



Paper: Optimal Kronecker-Sum Approximation of Real Time Recurrent Learning

Poster: Online & Untruncated Gradients for RNNs

Frederik Benzing*, Marcelo Matheus Gauy*, Asier Mujika, Anders Martinsson, Angelika Steger

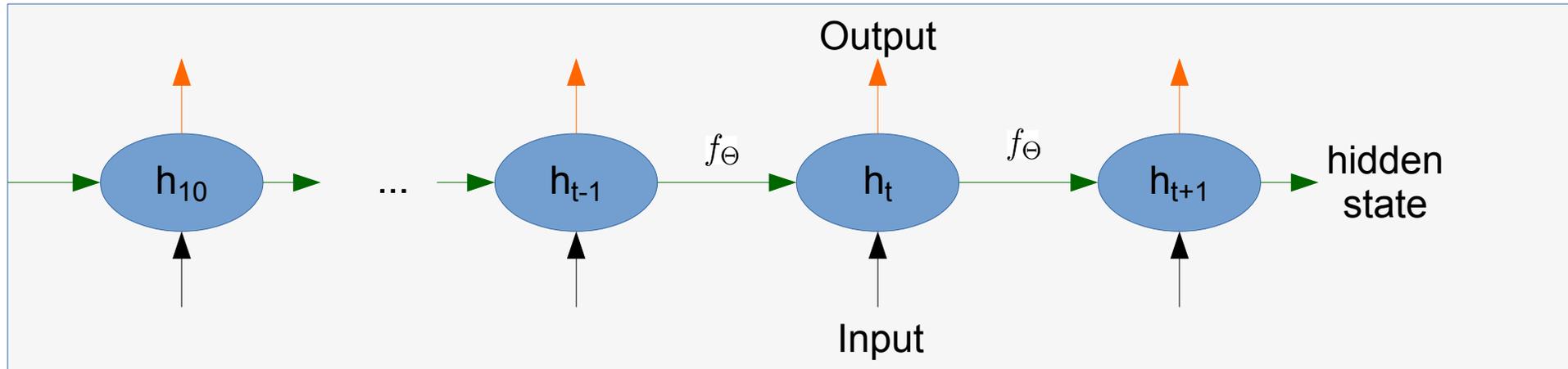
Recurrent Neural Nets (RNNs)

- Model **temporal** and **sequential data** (RL, audio synthesis, language modelling,...)
- One of the key research challenges:
Learn **Long-Term dependencies**

Training RNNs

Truncated Backprop Trough Time (TBPTT)

(Williams & Peng, 1990)



- Introduces arbitrary **Truncation Horizon** → no longer term dependencies
- **Parameter Update Lock** during forward & backward pass

Forward Computing Gradients

Real Time Recurrent Learning (RTRL) (Williams & Zipser, 1989)

Forward compute $G_t = \frac{dh_t}{d\Theta}$ with recurrence $G_{t+1} = H_t G_t + F_t$

- **Untruncated Gradients**
- **Memory** is independent of sequence length
- **Online parameter updates** (no update lock)

BUT: Need n^4 Runtime and n^3 Memory (for n hidden units) → infeasible

It looks like you want to do RTRL.

$$G_t = \frac{dh_t}{d\Theta} \in \mathbb{R}^{n \times n^2}$$

$$G_{t+1} = H_t G_t + F_t$$



Approximate RTRL to save time & space

Online Recurrent Optimization (UORO) (Tallec & Ollivier, 2017)

- **Idea:** Don't store G_t precisely, but **approximately**

$$G_t \approx u_t \otimes w_t$$

$\begin{matrix} \nearrow & & \nearrow \\ n \times 1 & & 1 \times n^2 \end{matrix}$

and **unbiasedly** approximate **recurrence** equation.

- Memory: n^2
- Runtime: n^3

It looks like you want to do RTRL.

