Time Series and Sequence Learning

Temporal Convolutional Networks, cont'd

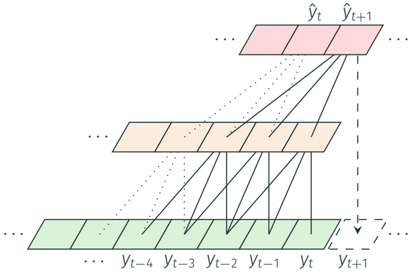
Fredrik Lindsten, Linköping University

Temporal Convolutional Network

2-layer TCN:

$$h_t^{(0)} = y_t,$$
$$h_t^{(1)} = \sigma(W^{(1)} H_t^{(0)} + b^{(1)}),$$
$$y_{t+1} = W^{(2)} H_t^{(1)} + b^{(2)} + \varepsilon_{t+1},$$

with $\varepsilon_{t+1} \sim N(0, \sigma_\varepsilon^2)$.

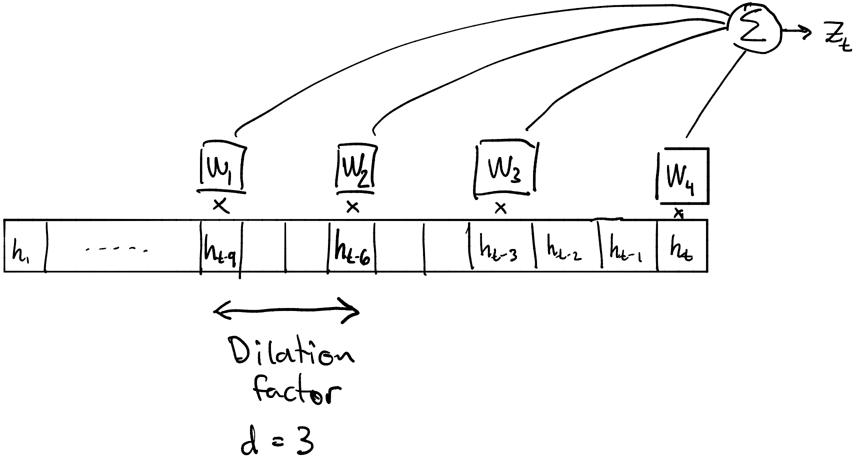


Can extend to multiple layers of hidden signals, $h_t^{(2)}$, $h_t^{(3)}$, ...

- ▲ Multiple layers $\Rightarrow$ very flexible models
- ▲ Receptive field increases with depth...
- ▼ ...but only linearly

Dilated convolutions

$$W = [w_1 \quad w_2 \quad w_3 \quad w_4]$$



Dilated convolutions

Formally

$$H_t = \begin{bmatrix} h_{t-(p-1)d} \\ \vdots \\ h_{t-2d} \\ h_{t-d} \\ h_t \end{bmatrix}, \qquad Z_t = W$$

TCN with dilated convolutions

TCN with dilated convolutions:

$$
\begin{aligned}
h_t^{(0)} &= y_t, \\
h_t^{(\ell)} &= \sigma(W^{(\ell)} H_t^{(\ell-1)} + b^{(\ell)}), && \ell = 1, \ldots, L-1, \\
y_{t+1} &= W^{(L)} H_t^{(L-1)} + b^{(L)} + \varepsilon_{t+1}, && \varepsilon_{t+1} \sim N(0, \sigma_\varepsilon^2).
\end{aligned}
$$

Here

$$H_t^{(\ell)} := \begin{bmatrix} h_{t-(p-1)d^{(\ell)}}^{(\ell)} & \cdots & h_{t-d^{(\ell)}}^{(\ell)} & h_t^{(\ell)} \end{bmatrix}^\top$$

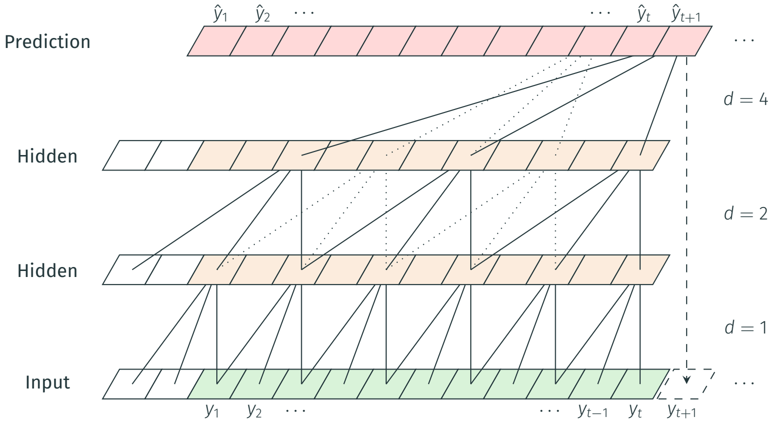
and $d^{(\ell)}$ is the dilation factor used in layer ℓ.

