Transfer Learning Method for Convolutional Neural Network in Automatic Modulation Classification

Yu Xu[1], Dezhi Li[1], Zhenyong Wang[1,2(✉)], Gongliang Liu[1], and Haibo Lv[1]

[1] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, Heilongjiang, China
{xu_yu, lidezhi, ZYWang, liugl, elitelv}@hit.edu.cn
[2] Shenzhen Academy of Aerospace Technology, Shenzhen, Guangdong, China

Abstract. Automatic modulation classification (AMC) plays an important role in many fields to identify the modulation type of signals, in which the deep learning methods have shown attractive potential development. In our research, we introduce convolutional neural network (CNN) to recognize the modulation of the input signal. We used real signal data generated by instruments as dataset for training and testing. Based on analysis of the unstable training problem of CNN for weak signals recognition with low SNR, a transfer learning method is proposed. Experiments results show that the proposed transfer learning method can locate better initial values for CNN training and converge to a good result. According to the recognition accuracy performance analysis, The CNN with the proposed transfer learning method has higher average classification accuracy and is more compatible for unstable training problem.

Keywords: Modulation classification · Convolutional neural network
Transfer learning method

1 Introduction

Automatic modulation classification is aiming to detect the modulation type of received signals in order to recover signals by demodulation. The dominant approach of signal modulation recognition can be categorized as likelihood-based (LB) methods and feature-based (FB) methods [1]. Many neural networks have been applied to the automatic modulation classification, which have shown better performances than the traditional LB methods and FB methods [2–6]. In the former research on a deep learning architecture based on convolutional neural network for modulation classification of wireless signals, it is found that the same programs get different results under the same experiment conditions. Even sometime the result is terrible. The phenomenon is defined as Unstable Training Problem in our paper, which comes out in training process with the random seed. Because the same initializer is involved at the beginning of training process, it is important to focus on the robustness of modulation classification based on CNN against noise in training process.

X. Gu et al. (Eds.): MLICOM 2017, Part II, LNICST 227, pp. 371–380, 2018.
https://doi.org/10.1007/978-3-319-73447-7_41

In former experiments under conditions of various SNR, the modulation classification method based on CNN shows fairly good recognition performances for SNR ≥ 0 dB wireless signals, in which the unstable training problem is rare. As the SNR of wireless signals drops, the unstable training problem is becoming obvious and it is hard to carry on training process of the CNN. The reason of the unstable training problem is regarded as infeasible robustness of the neural network to noise-like dataset [7]. For the Unstable Training Problem, if the SNR is very low, the features of modulation are influenced by noise and are hard to be extracted. The modulation classification method based on CNN will encounter numerical difficulties and we can't get a satisfied performance, because the signal dataset cannot provide a good estimation of gradient to training.

Transfer learning method refers to the situation where some learned knowledge is transferred to improve generalization in another setting. The objective is to inherit extracted information from advantage of data in the first setting, so as to help learning process or even making predictions directly in the second setting. The motivation is that the same representation may be useful in both settings.

The main idea of this paper is to investigate transfer learning to solve the unstable training problem in training process of a stacked convolutional neural network of deep learning architecture for modulation classification of wireless signals automatically. The rest of the paper is organized as follows. In Sect. 2, based on analysis of Stochastic Gradient Descent and initialization problem of training process of CNN, the transfer learning is investigated. Based on real sampled data of wireless signals, an improved CNN architecture is trained and proposed with transfer learning in Sect. 3. In Sect. 4, according to the experiment results, the effect of transfer learning to solve unstable training problem is analyzed in term of recognition accuracy in various SNR conditions. Finally, conclusions are drawn in Sect. 5.

