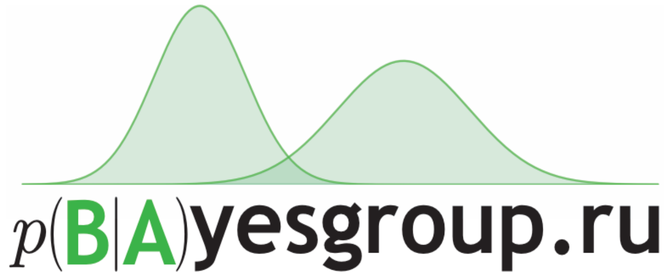
Structured Sparsification of Gated Recurrent Neural Networks

Ekaterina Lobacheva* Nadezhda Chirkova*

Alexander Markovich Dmitry Vetrov

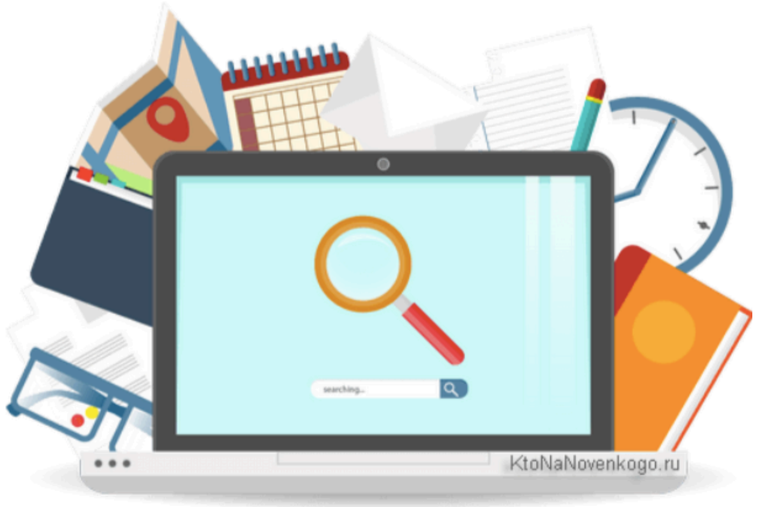
NATIONAL RESEARCH UNIVERSITY

SAMSUNG Research



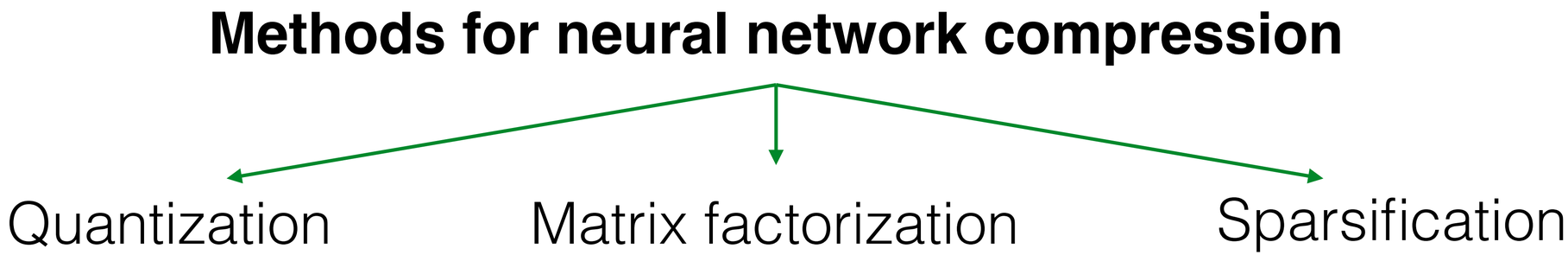
Compression of neural networks

- Deep neural networks achieve state-of-the-art performance in a variety of domains
- Model quality scales with model and dataset size
- State-of-the-art models usually incorporate **tens of millions of parameters**
- But **resources** (memory, processing time) **may be limited**



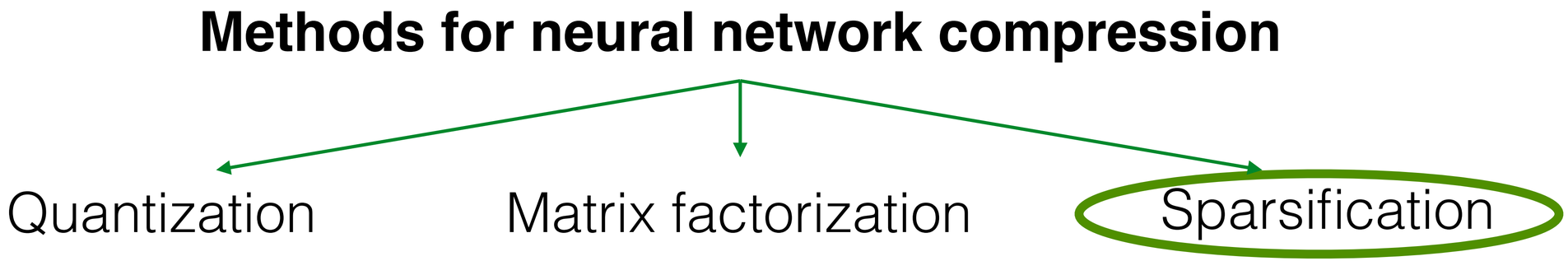
Compression of neural networks

- Deep neural networks achieve state-of-the-art performance in a variety of domains
- Model quality scales with model and dataset size
- State-of-the-art models usually incorporate **tens of millions of parameters**
- But **resources** (memory, processing time) **may be limited**



Compression of neural networks

- Deep neural networks achieve state-of-the-art performance in a variety of domains
- Model quality scales with model and dataset size
- State-of-the-art models usually incorporate **tens of millions of parameters**
- But **resources** (memory, processing time) **may be limited**

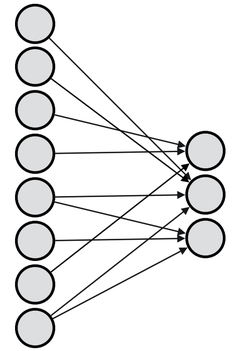


Sparsification of neural networks

Unstructured:
Learn sparse
weight matrices

⇨

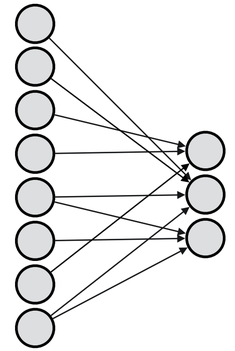
reduce
network
size



Sparsification of neural networks

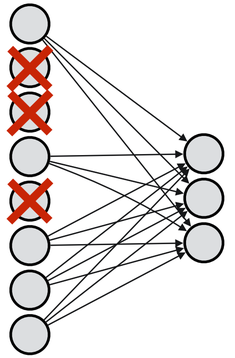
Unstructured:
Learn sparse
weight matrices

⇨ reduce network size



Structured:
Remove all weights
associated with
a structure unit
(e. g. neuron)

⇨ remove structure unit ⇨ speed up prediction



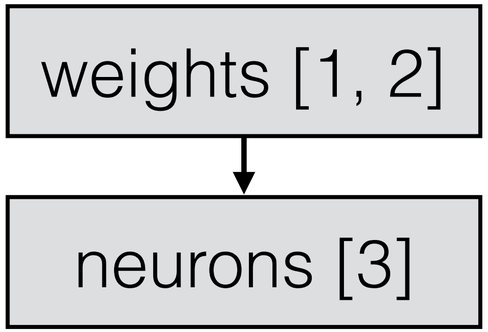
Problem setting

Goal: learn structured sparsity in gated RNNs (LSTM, GRU…)

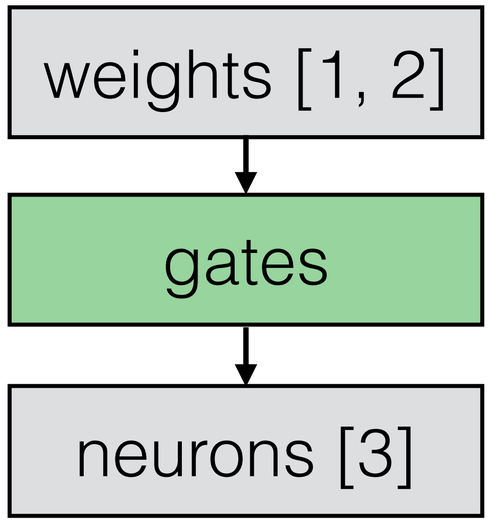
Problem setting

Goal: learn structured sparsity in gated RNNs (LSTM, GRU…)

What structure units to sparsify
in vanilla RNN?



in gated RNN?

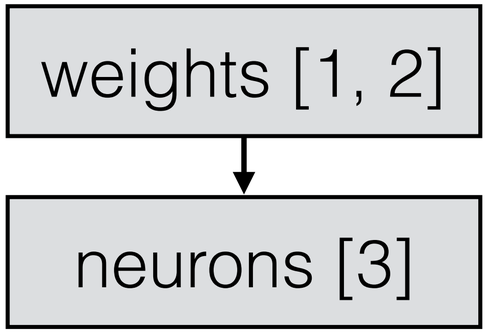


[1] Narang et al. Exploring sparsity in recurrent neural networks. ICLR 2017
[2] Chirkova et al. Bayesian sparsification of recurrent neural networks. EMNLP 2018
[3] Wen, et al. Learning intrinsic sparse structures within long short-term memory. ICLR 2018

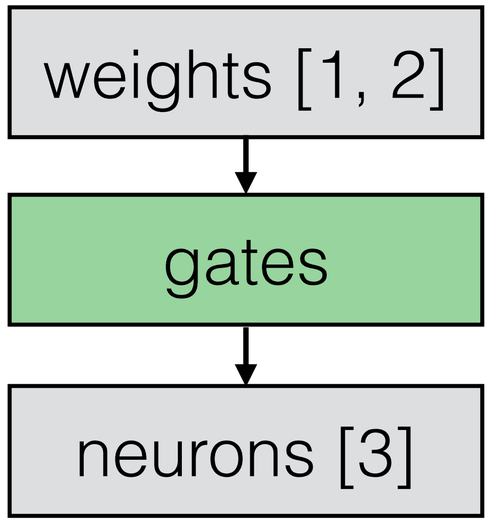
Problem setting

Goal: learn structured sparsity in gated RNNs (LSTM, GRU…)

What structure units to sparsify
in vanilla RNN?



