Hierarchical Multi-scale Attention Networks for Action Recognition

Shiyang Yan[a,b,*], Jeremy S. Smith[a], Wenjin Lu[b], Bailing Zhang[b]

[a] *Electrical Engineering and Electronic, University of Liverpool, Liverpool, United Kingdom*
[b] *Department of Computer Science and Software Engineering, Xi'an Jiaotong-liverpool University, SuZhou, JiangSu Province, China*

Abstract

Recurrent Neural Networks (RNNs) have been widely used in natural language processing and computer vision. Amongst them, the Hierarchical Multi-scale RNN (HM-RNN), a recently proposed multi-scale hierarchical RNN, can automatically learn the hierarchical temporal structure from data. In this paper, we extend the work to solve the computer vision task of action recognition. However, in sequence-to-sequence models like RNN, it is normally very hard to discover the relationships between inputs and outputs given static inputs. As a solution, the attention mechanism can be applied to extract the relevant information from the inputs thus facilitating the modeling of the input-output relationships. Based on these considerations, we propose a novel attention network, namely Hierarchical Multi-scale Attention Network (HM-AN), by incorporating the attention mechanism into the HM-RNN and applying it to action recognition. A newly proposed gradient estimation method for stochastic neurons, namely Gumbel-softmax, is exploited to implement the temporal boundary detectors and the stochastic hard attention mechanism. To amealiate the negative effect of the temperature sensitivity of the Gumbel-softmax, an adaptive temperature training method is applied to improve the system performance. The experimental results demonstrate the improved effect of HM-AN over LSTM with

*Corresponding author

Email addresses: `Shiyang.Yan@xjtlu.edu.cn` (Shiyang Yan), `J.S.Smith@liverpool.ac.uk` (Jeremy S. Smith), `Wenjin.Lu@xjtlu.edu.cn` (Wenjin Lu), `Bailing.Zhang@xjtlu.edu.cn` (Bailing Zhang)

attention on the vision task. Through visualization of what has been learnt by the network, it can be observed that both the attention regions of the images and the hierarchical temporal structure can be captured by a HM-AN.

Keywords: Action recognition, Hierarchical multi-scale RNNs, Attention mechanism, Stochastic neurons.

1. Introduction

Action recognition in videos is a fundamental task in computer vision. Recently, with the rapid development of deep learning, and in particular, deep convolutional neural networks (CNNs), a number of models [1] [2] [3] [4] have been proposed for image recognition. However, for video-based action recognition, a model should accept inputs with variable length and generate the corresponding outputs. This special requirement makes the conventional CNN model that caters for a one-versus-all classification unsuitable.

For decades RNNs have been applied to sequential applications, often with good results. However, a significant limitation of the vanilla RNN models, which strictly integrate state information over time, is the vanishing gradient effect [5]: the ability to back propagate an error signal through a long-range temporal interval becomes increasingly impossible in practice. To mitigate this problem, a class of models with a long-range dependencies learning capability, called Long Short-Term Memory (LSTM), was introduced by Hochreiter and Schmidhuber [6]. Specifically, LSTM consists of memory cells, with each cell containing units to learn when to forget previous hidden states and when to update hidden states with new information.

Much sequential data often has a complex temporal structure which requires both hierarchical and multi-scale information to be modeled properly. In language modeling, a long sentence is often composed of many phrases which further can be decomposed into words. Meanwhile, in action recognition, an action category can be described by many sub-actions. For instance, 'long jump' contains 'running', 'jumping' and 'landing'. As stated in [7], a promising approach to model such hierarchical representation is the multi-scale RNN. One popular approach of implementing multi-scale RNNs is to treat the hierarchical timescales as pre-defined parameters. For example, Wang et al. [8] implemented a multi-scale architecture by building a multiple layers LSTM in which higher layers skip several time steps. In their paper, the skipped number of time steps is the parameter to be pre-defined. However, it is often impractical to pre-define such timescales without learning, which also leads to a poor generalization capability. Chung et al. [7] proposed a novel RNN structure, Hierarchical Multi-scale Recurrent Neural Network (HM-RNN), to automatically learn time boundaries from data. These temporal boundaries are similar to rules described by discrete variables inside RNN cells. Normally, it is difficult to implement training algorithms for discrete variables. Popular approaches include unbiased estimator with the aid of REINFORCE [9]. In this paper, we re-implement the HM-RNN by applying the recently proposed Gumbel-sigmoid function [10] [11] to realize the training of stochastic neurons due to its efficiency [12].

In the general RNN framework for sequence-to-sequence problems, the input information is treated uniformly without discrimination on the different parts. This will result in the fixed length of intermediate features and hence subsequent sub-optimal system performance. The practice is in sharp contrast to the way humans accomplish sequence processing tasks. Humans tend to selectively concentrate on a part of information and at the same time ignores other perceivable information. The mechanism of selectively focusing on relevant contents in the representation is called attention. The attention based RNN model in machine learning was successfully applied in natural language processing (NLP), and more specifically, in neural translation [13]. For many visual recognition tasks, different portions of an image or segments of a video have unequal importance, which should be selectively weighted with attention. Xu et al. [14] systematically analyzed stochastic hard attention and deterministic soft attention models and applied them in image captioning tasks, with improved results compared with other RNN-like algorithms. The hard attention mechanism requires a stochastic neuron which is hard to train using the conventional back propagation algorithm. They applied REINFORCE [9] as an estimator to implement hard attention for image captioning.

The REINFORCE is an unbiased gradient estimator for stochastic units, however, it is very complex to implement and often has high gradient variance during training [12]. In this paper, we study the applicability of Gumbel-softmax [10] [11] in hard attention because Gumbel-softmax is an efficient way to estimate discrete units during the training of neural networks. To mitigate the problem of temperature sensitivity in Gumbel-softmax, we apply an adaptive temperature scheme [12] in which the temperature value is also learnt from the data. The experimental results verify that the adaptive temperature is a convenient way to avoid manual searching for the parameter. Additionally, we also test the deterministic soft attention [14] [15] and stochastic hard attention implemented by REINFORCE-like algorithms [16] [17] [14] in action recognition. Combined with HM-RNN and the two types of attention models, we systematically evaluate the proposed Hierarchical Multi-scale Attention Networks (HM-AN) for action recognition in videos, with improved results.

Our main contributions can be summarized as follows:

- We propose a Hierarchical Multi-scale Attention Network (HM-AN) by implementing HM-RNN with Gumbel-sigmoid to realize the discrete boundary detectors.
- We also propose four methods of realizing an attention mechanism for action recognition in videos, with improved results over many baselines.
- By incorporating Gumbel-softmax and Gumbel-sigmoid into HM-RNN, we make the stochastic neurons in the networks end-to-end trainable by error back propagation.
- For the hard attention model based on Gumbel-softmax, we propose to use an adaptive temperature for the Gumbel-softmax, which generates much improved results over a constant temperature value.

- Through visualization of the learnt attention regions, the boundary detectors of HM-AN and the adaptive temperature values, we provide insights for further research.

2. Related Works

2.1. Hierarchical RNNs

The modeling of hierarchical temporal information has long been an important topic in many research areas. The most notable model is LSTM proposed by Hochreiter and Schmidhuber [6]. LSTM employs the multi-scale updating concept, where the hidden units' update can be controlled by gating such as input gates or forget gates. This mechanism enables the LSTM to deal with long term dependencies in the temporal domain. Despite this advantage, the maximum time steps are limited to within a few hundred because of the leaky integration which makes the memory for long-term gradually diluted [7]. Actually, the maximum time steps in video processing is several dozen frames which makes the application of LSTM in video recognition very challenging.

To alleviate this problem, many researchers tried to build a hierarchical structure explicitly, for instance, Hierarchical Attention Networks (HAN) proposed in [8], which is implemented by skipping several time steps in the higher layers of the stacked multi-layer LSTMs. However, the number of time steps to be skipped is a pre-defined parameter. How to choose these parameters and why to choose a certain number are unclear.

More recent models like clockwork RNN [18] partitioned the hidden states of a RNN into several modules with different timescales assigned to them. The clockwork RNN is more computationally efficient than the standard RNN as the hidden states are updated only at the assigned time steps. However, finding the suitable timescales is challenging which makes the model less applicable.

To mitigate the problem, Chung et al. [7] proposed the Hierarchical Multi-scale Recurrent Neural Network (HM-RNN). The HM-RNN is able to learn the temporal boundaries from data, which allows the RNN model to build a hierarchical structure and enables long-term dependencies automatically. However, the temporal boundaries are stochastic discrete variables which are very hard to train using the standard back propagation algorithm.

