electronics

MDPI

Article

Improving Heterogeneous Network Knowledge Transfer Based on the Principle of Generative Adversarial

Feifei Lei [1], Jieren Cheng [1,2,*], Yue Yang [1,2], Xiangyan Tang [1,2,3], Victor S. Sheng [4] and Chunzao Huang [1]

[1] School of Compute Science and Cyberspace Security, Hainan University, Haikou 570228, China; 20181685310238@hainanu.edu.cn (F.L.); 18085212210037@hainanu.edu.cn (Y.Y.); 992747@hainanu.edu.cn (X.T.); 20171684310262@hainanu.edu.cn (C.H.)

[2] Hainan Blockchain Technology Engineering Research Center, Haikou 570228, China

[3] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

[4] Department of Computer Science, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA; victor.sheng@ttu.edu

* Correspondence: chengjieren@hainanu.edu.cn

Abstract: Deep learning requires a large amount of datasets to train deep neural network models for specific tasks, and thus training of a new model is a very costly task. Research on transfer networks used to reduce training costs will be the next turning point in deep learning research. The use of source task models to help reduce the training costs of the target task models, especially heterogeneous systems, is a problem we are studying. In order to quickly obtain an excellent target task model driven by the source task model, we propose a novel transfer learning approach. The model linearly transforms the feature mapping of the target domain and increases the weight value for feature matching to realize the knowledge transfer between heterogeneous networks and add a domain discriminator based on the principle of generative adversarial to speed up feature mapping and learning. Most importantly, this paper proposes a new objective function optimization scheme to complete the model training. It successfully combines the generative adversarial network with the weight feature matching method to ensure that the target model learns the most beneficial features from the source domain for its task. Compared with the previous transfer algorithm, our training results are excellent under the same benchmark for image recognition tasks.

Keywords: deep learning; transfer learning; generative adversarial nets; heterogeneous network

check for updates

Citation: Lei, F.; Cheng, J.; Yang, Y.; Tang, X.; Sheng, V.S.; Huang, C. Improving Heterogeneous Network Knowledge Transfer Based on the Principle of Generative Adversarial. *Electronics* **2021**, *10*, 1525. https://doi.org/10.3390/electronics10131525

Academic Editors: Jungong Han and Guiguang Ding

Received: 29 April 2021
Accepted: 18 June 2021
Published: 24 June 2021

1. Introduction

The objective of deep learning models is to optimize the function of large datasets. Models can learn an optimal function using the general process described above. It is foreseeable that traditional machine learning processes will be stretched when the number of tasks is very large or the learning process is very slow. How to maximize the use of previously learned tasks to help the learning of new tasks? How to achieve a better learning effect on the target task on a small network? The initial method uses a learned source task to apply it directly to the target task, and the model achieves the learning of the target task through fine-tuning on the target task [1]. This has been proven to be an effective way. The task of transfer learning is to reduce the amount of data and improve the generalization of the model, so that it can quickly converge and transfer other tasks with a small amount of training data. At present, a lot of work has made a good performance for the homogeneous transfer networks, from the early method of fine-tuning [1] to the fixed network feature extraction layer, and where the learning distance is increased in the classification layer [2,3] through domain adversarial [4–8] idea of learning implicitly distributed distances; therefore, deep transfer learning methods have quickly become an active research field. If the network architectures of the source task and the target task are quite different, there exists no direct method of fine-tuning. A general algorithm is needed to enable the heterogeneous network to complete the transfer. Nowadays, there are several earlier works [9–13], which can be applied to the challenging scenario of knowledge transfer between heterogeneous models and tasks: Attention transfer [12] and Jacobian matching [13] use attention maps generated from feature maps or Jacobians for transferring the source knowledge. L2W-FF [9] further implements the matching rules of knowledge transfer in an automatic way instead of manually adjusting the transfer configuration considering the differences in architecture and tasks between the source and the target domain. Our motivation is that these heterogeneous network transfers are currently mainly driven by feature matching traditional algorithms. We believe that we can use generative adversarial ideas and use network-driven networks combined with weight feature matching to more effectively perform knowledge transfer. Deep neural networks are more powerful for learning general features and transferable features, but some experiments [1] prove that deep features must eventually transition from general features to specific features along the network. As the domain difference increases, feature transfer ability drops significantly in higher layers. Our research topic is to realize the transfer from source task to target task around heterogeneous network. According to previous theories [1], we divide the network layer characteristics into two parts, namely general feature layer and specific feature layer. In our experiments, the general feature layer refers specifically to the low-level convolutional layer and the specific feature layer refers to the high-level fully connected layer. Our new method is to use a combination of generative adversarial network, and feature matching for different network feature maps to improve transmission capacity. The experimental results on benchmarks are excellent. Our contributions are as follows.

