pytorch-partial-crf

Apr 28, 2021

Contents:

1 Installation **1**

2 Getting started **3**

2.1 Computing log likelihood 3

2.2 Decoding 3

CHAPTER 1

Installation

Install with pip:

```
pip install pytorch_partial_crf
```

CHAPTER 2

Getting started

This package provides an implementation of a Partial/Fuzzy CRF layer for learning incompleted tag sequences, and a linear-chain CRF layer for learning tag sequences.

```
import torch
from pytorch_partial_crf import PartialCRF
num_tags = 6  # number of tags is 6
model = PartialCRF(num_tags)
```

2.1 Computing log likelihood

```
batch_size = 3
sequence_length = 5
emissions = torch.randn(batch_size, sequence_length, num_tags)
# Set to -1 if it is unknown tag
tags = torch.LongTensor([
      [1, 2, 3, 3, 5],
      [-1, 3, -1, 2, 1],
      [1, 0, -1, 4, -1],
])  # (seq_length, batch_size)
model(emissions, tags) # Computing log likelihood
```

2.2 Decoding

Viterbi decode

```
model.viterbi_decode(emissions)
```

Restricted viterbi decode

```
possible_tags = torch.randn(batch_size, sequence_length, num_tags)
possible_tags[possible_tags <= 0] = 0 # `0` express that can not pass.
possible_tags[possible_tags > 0] = 1 # `1` express that can pass.
possible_tags = possible_tags.byte()
model.restricted_viterbi_decode(emissions, possible_tags)
```

Marginal probabilities

```
model.marginal_probabilities(emissions)
```