CNN and RNN Based Neural Networks for Action Recognition

To cite this article: Chen Zhao et al 2018 J. Phys.: Conf. Ser. 1087 062013

View the article online for updates and enhancements.
CNN and RNN Based Neural Networks for Action Recognition

Chen Zhao, JunGang Han, Xuebin Xu
Xi’an University Of Posts and Telecommunications, Xi’an, Shanxi 710121, China
awp4211@gmail.com

Abstract. In this paper, we propose a novel approach to recognize human actions. Action recognition from videos has not been addressed extensively and effectively, primarily due to the tremendous variations that result from background and scale, etc. In order to classify the videos, at first we use very deep Convolution Neural Networks (CNNs) to extract the features of the frame in videos. Then, we use multilayered Recurrent Neural Networks (RNNs) with a type of Long Short-Term Memory (LSTM) units to process the sequence of the extracted features by CNNs. Finally, we integrate both of the CNNs and RNNs for our model. We evaluate the model on UCF-11 (YouTube Action) dataset and analyze which type of the model is fit for classify the videos. And the model we trained gets a higher accuracy.

1. Introduction

It has been proved that humans do not focus their attention on an entire scene at once when make cognition in a scene. Instead, human focus on a sequence of the scene to extract relevant information to recognize it [1]. But most traditional computer vision algorithms only pay attention to just one picture or a part of the video. In fact, people perceive an image by a large number of features extracted from the image, but unfortunately, it is hard to know which feature is more important than the others, and this make it difficult to classify videos by computer program.

Classifying videos instead of images adds a temporal dimension to the problem. With the recent surge of interest in deep neural networks, it is possible to solve the classification problem with temporal dimension.

CNN can directly identify the visual pattern from the original image and it needs very little pretreatment work [2]. CNN can simultaneously extract and train a variety of features of computer vision and with the extracted features, CNN can imitate the process of people’s recognition of the pictures. By now, CNN has been successfully applied to handwritten character recognition [3], face recognition [4] and etc. In recent year, Google’s GoogLeNet [5] makes the quality of image recognition and object detection proved at a dramatic pace.

In this paper we will describe how our model handle the above problems to recognize actions. We describe how our model dynamically pools convolutional features and how to using these sequences to abstract the features for understanding the videos.

2. Related Work

In the field of action recognition, the main focus is on realistic datasets collected from websites, movies and TV shows [6] [7]. Action recognition has been a challenging problem for many years. Many spatio-temporal algorithms have been proposed such as HOG3D [8] and 3D-SIFT [9]. Meanwhile, Jingen Liu used diffusion distance [10] to learn semantic visual vocabularies and present a systematic framework for recognizing realistic actions from videos [11].
It is important to extract features of image in video frames. In recent year, CNNs have been highly successful in image classification and multiscale image feature extraction [12]. For larger datasets such as ImageNet, the recent trend has been to increase the number of layers, layer size and filter size while using dropout [13] to handle the problem of overfitting. When the depth of networks is increased, the gradient often vanish or explode [14]. The problem makes us can’t train model normally and hamper convergence from the beginning. To handle this problem, residual connection was introduced by He et al.in [15] in which they give convincing method and practical evidence for the advantages of the using additive merging of signals for image recognition. Google integrates the Inception module with residual module to generate the Inception Residual block [16].

Hidden Markov Models (HMMs) is typically used for sequence processing. Recently recurrent neural networks (RNNs) hold more promise for recognizing patterns that are defined by temporal distance [17]. Hochreiter and Schmidhuber proposed a novel type of RNN unit called Long Short-Term Memory (LSTM) [18] unit that works better than traditional RNNs on tasks involving long time lags.

In this paper, we combine CNNs and RNNs to recognize human actions. And we got remarkably improved precision on recognition of human actions.

3. The Model and Architecture

We have tested various inception nets, residual nets and plain nets, and have observed consistent phenomena. The total deep neural networks for action recognition is shown in Figure 1 which contains the following steps:

Step 1: use different deep convolution neural networks to extract various features of the images;
Step 2: reshape the feature matrix to sequence;
Step 3: set the feature matrix as the input of multilayered recurrent neural networks;
Step 4: use softmax regression classifier to classify the videos.

