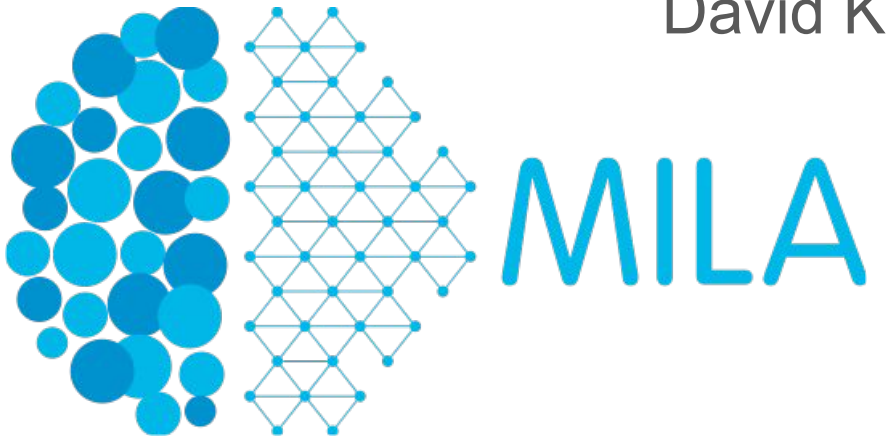
Regularizing RNNs by Stabilizing Activations

David Krueger, Roland Memisevic



Stability: a generic prior for temporal models

- Sequential representation $h := h_0, h_1, \ldots$ is *stable* if it does not grow exponentially:

$$||h_t|| \notin \Omega(e^t)$$

The Norm-stabilizer

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$

The Norm-stabilizer

Hidden states

Regularization strength hyperparameter

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$

(In Theano):

Axes: tstep, batch, num_hids (tbn)

```
hidden_norms = T.sum((hidden_states**2 + 1.e-9), axis=-1)**.5
cost += beta * T.mean((hidden_norms[1:] - hidden_norms[:-1])**2)
```

Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
- Things we’re not doing
- Experiments

Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
- Things we're not doing
- Experiments

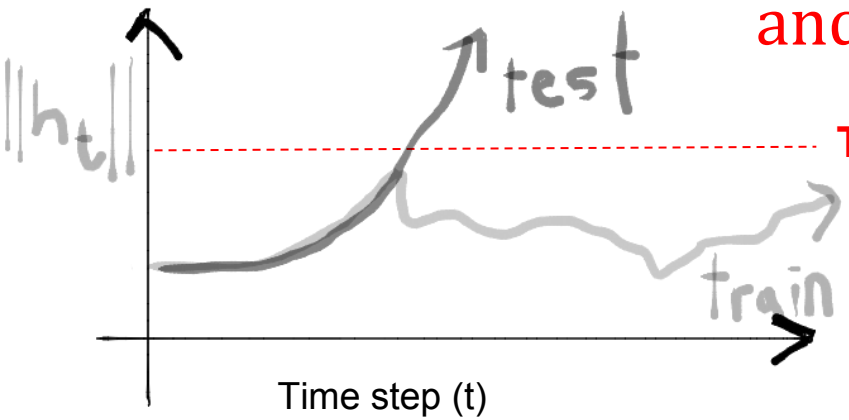
Why is stability important?

- Instability $\Rightarrow$ past observations too influential
  - Gradients explode
  - Current observations ignored outside *region of stability:* e.g. $\{h : \exists x \; s.t. \; \|\sigma(W_x x + w_h h)\| \leq \|h\|\}$ for a network with 1 hidden unit

Stability doesn't come for free!