2 Training Process Based on Transfer Learning

2.1 Stochastic Gradient Descent

Nearly all of deep learning is powered by important Stochastic Gradient Descent (SGD) algorithm. Stochastic Gradient Descent is typical and preferred to training process for neural networks. One row of data is inputted into the network at a time. The network activates neurons forward to produce an output value finally. Then the output value is compared to the expected output value to generate an error value. The error is backward propagated through the network, in which the weights of layer are updated one after another, according to the contributed amount to the error. The process is repeated for all of the examples in the training data to get a trained network of the intended goal.

The weights in the network can be updated from the calculated errors for each training example, which can result in fast but also chaotic changes to the network. On the other hand, the errors can be saved up across all of the training examples and the network can be updated at the end.

The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function. For example, the negative conditional log-likelihood of the training data can be written as

$$J(\theta) = E_{x,y\sim p_{data}} L(x, y, \theta) = \frac{1}{m}\sum_{i=1}^{m} L(x^{(i)}, y^{(i)}, \theta) \tag{1}$$

where L is the per-example loss:

$$L(x, y, \theta) = -\log p(y \mid x; \theta) \tag{2}$$

For these additive cost functions, gradient descent requires computing

$$\nabla_\theta J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta L(x^{(i)}, y^{(i)}, \theta) \tag{3}$$

Considering the computational cost, we use a small set of samples to approximately estimate the true gradient. Specifically, on each step of the algorithm, we can sample a minibatch of examples $B = \{x^{(1)}, \ldots, x^{(m\prime)}\}$ drawn uniformly from the training set. The minibatch size m' is typically chosen to be a relatively small number of examples, ranging from 1 to a few hundred.

Based on examples from the mini-batch B, The estimate of the gradient is formed as

$$\mathrm{g} = \frac{1}{m'}\nabla_\theta \sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta) \tag{4}$$

The stochastic gradient descent follows the estimated gradient downhill:

$$\theta \leftarrow \theta - \varepsilon g \tag{5}$$

where ε is the learning rate.

2.2 Initialization Problem of Training Process

Training algorithms for deep learning models are usually iterative in nature and thus require us to specify some initial point from which to begin the iterations. However, training deep models is a sufficiently difficult task that most algorithms strongly affected by the choice of initialization. The initial point can determine whether the algorithm converges with some initial points begin so unstable that the algorithm encounters numerical difficulties and fails. When learning does converge, the initial point can determine how quickly learning converges and whether it converges to a point with high or low cost.

Different task have different solution space, but they are similar in some extents. Neural network training is non-deterministic, and converges to a different function every time it is run. As can be seen from the Fig. 1, the training process may encounter

many numerical difficulties like local maximum or shoulder location. The state when we stop training not only depends on the optimization algorithm. The initial values where we start iteration also have a great impact on the result. A bad initial value can result in numerical difficulties.

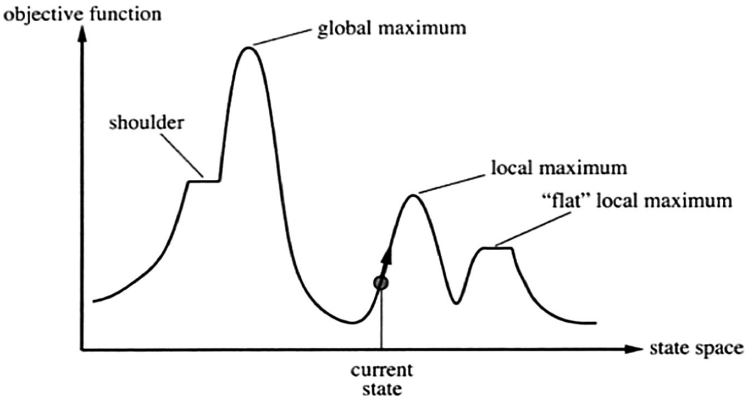

Fig. 1. One-dimensional solution space state illustration

Moreover, when training our neural network, it is possible that training initializes the model in a location which is surrounded by areas where the cost function varies so much from one sample point to another that mini-batches give only a very noisy estimate of the gradient, or region surrounded by areas where the Hessian matrix is so poorly conditioned that gradient descent methods must use very small steps. It slow down the training by requiring lower learning rates and careful parameter initialization, and makes it hard to train models.