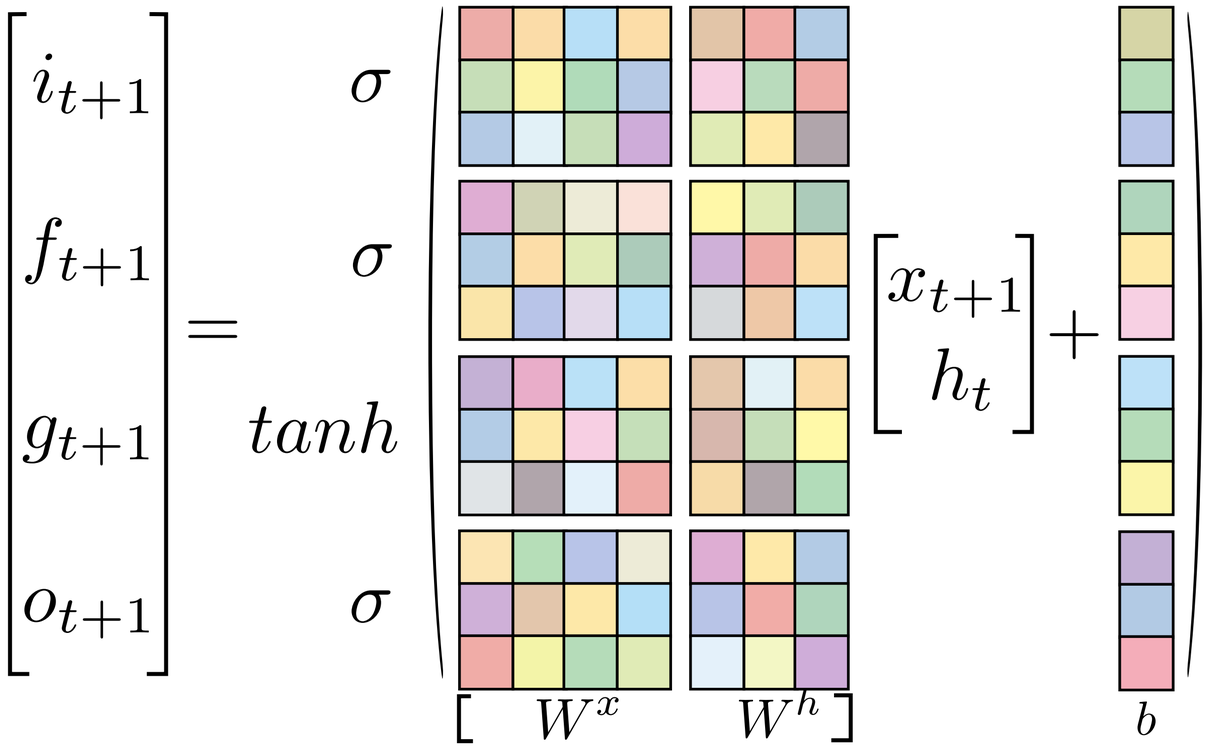
in gated RNN?



Main idea: introduce intermediate level of sparsification — gates

[1] Narang et al. Exploring sparsity in recurrent neural networks. ICLR 2017
[2] Chirkova et al. Bayesian sparsification of recurrent neural networks. EMNLP 2018
[3] Wen, et al. Learning intrinsic sparse structures within long short-term memory. ICLR 2018

LSTM

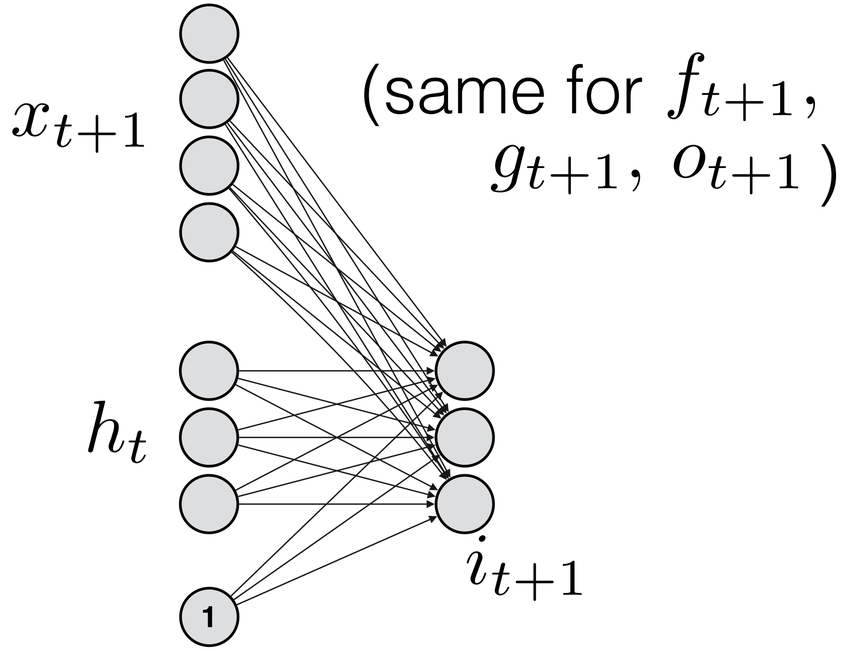


$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$

$$h_{t+1} = o_{t+1} \odot g_{t+1}$$

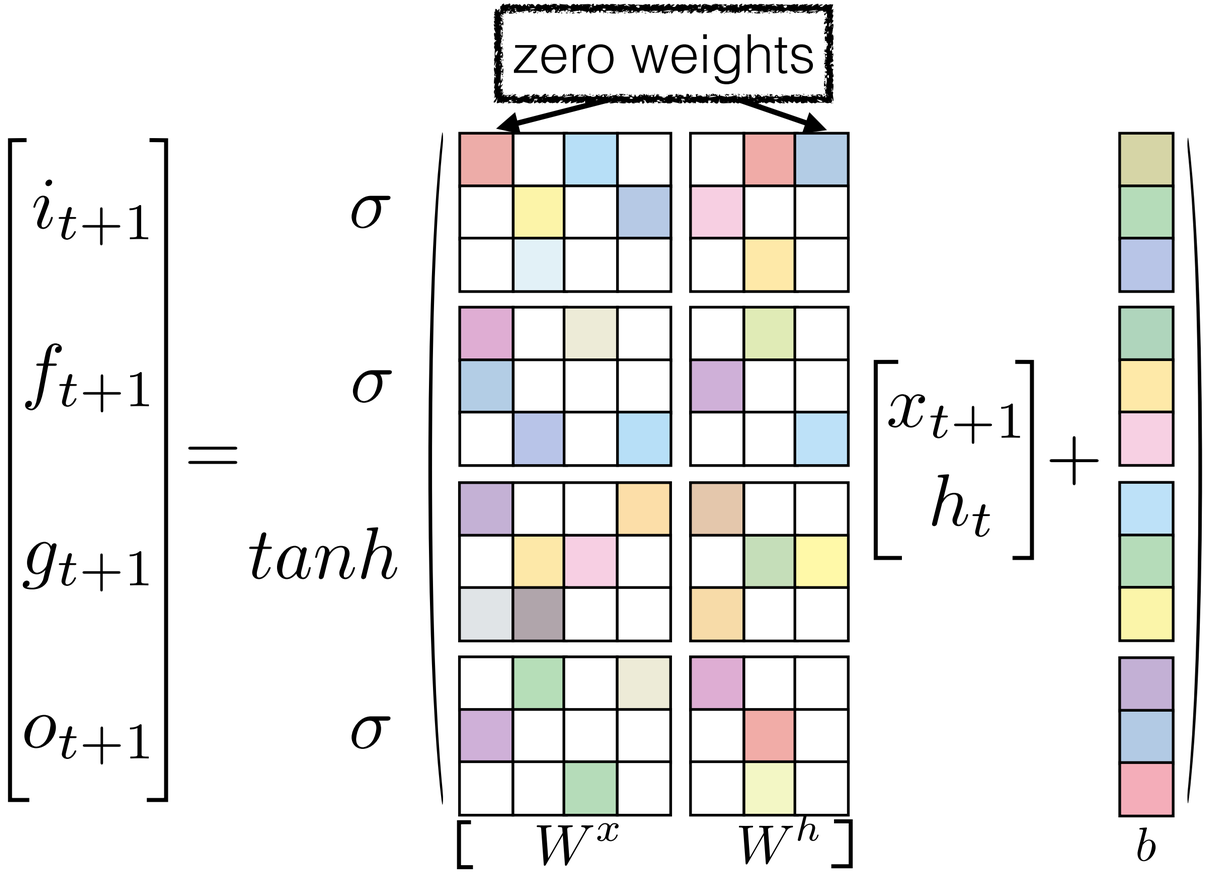
LSTM

$$\begin{bmatrix} i_{t+1} \\ f_{t+1} \\ g_{t+1} \\ o_{t+1} \end{bmatrix} = \begin{matrix} \sigma \\ \sigma \\ tanh \\ \sigma \end{matrix} \left(\begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & W^x & & & & W^h & & \end{bmatrix} \begin{bmatrix} x_{t+1} \\ h_t \end{bmatrix} + \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ b \end{bmatrix} \right)$$