A popular approach to train the discrete neurons is the REINFORCE-like [19] algorithms. This is an unbiased estimator but often with high gradient variance [7]. The original HM-RNN applied a straight-through estimator [9] because of its efficiency and simplicity in implementation. Instead, in this paper, we applied the more recent Gumbel-sigmoid [10] [11] to estimate the stochastic neurons. This is much more efficient than other approaches and achieved state-of-the-art performance among many other gradient estimators [10].

2.2. Attention Mechanism

One important property of human perception is that we do not tend to process a whole scene, in its entirety, at once. Instead humans pay attention selectively on parts of the visual scene to acquire information where it is needed [16]. Different attention models have been proposed and applied in object recognition and machine translation. Mnih et al. [16] proposed an attention mechanism to represent static images, videos or as an agent that interacts with a dynamic visual environment. Also, Ba et al. [17] presented an attention-based model to recognize multiple objects in images. These two models are all with the aid of REINFORCE-like algorithms.

The soft attention model was proposed for the machine translation problem in NLP [13], and Xu et al. [14] extended it to image caption generation as the task is analogous to 'translating' an image into a sentence. Specifically, they built a stochastic hard attention model with the aid of REINFORCE and a deterministic soft attention model. The two attention mechanisms were applied to the image captioning task, with good results. Subsequently, Sharma et al. [15] built a similar model with soft attention applied to action recognition from videos.

There are a number of subsequent works on the attention mechanism. For instance, in [20], the attention model is utilized for video description generation by softly weighting the visual features extracted from the frames in each video. Li et al. [21] combined a convolutional LSTM [22] with the soft attention mechanism for video action recognition and detection. Teh et al. [23] extended the soft attention into CNN networks for weakly supervised object detection.

One important reason for applying soft attention instead of hard version is that the stochastic hard attention mechanism is difficult to train. Although the REINFORCE-like algorithms [19] are unbiased estimators to train stochastic units, their gradients have high variants. To solve this problem, recently, Jang et al. [10] proposed a novel categorical re-parameterization technique using the Gumbel-softmax distribution. The Gumbel-softmax is a superior estimator for categorical discrete units [10]. It has been proved to be efficient and has high performance [10].

2.3. Action Recognition

Action recognition has received significant attention recently. Most approaches focused on the design of novel features, trajectory-based features [24], CNN based features [25] [26] [27]. For example, [28] built a simple representation to explicitly model the motion relationships, with outstanding results with popular classifier like SVM on several benchmark datasets.

Some researches built model to better exploit these powerful features by fusing operation. For instance, [29] proposed a regularized Deep Neural Network (DNN) to fuse the CNN features, the trajectory features and the audio features for action categorization, with promising results. [26] [27] fused CNN features and motion features for better recognized action categories in video.

RNNs have been popular for speech recognition [30], image caption generation [14], and video description generation [20]. There have also been efforts made for the application of LSTM RNNs in action recognition. For instance, [31] proposed an end-to-end training system using CNN and RNN deep both in space and time to recognize activities in video. [32] also explicitly models the video as an ordered sequence of frames using LSTM. Most of the previous work treat image features extracted from CNNs as static inputs to a RNN to generate action labels at each frame. The attention mechanism is able to discriminate the relevant features from these static inputs and can improve the system performance. On the other hand, the interpretation of CNN features will be much easier if the attention mechanism can be applied to action recognition because the attention mechanism automatically focuses on specific regions to facilitate the classification.

In this paper, we re-implement the HM-RNN to capture the hierarchical structure of temporal information from video frames. By incorporating the HM-RNN with both stochastic hard attention and deterministic soft attention, the long-term dependencies of video frames can be captured.

Research related to ours also includes the attention model proposed by Xu et al. [14] and [33]. [14] first applied both stochastic hard attention and deterministic soft attention mechanisms for spatial locations of images for image captioning. [33] instead used weighting on image patches to implement region-level attention. In this paper, similar to [14], both stochastic hard attention and deterministic soft attention are studied. However, when implementing hard attention, [14] borrowed the idea of REINFORCE whilst we also propose to apply the more recent Gumbel-softmax to estimate discrete neurons in the attention mechanism.

3. The proposed methods

In this section, we first re-visit the HM-RNN structure proposed in [7], then introduce the proposed HM-AN networks, with details of Gumbel-softmax and Gumbel-sigmoid to estimate the stochastic discrete neurons in the networks.

3.1. HM-RNN

HM-RNN was proposed in [7] to better capture the hierarchical multi-scale temporal structure in sequence modeling. HM-RNN defines three operations

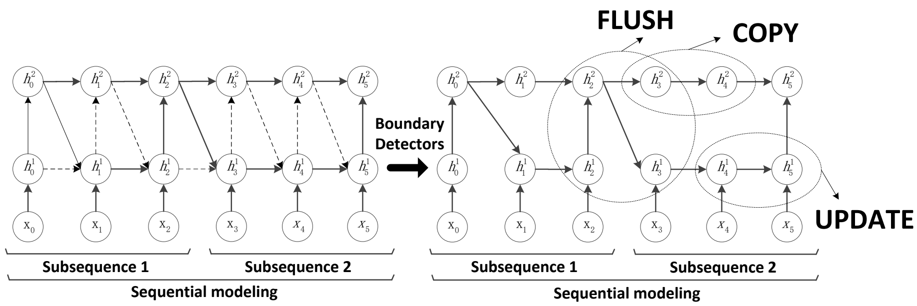

Figure 1: Network Structure: After the networks discover the implicit boundary relations of the multi-scale property, boundary detectors can set the networks into an explicit multi-scale architecture.

depending on the boundary detectors: UPDATE, COPY and FLUSH. The selection of these operations is determined by the boundary state z_t^{l-1} and z_{t-1}^l, where l and t represent the current layer and time step, respectively:

$$\begin{aligned} &UPDATE, && z_{t-1}^l = 0 \text{ and } z_t^{l-1} = 1; \\ &COPY, && z_{t-1}^l = 0 \text{ and } z_t^{l-1} = 0; \\ &FLUSH, && z_{t-1}^l = 1. \end{aligned} \tag{1}$$

The updating rules for the operation UPDATE, COPY and FLUSH are defined as follows:

$$c_t^l = \begin{cases} f_t^l \odot c_{t-1}^l + i_t^l \odot g_t^l, & UPDATE \\ c_{t-1}^l, & COPY \\ i_t^l \odot g_t^l, & FLUSH \end{cases} \tag{2}$$

The updating rules for hidden states are also determined by the pre-defined operations:

$$h_t^l = \begin{cases} h_{t-1}^l, & COPY \\ o_t^l \odot c_t^l, & otherwise \end{cases} \tag{3}$$

The (i, f, o) indicate the input, forget and output gate, respectively. g is called the 'cell proposal' vector. One of the advantages of HM-RNN is that the updating operation (UPDATE) is only executed at certain time steps instead of all the time, which significantly reduces the computation cost.

The COPY operation simply copies the cell memory and hidden state from the previous time step to the current time step in the upper layers until the end of a subsequence, as shown in Fig. 1. Hence, the upper layer is able to capture coarser temporal information. Also, the boundaries of subsequence are learnt from the data which is a big improvement over other related models. To start a new subsequence, the FLUSH operation needs to be executed. The FLUSH operation firstly forces the summarized information from the lower layers to be merged with the upper layers, then re-initialize the cell memories for the next subsequence.

In summary, the COPY and UPDATE operations enable the upper and lower layers to capture information on different time scales, thus realizing a multi-scale and hierarchical structure for a single subsequence. The FLUSH operation is able to summarize the information from the last subsequence and forward them to the next subsequence, which guarantee the connection and coherence between parts within a long sequence.

The values of gates (i, f, o, g) and the boundary detector z are obtained by:

$$\begin{pmatrix} i_t^l \\ f_t^l \\ o_t^l \\ g_t^l \\ z_t^l \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \\ hardsigm \end{pmatrix} f_{slice} \begin{pmatrix} s_t^{recurrent(l)} + \\ s_t^{top-down(l)} + \\ s_t^{bottom-up(l)} + \\ b_l \end{pmatrix} \tag{4}$$

where

$$s_t^{recurrent(l)} = U_l^l h_{t-1}^l \tag{5}$$

$$s_t^{top-down(l)} = U_{l+1}^l (z_{t-1}^l \odot h_{t-1}^{l+1}) \tag{6}$$

$$s_t^{bottom-up(l)} = W_{l-1}^l (z_t^{l-1} \odot h_t^{l+1}) \tag{7}$$

and the hardsigm is estimated using the Gumbel-sigmoid which will be is explained later. In the equation, the U_l and W_l are the weight matrices, and b_l is the bias matrix.

