

## tf.contrib.signal.mfccs\_from\_log\_mel\_spectrograms

```
mfccs_from_log_mel_spectrograms(  
    log_mel_spectrograms,  
    name=None  
)
```

Defined in [tensorflow/contrib/signal/python/ops/mfcc\\_ops.py](#).

See the guide: [Signal Processing \(contrib\) > Computing Mel-Frequency Cepstral Coefficients \(MFCCs\)](#)

Computes [MFCCs](#) of `log_mel_spectrograms`.

Implemented with GPU-compatible ops and supports gradients.

[Mel-Frequency Cepstral Coefficient \(MFCC\)](#) calculation consists of taking the DCT-II of a log-magnitude mel-scale spectrogram. [HTK's](#) MFCCs use a particular scaling of the DCT-II which is almost orthogonal normalization. We follow this convention.

All `num_mel_bins` MFCCs are returned and it is up to the caller to select a subset of the MFCCs based on their application. For example, it is typical to only use the first few for speech recognition, as this results in an approximately pitch-invariant representation of the signal.

For example:

```
sample_rate = 16000.0  
# A Tensor of [batch_size, num_samples] mono PCM samples in the range [-1, 1].  
pcm = tf.placeholder(tf.float32, [None, None])  
  
# A 1024-point STFT with frames of 64 ms and 75% overlap.  
stfts = tf.contrib.signal.stft(pcm, frame_length=1024, frame_step=256,  
                               fft_length=1024)  
spectrograms = tf.abs(stft)  
  
# Warp the linear scale spectrograms into the mel-scale.  
num_spectrogram_bins = stfts.shape[-1].value  
lower_edge_hertz, upper_edge_hertz, num_mel_bins = 80.0, 7600.0, 80  
linear_to_mel_weight_matrix = tf.contrib.signal.linear_to_mel_weight_matrix(  
    num_mel_bins, num_spectrogram_bins, sample_rate, lower_edge_hertz,  
    upper_edge_hertz)  
mel_spectrograms = tf.tensordot(  
    spectrograms, linear_to_mel_weight_matrix, 1)  
mel_spectrograms.set_shape(spectrograms.shape[:-1].concatenate(  
    linear_to_mel_weight_matrix.shape[-1:]))  
  
# Compute a stabilized log to get log-magnitude mel-scale spectrograms.  
log_mel_spectrograms = tf.log(mel_spectrograms + 1e-6)  
  
# Compute MFCCs from log_mel_spectrograms and take the first 13.  
mfccs = tf.contrib.signal.mfccs_from_log_mel_spectrograms(  
    log_mel_spectrograms)[..., :13]
```

Args:

- `log_mel_spectrograms`: A `[..., num_mel_bins] float32 Tensor` of log-magnitude mel-scale spectrograms.
- `name`: An optional name for the operation.

Returns:

A `[..., num_mel_bins] float32 Tensor` of the MFCCs of `log_mel_spectrograms`.

Raises:

- `ValueError`: If `num_mel_bins` is not positive.

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

## Stay Connected

Blog

GitHub

Twitter

## Support

Issue Tracker

Release Notes

Stack Overflow

English

[Terms](#) | [Privacy](#)