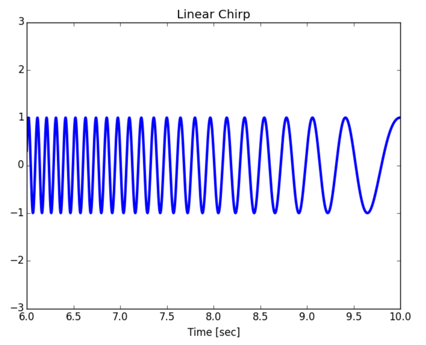
Learning filter widths of spectral decompositions with wavelets

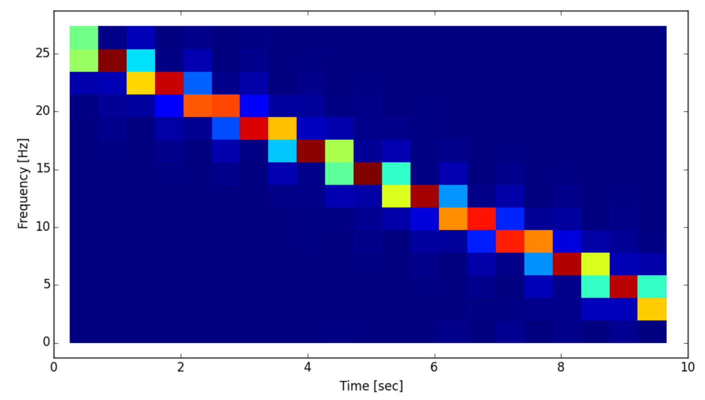
Haidar Khan[1], Bülent Yener[1]

1: Department of Computer Science, Rensselaer Polytechnic Institute, USA

Time series are often represented using spectral decompositions. . .



A linear chirp signal . . .



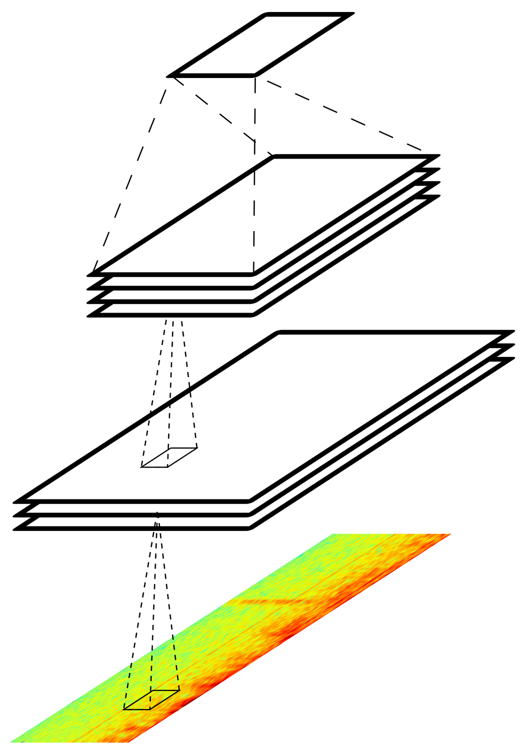
. . . represented as a spectrogram

. . . which are used to perform classification

Output

Convolution

Input (spectrogram)