3.2. HM-AN

The sequential problems inherent in action recognition and image captioning in computer vision can be tackled by a RNN-based framework. As previously explained, HM-RNN is able to learn the hierarchical temporal structure from data and enable long-term dependencies. This inspired our proposal of the HM-AN model.

As attention has been proved very effective in action recognition [15], in HM-AN, to capture the implicit relationships between the inputs and outputs in sequence to sequence problems, we apply both hard and soft attention mechanisms to explicitly learn the important and relevant image features regarding the specific outputs. A more detailed explanation is as follows.

3.2.1. Estimation of Boundary Detectors

In the proposed HM-AN, the boundary detectors z_t are estimated with Gumbel-sigmoid, which is derived directly from the Gumbel-softmax proposed in [10] and [11].

The Gumbel-softmax replaces the argmax in the Gumbel-Max Trick [34] [35] with the following Softmax function:

$$y_i = \frac{exp(log(\pi_i + g_i)/\tau)}{\sum_{j=1}^{k} exp(log(\pi_j + g_j)/\tau)} \tag{8}$$

where $g_1, ..., g_k$ are $i.i.d.$ sampled from the distribution Gumbel (0,1), and τ is the temperature parameter. k indicates the dimension of the generated Softmax vector.

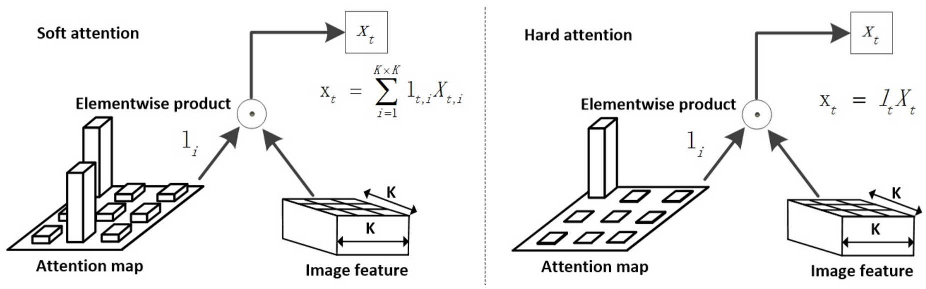

Figure 2: The attention mechanism: Soft attention assign weights on different locations of features using softmax whilst the values of the hard attention map are either 1 or 0 which means only one important location is selected.

To derive the Gumbel-sigmoid, we firstly re-write the Sigmoid function as a Softmax of two variables: π_i and 0.

$$\begin{aligned} sigm(\pi_i) &= \frac{1}{(1+exp(-\pi_i))} = \frac{1}{(1+exp(0-\pi_i))} \\ &= \frac{1}{1+exp(0)/exp(\pi_i)} = \frac{exp(\pi_i)}{(exp(\pi_i)+exp(0))} \end{aligned} \tag{9}$$

Hence, the Gumbel-sigmoid can be written as:

$$y_i = \frac{exp(log(\pi_i + g_i/\tau)}{exp(log(\pi_i + g_i)/\tau) + exp(log(g')/\tau)} \tag{10}$$

where g_i and g' are independently sampled from the distribution Gumbel (0,1).

To obtain a discrete value, we set values of $z_t = \widetilde{y_i}$ as:

$$\widetilde{y_i} = \begin{cases} 1 & y_i \geq 0.5 \\ 0 & otherwise \end{cases} \tag{11}$$

In our experiments, all the boundary detectors z_t are estimated using the Gumbel-sigmoid with a constant temperature of 0.3.

3.2.2. Deterministic Soft Attention

To implement soft attention over image regions for the action recognition task, we applied a similar strategy to the soft attention mechanism in [15] and

[14].

Specifically, the model predicts a Softmax over K×K image locations. The location Softmax is defined as:

$$l_{t,i} = \frac{exp(W_i h_{t-1})}{\sum_{j=1}^{K\times K} exp(W_j h_{t-1})} \qquad i = 1, ..., K^2 \tag{12}$$

where i means the ith location corresponding to the specific regions in the original image.

This Softmax can be considered as the probability with which the model learns the specific regions in the image, which is important for the task in hand. Once these probabilities are obtained, the model computes the expected values over image features at different regions:

$$x_t = \sum_{i=1}^{K^2} l_{t,i} X_{t,i} \tag{13}$$

where x_t is considered as inputs of the HM-AN networks. In our HM-AN implementations, the hidden states used to determine the region softmax is defined for the first layer, i.e., h^1_{t-1}. The upper layers will automatically learn the abstract information of input features as previously explained. The soft attention mechanism can be visualized in the left side of Fig. 2.

3.2.3. Stochastic Hard Attention

REINFORCE-like algorithm. Stochastic hard attention was proposed in [14]. Their hard attention was realized with the aid of a REINFORCE-like algorithm. In this section, we also introduce this kind of hard attention mechanism.

The location variable l_t indicates where the model decides to focus attention on the t^{th} frame of a video. $l_{t,i}$ is an indicator of a one-hot representation which can be set to 1 if the i^{th} location contains a relevant feature.

Specifically, we assign a hard attentive location of $\{\alpha_i\}$:

$$\begin{aligned} p(l_{i,t} = 1 | l_{j<t,a}) &= argmax(\alpha_{t,i}) \\ &= argmax\left(\frac{exp(W_i h_{t-1})}{\sum_{j=1}^{K\times K} exp(W_j h_{t-1})}\right) \end{aligned} \tag{14}$$

where a represents the input image features.

We can define an objective function L_l that is a variational lower bound on the marginal log-likelihood $log\ p(y|a)$ of observing the action label y given image features a. Hence, L_l can be represented as:

$$\begin{aligned} L_l &= \sum_l p(l|a) log\ p(y|l,a) \\ &\leq log\ \sum_l p(l|a) p(y|l,a) \\ &= log p(y|a) \end{aligned} \tag{15}$$

$$\begin{aligned} \frac{\partial L_l}{\partial W} = \sum_l p(l|a)[&\frac{\partial log\ p(y|l,a)}{\partial W} + \\ &log\ p(y|l,a)\frac{\partial log\ p(l|a)}{\partial W}] \end{aligned} \tag{16}$$

Ideally, we would like to compute the gradients of Equation 16. However, it is not feasible to compute the gradient of expectation in Equation 16. Hence, a Monte Carlo approximation technique is applied to estimate the gradient of the operation of expectation.

Therefore, the derivatives of the objective function with respect to the network parameters can be expressed as:

$$\begin{aligned} \frac{\partial L_l}{\partial W} = \frac{1}{N}\sum_{n=1}^{N}[&\frac{\partial log\ p(y|\tilde{l}_n,a)}{\partial W} + \\ &log\ p(y|\tilde{l}_n,a)\frac{\partial log\ p(\tilde{l}_n|a)}{\partial W}] \end{aligned} \tag{17}$$

where $\tilde{l}$ is obtained based on the argmax operation as in Equation 14.

Similar with the approaches in [14], a variance reduction technique is used. With the k^{th} mini-batch, the moving average baseline is estimated as an accumulation of the previous log-likelihoods with exponential decay:

$$b_k = 0.9 \times b_{k-1} + 0.1 \times log\ p(y|\tilde{l}_k,a) \tag{18}$$

The learning rule for this hard attention mechanism is defined as follows:

$$\begin{split}\frac{\partial L_l}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}[\frac{\partial log\ p(y|\tilde{l}_n, a)}{\partial W} + \\ \lambda(log\ p(y|\tilde{l}_n, a) - b)\frac{\partial log\ p(\tilde{l}_n|a)}{\partial W}]\end{split} \tag{19}$$

where λ is a pre-defined parameter.

As pointed out in Ba et al. [17], Mnih et al. [16] and Xu et al. [14], this is a formulation which is equivalent to the REINFORCE learning rule [19]. For convenience, it is abbreviated as REINFORCE-Hard Attention in the following sections.

Gumbel Softmax. In the hard attention mechanism, the model selects one important region instead of taking the expectation. Hence, it is a stochastic discrete unit which cannot be trained using back propagation. [14] applied REINFORCE to estimate the gradient of the stochastic neuron. Although REINFORCE is an unbiased estimator, the variance of the gradient is large and the algorithm is complex to implement. To solve these problems, we propose to apply Gumbel-softmax to estimate the gradient of the discrete units in our model. Gumbel-softmax is better than REINFORCE and much easier to implement [10].

