Interpretable RNN

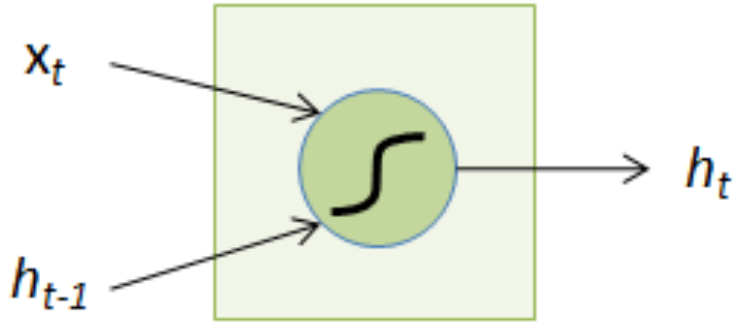
Hamed Benazha

What is a sequence ?

- A lot of things can be considered as sequences
- Time series
- Natural Language, speech
- We can even convert non sequences to sequences. For example an image can be converted to a sequence of pixels

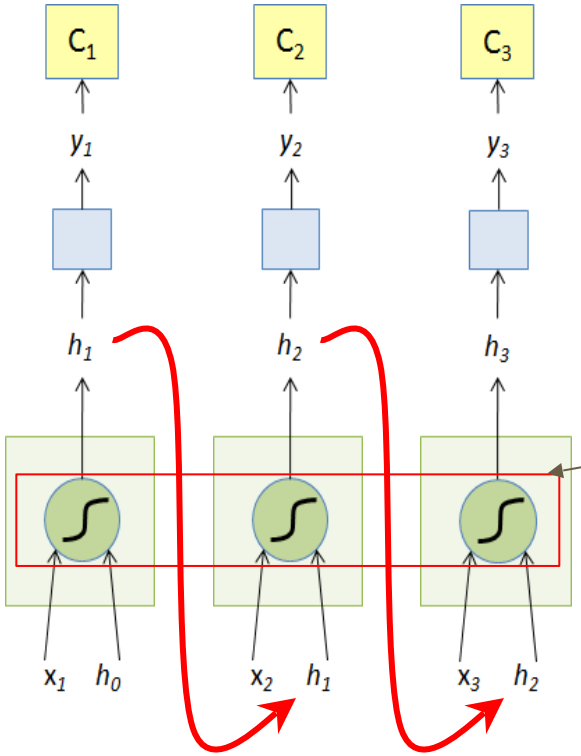
RNN

- RNN can model sequence with variable length
- They are also efficient because weight are shared at each time-steps



$$h_t = f(W_h h_{t-1} + W_x x_t)$$

Unfolded RNN



$$h_t = f(W_h h_{t-1} + W_x x_t)$$

Weights shared over time!

Discrete RNN for interpretability purpose

- RNN’s transition function looks like continuous version of DFA’s transition function
- One way to make a RNN model inherently interpretable is to discretize its transition function
- A discrete RNN is equivalent to a discrete finite automata

Vector Quantization

- Vector Quantization introduce a discrete structure inside a neural network
- The codebook is a list of vectors associated with an index
- Input are projected to the closest vector in the codebook
- Mathematically, this is written
- $k = argmin_i ||x - e_i||$
- $y = e_k$
- Where x is the input, e_i is the ith codebook vector, and y the quantized vector

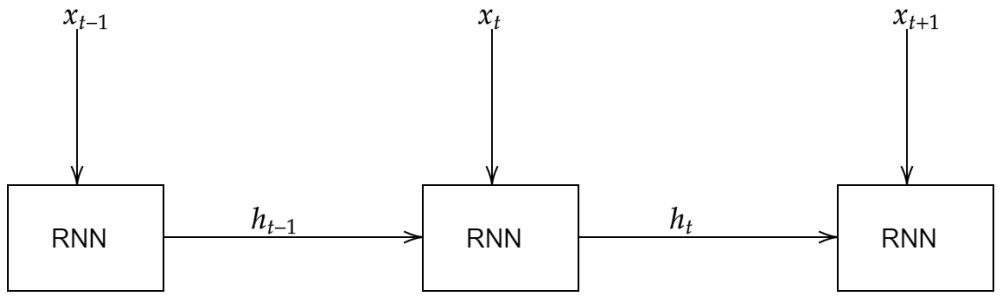
Training the VQ

- codebook vectors are learned via gradient descent
- To have an optimal quantization, we want to align inputs to codebook
- The problem is bidirectional. The VQ's input should align with the codebook and vice-versa
- $L_{VQ} = ||sg[x] - e||_2^2 + ||x - sg[e]||_2^2$
- Where sg is the stop gradient
- The first term is the codebook alignement, the second term is the input alignment