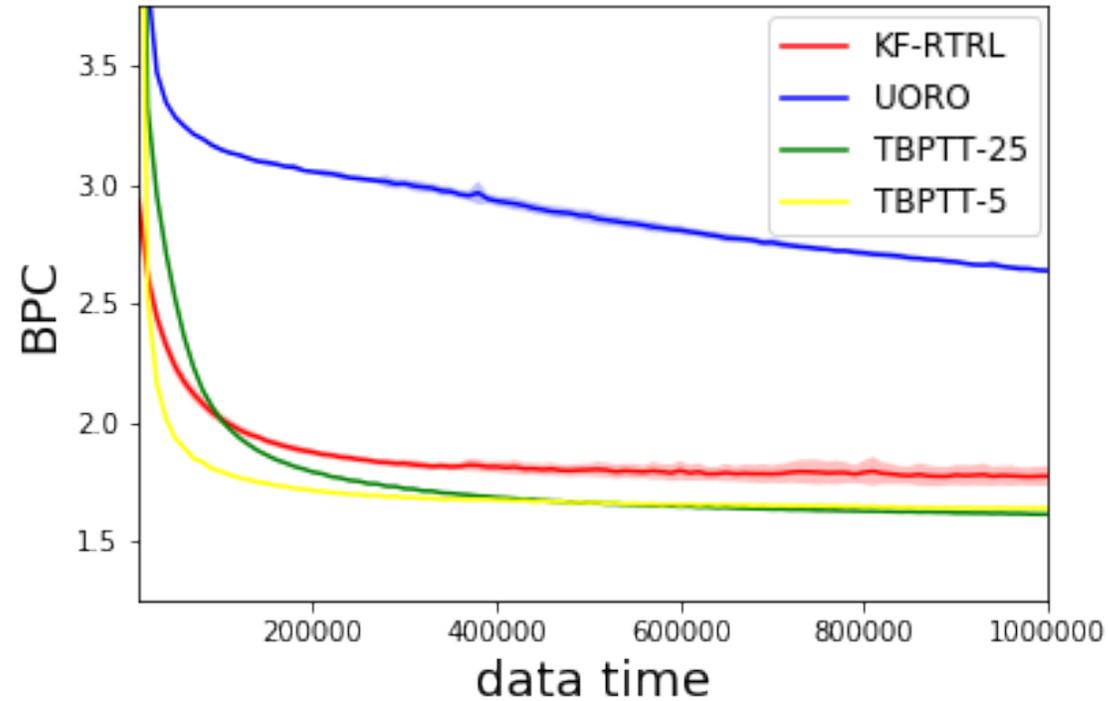
$$G_t = \frac{dh_t}{d\Theta} \in \mathbb{R}^{n \times n^2}$$

$$G_{t+1} = H_t G_t + F_t$$



Does it work? Part I

UORO (Tallec & Ollivier, 2017) and KF-RTRL (Mujika et al., 2018)

