LEARNING LOGIC PROGRAM REPRESENTATION FOR DELAYED SYSTEMS WITH LIMITED TRAINING DATA

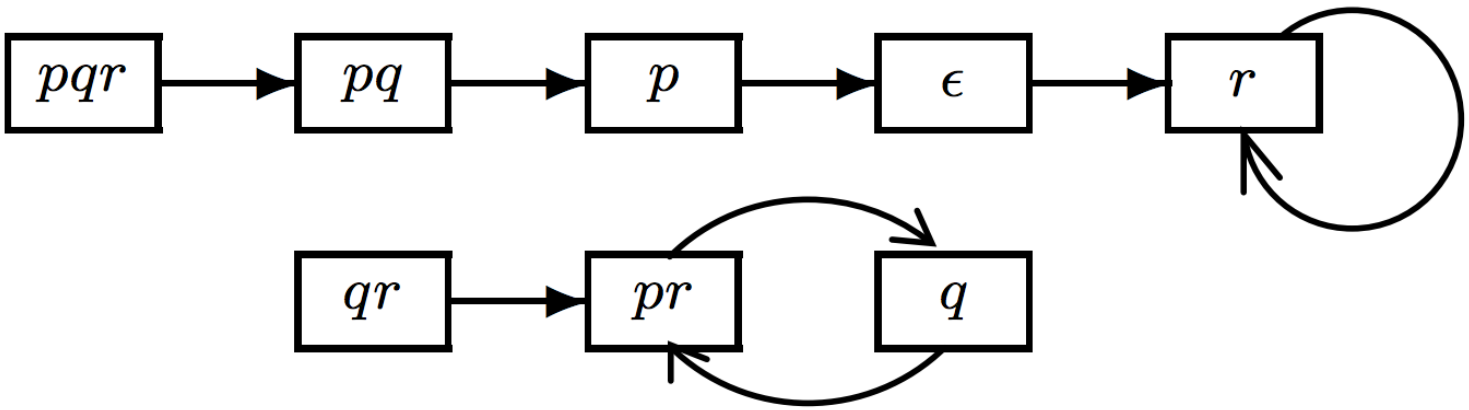
Yin Jun Phua, Tony Ribeiro, Sophie Tourret, and Katsumi Inoue

Tokyo Institute of Technology, Japan

National Institute of Informatics, Japan

Laboratoire des Sciences du Numérique de Nantes, France

LFIT



$$p \leftarrow q$$

$$q \leftarrow p \wedge r$$

$$r \leftarrow \neg p$$

[Inoue *et al.* 2014]

DEALING WITH BIOLOGICAL DATA

- Data are often difficult/expensive to obtain
- Too little data point with respect to the number of variables
- Data are often noisy
- Biological effects often take time to manifest, so delay should be considered