Another related method is greedy layer-wise unsupervised pre-training. This greedy learning procedure could be used to find a good initialization for a joint learning procedure over all the layers [8]. In the context of a supervised learning task, it can be viewed as regularization and a form of parameter initialization.

Unsupervised pre-training takes the parameters into a region that would otherwise be inaccessible and it brings much success. Neural networks that receive unsupervised pre-training consistently halt in the same region of function space, while neural networks without pre-training consistently halt in another region. The region where pre-trained networks arrive is smaller, suggesting that pre-training reduces the variance of the estimation process, which can in turn reduce the risk of severe over-fitting [9].

2.3 Transfer Learning Method

Transfer learning refers to the situation where what has been learned in one setting P_1 is exploited to improve generalization in another setting P_2. In transfer learning, the learner must perform two or more different tasks, but we assume that many of the factors that explain the variations in P_1 are relevant to the variations that need to be

captured for learning P_2. Using the same representation in both settings allows the representation to benefit from the training data that is available for both tasks.

A brief example of transfer learning is show in Fig. 2. In most case, one task is easy to fulfill while another is hard due to the low quality of dataset. We can use transfer learning through sharing weight between the two learning process. Sometimes what is shared among the different tasks is not the input but the output. In cases like these, it makes more sense to share the upper layers of the neural network, as illustrated in Fig. 3. The transfer learning may help to learn representations that are useful to quickly generalize.

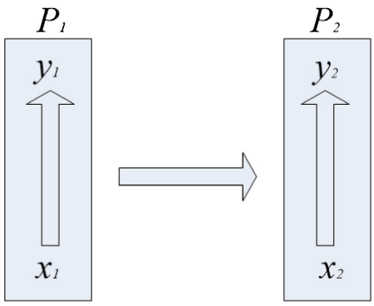

Fig. 2. An example of transfer learning

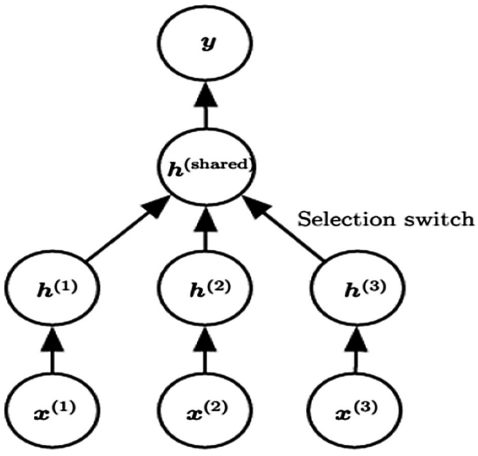

Fig. 3. An example of transfer learning sharing the output

3 The CNN for Modulation Classification

To meet the requirements of modulation classification, our network architectures are mainly inspired by ALEXNET [10], as shown in Fig. 4.

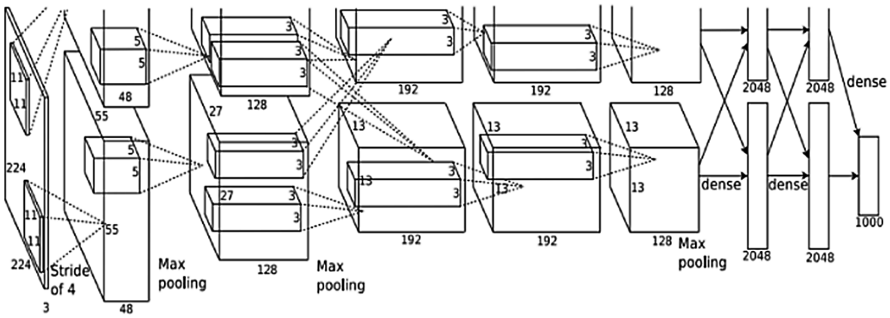

Fig. 4. The architecture of ALEXNET

3.1 Signals Data Sampled and Process

Because digital modulation has better immunity performances to interference, which is mostly discussed in the literatures for modulation classification. Here, it is assumed that there is a single carrier-transmitted signal in additive white Gaussian noise (AWGN) channel. The modulation types include 2ASK, BPSK, QPSK, 8PSK and 16QAM.