We use generative adversarial thinking methods to achieve transfer to heterogeneous networks. As a result of the heterogeneity of the network, we apply the point-to-convolution layer to the target domain network to complete the linear transformation of the target domain feature map and realize the domain discriminant network drive that incorporates the principle of generative transfer.

We successfully combine the generative transfer network with the weight feature matching method and propose a new objective function optimization scheme to complete the model training, ensuring that the target model learns the low-level features that are most beneficial to its own task from the source domain.

2. Related Work

We review two dominant research directions for transfer learning.

2.1. Feature Matching

Feature matching can be understood as a linear transformation, which is a good method in transfer learning. There are also many related researchers in the field of transfer learning exploring feature matching. At the beginning, it was only manual layer-to-layer matching of heterogeneous networks [10–12]; however, there are unavoidable disadvantages with this method. One reason is that too many extra operations will be added. Another reason is that the transferred knowledge is not necessarily available for the target task, which may cause the knowledge that is not conducive to the target task to be transferred into it, thereby reducing the transfer effect. Later, some scholars [9] proposed feature matching with additional weights and realized a method of automatically matching features based on weights. This method mainly updates the weight of the feature matching layer by continuously measuring the distance between the target domain and the source domain, thereby increasing the transfer of useful knowledge and weakening the transfer of knowledge with little relevance. Our model is based on predecessors, using weight feature matching and generative adversarial network mechanisms to improve the effect of knowledge transfer.

2.2. Generative Adversarial Net

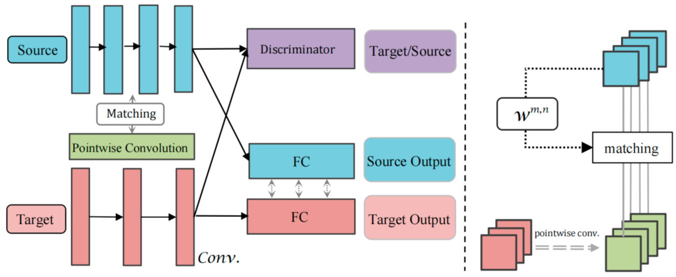
The generative adversarial network is composed of a generative network and a discriminant network. The generator is used to generate fake samples, and the discriminator is used to distinguish between true and false simples. The two game each other until the system reaches a Nash equilibrium. In transfer learning, there is a source domain and a target domain. The target domain can be directly assumed as the sample generated by the generator. The original generator is responsible for extracting features and continuously learning the knowledge of the source domain data, making the discriminator unable to distinguish between the two-domain data. From [10] and others, the idea of adversarial was first introduced into the field of transfer learning, mainly used for the adaptive problem of an important branch of transfer learning. Domain adaptation focuses on the same feature space. Given a labeled source domain D_s and an unlabeled target domain D_t, it is assumed that their feature space and category space are same, but the edge distribution of the domain is different. The labeled source domain data is used to predict the target domain label. Subsequently, there are many different applications and productions in the field of transfer learning, such as image attribute transfer [14] and super-resolution image reconstruction [15]. These are all domain adaptation issues, since domain adaptation is to transfer knowledge with the same feature space, category space and the homogeneous network. Its core function is to adapt the feature distribution of the target domain to the source domain feature distribution, thereby completing the domain feature in-variance. In this paper, we propose a new heterogeneous transfer network, which combines the idea of generative adversarial to perform transfer learning on the heterogeneous network and linear transformation of target domain features. The domain discriminator is used to drive the common layer characteristics of the source domain and the target domain. This can make the target domain more effective and accurate in order to learn the common layer characteristics of the source domain.

The rest of the paper consists of the following parts. In Section 3, we describe our heterogeneous transfer network structure principle and training method. In Section 4, we show the experimental results and evaluations under different configurations. Section 5 explains the conclusion.

3. Our Approach

3.1. Motivation

According to the experiment of previous research [1] in the neural networks, as the domain difference increases, the features learned by the network are gradually proprietary, which means that the transfer ability will significantly decrease as the number of network layers deepens. We divide the network layer features into two parts, general feature layer and specific feature layer. In our experiments, the general feature layer refers specifically to the low-level convolutional layer and the specific feature layer refers to the high-level fully connected layer. We propose a novel transfer network for transfer training. Our goal is to use a combination of generative adversarial nets (GAN), distribution adaptation (this part mainly pays attention to the sample of the same feature and category spaces, same conditional probability distribution and the different edge distributions), and feature matching for different network feature layers to improve the transfer effect. And the generality of the model is proved through testing and evaluation on different general data sets. Our novel method is shown in Figure 1. Deep learning has strongly developed in the two major areas of natural language processing and computer vision [16–20]. Here we mainly use the convolutional neural network commonly used in the field of computer vision as the experimental model of transfer to verify the migration. This method is suitable for convolutional neural networks, but it is not limited to this.