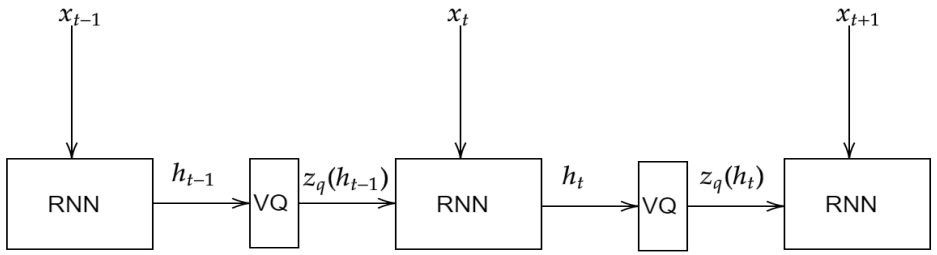
VQ-RNN

- Using the same concept, we want here to use the VQ in the RNN to discretize the hidden state space
- We can choose the number of vectors in the codebook and thus the arity of the hidden space
- A discrete RNN is inherently a Discrete Finite Automata

Vanilla RNN



VQ-RNN

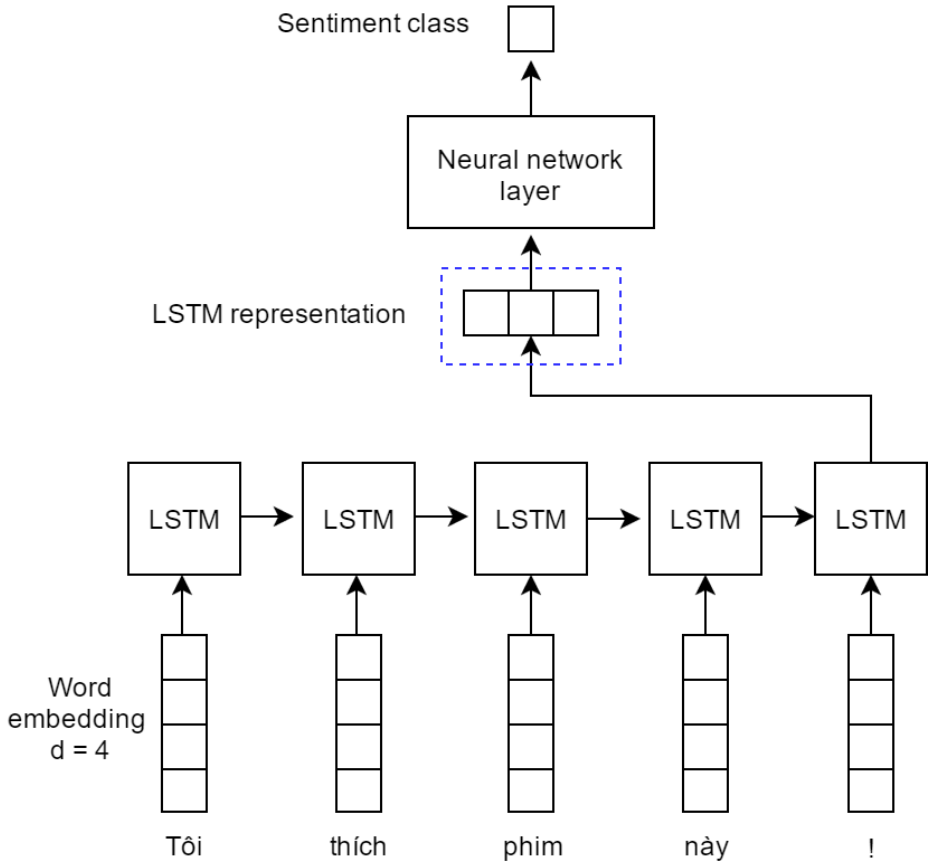


Equivalence with a DFA

- VQ-RNN is a DFA if the input is discrete (it's the case with NLP)
- We denote H the discrete space of hidden states
- We denote Σ the discrete space of the inputs
- We recall that a DFA is a 5-tuple $(Q,\ \Sigma,\ \delta,\ q_0,\ F)$
- Our VQ-RNN cell is now a function : $H \times \Sigma \rightarrow H$
- The associated DFA is then $(H,\ \Sigma,\ VQ,\ \overrightarrow{0},\ H)$

Application to interpretability

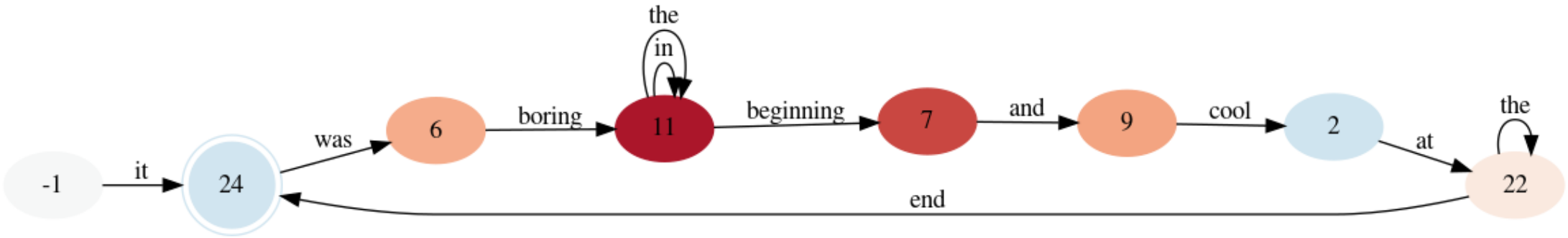
- A RNN for classification can be split up in two part
- The feature extraction part done by the RNN
- And the classification part done by a FC NN
- Usually only the last time-step is used to do the classification



