

---

# **tensorflow\_hmm Documentation**

***Release 0.1.1***

**Zach Dwiell**

**Jan 23, 2018**



---

## Contents

---

<b>1</b>	<b>tensorflow_hmm package</b>	<b>3</b>
1.1	Submodules . . . . .	3
1.2	tensorflow_hmm.hmm module . . . . .	3
1.3	Module contents . . . . .	4
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



Contents:



## 1.1 Submodules

### 1.2 tensorflow\_hmm.hmm module

**class** tensorflow\_hmm.hmm.**HMM** (*P*, *p0=None*, *length=None*)

Bases: object

A class for Hidden Markov Models.

The model attributes are: - *K* :: the number of states - *P* :: the *K* by *K* transition matrix (from state *i* to state *j*, (*i*, *j*) in [1..*K*])

- *p0* :: the initial distribution (defaults to starting in state 0)

**class** tensorflow\_hmm.hmm.**HMMNumpy** (*P*, *p0=None*, *length=None*)

Bases: tensorflow\_hmm.hmm.HMM

**forward\_backward** (*y*)

**viterbi\_decode** (*y*)

**viterbi\_decode\_batched** (*y*)

Expects inputs in [B, N, K] layout

**class** tensorflow\_hmm.hmm.**HMMTensorflow** (*P*, *p0=None*, *length=None*)

Bases: tensorflow\_hmm.hmm.HMM

**forward\_backward** (*y*)

runs forward backward algorithm on state probabilities *y*

*y* [np.array][shape (T, K) where T is number of timesteps and] K is the number of states

(posterior, forward, backward) posterior : list of length T of tensorflow graph nodes representing the posterior probability of each state at each time step

**forward** [list of length T of tensorflow graph nodes representing] the forward probability of each state at each time step

**backward** [list of length T of tensorflow graph nodes representing] the backward probability of each state at each time step

**viterbi\_decode** (y)

Runs viterbi decode on state probabilities y.

**y** [np.array][shape (T, K) where T is number of timesteps and] K is the number of states

(s, pathScores) s : list of length T of tensorflow ints : represents the most likely state at each time step.

**pathScores** [list of length T of tensorflow tensor of length K] each value at (t, k) is the log likelihood score in state k at time t. sum(pathScores[t, :]) will not necessary == 1

**viterbi\_decode\_batched** (y, onehot=False)

Runs viterbi decode on state probabilities y in batch mode

**y** [np.array][shape (B, T, K) where T is number of timesteps and] K is the number of states

**onehot** [boolean][if true, returns a onehot representation of the] most likely states, instead of integer indexes of the most likely states.

(s, pathScores) s : list of length T of tensorflow ints : represents the most likely state at each time step.

**pathScores** [list of length T of tensorflow tensor of length K] each value at (t, k) is the log likelihood score in state k at time t. sum(pathScores[t, :]) will not necessary == 1

tensorflow\_hmm.hmm.**tf\_map** (fn, arrays)

Apply fn to each of the values in each of the arrays. Implemented in native python would look like:

```
return map(fn, *arrays)
```

more explicitly:

```
output[i] = fn(arrays[0][i], arrays[1][i], ... arrays[-1][i])
```

This function assumes that all arrays have same leading dim.

## 1.3 Module contents



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### t

`tensorflow_hmm`, 4

`tensorflow_hmm.hmm`, 3



## F

`forward_backward()` (tensorflow\_hmm.hmm.HMMNumpy method), 3  
`forward_backward()` (tensorflow\_hmm.hmm.HMMTensorflow method), 3

## H

HMM (class in tensorflow\_hmm.hmm), 3  
HMMNumpy (class in tensorflow\_hmm.hmm), 3  
HMMTensorflow (class in tensorflow\_hmm.hmm), 3

## T

tensorflow\_hmm (module), 4  
tensorflow\_hmm.hmm (module), 3  
tf\_map() (in module tensorflow\_hmm.hmm), 4

## V

`viterbi_decode()` (tensorflow\_hmm.hmm.HMMNumpy method), 3  
`viterbi_decode()` (tensorflow\_hmm.hmm.HMMTensorflow method), 4  
`viterbi_decode_batched()` (tensorflow\_hmm.hmm.HMMNumpy method), 3  
`viterbi_decode_batched()` (tensorflow\_hmm.hmm.HMMTensorflow method), 4