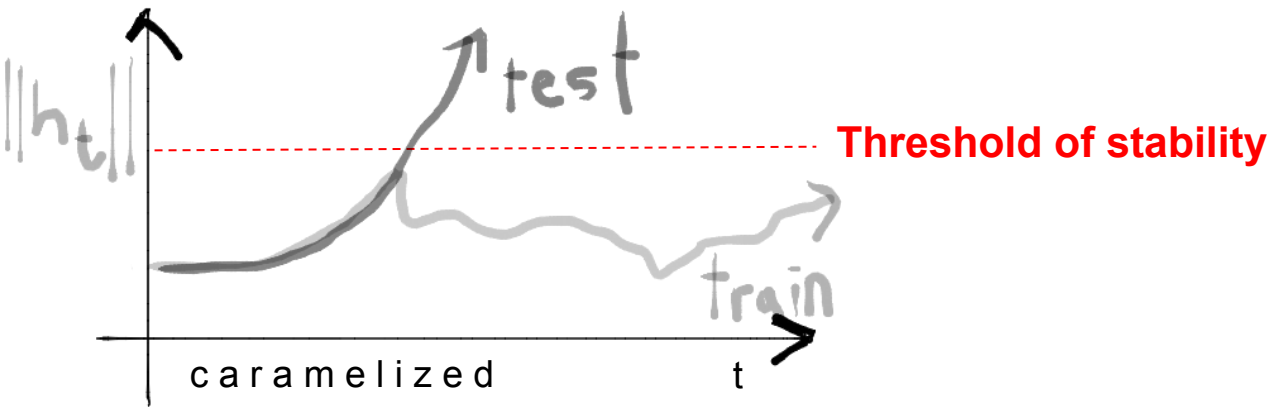
- W_h is exponentiated
- "Explosive potential" of RNN dynamics can be "defused" on training sequences…

and unleashed at test time!



Why is stability important? (example)

- Train sequence: “caramel apple”
- Test sequence: “caramelized onions”

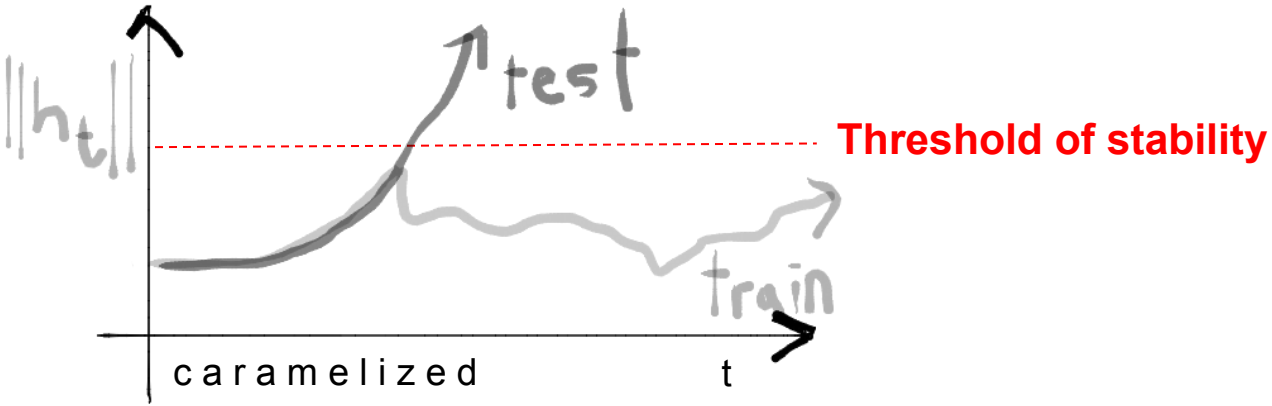


Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
- Things we're not doing
- Experiments

Why does stability help generalization?

