E-sports analysis data acquisition algorithm based on convolutional neural network

Liulei Li¹, Wanning Zhu¹*, Cheng Yu¹, and Liqing Nong²

¹Department of Software Engineering, Jinling Institute of Technology, 211100, Nanjing, China
²Secretary major, Guangxi University of Finance and Economics, 530003, Nanning, China

Abstract. At present, e-sports has become one of the most important industries. How to analyze e-sports data has become an urgent problem to be solved. Currently, some hot competition items do not provide data interfaces, so that the training set required for data analysis cannot be directly and accurately acquired. Data can only be obtained by watching video games in person. This method is obviously inefficient and the accuracy cannot be guaranteed. This paper proposes a data acquisition algorithm based on convolutional neural network algorithm. It also introduces transfer learning, improves the sample training method and data acquisition method, and finally solves the problem of data acquisition. According to the test, this algorithm achieves about 91% accuracy.

1 Introduction

In the League of Legends s7 finals that just ended in 2017, the number of viewers in each channel reached 60 million. It can be seen that the impact of e-sports on human beings is increasing, and the economic benefits brought by it are also growing rapidly. Therefore, it is necessary to use the achievements made by people in data mining at the present stage to analyze the data of e-sports, and then present them to the audience for more exciting games and gain greater economic benefits. Due to the particularity of e-sports itself, the use of big data to conduct tactical analysis of e-sports was carried out late until 2012 when the EHOME team began to formally use big data to conduct tactical analysis on opponents in dota2 competitions. Data acquisition is the starting point of analysis of e-sports tactics work. However, League of Legends, one of the most popular international competitions, does not provide open data interfaces, making the use of big data for tactical analysis difficult from the first step. The algorithm proposed in this article solves this problem and achieves a correct rate of 91%.

2 Convolutional neural network analysis game video export position

2.1 Knowledge preparation

*Corresponding author: zhuwanning@jit.edu.cn
2.1.1 Inception convolutional neural network architecture

Since AlexNet made a historical breakthrough in 2012, before the GoogLeNet came out, the mainstream network architecture breakthrough was generally a deeper network and a wider network. But purely deepening the network has many disadvantages. Inception-v3 [1] was born in such circumstances, increasing the depth and width of the network while reducing the parameters.

2.1.2 Transfer learning

In most of the tasks of machine learning, we assume that the data used in training and testing obeys the same distribution and originates from the same domain. However, in practical applications, this assumption is difficult to establish and often encounters some problems. At this time, knowledge transfer has played a role, that is, to transfer knowledge in the B field to the A field and improve the classification effect of the A field. Transfer learning [2], as a new learning paradigm, was proposed to solve this problem.

2.2 Position analysis algorithm introduction

For the currently proposed MOBA class e-sport data analysis algorithm, a very important type of data is the hero's real-time location in the game. The hero's real-time location can be used to infer the player's real-time action and subsequent actions. The algorithm proposed in this paper solves the real-time position acquisition problem of the hero. For any hero-view video of any game, picture of each frame of this video can be input to this algorithm and be analyzed to obtain this picture's position in the entire large map.