The signal data are produced by vector signal generator SMU200A. The sampling rate is 1 GHz. All the signal data of different modulation types have the same carrier frequency of 100 MHz and bandwidth of 25 MHz. Every sample has 2000 raw points and there are 25000 samples in total, 5000 samples for each modulation type. The only preprocess is to rescale the amplitude to the range of −2 V to 2 V. The spectrum map of sampled BPSK signal is shown in the Fig. 5.

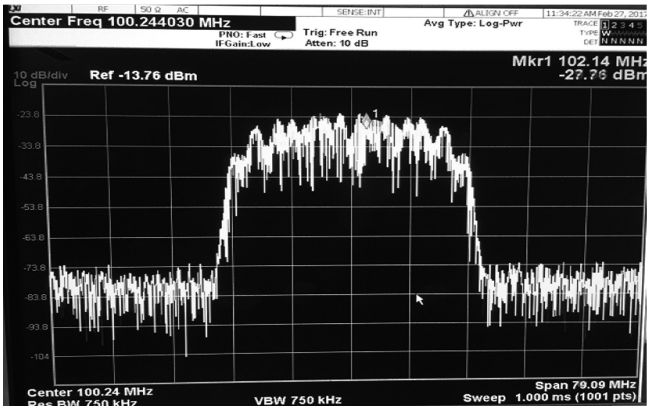

Fig. 5. The spectrum map of BPSK signal

For most classification and regression process, there is still possibility to get results even with small random noise added to the input. However, neural networks are proved not robust to noise [7]. One way to improve the robustness of neural networks is simply

to do training process with input random noise data. So in training procedure to improve the robustness, training data of same SNR are included, which are also used to test the performance of proposed method in different SNR conditions.

When the network layers are not deep, it is not likely to encounter the problems like vanishing/exploding gradients. The principle of maximum likelihood is taken as the cost function, which means the cross-entropy between the training data and the prediction of the model is regarded as the cost function. The weights are initialized with Gaussian distribution initializers, which have zero means and unit variances. The SGD is involved with a mini-batch size of 256. The weight decay is 0.0001 and the momentum is 0.9. The learning rate starts from 0.1. When there are errors plateaus occur, the learning rate descends at rate of 10 times.

As for the testing process, it is typically to use a simple separation of the same sampled data into training and testing datasets. In experiments, 80% data of the sampled signal is assigned to training dataset and 20% data of the sampled signal is assigned to testing dataset. Finally when the training is halt, we get the accuracy through inputting the testing datasets and statistical the accuracy.

3.2 The CNN Architecture

It is found that the removal of the fully-connected layers of ALEXNET will reduce the amount of weight parameters and get little impact of the recognition accuracy performance. In this paper, the large kernel size is designed for better performances and acceptable complexity. Moreover, after investigating the deep neural network with more than 30 layers, it is found there are over-fitting problems. It is possible to apply a shallow neural network to compete modulation recognition for signals with reasonable SNR.

Based on the analysis above, the number of input neuron is set to 2000, which means every sample has 2000 raw points. The CNN is proposed with 3 convolutional layers, and each convolutional layer is followed by a max pooling layer. At the end of the CNN network, a 5-way fully-connected layer with softmax is used to output the probability of 5 kinds of signal modulations classification. The convolutional layers have filter kernels with length of 40. 64 filter kernels are used in both the input layer and the second layer. For the third layer, the filter kernels are increased to length of 128. The max pooling layers perform down-sampling with stride of 2 and pool width of 3 to get overlapping pooling. We do not use the any regularization like dropout. So, the CNN consists of 4 weighted layers, as shown in Fig. 6.

