An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling

Shaojie Bai, J. Zico Kolter, Vladlen Koltun

14 January 2020

Presented by Lucian M. Sasu, Ph.D.
ML Reading Group, 14 January 2021

https://arxiv.org/pdf/1803.01271.pdf

Outline

Metadata

- Venue: CoRR, https://arxiv.org/pdf/1803.01271.pdf, 2018
- Authors' affiliations:
  - ML Department, Carnegie Mellon University
  - CS Department, Carnegie Mellon University
  - USA Intel Labs, Santa Clara
- Visibility:
  - 906 citations as reported by Semantic Scholar
  - 155 highly influential – Semantic Scholar
  - 1022 citations — Google Scholar
- Pytorch code by paper's authors: https://github.com/locuslab/TCN

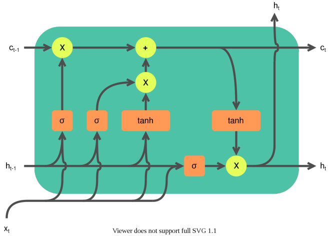
Aims and findings

- traditionally, a sequence modeling task is solved through recurrent neural networks (LSTM, GRU)
- at the moment of writing, some conv-based networks obtained state of the art results in audio synthesis, language modeling, machine translation
- paper's target: empirical evaluation of convolutional and recurrent architectures on RNNs benchmark tasks
- approach: Temporal Convolutional Networks (TCNs)
- in-depth analysis of memory retention of TCNs: longer memory than RNNs

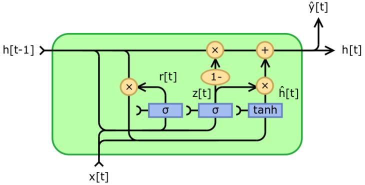
Outline

Background: RNNs

- RNNs: a vector of learned hidden states is propagated through time
- vanilla RNNs – difficult to train
- current architectures: Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) networks



(a) LSTM cell; source



(b) GRU cell; source

Figure 1: LSTM and GRU cells

Background: RNNs

- multiple empirical studies for recurrent networks: which is the best?
  - tests with tens of thousands of architectures
  - not trivial to find architectures better than LSTMs
  - models post–LSTM are often surpassed by classical LSTMs
  - variants and hybridizations with other techniques: Convolutional LSTM, Quasi–RNN, dilated RNNs

Background: TCNs

- Temporal Convolutional Networks (TCNs): "a simple descriptive term for a family of architectures"
- features:
  1. causal convolutions, no leakage from future to past
  2. can take a sequence of any length and produce a sequence of same length
  3. no gating mechanisms
  4. bonus: hints on how to build very long effective history sizes
  5. simpler than other CNN–based nets
- *not to be confused with [1]*

Background: Sequence modeling in TCNs

- input: input sequence $x_0, \ldots, x_T$
- associated output: sequence $y_0, \ldots, y_T$
- causal constraint is satisfied: value y_t is predicted based on $x_0, \ldots, x_t$ and not on $x_{t+1}, \ldots, x_T$
- training goal: find a network $f : \mathcal{X}^{T+1} \rightarrow \mathcal{Y}^{T+1}$, that produces the mapping $\hat{y}_0, \ldots, \hat{y}_T = f(x_0, \ldots, x_T)$ s.t. f minimizes a certain loss function (NLL, MSE, ...)
- such a topic is also covered by atoregressive prediction, but fall outside machine translation, which may peek into the future

Background: Sequence modeling in TCNs

- 1 dimensional fully convolutional network architecture [2]
- the same output length is obtained through padding with zeros;
- each hidden layer and output layer have the same length as the input

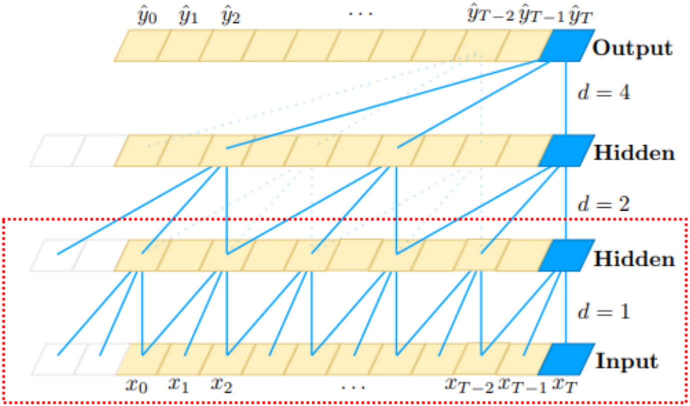


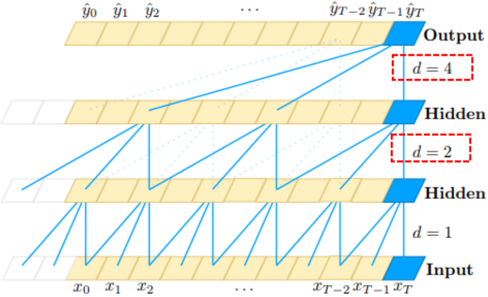
Figure 2: 1D fully convolutional, padding at the left, non–dilated convolution.
Source: [3]