3.1. Convolutional neural networks

We extract the last convolutional layer obtained by inputting the video frames through kinds of GoogLeNets. The last convolution layer has K convolution maps and the features form a feature cube with the shape of W*H*K. After convolutional neural network, we use a down sampling method such as average pooling with kernel size W*H by padding method of ‘VALID’ to make sure the feature cube to 1-dimensional vector(1*1*K). The 3-dimensional matrix can easily reshape to 1-dimensional vector, and the vector is the feature of the picture. We define the feature vector as:

\[ X_t = \{X_t^1, X_t^2, \ldots, X_t^K\}, X_t^i \in \mathbb{R} \]

As for the detail of convolutional neural networks, we use a variety kind of residual and inception blocks shown in Figure 2, Figure 3 and Figure 4. All the convolutions not marked with ‘V’ in the figures are same-padded meaning that their output size is same as the original input. And if the kernel size not equals to 1*1, we use the ReLU activation function to the layer. The 1*1 convolution without activation is used for controlling the filter’s size to fit the depth of input and output.
First International Conference on Advanced Algorithms and Control Engineering

IOP Conf. Series: Journal of Physics: Conf. Series 1087 (2018) 062013
doi:10.1088/1742-6596/1087/6/062013

Figure 2  The Inception_Residual_A block for 35*35 grid which is used in lower convolution part

Figure 3  The Inception_Residual_B block for 17*17 grid which is used in middle convolution part

Figure 4  The Inception_Residual_C block for 8*8 grid which used in higher convolution part

Before we push image to the Inception-Residual block, we reduce the image size by Reduction block in Figure 5 (The digit after ‘/’ represents the convolution or max pooling’s stride. If the stride is 2 then the image size will scale down.).
Figure 5  The Reduction block connects the original inputs and inception blocks

The fully convolutional neural network structure in our model is shown in Figure 6. Between the Inception_Residual blocks, we insert the ReductionA block and ReductionB block which same as InceptionV4 [16]. And the fully convolutional neural network structure in our model is shown in Figure 6.

Figure 6  Overall CNN in our model

3.2. A different type of LSTM cell
The LSTM has complicated dynamics that allow it to easily “memorize” information for an extended number of time steps. The “long term” memory is stored in a vector of memory cell $c_t \in \mathbb{R}^n$. The LSTM cell’s detail structure is shown in Figure 7. Let $h_t \in \mathbb{R}^n$ be a hidden state in layer 1 in time step t. A LSTM cell contains an input gate, a forget gate and a output gate. LSTM’s architecture used in our experiments is an improvement model of Graves et al. [19]:

$$
\text{LSTM: } h_t^{1:t-1}, h_{t-1}, c_{t-1} \rightarrow h_t^t, c_t^t \quad (1)
$$

$$
i_t^t = \text{input} \_ \text{gate} = \text{input} \ast W_i \quad (2)
$$

$$
f_t^t = \text{forget} \_ \text{gate} = \sigma(f_{t-1}^t + \text{forget} \_ \text{bias}) \quad (3)
$$

$$
\sigma_t^t = \text{output} \_ \text{gate} = \text{output} \ast W_o \quad (4)
$$

$$
c_t^t = f_t^t \odot c_{t-1}^t + i_t^t \odot \sigma(input \ast W_i) \quad (5)
$$

$$
h_t^t = o_t^t \odot \sigma(c_t^t) \quad (6)
$$

where $\sigma$ is the symbol of sigmoid activation function.
Figure 7 shows the LSTM cell used in this paper. Each gate’s activation function is sigmoid function. By a lot of practical evidence, in fact, we can’t get high precision with single layer LSTM cells based RNN. So we choose to use multilayered RNN which is shown in Figure 8, and we denote the state of last LSTM cell in last layer as the final state, which is used to classify actions.

3.3. Fully architecture of our model
The overall architecture is shown in Figure 9. The number of frames in video is defined as ‘fps’, and we feed our model with times of ‘fps’ images once.
4. Experiment

4.1. Dataset and pretreatment

We have used UCF-11 dataset in our experiments. UCF-11 contains 11 action categories: basketball shooting, biking and etc. For each video, we sample 24 frames (‘fps’) randomly. Then we use gray images and pretreatment each image with histogram equalization algorithm. There is a sequence of one video after pretreatment in Figure 10.