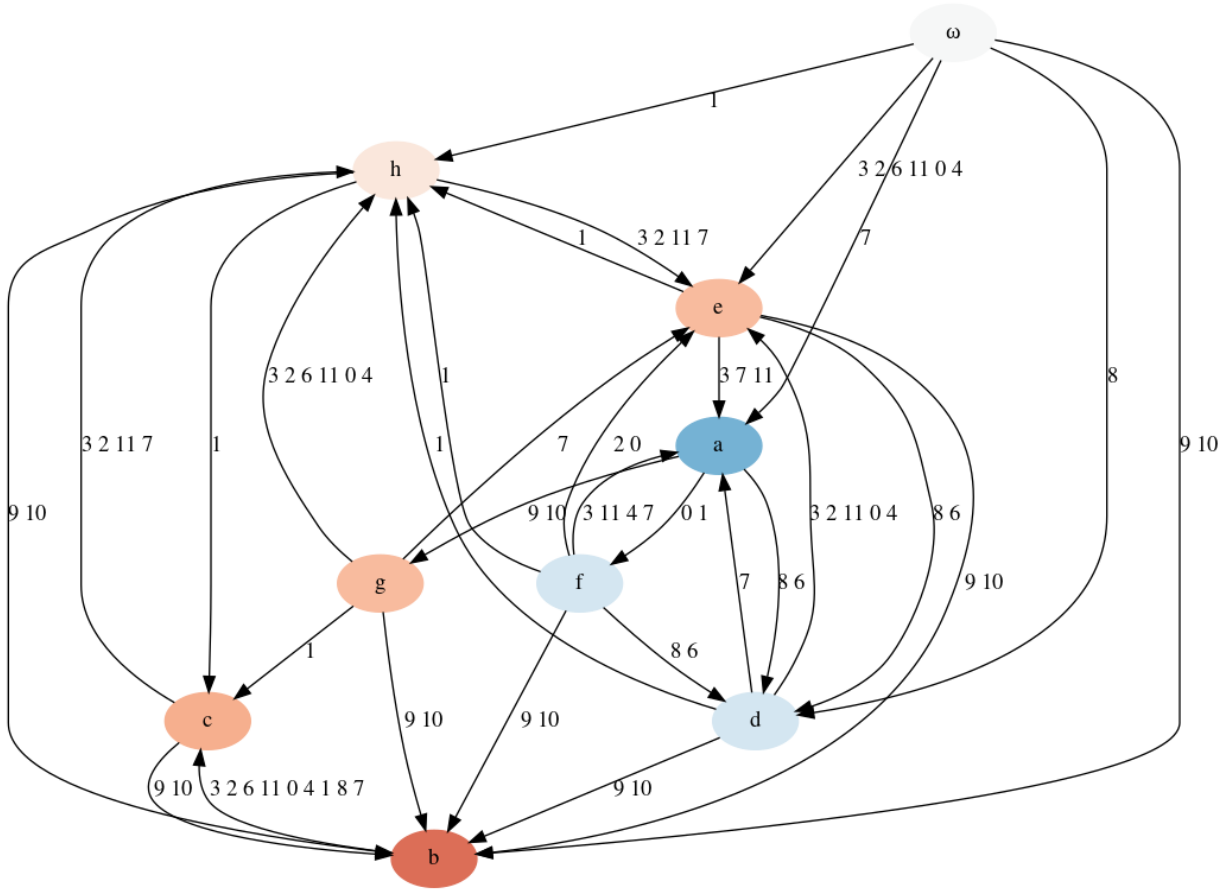
Application to interpretability

- The classification part is $c: H \rightarrow [0,1]$
- It can be seen as a confidence on a hidden state.
- If we apply this function to the hidden state, we can get a confidence score on each state
- By analysing the state transition, we can thus analyse how the confidence of the network change through the sentence

Local explanation



Global explanation



Global interpretability

- As we seen, the get a local interpretability/explainability
- We can also get a global explicability (without the interpretability), but the DFA will be huge
- The VQ-RNN can be applied to a wide range of problem, not only binary classification, it can be applied to multi-label classification, multiclass, seq2seq…
- It’s also not limited to 1 layer deep RNN, but it can be generalized to very deep RNN

Thank you