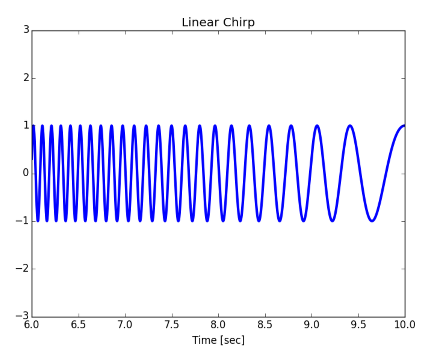


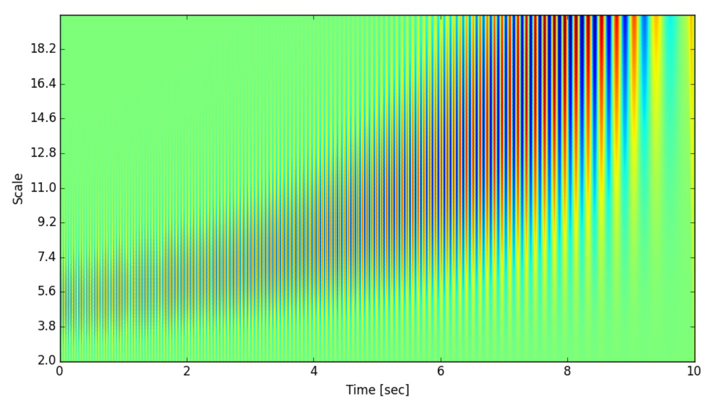
The spectrogram is used as input to a convolutional network. The parameters of the spectrogram are optimized as hyperparameters.

The wavelet transform can also be used to perfom spectral decomposition

Definition. The Ricker wavelet function ψ_s is given by

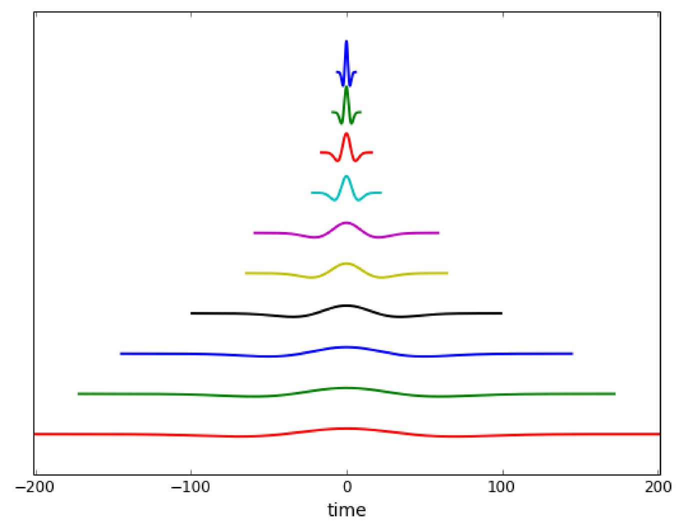
$$\psi_s(t) = \frac{2}{\sqrt{3s}\pi^{\frac{1}{4}}}\left(1-\frac{t^2}{s^2}\right)e^{-\frac{t^2}{2s^2}} \quad (1)$$





Wavelet filters are both localized and differentiable. . .

Convolution of a signal with stretched and contracted wavelet functions yields a multiresolution spectral decompositon of the signal.
The s parameter is used to control the width of the wavelet function filters.



ψ_{s_i} is the wavelet function at scale s_i discretized over a grid of K points. The gradient of the filter $\psi_{s_i,k} = AMG$ with respect to s_i can be written as:

$$\frac{\delta \psi_{s_i,k}}{\delta s_i} = A(M\frac{\delta G}{\delta s_i} + G\frac{\delta M}{\delta s_i}) + MG\frac{\delta A}{\delta s_i} \quad (2)$$

with A, M, and G defined as:

$$A = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3s_i}} \quad M = (1-\frac{t_k^2}{s_i^2}) \quad G = e^{-\frac{t_k^2}{2s_i^2}} \quad (3)$$

. . . allowing the filter widths to be learned using gradient descent

Gradient descent involves calculating $\delta E/\delta s_i$, where E is the differentiable loss function being minimized. The backpropagation algorithm yields $\delta E/\delta z_{ij}$. We can write the partial derivative of E with respect to each scale parameter s_i as:

$$\frac{\delta E}{\delta s_i} = \sum_{k=1}^{K} \frac{\delta E}{\delta \psi_{s_i,k}} \frac{\delta \psi_{s_i,k}}{\delta s_i} \quad (4)$$

The gradient with respect to the filter $\psi_{s_i,k}$ can be written using $\delta E/\delta z_{ij}$:

$$\frac{\delta E}{\delta \psi_{s_i,k}} = \sum_{j=1}^{N} \frac{\delta E}{\delta z_{ij}} \frac{\delta z_{ij}}{\delta \psi_{s_i,k}} = \sum_{j=1}^{N} \frac{\delta E}{\delta z_{ij}} x_{j+k} \quad (5)$$

Gradient descent is used to update the scale parameters: $s_i^* = s_i - \gamma\frac{\delta E}{\delta s_i}$.