(a) (b)

Figure 1. (**a**) Representation of Heterogeneous transfer network model. (**b**) Schematic representation of weight feature matching: Linear transformation φ_θ the target domain feature map. $w_c^{m,n}$ denote the matching weight of the c-th channel between the feature map of the m-th layer of the source network and the feature map of the n-th layer of the target network after linearly transform φ_θ.

In Section 3.2, we describe the transfer process based on generative adversarial networks (the main function is the domain offset of the middle layer of the network), and Section 3.3 focuses on weight feature matching. In Section 3.4, the domain adaptation method of the high-level network is described. In the last Section 3.5, we specifically describe the experimental training process of our model.

3.2. Generative Adversarial Nets

The goal of traditional GAN is to generate training samples. Since there is a source domain and a target domain naturally in transfer learning, we can avoid the process of generating samples and directly treat the data of the target domain as the generated samples. At this time, the function of the generator changes and does not generate new samples; it plays the function of feature extraction: learning the characteristics of the domain data continuously, making the discriminator unable to distinguish between the two domains. In this way, the original generator can also be called a feature extractor.

A discriminate mechanism is added to the training of the neural network. The goal is to make the discriminator unable to distinguish the difference between the two fields, continuously promote the knowledge transfer of the target domain network and accelerate the driving of the target network to learn the common characteristics of the source domain and the target domain.

Traditional transfer problems generally use fixed feature representations, but the adversarial transfer network in this paper focuses on how to select transferable features between different domains and make tidy target networks learn knowledge through source network more accurately and quickly. In other words, a good transferable feature should meet two conditions: firstly, in the face of these features, it is impossible to distinguish whether they come from the target domain or the source domain; secondly, using these features to complete the classification task better. Therefore, the network loss consists of two parts: training loss (label predictor loss) and domain discriminate loss [21].

We further define a discriminator network D_{θ_d}, a source domain network S_{θ_s}, and a target domain network T_{θ_t}. θ_d, θ_s, and θ_t represent the parameters of the D_{θ_d}, S_{θ_s}, and T_{θ_t}, respectively. Our ultimate goal is to be able to predict labels y^t given the input I^t for the target distribution. We assume that the model works with input samples $I^t \in T$, where T is input space and certain output of y^t from the label space Y^t. For sample I_n^t, $n = 1, ..., N$, we describe the sample by a real-valued tensor of size W × H × C. We assume that there are two distributions $P_S\left(I^s, y^s\right)$ and $P_t\left(I^t, y^t\right)$ on $X \otimes Y$, which will be referred to as the source distribution and the target distribution. We denote with d_i ($d_i \in \{0, 1\}$) the binary variable (discriminator output) for the *i-th* example. If $d_i = 1$, it proves that the sample comes from the source distribution $I_i^s \sim P_s(I^s)$. If $d_i = 0$, it proves that the sample comes from the source distribution $I_i^t \sim P_t\left(I^t\right)$.

$$min_{\theta_t} max_{\theta_d} V\,(T,\ D) = E_{I^s \sim P_s(I^s)}\left[\log D_{\theta_d}(I^s)\right] + E_{I^t \sim P_G(I^t)}\left[\log\left(1 - D_{\theta_d}\left(T_{\theta_t}\left(I^t\right)\right)\right]\right. \quad (1)$$

$$min_{\theta_t} max_{\theta_d} V\ (T,\ D) = E_{I^s \sim P_s(I^s)}\ \left[\log D_{\theta_d}\ (I^s)\right] + E_{I^t \sim P_G(I^t)} \left[\log\left(-D_{\theta_d}\ \left(T_{\theta_t}\ \left(I^t\right)\right)\right]\right. \tag{2}$$

3.3. Weight Feature Matching

If the convolutional neural network is well trained for the task, its intermediate feature space should have useful knowledge for the task [9–13]. Many predecessors have studied neural network feature matching, some researchers have manually matched the features [10–13], and some have discerned automatic matching of features [9]. Intermediate feature mapping of the m-th layer of the pre-trained source network is used to mean $S^m_{\theta_s}\ (I^s)$ and feature mapping of the n-th layer of the target network is used to mean $S^n_{\theta_t}\ \left(I^t\right)$. We minimize the following l^2 objective, similar to that used in FitNet [10] and L2t-ww [9] to transfer the knowledge from $S^m_{\theta_s}(I^s)$ to $S^n_{\theta_t}\left(I^t\right)$.

$$\|\varphi_\theta\left(T^n_{\theta_t}\ \left(I^t\right)\right) - S^m_{\theta_s}(I^s))\|^2_2 \tag{3}$$