We can simply replace the Softmax with Gumbel-softmax in Equation 12 and remove the process of taking expectation to realize the hard attention.

$$l_{t,i} = \frac{exp(log(W_i h_{t-1} + g_i)/\tau)}{\sum_{j=1}^{K \times K} exp(log(W_j h_{t-1} + g_j)/\tau)} \qquad i = 1...K^2 \tag{20}$$

The Gumbel-softmax will choose a single location indicating the most important image region for the task. However, the search space for the temperature parameter is too large to be manually selected. The temperature is a sensitive parameter as explained in [10]. Hence in this paper we applied an adaptive temperature as in [12]. The adaptive temperature determines the value depending on the current hidden states. In other words, instead of being treated

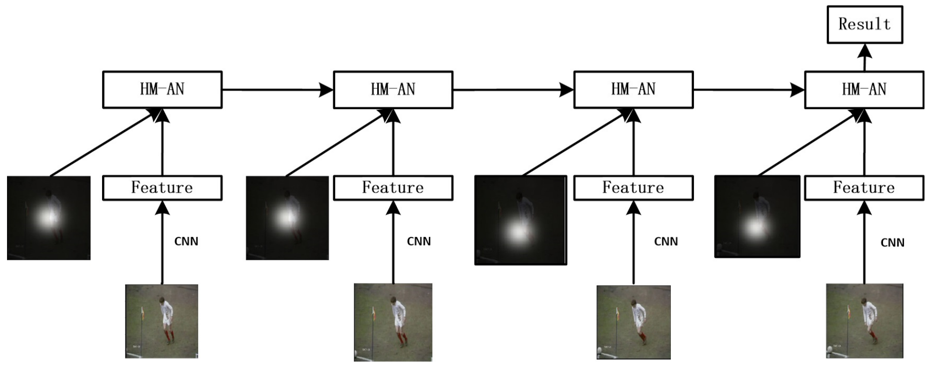

Figure 3: Action recognition with HM-AN.

as a pre-defined parameter, the value of temperature is learnt from the data. Specifically, we use the following mechanism to determine the temperature:

$$\tau = \frac{1}{Softplus(W_{temp} h_t^1 + b_{temp}) + 1} \tag{21}$$

where h_t^1 is the hidden state of the first layer of our HM-AN. Equation 21 generates a scalar for the temperature. In the equation, adding 1 can enable the temperature to fall into the scope of 0 and 1. The hard attention mechanism can be seen in the right hand side of Fig. 2.

3.3. Application of HM-AN in Action Recognition

The proposed HM-AN can be directly applied in video action recognition. In video action recognition, the dynamics exist in the inputs, i.e., the given video frames. With the attention mechanism embedded in RNN, the important features of each frames can be discovered and discriminated in order to facilitate recognition.

For action recognition, the HM-AN applies the cross-entropy loss for recognition.

$$LOSS = -\sum_{t=1}^{T}\sum_{i=1}^{C} y_{t,i} log(\hat{y}_{t,i}) \tag{22}$$

where y_t is the label vector, $\hat{y}_t$ is the classification probabilities at time step t. T is the number of time steps and C is the number of action categories. The system architecture of action recognition using HM-AN is shown in Fig. 3

4. Experiments

In this section, we first explain our implementation details then report the experimental results on action recognition.

4.1. Implementation Details

We implemented the HM-AN using the Theano platform [36] and all the experiments were conducted on a server embedded with a Titan X GPU. In our experiments, HM-AN is a three layer stacked RNN. The outputs are concatenated by hidden states from three layers and forwarded to a softmax layer.

In addition to the baseline approach (LSTM networks), four versions of HM-AN were implemented for the purpose of comparison:

- Softmax regression. This is to perform a general image classification task based on spatial features.
- LSTM with soft attention (Baseline). The baseline approach is set as a one layer LSTM networks with the soft attention mechanism.
- Deterministic soft attention in HM-AN (Soft Attention). This is to determine how soft attention mechanism performs with the HM-AN.
- Stochastic hard attention with reinforcement learning in HM-AN (REINFORCE-Hard Attention). This type of hard attention mechanism is described in Section 3.2.3.
- Stochastic hard attention with a 0.3 temperature for Gumbel-softmax in HM-AN (Constant-Gumbel-Hard Attention). A constant temperature is applied in Gumbel-softmax to accomplish the proposed hard attention model.
- Stochastic hard attention with adaptive temperature for Gumbel-softmax in HM-AN (Adaptive-Gumbel-Hard Attention). The temperature is set as a function of the hidden states of RNN.

For the experiments, with the help of the MatConvNet platform [37], we firstly extracted frame-level CNN features from the last convolutional layer (res5cx) based on Residue-152 Networks [4] trained on the ImageNet [38] dataset. The images were resized to 224×224, hence the dimension of each frame-level features is 7×7×2048. For the network training, we applied a mini-batch size of 64 samples at each iteration. For each video sequence, the baseline approach randomly selected a sequence of 30 frames for training while the proposed approaches selected a sequence of 60 frames for training in order to verify the proposed HM-AN's capability to capture long-term dependencies. Actually, the optimal length for LSTM with attention is 30 and increasing the number will seriously deteriorate the performance. In order to determine the optimal length of sequence feeding into the networks, we perform several trials as described in Section 4.2.2, determining that the optimal length for the HM-AN is 60. We applied the back propagation algorithm through time and Adam optimizer [39] with a learning rate of 0.0001 to train the networks. The learning rate was changed to 0.00001 after 10,000 iterations. At test time, we compute class predictions for each time step and then average those predictions over 60 frames. Table 1 provides a detailed description of the network configuration. Table 2 shows the number of iterations and epoches needed for convergence on different datasets.

4.2. Experimental Results and Analysis

4.2.1. Datasets

We evaluated our approach on three widely used datasets, namely UCF Sports [40], the Olympic Sports datasets [41] and the more difficult Human Motion Database (HMDB51) dataset [42]. Fig. 4 provides some examples of the three datasets used in this paper. The UCF Sports dataset contains a set of actions collected from various sports which are typically featured on broadcast channels such as ESPN or BBC. This dataset consists of 150 videos with a resolution of 720 × 480 and contains 10 different action categories. The Olympic Sports dataset was collected from YouTube sequences [41] and contains 16 dif-

Table 1: Networks Structure Configuration.

Input to HM-AN		Size of Inner Units of HM-AN	
Inputs	$7 \times 7 \times 2048$	Hidden Unit Size	2048
Output Layers		Cell Memory Size	2048
1st Layer Outputs	2048	Gate Size (i, f, o, g)	2048
2nd Layer Outputs	2048	Boundary Detector Size	2048
3rd Layer Outputs	2048	Training Parameters	
Concatenation Layer	6144	Dropout	0.5
Fully connected Layer 1	1024	Learning Rate	0.00001
Fully connected Layer 2	Class Categories	Video Sequence Length	60

Table 2: Number of Iterations and Epoches for Convergence on Different Datasets.

Dataset	Iterations	Epoches
UCF Sports	400	2
Olympic Sports	2500	2
HMDB51	10000	2

ferent sports categories with 50 videos per class. Hence, there are a total of 800 videos in this dataset. The HMDB51 dataset is a more difficult dataset which provides three train-test splits each consisting of 5100 videos. These sequences are labeled with 51 action categories. The training set for each split has 3570 videos and the test set has 1530 videos.

For the UCF Sports dataset, as there is lack of training-testing split for evaluation, we manually divide the dataset into training and testing sets. We randomly selected 75 percent for training, and left the remaining 25 percent for testing. We then report the classification accuracy on the testing dataset.

As for Olympic Sports dataset, we used the original training-testing split with the 649 sequences for training and 134 sequences for testing provided in

(a) UCF Sports dataset

(b) Olympic Sports dataset

(c) HMDB51 dataset

Figure 4: Some examples from the datasets used in this paper.

the dataset. Following the practice in [41], we evaluated the Average Precision (AP) for each category on this dataset.

When evaluating our method on HMDB51, we also followed the original training-testing split and report the classification accuracy on the testing set.

4.2.2. Results

UCF Sports dataset. We firstly tested the performance of the LSTM with soft attention proposed in [15] on the UCF Sports dataset and obtained 70.0% accuracy. All the experimental settings were the same as those in [15]. Then we evaluated the proposed four approaches mentioned previously. As described in [15], the optimal sequence length is 30 frames.