![Figure 10 Video frames for 24(fps) time-steps for an example of playing basketball after pretreatment](image)

‘fps*299*299*1’ means that one video in dataset has fps(=24) frames, we resize the pictures to 299 pixels in height and 299 pixels in width and we only use gray scale pictures.

All videos in the dataset are resized to 224*224 resolution before to be fed to InceptionV1 [5] and InceptionV2 [20] model. And all videos are resized to 299*299 before to be fed to InceptionV3 [20], InceptionV4 our model.

4.2. Training details and precision

We implemented the model in Tensorflow framework and we choose AdamOptimizer [21] and RMSprop algorithm to train our model. Each model convergences within 100 epochs. The results were obtained on a machine with two Intel Xeon E5-2620v3 CPUs and a NVIDIA Tesla K80c GPU. Training each model 100 epochs cost about 4000 minutes. Our code based on Tensorflow is available at: github@awp4211/TensorFlowLearning/UFC11.

The recognition precision on different model is described in Table 1 where ‘n_lstm’ represents the number of LSTM layers we use in multilayered RNN. We also trained the networks with 1 to 4 LSTM layers, but the precision is very low (below 10%).

By contrast, testing in UCF-11 dataset, Jinggen Liu only get the best result of 76.1% in [10] with the method of Diffusion Distance and 71.2% in [11]. Shikhar get the best result of 84.96% by using Soft Attention model [22]. So, by using deep neural network, the percision can easily improved. Traditional computer visual algorithm is a type of shallow neural network and it works no better than deep neural network.

| n_lstm | CNN | InceptionV1 | InceptionV2 | InceptionV3 | InceptionV4 | Our Model |
|--------|-----|-------------|-------------|-------------|-------------|-----------|
| 5      | 5   | 20.57%      | 24.45%      | 56.40%      | 51.07%      | 97.14%    |
| 7      | 7   | 68.34%      | 31.33%      | 76.98%      | 95.05%      | 97.13%    |
| 9      | 9   | 69.25%      | 68.56%      | 83.79%      | 33.59%      | 97.56%    |
| 11     | 11  | 71.58%      | 37.48%      | 86.88%      | 96.84%      | 97.37%    |
| 13     | 13  | 63.30%      | 25.53%      | 85.37%      | 8.59%       | 97.20%    |
5. Conclusion

In this paper we developed a new method to recognize action from videos with both CNNs and RNNs. The experiments verified that our framework is effective. Deep convolutional neural networks can effectively extract the images’ features and deep LSTM cell based recurrent neural networks do well in processing sequenced data. We show that the Inception_Residual model improve the performance of CNN. We also show that the multilayered LSTM architectures give better performance than single-layered LSTM.

References

[1] R. A. Rensink. The dynamic representation of scenes. Visual Cognition, 7(1-3):17–42, 2000.
[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. ICML, 2015.
[3] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
[4] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: a convolutional neural-network approach.[J].
[5] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]// Computer Vision and Pattern Recognition. IEEE, 2014:1-9.
[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008,pp. 1–8.
[7] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, “Structured learning of human interactions in tv shows,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, no. 12, pp. 2441–2453, 2012.
[8] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In BMVC 2008-19th British Machine Vision Conference, pages 275–1. British Machine Vision Association, 2008.
[9] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th international conference on Multimedia, pages 357–360. ACM, 2007.
[10] Liu, Jingen, Y. Yang, and M. Shah. "Learning semantic visual vocabularies using diffusion distance." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on IEEE, 2009:461-468.
[11] Liu J, Luo J, Shah M. Recognizing realistic actions from videos in the Wild[J]. 2009:1996-2003. IEEE Transactions on Neural Networks, 1997, 8(1):98.
[12] Wu R, Yan S, Shan Y, et al. Deep Image: Scaling up Image Recognition[J]. Computer Science, 2015.
[13] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4):págs. 212-223.
[14] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 1994, 5(5):157-166.
[15] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.
[16] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. 2016.
[17] Haşim Sak, Andrew Senior, Françoise Beaufays. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition[J]. Computer Science, 2014:338-342.
[18] Graves A. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[19] Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on,pp. 6645–6649. IEEE, 2013.
[20] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[J]. Computer Science, 2015.
[21] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
[22] Sharma S, Kiros R, Salakhutdinov R. Action Recognition using Visual Attention[J]. Computer Science, 2015.