TensorFlow

TensorFlow API r1.4

tf.contrib.signal.stft

```
stft(
    signals,
    frame_length,
    frame_step,
    fft_length=None,
    window_fn=functools.partial(window_ops.hann_window, periodic=True),
    pad_end=False,
    name=None
)
```

Defined in `tensorflow/contrib/signal/python/ops/spectral_ops.py`.

See the guide: Signal Processing (contrib) > Computing spectrograms

Computes the Short-time Fourier Transform of `signals`.

Implemented with GPU-compatible ops and supports gradients.

Args:

- `signals`: A `[..., samples]` `float32` `Tensor` of real-valued signals.
- `frame_length`: An integer scalar `Tensor`. The window length in samples.
- `frame_step`: An integer scalar `Tensor`. The number of samples to step.
- `fft_length`: An integer scalar `Tensor`. The size of the FFT to apply. If not provided, uses the smallest power of 2 enclosing `frame_length`.
- `window_fn`: A callable that takes a window length and a `dtype` keyword argument and returns a `[window_length]` `Tensor` of samples in the provided datatype. If set to `None`, no windowing is used.
- `pad_end`: Whether to pad the end of `signals` with zeros when the provided frame length and step produces a frame that lies partially past its end.
- `name`: An optional name for the operation.

Returns:

A `[..., frames, fft_unique_bins]` `Tensor` of `complex64` STFT values where `fft_unique_bins` is `fft_length // 2 + 1` (the unique components of the FFT).

Raises:

- `ValueError`: If `signals` is not at least rank 1, `frame_length` is not scalar, or `frame_step` is not scalar.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**