One of the expectations of using HM-AN is to enable long-term dependencies. In order to find the optimal length for HM-AN, we performed certain experiments. As shown in Table 3, the optimal length of the video sequence is

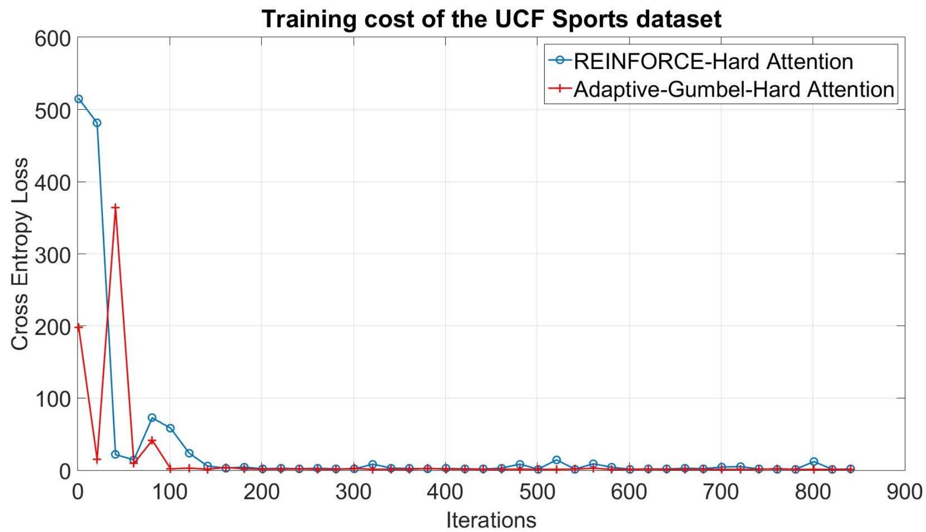

Figure 5: Training cost of the UCF Sports dataset.

60 frames. Increasing or decreasing the length would cause a drop in the overall result accuracy.

HM-AN with stochastic hard attention which is realized with REINFORCE-like algorithm improves the results to 82.0%. HM-AN with soft attention is similar to the REINFORCE-Hard Attention, with an accuracy of 81.1%. The hard attention mechanism realized by Gumbel-softmax with adaptive temperature achieves 82.0% accuracy, similar to our REINFORCE-Hard Attention model. However, the Constant-Gumbel-Hard Attention which uses Gumbel-softmax with constant temperature value of 0.3 only yields 76.0% accuracy, which indicates the significant role of adaptive temperature in maintaining the system performance. Fig. 5 shows the curves of training cost cross entropy for the Adaptive-Gumbel-Hard Attention approach and REINFORCE-Hard Attention approach, respectively. It can be seen from the figure that the REINFORCE-Hard Attention converges marginally slower than the approach of Adaptive-

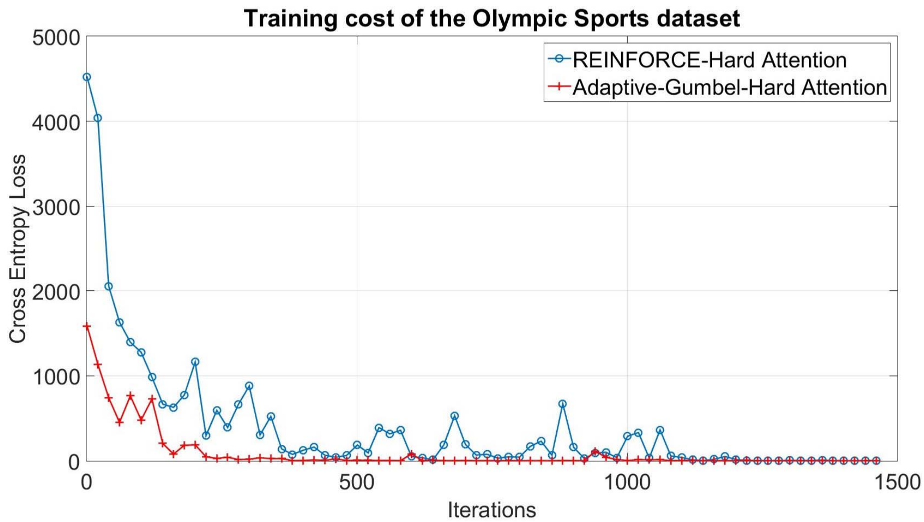

Figure 6: Training cost of the Olympic Sports dataset.

Gumbel-Hard Attention.

As shown in Table 4, we compare our model with the methods proposed in [43] in which a convolutional LSTM attention network with hierarchical architecture was used for action recognition. The hierarchical architecture in [43] was pre-defined whilst our model is able to learn the hierarchy from the data. The improvements demonstrated by our methods are obvious as shown in Table 4.

Olympic Sports dataset. The Olympic Sports dataset is of medium size. Results from this dataset are shown in Table 5. The mAP result of baseline approach is 73.7%. Our method HM-AN with Soft attention achieves 82.4% mAP. However, unlike the UCF Sports dataset, the mAP result of REINFORCE-Hard Attention is 77.1%, which is lower than the approach of Soft Attention. The Constant-Gumbel-Hard Attention, which is implemented by Gumbel-softmax with a constant temperature of 0.3, obtains a mAP value of 82.3%. By mak-

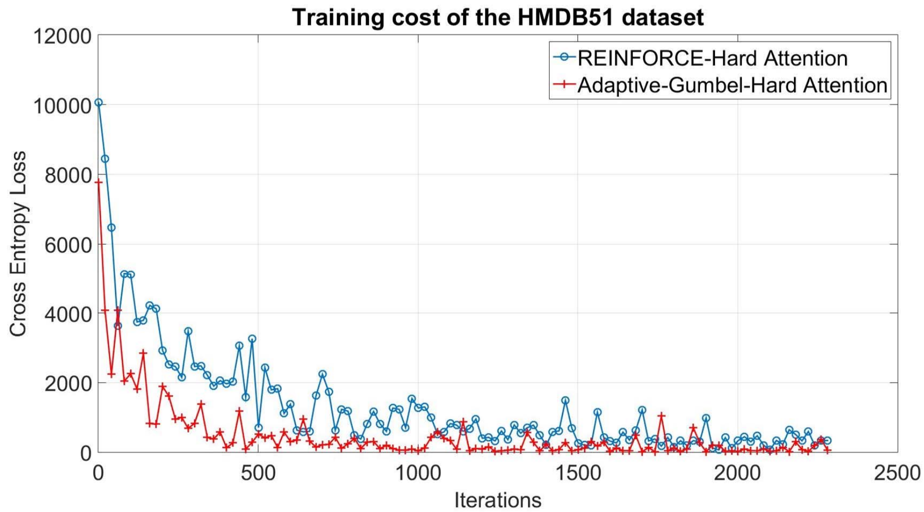

Figure 7: Training cost of the HMDB51 dataset.

Table 3: Accuracy on UCF Sports using Adaptive-Gumbel-Hard Attention with different sequence lengths.

Sequence Length	Accuracy
30 frames	70.0%
40 frames	74.0%
50 frames	78.0%
60 frames	**82.0%**
70 frames	80.1%

ing the temperature value of Gumbel-softmax adaptive, the proposed model achieves 82.7% mAP, the highest among all our experimental results. Again, our proposed methods show superior performance compared to the hand-designed hierarchical model in [43].

