Evolving Deep Neural Networks for Movie Box-office Revenues Prediction

Yao Zhou* and Gary G. Yen †
*Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China 610065
†School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, USA 74078
Email: yaozhou@okstate.edu, gyen@okstate.edu

***Abstract*—Reliable prediction of movie box-office revenues can greatly reduce the financial risk in the film industry, but accurate prediction is not easy to obtain. Recently, deep neural networks has been applied on movie box-office revenues prediction problems as a promising solution. However, the architecture has a significant impact on its performance, and generally involves a heavy burden of manually designing which is unable to traverse the space of possible architectures efficiently. As a result, the applicability and performance of deep neural networks are severely limited. This paper proposes a new evolutionary algorithm for evolving deep neural networks for movie box-office revenues prediction. In particular, a deep neural network that fuses features extracted from movie posters by a convolutional neural network is introduced first, then a set of novel genetic operators are designed correspondingly. The proposed method can automate the deep neural network architecture designing and aim to search the optimal architecture for movie box-office revenues prediction. Experiments carried out on the Internet Movie Database (IMDB) dataset show that the proposed algorithm achieves superior performance compare to other competitive approaches.**

I. Introduction

The ability to predicting the ultimate revenues affects the marketing activities decision and has been a major research focus in the film industry. Reliable prediction of movie box-office revenues can reduce the financial risk, provide guidance that can assist movie producers to make managerial decisions in the production process, such as budget and screen allocations, as well as consumers to make their choices. Although movie financial success has been considered as an unpredictable problem [1], research works have been investigated on developing approaches to predict movie box-office revenues [2], [3].

Deep neural networks (DNNs) has been successfully employed in a variety of applications [4], [5], including image recognition [6], natural language processing [7] and intelligent game playing [8], to name a few, which is mainly credit to the availability of large amount of labeled data [9] and advancement of computing power (e.g., graphic processing units). As a variant of the DNNs, convolutional neural networks (CNNs) [10] eliminated the dependence on feature designing, which is able to automatically learn representations from high dimensional data in a data-driven fashion. Thus discriminative features related to the given problem can be effectively learned from data, instead of crafting with prior domain-knowledge, which could robustly characterize features for a given task. Recently, DNN has been exploited for movie box-office revenues prediction [11], and emerged as a superior approach for modeling complex relations exihibited in movie box-office revenues data.

Although neural networks has achieved remarkable performance on movie box-office revenues prediction problems, a well-known fact is that the architecture configuration of neural networks has a significant influence on its performance [1], [12], [13]. In order to obtain a good architecture, many parameters require to be carefully designed by developer, including the number of layers, the number of nodes in each layer, activation functions, etc. With the advances in deep neural networks, this problem becomes even more severe since there are much more parameters need to be set (e.g., dropout probabilities [14]). Generally, the DNN architectures are manually configured with a trial-and-error fashion, which is considered low efficient and make it infeasible to test all possible combinations within a reasonable time. On the other hand, with a rapidly increased number of DNN based applications, a large amount of well-designed DNN architectures can be used as reference models. Therefore, adapting existing DNN architectures to a new problem that hasn't been addressed by DNN yet could be a promising way for accelerating DNN architectures designing.

In order to find an optimal or near optimal DNN architecture, a great effort has been made on automated DNN architecture designing. In particular, evolutionary algorithms have been widely explored as the main idea for this purpose, which have drawn attentions from researchers in the field of image classification and natural language processing [15], [16], [17]. Specifically, by designing an appropriate genotype as DNN architecture representation, as well as corresponding genetic operators, evolutionary algorithm can be used as a heuristic searching tool for searching the optimal DNN architecture in a specified search space, which is much more efficient than manually designing and may obtain a DNN architecture with improved performance. However, as it has not been explored in the problem domain of movie box-office revenues prediction, developing an evolutionary algorithm to search near optimal DNN architectures under those applicable considerations still remains to be investigated.

In this work, a new evolutionary algorithm for optimizing DNN architectures to predict movie box-office revenues is presented. In particular, a genotype representation is designed

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

to encode the most important parameters of possible DNN architectures, including the number of layers, the number of nodes in each layer, activation functions, and dropout probabilities. Besides, the incorporation of a CNN is considered as part of the genotype representation as well, due to it acts as feature extractor for movie poster and needs to be appropriately integrated with other features in the DNN. Moreover, a reference model based method is proposed for individual initialization by transferring prior-knowledge on DNN architecture designing, and a novel crossover operator is designed to effective exchange information between individuals by considering all possible combinations. Other genetic operators including mutation, evaluation and selection are adapted to the designed genotype representation as well for generating competitive offspring. Lastly, experiments on the Internet Movie Database (IMDB) are carried out and comparison with other competitive approaches verifies the effectiveness of proposed algorithm.

