Image and Vision Computing 60 (2016) 58–63

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A novel companion objective function for regularization of deep convolutional neural networks☆

Weichen Sun[a], Fei Su[a,b,*]

[a]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[b]Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing, China

ARTICLE INFO

Article history:
Received 15 May 2016
Received in revised form 22 August 2016
Accepted 16 November 2016
Available online 24 November 2016

Keywords:
Convolutional neural network
Regularization
Companion objective function
Classification

ABSTRACT

Regularization is an essential technique discussed in an attempt to solve the overfitting problem in deep convolutional neural networks (CNNs). In this paper, we proposed a novel companion objective function as a regularization strategy to boost the classification performance in deep CNNs. Three aspects of this companion objective function are studied. Firstly, we proposed two kinds of auxiliary supervision for convolutional filters and non-linear activations respectively in the companion objective function. Both of them enhanced the performance by aleviating the overfitting problem and auxiliary supervision for non-linear activations provided more efficiency. Secondly, regularization of auxiliary supervision in the pre-training phrase is discussed. With the assistance of auxiliary supervision, CNNs could obtain a more favorable initialization for end-to-end supervised fine-tuning. Finally, this companion objective function is verified to be compatible with other regularization strategies such as dropout and data augmentation. Experimental results on benchmark datasets (CIFAR-10 and CIFAR-100) demonstrated advantages of our proposed companion objective function as a regularization approach.

1. Introduction

Convolutional Neural Networks(CNNs) have been successfully applied in practice and across a wide range of computer vision tasks, such as image classification [1–3], object detection [4–6], semantic segmentation [7,8], object tracking [9,10] and visual question answering [11,12]. The origin of CNNs [13] goes back to the 1986 and there has been a resurgence of interest in various neural networks named deep learning since 2006. Deep learning techniques are a class of machine learning techniques that model hierarchical abstractions in input data with the help of multiple hidden layers. Significant performance gain of deep learning is mainly due to the increased availability of labeled data and processing power. CNNs are biologically-inspired variants of neural networks and composed of alternating convolutional layers and pooling layers [14]. Taking into account the spatial structure of images, each convolutional layer supplies a particularly well-adapted architecture of local receptive fields and shared weights. Hence, neurons in each layer will only be connected to a small region of the lower layer, instead of all neurons in a fully-connected manner.

In the presence of large amount of labeled datasets, the capacity of CNNs could be improved easily by increasing their depths and widths, which implies growing parameters need to be tuned [2,3]. Due to their large number of parameters, it takes significantly longer to train CNNs while running the risk of overfitting. Overfitting is a major problem that occur during neural network training and it means the model starts to memorize training data rather than learning to generalize from trend, which yields worse generalization. A wide range of regularization techniques, such as data augmentation [15,16], unsupervised pre-training [17], early stopping [18], parameter norm penalities [19], bagging [20], dropout [21], stochastic pooling [22], multi-task learning [23,24], various objective functions [3,25,26] and so on, have been proposed to keep overfitting at bay and enhance the performance of CNNs from various angles. Data augmentation [15,16] is an effective strategy which augments training datasets by label-preserving image transformations such as scaling and rotation. The additional training data help the neural network to learn more invariant features and improve generalization to unseen data with noise. Unsupervised pre-training [17] is an unusual form of regularization by defining a favorable initialization point

☆ This paper has been recommended for acceptance by Jiwen Lu.

* Corresponding author at: School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China.
E-mail address: sufei@bupt.edu.cn (F. Su).

http://dx.doi.org/10.1016/j.imavis.2016.11.012
0262-8856/© 2016 Published by Elsevier B.V.