First of all, as shown in Figure 1, the large map is divided into multiple regions, and the picture of each frame in the input video are matched with these regions. According to the principle of convolutional neural network described by Zeiler MD, Fergus R [3], CNN processing of input images is actually performing edge detection and then rising to feature extraction, thus classifying the input image. The size of the divided area is considered to be the size of the picture displayed by one screen in the normal game. Then determine the number of coordinates, that is, the number of regions, combined with the size of a single region and the impact of regional overlap on the classification results, divide the large map into 31 (horizontal) * 31 (vertical) = 961 regions. The red box shown in the figure is the size of one region. CNN training requires a large number of samples. If there is not a large number of samples, there is no advantage compared to other algorithms [4]. However, we do not have enough manpower to mark enough samples to train 961 categories. Therefore, this paper uses transfer learning to solve the problem of insufficient samples, which mainly reflected in the parameters of the network initialization to the parameters that Inception-v3 victory in the ILSVC contest. Because there are many kinds of objects in the ILSVC competition and training sets are abundant, the parameter generalization ability is good. Inspired by Krizhevsky A [5]'s framework for achieving good performance in the ILSVC contest, this article made a comparison between following two changes. The first one adds a 1*1 convolutional layer to the original network, mainly considering that although Inception can extract effective features, after all, the sample that trained its feature extraction ability has a very different style from the sample style of this article. So through the sample of this question to train and update the parameters, making it more suitable for the problem sample environment. The second addition of two 1*1 convolutional layers is to prevent the under-fitting condition that may occur with the first network. And the number of first-level units added is 256, and the second-level is 961. This forms a bottleneck unit [6]. This increases the depth of the network and prevents under-fitting. At the same time, it
also reduces the number of parameters.

Fig. 1. Map division.

Fig. 2. Network structure.

2.3 Position analysis algorithm

The structure of the network is shown in Figure 2. This algorithm divides 961 categories so create 961 new folders. Each folder contains a category of pictures. All folders are collectively referred to as folders:
define first_layer = the first 1*1 convolution layer that added,
second_layer = the second convolution layer that added,
batch_size = number of samples for one training,
learning_rate = learning rate of this network, step = total number of training steps
initialize first_layer and second_layer with all parameters = 0
define imagelist, images, class_truths, logits
for every folder in folders:
  for every file in folder:
    if file.suffix in [png, jpg]
      file = file resize to 90px*160px
tensorout = run file in inception-v3
    imagelist[folder.name] add tensorout
  for _ in range(step):
    for _ in range(batch_size):
      class_index = random index of class in imagelist
      image = random image in imagelist[class_index]
      class_truth = [0,0,0,0,...,0,0]
      class_truth[class_index] = 1
      images.append(image)
      class_truths.append(class_truth)
      logits = commit convolution between images and first_layer
      if second_layer != null:
        logits = commit convolution between logits and second_layer
      loss = calculate softmax crossentropy with logits=logits labels=class_truths
      commit gradient_descent on cross_entropy with learningrate=learning_rate

2.4 Algorithm analysis

The added layers’ parameters are initialized to 0. The parameters of these two convolutional layers will be updated during training and learning. Then for the 961 folders that contain the training set, each folder is scanned and stored in a two-dimensional array. Then the picture will be pre-processed. Preprocessing Step 1: Reduce the picture to 90x160 pixels. Step 2, input it to the original Inception-v3 network and record the output.

Finally is training, the preprocessed picture is first convoluted on the first 1*1 convolutional layer. If there is a second 1*1 convolutional layer, convolution is performed on the second 1*1 convolutional layer. The softmax[7] normalization process is finally performed, and the obtained data is the CNN judgment of which area this image belongs to. The parameters of the last increased 1*1 convolutional layer are updated by gradient descent and back propagation.

3 Experimental results and analysis

The experimental results are shown in the following table. The first line indicates whether to add a layer of convolution or two layers. The first column is the learning-rate.

<table>
<thead>
<tr>
<th></th>
<th>One convolution layer</th>
<th>Two convolution layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>71.2%</td>
<td>1.7%</td>
</tr>
<tr>
<td>0.03</td>
<td>86.3%</td>
<td>2.1%</td>
</tr>
<tr>
<td>0.01</td>
<td>91.2%</td>
<td>2.2%</td>
</tr>
<tr>
<td>0.003</td>
<td>42.1%</td>
<td>0%</td>
</tr>
<tr>
<td>0.001</td>
<td>22.1%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Then the picture will be pre-processed. Preprocessing Step 1: for every folder in folders:
- define imagelist, images, class_truths, logits that contain the training set, each folder is scanned and stored in a two-dimensional array. The convolutional layers will be updated during training and learning. Then for the 961 folders:
  - initialize first_layer and second_layer with all parameters = 0
  - for_inrange(step):
  - finally is training, the preprocessed picture is first convoluted on the first 1*1 convolutional layer. If there is a second 1*1 convolutional layer, convolution is performed on the second 1*1 convolutional layer. The softmax normalization process is finally performed, and the obtained data is the CNN judgment of which area this image belongs to.