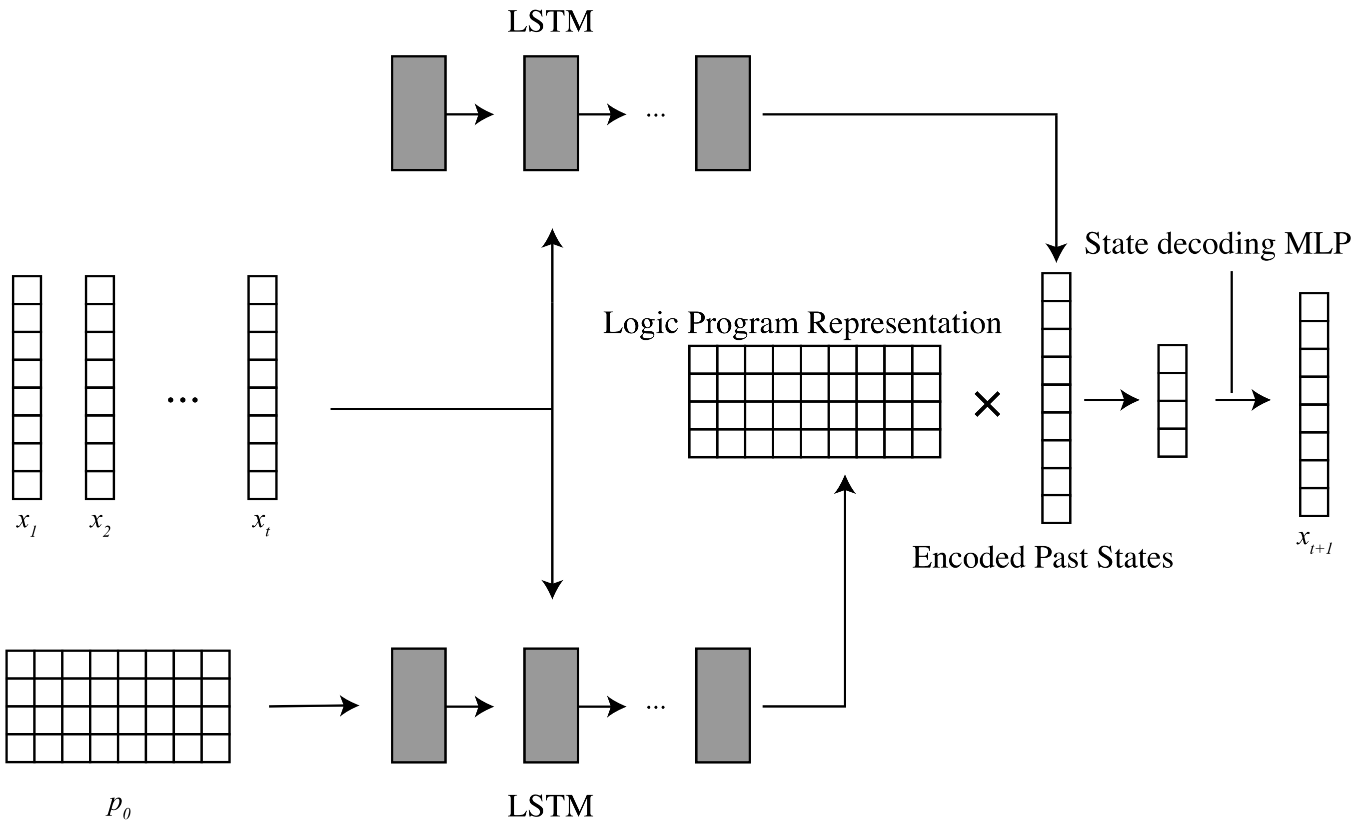
CURRENT ALGORITHMS

- Logical method [Inoue *et al.* 2014], [Ribeiro *et al.* 2014], [Ribeiro *et al.* 2015]
  - Produces NLP as output
  - Difficult to apply to small amount of data
  - Cannot deal with noisy data
- Neural network [Gentet *et al.* 2016]
  - Separate algorithm to obtain NLP
  - Applicable to small amount of data
  - Do not deal with delay

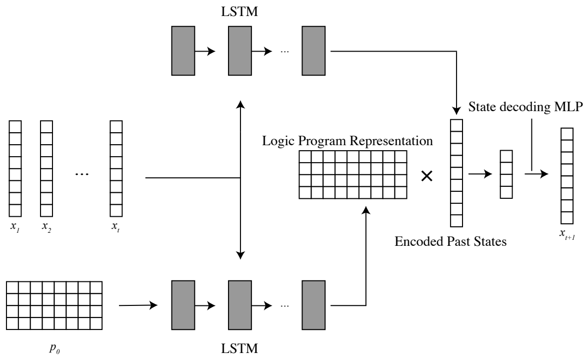
Ideally...