for traditional supervised training rather than supervised training straightly from randomly initialization point. Furthermore, unsupervised pre-training takes full advantages of unlabeled data which are always abundant and easily acquired. Early stopping [18] is a simple form of regularization applied to avoid overfitting when training in an iterative method, such as stochastic gradient descent. Parameter norm penalties [19] are regularization based on limiting the capacity of neural networks by adding parameter norm penalties, such as L1 or L2, to the objective function. As a result, the weight matrix appears some degree of sparsity with weights being updated iteratively. Bagging [20] is also a frequently regularization method to reduce generalization error by combining several models. Generalization error is reduced owing to errors from different uncorrelated models which are enforced by different assumptions, hyper-parameters or training strategies. Dropout training [21] is a variant of ensemble method which combines different network topologies by randomly dropping out nodes of the neural network during training. In this sense, each training example just influences one network topology in each iteration. All network topologies are sub-networks of a neural network and they share the same weights. Stochastic pooling [22] is a specified type of regularization for convolutional layers that enable the pooling layer after each convolutional layer a stochastic process rather than the deterministic operation in pooling layers such as average and max. Max pooling only captures the strongest activation of each pooling region. However, stochastic pooling could take non-maximal activations into account and thus represent multi-modal distributions of activations within each pooling region. Multi-task learning [23,24] profits from a regularization effect by leveraging supervised data from various tasks. In this way, generic parameters shared across all given tasks and task-specific parameters could be trained together with augmented training data. Despite deep learning has been shown to outperform traditional methods in numerous fields, in fact, more effective regularization strategies for deep learning play a major rule in improving generalization and avoiding overfitting.

In this work, a novel companion objective function as a regularization is proposed to facilitate the training of feature extractors, especially in lower layers. The key point of this kind of companion objective function is that we train several neural networks simultaneously with increasing depths and shared weights. Deeper neural networks are built based on shallower neural networks. It is similar with the dropout regularization but dropout training generates sub-networks with the same depth. An alternative view of this kind of companion objective function is that auxiliary supervised classifiers helps to address the vanishing gradient problem [27,28] during neural network training with the back-propagation algorithm. In addition, this kind of companion objective function could cooperate with other forms of regularization such as dropout, data augmentation and so forth. Experimental results on the classification benchmark dataset, CIFAR-10 [29], demonstrate the effectiveness of our proposed regularization strategy, which significantly outperform the results of the current state of the art technique. In order to prove that effects of our method are not just akin to a particular dataset, we have experimented on the dataset CIFAR-100 [29], results also indicate the promising performance of our proposed method.

2. Related work

There has been very limited achievements in the past for regularization methods from the angle of objective functions. Hinton et al. [30] showed that Restricted Boltzmann Machines (RBMs) [31] can be stacked to form so-called Deep Belief Networks (DBNs) [17] and introduced a greedy layer-wise unsupervised learning algorithm for DBNs. However, unsupervised greedy layer-wise training indicates inadequate. Bengio et al. [32] proposed a variant of greedy layer-wise training based on partial supervision and significant improvements support that the greedy supervised layer-wise training strategy helps to optimize deep networks. Better generalization is also obtained because this strategy initializes each layer with better representations of abstractions. It is equivalent to define several objective functions and each of them is optimized by the back-propagation algorithm in a sequence from bottom to top. Multiple objective functions optimized in a separated manner are applied as a regularization and help to optimize deep networks. But former optimizations are independent to the final optimization of the whole network, neglecting the truth that former optimizations are conducted to promote the final optimization.

Based on a hypothesis that a classifier trained on more discriminative features displays better performance than one trained on less discriminative features, Lee et al. [25] introduced a companion objective function which enforces direct and early supervision for both hidden layers and the output layer. This companion objective function acts as a kind of feature regularization, which simultaneously minimizes final classification error while enforcing hidden layers to learn more discriminative features. In addition, the exploding and vanishing gradients problems could be alleviated as there is a supervision for each hidden layer from the ground truth labels. The GoogLeNet [3] also introduces several auxiliary supervised classifiers which are selectively connected to the middle level convolutional layers and has won the first place in the Large Scale Visual Recognition Challenge 2015 (ILSVRC2015). This design encourages the lower and middle level convolutional features to be trained directly from the auxiliary supervision, avoiding gradient information vanished in very deep layers in back-propagation procedure.