This is Equation (3) of the equation: we used pointwise convolution to linearly transform φ_θ the target domain feature map $T^n_{\theta_t}\left(I^t\right)$. This process produces parameter θ. We set weights for feature matching between channels to focus on the more closely related channels. We use $w^{m,n}_c$ to denote the matching weight of the *c-th* channel between the feature map of the *m-th* layer of the source network $S^m_{\theta_s}(I^s)$ and the feature map of the *n-th* layer of the target network $\varphi_\theta\ \left(T^n_{\theta_t}\left(I^t\right)\right)$ after linearly transforming φ_θ. $\mathcal{L}^{m,n}_{fm}\ \left(\theta|I^t, w^{m,n}\right)$ is used to represent the loss function of weight feature matching.

$$\mathcal{L}^{m,n}_{fm}\ \left(\theta_t|I^t, w^{m,n}\right) = \frac{1}{HW}\sum_c w^{m,n}_c \sum_{i,j}\left(\varphi_\theta\left(T^n_{\theta_t}\left(I^t\right)\right) - S^m_{\theta_s}\ (I^s)\right)^2 \tag{4}$$

3.4. Maximum Mean Discrepancy

This part is mainly for the same feature space of the source domain and target domain, $S = T$, and their category spaces are also the same where $Y^s = Y^t$, and the conditional probability distribution is also the same $P_s\ (Y^s|I^s) = P_T\ \left(Y^t\middle|I^t\right)$. But the edge distributions of these two domains are different, $P_S\ (I^s)\ \neq\ P_T\ \left(I^t\right)$. It also can be seen as a domain adaptation problem. Domain adaptation is also an important part of the field of transfer learning, and it is commonly used in many unsupervised and less-supervised tasks. When we encounter other types of transfer data, we can manually set the hyperparameter γ to 0. For high-level networks, features are the most exclusive, so domain adaptation of high-level networks is inevitable. Most of the documents [2,3] have carried out various transfer experiments on the transfer of high-level features. Here, we use the most widely used MMD measurement criteria to transfer the upper layer such as FC layer. Use the same method as DDC model [2]. This model adds a distance loss to the final classification layer to reduce the use of a kernel function method, the maximum mean discrepancy (MMD), which measures the two distributions of the source domain and the target domain in the regeneration Hilbert. The distance of space is a nuclear learning method. $\phi\ (\cdot)$ is a mapping used to map the original variable to the reproducing kernel Hilbert space (RKHS) [22]. The Hilbert space is complete for the inner product of the function, the reproducing nuclear Hilbert space is a Hilbert space with reproducibility $\langle K(x,\cdot), K(y,\cdot)\rangle_H = K(x,y)$. After expanding the square, the inner product in the RKHS space is converted into a kernel function, so MMD can be directly calculated by the kernel function.

$$\mathcal{L}_{MMD}\ \left(\theta_t|I^t, \theta_s|I^s\right) = \|\frac{1}{|I^s|}\sum_{I^s\in S}\phi\ (I^s) - \frac{1}{|I^t|}\sum_{I^t\in T}\phi\ \left(I^t\right)\|^2 \tag{5}$$

3.5. Model Holistic Training

Our final loss $\mathcal{L}_s$ to train a target model is given as follows. In particular, when we train data, the same feature space of the source domain and target domain, $S = T$, and their category spaces are also the same where $Y^s = Y^t$, and the conditional probability distribution is also the same $P_s\left(Y^s|I^s\right) = P_T\left(Y^t\middle|I^t\right)$. But the edge distributions of these two domains are different, $P_S(I^s) \neq P_T\left(I^t\right)$, we should make $\gamma \neq 0$. $\mathcal{L}_{org}\left(\theta_t|I^t, y^t\right)$ is the original loss (e.g., cross entropy) in the target model, $\mathcal{L}_g\left(\theta_t|I^t, \theta_s|I^s\right)$ is the loss of generative adversarial net and λ, $\beta > 0$, γ $(\gamma = 0)$ is a hyper-parameter:

$$\mathcal{L}_s = \mathcal{L}_{org}\left(\theta_t|I^t, y^t\right) + \lambda\mathcal{L}_g\left(\theta_t|I^t, \theta_s|I^s\right) + \beta\mathcal{L}_{fm}^{m,n}\left(\theta_t|I^t, w^{m,n}\right) + \gamma\mathcal{L}_{MMD}\left(\theta_t|I^t, \theta_s|I^s\right) \quad (6)$$