1. Can apply to small amount of data
2. Can deal with delay
3. Can deal with noisy data
4. Can output NLP directly
5. Can compute fast

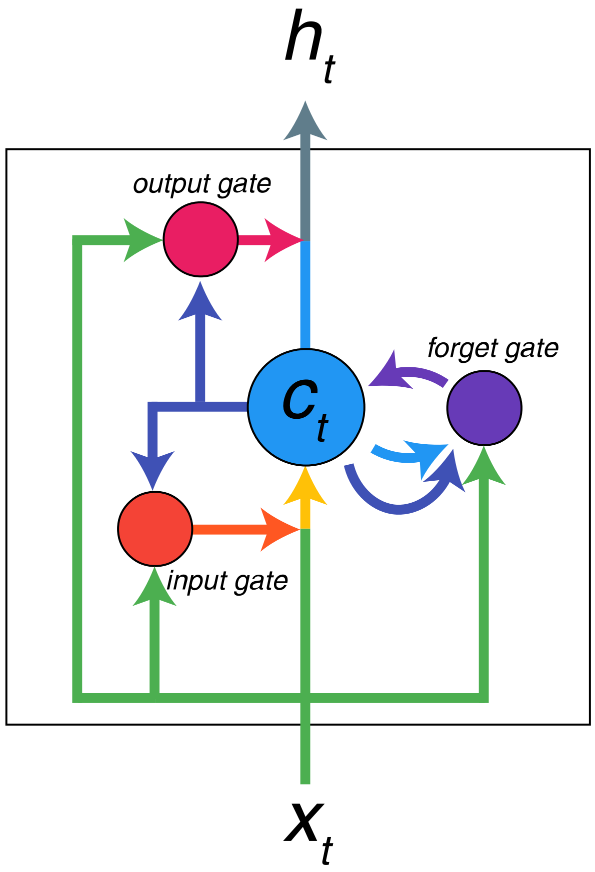
PROPOSED MODEL

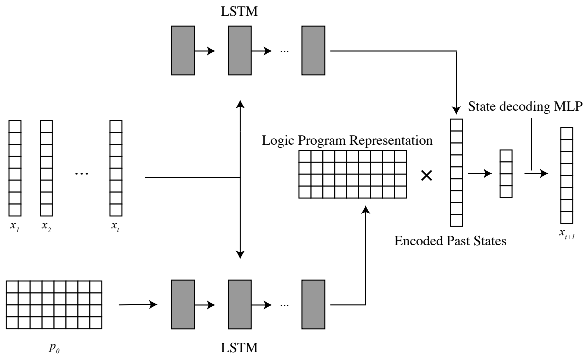


LSTM



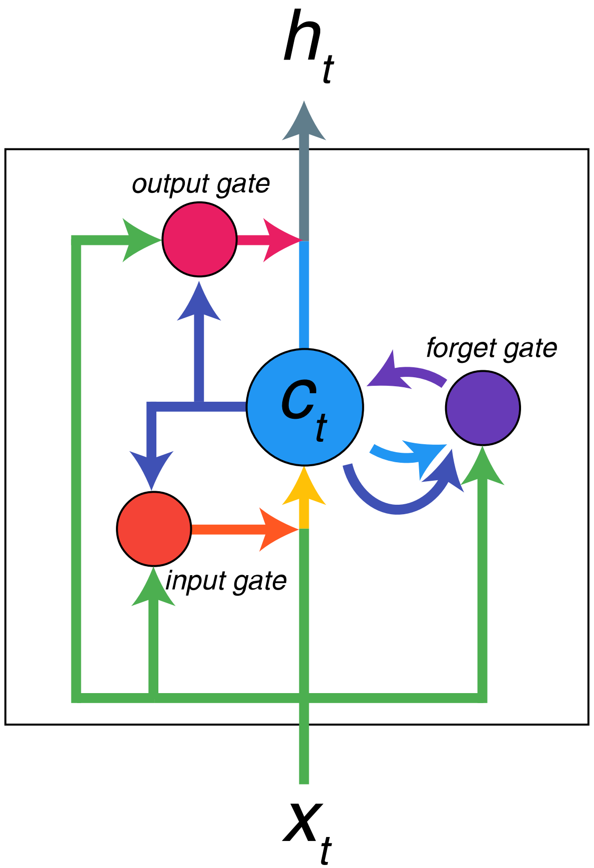
- Long Short-Term Memory (LSTM) is a form of Recurrent Neural Network
- Can learn long-term dependencies, and do not suffer from vanishing gradient problem
- Popular in many sequence-to-sequence mapping tasks such as machine translation
- LFIT can be viewed as mapping a sequence of state transitions to the correct logic program