- Explosive potential is always punished/forbidden
  - Even when defused
- Allows generalization to longer sequences



Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
  - Enforce or **encourage**
- Things we're not doing
- Experiments

Stability in RNNs

- Most RNNs enforce stability via:
  - Bounded nonlinearities
    - LSTM, GRU, Tanh-RNN
  - Unitary transition matrix
    - Unitary RNN -- Arjovsky, Shah, Bengio (concurrent work)

Stability in RNNs

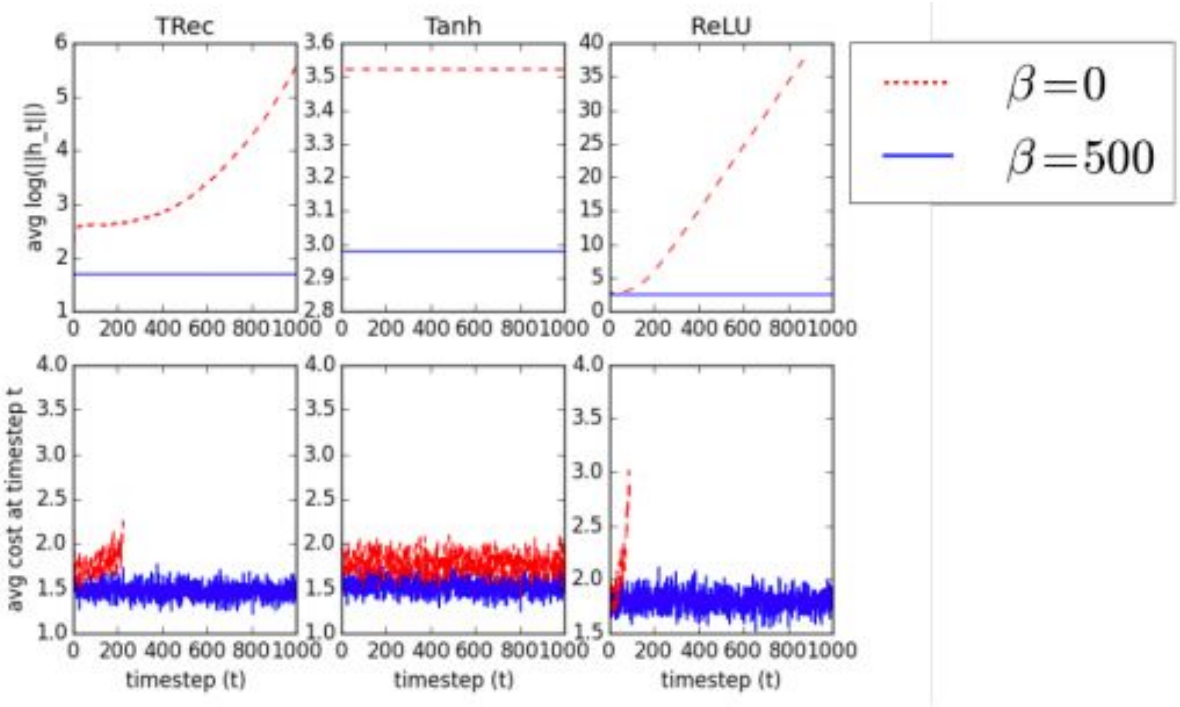
- Most RNNs enforce stability via:
  - Bounded nonlinearities
    - Disadvantage: saturation $\Rightarrow$ no gradient!
  - Unitary transition matrix
    - Disadvantage: no forgetting! (...via W_h)

Stability in RNNs

- Most RNNs enforce stability via:
  - Bounded nonlinearities
    - Disadvantage: saturation $\Rightarrow$ no gradient!
    - Use ReLU $\rightarrow$ Identity RNN -- Le, Jaitly, Hinton (2015)
      - Disadvantage: can be unstable
  - Unitary transition matrix
    - Disadvantage: no forgetting! (...via W_h)

IRNN instability

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$



Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
- Things we're not doing
- Experiments

Things we're not doing:

- Slow Feature Analysis (SFA)
- Enforcing stability
- Encouraging stability everywhere
- Encouraging orthogonal W_h

More flexibility = good?

Things we're not doing (1):

- Slow Feature Analysis (SFA)
  - $h_t = -h_{t-1}$ makes norm-stabilizer happy!