Table 4: Accuracy on UCF Sports

Methods	Accuracy
Softmax Regression (Residue-152 Features)	66.0%
Baseline (Residue-152 Features)	70.0%
Conv-Attention [43] (Residue-152 Features)	72.0%
CHAM [43] (Residue-152 Features)	74.0%
Soft Attention (Residue-152 Features)(Ours)	81.1%
REINFORCE-Hard Attention (Residue-152 Features)(Ours)	**82.0%**
Constant-Gumbel-Hard Attention(Residue-152 Features) (Ours)	76.0%
Adaptive-Gumbel-Hard Attention (Residue-152 Features)(Ours)	**82.0%**

HMDB51 dataset. HMDB51 is a more difficult and larger dataset. First of all, we test the accuracy of softmax regression based on Residue-152 networks, with 38.2% accuracy, which improved this approach based on GoogleNet features by 4.7%. This is consistent with previous findings where the Residue-152 networks reported 23.0% top 1 error on ImageNet dataset [38], which is 11.2% percent less than the GoogleNet results (34.2%) [44] [4]. However, all the subsequent experiments are all performed using features from Residue-152 features, which verify that the performance gain is from the proposed model instead of the advanced image features. The performance of the baseline approach is shown in Table 7, with 40.8% accuracy. The three layer LSTMs with soft attention based on GoogleNet features was reported in [15], with 41.3% accuracy. To make the comparison fair, we also tested three layer LSTMs with soft attention on Residue-152 features. However, we were not able to obtain a very obvious improvement on the final result, with 42.4% accuracy (1.1% gains over the result from [15]). Our HM-AN model with soft attention improves the accuracy to 43.8%. We then applied the REINFORCE-Hard Attention approach on this dataset. The result accuracy turns out to be lower than the HM-AN with soft attention. Moreover, the model with REINFORCE-like algorithm converges slower than the Gumbel-softmax with adaptive temperature, also with more

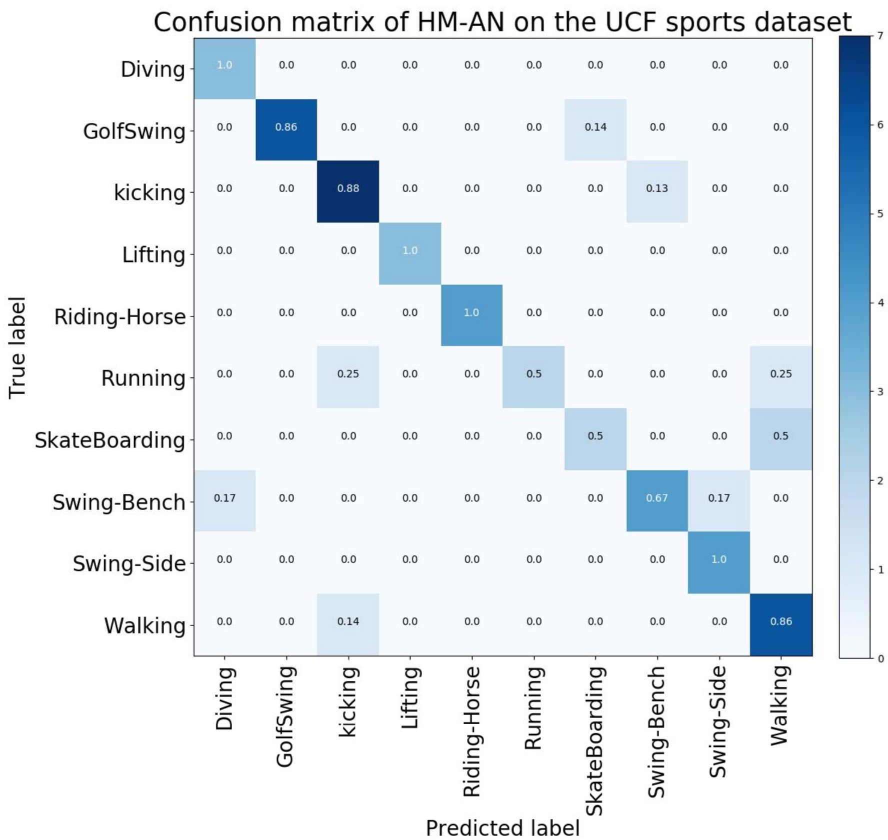

Figure 8: Confusion Matrix of HM-AN with Adaptive-Gumbel-Hard Attention on the UCF Sports dataset.

oscillations on the training cost, which is shown in Fig. 7. With a constant temperature value of 0.3 for hard attention, the model achieves 44.0% accuracy. Again, the improvement by adding adaptive temperature is obvious, with 44.2% accuracy on the HMDB51 dataset. The accuracy results are further summarized in Table 7.

We also compare the performance of the proposed HM-AN with some published models related to ours. Our proposed approach shares similarity with the spatial convolutional net from the two-stream scheme [26]. The difference is that the two-stream approach performs fine-tuning on the CNN model, with

Table 5: AP on Olympics Sports

Class	Vault	Triple Jump	Tennis serve	Spring board	Snatch
Softmax Regression (Residue-152 Features)	97.7%	100.0%	42.8%	58.4%	31.7%
Baseline (Residue-152 Features)	97.0%	88.4%	52.3%	60.0%	23.2%
Conv-Attention (Residue-152 Features) [43]	97.0%	94.0%	49.8%	66.4%	26.1%
CHAM (Residue-152 Features) [43]	97.0%	98.9%	49.5%	69.2%	47.8%
Soft Attention (Residue-152 Features)(Ours)	99.0%	100.0%	60.7%	64.2%	38.6%
REINFORCE-Hard Attention (Residue-152 Features) (Ours)	100.0%	95.0%	50.8%	56.3%	28.6%
Constant-Gumbel-Hard Attention (Residue-152 Features) (Ours)	97.0 %	99.0%	62.6 %	58.7%	40.3%
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	98.1%	98.9%	62.1%	64.3%	45.4%
Shot put	Pole vault	Platform 10m	Long jump	Javelin Throw	High jump
61.5%	88.8%	85.6%	96.6%	95.0%	79.7%
67.4%	69.8%	84.1%	100.0%	89.6%	84.4%
60.0%	100%	86.0%	98.0%	87.9%	80.0%
79.8%	60.8%	89.7%	100.0%	95.0%	78.7%
77.2%	85.4%	91.5%	98.9%	97.0	77.2%
90.6%	100.0%	86.7%	100.0%	89.7%	77.5%
87.8%	100.0%	93.1%	100.0%	93.2%	82.8%
84.1%	100.0%	94.8%	100.0%	95.3%	86.2%
Hammer throw	Discus throw	Clean and jerk	Bowling	Basketball layup	mAP
32.9%	84.2%	78.0%	41.5%	89.3%	72.7%
38.0%	100.0%	76.0%	60.0%	89.8%	73.7%
36.6%	97.8%	100.0%	46.8%	81.2%	75.5%
37.9%	97.0%	84.8%	46.7%	89.1%	76.4%
44.1%	94.2%	83.8%	63.9%	89.2%	77.1%
52.9%	95.8%	92.4%	69.4%	98.1%	82.4%
54.7%	95.8%	91.3%	60.5%	100.0%	82.3%
53.8%	95.8%	84.9%	62.5%	97.0%	**82.7%**

an improved accuracy of 40.5%. Recent research on the two-stream approach [27] reported better results, with 47.1% accuracy. However, the evaluation of the two-stream method is based on each video whilst our evaluation is based on 60 frame sequences. The sequence-based accuracy is normally lower than the video-based accuracy as described in [45]. We only list the video-based approaches for reference since the evaluation of them is different from sequence-based approaches.

For sequence-based approaches, the methods not from the RNN family but only with the spatial image, show poor performance as illustrated in Table 8. Specifically, the softmax regression approach [15] directly uses extracted image features of each frame and performs softmax regression on them, with 33.5% accuracy. The softmax regression approach based on image features from Residue-152 networks improves the accuracy to 38.2%. [15] reported that the LSTM without attention achieves 40.5% accuracy [15]. When adding the

Table 6: Accuracy of Softmax Regression on HMDB51 based on Different Features

Image Features	Accuracy
GoogleNet	33.5%
Residue-152 Network	38.2%

soft attention mechanism, an improved accuracy of 41.3% can be obtained. The Conv-Attention [43] and ConvALSTM [21] both use convolutional LSTM with attention. The differences are that Conv-Attention extracts features from Residue-152 Networks [4] without fine-tuning whilst ConvALSTM extracts image features from a fine-tuned VGG16 model. The ConvALSTM leads Conv-Attention by a small margin, with 43.3% accuracy. As explained previously, CHAM [43] has a hand-designed hierarchical architecture, which is in contrast with ours in which the temporal hierarchy is formed through training. Our best setting (Adaptive-Gumbel-Hard Attention) reports the highest accuracy (44.2%) among methods from the RNN family and leads the CHAM results (43.4%) by 0.8 percent. In sequence-based approaches, the one that outperforms ours is the Long-term temporal convolutions [45], with 52.6% accuracy. This method has a 3D-convolution architecture, and is trained directly on the specific dataset, which is very different from our approach.

Analysis and Visualization. We tested four approaches (Soft Attention, REINFORCE-Hard Attention, Constant-Gumbel-Hard Attention and Adaptive-Gumbel-Hard Attention) on three different datasets: UCF Sports dataset, the Olympic Sports dataset and the HMDB51 dataset. On the UCF Sports dataset, the REINFORCE-Hard Attention and Adaptive-Gumbel-Hard Attention generate satisfactory results and show better performance than the soft attention and Constant-Gumbel-Hard Attention. This indicates that the adaptive temperature is an efficient method to improve performance in the implementation of Gumbel-softmax based hard attention.

