tRustNN: Building trust in Recurrent Neural Networks through data-driven, human-interpretable visualizations

Ioannis Chalkiadakis
Supervisor: Mike Chantler

Edinburgh Center for Robotics

ic14@hw.ac.uk

July 5, 2017

Motivation

- RNN successes in robotics:
  - Learning by demonstration, [1] (social robotics)
  - Object tracking, [2]
  - Object tracking for autonomous vehicles, [3], [4]
- Largely vague/unexplored
- Can we trust them for critical applications?

Research questions I

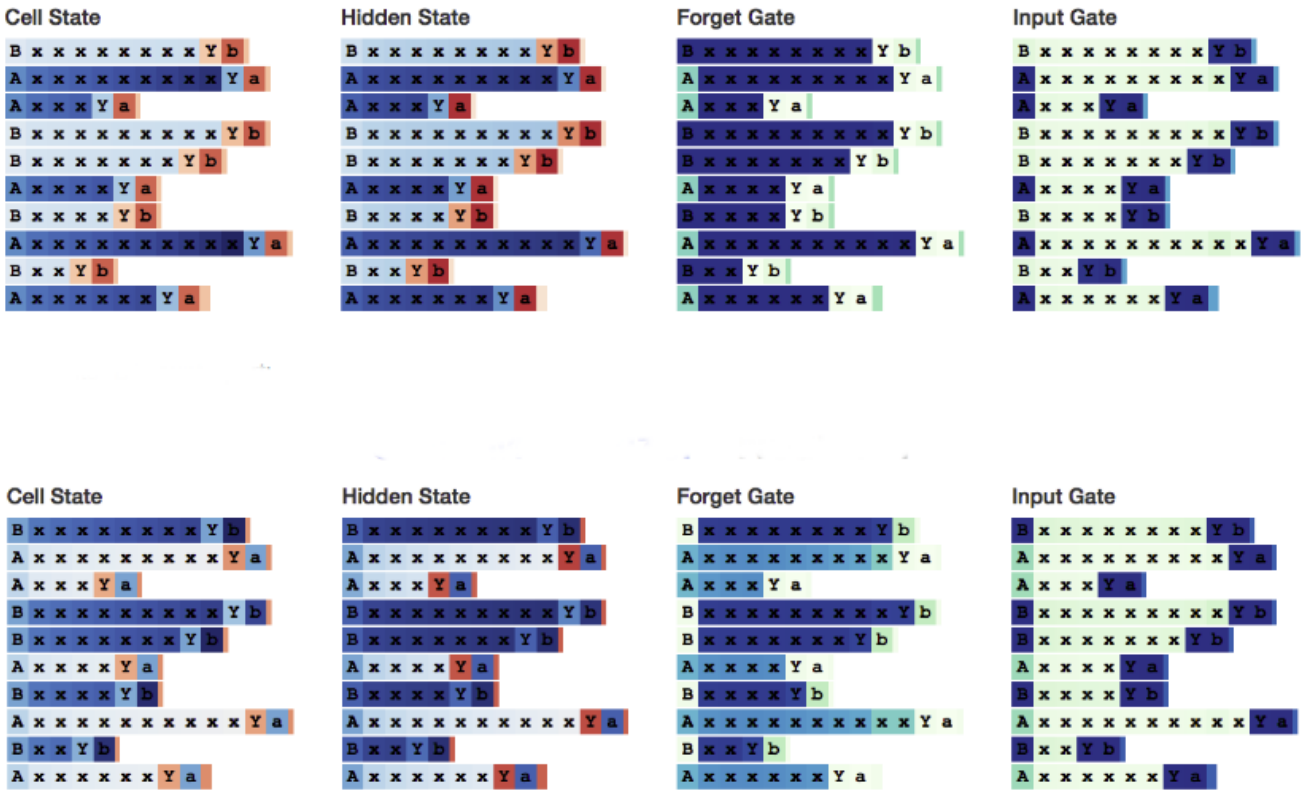
How can we employ visualization methods to interpret the strengths and weaknesses of recurrent neural networks, thus helping (non)experts trust and improve them?

- Research goal: Given a trained network, we want to provide the developer (end-user) with a conceptual model of the task and how it reflects on the network
  - alleviate user cognitive load [5]
  - RNN as state machine [6]