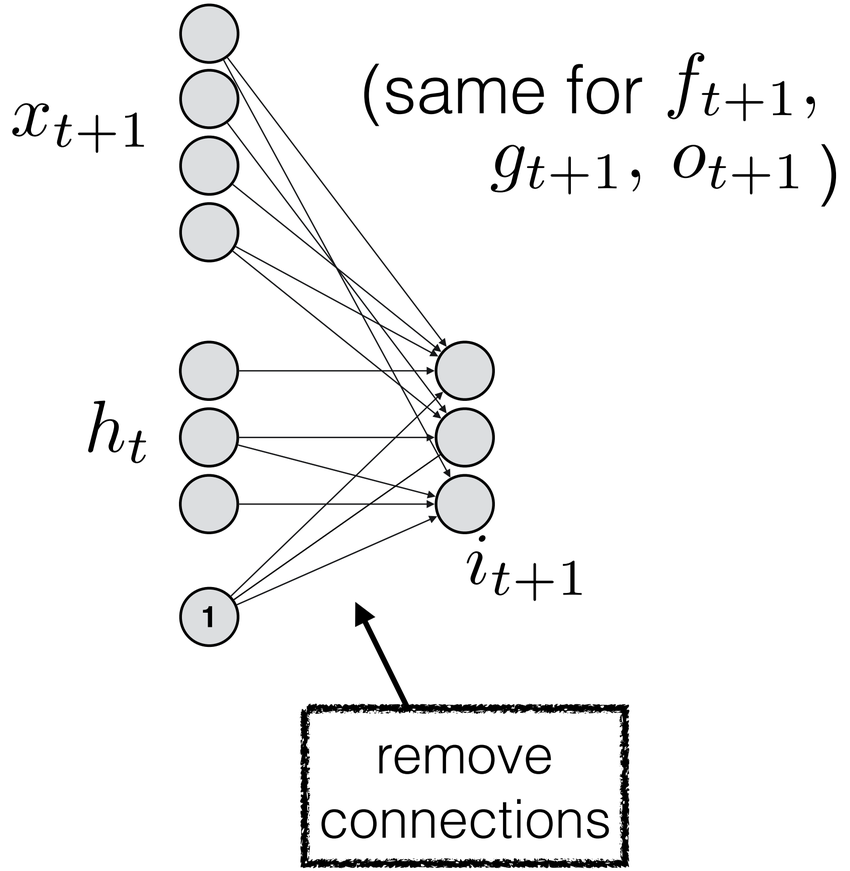


$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$

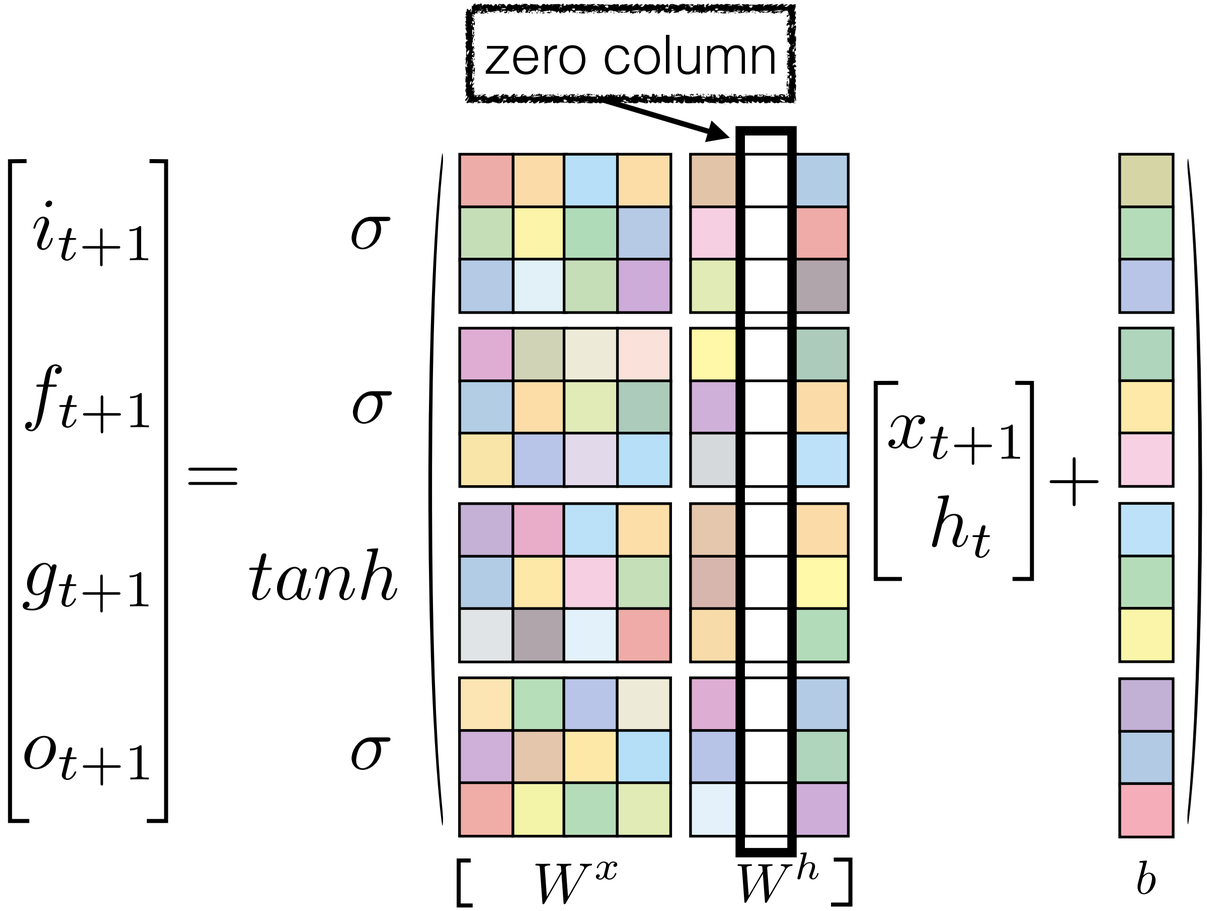
LSTM: remove individual connections



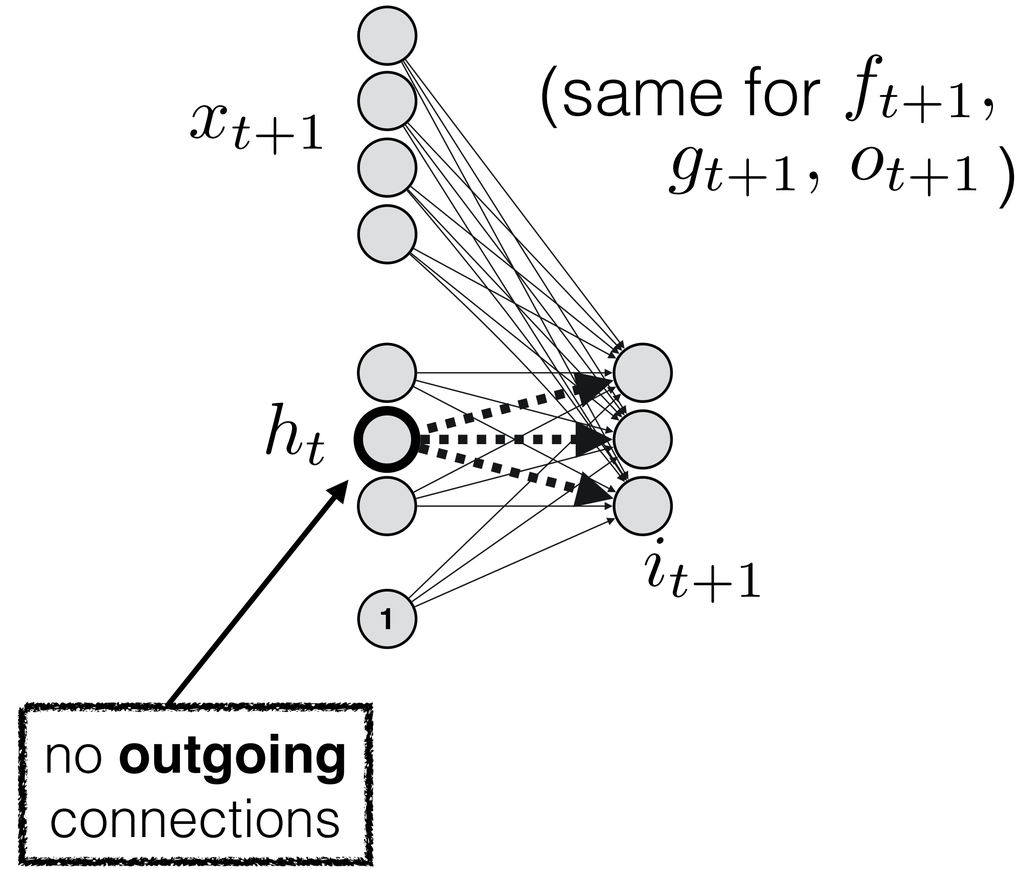
$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$



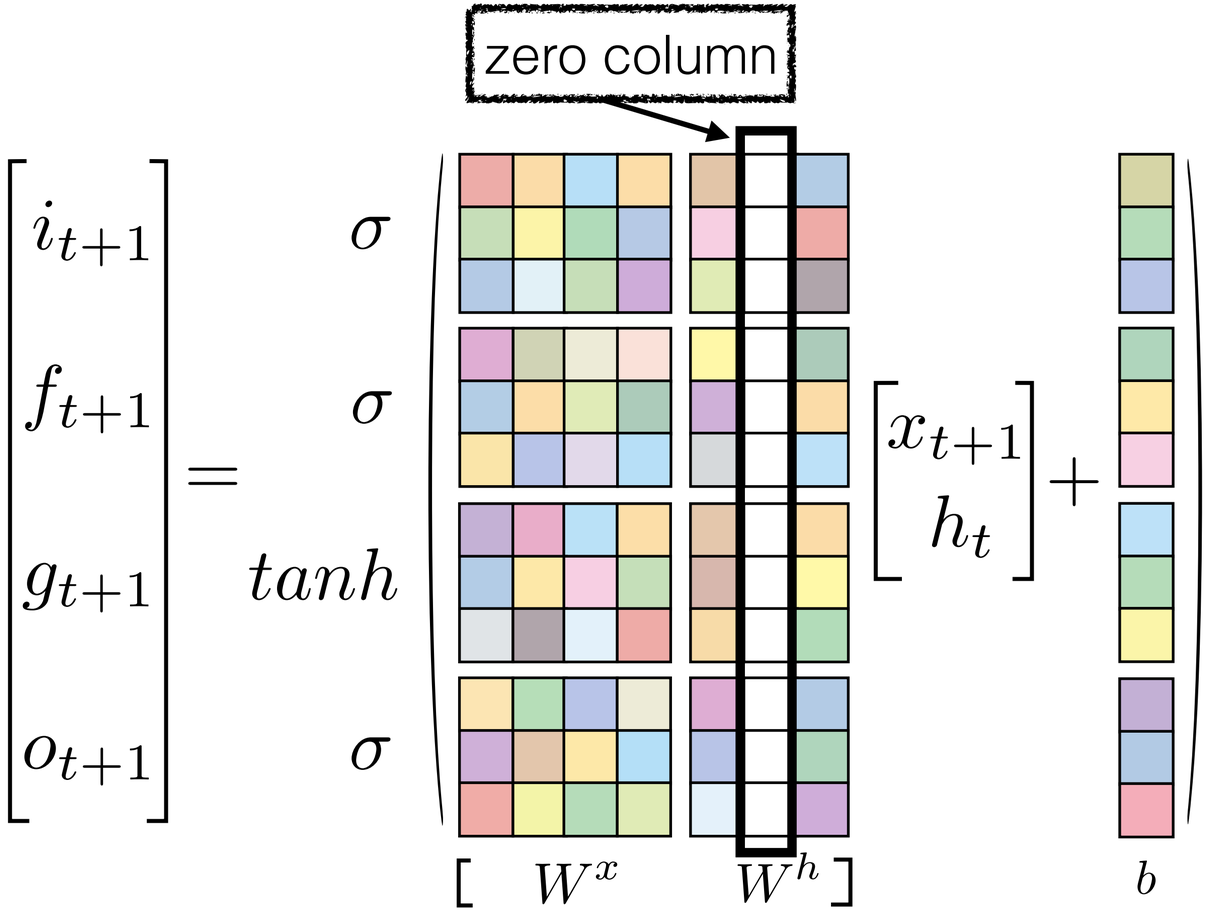
LSTM: remove neurons



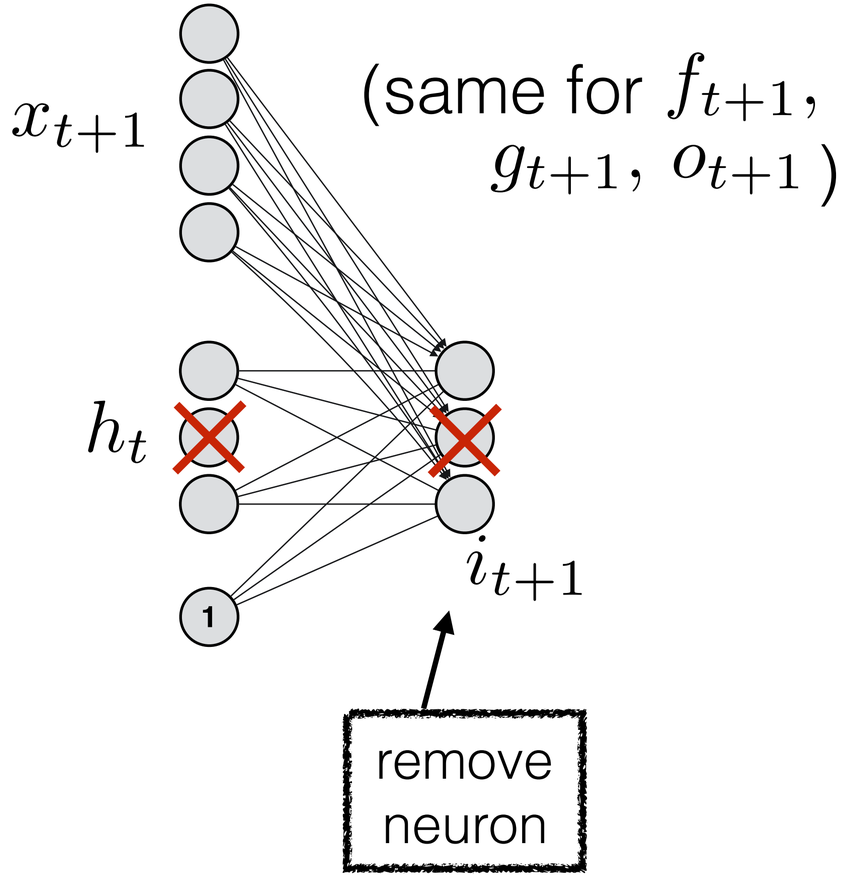
$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$