Table 7: Accuracy on HMDB51

Methods	Accuracy
Softmax Regression (Residue-152 Features)	38.2%
Baseline (Residue-152 Features)	40.8%
Three LSTM Layers with Attention (Residue-152 Features)	42.4%
Soft Attention (Residue-152 Features)(Ours)	43.8%
REINFORCE-Hard Attention (Residue-152 Features)(Ours)	41.5%
Constant-Gumbel-Hard Attention (Residue-152 Features)(Ours)	44.0%
Adaptive-Gumbel-Hard Attention (Residue-152 Features)(Ours)	**44.2%**

Table 8: Comparison with related methods on HMDB51

Methods	Accuracy	Spatial Image Only	Fine-tuning of CNN model
Video Accuracy			
Spatial Convolutional Net (8 Layers CNN model) [26]	40.5%	Yes	Yes
Spatial Convolutional Net (VGG 16) [27]	47.1%	Yes	Yes
Composite LSTM Model [46]	44.0%	Yes	No
Trajectory-based modeling [47]	40.7%	No	No
Deep 3D CNN [48]	**51.9%**	Yes	Yes
Sequence Accuracy			
ConvALSTM (VGG16 model) [21]	43.3%	Yes	Yes
Long-term temporal convolutions [45]	**52.6%**	Yes	Yes
Softmax Regression (GoogleNet Features) [15]	33.5%	Yes	No
Average pooled LSTM [15] (GoogleNet Features)	40.5%	Yes	No
Three LSTM Layers with Attention (GoogleNet Features) [15]	41.3%	Yes	No
Three LSTM Layers with Attention (Residue-152 Features)	42.4%	Yes	No
Conv-Attention (Residue-152 Features) [43]	42.2%	Yes	No
CHAM (Residue-152 Features) [43]	43.4%	Yes	No
Adaptive-Gumbel-Hard Attention (Residue-152 Features) (Ours)	**44.2%**	Yes	No

On both of the Olympic Sports dataset and HMDB51 dataset, the best approach is the Adaptive-Gumbel-Hard Attention while the REINFORCE-Hard Attention is even worse than the soft attention mechanism. On the bigger datasets, the advantages of Gumbel-softmax include small gradient variance and simplicity, which are obvious compared with the REINFORCE-like algorithms. This shows that Gumbel-softmax generalizes well on large and complex datasets. This is reflected not only by the result accuracy, but also by the training cost

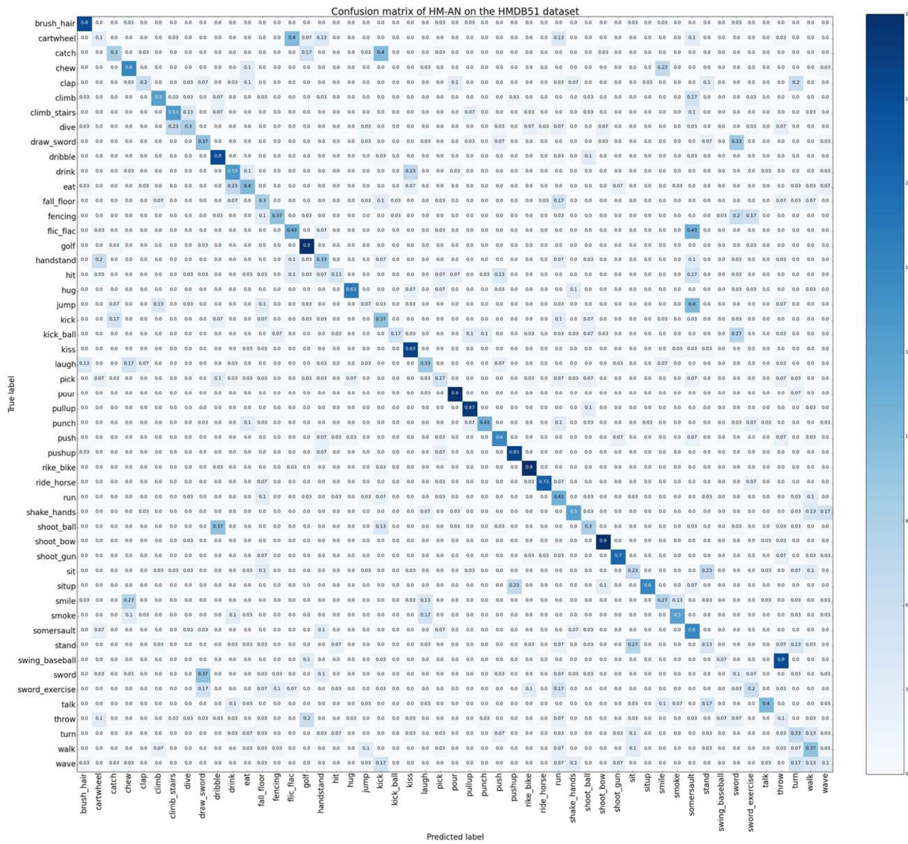

Figure 9: Confusion Matrix of HM-AN Adaptive-Gumbel-Hard Attention on the HMDB51 dataset.

curves in Fig. 6 and Fig. 7. This conclusion is also consistent with the findings in other recent research [12] which also applied both REINFORCE-like algorithms and Gumbel-softmax as estimators for stochastic neurons.

The visualization of attention maps and boundary detectors learnt by the HM-AN is shown in Fig. 10. In the attention maps, the brighter an area is, the more important it is for the recognition. The soft attention captures multi-regions while the hard attention selects only one important region. As can be seen from the figure, in different time steps, the attention regions are different which means the model is able to select region to facilitate the recognition through time automatically. The z_1, z_2 and z_3 in the figure indicate the boundary detectors in the first layer, the second layer and the third layer, respectively. In the figure, for the boundary detectors, the black regions indicate there exists a boundary in the time-domain whilst the grey regions show the UPDATE operation can be performed. The multi-scale properties in the time-domain can be captured by the HM-AN as different layers show different boundaries.

From the reported results, we find that on all three datasets, the Constant-Gumbel-Hard Attention approach is worse than the approach of Adaptive-Gumbel-Hard Attention. This is because we do not know initially which temperature parameter is the optimal for the dataset. To provide a better understanding of the network, we showed how the adaptive temperature change along with the test samples on three datasets, as shown in Fig. 11. From the figure, we can see that the adaptive temperature is about 0.6, which is very different from the pre-defined 0.3 temperature in Constant-Gumbel-Hard Attention.

On the UCF Sports dataset, the Constant-Gumbel-Hard Attention is significantly worse than other approaches, including the REINFORCE-Hard Attention, with only 76.0% accuracy. As shown in Fig. 11, the temperature from the UCF Sports dataset is slightly higher than the other two datasets, which means the 0.3 pre-defined temperature parameter is not an appropriate option. In addition, the approach of Adaptive-Gumbel-Hard Attention makes the networks converge much quicker as shown in Fig. 5, Fig. 6 and Fig. 7, which also explains the higher accuracy results of this method.