The rest of this paper is organized as follows. Section II briefly reviews related work, while Section III presents the evolutionary algorithm designed for optimizing the architecture of DNN. Experimental setting, dataset description, experimental results and analysis are detailed in Section IV. Finally, Section V concludes this paper and provides some future directions.

II. Related Work

Movie box-office revenues prediction has been widely investigated over the last decade [1], [3]. Existing methods for predicting movie box-office revenues can be categorized according to the features and prediction techniques adopted. For instance, script and allocated production budget were selected as pre-production features, and a natural language processing (NLP) and kernel combined approach was adopted to assess the box-office potential [18], which resulted in better performance than several existing benchmark methods for movie box-office prediction. In addition to genre, budget, actors, Motion Picture Association of America (MPAA) rating and critics' reviews, features of movie trailer are investigated as well, and then correlation analysis was exploited for the financial success prediction [19]. Moreover, effects on movie box-office revenues from social media such as Facebook, Twitter, and YouTube have been empirically investigated with ordinary least square regression models [20]. Also, from the aspect of evolutionary algorithms, a genetic algorithm based approach was exploited for selecting features in movie box-office revenues prediction problem [21]. In particular, a fitness function which considers both model's prediction error and the number of selected variables were adopted to guide the search direction, which can alleviate feature selection task from searching in a large number of social network service (SNS) variables. Although the above methods exploited movie-related data from advertising and social media, the prediction techniques employed were mainly based on linear algorithms, which may limit their representation ability for processing high-dimensional and heterogeneous data from multiple modalities.

Due to the remarkable capability of modeling the complex relations, artificial neural networks (ANNs) has been extensively adopted for movie box-office revenues prediction. For example, a multi-layer perceptron (MLP) neural network has been developed for box-office revenues prediction [1], which consists of two hidden layers with sigmoid activation function, and each of them use 18 and 16 nodes, respectively. This parameterization was achieved by some preliminary experiments. The built neural network accepts many types of movie-related metadata as input, and classifies the given movie to a predefined category which indicates the degree of financial success. In addition, another neural network based approach with different architecture has been proposed to predict the financial success of movies [12]. It empirically employed 30 and 10 nodes in the first and second hidden layers, respectively. Although ANNs appears to be a promising solution, it is daunting to researchers for tuning the hyper-paramseters of network architecture.

Recently, contributed to the advances of new functionalities [22], normalization techniques [23], and connection schemes [24], DNNs has achieved considerable performance improvement than conventional ANNs on a variety of tasks. In particular, a DNN based method has been proposed for movie box-office revenues prediction [11] by considering the aspects of features and prediction techniques. The main idea is to construct a CNN model to extract features from movie posters, then a DNN is designed to incorporate those features for predicting movie box-office revenues, which resulted in better performance than other competitive approaches. However, as DNNs are generally more complex than ANNs, the problem of finding an optimal or near optimal architecture has become challenging in this field. Therefore, it is necessary to develop an automated approach for searching best configurations of DNN for movie box-office revenues prediction, which is expected to relieve the burden of manual-tuning, and further improve the performance in prediction.

Multiple approaches have been proposed by employing evolutionary algorithms to optimize neural network architectures and parameters [25] in different application domains. For instance, a genetic algorithm has been developed to evolve most essential parameters to train a DNN for malware classification [26]. Also, evolutionary techniques such as gravitational search algorithm (GSA) [27] and particle swarm optimization (PSO) [28] has been studied for optical character recognition (OCR) task with CNNs [29], where evolutionary algorithms are used to search optimal weights of CNNs. Additionally, a genetic algorithm and back-propagation hybrid approach is proposed for human action recognition using CNNs [30]. Different from existing research works, in this study, we proposed a set of novel genetic operators for optimizing critical parameters of DNNs, as well as discovering the best feature fusion scheme in the proposed evolutionary algorithm for movie box-office revenues prediction problems, which has been less investigated.

III. The Proposed Method

In this section, the proposed method will be detailed in two parts: the basic DNN architecture for movie box-office revenues prediction and the evolutionary algorithm for optimizing the DNN architecture.