The most popular activation function employed in CNNs mentioned above is rectified linear unit(ReLU) [33]. ReLU is a piecewise linear function which keeps positive inputs invariant and projects negative inputs to zeros. Consequently, activations of neural nodes are sparse and the gradient is less likely to vanish. However, ReLU has an undesirable property that back-propagation will be blocked once the unit is not activated and a vanishing error back flow has almost no effect on weight updates of lower layers. Authors in [34] introduced a so-called maxout activation function which assigns a non-zero slope to both positive part and negative part. Hence it facilitates the optimization procedure by partly preventing active hidden units from transiting to inactive. Weichen et al. [35] improved the fitting performance of the activation function by employing a trainable activation function called Multi-layer Maxout Network (MMN). Compared to fixed activations, MMN is a multi-layer variant of maxout and is capable of approximating an arbitrary function. On the other hand, MMN increases the number of trainable parameters and the depth of CNNs simultaneously, implying more risks of overfitting. Therefore, it is imperative to develop more efficient regularization techniques.

In this paper, we proposed a novel companion objective function as a regularization aiming to enhance the representation of high-level features by obtaining more discriminative low-level features. Auxiliary supervision is utilized on regularizing both convolutional filters and activation functions together with global supervision. To sum up, the proposed companion objective function has beneficial characteristics both for optimization and regularization. Moreover, it can be applied simultaneously in conjunction with other regularization strategies such as dropout and data augmentation.

3. Our proposed companion objective function

The architecture of our fully convolutional neural network with auxiliary supervision on MMNs is shown in Fig. 1. MMN is a micro neural network introduced as a trainable activation function and its architecture is shown in Fig. 2. We can employ SVM classifiers or

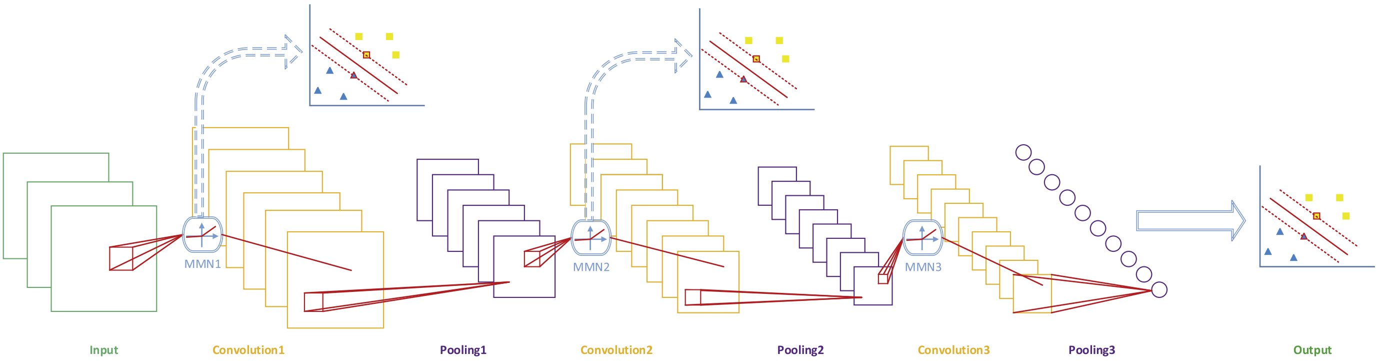

Fig. 1. Fully convolutional neural network with auxiliary supervision on MMNs.

softmax classifiers as the auxiliary supervision since the SVM is more local objective, which could be thought of either as a bug or a feature, while the softmax classifier is more global objective. In this section, we will discuss how auxiliary classifiers appended to our model helps parameters to be modified through training and how to use companion supervision as a regularization efficiently.