5. Conclusion

In this paper, we proposed a novel RNN model, HM-AN, which improves HM-RNN with attention mechanism for visual tasks. Specifically, the boundary detectors in HM-AN are implemented by the recently proposed Gumbel-sigmoid. Two versions of the attention mechanism were implemented and tested. Our work is the first attempt to implement hard attention in vision tasks

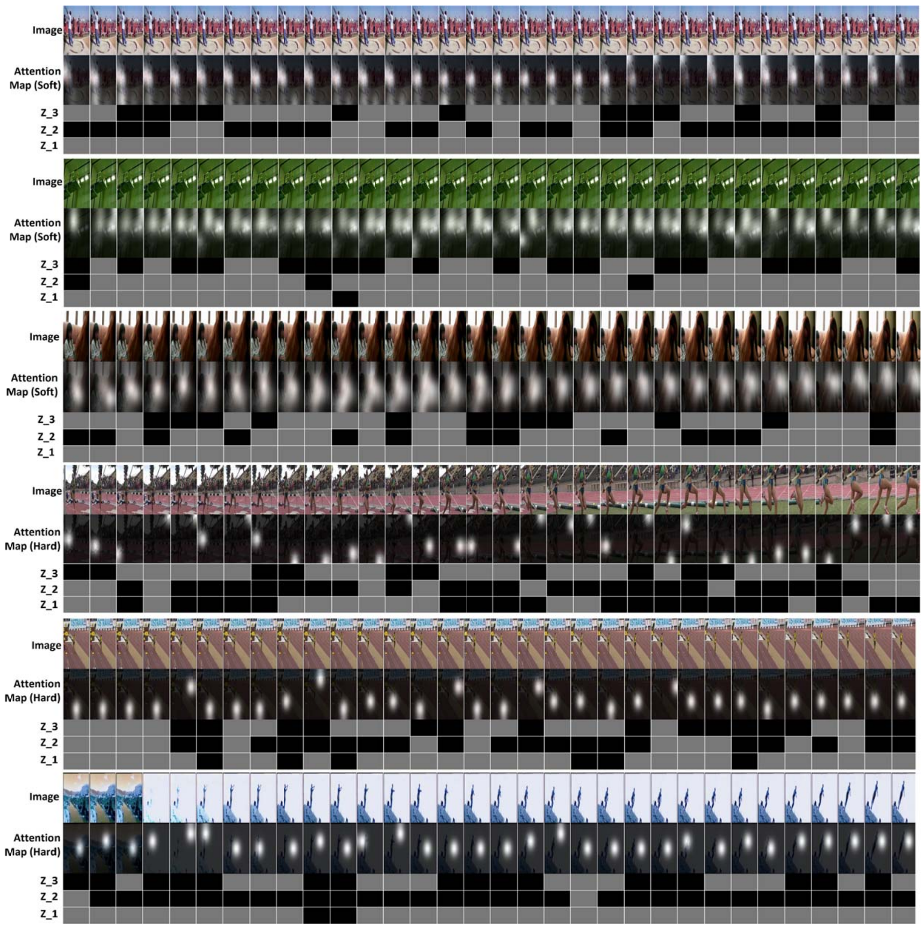

Figure 10: Visualization of attention maps and detected boundaries for action recognition.

with the aid of Gumbel-softmax instead of REINFORCE algorithm. To solve the problem of sensitive parameter of softmax temperature, we applied adaptive temperature methods to improve the system performance. To validate the effectiveness of HM-AN, we conducted experiments on action recognition from videos. Through experimenting, we showed that HM-AN is more effective than LSTMs with attention. The attention regions of both hard and soft attention and boundaries detected in the networks provide visualization for the insights of what the networks have learnt. Theoretically, our model can be built based on

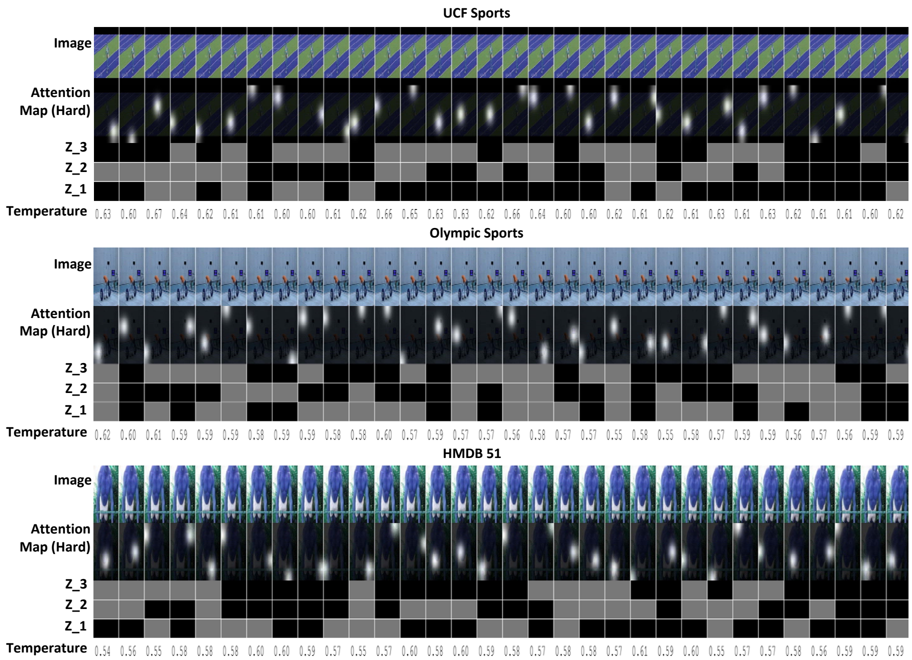

Figure 11: Visualization of temperature values with attention maps and detected boundaries for action recognition, the samples are randomly selected.

various features, e.g., Dense Trajectories, to further improve the performance. However, our emphasis in this paper is to prove the superiority of the model itself compared with other RNN-like models given same features. Hence, we chose to use deep spatial features only. Our work can facilitate further research on the hierarchical RNNs and its applications to computer vision tasks.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[5] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.

[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[7] J. Chung, S. Ahn, Y. Bengio, Hierarchical multiscale recurrent neural networks, arXiv preprint arXiv:1609.01704.

[8] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, B. Li, Hierarchical attention network for action recognition in videos, arXiv preprint arXiv:1607.06416.

[9] Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint arXiv:1308.3432.

[10] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144.

[11] C. J. Maddison, A. Mnih, Y. W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, CoRR abs/1611.00712. URL http://arxiv.org/abs/1611.00712

[12] C. Gulcehre, S. Chandar, Y. Bengio, Memory augmented neural networks with wormhole connections, arXiv preprint arXiv:1701.08718.

[13] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, ICLR 2015.

[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International Conference on Machine Learning, 2015, pp. 2048–2057.

[15] S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention, in: International Conference on Learning Representations (ICLR) Workshop, 2016.

[16] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in neural information processing systems, 2014, pp. 2204–2212.

[17] J. Ba, V. Mnih, K. Kavukcuoglu, Multiple object recognition with visual attention, in: International Conference on Learning Representations (ICLR), 2015.

[18] J. Koutnik, K. Greff, F. Gomez, J. Schmidhuber, A clockwork rnn, in: 31st International Conference on Machine Learning (ICML), 2014.

[19] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (3-4) (1992) 229–256.

[20] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Video description generation incorporating spatio-temporal features and a soft-attention mechanism, arXiv preprint arXiv:1502.08029.

[21] Z. Li, E. Gavves, M. Jain, C. G. Snoek, Videolstm convolves, attends and flows for action recognition, Computer Vision and Image Understanding 2018.

[22] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation

nowcasting, in: Advances in Neural Information Processing Systems, 2015, pp. 802–810.

[23] E. Teh, M. Rochan, Y. Wang, Attention networks for weakly supervised object localization, BMVC, 2016.

[24] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 3551–3558.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[26] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[27] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[28] Y.-G. Jiang, Q. Dai, W. Liu, X. Xue, C.-W. Ngo, Human action recognition in unconstrained videos by explicit motion modeling, IEEE Transactions on Image Processing 24 (11) (2015) 3781–3795.

[29] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, S.-F. Chang, Exploiting feature and class relationships in video categorization with regularized deep neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[30] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional lstm, in: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, IEEE, 2013, pp. 273–278.

[31] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625–2634.

[32] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4694–4702.

[33] K. Fu, J. Jin, R. Cui, F. Sha, C. Zhang, Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2017) 1–1. `doi:10.1109/TPAMI.2016.2642953`.

[34] E. J. Gumbel, J. Lieblein, Statistical theory of extreme values and some practical applications: a series of lectures.

[35] C. J. Maddison, D. Tarlow, T. Minka, A* sampling, in: Advances in Neural Information Processing Systems, 2014, pp. 3086–3094.

[36] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, arXiv preprint arXiv:1211.5590.

[37] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 689–692.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.

[39] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), 2015.

[40] M. Rodriguez, Spatio-temporal maximum average correlation height templates in action recognition and video summarization.

[41] J. C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: European conference on computer vision, Springer, 2010, pp. 392–405.

[42] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: Proceedings of the International Conference on Computer Vision (ICCV), 2011.

[43] S. Yan, J. S. Smith, W. Lu, B. Zhang, Cham: action recognition using convolutional hierarchical attention model, in: Proceedings of the IEEE Conference on Image Processing, 2017.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 00 (2015) 1–9. doi:doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298594.

[45] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[46] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised learning of video representations using LSTMs, in: ICML, 2015.

[47] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points, in: European Conference on Computer Vision, Springer, 2012, pp. 425–438.

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489–4497. doi:10.1109/ICCV.2015.510.