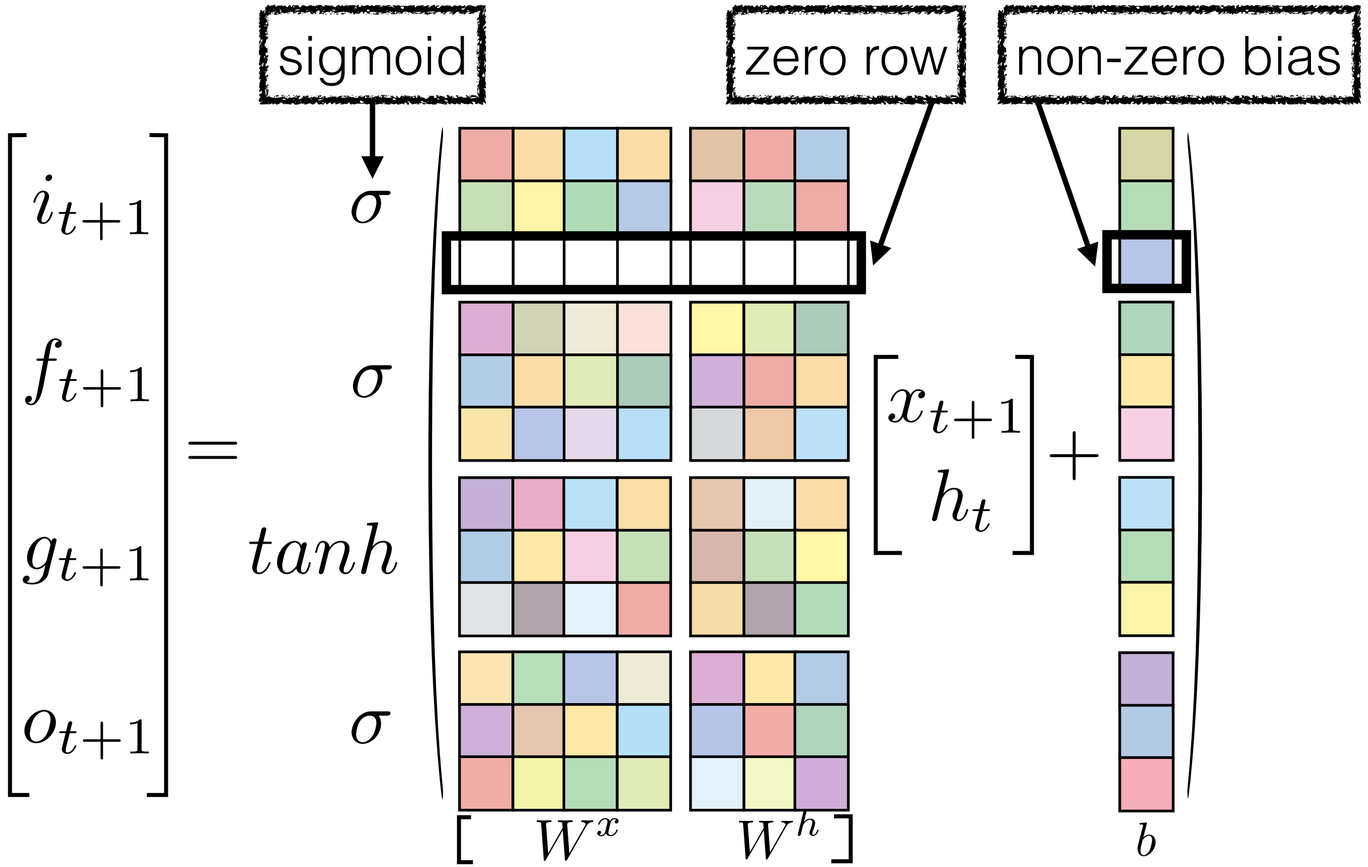
LSTM: remove neurons

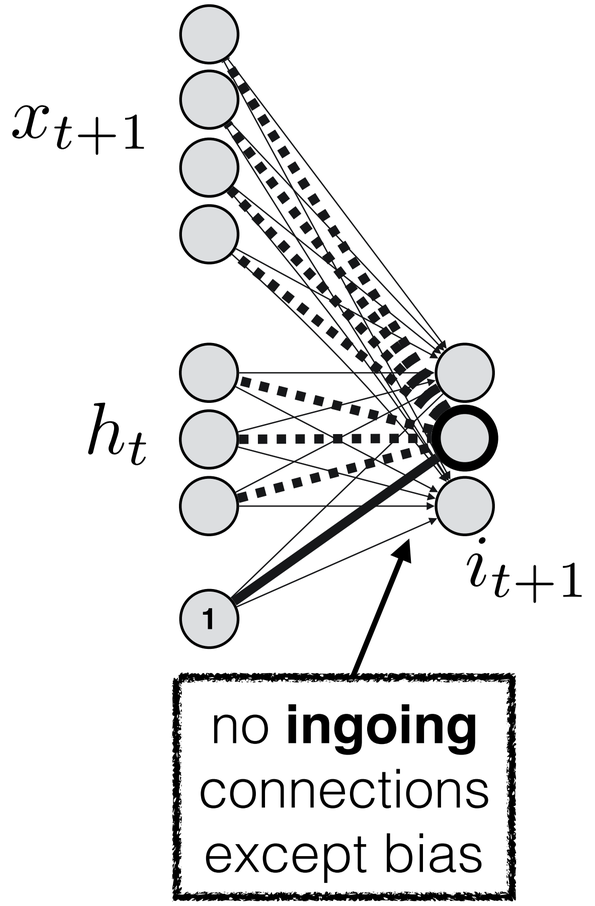


$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$



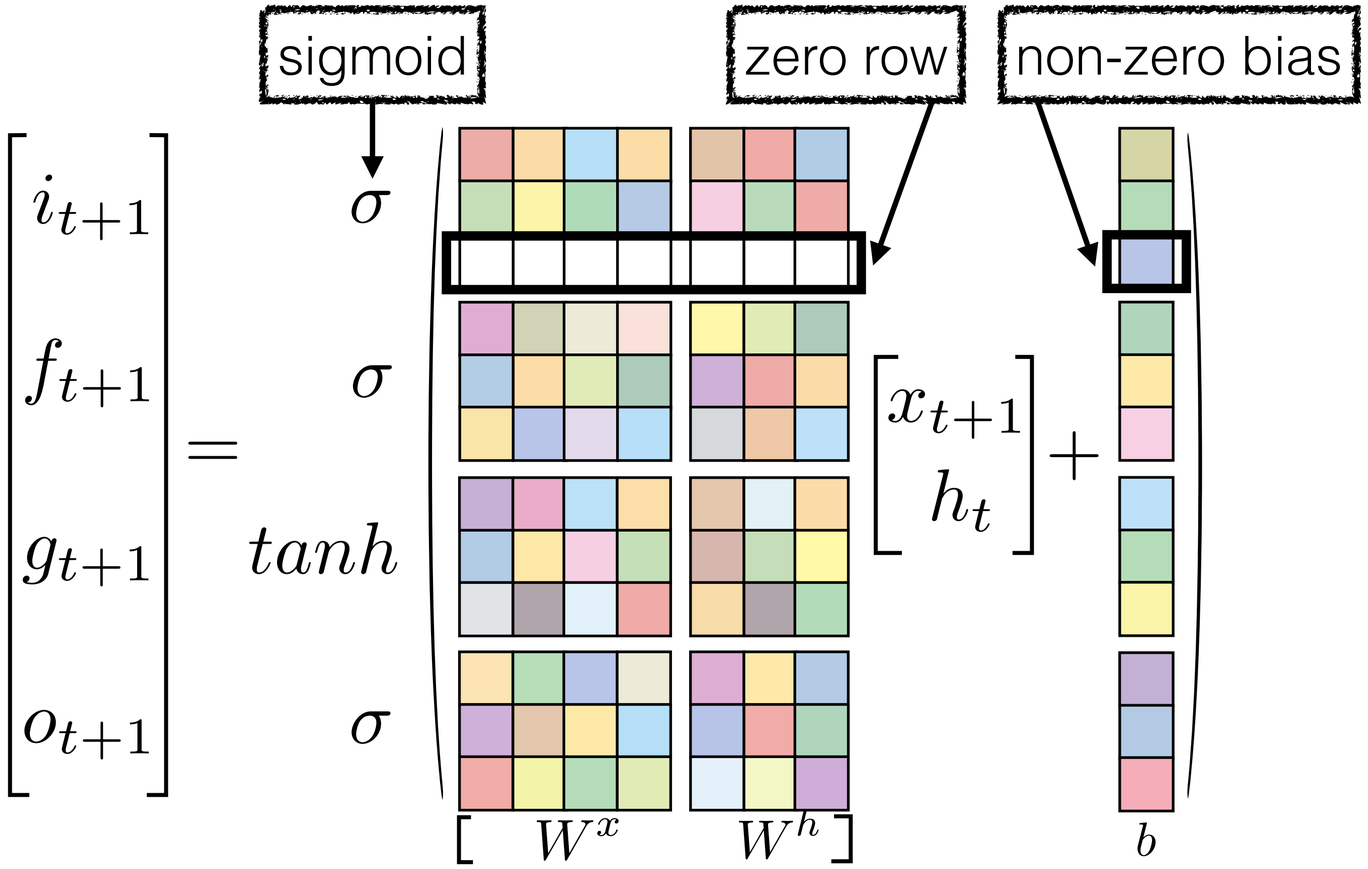
LSTM: make gates constant

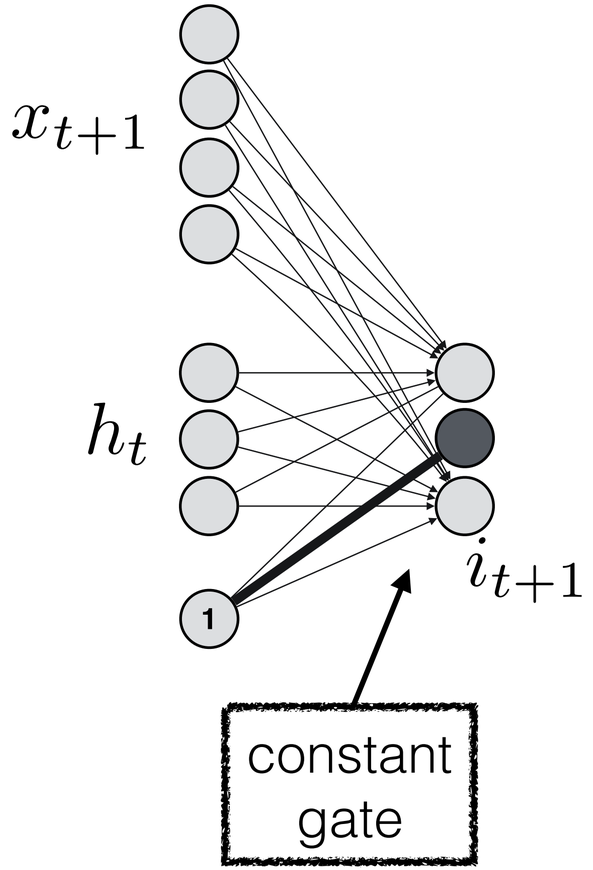




$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$

LSTM: make gates constant

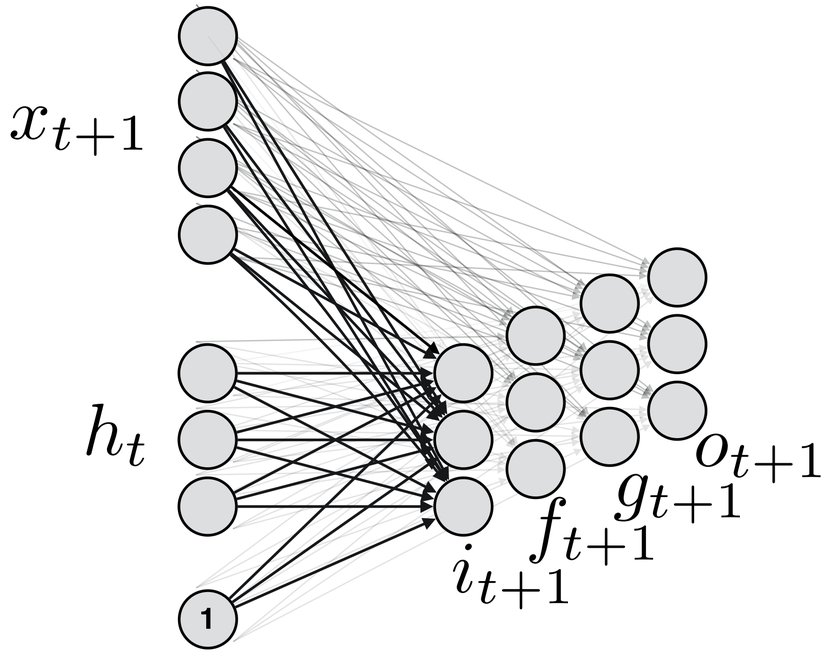




$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$