3.1. Definition

We define $S = \{(x, y) \in \{(x_i, y_i), i = 1, \ldots, N\}\}$ be a set of N samples, where $x_i \in R^n$ is the i-th raw input data, $y_i \in \{1, \ldots, K\}$ is the corresponding label, and K is the number of labels. The essence of a deep CNN is a composition of functions defined as:

$$F(x, W, w_{classifier}) = f_{classifier} \circ g_L \circ f_L \circ \cdots \circ g_l \circ f_l \circ \cdots \circ g_1 \circ f_1(x) \qquad (1)$$

where f_l is a linear transformation of each layer, $l \in [1, L]$, corresponding to the convolution operation on its input, and g_l is a non-linear activation function, corresponding to the MMN of each layer. Here we absorb the pooling operation into f_l because it has no parameters to be trained. The parameter W consists of two categories of trainable parameters, which are W_l for convolutional filters and $\hat{W}_l$ for non-linear activation functions of each layer respectively. Here, we absorb the bias term into the parameter W_l. Suppose x_{l-1} is the output of the $(l-1)$-th layer, the weighted sum transfer function f_l of the l-th layer is defined as:

$$f_l(x_{l-1}) = x_{l-1}^T W_l \qquad (2)$$

Then the output of the l-th layer x_l, is given by:

$$x_l = g_l(f_l(x_{l-1})) \qquad (3)$$

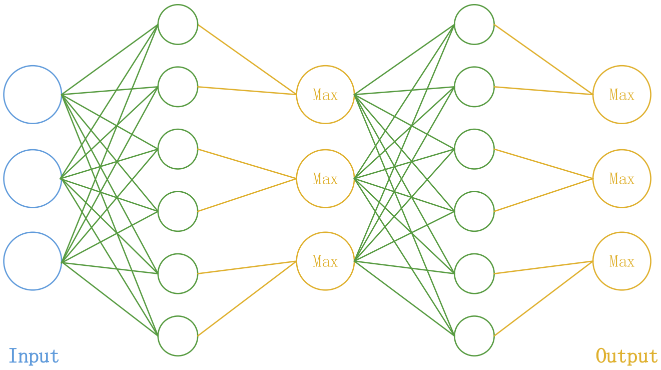

Fig. 2. Multi-layer Maxout Network (MMN).

As shown in Fig. 1, the function $f_{classifier}$ is a linear classifier added on the top of the model and $w_{classifier}$ is the normal vector to the hyperplane if we employ the SVM classifier.

3.2. Review of MMN

MMN is a trainable activation function with a multi-layer structure. Given an input $x \in R^n$ (x is either the raw input vector or the state vector of a hidden layer), the activation $g(x)$ of a hidden unit is calculated as follows:

$$h_j(x) = \max_{i \in \{1,2\}} x^T W_i^{(1)} \qquad (4)$$

$$g(x) = \max_{j \in \{1,2\}} h_j(x)^T W_j^{(2)} \qquad (5)$$

Here, the MMN a two-layer network and $h_j(x)$ is a hidden unit performing the max operation over two functions.

MMN is a powerful trainable approximator and could be trained to approximate any complex activation function well by increasing the parameter k_m and the depth, even the potential activation function is non-convex such as a w-shaped curve.

3.3. Companion objective function

We define the weight vector W for both convolutional filters and MMNs as:

$$W = \left(W_1, \hat{W}_1, W_2, \hat{W}_2, \cdots, W_L, \hat{W}_L\right) \qquad (6)$$

Supposing that SVM classifiers are applied to provide auxiliary supervision, all auxiliary SVM classifiers can be divided into two parts with their parameters w for linear weighted sum functions and $\hat{w}$ for trainable non-linear activation functions respectively:

$$w = (w_1, w_2, \cdots, w_{L-1}) \qquad (7)$$

$$\hat{w} = (\hat{w}_1, \hat{w}_2, \cdots, \hat{w}_{L-1}) \qquad (8)$$

As a result, the companion objective function of this model is defined as a weighted sum of various losses:

$$L(W, w_{classifier}) + \|w_{classifier}\|^2 + \alpha C_1 + \beta C_2 \qquad (9)$$

where

$$C_1 = \sum_{l=1}^{L-1} \left[l(W, w_l) + \|w_l\|^2 - \gamma \right]_+ \qquad (10)$$

$$C_2 = \sum_{l=1}^{L-1} \left[\hat{l}(W, \hat{w}_l) + \|\hat{w}_l\|^2 - \delta \right]_+ \tag{11}$$

