First Step Toward Model-free, Anonymous Object Tracking with Recurrent Neural Networks

mingzailao

2016-09-08 Thu

Outline

Introduction

Goal

Build a model that can track an anonymous object in an image sequence

Key

1. Integrates convolutional network with recurrent network.
2. Deploys attention at multiple representation layers.

Filtering-based Visual Object Tracking

In filtering-based object tracking, it is natural to establish a probabilistic graphical model, or often referred to as a state-space model, by defining

1. Observationmodel:
$$p(\phi(x_t)|z_t)$$
2. Transition model:
$$p(z_t|z_{t-1})$$

where ϕ_{x_t} and z_t are the observation and the latent state at time

Filtering-based Visual Object Tracking

Kalman filter

Kalman filter assume that both the observation and transition models are Gaussian distributions such that

$$\phi(x_t)|z_t \sim \mathcal{N}(W_x z_t, C_x), z_t|z_{t-1} \sim \mathcal{N}(W_z z_{t-1}, C_z)$$

where $\mathcal{N}(\mu, C)$ is a Guassian distribution with its mean μ and convariance matrix C. once the model is defined, the goal is to infer the following posterior distribution:

$$p(z_t|\phi(x_1), \phi(x_2) \cdots, \phi(x_t)) \tag{1}$$

Filtering-based Visual Object Tracking

Kalman filter

In other words, we are try to find the distribution over the potential object location in the t-th video frame given the detected objects(represented by the feature vectors) up to the $(t-1)$-th frame.

Tracking-by-Detection

Key

It is a discriminative model that directly approximates the posterior distrobution over the underlying location/state of the object in Eq. (1)

Limitations

1. drifting
2. occlusion

Tracking-by-Detection

Reason

Assuming Markov property over the poterior distribution in Eq. (1), meaning

$$p(z_t|\phi(x_1), \phi(x_2), \cdots, \phi(x_t)) \approx p(x_t|\phi(x_{t-1})) \qquad (2)$$

Because the tracking model considers only two consecutive frame(or more precisely concentrates heavily on a latest pair of consecutive frames only), it is not possible for the model to adjust dor accumulated error over many subsequent frames.

Visual Object Tracking with Deep Neural Networks

Aim

Building a system that avoids the limitations of the conventional visual tracking systems.

Nets

Deep Recurrent Network

1. input : raw video frames of pixel intensities
2. output: coordinates of a bounding box of an object being tracked for each frame

Visual Object Tracking with Deep Neural Networks

Mathematic

The proposed model factorizes the full tracking probability into:

$$p(z_1, z_2, \cdots, z_t | x_1, x_2, \cdots, x_t) = \prod_{t=1}^{T} p(z_t | z_{<t}, x_{\leq t}) \tag{3}$$

where z_t and x_t are the location of the object and on input frame, respectively, at time t.

Model Description

Technological Process

At each time step t, an input frame x_t is first processed by a convolutional network,

$$\phi(x_t) = conv_{\theta_c}(m(x_t, \tilde{z}_{t-1})) \tag{4}$$

1. $conv_{\theta_c}(\cdot)$: a convolutional network with its paramaters θ_c.
2. $m(\cdot, \cdot)$: a preprocessing routine for the raw frame.
3. $\tilde{z}_{t-1}$: the predicted location of an object from the previous frame x_{t-1}.

Model Description

Technological Process

This feature vector of the input frame is fed into a recurrent neural network. The recurrent neural network updates its internal memory vector h_t based on the previous memory vector h_{t1}, previous location of an object $\tilde{z}_{t-1}$ and the current frame $\phi(x_t)$:

$$h_t = rec_{\theta_r}(h_{t-1}, \tilde{z}_{t-1}, \phi(x_t)) \tag{5}$$

1. rec_{θ_r} : a recurrent activation function such as gated recurrent units, LSTM units or a simple sigmoid function, parametrized with paramaters θ_r.

Model Descripteion

Technological Process

This formulation lets the recurrent neural network to summarize the history of predicted locations $z_{<t}$ and input frames $x_{\leq t}$ up to time step t.

