Muiltple Object Tracking using RNNs

Khizer Amin

Introduction

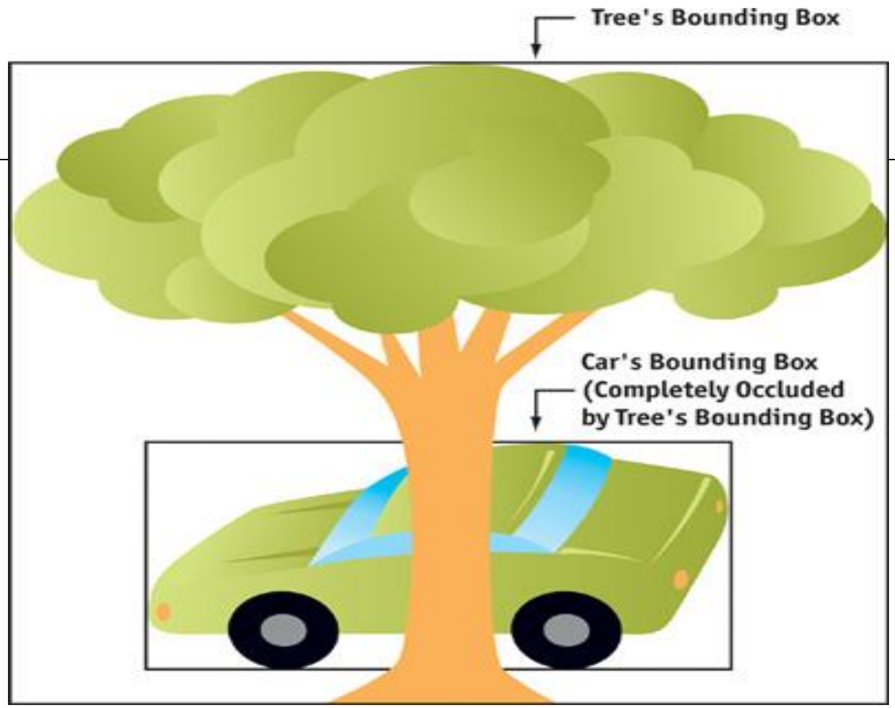
- Why tracking?
  - Association: keep the object identity across frames
  - SpeedUp: Local prediction (fast for intermediate frames)

- Why RNN/LSTMs
  - Natural language processing
  - ANNs can‘t deal with the temporal or sequential data.
  - Memory (They retain context by having memory)

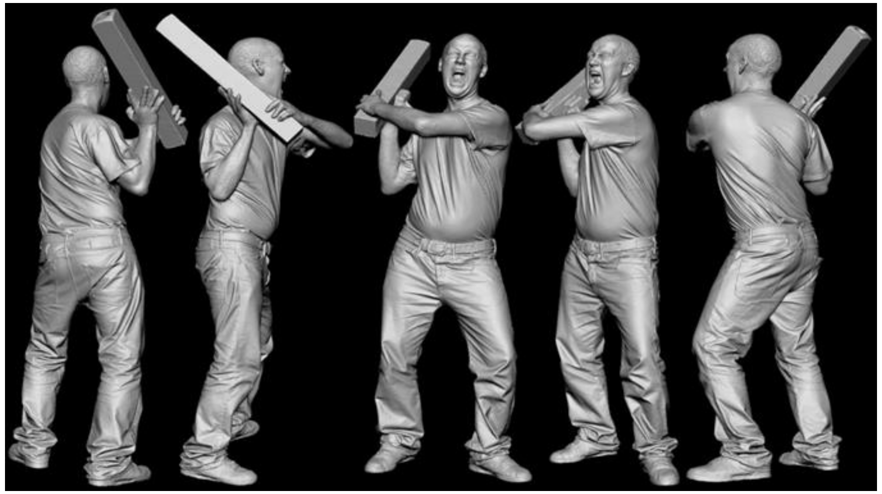
- Applications
  - Video surveillance
  - Traffic management
  - Medical imaging

Challenges for Tracking

- Occlusion
  - It is a classic reseach problem in computer vision.

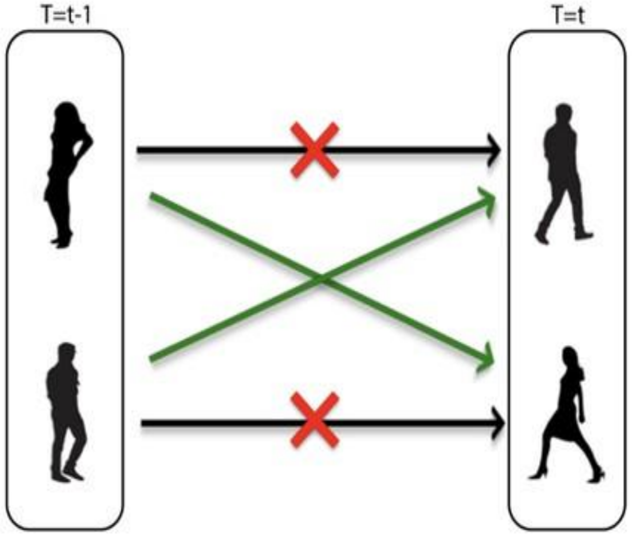


- Different view points
  (camera motion)



Challenges for Tracking

- A varying number of targets
  (birth/death of targets)

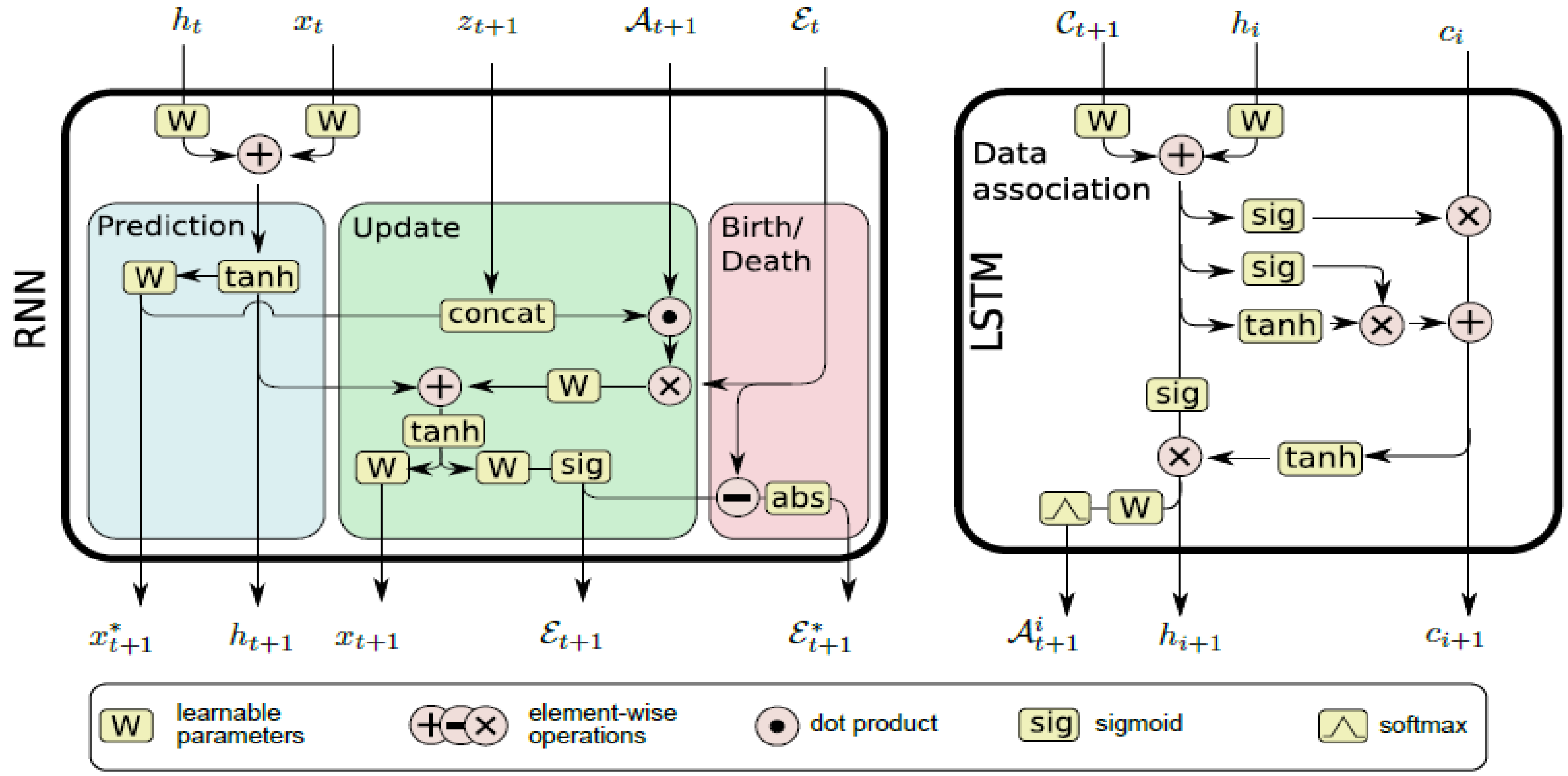


- Data Association

Related Work	$\mathbf{MOTA^1}$	Differences
Online Deep Tracking Metric learning (A. Sadeghian, A. Alahi, S. Savarese - 2016) **(RNN based)** Tracking multiple target using cues(their appearance, motion and inter-relations)	37.6%	– Using 3 RNNs as LSTMs for each cue. – Matching targets in each frame.
Learning to Track: Online Multi-Object Tracking by Decision Making. (Y. Xiang, A. Alahi, S. Savarese - 2015) Tracking using Markov decision process, where each object‘s lifetime is modeled by an MDP.	30.3%	– MDP handle the birth/death and appearance/disappearanc e of objects in tracking. – Data Association using Reinforcement learning.
Joint Probabilistic Data Association Revisited (JPDA) (Rezatofighi et al. 2015) Associating the detected measurements in each time frame with existing targets using a joint probabilistic score	23.8%	– Reformulate the DA assignment score as a integer linear problem
Muiltple Object 3D Position Tracking using RNNs **(Our approach)** Tracking based on RNN(motion, prediction and update) and LSTM(data association) **(This was the first approach of the time which employs RNNs/LSTMs for online MOT, and also non trivial for DL approaches at that time)**	**19.0%**	– Time varying nr. of targets. – State estimation of targets. – Data association

1. Multiple-Object tracking Accuracy

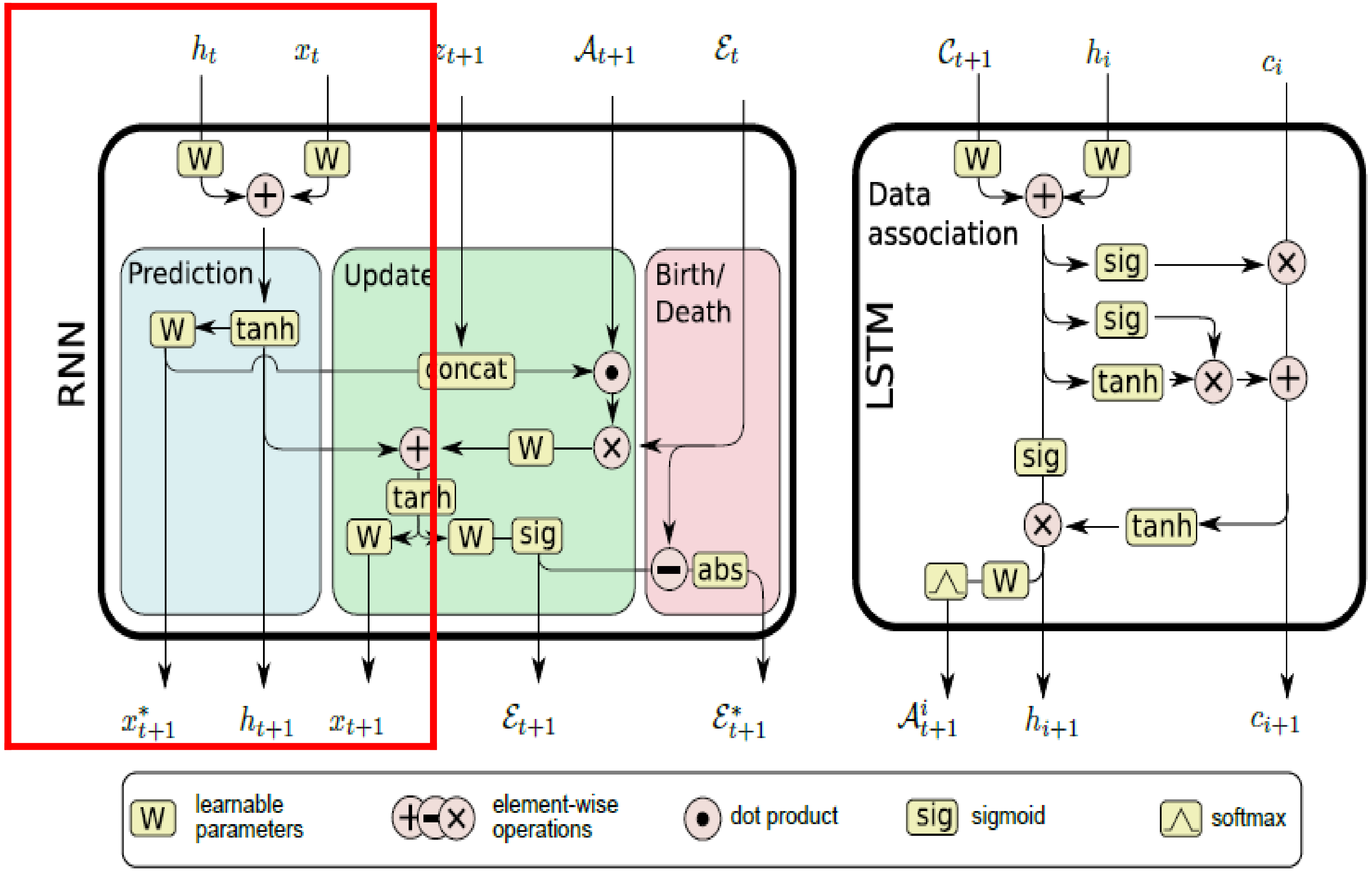
Architecture



RNN Stage - Prediction Layer

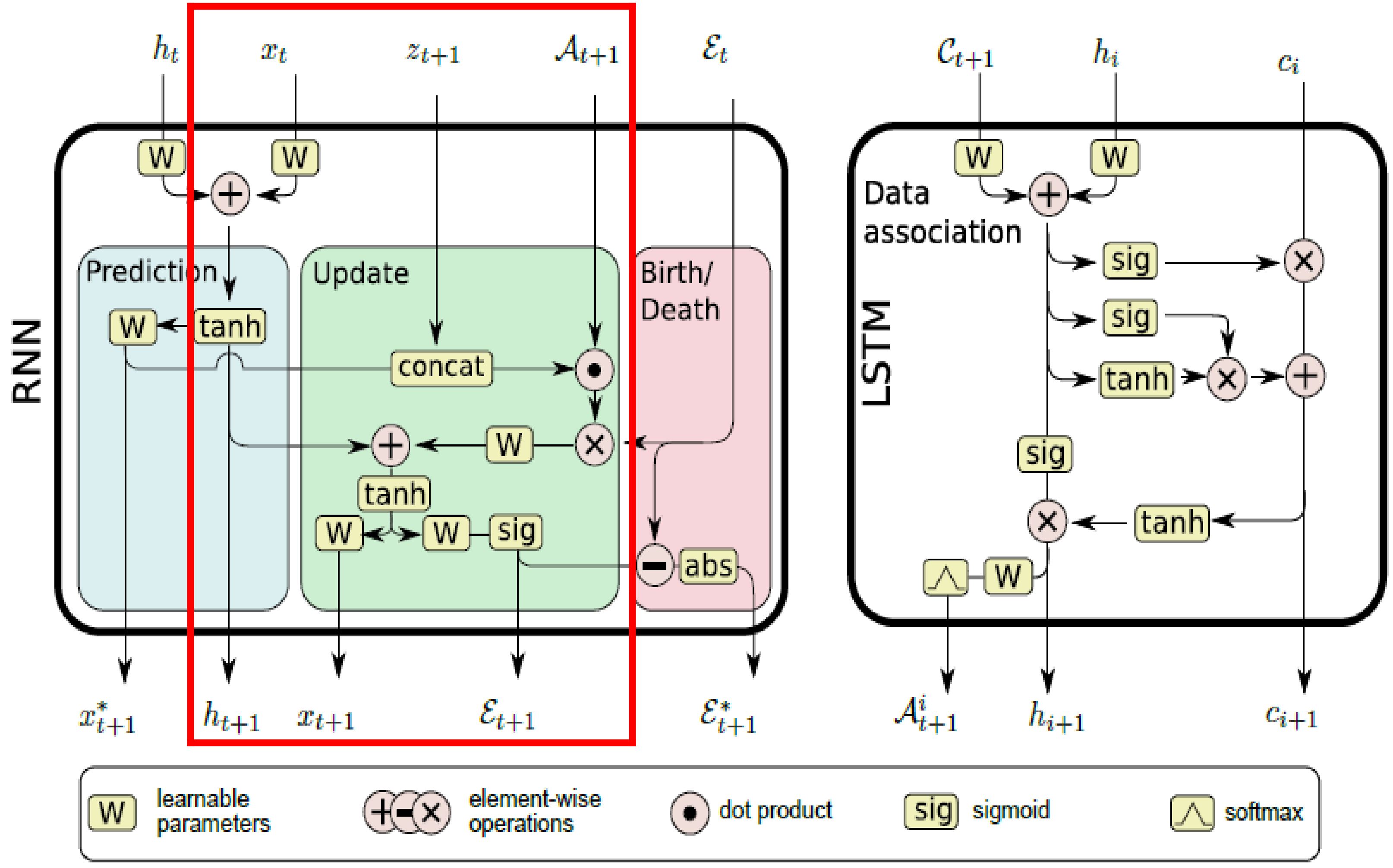
- Using available ground truth Bboxes from current and the previous state.

- Predicting the target motion in the absence of the measurements.



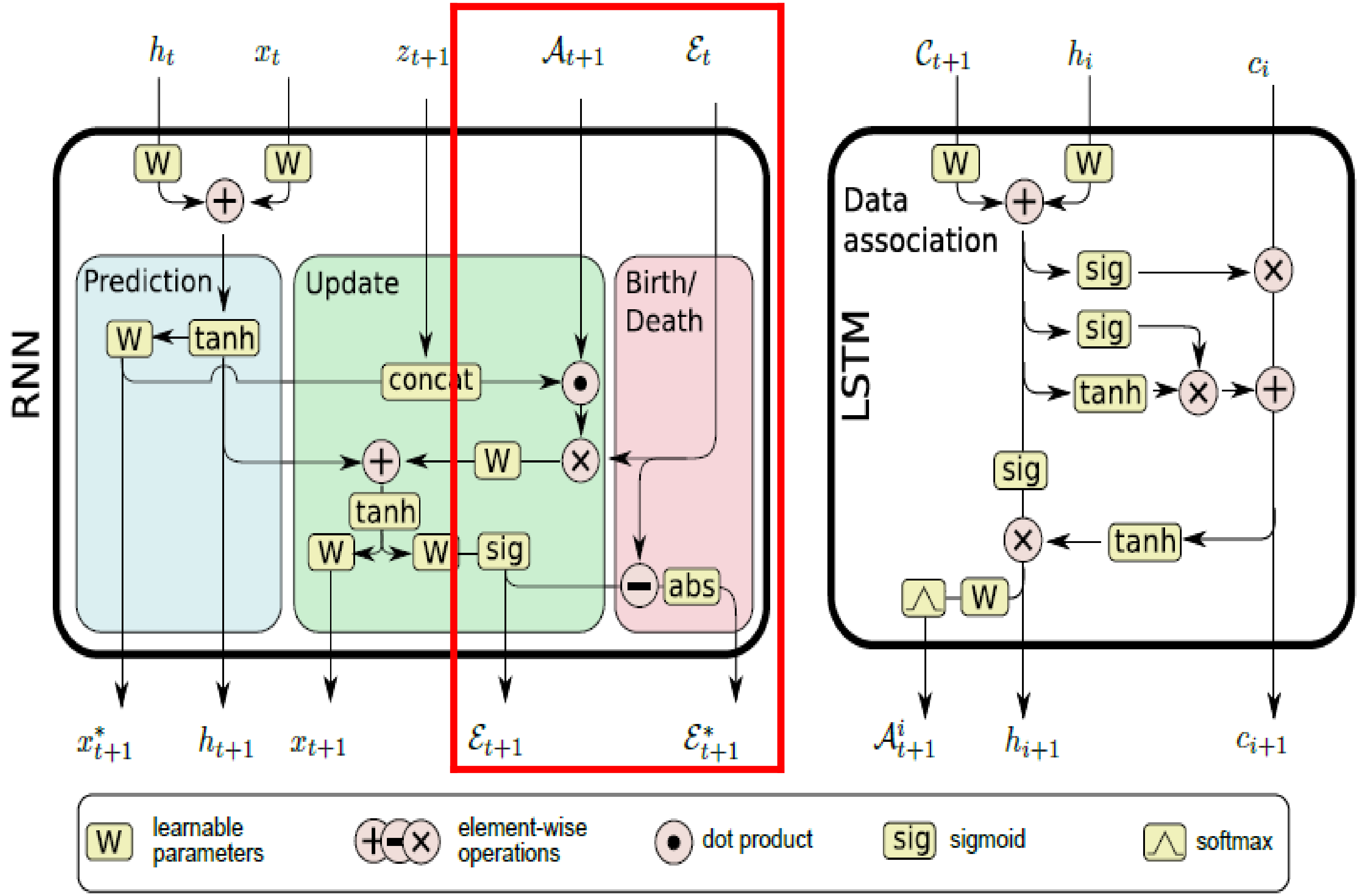
RNN Stage – Update Layer

- Updating the state for the target ‘x’ by updating the current status of the targets Bbox coordinates by introducing the measurements data ‘z’.
- Comparison between the measurement and target coordinates.

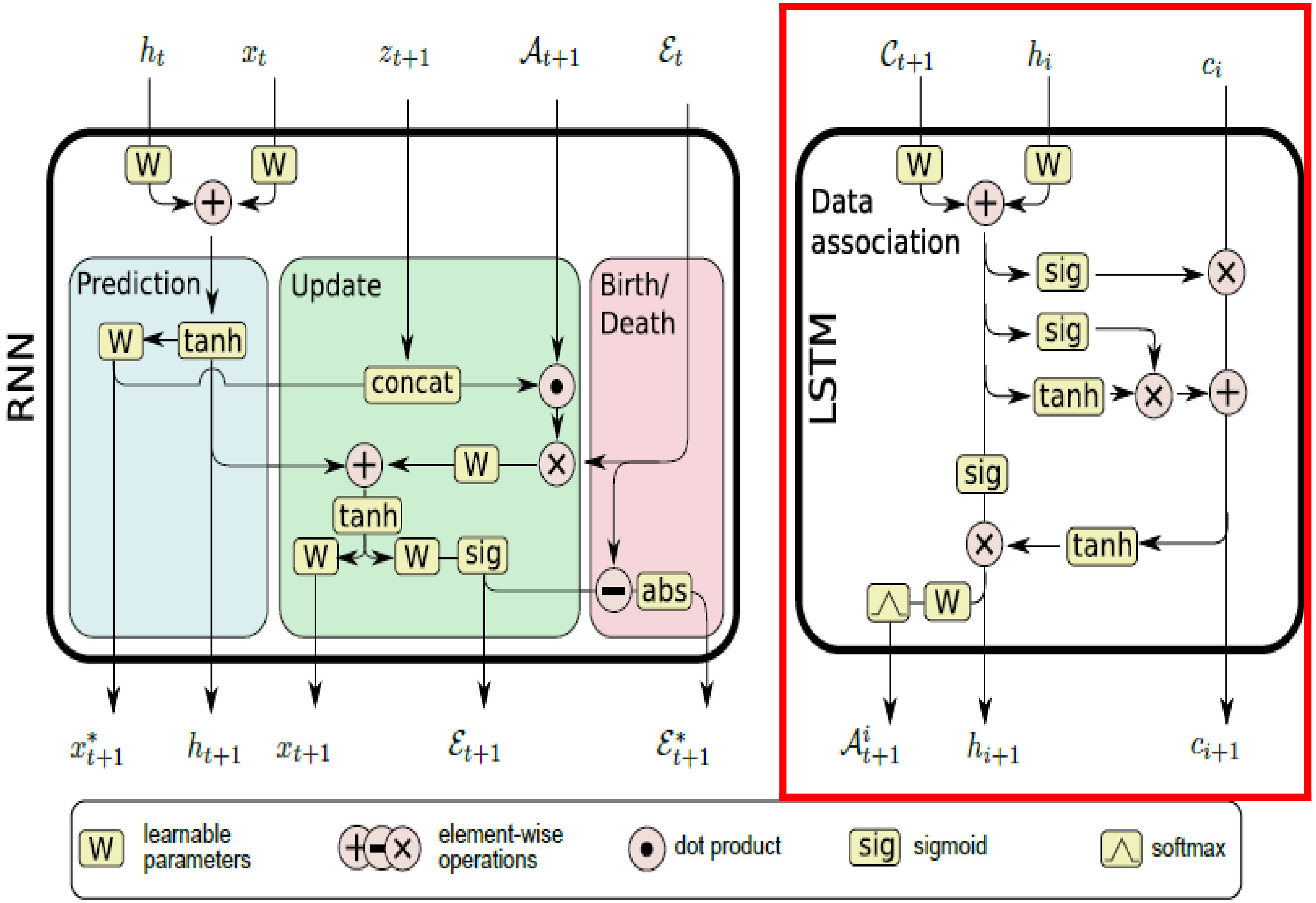


RNN Stage – Birth/Death Layer

- Targets can appear or disappear from a frame.
- Preserving target's identity and association.
- Identifying target's track initiation and termination.



LSTM Stage



- Performing data association.
- Creating the asscociation matrix 'A' using the LSTM cell.
- Using to 'A' matrix to preserve the context and identity of a target over time.

Data Association using LSTM

- LSTM has a strong memory unit.

- LSTM cell outputs the association vector between predicted targets and detected measurements.

$$C = ||x^i - z^j||_2$$

- The association vector then passed to the RNN update layer to estimate the state of the target 'x_{t+1}'.

- **The process to train LSTM for data association is complex. Model can be simplified by removing LSTM based data association.**

Data Association - Alternative approach

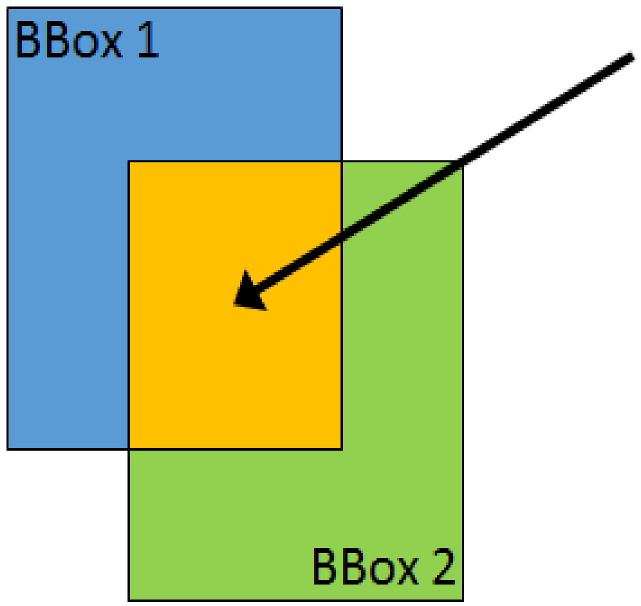
- **Intersection over Union (IoU)**

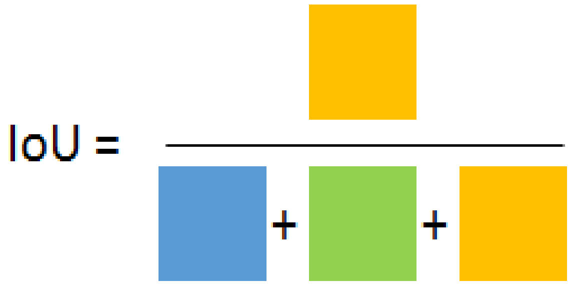
- **Overlap of predicted targets and detected measurements:**

$$IoU = \frac{Intersection\ area\ of\ two\ bounding\ boxes}{Union\ of\ areas\ of\ both\ bounding\ boxes}$$

Data Association - Alternative approach

- Finding similarities between Bounding boxes.





Experiment

- **Training Data**

  - Data with sequence of images is hard to get.

  - MOTChallenge 2015 is used with 22 video sequences.

- **Implementation:**

  - Keras-Tensorflow

  - The orignal paper paper was inspired by Andrea Karpathy RNN algorithm for textual data.

Evaluation Metrics

- Two standard metrics:

- **MOTA**

$$1 - \frac{\sum_t (m_t + f_{pt} + mme_t)}{\sum_t g_t}$$

- **MOTP**

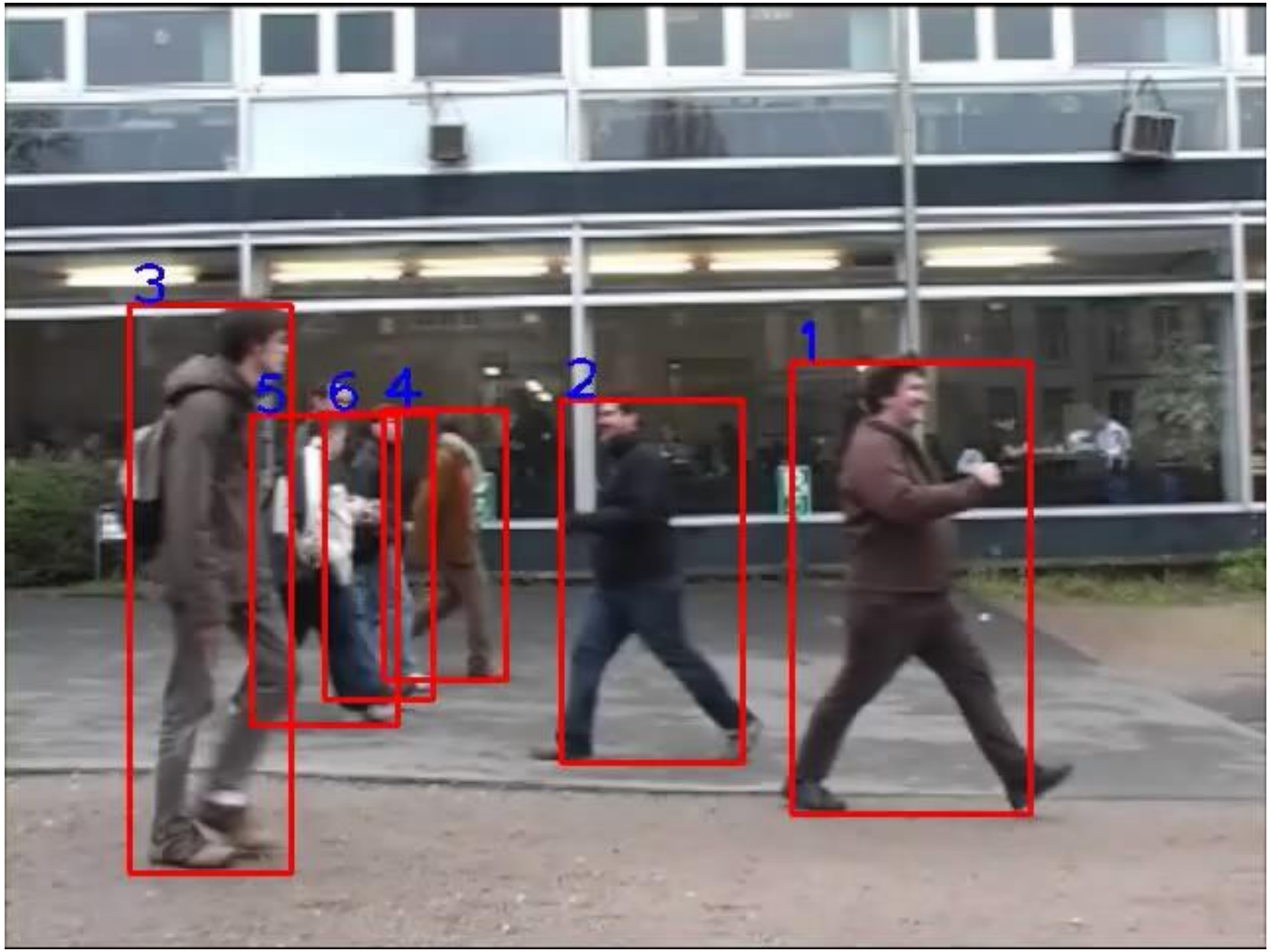
$$\frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}$$

Results

Method	MOTA (%)	MOTP (%)
AMIR15(SadeghianAS17)	37.6	71.7
MDP(xiang_iccv15)	30.3	71.3
JPDMm(Rezatofighi2016)	23.8	68.2
RNN_LSTM(Milan:2017)	**19.0**	71.0
Our implementation		
Prediction Layer	5.1	30.3
Update Layer with IoU	17.6	70.8

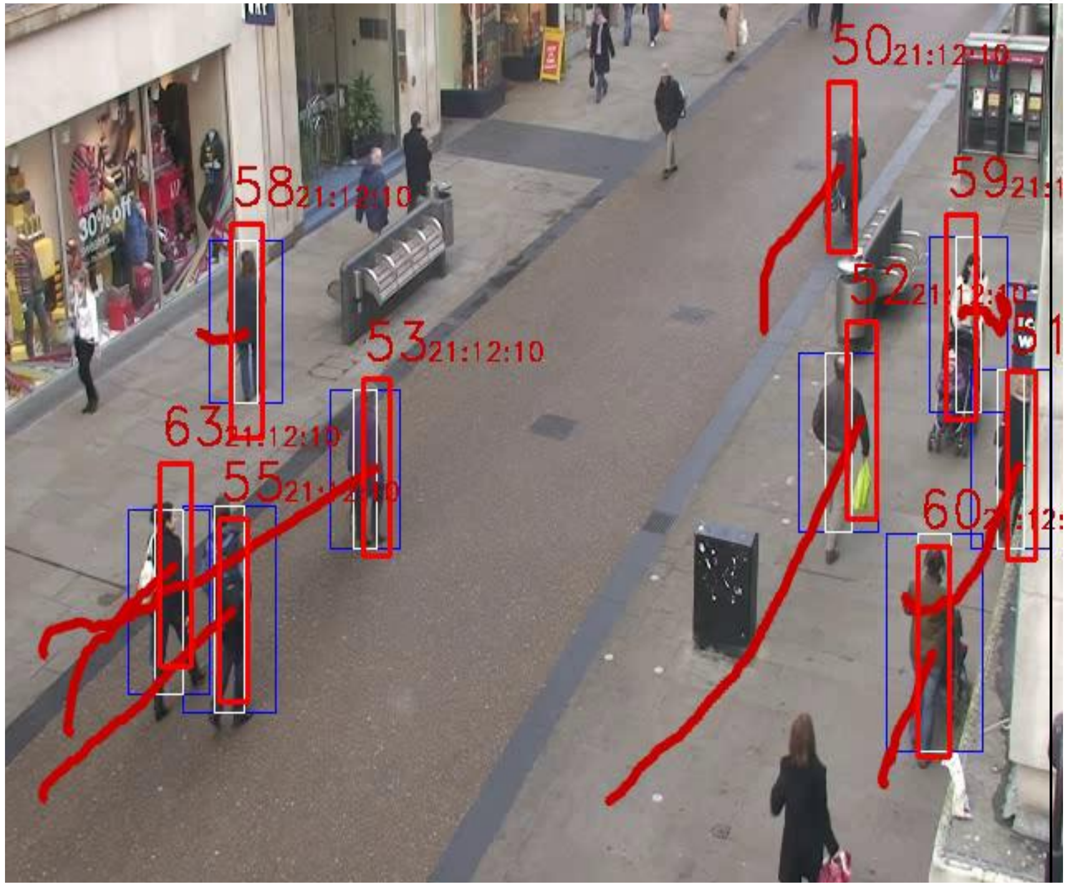
Table 4.1: Tracking results on MOTChallenge test set.

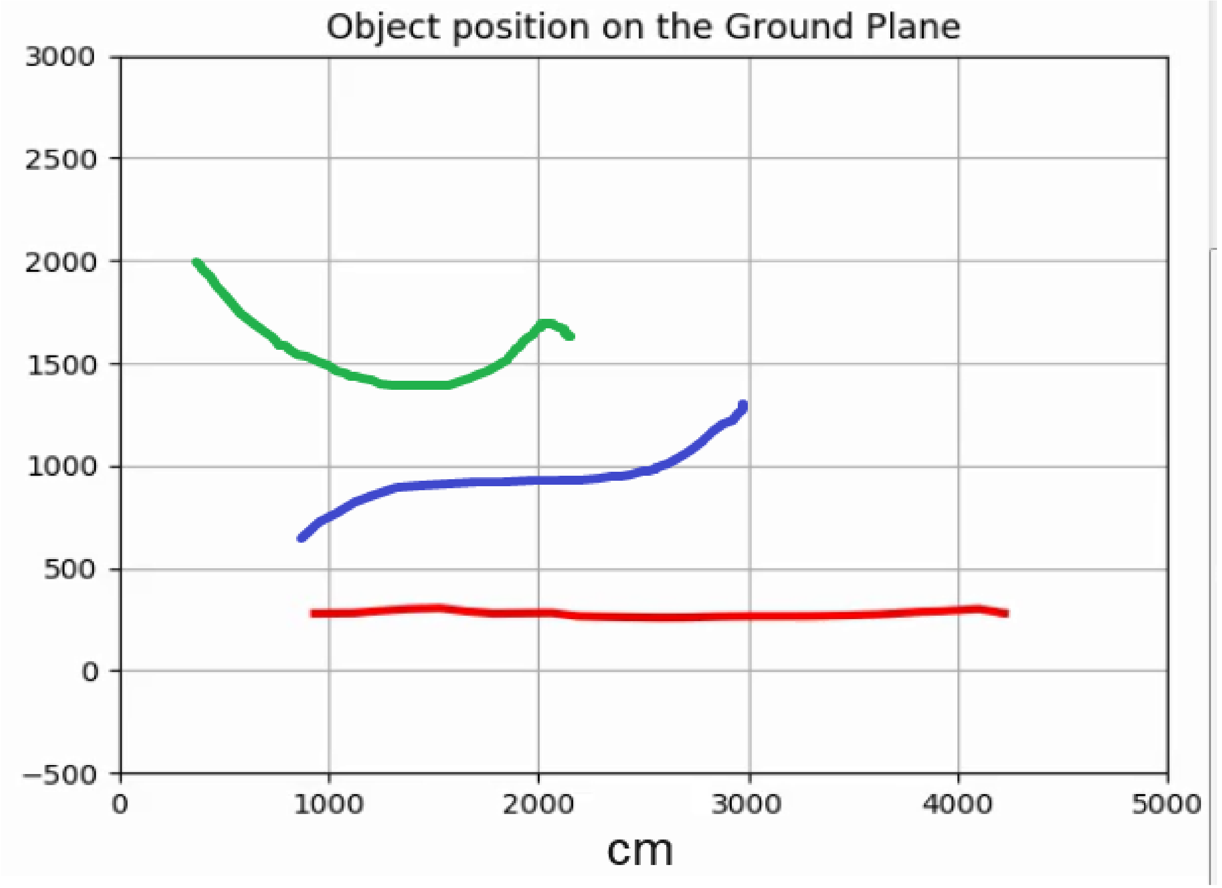
Demonstration



Next steps

Conversion (2D -> 3D)





Thank you for the attention