$\rightarrow$ explore, interpret and ultimately improve the network

RNN as state machine I

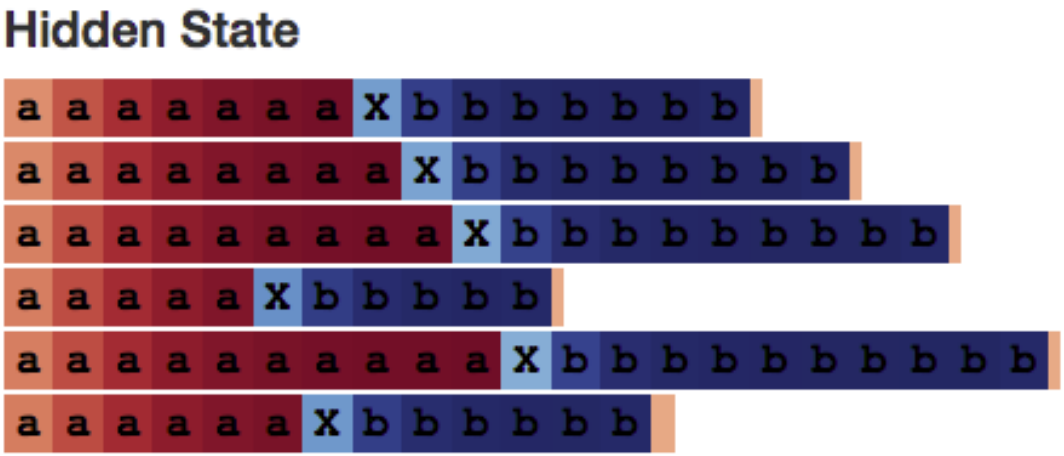
Figure: Remembering states



From: *http://blog.echen.me/2017/05/30/exploring-lstms/*

RNN as state machine II

Figure: Counting

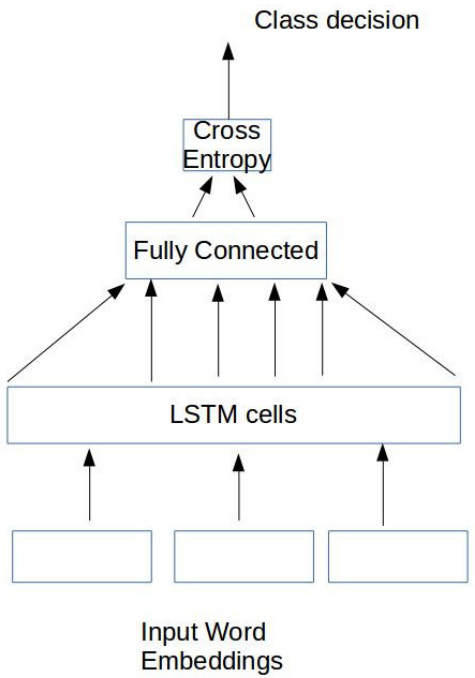


From: *http://blog.echen.me/2017/05/30/exploring-lstms/*

Project details I

- Task domain: Natural Language, emotion classification: well-explored, easier to visualize that other robotics-related applications
- Use an LSTM-RNN to classify movie reviews as positive or negative

Figure: LSTM-RNN for sentiment classification



Project details II

- RNN learn temporal sequences in data, can infer by considering multiple steps back in time
- LSTM cell: overcome common training issues: exploding/vanishing gradients, instability
- cells = state + 'gates'

Status report - Proposal goals and WPs

- WP1: Client-server application allowing selection of input features and loading trained network
- WP2: Visualization of input space - neural net, clustering
- WP3: Partial interactivity of visualizations
- WP4: Exploration of net's decision (LRP, [7]).
- WP5 - WP6: user experiments + evaluation (TODO)

Next steps - Open questions I

- Input space
- RNN/neuron space
- Interpretability

Formulate a task for users and that will allow us to evaluate the interpretability of the network's visualizations.

Review

- RNN as black box
- Conceptual model linking task to RNN
- Help developers spot weaknesses and improve RNN
- Help users understand and trust RNN

Key References I


Rouhollah Rahmatizadeh, Pooya Abolghasemi, and Ladislau Bölöni,
"Learning manipulation trajectories using recurrent neural networks,"
CoRR, vol. abs/1603.03833, 2016.


Peter Ondruska and Ingmar Posner,
"Deep tracking: Seeing beyond seeing using recurrent neural networks,"
CoRR, vol. abs/1602.00991, 2016.


Julie Dequaire, Dushyant Rao, Peter Ondruska, Dominic Zeng Wang, and Ingmar Posner,
"Deep tracking on the move: Learning to track the world from a moving vehicle using recurrent neural networks,"
CoRR, vol. abs/1609.09365, 2016.

Key References II

Peter Ondruska, Julie Dequaire, Dominic Zeng Wang, and Ingmar Posner,
"End-to-end tracking and semantic segmentation using recurrent neural networks,"
CoRR, vol. abs/1604.05091, 2016.

Mary Hegarty,
"The cognitive science of visual-spatial displays: Implications for design,"
topiCS, vol. 3, no. 3, pp. 446–474, 2011.

K. Arai and R. Nakano,
"Stable behavior in a recurrent neural network for a finite state machine,"
Neural Netw., vol. 13, no. 6, pp. 667–680, July 2000.

Key References III

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek,
"Explaining recurrent neural network predictions in sentiment analysis,"
arXiv, , no. 1706.07206, 2017.

Backup I

- LSTM
  - Cell state == long-term memory
  - Cell output == short-term memory
  - Forget gate : controls what we keep in long-term memory
  - Input gate : how does input at timestep t affect long-term memory
  - Output gate : what will be passed on to next cells
- Open q's
  - Which neurons are activated the most for high-relevance inputs? Do they form a meaningful cluster? Under what conditions would they form one (cluster number, relevance threshold)?
  - How far back in time (word sequence) does the network look? What is this depth determined by?
  - Could we come up with a more effective/interpretable way of distributing relevance over dimensions in LRP?

Backup II

- Show clustering in embedding input space on the LRP-wordcloud + associate clusters with neuron space - can we identify cluster-states in neuron space?
- Does the order of input words matter or does the network self-learn an ordering? (current order is mainly dictated by word2vec embeddings)
- What happens if we add an extra fully connected layer before LSTM? Could that make the network learn a different final embedding, or encode a different order of words?