Firstly, the resulting parameter θ_t^T is learned only using the knowledge of the source model, thus we updated the target model for T times via gradient-based algorithms for minimizing $\mathcal{L}_{fm}^{m,n}$ and $\mathcal{L}_g$. We designed a new type of training scheme to update the network parameters of feature matching and generative discriminate network emphatically by setting a hyper parameter value (T). The purpose of this process is obvious and important; it enhances the influence of the regularization term $\mathcal{L}_{fm}^{m,n}$ and $\mathcal{L}_g$. on the target model parameters, and because the source domain data is not used, the target features are completely provided by the source domain. Secondly, we used θ_t^{T+1} from θ_t^T to update and minimize $\mathcal{L}_{org}\left(\theta_t|I^t, y^t\right)$ once. Thirdly, we measured $\mathcal{L}_{org}\left(\theta_t|I^t,\ y^t\right)$ and updated $w_c^{m,n}$, θ_d to minimize $\mathcal{L}_{fm}^{m,n}$ and $\mathcal{L}_g$. To train the target model, we alternatively updated the target model parameters θ_t and parameters $w_c^{m,n}$ to make the ability of our model. The purpose is to increase the influence of the source domain network on the target domain network training and help the target domain network training quickly. The proposed training scheme is formally outlined in Algorithm 1.

Algorithm 1 Minibatch stochastic gradient descent training of model. Learning of θ_t, $w^{m,n}$, θ_d

```
Input: Dataset $D_{train} = \left\{\left(I_i^t, y_i^t\right)\right\}$, learning rate $\alpha$
Repeat
Sample a batch $B \subset D_{train} with\ |B| = \mathbf{B}$
        Update $\theta_t$ to minimize $\frac{1}{B}\sum_{(I_i^t, y_i^t)\in B} \mathcal{L}_s$
for t = 0 to T − 1 do
        $\theta_t^{T+1} \leftarrow \theta_t^T - \alpha\nabla_\theta\frac{1}{B}\sum_{(I_i^t, y_i^t)\in B} = \mathcal{L}_{fm}^{m,n}\left(\theta_t|I^t, w^{m,n}\right)$
    Update the discriminator by ascending its stochastic gradient:
        $\nabla_{\theta_d}\frac{1}{B}\sum_{(I_i^t, y_i^t)\in B}\left[\log D_{\theta_d}(I^s) + \log(-D_{\theta_d}(T_{\theta_t}(I^t))\right]$
    Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_t}\frac{1}{B}\sum_{(I_i^t, y_i^t)\in B}\log(-D_{\theta_d}(T_{\theta_t}(I^t))$
end for
$\theta_t^{T+1} \leftarrow \theta_t^T - \alpha\nabla_{\theta_t}\frac{1}{B}\sum_{(I_i^t, y_i^t)\in B}\mathcal{L}_{org}\left(\theta_t|I^t, y^t\right)$
        Update $w_c^{m,n}$ using $\alpha\nabla_\phi\frac{1}{B}\sum_{(, I_i^t, y_i^t)\in B}\mathcal{L}_{fm}^{m,n}\left(\theta_t|I^t, w^{m,n}\right)$
until done
```

4. Experiments

Our experiments are mainly divided into two parts. The first part is the transfer experiment on the public benchmark [23–27], and compared with the experimental results of the heterogeneous transfer network of the predecessors. The second part obtains the transfer effect of different network layers by using the different parameter regularization methods adopted by our model, and discusses the transfer characteristics and transfer methods of each layer of the network.

4.1. Setup

In order to evaluate our model and other models more easily, we chose classical and universal dataset tests and a backbone with superior performance as our source and target domain heterogeneous network. We performed experiments on 32 × 32 image classification tasks, using the Tiny ImageNet [27] dataset as a source task, and CIFAR-10 CIFAR-100 [28] and STL-10 [24] datasets as target tasks. Tiny ImageNet has 200 classes. Each class has 500 training images, 50 validation images, and 50 test images. The sample size used in this experiment is 32 × 32 × 3. CIFAR-10 datasets have 10 classes. There are 5000 training images. CIFAR-100 datasets have 100 classes. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 super classes. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs). STL-10 [24] datasets have 10 classes. There are 500 training images, with 800 test images per class. The same is the case with L2T-ww [9] we resize them into 32 × 32 when training and testing. We trained 32-layer ResNet [29] on the source tasks and 9-layer VGG [30] on the target tasks. At the same time, we conducted experiments on a deeper target network, training 34-layer ResNet [29] on the source tasks and 19-layer VGG [30] on the target tasks. We performed experiments on 224 × 224 × 3 image classification tasks, and used the ImageNet [23] dataset as a source task, PASCAL VOC2007 [31] and CUB200 [25] datasets as target tasks. In order to reflect the key role played by every part in transfer learning intuitively, we used the MNIST dataset [32] as a source task, and MNIST-M dataset as a target task.