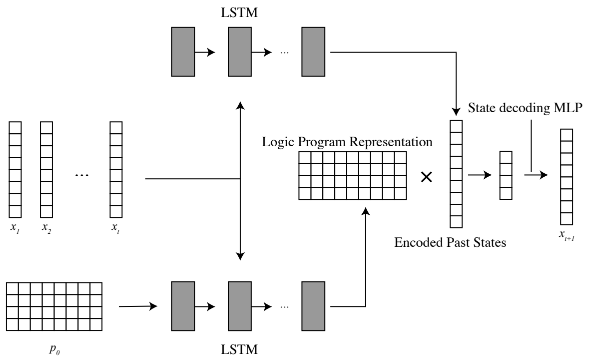




LSTM

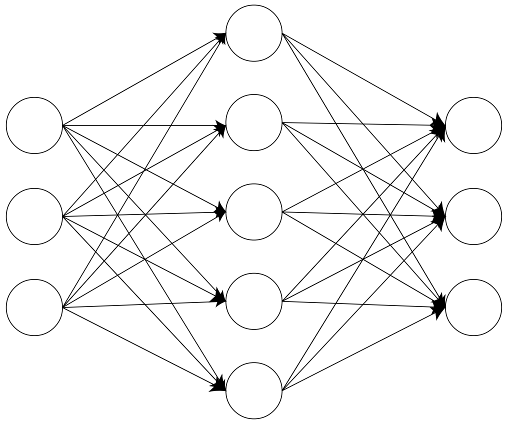
- 3 gates in the memory cell controls the internal state
- In our model, two LSTMs used:
  1. Autoencoder for state transitions
  2. For performing LFIT

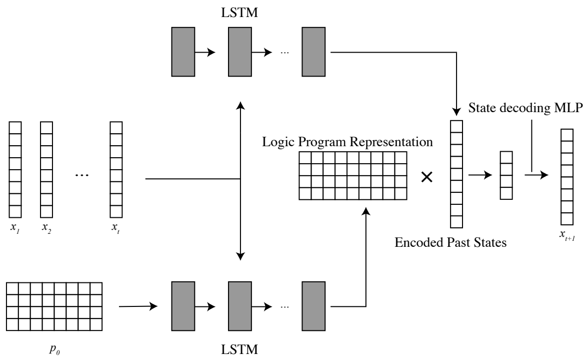




MLP

- Multi-layer Perceptron is a class of feed-forward neural network
- Aside from the input nodes, each layer is activated by rectified linear units
- In our model, we used an MLP with 2 hidden layers, each with 32 units