A. Basic DNN architecture

The basic DNN architecture for movie box-office revenues prediction is constructed according to the features selected. In this work, commonly used movie-related data, such as genres, duration, star value, budget, etc. are adopted. Besides, a movie poster is always incorporated as a movie-related data, since it conveys almost all important information about a movie, which may significantly affect the viewers' decision, and can implicitly contribute to the financial success of a movie. In particular, a CNN is firstly constructed as a feature extractor for movie posters. Then the features of movie posters are integrated with other movie-related data in a DNN as shown in Fig. 1.

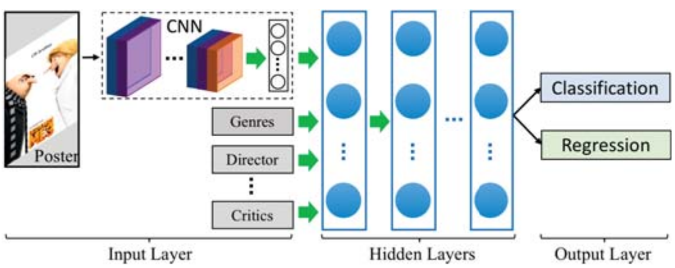


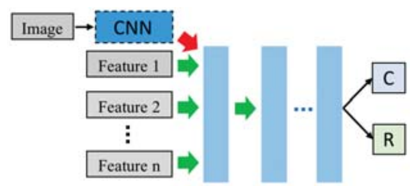
Fig. 1: The basic DNN architecture for movie box-office revenues prediction. It consists of three stages, where movie posters are processed by a CNN, and the extracted features are integrated with other movie-related data and feed to hidden layers. In the output layer, classification and regression are performed simultaneously as a multi-task learning paradigm.

The basic DNN architecture can be grouped into three stages. The first stage describes the input of DNN, where movie poster is firstly processed by a CNN, then integrated with other movie-related data and feed to the second stage which consists of hidden layers. There could be multiple hidden layers with different numbers of nodes and different activation functions, as well as regularization techniques such as dropout [14]. Thus, it can provide the capability of modeling arbitrary nonlinear relations which may exist in the complex movie-related data. In the third stage, a multi-task learning paradigm is employed by performing classification and regression simultaneously. As movie box-office revenues is originally in continuous form, and later be discretized to different financial success categories, both of them can be utilized for training the DNN. Here the classification act as the main task, while regression is used as a regularizer. Accordingly, a joint cost function could be defined for gradient based optimization methods. Formally, the parameters of the DNN is denoted as θ, and for given data samples x, the output of classification and regression are denoted as $Pr(\theta, x)$ and $a(\theta, x)$, respectively. Then the optimization can be achieved as follows:

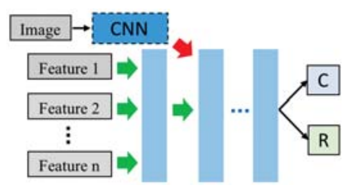
$$\arg\min_{\theta} \left[-w_1 * c \cdot \log \Pr(\theta, x) + w_2 * \|y - a(\theta, x)\|^2 \right] \quad (1)$$

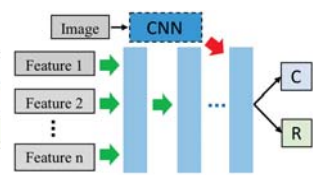
where c and y represent ground truth of movie box-office revenues in categorized and continuous forms, respectively. w_1 and w_2 are weighting coefficients.

On the other hand, there are multiple options for feature fusion. The scheme depicted in Fig. 2a terms as low level fusion, which means the fusion happens at the first layer. On the contrary, the fusion can happen at the last layer as well, which is represented as high level fusion (Fig. 2c). Also, it is possible to fuse features at any of the intermediate layers, termed as middle level fusion (Fig. 2b). A comparison of those three fusion schemes is depicted in Fig. 2. As it can be seen, the features fused at lower level will be passed through more nonlinear transformations than those fused at higher levels, while it may be prone to over-fitting when using many hidden layers. In contrast, those fusions happened at higher level may lead the features to be less representative, and result in performance deterioration. Therefore, it is necessary to find an appropriate fusion scheme for the basic DNN architecture.



(a) Low level fusion





(b) Middle level fusion (c) High level fusion

Fig. 2: Comparison of low level fusion (a), middle level fusion (b) and high level fusion (c). The main difference is the position where the fusion happens. In the output layer, C and R represent classification and regression, respectively.