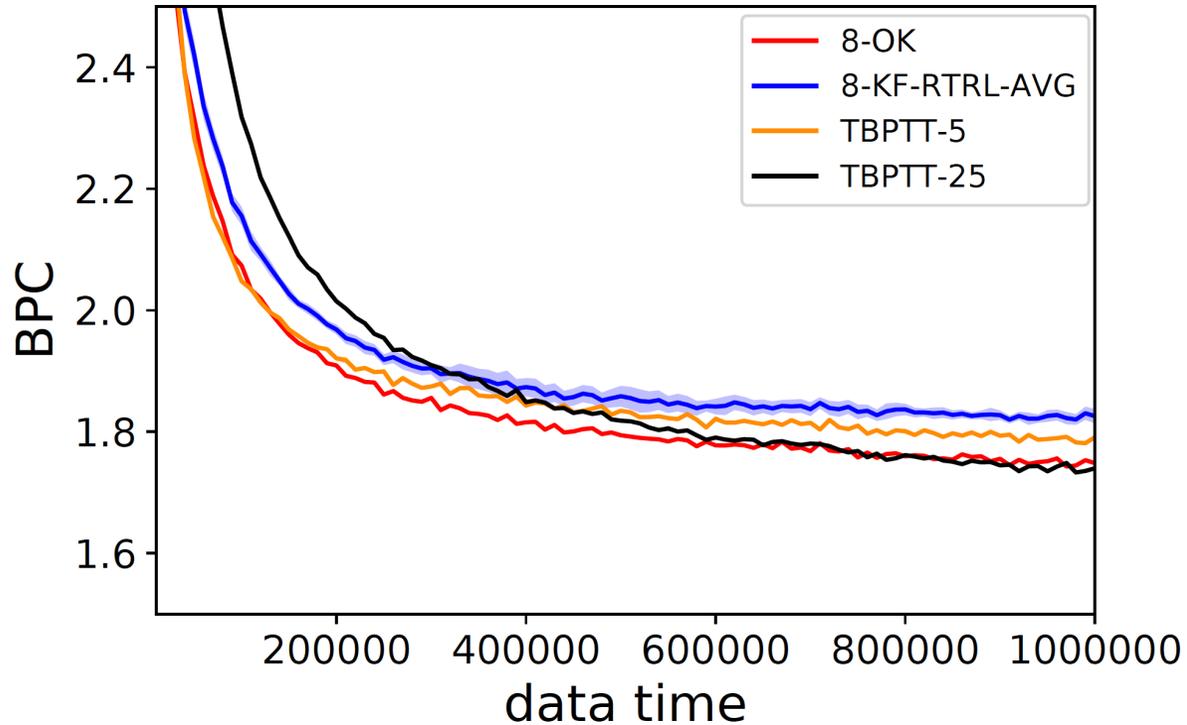


Character-level PTB

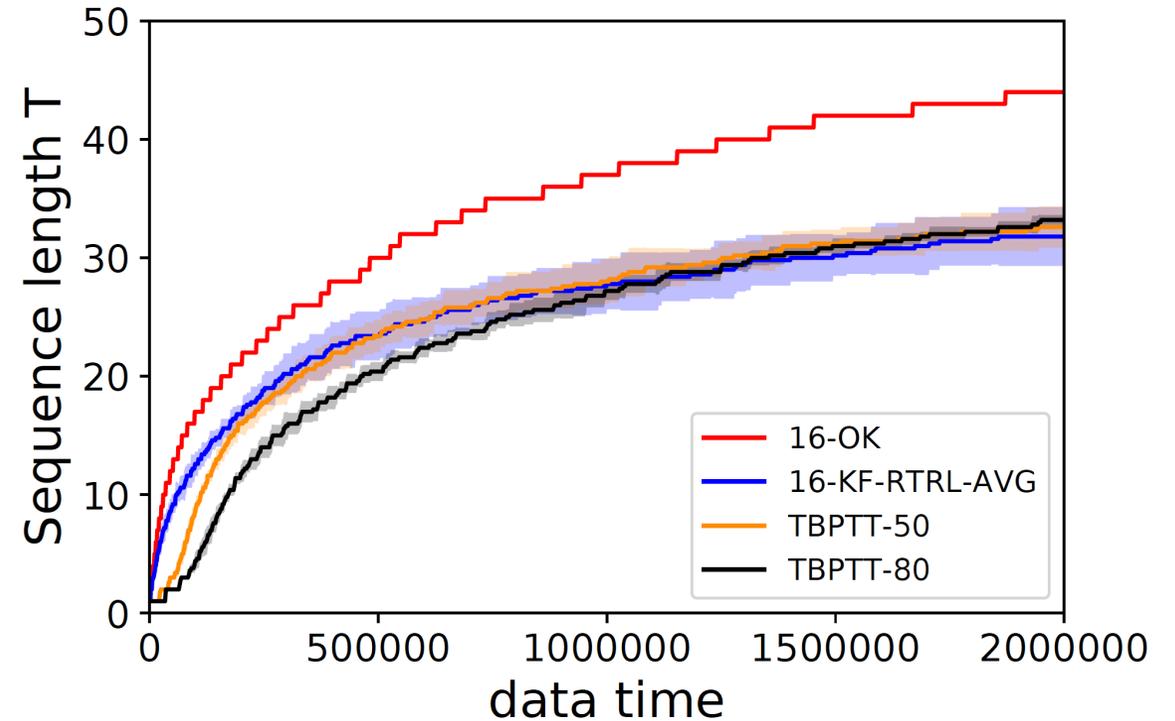
Does it work? Part II

Provably optimal approximation – Optimal Kronecker-Sum (OK)

(our contribution)



Character-level PTB



Copy Task

Input: #01101-----
Output: -----#01101

What to remember

- Truncated BPTT has problems (truncation, update lock)
- RTRL as online & untruncated alternative, but too costly
- Our OK approx of RTRL reduces costs by factor n
 - No performance loss
 - Break update lock \rightarrow faster convergence
 - Theoretically optimal (for certain class of approx)
- Still need to reduce computational costs

	Memory	Runtime	Unbiased&Online
RTRL	n^3	n^4	✓
r -OK	rn^2	rn^3	✓
TBPTT- T	Tn	Tn^2	✗

It looks like you got interested in RTRL. Have a look at **Poster #166**.

