tf.nn.ctc_greedy_decoder

```
ctc_greedy_decoder(
    inputs,
    sequence_length,
    merge_repeated=True
)
```

Defined in `tensorflow/python/ops/ctc_ops.py`.

See the guide: Neural Network > Connectionist Temporal Classification (CTC)

Performs greedy decoding on the logits given in input (best path).

> **Note:** Regardless of the value of merge_repeated, if the maximum index of a given time and batch corresponds to the blank index `(num_classes - 1)`, no new element is emitted.

If `merge_repeated` is `True`, merge repeated classes in output. This means that if consecutive logits' maximum indices are the same, only the first of these is emitted. The sequence `A B B * B * B` (where '*' is the blank label) becomes

- `A B B B` if `merge_repeated=True`.
- `A B B B B` if `merge_repeated=False`.

Args:

- `inputs`: 3-D `float` `Tensor` sized `[max_time x batch_size x num_classes]`. The logits.
- `sequence_length`: 1-D `int32` vector containing sequence lengths, having size `[batch_size]`.
- `merge_repeated`: Boolean. Default: True.

Returns:

A tuple `(decoded, neg_sum_logits)` where `decoded`: *A single-element list.* `decoded[0]` *is an* `SparseTensor` *containing the decoded outputs s.t.:*
`decoded.indices`: *Indices matrix* `(total_decoded_outputs x 2)`. *The rows store:* `[batch, time]`.
`decoded.values`: *Values vector, size* `(total_decoded_outputs)`. *The vector stores the decoded classes.*
`decoded.shape`: *Shape vector, size* `(2)`. *The shape values are:* `[batch_size, max_decoded_length]` `neg_sum_logits`: A `float` matrix `(batch_size x 1)` containing, for the sequence found, the negative of the sum of the greatest logit at each timeframe.

Last updated November 2, 2017.

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**