$$h_{t+1} = o_{t+1} \odot g_{t+1}$$

Putting everything together

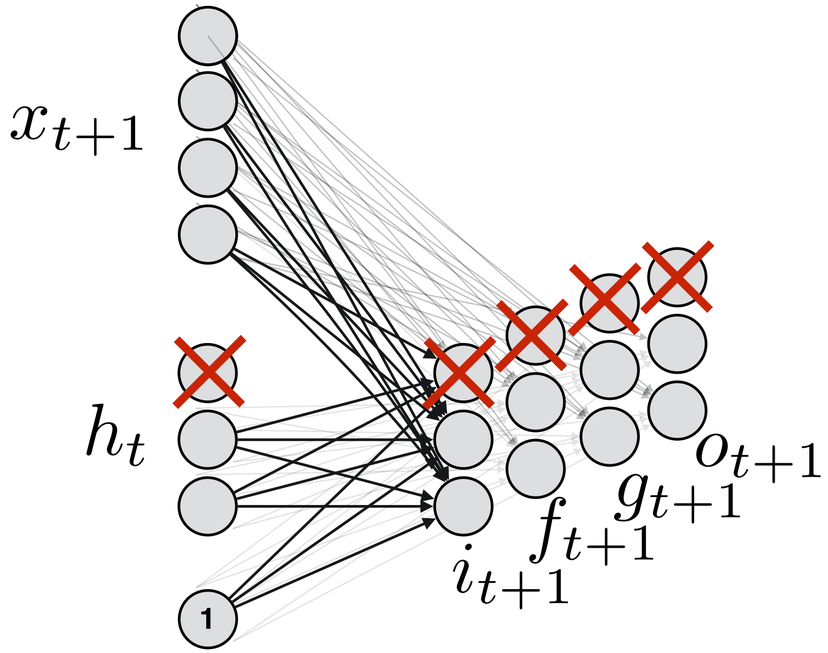
LSTM:



Levels of sparsification:

Putting everything together

LSTM:

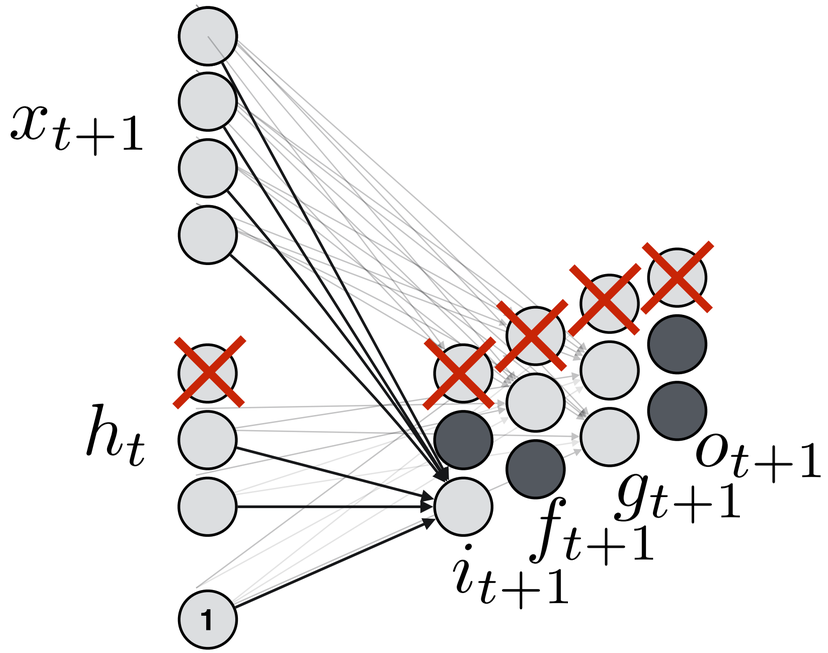


Levels of sparsification:

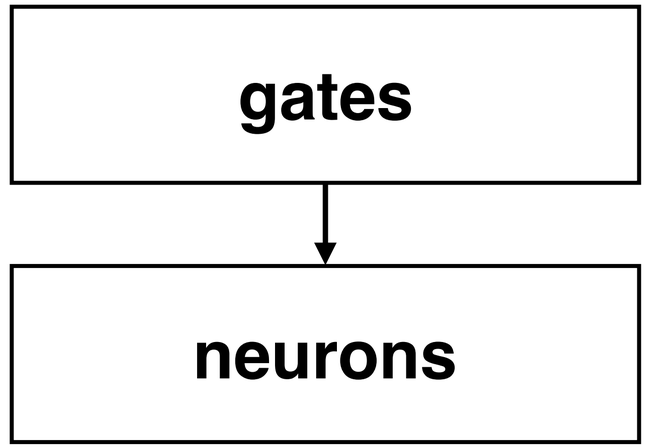
neurons

Putting everything together

LSTM:

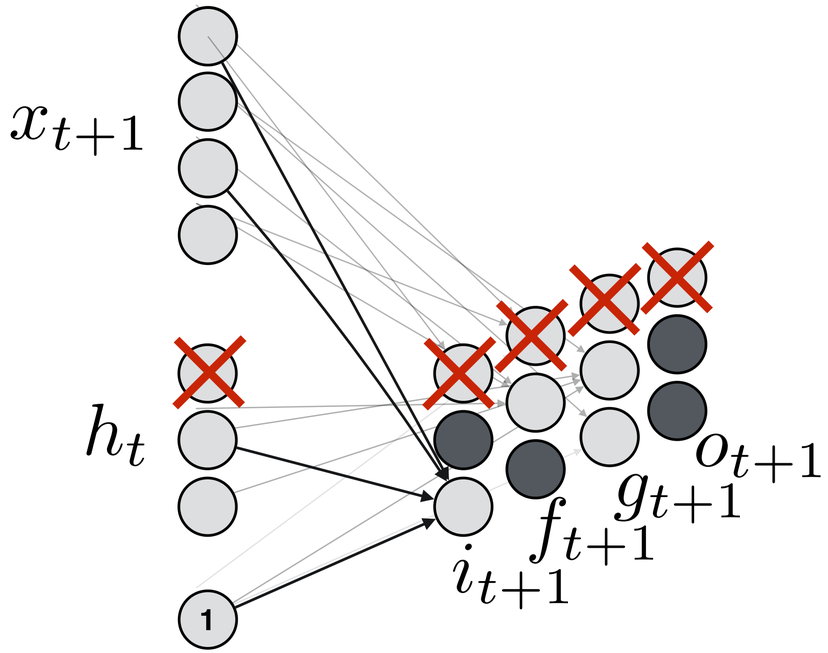


Levels of sparsification:

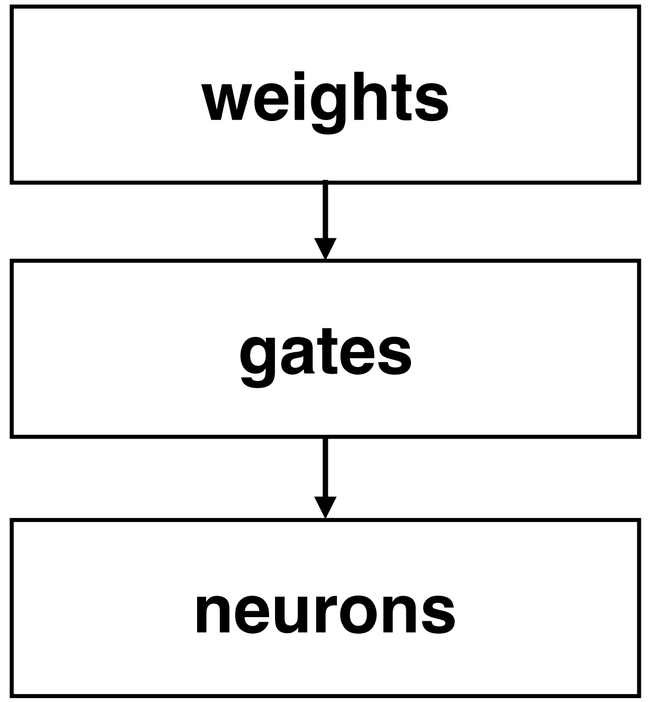


Putting everything together

LSTM:



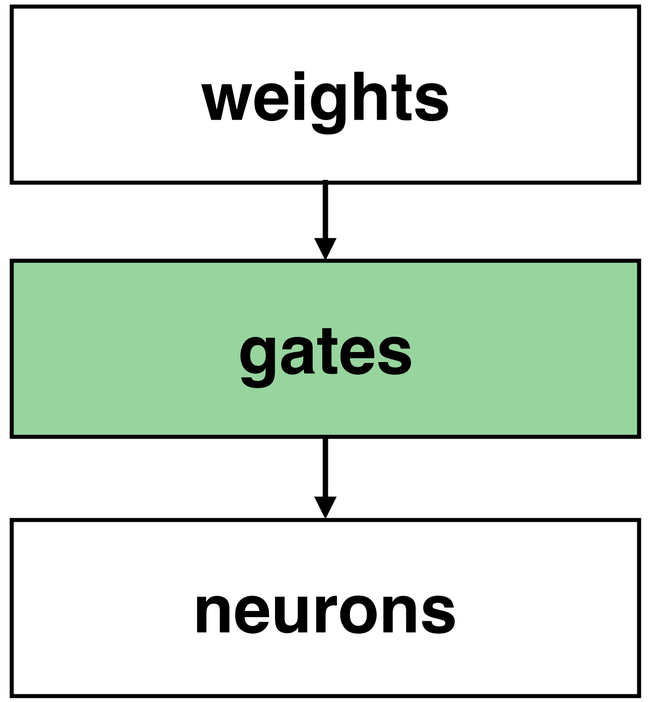
Levels of sparsification:



Putting everything together

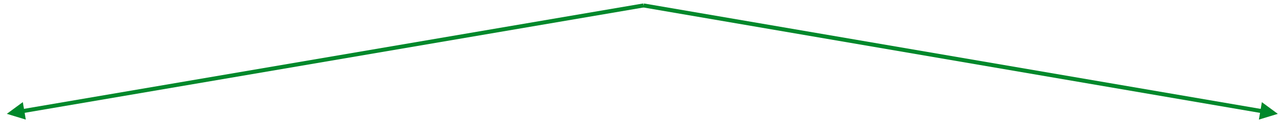
- Better neuron-wise compression
- Less computations on testing stage
- Show that LSTM uses different gates in different tasks

Levels of sparsification:



Implementation of the idea

Implementation in two sparsification paradigms



Magnitude pruning

Based on (Wen et al., 2018)

Details on the next slides

Bayesian sparsification

Based on (Chirkova et al., 2018) and (Louizos et al., 2017)

Details in the paper and poster

(Wen et al., 2018) Wen, et al. Learning intrinsic sparse structures within long short-term memory. ICLR 2018
(Chirkova et al., 2018) Chirkova et al. Bayesian sparsification of recurrent neural networks. EMNLP 2018
(Louizos et al., 2017) Bayesian compression for deep learning. NeurIPS 2017

Implementation in magnitude pruning

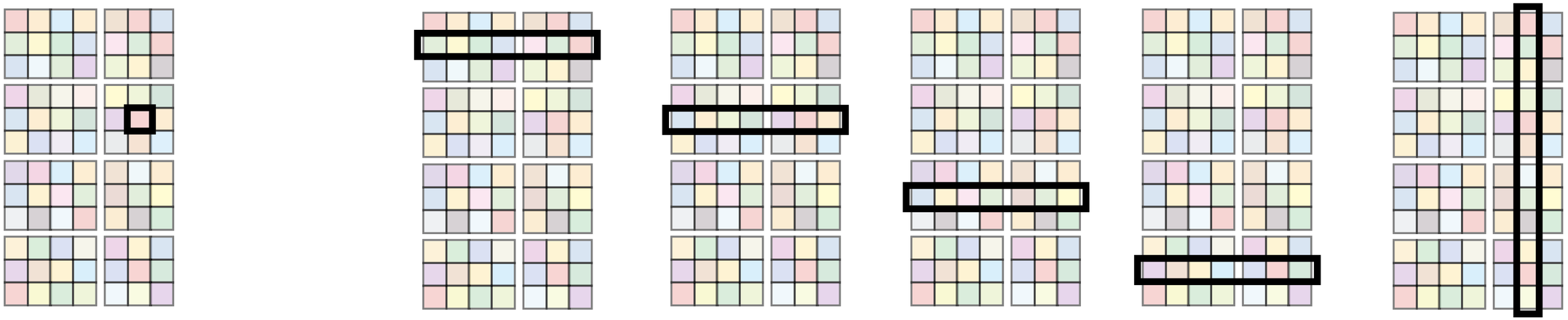
Regularization term:

$$\lambda_1 \|w\|_1 + \lambda_2 \sum_{\eta \in H} \left(\|w_{\eta,i}\|_2 + \|w_{\eta,f}\|_2 + \|w_{\eta,g}\|_2 + \|w_{\eta,o}\|_2 + \|w_{\eta,h}\|_2 \right)$$

individual weights

four gate groups

neuron group



Implementation in magnitude pruning

During training:

- Add group Lasso and Lasso regularization
- On each forward pass, zero out weights with $|w| < \tau$
- We do not mask gradients for the pruned weights

After training:

- Zero out weights with $|w| < \tau$
- Zero row $\longrightarrow$ mark the gate as constant
- Zero column $\longrightarrow$ remove the neuron

Hyperparameters:

- threshold τ
- regularization strength λ_1, λ_2

Experiments: Bayesian approach

text classification

text generation

Task	Method	Quality	Neurons	Gates
IMDb	Original	**84.1**	128	512
Accuracy %	Bayes W+N	83.98	5	12
	Bayes W+G+N	83.98	**4**	**6**
AGNews	Original	**90.6**	512	2048
Accuracy %	Bayes W+N	88.55	17	62
	Bayes W+G+N	88.41	**14**	**39**
Char PTB	Original	1.454	1000	4000
Bits-per-char	Bayes W+N	1.430	**390**	**1560**
	Bayes W+G+N	**1.425**	404	1563
Word PTB	Original	114.41	200 – 200	800 – 800
Perplexity	Bayes W+N	104.81	68 – 110	272 – 392
	Bayes W+G+N	**104.45**	**52 – 108**	**197 – 349**

Bayes W+N: Weights (Chirkova et al., 2018) + Neurons (Louizos et al., 2017)

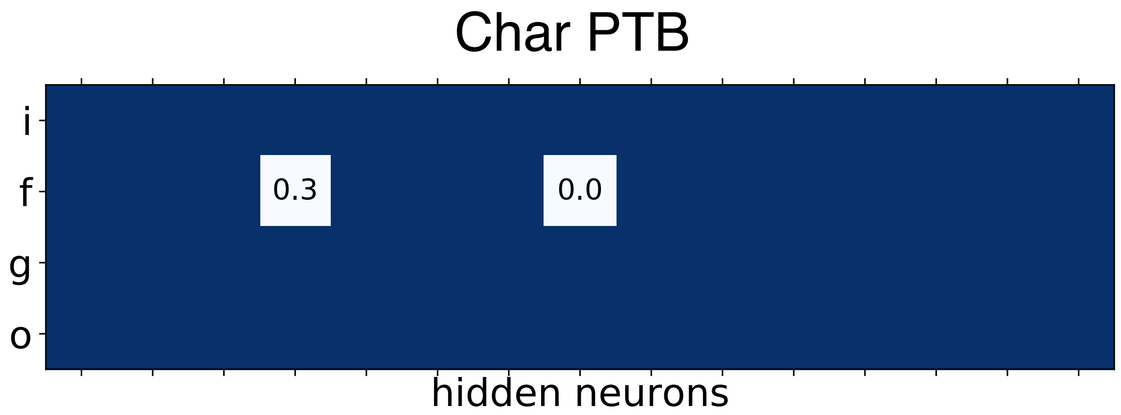
Experiments: pruning approach

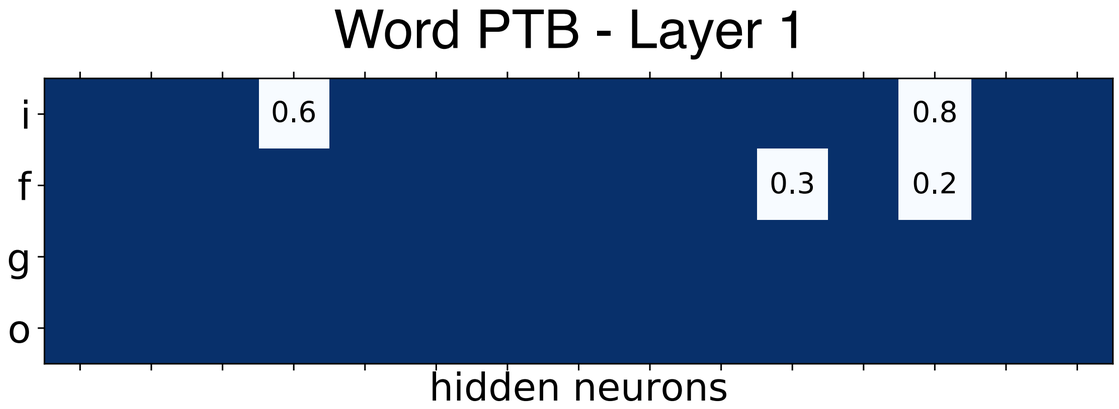
text generation

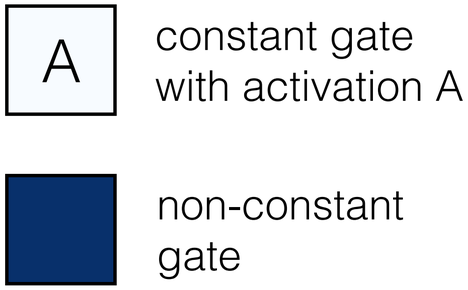
Task	Method	Quality	Neurons	Gates
Word PTB	Original	114.41	200 – 200	800 – 800
Perplexity	Prun. W+N	106.25	72 – 123	288 – 492
	Prun. W+G+N	**105.64**	**64 – 115**	**193 – 442**
Word PTB	Original	78.57	1500 – 1500	6000 – 6000
(large)	Prun. W+N	**77.62**	324 – 394	1296 – 1576
Perplexity	Prun. W+G+N	**77.82**	**252 – 394**	**881 – 1418**

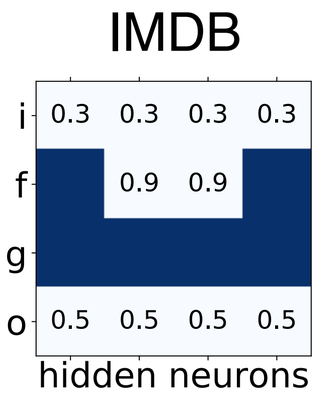
Pruning W+N: Weights + Neurons (Wen et al., 2018)