Norm Stabilizer

SFA

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2 \neq \beta \frac{1}{T} \sum_{t=1}^{T} (h_t - h_{t-1})^2$$

Things we're not doing (2):

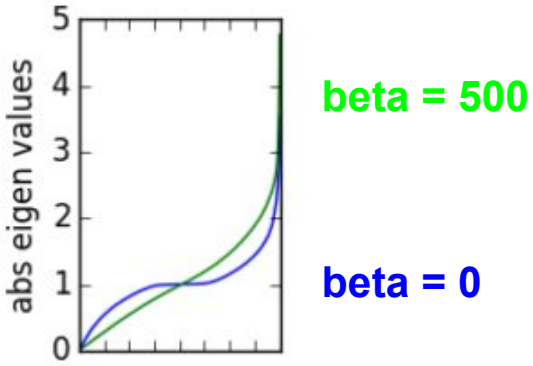
- Slow Feature Analysis (SFA)
- Enforcing stability
  - … just encouraging stability

Things we're not doing (3):

- Slow Feature Analysis (SFA)
- Enforcing stability
- Encouraging stability everywhere
  - … just around the data

Things we're not doing (4):

- Slow Feature Analysis (SFA)
- Enforcing stability
- Encouraging stability everywhere
- Encouraging orthogonal W_h
  - See sorted eigen-values →



Things we're not doing:

- Slow Feature Analysis (SFA)
- Enforcing stability
- Encouraging stability everywhere
- Encouraging orthogonal W_h

More flexibility = good?

Outline:

- Why is stability important?
- Why does it help generalization?
- How to achieve stability?
- Things we're not doing
- Experiments

Tasks:

- Character-level language modelling (next-step prediction) on Penn Treebank
- Phoneme recognition on TIMIT
- Adding problem from the original LSTM paper (Hochreiter and Schmidhuber, 1997)

IRNN Performance (Penn Treebank)

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$

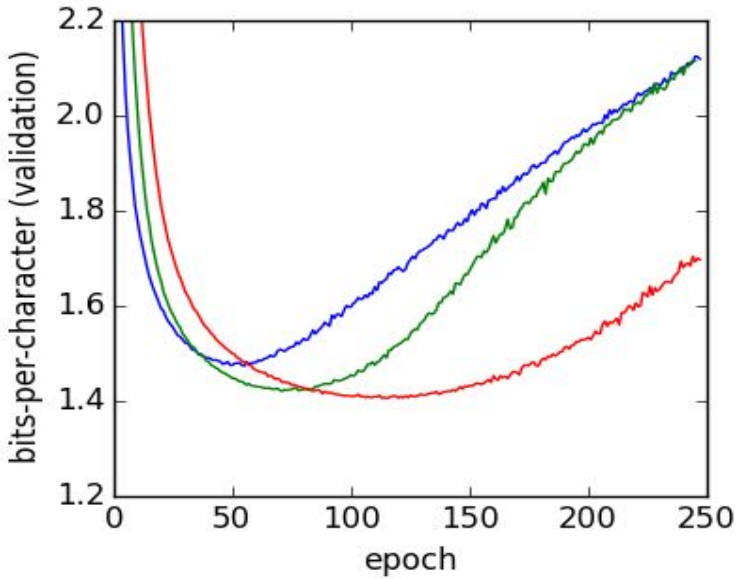
	$lr = .002, gc = 1$	$lr = .002$	$lr = .0002, gc = 1$	$lr = .0002$
tanh, $\beta = 0$	1.71	**1.55**	2.15	2.15
tanh, $\beta = 500$	1.57	2.70	1.79	1.80
ReLU, $\beta = 0$	1.78	1.69	1.93	1.93
ReLU, $\beta = 500$	1.74	1.73	**1.65**	2.04
TRec, $\beta = 0$	1.62	1.63	1.95	1.88
TRec, $\beta = 500$	**1.48**	1.49	1.56	1.56