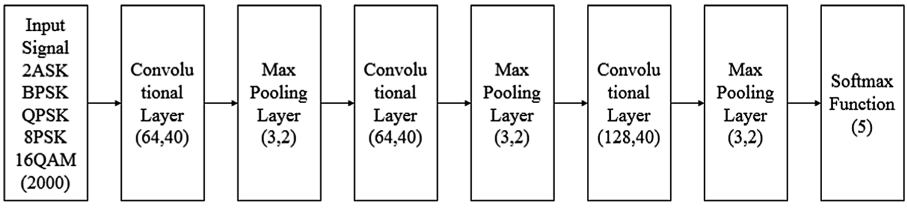

Fig. 6. The CNN Structure

4 Experiments and Results Analysis

We initialize the weights with the same initializers to explore the performance of the proposed neural network architecture in different SNR conditions and encounter the unstable training problem. We take the experiment with the SNR ranks −1 to −3 as an example to describe this phenomenon. The result of three times in different SNR conditions is list in Table 1, we report 'failed' when the accuracy is lower than 40%.

Table 1. The accuracy comparison under various SNR conditions

SNR	The 1st experiment	The 2nd experiment	The 3rd experiment
−1 dB	99.86%	99.44%	99.50%
−2 dB	failed	97.31%	92.10%
−3 dB	92.08%	failed	94.45%

As can be seen in Table 1, when the SNR is high, the experiment works well and the proposed CNN gets satisfied recognize accuracy at approximate 100%. We contribute the reason for this good result to the simple architecture of shallow network. The solution space is not very complicate. The dataset can provide a good estimation of gradient to update the parameter so the initial point does not matter. An approximate global maximum point can be achieved. As the drop of the SNR, the model encounters the problem which we called unstable training. In this case, we think that the mini-batches give only a very noisy estimate of the gradient and it easy to fall into a local minimum point which blocks the training. The algorithm cannot calculate a right direction to move which contribute to a failure. The results fluctuated severely and a method based on transfer learning is proposed in the following.

We apply the weights trained in the high SNR condition as the initial weights when we training the model in the low SNR conditions, which is a kind of transfer learning. We got the convergent point in high SNR condition and the gradient become small. There are two tasks which have a same target to recognize the modulation type of the signal but have the different input. To some extent one task has a more clear input while the input of another task can be seen as a more polluted signal suffer from AWGN channel. Our experiment results show that the unstable training problem is well addressed in this setting. Besides, we also get a lift in the performance of accuracy.

We evaluate our method based on transfer learning by using the pretraining initial weights and the results is show in Fig. 7. The line 1 is the performance of the original method using Gaussian distribution random initializers without pretraining. Due to the unstable training problem, we run the program 5 times and choose the best result as the final result. We first trains the model in 0 dB and it exists no difficult to get a satisfied recognize accuracy approximate 100%. We save the weights as the initial weights when we experiment the performance in other low SNR conditions and the result is show in line 2. It is obvious that we obtain the accuracy gains from this method. We argue that the initial weights bring the network the simple capacity to recognize the true modulation type of the signal data and that is otherwise hard to access. We use the weights based on iterative SNR to explore the performance of the neural network in

line 3. For example, we initial the weights trained in the SNR of −1 dB when we train the model in the SNR of −2 dB and recursively. Somehow surprisingly, the results are improved by healthy margins and report in Fig. 7. In our opinion, the initial weights can not only have the ability to recognize the modulation type but also extract features which is covered by noise simply. So this method can lift the performance of the proposed neural network architecture to recognize the modulation type of signal a lot.