The overall loss is

$$L(W, w_{classifier}) = \sum_{f_{classifier} \neq y_i} [1 - < w_{classifier}, \varphi(x_i, f_{classifier}) - \varphi(x_i, y_i) >]_+^2 \tag{12}$$

and the companion losses are

$$l(W, w_l) = \sum_{f^l_{classifier} \neq y_i} \left[1 - < w_l, \varphi(x_i, f^l_{classifier}) - \varphi(x_i, y_i) > \right]_+^2 \tag{13}$$

$$\hat{l}(W, \hat{w}_l) = \sum_{\hat{f}^l_{classifier} \neq y_i} \left[1 - < \hat{w}_l, \varphi(x_i, \hat{f}^l_{classifier}) - \varphi(x_i, y_i) > \right]_+^2 \tag{14}$$

We name C_1 and C_2 as companion losses for convolutional layers and MMN activation functions respectively. α and β are hyper-parameters that balance the influence of losses in the companion objective function. Eqs. (10) and (11) denote squared hinge losses of two different types mentioned above, in which $f^l_{classifier}$ and $\hat{f}^l_{classifier}$ are the outputs of auxiliary SVM classifiers at the l-th layers, γ and δ are thresholds controlling the influence of different companion losses. Once the companion loss is less than the threshold during training, it will be not included in the calculations of the overall objective function. When all the companion losses are omitted, the objective function equals to the regular objective function of end-to-end supervised learning. In Eq. (12), $L(W, w_{classifier})$ and $\|w_{classifier}\|^2$ are the squared hinge loss and the margin of the SVM classifier at the top of the deep CNN. Here φ is the joint feature function and we omit the influence of balance parameters between the squared hinge loss and the margin for simplicity. In Eq. (13), $l(W, w_l)$ and $\|w_l\|^2$ are the squared hinge loss and the margin respectively of the SVM classifier added to the l-th hidden convolutional layer of the deep CNN. In Eq. (14), $\hat{l}(W, \hat{w}_l)$ and $\|\hat{w}_l\|^2$ are the squared hinge loss and the margin of the SVM classifier, which are added to the trainable non-linear activation function of the l-th hidden layer.

There are two kinds of auxiliary losses to be considered in our proposed objective function Eq. (9). The contributions are as follows: (1) The proposed companion objective function helps to address the vanishing gradient problem when modifying both parameters for linear transformations and parameters for nonlinear activation functions. Layer-wise auxiliary losses act as a compensation for the lower layers when few gradient information is back propagated through higher layers and could update parameters of each layer more efficiently, especially for those lower layers. (2) Both kinds of companion losses act as regularization, which prompt to obtain more discriminative low-level features enhancing the representation of high-level features. Hence more discriminative features could be obtained by training convolutional filters and non-linear activation functions jointly, particularly non-linear activation functions, which plays an important role in making the deep architecture reasonable. (3) The companion supervision could achieve a good initialization for conventional fine-tuning and lead to a performance enhancement without losing much information. (4) Our objective function can also be treated as a trade-off between greedily layer-wise unsupervised pre-training and standard end-to-end supervised learning and overcome weaknesses of them.

4. Experiments

To verify the performance of our proposed companion objective function, we evaluate it on the image classification benchmark datasets CIFAR-10 and CIFAR-100. Experimental results are compared with those methods proposed in literatures [25,34,36,37] as well as the current state of the art. Furthermore, we demonstrate that the companion objective function plays a role as a regularization in deep models with extensive experiments. Our experiments are based on Caffe [38], which is a popular deep learning framework.

4.1. Experimental setup

In experiments on CIFAR-10 and CIFAR-100, we introduce four models with different objective functions to reflect influence of auxiliary supervision added to different components in a deep CNN. In order to control the influence of auxiliary supervision to the final classification, both α and β in Eq. (9) equal 0.3, which is selected by a grid search over 0,0.1,…,0.5 on a validation set. We found that these hyper-parameters selections are crucial and either too large or too little auxiliary supervision would result to the performance decreasing.