B. Evolutionary algorithm for optimizing DNN architecture

In order to train a DNN, important parameters including the number of layers, the number of nodes and activation functions of each layer require to be set at the beginning. That is, the DNN architecture needs to be determined before the training. However, the optimal or near optimal DNN architecture in the large search space is unknown and may vary from problem to problem. Instead of manually designing the DNN architecture in a trial-and-error fashion, an evolutionary algorithm that employs heuristic search over the possible parameter configurations is proposed for discovering optimal

DNN architectures, where each individual represents the parameters for constructing a DNN architecture. In particular, the fusion scheme is encoded in the individual representation in an innovative manner, and optimized accordingly. Two objectives of each individual are defined for optimization, one is the classification accuracy on the validation dataset, while the other is the complexity of the evolved network architecture.

As a DNN could be constructed with as many layers as possible when the memory capacity allows, the search space would be unbound, which is problematic for pursuing possible best architecture. Meanwhile, it is a common exercise to build a DNN architecture by referring to existing successful ones, rather than beginning from scratch. Based on the above considerations, the searching process begins with a reference DNN architecture which has been empirically proven to have a good performance. Genetic operators employed in the evolutionary algorithm are detailed as follows.

1) Genotype design: Genotype in the evolutionary algorithm is designed to represent the parameters of DNN architectures. As the number of layers in DNNs can be different from each other, a sequential varied-length encoding strategy is adopted for this situation. That is, the length of a genotype representation is the same to the depth of a DNN. In particular, a gene in the genotype representation encodes parameters including the number of nodes, activation function and dropout probability of each layer, as well as the fusion scheme, which is depicted in Fig. 3.

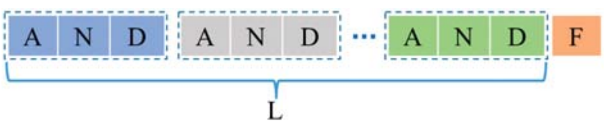


Fig. 3: The genotype representation of a DNN. It consists of L layers, where the parameters A, N, and D denote the activation function, the number of nodes of each layer, and the probability of each dropout layer, respectively. The fusion scheme is represented as F, which is a number between 0 and L that indicates the position to perform feature fusion.

2) Population initialization: The population consists of individuals for a given size. Mostly, the individuals are randomly initialized. However, it may generate many trivial initial solutions which would lead to performance deterioration of the evolutionary algorithm. Instead, a reference model based method is adopted for initializing individuals. In particular, each individual is initialized by randomly selecting part of the layers from the reference model, then the parameters of the layers (e.g. activation function, number of nodes in each layer, and dropout probability) are re-initialized with random perturbations. The fusion scheme is randomly initialized in the range of 0 to the number of layers.

3) Crossover: Based on the presented genotype for representing varied-depth DNN architectures, a crossover operator is needed to be designed for effectively exchanging information between individuals. However, the varied length genotype representation poses a problem when applying conventional crossover operators. To this end, a novel crossover operator is proposed to effectively recombine information of parent genotypes. The basic idea is to ensure that all possible recombinations of genes can be covered even though the parent genotypes are differ in length. An illustrative example is shown in Fig. 4. Two parent individuals are denoted as A and B whose length are n and m, respectively. The fusion parameter is represented as F. Assuming $m < n$, then possible combinations can be constructed by matching all genes in B to a same sized subset of genes in A, which result in a number of $\frac{n!}{(n-m)!}$ possible combinations. Each of them consists of a set of pairs that represent recombination operations. Then a probability is set to determine if a pair in the combination set will be selected as the final crossover operation. Since the validity of F depends on the depth of a DNN, and inappropriate crossover will lead to invalid solutions, F is kept unchanged during crossover.

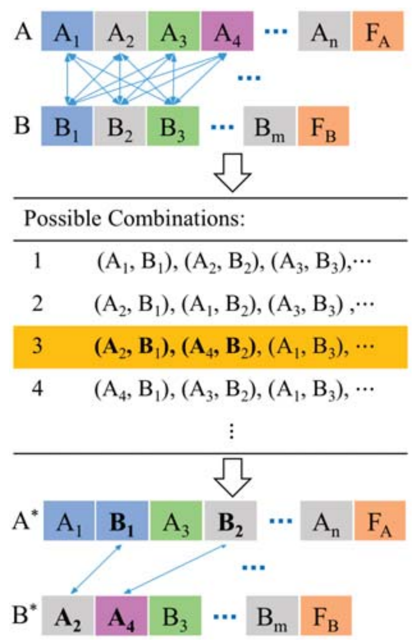


Fig. 4: The proposed crossover operator. Possible combinations for the parent genotypes A and B are shown in the table. Assuming the third one is adopted for recombination, then (A_2, B_1) and (A_4, B_2) are independently selected as the final crossover operations by a given probability, and result in offspring A^* and B^*.

