Slightly more complicated segmentation.

LSS 12 April 2109

Again, this is based on work I did in the 1990’s. It extends the technique described in SimpleSegmentation.doc to include leaky integrate and fire neurons to integrate the signals across adjacent channels (where “adjacent” means whatever the topology of the network implies). Here adjacency uses simple convergence: the parameter convergence (alterable, like most parameters through the varargin system) determines how many onset signals (and offset signals) are input to each LIF neuron. There are always the same number of LIF neurons, one per bandpass channel, so those near the “edge” will not receive as many inputs. The weights from the channels to these neurons should reflect this, but currently they don’t.

In the function described here, findsegments\_2.m the onset and offset signals are applied to an LIF array.

Altogether, the function:

1. Calculates the onset/offset signal for each band
2. Computes the onset signal and the offset signal from this (max(0, abs()), and max(0, abs(-)) (optional: take log(1+ this signal) to compress the signals, attempting to compensate for the large dynamic range in audio signals)
3. Applies a convergence to these multiple channel signals to produce the converged input to an LIF neuron (where there is one LIF neuron for each channel, that is, one set for onsets, and one set for offsets). Note that if the convergence is 0, then each LIF neuron receives input only from its own channel. The weight value supplied (onset\_wt, and offset\_wt) is shared out evenly between the inputs, so that the total input to each LIF neuron is approximately independent of the convergence value.
4. Each LIF neuron creates a spike train: the parameters governing the LIF neurons are the dissipation (1/time constant), onset\_diss and offset\_diss, and the refractory periods, onset\_rp and offset\_rp. The threshold is fixed at 1.

Summarized onsets (and offsets), are created by simply summing across bands, using shorter time buckets than the (1/sampling frequency) used in the LIF neurons. To avoid issues related to precise timing of “buckets”, they are summarized into sets of buckets (ceil(summaryintegratelength/summarysteplength)of them).

1. We then use findpeaks to find the peak times which will be candidate starts of segments (onsets) and ends of segments (offsets),
2. We then produce the segmentation from these candidates. We demand that each segment be at least minseglength long (default is 0, alterable using varargin). To achieve this, we start with the first candidate start of segment, then seek an nd of segment at least minseglength later. If shortestsegment is true, we use this one: if it is false, we seek forward in the candidate end segments, seeking the last candidate before the next candidate segment start.
3. The segments found are returned either as an empty array or as an n by 2 array.

New parameters and default values (alterable using varargin)

% new parameters for LIF based onset and offset

onset\_diss = 100 ; % dissipation for onset neurons

onset\_rp = 0.05 ; % refractory period for onset cells

onset\_wt = 40.0 ; % onset weight

offset\_diss = 100 ; % dissipation for offset neurons

offset\_rp = 0.05 ; % refractory period for onset cells

offset\_wt = 40.0 ; % offset weight

convergence = 4 ; % convergence (no of inputs to each neuron = 2\*convergence + 1)

logonset = false ; % default is not to take logs of onset signals

% new parameters for calculating actual segments

summarysteplength = 0.005 ; % step length used in summarising onset and offset spikes: 5ms default

summaryintegratelength = 0.02; % width of histogram used in summarising onset and offset spikes: 20ms default

shortestsegment = false ; % used to enable earlier or later candidate offset (segment end) times to be used

The appropriate values for parameters is not clear. For example, onset\_wt and offset\_wt seem to work well at 100 for some signals (e.g. noise free speech), and 150 for others (e.g. speech with a loud interfering tone). How might these be set?