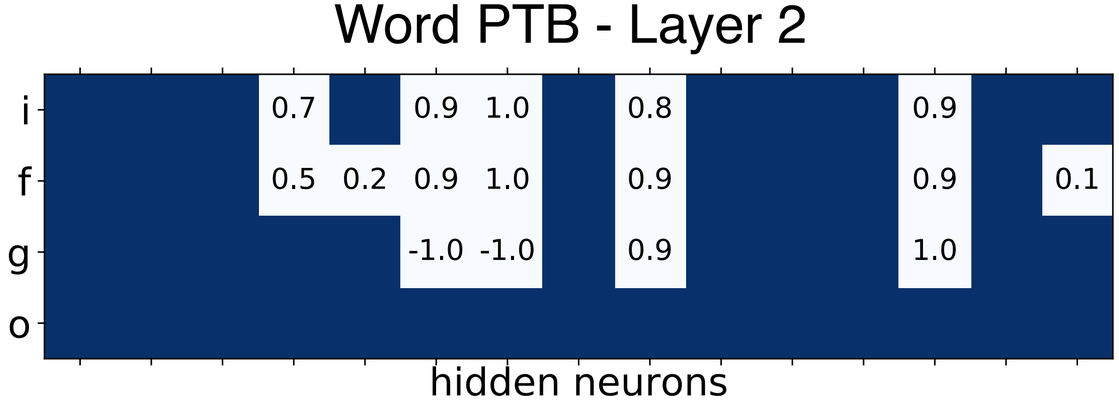
Resulting gate structures: different tasks





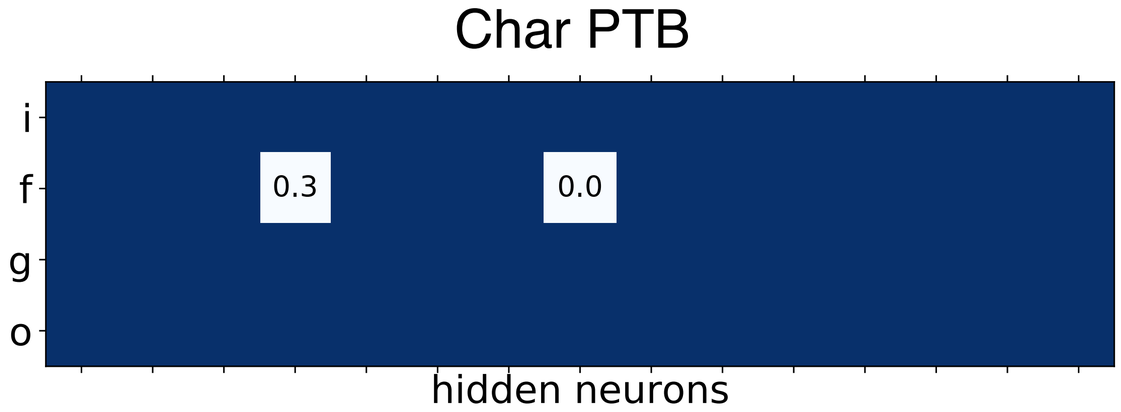


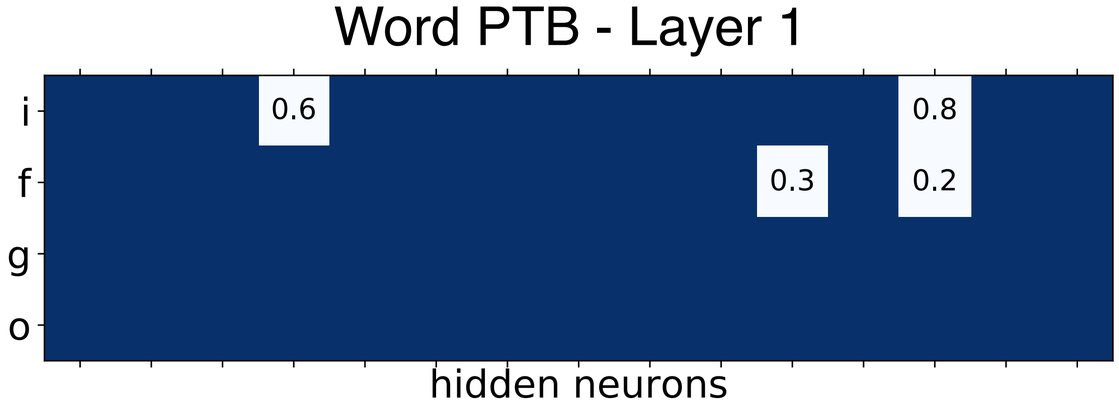




Different gates are important for different tasks

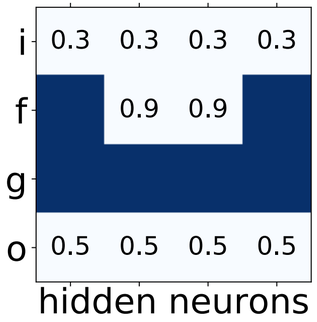
Resulting gate structures: different tasks



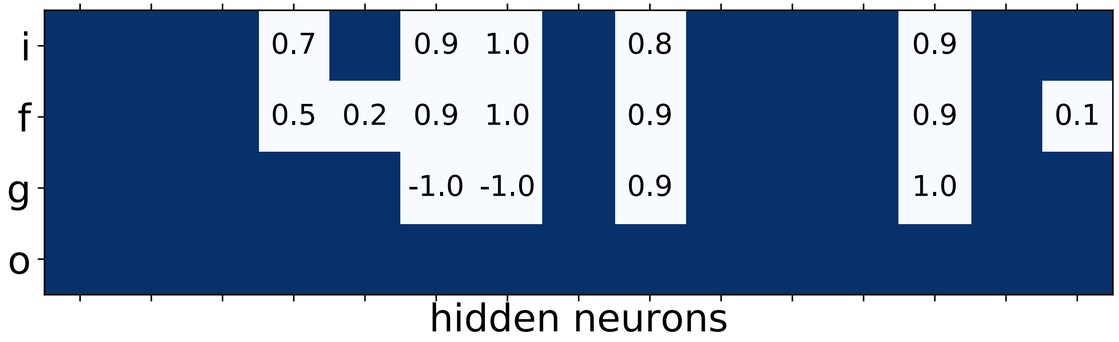


constant gate
with activation A

non-constant
gate

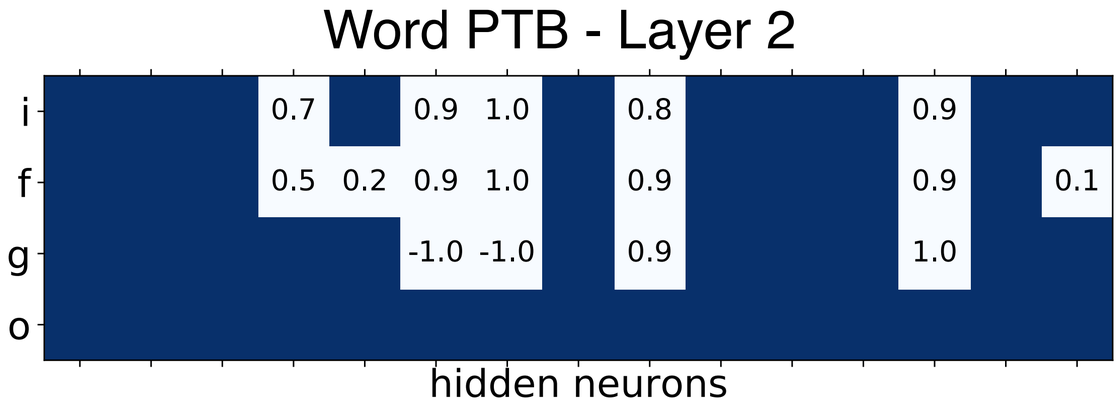


text classification



text generation

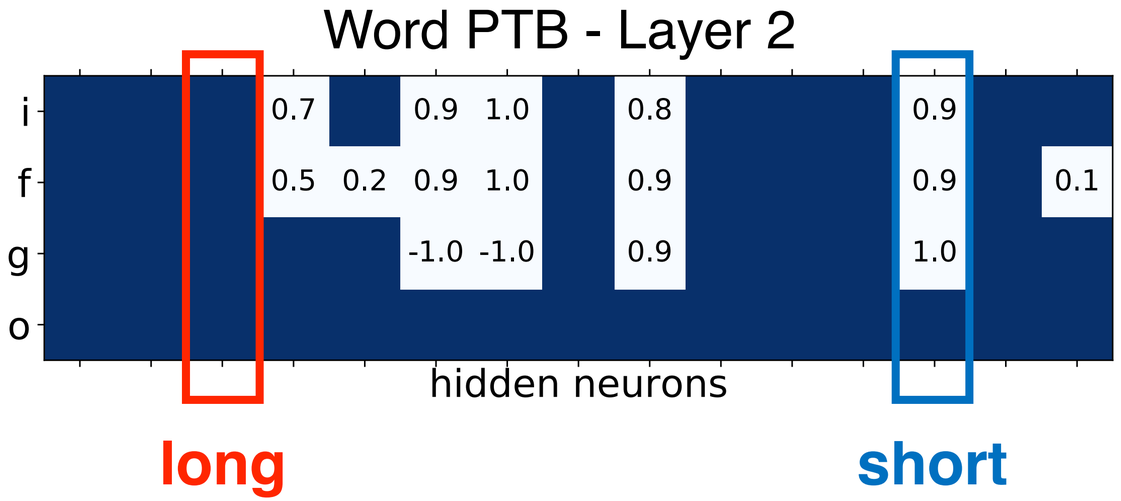
Resulting gate structures: text generation