4) Mutation: The presented crossover operator provided the capability of exploitation on existing varied-length genotypes. To avoid being trapped in local optimum, the ability of exploration is required as well. Accordingly, a mutation operator is designed and applied on each of the parameters encoded in the genotype. In particular, the number of nodes in each layer, dropout probability, activation function and fusion scheme are perturbed with a given probability, which encourages exploration. Besides, another probability determines to add or remove a layer, where the number of nodes of newly added layer will be initialized around the minimum value of that in the reference model, and the corresponding activation function and dropout probability of the new layer will be randomly initialized with a valid value. A layer can be removed only when the number of hidden layers is greater than 1 and the fusion scheme will be adjusted accordingly to ensure its validity.

5) Evaluation and Selection: Evaluation aims at measuring the objectives of each individual, and later provides guidance for choosing parent solutions in the selection process. As an intuitive indicator of the quality of a DNN architecture, accuracy on validation dataset is adopted as the first objective to be maximized. On the other hand, model complexity is important as well, due to its effects on the efficiency and generalization ability of a DNN, thus selected as the second objective to be minimized. Based on the above two-objective designing, NSGA-II selection strategy [31] is employed for promoting convergence and diversity. Moreover, elites from the population are saved and then passed onto the next generation, in order to maintain the stability of the evolution process.

Based on the genetic operators listed above, the proposed method for evolving DNN architecture for movie box-office revenues prediction can be described by the framework as shown in Algorithm 1.

Algorithm 1: Framework of proposed method

Input : The reference DNN $\mathcal{M}$, the number of generations T, the population size N, the crossover and mutation probabilities p_c and p_m, and the dataset $\mathcal{D}$.

Output: Best DNN architecture from evolution

```
P_0 ←initialize the population from M with size N;
for t = 0, 1, ··· , T do
    Q_t ← generate offspring population from P_t using
      designed crossover and mutation operators with
      probability of p_c and p_m, respectively;
    R_t ← P_t ∪ Q_t;
    F_t ←evaluation(R_t, D);
    P_{t+1} ←selection(R_t, F_t);
    terminate when stopping condition satisfied
end
```

IV. Experiments

This section details the experimental settings, including dataset description, DNN training configuration, and parameters setting for the evolutionary algorithm. Also, results and analysis about the experiments will be described, as well as comparisons with some chosen competitive methods.

A. Dataset description

The dataset used in experiments is acquired from the Internet Movie Database website (*imdb.com*), and the corresponding box-office revenues for each movie are obtained from *the-numbers.com*. Totally $3,807$ movie samples are collected, and uniformly categorized into six groups according to their ranked box-office revenues (i.e., from \$200,000 to \$760,505,847.) [12]. The movie-related metadata used are described in Table I. A number of 27 attributes are selected as predictive indicators. In particular, numeric data are normalized to the interval of $[0, 1]$, and movie posters are pre-processed by substracting the mean value used in [32].

B. Performance metrics

For the aim of evaluating the performance of the proposed method, average percent hit rate (APHR) [1] is deployed as the evaluation metric, which indicates the percentage of correct classifications. In particular, two types of APHR are employed as performance metrics with respect to different classes:

- Absolute accuracy: This metric measures the exact hit rate (termed as Bingo), meaning that only a classification exactly matches the ground truth would be considered as correct.
- Relative accuracy: Based on Bingo, and with further consideration of those classification results in adjacent classes (1-away).

Formally, the APHR can be computed as follows:

$$APHR_{Bingo} = \frac{1}{n}\sum_{i=1}^{K} c_i \tag{2}$$

$$APHR_{1-away} = APHR_{Bingo} + \frac{1}{n}\sum_{i=1}^{K} c_{i-1} + c_{i+1} \tag{3}$$

where *n* represents the total number of samples, *K* is the total number of classes, and c_i denotes the total number of samples correctly classified as class *i*, when $i \leq 1$ or $i \geq K$, $c_i = 0$.

C. Parameters setting

The reference model [11] used for initializing individuals is shown in Table II. It can be considered as a way for transferring prior-knowledge, because a DNN is generally built by referring to existing models, rather than from scratch. The upper bounds of layers and nodes for each individual are set to be a doubled value of that in the reference model, and the set of predefined activation functions is composed of *sigmoid*, *tanh* and *relu* [22]. Concatenation and Adam [33] are adopted for feature fusion and optimizing weight parameters of the DNN, respectively. To balance the importance of the two tasks in the output layer, learning rates with regard to loss function is set to 0.05, and w_1, w_2 in Eq. 1 are set to 0.5 and 0.005, respectively. Empirically, the batch size and training steps are set to 200 and 1000, respectively. The DNN architecture is implemented within Tensorflow [34], and running on a computer with a GPU of NVIDIA GeForce GTX TITAN X.

In Algorithm 1, the size of population $\mathcal{P}$ and the generated offspring population $\mathcal{Q}$ are both set to 50, the number of generations T is set to 100, and stopping condition satisfies when the performance doesn't improve for over 15 generations. To evaluate the average performance of proposed algorithm, 20

TABLE I: Descriptions of movie-related metadata

Genre	Possible genres: Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, etc.	Vector
The production budget	Investment of a film, which is not typically published, but can be estimated from advertisements and social media	Numeric
Cast & Crew	Impact of superstar actor/actress, famous director, etc. Facebook "likes" of them are used as a measurement.	Numeric
Duration/Running time	A numerical variable. Statistical analysis shows that it is, on average, around 100 min.	Numeric
User rating	Voting and scores of movies, which reflect the sentiments of online users.	Numeric
Social commentary	Professional movie critiques and viewer reviews. For the sake of efficiency, amount of those comments were adopted.	Numeric
Movie poster	It delivers most of the important information about a movie, and act as an advertising medium.	Image

independent runs are tested, and each run take around 6 hours. In order to accelerate the algorithm, 20 training steps are performed in the evaluation. The probabilities of crossover and mutation are 1 and $1/n$, respectively, where n is the length of the individual, and a layer has 50% chance to be added or removed. Moreover, 10% training data is kept for validation, and relative accuracy (*1-away*) is adopted as an objective to be maximized, while the number of layers of a DNN is adopted as another objective to be minimized.

TABLE II: The reference DNN architecture. FC represents fully connected layer.

Layer	Type	Units
1	Input	-
2	FC+relu	72
3	Dropout(0.6)	-
4	FC+relu	128
5	Dropout(0.6)	-
6	FC+relu	256
7	Dropout(0.6)	-
8	FC+relu	128
9	Dropout(0.6)	-
10	FC+relu	72
11	Dropout(0.6)	-
12	Softmax	6

D. Experimental results

The effectiveness of the CNN component in the reference DNN architecture is investigated first. As it can be seen from Table III, the performance get improved when CNN is integrated for extracting features from movie posters.

TABLE III: Performance comparison of DNN architectures with or without the CNN component.

Architecture		1(%)	2(%)	3(%)	4(%)	5(%)
w/o CNN	Bingo	52.44	49.12	53.89	50.13	51.67
	1-Away	87.45	87.33	85.44	86.22	85.78
with CNN	Bingo	**53.22**	**50.00**	**54.56**	**50.44**	**52.78**
	1-Away	**90.33**	**88.11**	**87.00**	**88.78**	**88.78**

Once the evolutionary algorithm is terminated, the best individual is picked out according to its accuracy achieved on validation dataset. As shown in Fig. 5, the best DNN architecture consists of 3 hidden layers, where there are 36, 117, and 194 nodes established at the first, second, and last layer, respectively. The *relu* is selected as the best activation function for all hidden layers, and the best dropout probabilities on these three layers are found to be 0.0, 0.4, and 0.4, respectively. Moreover, fusing features at the first hidden layer is selected as the best fusion scheme, which is considered a type of middle level fusion.

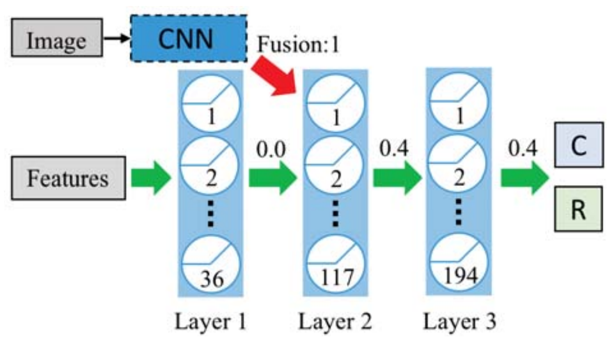


Fig. 5: The best network architecture found from evolution. Three hidden layers are established with 36, 117, 194 nodes, respectively. The number between each two layers indicate the probability of dropout, *relu* activation function is employed for each layer, and fusion happens at the first hidden layer.

Compared to the reference DNN architecture, the best architecture from evolution has a smaller number of layers and nodes, and uses an optimized fusion scheme. An interesting parameter in the evolved DNN architecture is that the dropout probability of the first hidden layer is set to 0, which implies there is no need to add dropout at the first layer to enhance the generalization ability for this problem. Also, the *relu* activation function is found to be better than *tanh* and *sigmoid* in the DNN for movie box-office revenues prediction, which is consistent with a previous research finding [11].

To justify the effectiveness of the proposed method for evolving DNN, existing movie box-office revenues prediction methods including Support Vector Machines (SVM) [35], Random Forest (RF) [36], Multi-layer BP network (MLBP) [12], and a manually designed DNN [11] are trained on the same dataset as competitors, all features including movie posters are adopted, and a 5-fold cross-validation scheme is employed to reduce the sensitivity of data splitting.

As it can be seen from Table IV, both the mean and best performance of evolved DNN achieved better performance than that of SVM, RF, and MLBP, contributed to the superior capability of modeling high complex relation and good gen-

TABLE IV: Comparison of different methods for movie box-office revenues prediction. (The proposed method is termed as *Evolved DNN*, and numbers in first row represent different folds.)

Classifier		1(%)	2(%)	3(%)	4(%)	5(%)	Average(%)
SVM [35]	Bingo	33.07	35.19	34.39	33.33	35.85	34.37
	1-Away	63.62	65.74	65.61	64.02	65.21	64.84
Random Forest [36]	Bingo	48.28	45.76	47.35	49.21	48.28	47.78
	1-Away	85.19	81.09	82.01	83.86	82.14	82.86
MLBP [12]	Bingo	49.38	48.56	51.78	49.45	51.14	50.06
	1-Away	88.05	85.13	84.89	85.22	86.36	85.93
DNN [11]	Bingo	53.22	50.00	54.56	50.44	52.78	52.20
	1-Away	90.33	88.11	87.00	88.78	88.78	88.60
Evolved DNN (mean)	Bingo	**54.27**	**52.03**	**55.45**	**51.37**	**53.87**	**53.40**
	1-Away	**91.72**	**90.01**	**88.83**	**90.14**	**90.80**	**90.30**
Evolved DNN (best)	Bingo	**55.60**	**54.32**	**56.39**	**52.53**	**55.13**	**55.03**
	1-Away	**92.12**	**91.14**	**89.49**	**91.54**	**90.80**	**91.33**

eralization with an optimized architecture. It is worth noting that the best architecture from evolution outperforms the hand-designed DNN [11] as well, which demonstrated that the proposed evolutionary algorithm can find a DNN architecture with improved performance than that from trial and error, and the requirement of hand-crafting hyper-parameters of network architectures potentially limits the performance and applicability of DNN techniques. In particular, although the proposed evolutionary algorithm initialized each individual from a reference model, it is able to find a new architecture that achieves better performance than the referenced one, which provides the inspiration that incorporating prior-knowledge could be a promising means to improve the performance of evolutionary algorithms for searching DNN architectures.

E. Feature visualization

As a feature extractor, the CNN is firstly fine-tuned with a task for classifying movie posters, which outputs a result that indicates the movie is financial successful or not. In particular, the VGG-16 network [32] is adopted as the basic CNN architecture by removing its fully connected layers, then connected with a global average pooling layer and a fully connected layer with two nodes. Accordingly, the training set is divided into two categories based on the rank of each movie's box-office revenues. After the training converged, it achieved an accuracy of 65% on the validation set. Due to the limitation of a relative small number of movie posters, the CNN architecture is manually designed and fine-tuned from an off-the-shelf model [32] , and only fusion scheme is evolved by the proposed algorithm. The optimization of the CNN architecture is one of our future works. Meanwhile, in order to provide an intuitive understanding about the relations between movie posters and movie box-office revenues, features learned in the CNN are visualized [37] and shown in Fig. 6. As it can be seen from the visualization results, features related to financial success are mainly focusing on text areas, which indicates that the text designing may have a potential impact on movie box-office revenues.

Fig. 6: Visualization of learned features in CNNs. Features related to financial success are highlighted by the red areas and shown on the left side, and the original movie posters are shown on the right side.