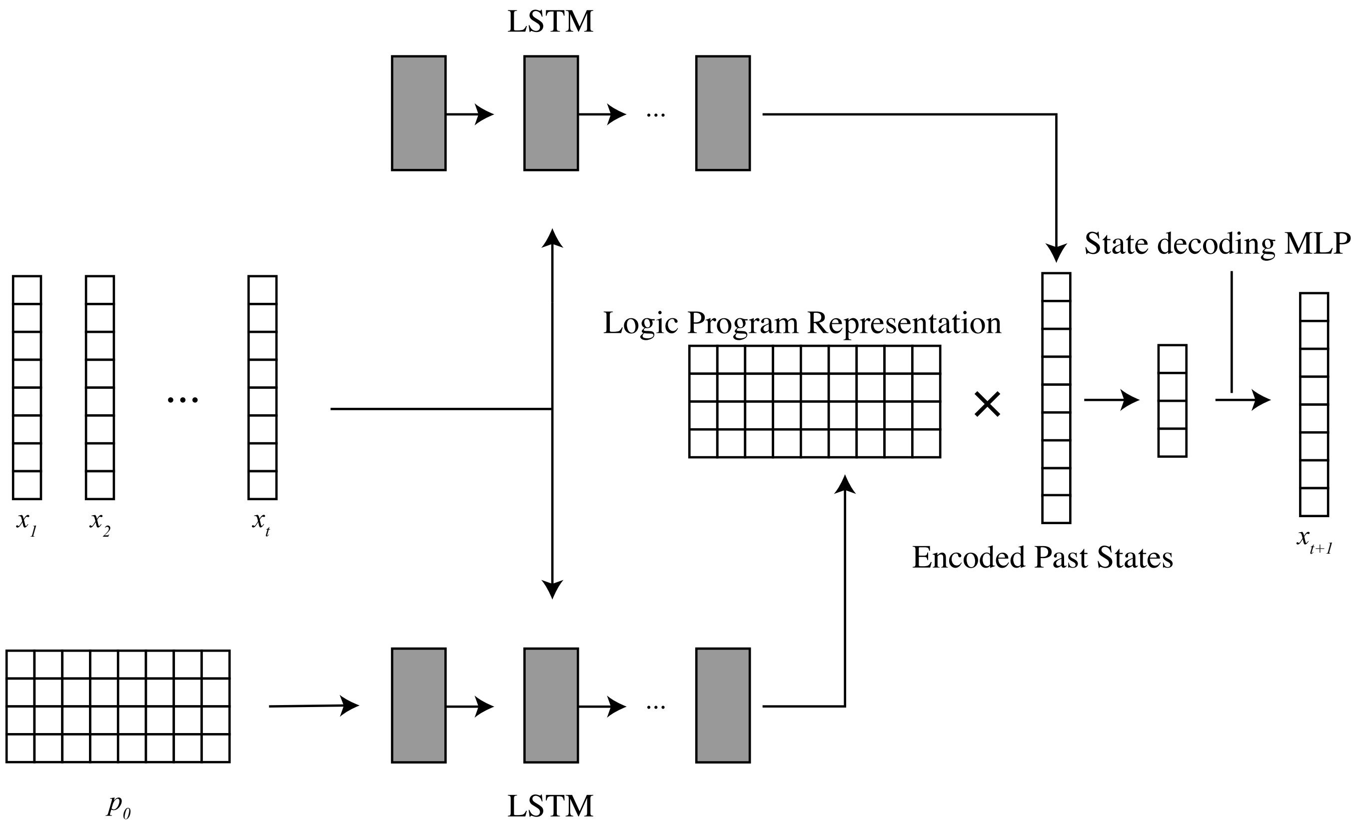


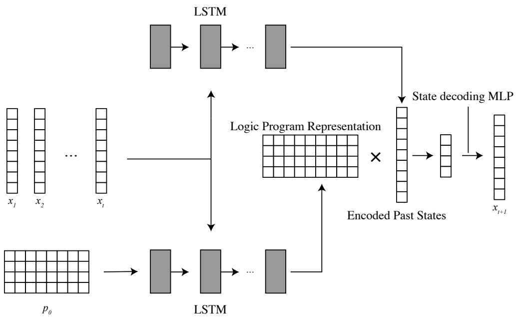


TRAINING THE MODEL

1. Randomly generate NLPs
2. Randomly initialize the state and generate time series
3. Train model on randomly generated data until model converges