- The relevant spectral content is learned with backpropagation.
- The kernel support (filter size) is adapted via the scale parameter.
- The number of hyperparameters to be optimized with expensive crossvalidation is reduced.

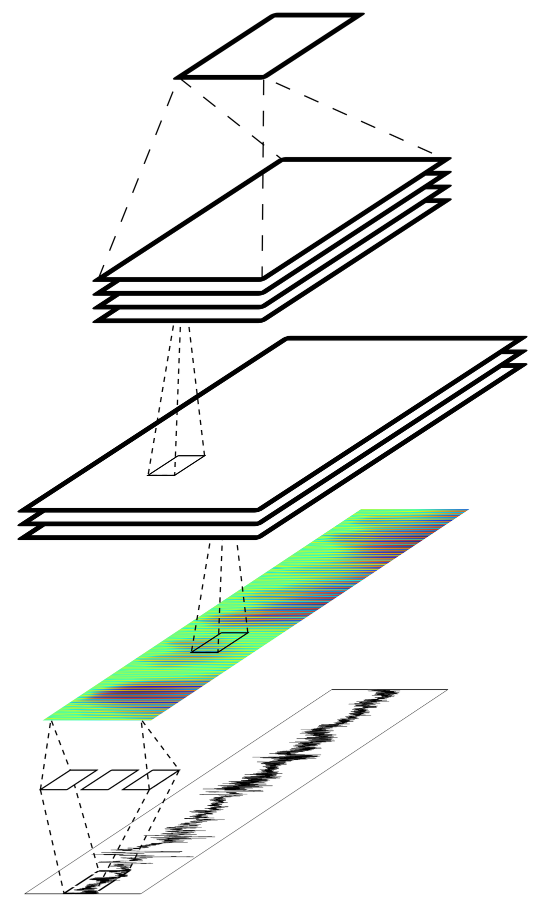
Possible to perform end-to-end time series classification

Output

Convolution

Wavelet deconvolution

Input (signal)

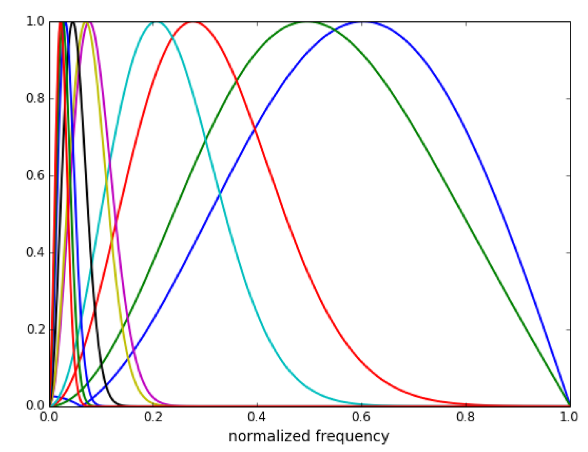


Learning the spectral decomposition yields improvements in accuracy

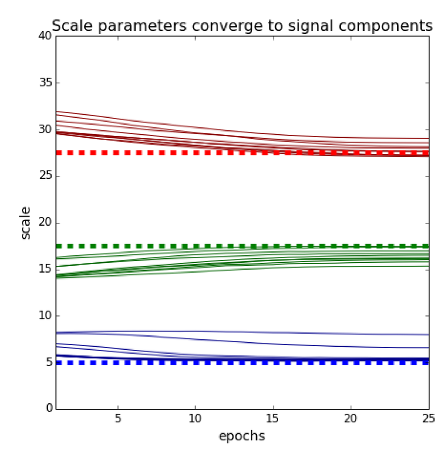
Method	PER on the TIMIT dataset
DNN + RNN	18.8
CNN	18.9
WD + CNN	18.1
LSTM RNN	17.7
Hierarchical CNN	**16.5**

Method	Test error on the Haptics dataset
BOSS	0.536
ResNet	0.495
COTE	0.488
FCN	0.449
WD + CNN	**0.425**

The resulting wavelet functions are interpretable



Learned wavelet filters



Convergence to generating frequencies