V. CONCLUSION

Deep neural networks has been widely employed in a variety of domains. However, its performance is sensitive to network architecture designing. In particular, problems using multiple data modalities require complicated network architecture designing. This paper proposed a novel evolutionary algorithm for evolving deep neural networks for movie box-office revenues prediction, which uses movie-related multi-modal data to predict its financial success. First, a deep neural network architecture integrated with a convolutional neural network is introduced, where features of movie posters can be fused with other movie-related metadata for box-office revenues prediction. Then, a set of novel genetic operators is designed to optimize the architecture of deep neural networks. In particular, a genotype representation is presented for encoding parameters of the network architecture, and a reference model based method is proposed for initializing each individual. Moreover, novel crossover and mutation operators are designed for effectively exploring the searching space. Experimental results show that the proposed method is able to find the deep neural network architecture which achieves improved performance than existing competitive methods. Future investigations will focus on reducing the compuation cost, applying it for other multimodal prediction problems, as well as corresponding genetic operators designing.

Acknowledgment

This work is supported in part by the China Scholarship Council, and in part by the Miaozi Project in Science and Technology Innovation Program of Sichuan Province, China.

References

[1] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, no. 2, pp. 243–254, 2006.

[2] E. A. Antipov and E. B. Pokryshevskaya, "Are box office revenues equally unpredictable for all movies? evidence from a random forest-based model," *Journal of Revenue and Pricing Management*, vol. 16, no. 3, pp. 295–307, 2017.

[3] J. Duan, X. Ding, and T. Liu, "A Gaussian copula regression model for movie box-office revenues prediction," *Science China Information Sciences*, vol. 60, no. 9, p. 092103, 2017.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] Y. Zhou, H. Mao, and Z. Yi, "Cell mitosis detection using deep neural networks," *Knowledge-Based Systems*, vol. 137, pp. 19–28, 2017.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in neural information processing systems (NIPS)*, 2014, pp. 3104–3112.

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, C. Yutian, L. Timothy, H. Fan, S. Laurent, G. van den Driessche, G. Thore, and H. Demis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[9] J. Deng, W. Dong, R. Socher, J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] Y. Zhou, L. Zhang, and Z. Yi, "Predicting movie box-office revenues using deep neural networks," *Neural Computing and Applications*, 2017, doi:10.1007/s00521-017-3162-x.

[12] L. Zhang, J. Luo, and S. Yang, "Forecasting box office revenue of movies with bp neural network," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6580–6587, 2009.

[13] M. Ghiassi, D. Lio, and B. Moon, "Pre-production forecasting of movie revenues with a dynamic artificial neural network," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3176–3193, 2015.

[14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. Le, and A. Kurakin, "Large-scale evolution of image classifiers," *arXiv preprint arXiv:1703.01041*, 2017.

[16] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," *arXiv preprint arXiv:1711.00436*, 2017.

[17] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," *arXiv preprint arXiv:1703.00548*, 2017.

[18] J. Eliashberg, S. K. Hui, and Z. J. Zhang, "Assessing box office performance using movie scripts: A kernel-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2639–2648, 2014.

[19] A. Tadimari, N. Kumar, T. Guha, and S. S. Narayanan, "Opening big in box office? trailer content can help," in *Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2777–2781.

[20] C. Oh, Y. Roumani, J. K. Nwankpa, and H.-F. Hu, "Beyond likes and tweets: Consumer engagement behavior and movie box office in social media," *Information & Management*, vol. 54, no. 1, pp. 25–37, 2017.

[21] T. Kim, J. Hong, and P. Kang, "Box office forecasting using machine learning algorithms based on sns data," *International Journal of Forecasting*, vol. 31, no. 2, pp. 364–390, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of 2015 International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[25] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[26] A. Martín, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "Evolving deep neural networks architectures for android malware classification," in *Proceedings of 2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1659–1666.

[27] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[28] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 760–766.

[29] L.-O. Fedorovici, R.-E. Precup, F. Dragan, and C. Purcaru, "Evolutionary optimization-based training of convolutional neural networks for OCR applications," in *Proceedings of 2013 IEEE International Conference on System Theory, Control and Computing (ICSTCC)*, 2013, pp. 207–212.

[30] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," *Pattern Recognition*, vol. 59, pp. 199–212, 2016.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[33] D. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[35] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.

[36] T. K. Ho, "Random decision forests," in *Proceedings of 1995 IEEE International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2921–2929.