PROPOSED MODEL





EVALUATION

1. Accuracy of the prediction
2. Quality of the learned NLP representation

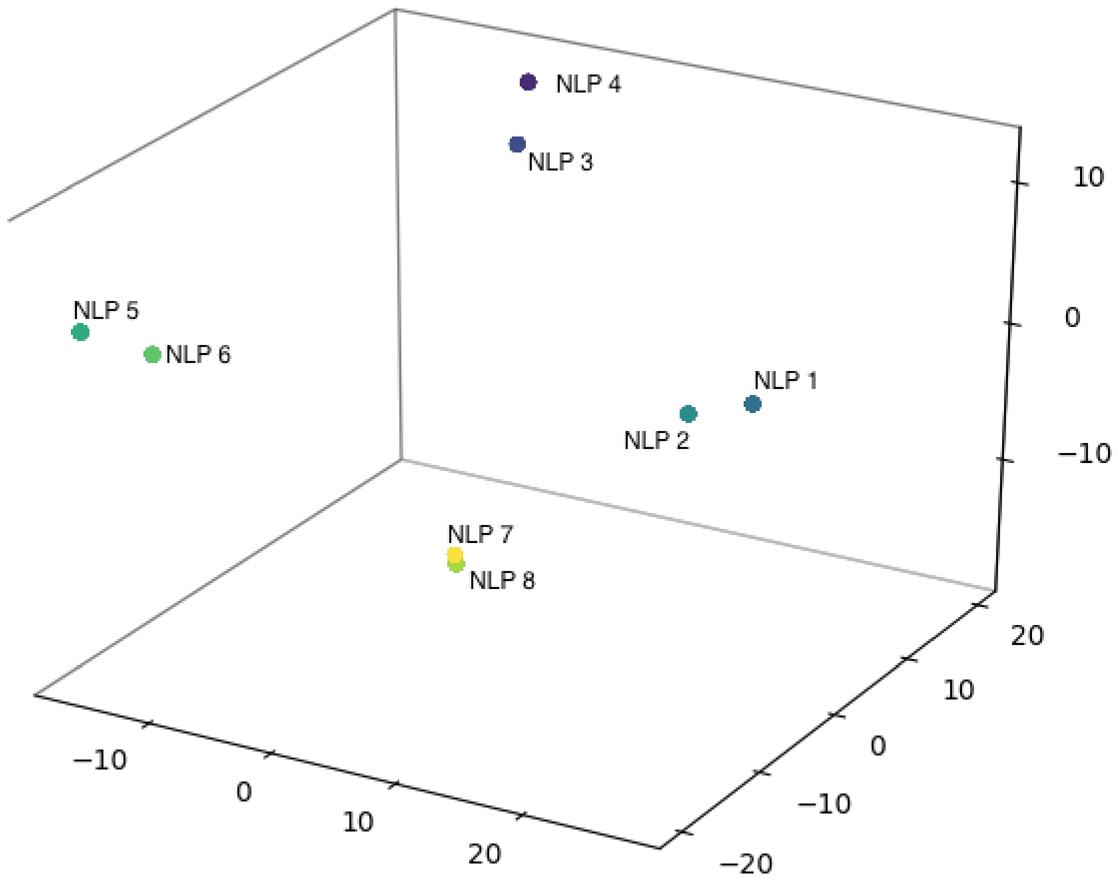
EXPERIMENT SETTING

1. Created an artificial boolean network based on mammalian cell cycle
2. Initialized with random initial state, and a time series is generated from the initial state
3. Noise are added randomly to the time series
4. The model is then used to attempt to recover the time series from several initial values

ACCURACY

Dataset	MSE (Original)	MSE (Noisy)
1	0.14	0.20
2	0.19	0.20
3	0.20	0.15
4	0.19	0.14
5	0.19	0.20

QUALITY



NLP 4
NLP 3
NLP 5
NLP 6
NLP 1
NLP 2
NLP 7
NLP 8
10
0
−10
−10
0
10
20
20
10
0
−10
−20

QUESTIONS RECEIVED (1)

- How is this a classification task and what are the competing models?

  A: In a sense this model is trying to classify a series of state transition with the corresponding logic program that can explain it. Currently there are no competing models.

QUESTIONS RECEIVED (2)

- There are no comparison between smaller dataset and larger dataset?

  A: No experiments were done on this. Logical methods require every single possible transition compared to neural network methods which require less amount of data than that.

CONCLUSION

- Contribution:
  - Proposed a method for learning from small amount of data
  - Proposed a method that utilizes neural network and can learn from transitions in delayed setting
- Future work:
  - Input partial logic program as background knowledge
  - Extract normal logic program from the learned program representation