2.4 Algorithm analysis
- The added layers’ parameters are initialized to 0. The parameters of these two layers are initialized to 0. The parameters of these two layers are initialized to 0. The parameters of these two layers are initialized to 0.
- The experimental results are shown in the following table. The first line indicates whether the network parameters are randomly initialized. The second experiment initialization parameters are AlexNet's parameters trained on the Imagenet data set, and a full-connection layer is added, only the last layer added is updated during training. The experimental results are shown in Table 2. It can be seen that the network with randomly initialized parameters in the test, the correct rate of verification quickly reaches 100%, but the test set performance is very poor, and it can be judged that the network is overfitting because the training set is too small. While the network parameters are initialized to the parameters of the network trained by the Imagenet data set, although the accuracy of the verification set is steadily rising and high, the actual test found that the accuracy is not high. From this, it can be found that it is necessary to choose a more advanced and sophisticated network as the initial structure of the small sample migration learning network. Finally, there is a problem that the correct Inception-v3 network in the test set does not meet the Inception-v3 performance on the ILSVRC. There are two reasons for this: 1. This is the transfer learning test. The sample is a picture in the game, which differs greatly from the style of the ILSVC training set. 2. MOBA game maps are similar in many places. The colors are close and the features are mostly curved walls. The maps are symmetrical, which further increases the difficulty of distinguishing.

<table>
<thead>
<tr>
<th>Table 2. Accuracy of two methods to initialize parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate accuracy</td>
</tr>
<tr>
<td>Test accuracy</td>
</tr>
</tbody>
</table>

From the result of Table 1, we can find that the highest Top-5 correct rate is about 91%. To determine whether this improvement can solve the problem of obtaining higher accuracy in the case of insufficient samples, we have conducted two other experiments in this paper. Both experiments use the classical AlexNet network architecture. The difference is that the first experimental network parameters are randomly initialized. The second experiment initialization parameters are AlexNet's parameters trained on the Imagenet data set, and a full-connection layer is added, only the last layer added is updated during training. The experimental results are shown in Table 2. It can be seen that the network with randomly initialized parameters in the test, the correct rate of verification quickly reaches 100%, but the test set performance is very poor, and it can be judged that the network is overfitting because the training set is too small. While the network parameters are initialized to the parameters of the network trained by the Imagenet data set, although the accuracy of the verification set is steadily rising and high, the actual test found that the accuracy is not high. From this, it can be found that it is necessary to choose a more advanced and sophisticated network as the initial structure of the small sample migration learning network. Finally, there is a problem that the correct Inception-v3 network in the test set does not meet the Inception-v3 performance on the ILSVRC. There are two reasons for this: 1. This is the transfer learning test. The sample is a picture in the game, which differs greatly from the style of the ILSVC training set. 2. MOBA game maps are similar in many places. The colors are close and the features are mostly curved walls. The maps are symmetrical, which further increases the difficulty of distinguishing.

4 Conclusion

This article uses convolutional neural network to solve the traditional e-sports tactics analysis data acquisition difficult, slow, requires a lot of artificial problems. Can analyze a large number of game videos in a very short time, get a lot of data, and the correct rate reached 91%. This paper analyzes two possible causes of the failure to achieve a high level of accuracy. From these two perspectives, further research can be conducted to improve the accuracy of the network.

Acknowledgments

This article is funded by the Jinling Institute of Science and Technology, a high-level talent research startup fund, and a web user behavior analysis and research project based on quantum algorithms(No. jit-b-201624) and the Social Engineering Network Behavior Analysis(No.2017YFB0802800).

References