Background: Causal convolutions

- “causal”: no leackage from future into the past
- output at time t is obtained by convolving kernel and inputs at time $t, t-1, \ldots, t-k+1$ (here: for $d=1$)
- TCN = 1D FCN + causal convolution
- disadvantage: one needs a very large number of hidden layers or large kernels to process a long history

Background: Dilated convolutions

- looking deep in history: dilated convolutions
- results: exponentially large receptive fields



- formal: input $\mathbf{x} \in \mathbb{R}^n$, filter (kernel) $f : \{0, \ldots, k-1\} \rightarrow \mathbb{R}$, one defines dilated conv F on s as:

$$F(s) = (\mathbf{x} *_d f)(\mathbf{s}) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \tag{1}$$

Background: Dilated convolutions

- $d = 1 \Rightarrow$ regular convolution
- $d > 1 \Rightarrow$ larger range of receptive fields
- dilation increases exponentially with depth, $d = 2^i$, i being the index of the hidden layer
- result: there is at least one filter which touches each value of the input

Background: Residual connections, dropout

- residual connections [4] combine input $\mathbf{x}$ with its transformation $\mathcal{F}$; popular approach which improves results of DL architectures
- $o = Activation(\mathbf{x} + \mathcal{F}(\mathbf{x}))$

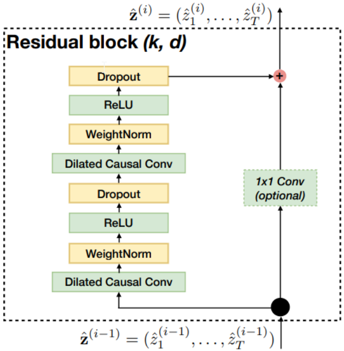


Figure 3: TCN residual block; source [3]

- dropout after each dilated convolution: a whole channel is zeroed out

TCN Advantages

- convs are parallel–friendly; a long sequence can be processed in parallel
- flexible receptive size: stacking more dilated layers, using larger dilation factors, increasing the filters size: about the same effect
- no vanishing/exploding gradients as in vanilla RNNs
- low memory requirement for training, unlike for partial results which feeds LSTM and GRU gates
- allows for variable length inputs, by sliding the kernel

TCN Disadvantages

- data storage during evaluation - the whole input sequence $x_0, \ldots, x_T$ must be maintained to build the estimation
- transfer learning might not work: a pretrained model with small k, d might be inappropriate for a problem where large history must be learned

Outline

Sequence Modeling Tasks

- adding problem
- sequential MNIST and P–MNIST
- copy memory
- JSB Chorales and Nottingham
- PennTreebank
- LAMBADA
- text8

Experimental setup and results

- quite the same TCN architecture used across experiments, with different depth of network and kernel size; exponential dilation $d = 2^i$
- recurrent networks: around the same number of parameters as TCN

Sequence Modeling Task	Model Size (≈)	Models			
		LSTM	GRU	RNN	**TCN**
Seq. MNIST (accuracy^h)	70K	87.2	96.2	21.5	**99.0**
Permuted MNIST (accuracy)	70K	85.7	87.3	25.3	**97.2**
Adding problem T=600 (loss$^\ell$)	70K	0.164	**5.3e-5**	0.177	**5.8e-5**
Copy memory T=1000 (loss)	16K	0.0204	0.0197	0.0202	**3.5e-5**
Music JSB Chorales (loss)	300K	8.45	8.43	8.91	**8.10**
Music Nottingham (loss)	1M	3.29	3.46	4.05	**3.07**
Word-level PTB (perplexity$^\ell$)	13M	**78.93**	92.48	114.50	88.68
Word-level Wiki-103 (perplexity)	-	48.4	-	-	**45.19**
Word-level LAMBADA (perplexity)	-	4186	-	14725	**1279**
Char-level PTB (bpc$^\ell$)	3M	1.36	1.37	1.48	**1.31**
Char-level text8 (bpc)	5M	1.50	1.53	1.69	**1.45**

Figure 4: Experimental results. Bold values on each line are the best.

- caveat: the canonical recurrent networks are not state of the art

Results

- adding problem: TCN has fastest convergence
- sequential MNIST and P–MNIST: better perfomance in convergence and final accuracy; outperforms state of the art
- copy memory: quick convergence; stress test for long sequences; TCN allows for much longer history, LSTM and GRU degenerate to random guessing
- JSB Chorales and Nottingham: better than canonic recurrent models, but under state of the art
- PennTreebank: for small datasets, LSTM wins; for larger ones, TCN is the best without extensive hyperparameter search
- LAMBADA: large ds, TCN wins; stress test for both local and non-local textual "understanding"
- text8: TCN beats canonical recurrent networks, but not SOTA

Outline

Conclusions

- dilated convolutions and residual connections might explain TCN performance; former CNNs did not work well in sequence tasks
- a clear winner when large history must be learned; under LSTM for short sequences
- small effort in hyperparameter tuning; see paper's supplementary material

Outline

Bibliography

Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager.
Temporal convolutional networks: A unified approach to action segmentation.
CoRR, abs/1608.08242, 2016.

Jonathan Long, Evan Shelhamer, and Trevor Darrell.
Fully convolutional networks for semantic segmentation.
CoRR, abs/1411.4038, 2014.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun.
An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.
CoRR, abs/1803.01271, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
CoRR, abs/1512.03385, 2015.