Model Description

Technological Process

With the new updated memory state h_t, the recurrent neural network computes the predictive distribution over the object's location(See Eq. (1)).

$$p(z_t|z_{<t}, x_{\leq t}) = out_{\theta_o}(h_t) \tag{6}$$

1. θ_o: a set of paramaters defining the output neural network.

We take the mean of this predictive distribution as a predicted location $\tilde{z}_t$ at time t:

$$\tilde{z}_t = \mathbb{E}[z|z_{<t}, x_{\leq t}] \tag{7}$$

Model Description

Preprocessing Input Frame x_t

This part explain the method that got $m(x_t, \tilde{z}_{t-1})$ in Eq. (4)
The most obvious and straightforward choice is to simply have an identity function: In this case, we use an identity function to preprocess each input frame x_t:

$$m(x_t, \tilde{z}_{t-1}) = x_t \tag{8}$$

One Possible Choice : attentive weight scheme

Weighting each pixel of the raw frame x_t such that a region surrounding the predicted location of an object in the previous frame is given higher weights.

Preprocessing Input Frame x_t

The Recurrent Network Outputs

1. (x_0, y_0) : top-left corner.
2. (x_1, y_1) : bottom-right corner.
3. s : log-scale .
4. r : log-ratio between the stride and the image size.
5. a : log-amplitude.

Preprocessing Input Frame x_t

The weights of each pixel

Given these outputs, we weight each pixel using a mixture of NN Guassians. Each Guassian (i, j) is centered at

$$\left(\frac{x_{0+}x_1}{2}+\left(i-\frac{N}{2}-0.5\right)\exp(r)K, \frac{y_0+y_1}{2}+\left(j-\frac{N}{2}-0.5\right)\exp(r)K\right) \tag{9}$$

and has the standard deviation of $\exp(s)\frac{K}{2}$, K corresponds to the width or height of a frame. These Gaussians are used to form a mask $G(z_{t1})$ which is used as

$$m(x_t, z_{t-1}) = x_t \cdot G(z_{t-1}) \tag{10}$$

Training

Optimizier

Stochastic gradient descent (SGD)

Steps

1. Select a random background image from a large set of image.
2. Randomly choose a shape of an object from a predefined set of generic shapes.
3. Create a sequence of frames by randomly moving the selected object with cluttered background and foreground.
4. (Optional) Add various types of noise, including motion and scale change of both object and clutters.

Training

Training examples

a pair of a video clip, which contains a randomly chosen background and a moving shape, and a sequence of ground-truth locations i.e. $((x_1, z_1^*), \cdots, (x_T, z_T^*))$.

Log-likelihood

$$\mathcal{L}(\theta_c, \theta_r, \theta_o) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=k+1}^{T} \log p(z_t^n = z_t^{*,n} | z_{<t}^{*,n}, x_{\leq t}^n) \tag{11}$$

Training

Another training criterion is possible

If our prediction $\tilde{z}_t$ as each step t is a differentiable function, we let the model freely track an object given a training video sequence and maximize the log-probability of the ground-truth location only at the last frame

$$\mathcal{L}(\theta_c, \theta_r, \theta_o) = \frac{1}{N} \sum_{n=1}^{N} \log p(z_T^n = z_T^{*,n} | \tilde{z}_{<T}^n, x_{\leq T}^n) \tag{12}$$

For preliminary experiments, using this strategy.

Training

A small problem for the strategy

There is no guarantee that any intermediate prediction made by the model correspond to the correct object location.

Solution: adding the auxiliary cost

$$\mathcal{L}(\theta_c, \theta_r, \theta_o) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=k+1}^{T} \log p(z_t^n = z_t^{*,n} | \tilde{z}_{<t}^n, x_{\leq t}^n) \tag{13}$$

Training

Predicting the $\tilde{z}_t$

In that course, the model predict two points $z_t = [x_0, y_0, x_1, y_1]$ in the input frame. We use a Gaussian distribution with an fixed, identity covariance matrix, whose mean is computed from h_t, In order to reduce variance, we do not sample from this distribution, but simply take the mean as the prediction:

$$\tilde{z}_t = \mathbb{E}[z_t | \tilde{z}_{<T}, x_{\leq T}] \tag{14}$$

Characteristics

First

The proposed model is trained end-to-end, including object representation extraction, object detection and object tracking.

Second

The proposed model works with anonymous objects by design.