In terms of optimizer settings for network training, all target networks are trained by stochastic gradient descent (SGD) with a momentum of 0.9. We used an initial learning rate 0.1 and 200 epochs for all experiments.

4.2. Results on Different Experiments

We compared our methods with the following prior methods: learning without forgetting (LwF) [11], and attention transfer (AT) [12]. In our experimental setup, every method came from scratch for baselines. Here, these models include the model [9] that can perform automatic feature matching. Attention transfer [12] and Jacobian matching [13] use attention maps generated from feature maps or Jacobians for transferring the source knowledge. L2W-FF [9] further implements the matching rules of knowledge transfer in an automatic way, taking into account the differences in architecture and tasks between the source and the target, without the need to manually adjust the transfer configuration. For small network experiments, the inputted sample size was 32 × 32 × 3. In order to verify the versatility of the model, two different migration tasks were performed: Tiny ImageNet→CIFAR-100 and Tiny ImageNet→STL-10 (see Table 1). For big network experiments, the sample size we inputted was 224 × 224 × 3. We used a pre-trained 34-layer ResNet on ImageNet. In order to verify the versatility of the model, two different transfer tasks were performed: ImageNet → Pascal VOC2007 and ImageNet → CUB-200 (see Table 2 and Figure 2).

Table 1. Classification accuracy (%) of small network.

Source Task	**Tiny ImageNet**	
Target task	CIFAR-100	STL-10
Scratch	67.69 ± 0.22	67.69 ± 0.22
AT	69.23 ± 0.09	69.23 ± 0.09
LwF	69.97 ± 0.24	69.97 ± 0.24
L2T-ww	70.96 ± 0.61	70.96 ± 0.61
ours	72.85 ± 0.25	70.99 ± 0.88

Table 2. Classification accuracy (%) of big network.

Source Task	**ImageNet**	
Target task	Pascal VOC	CUB 200
AT	79.22 ± 0.59	44.52 ± 0.09
LwF	79.55 ± 0.64	44.56 ± 0.24
L2T-ww	80.96 ± 0.61	46.96 ± 0.67
ours	83.33 ± 0.64	47.11 ± 0.69

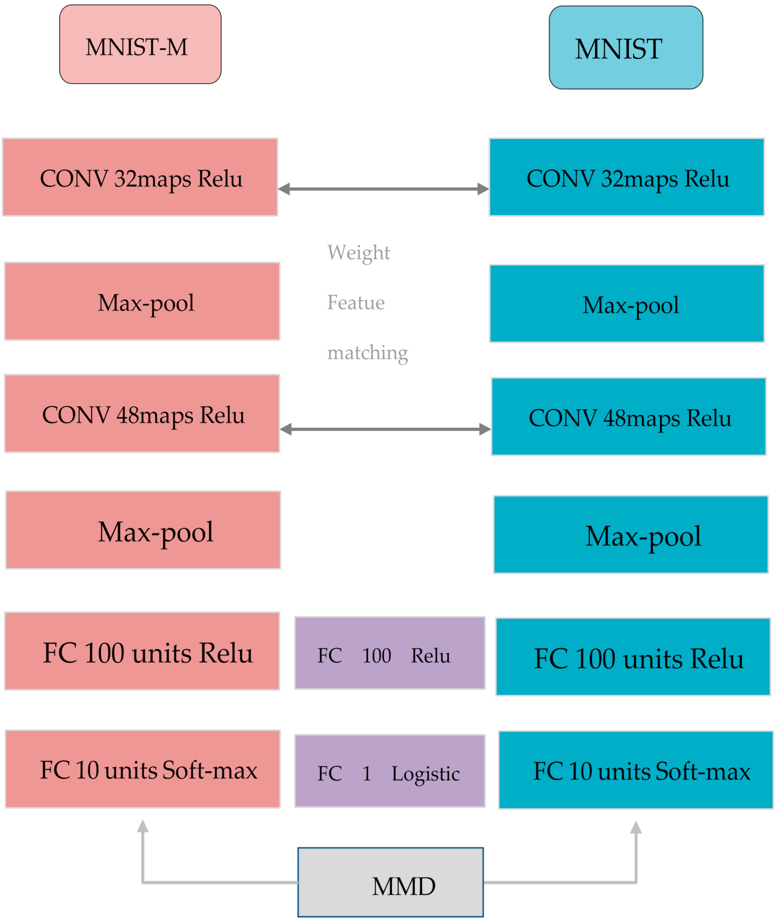
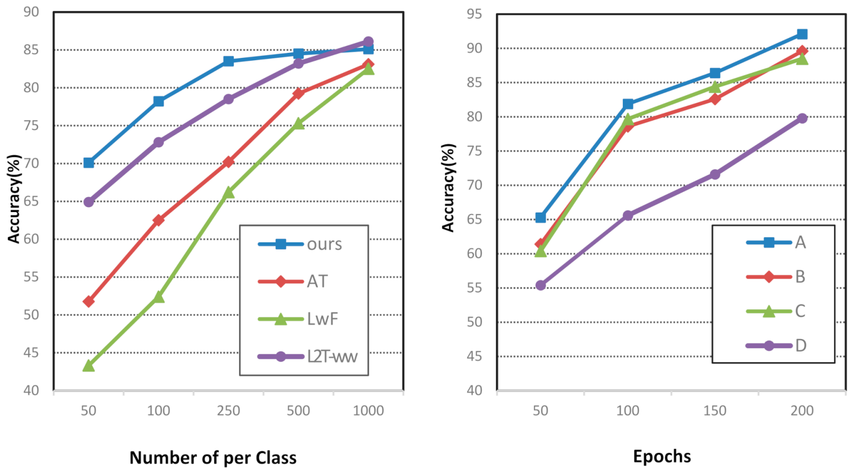