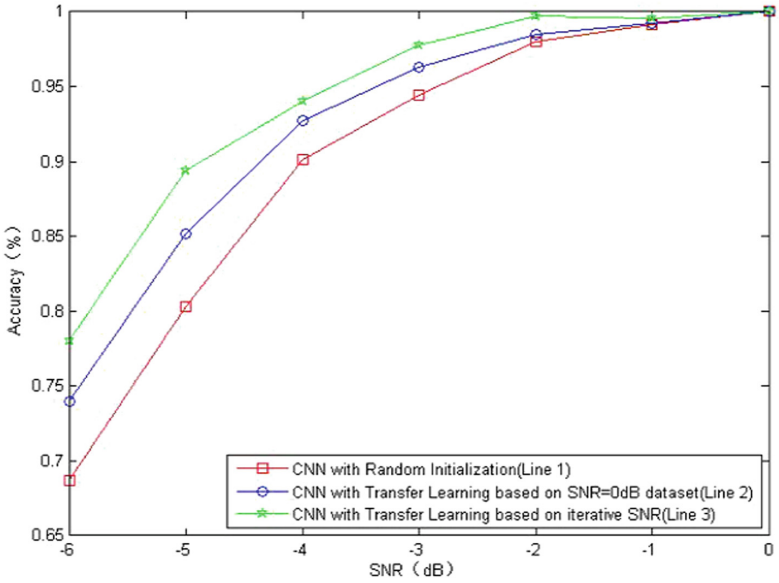

Fig. 7. Recognition accuracy comparison

5 Conclusion

The aim of our experiment is to design a method to fulfill the task of recognizing the modulation of input signal. We try to use stacked convolutional neural network to recognize the modulation of the input signal and we also get a good performance. We found that it is easily to encounter numerical difficulties when the SNR is low which is called unstable training is this paper. This motivates us to propose a transfer learning method to solve this problem. The unstable training problem is well addressed in this setting and we manage to obtain accuracy gains from this method.

Acknowledgement. This work was supported by National Natural Science Foundation of China. (No. 61601147, No. 61571316, No. 61371100) and the Fundamental Research Funds for the Central Universities (Grant No. HIT. MKSTISP. 2016013).

References

1. Hazza, A., Shoaib, M., Alshebeili, S.A., Fahad, A.: An overview of feature-based methods for digital modulation classification. In: 2013 1st International Conference on Communications, Signal Processing and their Applications, pp. 1–6. IEEE Press, New York (2013)
2. Zhu, X., Fujii, T.: A modulation classification method in cognitive radios system using stacked denoising sparse autoencoder. In: 2017 IEEE Radio and Wireless Symposium, pp. 218–220. IEEE Press, New York (2017)
3. Dai, A., Zhang, H., Sun, H.: Automatic modulation classification using stacked sparse auto-encoders. In: 13th IEEE International Conference on Signal Processing, pp. 248–252. IEEE Press, New York (2017)
4. Mendis, G.J., Wei, J., Madanayake, A.: Deep learning-based automated modulation classification for cognitive radio. In: 2016 IEEE International Conference on Communication Systems, pp. 1–6. IEEE Press, New York (2016)
5. O'Shea, T.J., Corgan, J., Clancy, T.C.: Convolutional radio modulation recognition networks. In: Jayne, C., Iliadis, L. (eds.) EANN 2016. CCIS, vol. 629, pp. 213–226. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44188-7_16
6. Fu, J., Zhao, C., Li, B., Peng, X.: Deep learning based digital signal modulation recognition. In: Mu, J., Liang, Q., Wang, W., Zhang, B., Pi, Y. (eds.) The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems. LNEE, vol. 322, pp. 955–964. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-08991-1_100
7. Tang, Y., Eliasmith, C.: Deep networks for robust visual recognition. In: 27th International Conference on Machine Learning, pp. 1055–1062. ACM, New York (2010)
8. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006). IEEE Press, New York
9. Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**(3), 625–660 (2010). ACM, New York
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: 2012 26th Annual Conference on Neural Information Processing Systems, vol. 25, no. 2, pp. 1097–1105. ACM, New York (2012)