Answer at each timestep

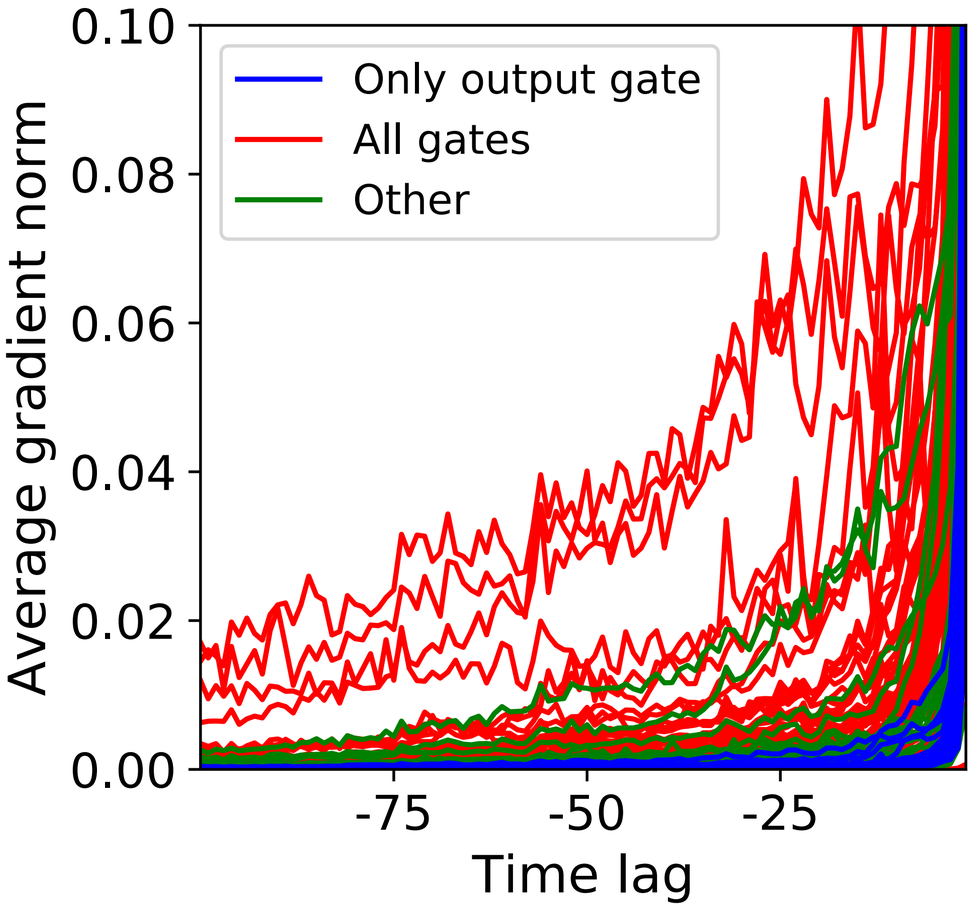
Output gates are important

Resulting gate structures: text generation

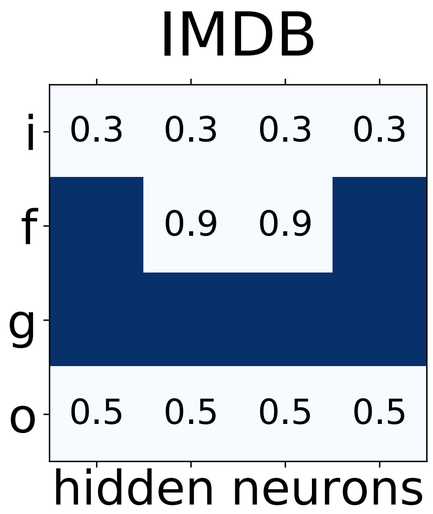


Answer at each timestep

Output gates are important
Both **long** and **short** dependencies are important

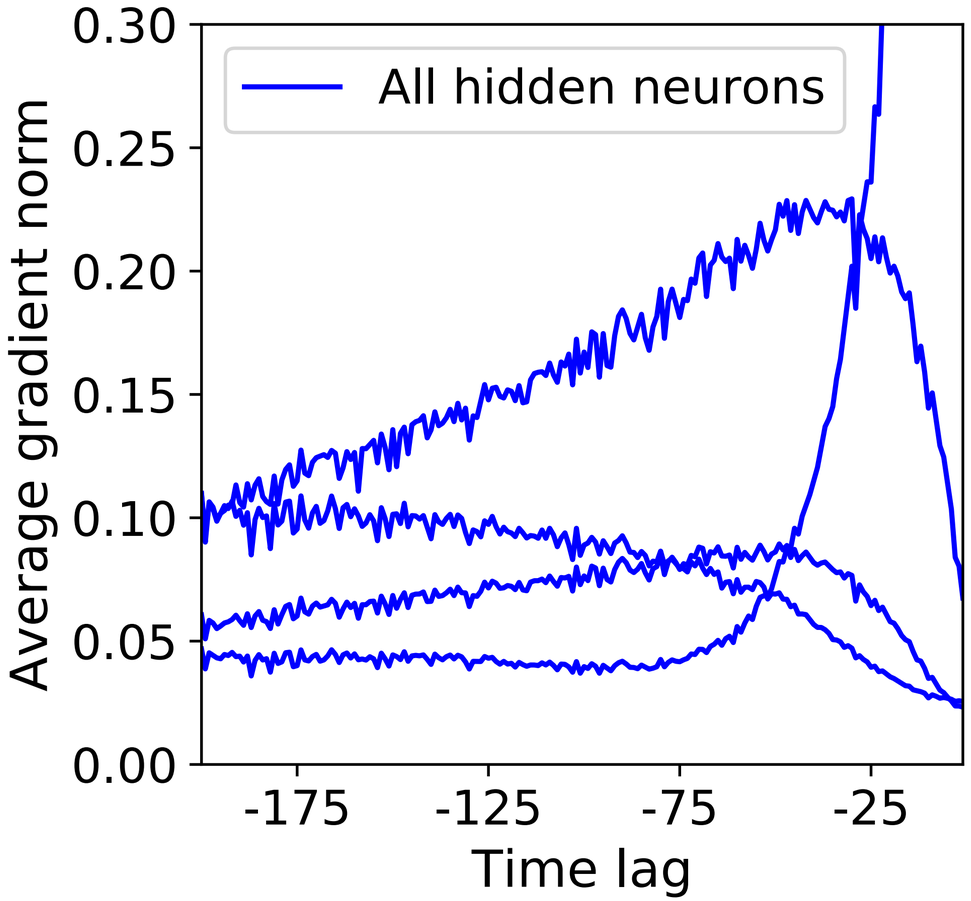


Resulting gate structures: text classification



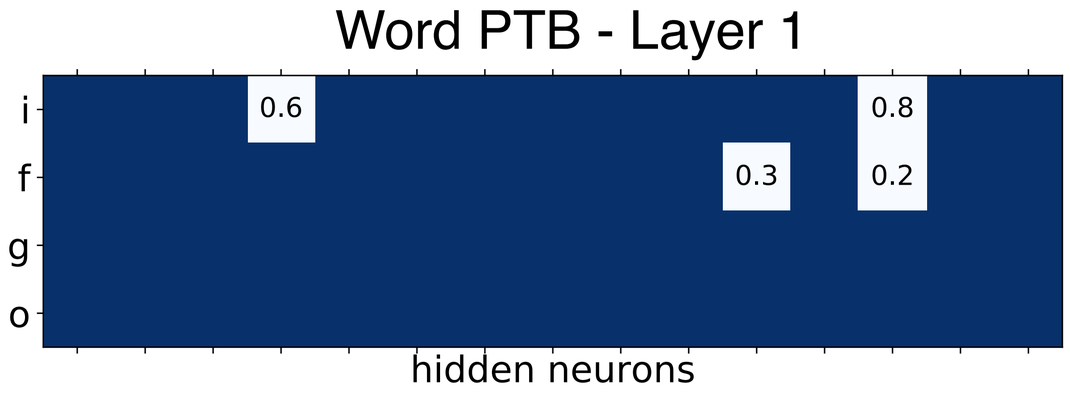
Answer at the end of the sequence

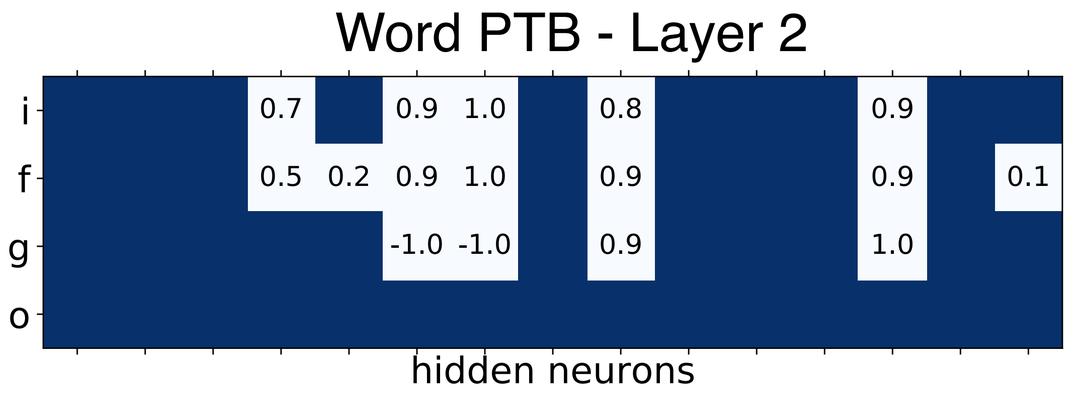
No need for output gate
Long dependencies are important



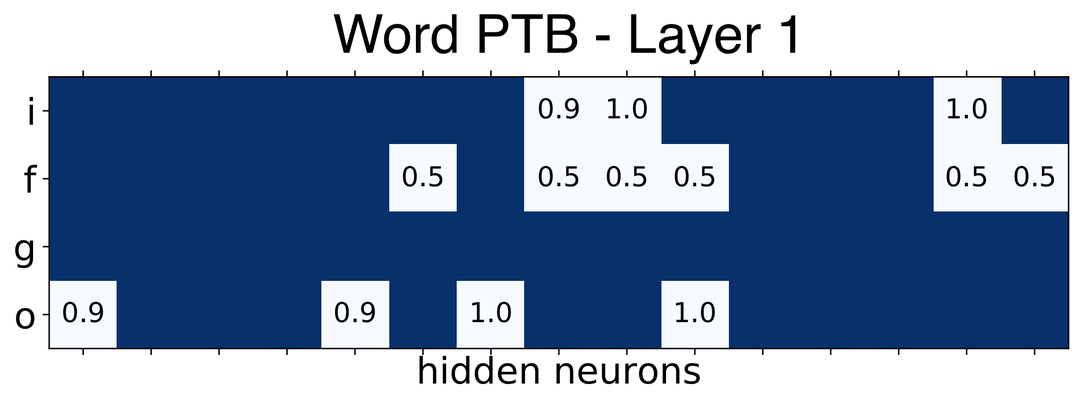
Resulting gate structures: Bayes vs pruning

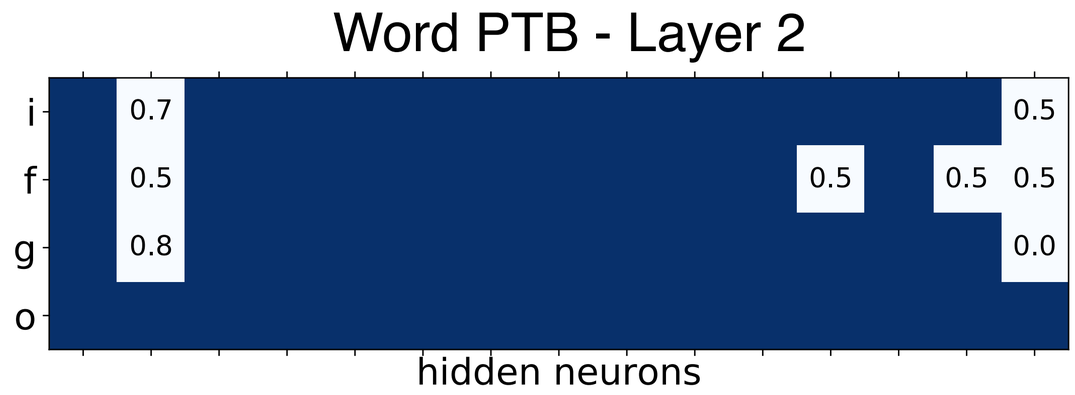
Bayesian approach





Pruning approach





Summary

Poster number:
284-ML8844

- Three levels of sparsity for gated RNNs:
  (1) sparsify weights; (2) make gates constant; (3) remove neurons

  Can be easily implemented in any sparsification framework
- Our implementations: magnitude pruning and Bayesian sparsification

- Improve compression and reveal LSTM specifics on different tasks

Contact us: elobacheva@hse.ru
nchirkova@hse.ru