Model A:

$$L(W, w_{classifier}) + \|w_{classifier}\|^2 \tag{15}$$

Model B:

$$L(W, w_{classifier}) + \|w_{classifier}\|^2 + 0.3 * C_1 \tag{16}$$

Model C:

$$L(W, w_{classifier}) + \|w_{classifier}\|^2 + 0.3 * C_2 \tag{17}$$

Model D:

$$L(W, w_{classifier}) + \|w_{classifier}\|^2 + 0.3 * C_1 + 0.3 * C_2 \tag{18}$$

Our CNN is similar to the model mentioned in [25], consisting of three convolutional layers and each of which is followed with a MMN as its trainable non-linear activation function. We also dropped the full-connected layers as introduced in [39] and employed a global averaged pooling layer to produce the output features. Before training, we preprocessed training images by the global contrast normalization (GCN). The training process was divided into four steps: Firstly, the model was pre-trained with a companion objective function for 50,000 iterations. Secondly, the model was fine-tuned with the simple objective function without any companion losses for 100,000 iterations. In both steps above, we used a fixed value of learning rate equal to 0.025. Then we continued fine-tuning the model for 20,000 iteration with the learning rate decreased by multiplying by 0.025. Finally, we set the value of learning rate to 0.0001 and fine-tuned the model for more 30,000 iterations. In optimization, we incorporated stochastic gradient descent (SGD) with a mini-batch of 128 instances. In order to avoid the over-fitting problem, we also adopted the dropout method to the input image as well as after first two pooling layers. The dropout rates were 20% for dropping out inputs and 50% otherwise.

4.2. CIFAR-10

The CIFAR-10 dataset consists of 32 × 32 color images in 10 classes, with 6000 images per class. There are 50,000 images for training and 10,000 images for testing. We selected 1000 images per class randomly from training images as the validation dataset in order to identify the hyper-parameters such as learning rates, the weight decay and iterations of supervised training with companion losses. A comparison of our model with previous methods is shown in Table 1.

Table 1
Comparisons on CIFAR-10.

Method	Test error (%)
Maxout [34]	11.68
Network In Network [39]	10.41
DSN [25]	9.78
ALL-CNN [40]	9.08
Model A	9.13
Model B	8.84
Model C	**8.46**
Model D	8.51

Table 3
Comparisons on CIFAR-100.

Method	Test error (%)
Maxout [34]	38.57
Tree based Priors [41]	36.85
Network In Network [39]	35.68
DSN [25]	34.57
Model A	33.46
Model B	31.94
Model C	32.90
Model D	**31.93**

The experimental results show that Model C with the companion objective function Eq. (17) outperformed others, even Model D which has more parameters to tune. It is obvious that MMN as a trainable activation function plays a more important role than others in feature extraction. In order to train DNNs efficiently and save memory, regularization in form of auxiliary supervision should be supplied to trainable activation function rather than other layers.

4.3. CIFAR-10 with data augmentation

In this part, we augmented the training set by randomly cropping 24 × 24 patches from the 32 × 32 image. Then our models are trained on random 24 × 24 patches and tested on the 24 × 24 patch cropped in the center of each 32 × 32 image. A performance comparison illustrated in Table 2 demonstrates our regularization strategy is effective with data augmentation in the classification task as well.

From the above results, we found that the performance improvement of our model with data augmentation is not as much as that listed in Table 1 (without data augmentation). The companion supervised information seems to be not so indispensable, which is reasonable. Because the motivation of the companion objective is to compensate the gradients for lower layers, they can learn more discriminative features for high layers. Once there are plenty of training data sets, the lower layer could also learn discriminative features without companion supervision.