Figure 2. The network detail architecture of MNIST → MNIST-M.

5. Discussion

In order to reflect the key role played by every part in transfer learning intuitively, our task was MNIST → MNIST-M (see Figure 2). This is the same feature space of the source domain and target domain, $S = T$, and their category spaces are also the same where $Y^s = Y^t$, and the conditional probability distribution is also the same where $P_s\ (Y^s|I^s) = P_T\ (Y^t|I^t)$. However, the edge distributions of these two domains are different, where $P_S\ (I^s) \neq P_T\ (I^t)$. We used the popular MNIST [32] dataset as the source domain, and MNIST-M was created by using each MNIST digit as a binary mask and inverting with it the colors of a background image. The background images are random crops uniformly sampled from the Berkeley Segmentation Data Set (BSDS500) [31].

In order to reflect the effect of knowledge transfer in the source network, we divided the training data and conducted a comparative experiment to control the number of datasets. We used Tiny ImageNet as a source task, and used CIFAR-10 datasets as target tasks. We divided the target domain dataset into five levels. We used $N \in \{50, 100, 250, 500, 1000\}$ training samples for each class and compared with previous models. The training of each class was done under the same hyperparameters. It can be seen from Figure 3a that our model has a higher accuracy rate with a smaller number of samples, thus has a greater advantage. There are two main reasons. First, compared with other previous methods, we do not use a one-step training gradient update in the training iteration scheme. We designed a new type of training scheme to update the network parameters of feature matching and generative discriminate network emphatically by setting a hyper parameter value (*T*). This increases the influence of the source network on the learning procedure of the target model, since the target features are solely trained without target labels. Secondly, our discriminator mechanism can drive the training efficiency of the target domain network faster, and is more conducive to training with fewer samples. This fully illustrates that our new heterogeneous transfer network is driven by generative adversarial network and weight feature matching is more obvious in transfer knowledge.



(**a**) (**b**)

Figure 3. (**a**) Classification accuracy (%) of Tiny ImageNet → CIFAR-10 with varying numbers of training samples per class in CIFAR-10. (**b**) Classification accuracy (%) of MNIST → MNIST-M. A is the reference.

To research the effectiveness of knowledge transfer between different layers and different transfer algorithms, we adopted the method of controlled variables and designed four comparative experiments. The training of each class is done under the same hyperparameters. Experiment A is trained under the complete training model system designed by our novel method. The B experimental model cancel the Generative Adversarial Nets, which are used to transfer the general feature layer and middle feature layer. The C experimental model cancel the weight feature matching that has a transfer effect on the middle feature layer. The D experimental model cancel the distribution adaptation that has a transfer effect on the proprietary feature layer. To visualize the difference between the three experiments, we have drawn the experimental results into a line chart (see Figure 3b). We evaluate the results of the experiment. Experiment D has the greatest impact on the model, followed by experiments B and C. This results also verifies the conclusions of some network feature studies [1]. In neural networks, as the domain difference increases, the features learned by the network are proprietary gradually, which means that the portability will decrease significantly as the number of network layers deepens.

6. Conclusions

This paper proposes a new, more optimized, heterogeneous transfer network model, which mainly uses the principle of generative adversarial, as well as the network-driven network to combine weight feature matching for more effective knowledge transfer on middle-level feature maps. We used the knowledge transfer of complex source networks to train simple target domain networks effectively and use less target domain data to complete the training of heterogeneous target networks on the basis of pre-trained complex networks. Progress has been made for improvement in the research of heterogeneous networks and our findings on the field of transfer learning are pivotal.

