d \times d}$ for concept query
 $A_1 \in \mathbb{R}^{d \times d}$ for entity query
 $h_l \in \mathbb{R}^{d \times 1}$ for projection vector

Hypernym Discovery

- List-wise Loss Function (custom implementation)

$$\mathcal{L} = - \sum_{x \in \mathcal{X}_{train}} \frac{1}{|Q_x|} \sum_{y \in Q_x} \log \frac{\exp(s_y)}{\sum_{y' \in Q_x \cup \mathcal{C}_x} \exp(s_{y'})}$$

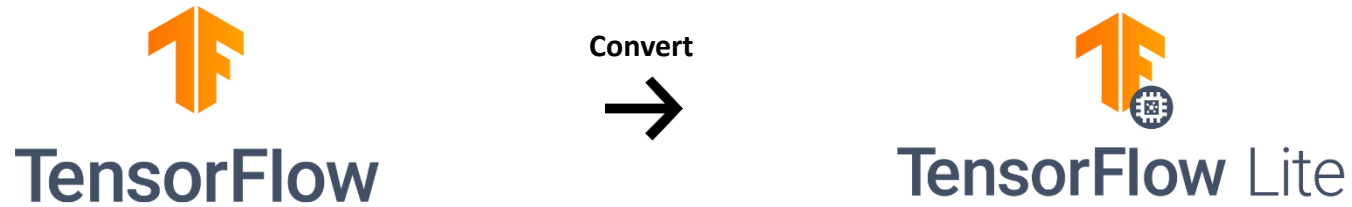


Future

- **Future works**
 - Preprocessing with POS tagged corpus
 - Different dataset and model

Scene Text Recognition

- TensorFlow Lite
 - Convert TensorFlow model to TensorFlow Lite model



```
import tensorflow as tf

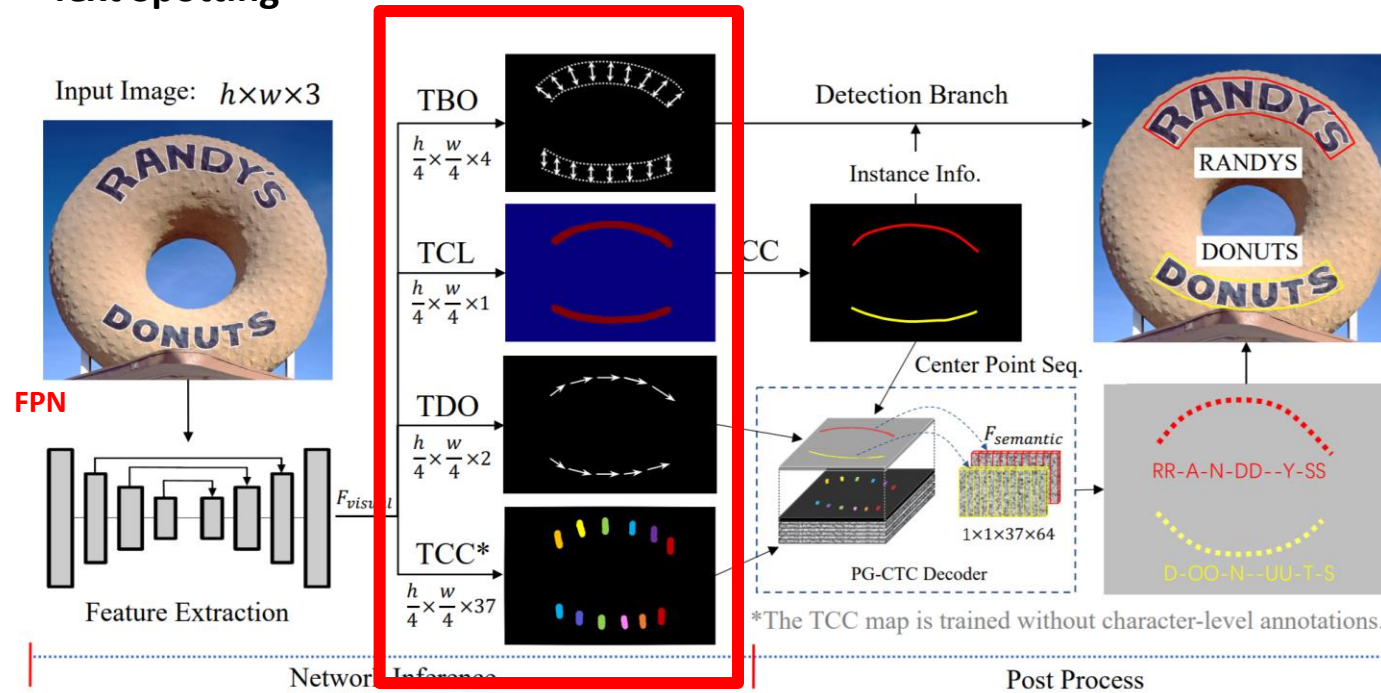
converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)
tflite_model = converter.convert()
```

- Training model on TensorFlow framework.
- Convert model to TensorFlow Lite with simple code

Scene Text Recognition

- PGNet-E

- Text Spotting



- TBO, TCL, TDO, TCC
 - Use output of FPN features

Thank you