Third

Training is done fully off-line.

Data Generation

Source

- All the datasets are based on the cluttered MNIST(`https://github.com/deepmind/mnist-cluttered`).
- Each video sequence used for training consists of 20 frames, and each frame is 100 Œ 100 large.
- The cluttered MNIST was chosen as a basis of generating further datasets, as one of the most important criterion we aim to check with the proposed approach is the robustness to noise.

Data Generation

Noise

In order to make sure that these clutters acts as both background noise and as objects hindering the sight of the models, we put some clutters in a background layer and the others in a foreground layer (overshadowing the target object.) Furthermore, the clutters move rather than stay in the initial positions to make it more realistic.

Data Generation

Object Moving

The target has a random initial velocity (v_{x_0}, v_{y_0}) and position (x_0, y_0), at each time frame, the position is changed by $(\Delta x_t, \Delta y_t) = (kv_{x,t-1}, kv_{y,t-1})$.

1. k : hyper-parameter (0.1 in our exper- iments) correlated to the frame rate.
2. $(\Delta v_{x,t}, \Delta v_{y,t}) \sim \mathcal{N}(0, v^{'} I)$ where $v' = 0.1$

Data Generation

Other Transformations

For example, at each time step, the target changes its scale by a random factor $f = p\exp(\tilde{f})$, where $\tilde{f} \sim \mathcal{U}(-0.5, 0.5)$ and $p = 0.1$ controls the magnitude of scale change.

Evaluate Model by Two Dataset

We evaluate our model on two different cases

MNIST-Single-Same

There is only a single digit moving around in each video sequence.

MNIST-Multi-Same

Contains frames of which each contains more than one digits. More specifically, we generate each video sequence such that there are two digits simultaneously moving around.

Novel Test Digit Classes

Test Datasets

As our goal is to build an anonymous object tracking system. We evaluate a trained model with sequences containing objects that do not appear during training time.
More specifically, we test the models on two sets of sequences containing one or two MNIST-2 digits, where one MNIST-2 digit is created by randomly overlapping two randomly selected normal MNIST digits on top of each other. We call these datasets MNIST-Single-Diff and MNIST-Multi-Diff, respectively.

Generalization to Longer Video Sequence

In all the cases, we evaluate a trained model on test sequences that are longer than the training sequences.

- some recent findings suggest that on certain tasks recurrent neural networks fail to generalize to longer test sequences.

We vary the lengths of test sequences among {20, 40, 80, 160}, while all the models are trained with 20-frame-long training sequences.

Models and Training

Train and Test

MNIST-{Single,Multi}-{Same,Diff}

5 models:

- RecTracker-ID(with no weight matrix,The tracker has a full, unadjusted view of the whole frame)
- RecTracker-Att-1(N=1)
- RecTracker-Att-3(N=3)
- ConvTracker
- KerCorrTracker

Network Architectures

Convolutional Network

1. Using a single convolutional layer with 3210 $\times$ 10 filters. These filters are applied with stride 5 to the input frame.
2. Do not use any pooling.
3. This convolutional layer is immediately followed by an element-wise tanh.
4. In the case of ConvTracker, a fully-connected layer with 200 tanh units follows the convolutional layer.

Notes

This fully-connected layer also receives as the input the predicted locations of the four preceding frames.

Network Architectures

Recurrent Network

Using 200 gated recurrent units to build a recurrent network, At each time step, the activation of the convolutional layer and the predicted object location $\tilde{z}_{t-1}$ in the previous frame are fed into the recurrent network.

Training Details

Epoch

50 or until the training cost stops improving

Training samples

Using a training set of 3,200,000 randomly-generated examples

Optimizier

RMSProp

minibatches

32

Result and Analysis

Evaluation metric

Intersection-over-union (IOU)

$$IOU(z_t^*, \tilde{z}_t) = \frac{|M^* \cap \tilde{M}|}{|M^* \cup \tilde{M}|} \tag{15}$$

Where M^* and $\tilde{M}$ are binary masks whose pixels inside a bounding box (either ground-truth * or predicted $\tilde{}$) are 1 and otherwise 0.

A higher IOU implies better tracking quality, and it is bounded between 0 and 1.
For each video sequence, we compute the average IOU across all the frames.