Author Contributions: Conceptualization, F.L. and Y.Y.; methodology, F.L.; software, F.L.; validation, F.L., Y.Y., and C.H.; data curation, F.L.; writing—original draft preparation, F.L.; writing—review and editing, J.C., X.T., and Y.Y.; project administration, Y.Y. and V.S.S.; funding acquisition, X.T. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research and Development Program of Hainan Province (Grant No. ZDYF2020040), the Major science and technology project of Hainan Province (Grant No. ZDKJ2020012), Hainan Provincial Natural Science Foundation of China (Grant Nos. 2019RC098 and 620MS021), National Natural Science Foundation of China (NSFC) (Grant No.61762033).

Data Availability Statement: [CUB200] Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset; http://www.vision.caltech.edu/visipedia/CUB-200-2011.html (access on 6 November 2011).

Acknowledgments: We would like to thank Jingyang Song and YunFeng Zhang for valuable comments on English writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014; JMLR.org. 2014; Volume 32, pp. 647–655.
2. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
3. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning Transferable Features with Deep Adaptation Networks. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; Volume 37, pp. 97–105.
4. Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; Erhan, D. Domain Separation Networks. In Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016; pp. 343–351.
5. Shen, J.; Qu, Y.; Zhang, W.; Yu, Y. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 4058–4065.
6. Yu, C.; Wang, J.; Chen, Y.; Huang, M. Transfer Learning with Dynamic Adversarial Adaptation Network. In Proceedings of the 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, 8–11 November 2019; pp. 778–786.
7. Li, Y.; Peng, X. Learning Domain Adaptive Features with Unlabeled Domain Bridges. *arXiv* **2019**, arXiv:1912.05004.
8. Ganin, Y.; Lempitsky, V.S. Unsupervised Domain Adaptation by Backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; Volume 37, pp. 1180–1189.
9. Jang, Y.; Lee, H.; Hwang, S.J.; Shin, J. Learning What and Where to Transfer. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 3030–3039.
10. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for Thin Deep Nets. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
11. Li, Z.; Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]
12. Zagoruyko, S.; Komodakis, N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
13. Srinivas, S.; Fleuret, F. Knowledge Transfer with Jacobian Matching. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4730–4738.
14. Xie, D.; Yang, M.; Deng, C.; Liu, W.; Tao, D. Fully-Featured Attribute Transfer. *arXiv* **2019**, arXiv:1902.06258.
15. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.P.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 105–114.
16. Cheng, J.L.; Liu, Y.; Tang, X.; Sheng, V.S.; Li, M. DDoS Attack Detection via Multi-Scale Convolutional Neural Network. *Comput. Mater. Contin.* **2020**, *62*, 1317–1333. [CrossRef]

17. Cheng, J.; Cai, C.; Tang, X.; Sheng, V.S.; Guo, W.; Li, M. DDoS Attack Information Fusion Method Based on CNN for Multi-Element Data. *Comput. Mater. Contin.* **2020**, *63*, 131–150. [CrossRef]
18. Yan, B.; Tang, X.; Liu, B.; Wang, J.; Zhou, Y.; Zheng, G.; Zou, Q.; Lu, Y.; Tu, W. An Improved Method for the Fitting and Prediction of the Number of COVID-19 Confirmed Cases Based on LSTM. *Comput. Mater. Contin.* **2020**, *64*, 1473–1490. [CrossRef]
19. Long, M.; Zeng, Y. Detecting Iris Liveness with Batch Normalized Convolutional Neural Network. *Comput. Mater. Contin.* **2019**, *58*, 493–504. [CrossRef]
20. Liu, Z.; Wang, X.; Lu, K.; Su, D. Automatic Arrhythmia Detection Based on Convolutional Neural Networks. *Comput. Mater. Contin.* **2019**, *58*, 497–509. [CrossRef]
21. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
22. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.-P.; Schölkopf, B.; Smola, A.J. Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy. In Proceedings of the 14th International Conference on Intelligent Systems for Molecular Biology 2006, Fortaleza, Brazil, 6–10 August 2006; 2006; pp. 49–57.
23. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
24. Coates, A.; Ng, A.Y.; Lee, H. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, FL, USA, 11–13 April 2011; Volume 15, pp. 215–223.
25. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*; Technical Report CNS-TR-2011-001; California Institute of Technology: Los Angeles, CA, USA, 2011.
26. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
27. Tiny-Imagenet. Available online: https://tiny-imagenet.herokuapp.com/ (accessed on 1 February 2014).
28. Krizhevsky, A. Convolutional Deep Belief Networks on CIFAR-10. 2012. Available online: http://www.cs.toronto.edu/~{}kriz/cifar.html(accessed on 8 April 2009).
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
31. Arbelaez, P.; Maire, M.; Fowlkes, C.C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [CrossRef] [PubMed]
32. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]