Best results

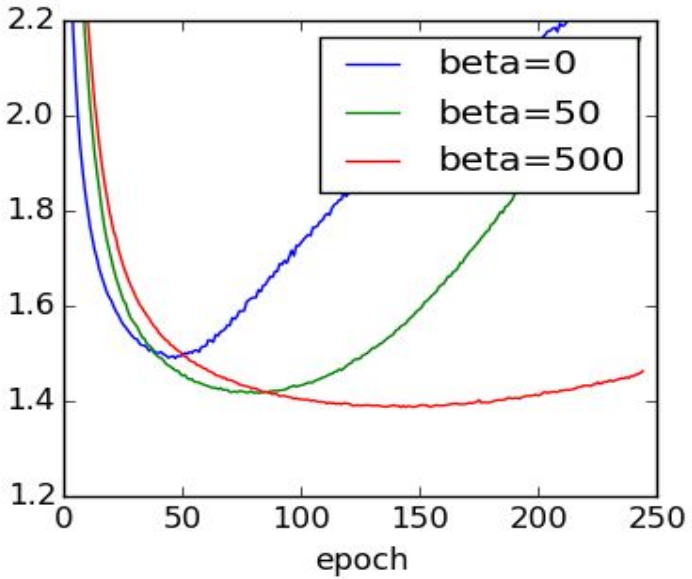
LSTM Performance (Penn Treebank)

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$

Penalize: Hidden State



Memory Cell

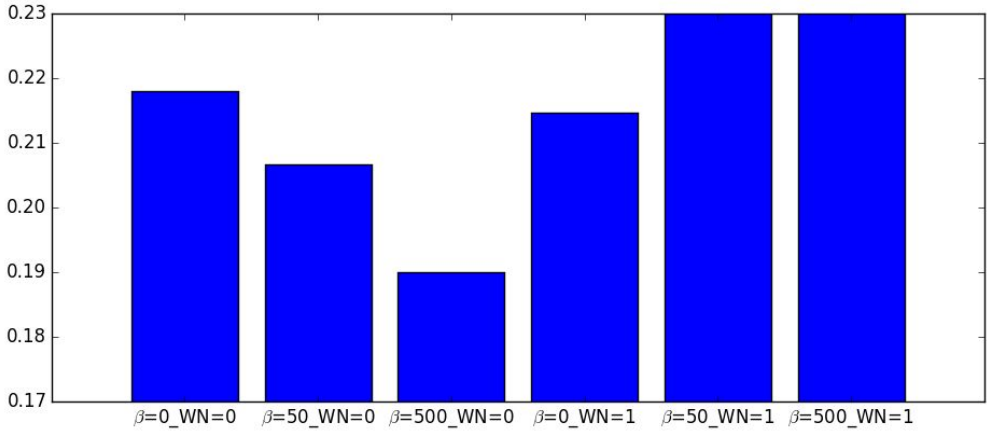


LSTM Performance (TIMIT)

- Use CTC, no beam search
- Average of 5 experiments

$$\beta \frac{1}{T} \sum_{t=1}^{T} (\|h_t\|_2 - \|h_{t-1}\|_2)^2$$

PER (phoneme error rate)



|-----Use weight noise-----|

Alternative Cost Functions

- Norm-stabilizer performed best
- Worth investigating other approaches to stability

Table 3: Performance (bits-per-character) of various costs designed to encourage norm stability.

	$(\Delta h_t)^2$	$(\Delta \|h_t\|_2)^2$	$(\frac{\Delta \|h_t\|_2}{\|h_t\|_2})^2$	$(\Delta \|h_t\|_1)^2$	$(\|h\|_2 - 5)^2$	$(\|h_0\|_2 - \|h_T\|_2)^2$
$\beta = 50$	1.84		1.60	2.96	1.49	3.81
$\beta = 500$	2.19	1.48	1.50	3.18	1.50	1.54

Thank you!

Any Questions?

LSTM hidden norms: