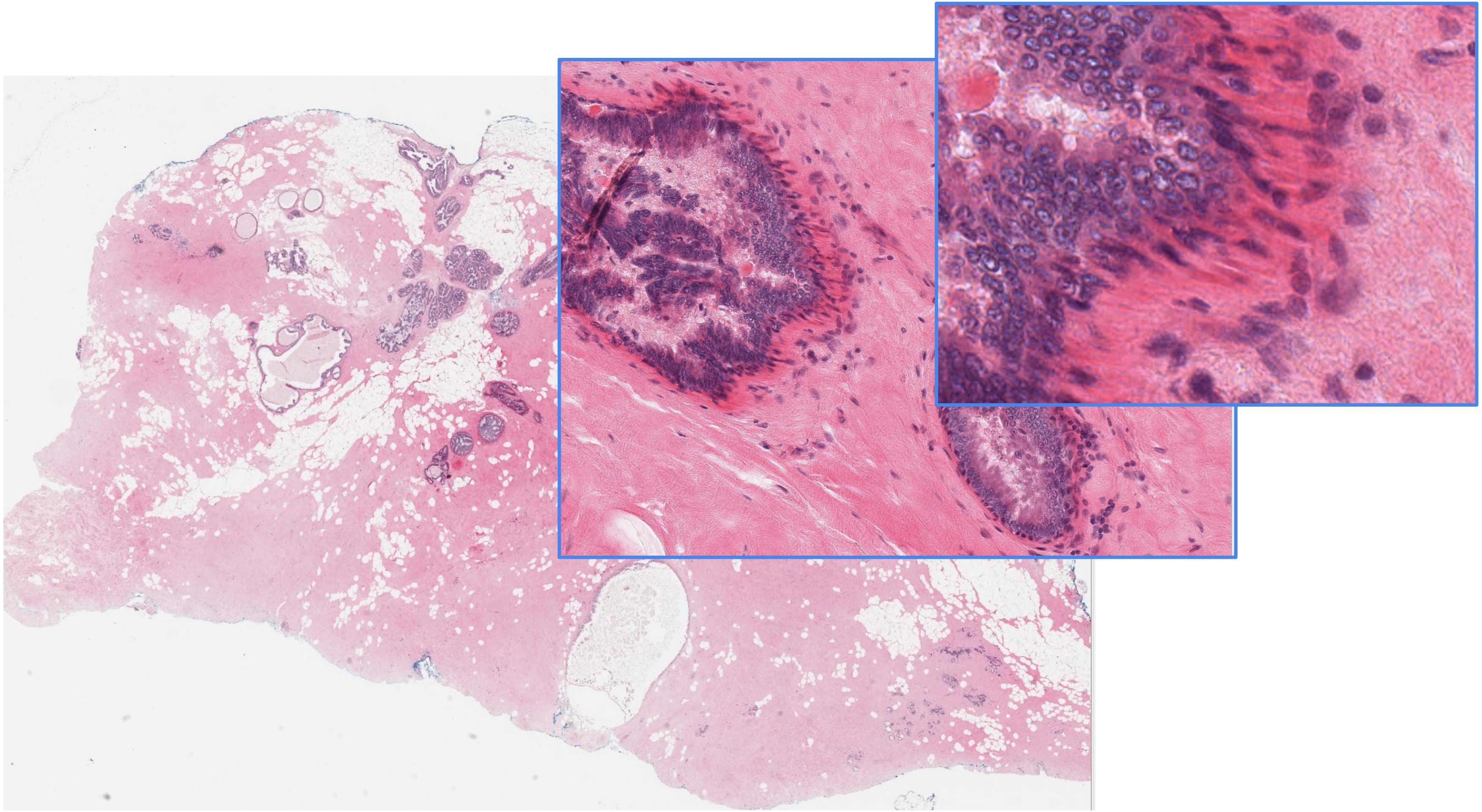
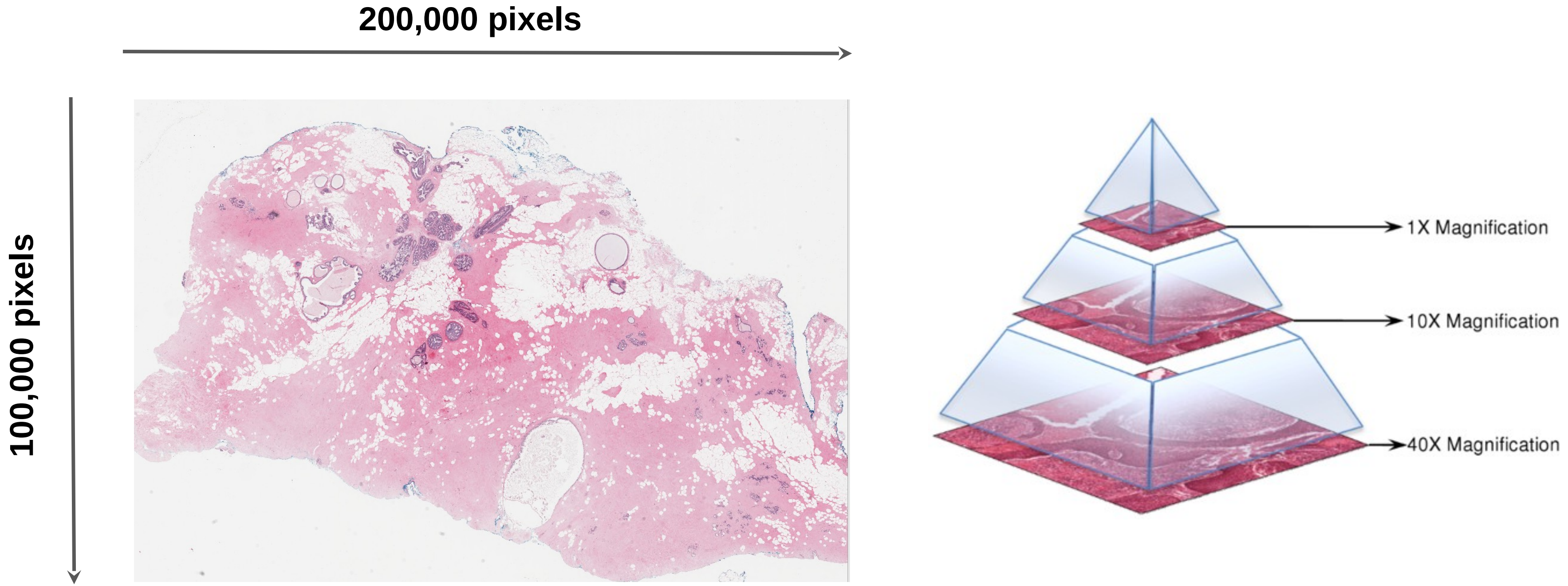
Recurrent Models of Visual Attention

Presented by Shazia Akbar
Sunnybrook Research Institute
Medical Biophysics, University of Toronto
Vector Institute

Toronto Deep Learning Series September 4, 2018




200,000 pixels
100,000 pixels
1X Magnification
10X Magnification
40X Magnification

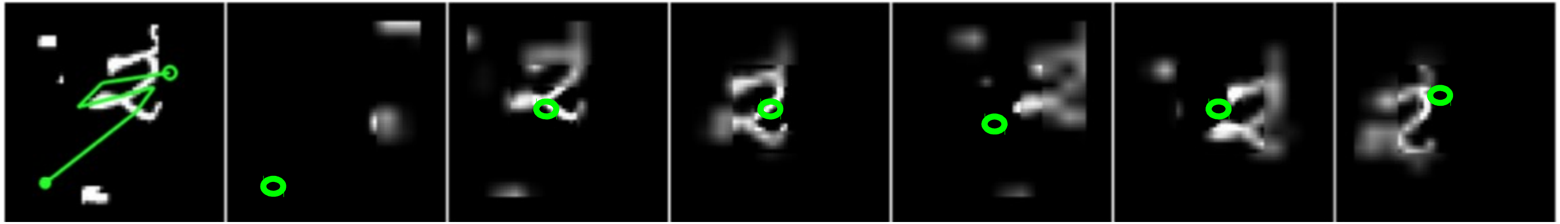
Recurrent Models of Visual Attention

Volodymyr Mnih **Nicolas Heess** **Alex Graves** **Koray Kavukcuoglu**
Google DeepMind

{vmnih,heess,gravesa,korayk} @ google.com

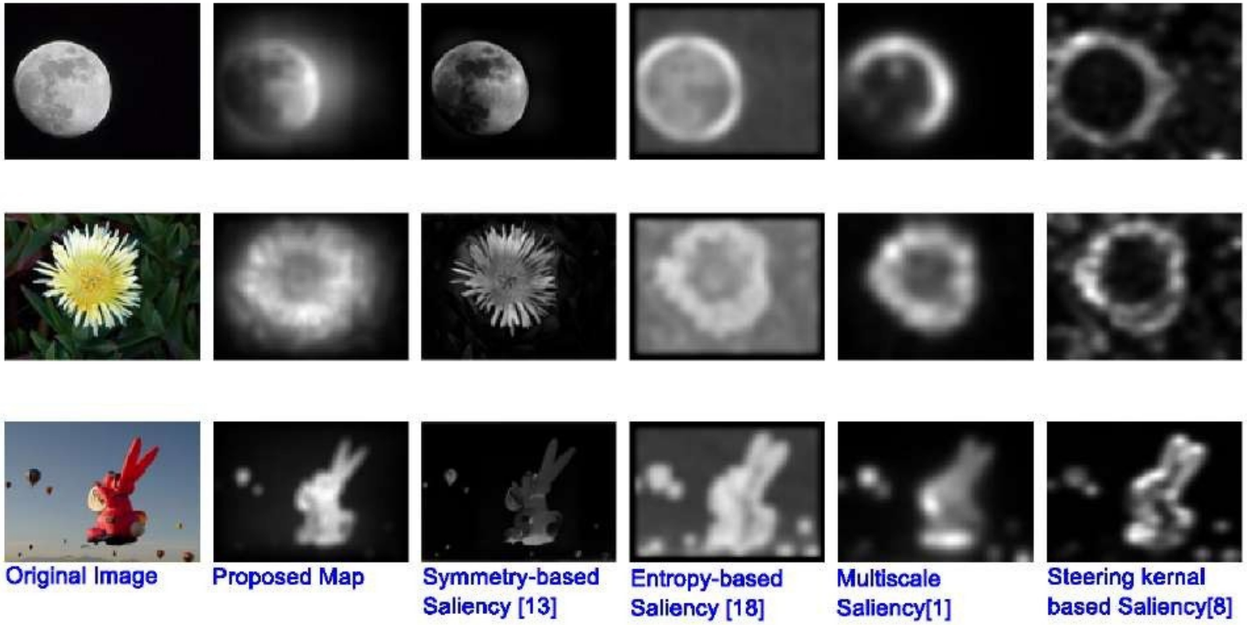
Abstract

Applying convolutional neural networks to large images is computationally expensive because the amount of computation scales linearly with the number of image pixels. We present a novel recurrent neural network model that is capable of extracting information from an image or video by adaptively selecting a sequence of regions or locations and only processing the selected regions at high resolution. Like convolutional neural networks, the proposed model has a



“The model sequentially chooses small windows of information on the data”

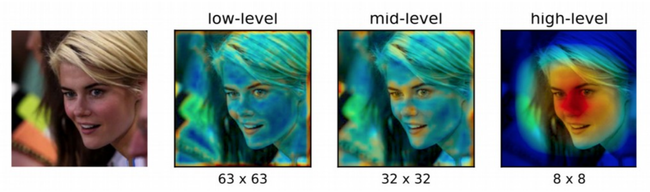
Saliency Maps

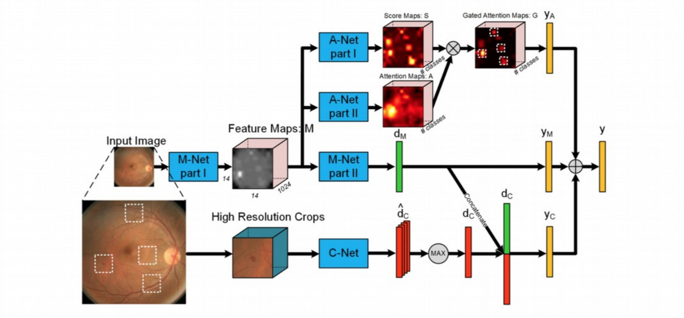


http://what-when-how.com/pattern-recognition-and-image-analysis/a-visual-saliency-map-based-on-random-sub-window-means-pattern-recognition-and-image-analysis/

Other Work

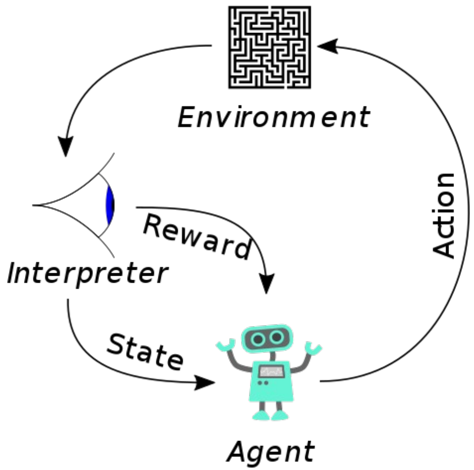
- Attention Maps: Lu et al [1], Zagoroyko & Komodakis [2], DRAW [4]
- Zoom-in-Net [3]
- ...



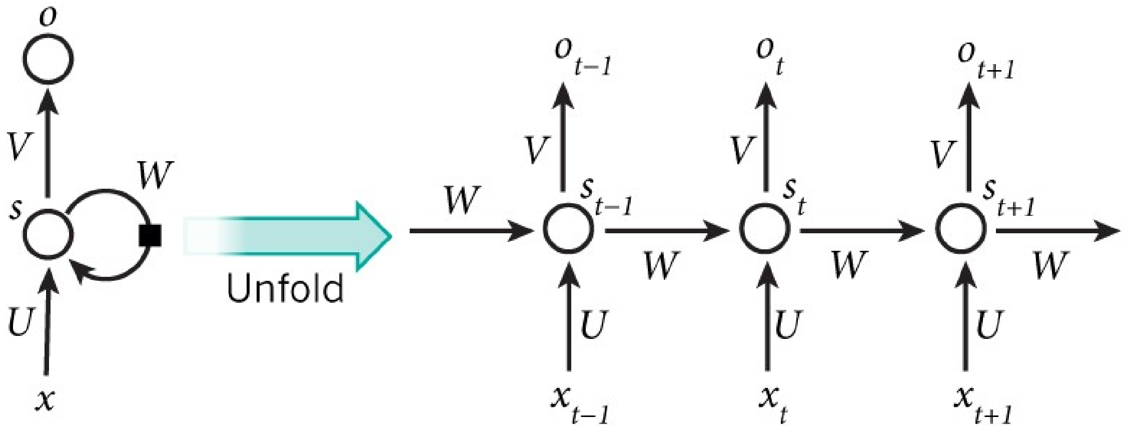


[1] Lu et al., Learning attention map from images, CVPR, 2012
[2] Zagoroyko & Komodakis, Paying more attention to attention, ICLR 2017
[3] Wang et al., Zoom-in-net: Deep Mining Lesions for Diabetic Retinopathy Detection
[4] Gregor et al., DRAW: A Recurrent Neural Network for Image Generation

RNNs



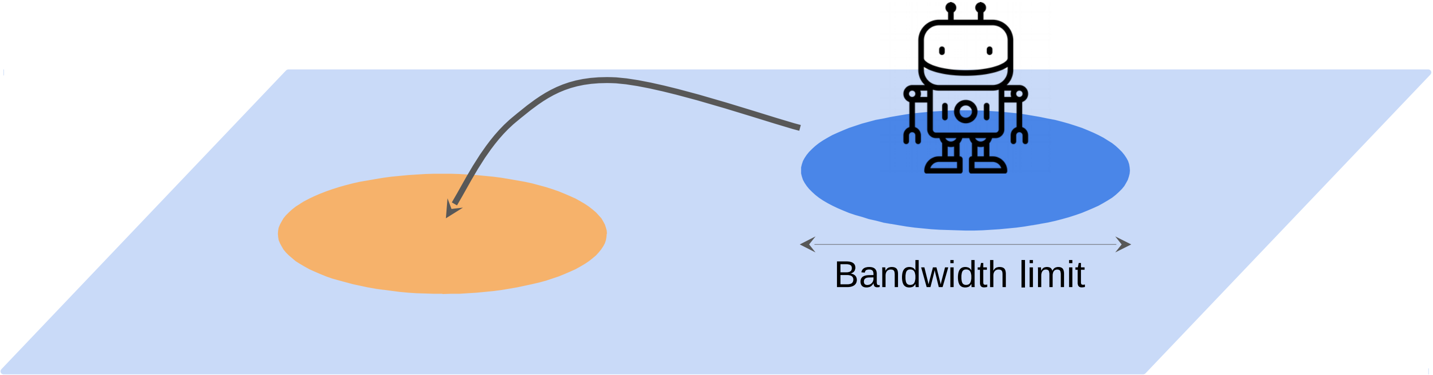
Environment
Action
Reward
Interpreter
State
Agent



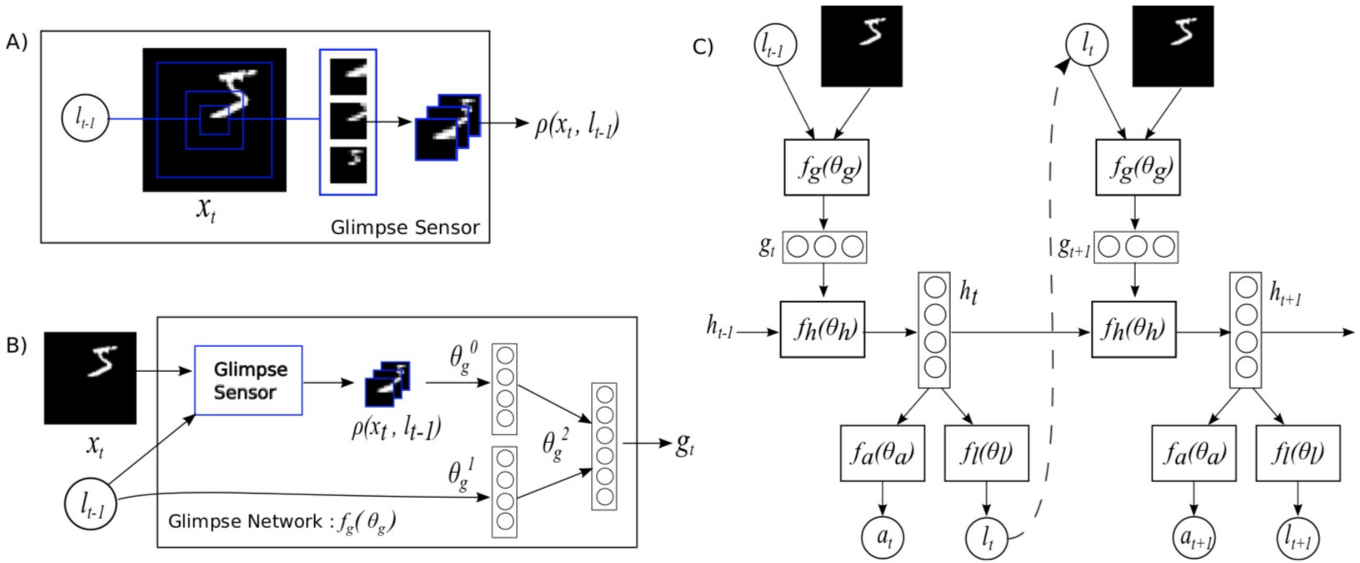
o
V
W
s
U
x
Unfold
o_{t-1}
o_t
o_{t+1}
V
W
s_{t-1}
s_t
s_{t+1}
U
x_{t-1}
x_t
x_{t+1}

Recurrent Attention Model (RAM)

State: **Image content, state of the game engine**
Action: **Classification decision, joystick controls**
Reward: **Correct decision?, points scored**



Recurrent Attention Model (RAM)



The Model

Sensor

Internal State

Action

Reward

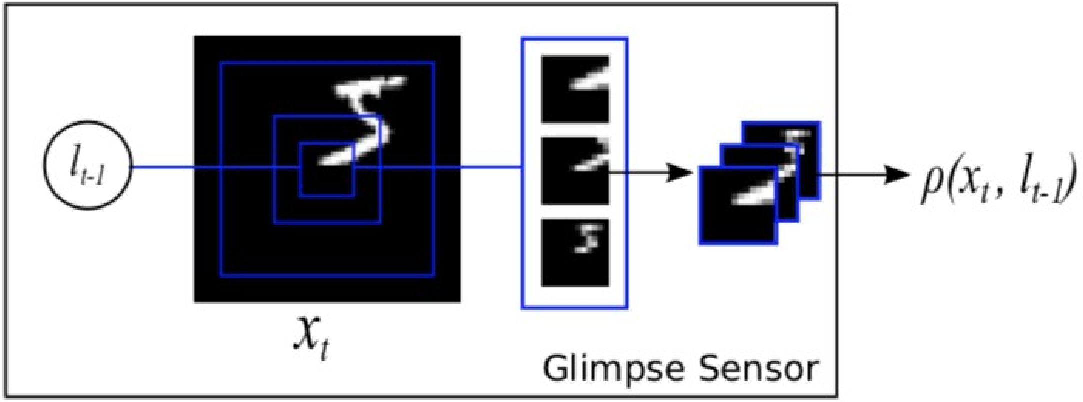
The Model

Sensor

Internal State

Action

Reward



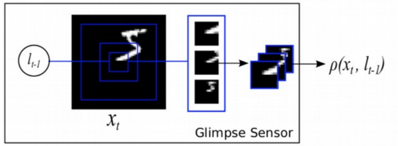
The Model

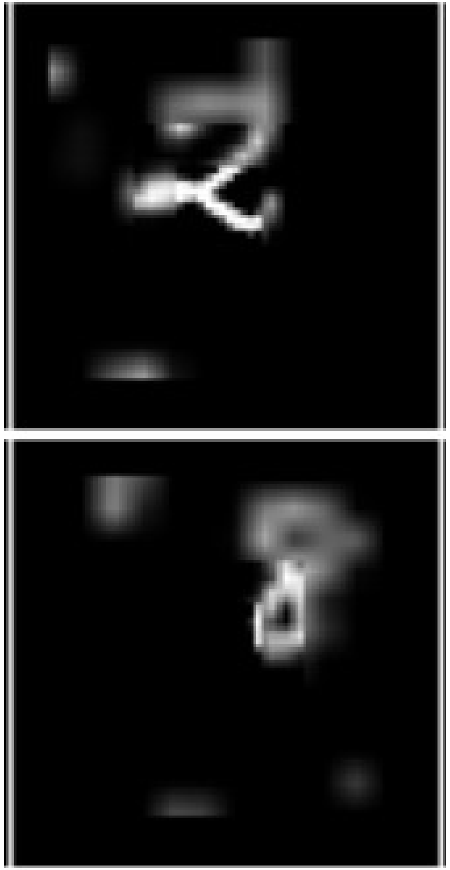
Sensor

Internal State

Action

Reward





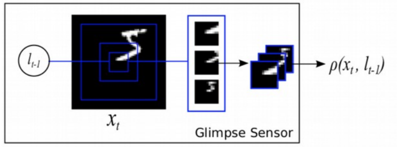
The Model

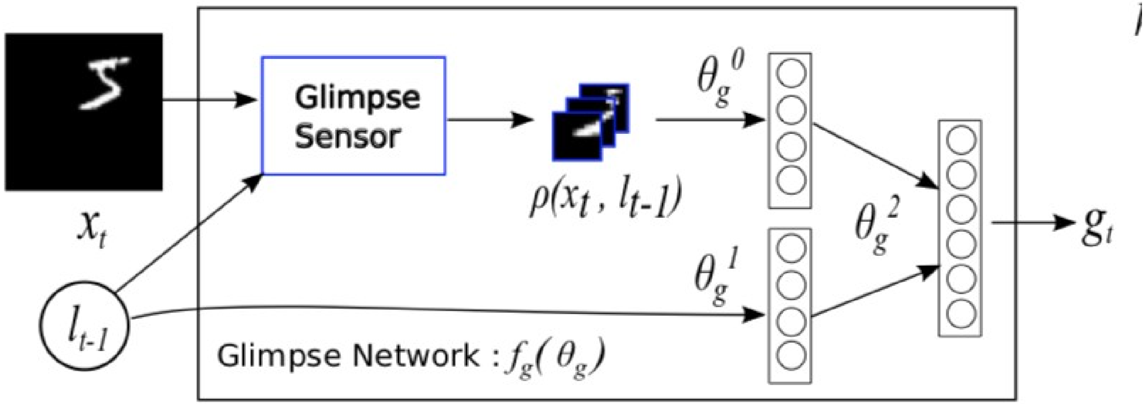
Sensor

Internal State

Action

Reward





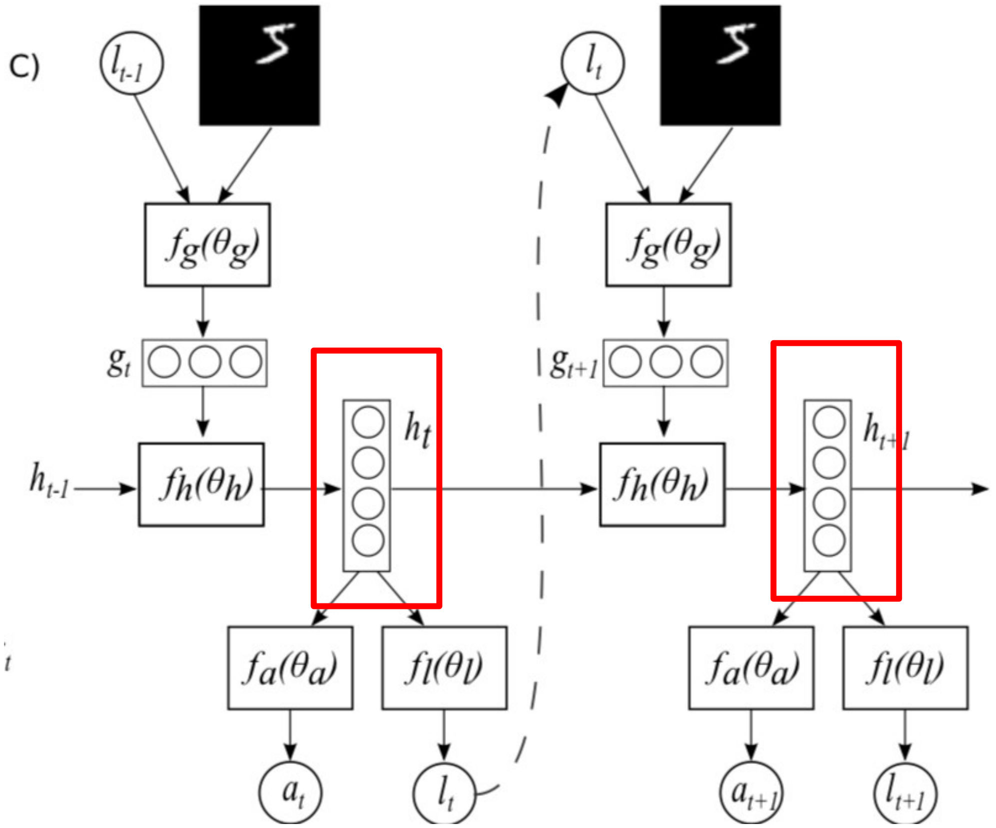
The Model

Sensor

Internal State

Action

Reward



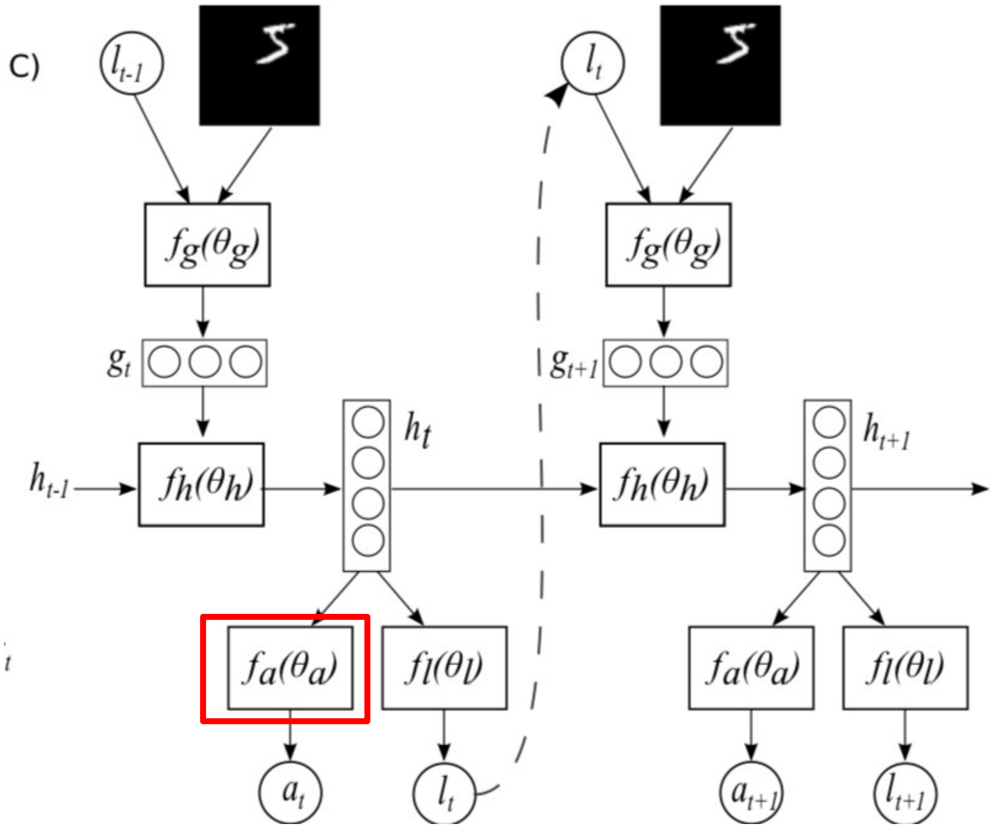
The Model

Sensor

Internal State

Action

Reward



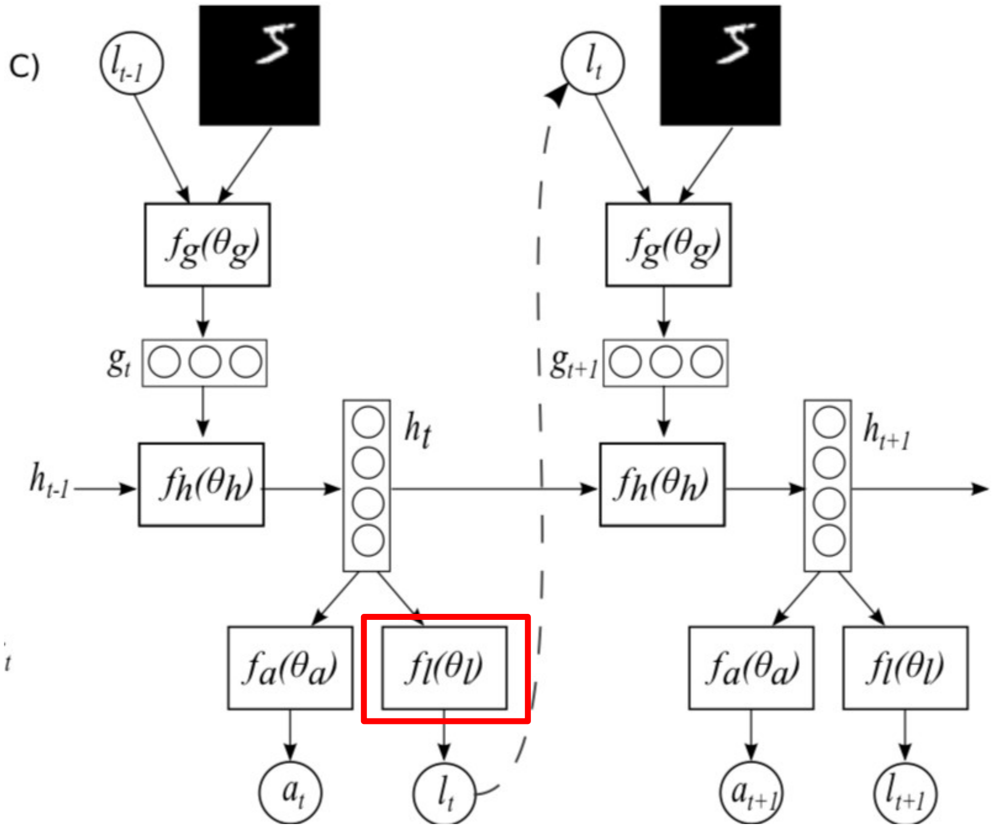
The Model

Sensor

Internal State

Action

Reward



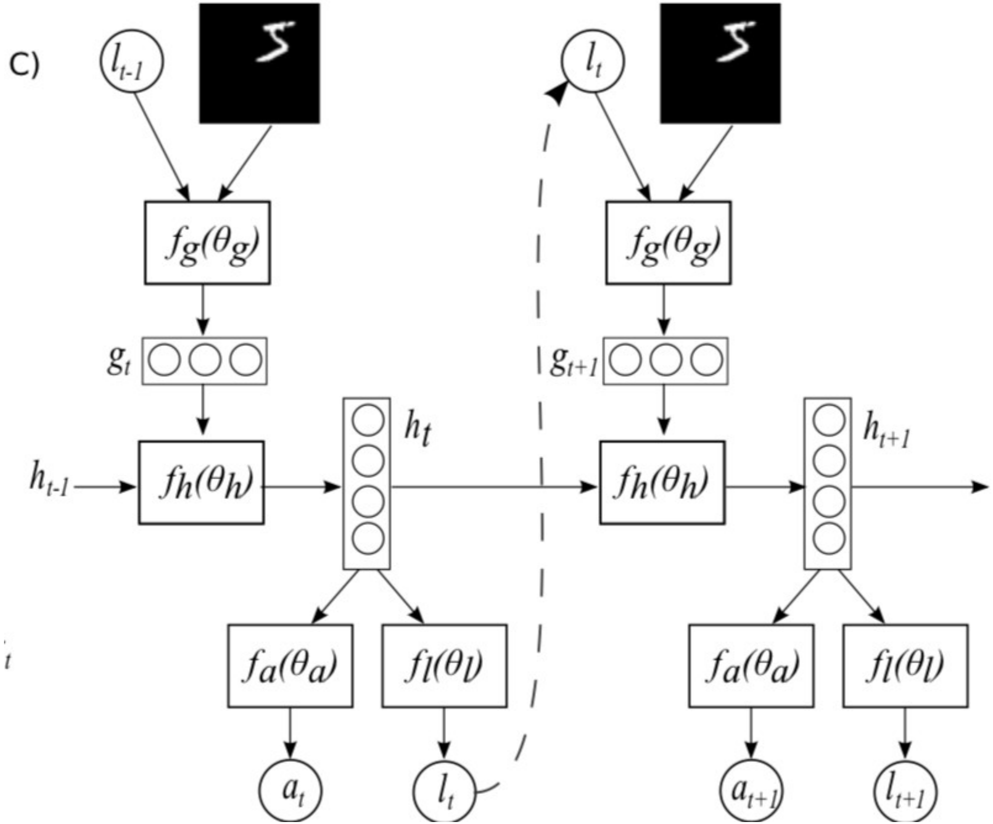
The Model

Sensor

Internal State

Action

Reward



Training

Maximize reward over distribution:

$$J(\theta) = \mathbb{E}_{p(s_{1:T};\theta)}[\sum r_t]$$

REINFORCE rule:

$$\nabla_\theta J = \sum_{t=1}^{T} \mathbb{E}_{p(s_{1:T};\theta)} \left[\nabla_\theta \log \pi(u_t|s_{1:t};\theta)R\right] \approx \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} \boxed{\nabla_\theta \log \pi(u_t^i|s_{1:t}^i;\theta)} R^i$$

Williams et al. Simple statistical gradient-following algorithms for connectionist reinforcement learning

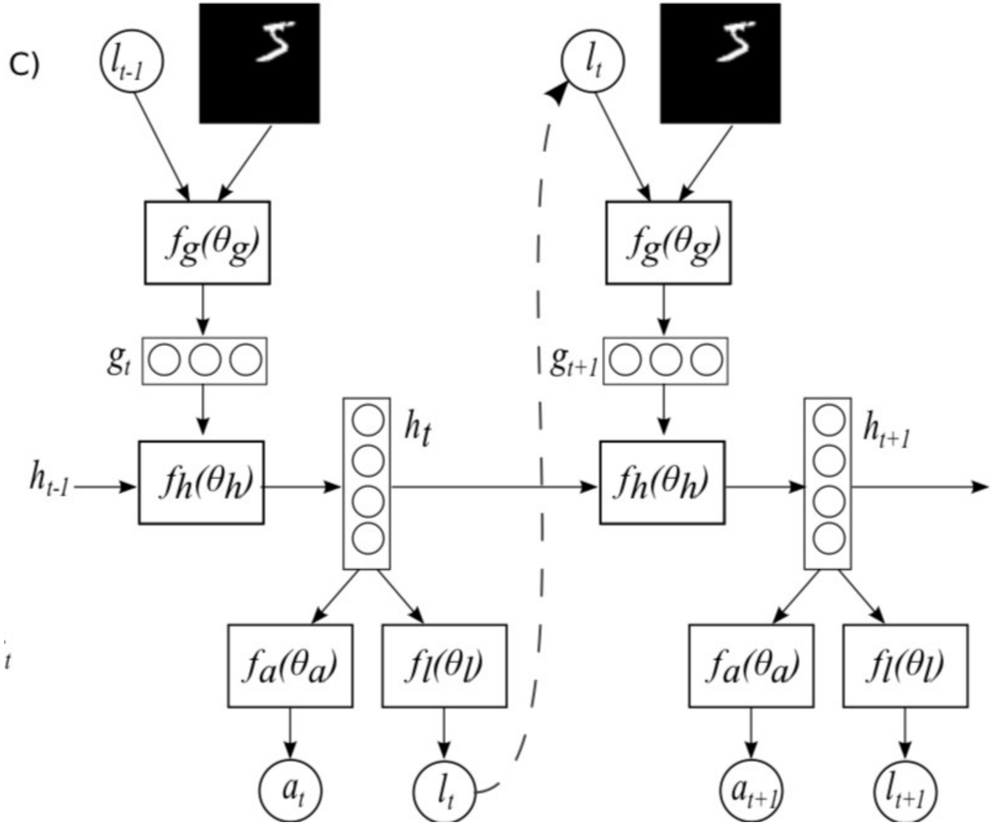
The Model

Sensor

Internal State

Action

Reward



Experiments

Glimpse network 2 FC layers (128 units in each), 1 FC layer (256 units)

Location network $f_l(h)$ two component Gaussian with fixed variance

Action network $f_a(h)$ linear softmax classifier (10 classes)

Internal state:

Classification: $Rect(Linear(h_{t-1}) + Linear(g_t))$

Game: LSTMs

Datasets

(a) Translated MNIST inputs.

(b) Cluttered Translated MNIST inputs.

60 x 60

100 x 100

Results

(a) 28x28 MNIST

Model	Error
FC, 2 layers (256 hiddens each)	1.69%
Convolutional, 2 layers	1.21%
RAM, 2 glimpses, 8×8, 1 scale	3.79%
RAM, 3 glimpses, 8×8, 1 scale	1.51%
RAM, 4 glimpses, 8×8, 1 scale	1.54%
RAM, 5 glimpses, 8×8, 1 scale	1.34%
RAM, 6 glimpses, 8×8, 1 scale	1.12%
RAM, 7 glimpses, 8×8, 1 scale	**1.07**%

(b) 60x60 Translated MNIST

Model	Error
FC, 2 layers (64 hiddens each)	6.42%
FC, 2 layers (256 hiddens each)	2.63%
Convolutional, 2 layers	1.62%
RAM, 4 glimpses, 12×12, 3 scales	1.54%
RAM, 6 glimpses, 12×12, 3 scales	**1.22**%
RAM, 8 glimpses, 12×12, 3 scales	**1.2**%

Results

(a) 60x60 Cluttered Translated MNIST

Model	Error
FC, 2 layers (64 hiddens each)	28.58%
FC, 2 layers (256 hiddens each)	11.96%
Convolutional, 2 layers	8.09%
RAM, 4 glimpses, 12×12, 3 scales	4.96%
RAM, 6 glimpses, 12×12, 3 scales	4.08%
RAM, 8 glimpses, 12×12, 3 scales	4.04%
RAM, 8 random glimpses	14.4%

(b) 100x100 Cluttered Translated MNIST

Model	Error
Convolutional, 2 layers	14.35%
RAM, 4 glimpses, 12×12, 4 scales	9.41%
RAM, 6 glimpses, 12×12, 4 scales	8.31%
RAM, 8 glimpses, 12×12, 4 scales	8.11%
RAM, 8 random glimpses	28.4%

Results

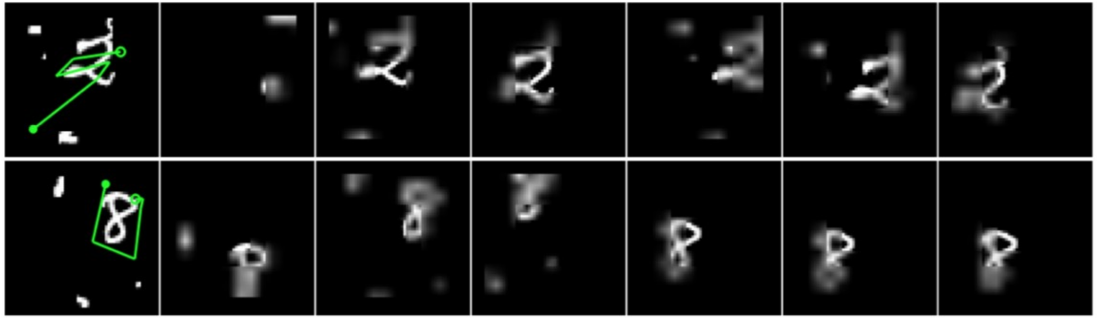
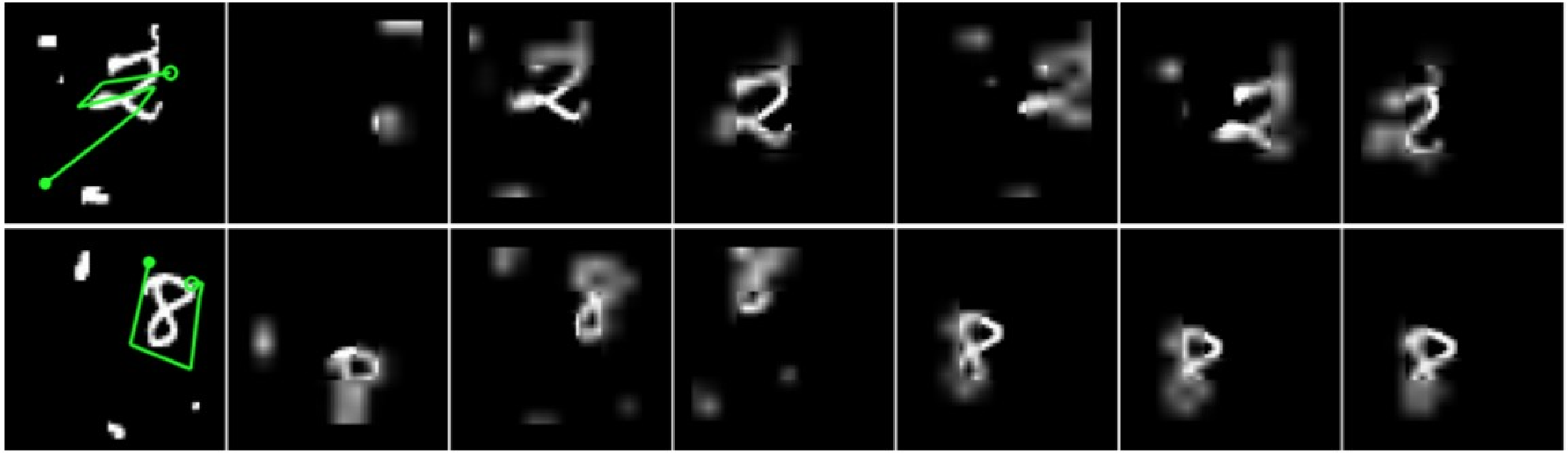
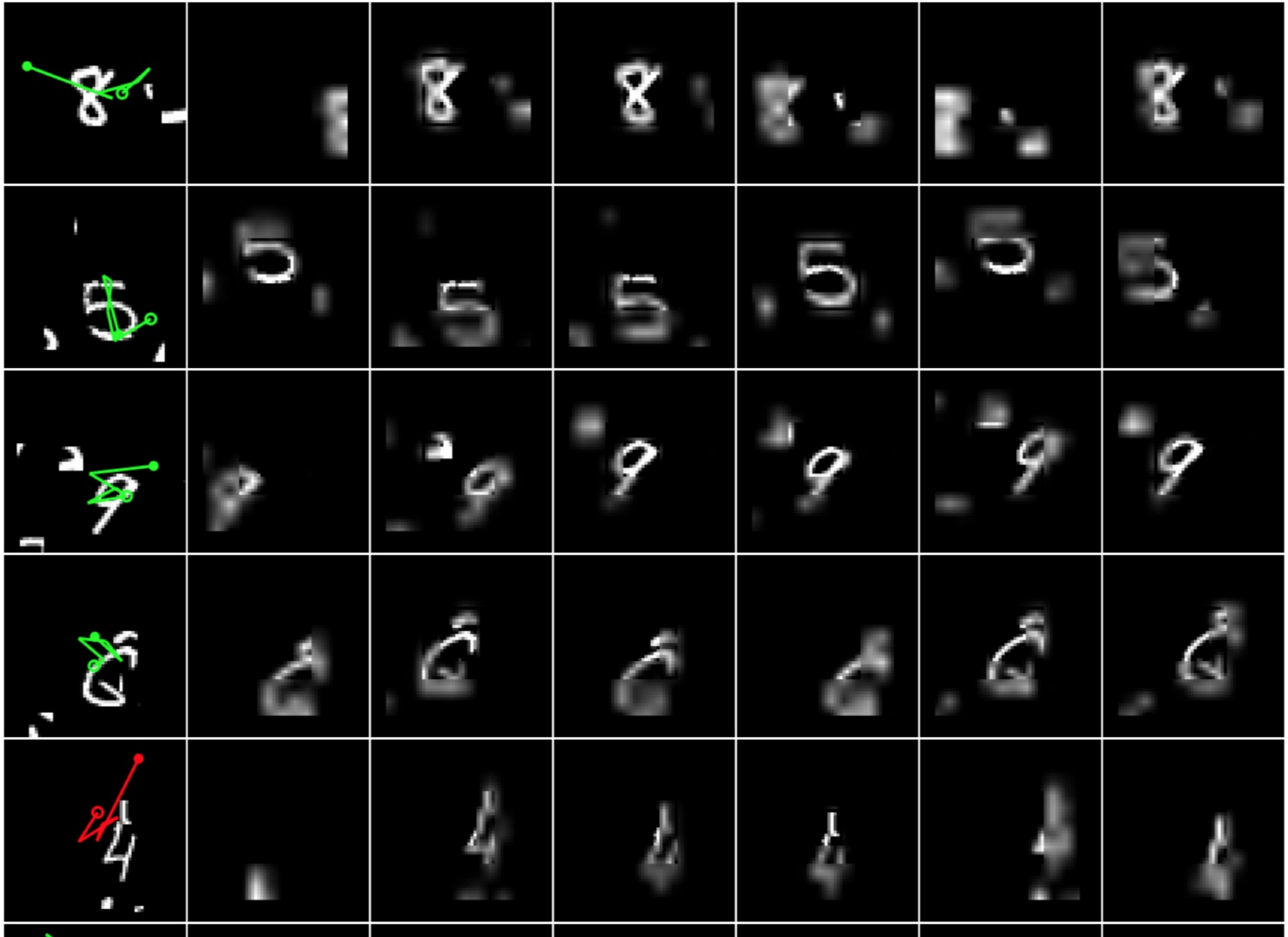


Figure 3: Examples of the learned policy on 60 × 60 cluttered-translated MNIST task. Column 1: The input image with glimpse path overlaid in green. Columns 2-7: The six glimpses the network chooses. The center of each image shows the full resolution glimpse, the outer low resolution areas are obtained by upscaling the low resolution glimpses back to full image size. The glimpse paths clearly show that the learned policy avoids computation in empty or noisy parts of the input space and directly explores the area around the object of interest.

Results





http://www.cs.toronto.edu/~vmnih/docs/attention.mov

Discussion Points

1. Alternative methods to downscaling
   a. E.g. number of ducts and locations
2. Termination of glimpses or reset?
   a. Possibly exploring multiple locations at same time
3. Visualizing the modeled internal state to the actual environment to see what has been learned from glimpses

Thank you

shazia.akbar@sunnybrook.ca

RNNs

Traditionally, input is received one at a time:

The cow jumped over the moon

{over, into, at, to}

Series of LSTM units enabling the network to look into the past (for a predetermined time)