4.4. CIFAR-100

The CIFAR-100 dataset is similar to the CIFAR-10 dataset, except it has 100 classes containing 600 images each [29]. There are 500 images for training, 100 images for testing per class. For CIFAR-100, we applied the same model and hyper-parameters as in CIFAR-10, except the last MMN layer outputs 100 feature maps instead of 10 feature maps corresponding to 100 different classes. The dropout layer of 20% are removed since the CIFAR-100 dataset is more challenging and the generalization benefits of randomly drop 20% of the input has diminishing returns. The experimental results demonstrate advantages of our proposed companion objective function and more challenging tasks require more auxiliary supervision. A summary in Table 3 shows the performance comparison of different models on the CIFAR-100 database.

Table 2
Comparisons on CIFAR-10 with data augmentation.

Method	Test error (%)
Maxout [34]	9.38
Probout [36]	9.39
DropConnect [37]	9.32
Network In Network [39]	8.81
DSN [25]	8.22
Model A	7.63
Model B	7.58
Model C	**7.49**
Model D	7.66

4.5. Regularization view

As we analyzed, both two kinds of auxiliary losses could be treated as regularization for features. We designed extensive experiments on CIFAR-10 to show their efficiency as regularization strategies. In order to reduce the interference of other regularization such as dropout and data augmentation. We refused the dropout method to the input image and ran experiments on CIFAR-10 without data augmentation. The compared four models were trained 50,000 iterations first (pre-training phase) and then fine-tuned by end-to-end supervised learning without any auxiliary losses till convergence (fine-tuning phrase). As shown in Fig. 3, at the beginning, the model without companion losses (Model A) is better than models trained with companion losses (Model B, Model C and Model D). All four models converged with fine-tuning. Experimental results are shown in Table 4, which proved that the companion objective function would not enhance the performance of models directly, but play a role of regularization to locate a good initialization of the network for latter supervised fine-tuning. What is more, the comparison among three different companion losses indicates that Model C is a more efficient regularization than others for pre-training our model. Moreover, the performance of deep neural networks can be greatly improved with a good initialization of weights prior to back-propagation.

5. Conclusion

In this paper, a new companion objective function for deep convolutional networks is proposed, as a regularization to extract more discriminative features by optimizing the objective function for both convolutional filters and non-linear activation functions. What is more, auxiliary supervision to various layers has been evaluated and auxiliary supervision for non-linear activation functions is more efficient. In addition, this companion object function helps to address the vanishing gradient problem in the companion supervised training procedure. Furthermore, it contributes to a good initialization for

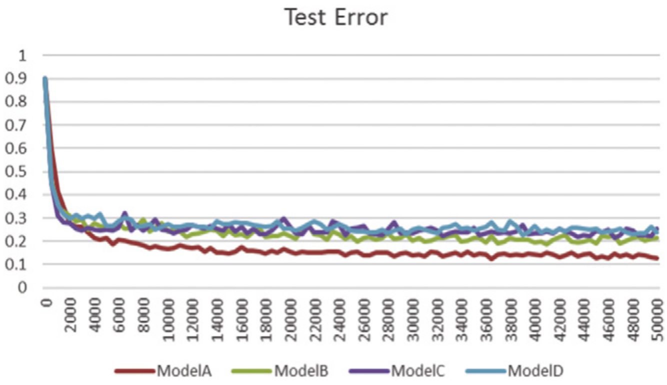

Fig. 3. Test error during 50,000 iterations.

Table 4
Regularization comparisons.

Model	Test error (%)	
	Pre-training	Fine-tuning
Model A	12.55	9.42
Model B	21.54	9.21
Model C	25.56	8.81
Model D	23.79	8.97

fine-tuning in deep neuron networks. In this framework, more discriminative features could be obtained. The proposed model takes both advantages of layer-wise supervised training and end-to-end supervised training. Experimental results show good performance of our model comparing to some state-of-the-art techniques.