Often we let the dilation increase with the depth of the network, e.g.,

$$d^{(\ell)} := 2^\ell.$$

TCN with dilated convolutions

By using dilated convolutions we can increase receptive field exponentially with depth.



Multi-channel TCN

All the tricks of the trade from deep learning can be used to extend the TCN model!

In particular:

Multi-dimensional hidden signals:

$$\text{Let } \dim h_t^{(\ell)} = c^{(\ell)} \geq 1.$$

- 1-dimension signal of length $n \iff$ "image" of size $n \times 1 \times 1$
- $c^{(\ell)}$-dimensional signal of length $n \iff$ "image" of size $n \times 1 \times c^{(\ell)}$
- Dimension of convolutional filter $W^{(\ell)} : p \times c^{(\ell-1)} \times c^{(\ell)}$

Residual connections

Residual connections: Direct linear term relating $h_t^{(\ell-1)}$ to $h_t^{(\ell)}$.

$$h_t^{(\ell)} = \sigma(W^{(\ell)} H_t^{(\ell-1)} + b^{(\ell)}) + V^{(\ell)} h_t^{(\ell-1)}.$$

Useful for training deep models — enables "easier flow of gradient information".

General TCN formulation

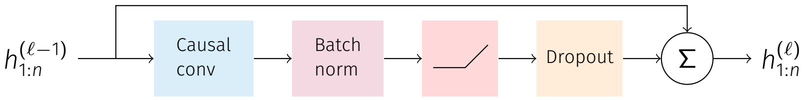
More generally, a TCN cen be written as:

$$\begin{aligned} h_t^{(0)} &= y_t, \\ h_t^{(\ell)} &= g_{\theta}^{(\ell)}(H_t^{(\ell-1)}), & \ell = 1, \ldots, L-1, \\ y_{t+1} &= g_{\theta}^{(L)}(H_t^{(L-1)}) + \varepsilon_{t+1}, & \varepsilon_{t+1} \sim N(0, \sigma_\varepsilon^2). \end{aligned}$$

with $H_t^{(\ell)} := \begin{bmatrix} h_{t-(p-1)d^{(\ell)}}^{(\ell)} & \cdots & h_{t-d^{(\ell)}}^{(\ell)} & h_t^{(\ell)} \end{bmatrix}^\top$ for

arbitrary nonlinear functions $g_{\theta}^{(1)}, \ldots, g_{\theta}^{(L)}$.

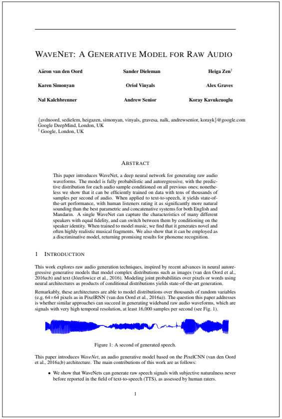
ex) One possible residual block of a TCN

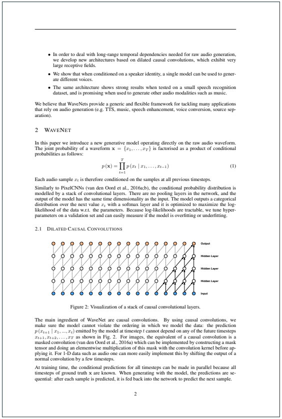


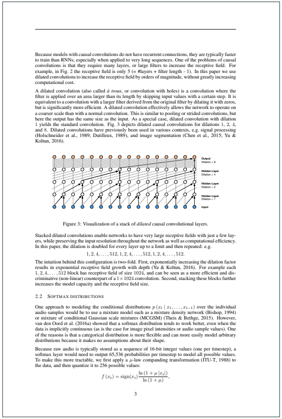
ex) WaveNet

WaveNet by DeepMind powers Google's text-to-speech technology.

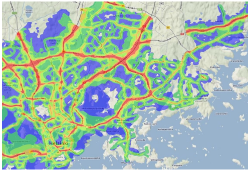
WaveNet: A Generative Model for Raw Audio. **Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu.** *arXiv:1609.03499*, 2016.

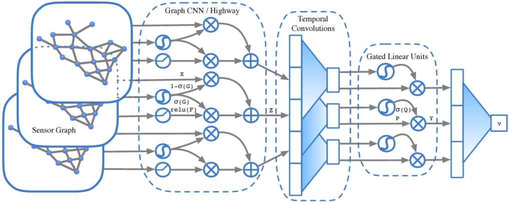






ex) Traffic prediction





Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. **Bing Yu, Haoteng Yin, Zhanxing Zhu** *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Deep Learning for time series analysis

So is this what we should always do for modeling sequential data?!

No!

- Only makes sense to use something as complex as TCN when classical methods fail — **Try Simple Things First!**
- Methods based on deep learning work best if we have multiple sequences, or one long sequence that can be split into segments
- For a single univariate time series, classical methods (AR, state space, NAR, ...) often work better.