Acknowledgments

This work is supported by Chinese National Natural Science Foundation (61372169, 61532018, 61471049), and Special Funds of Beijing Municipal Co-construction Project 2016.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012. pp. 1097–1105.
[2] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. 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, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 1–9.
[4] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 580–587.
[5] R. Girshick, Fast r-cnn, Proceedings of the IEEE International Conference on Computer Vision, 2015. pp. 1440–1448.
[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems, 2015. pp. 91–99.
[7] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 3431–3440.
[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic Image Segmentation With Deep Convolutional Nets and Fully Connected crfs, 2014. arXiv preprint arXiv:1412.7062
[9] N. Wang, D.-Y. Yeung, Learning a deep compact image representation for visual tracking, Advances in Neural Information Processing Systems, 2013. pp. 809–817.
[10] L. Wang, W. Ouyang, X. Wang, H. Lu, Visual tracking with fully convolutional networks, Proceedings of the IEEE International Conference on Computer Vision, 2015. pp. 3119–3127.
[11] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, D. Parikh, VQA: visual question answering, Proceedings of the IEEE International Conference on Computer Vision, 2015. pp. 2425–2433.
[12] M. Malinowski, M. Rohrbach, M. Fritz, Ask your neurons: a neural-based approach to answering questions about images, Proceedings of the IEEE International Conference on Computer Vision, 2015. pp. 1–9.
[13] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, Tech. rep., DTIC Document, 1985.
[14] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network, Neural Netw. 16 (5) (2003) 555–559.
[15] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012. pp. 3642–3649.
[16] H. Naceur, A. Delameziere, J. Batoz, Y. Guo, C. Knopf-Lenoir, Some improvements on the optimum process design in deep drawing using the inverse approach, J. Mater. Process. Technol. 146 (2) (2004) 250–262.
[17] G. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[18] Y. Bengio, Learning deep architectures for AI, Found. trends® Mach. Learn. 2 (1) (2009) 1–127.
[19] G. Hinton, A practical guide to training restricted Boltzmann machines, Momentum 9 (1) (2010) 926.
[20] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[21] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors, 2012. arXiv preprint arXiv:1207.0580
[22] M.D. Zeiler, R. Fergus, Stochastic Pooling for Regularization of Deep Convolutional Neural Networks, 2013. arXiv preprint arXiv:1301.3557
[23] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.
[24] T. Evgeniou, M. Pontil, Regularized multi-task learning, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004. pp. 109–117.
[25] C. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-Supervised Nets, 2014. arXiv preprint arXiv:1409.5185
[26] W. Sun, F. Su, Regularization of deep neural networks using a novel companion objective function, Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015. pp. 2865–2869.
[27] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, Neural Netw. IEEE Trans. 5 (2) (1994) 157–166.
[28] R. Pascanu, T. Mikolov, Y. Bengio, Understanding the exploding gradient problem, 2012, arXiv preprint axXiv:1211.5063.
[29] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features From Tiny Images, Computer Science Department, University of Toronto, Tech. Rep, 2009.
[30] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[31] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, Proceedings of the 24th International Conference on Machine Learning, ACM, 2007. pp. 791–798.
[32] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Proces. Syst. 19 (2007) 153.
[33] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010. pp. 807–814.
[34] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A.C. Courville, Y. Bengio, Maxout networks, ICML 28 (3) (2013) 1319–1327.
[35] W. Sun, F. Su, L. Wang, Improving deep neural networks with multilayer maxout networks, Visual Communications and Image Processing Conference, 2014 IEEE, IEEE, 2014. pp. 334–337.
[36] J. Springenberg, M. Riedmiller, Improving Deep Neural Networks with Probabilistic Maxout Units, 2013. arXiv preprint arXiv:1312.6116
[37] L. Wan, M. Zeiler, S. Zhang, Y. Cun, R. Fergus, Regularization of neural networks using dropconnect, Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013. pp. 1058–1066.
[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, Proceedings of the ACM International Conference on Multimedia, ACM, 2014. pp. 675–678.
[39] M. Lin, Q. Chen, S. Yan, Network in Network, 2014. arXiv preprint arXiv:1312.4400
[40] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for Simplicity: The All Convolutional Net, 2014. arXiv preprint arXiv:1412.6806
[41] N. Srivastava, R.R. Salakhutdinov, Discriminative transfer learning with tree-based priors, Advances in Neural Information Processing Systems, 2013. pp. 2094–2102.