《神经网络理论与应用》第七讲
Neural Network Theory and Applications (7)

主讲教师：吕宝粮
助教：李芮；李子怡、马天放、刘乐典

上海交通大学计算机科学与工程系

bllu@sjtu.edu.cn
http://bcmi.sjtu.edu.cn/~lubaoliang

2022年3月31日

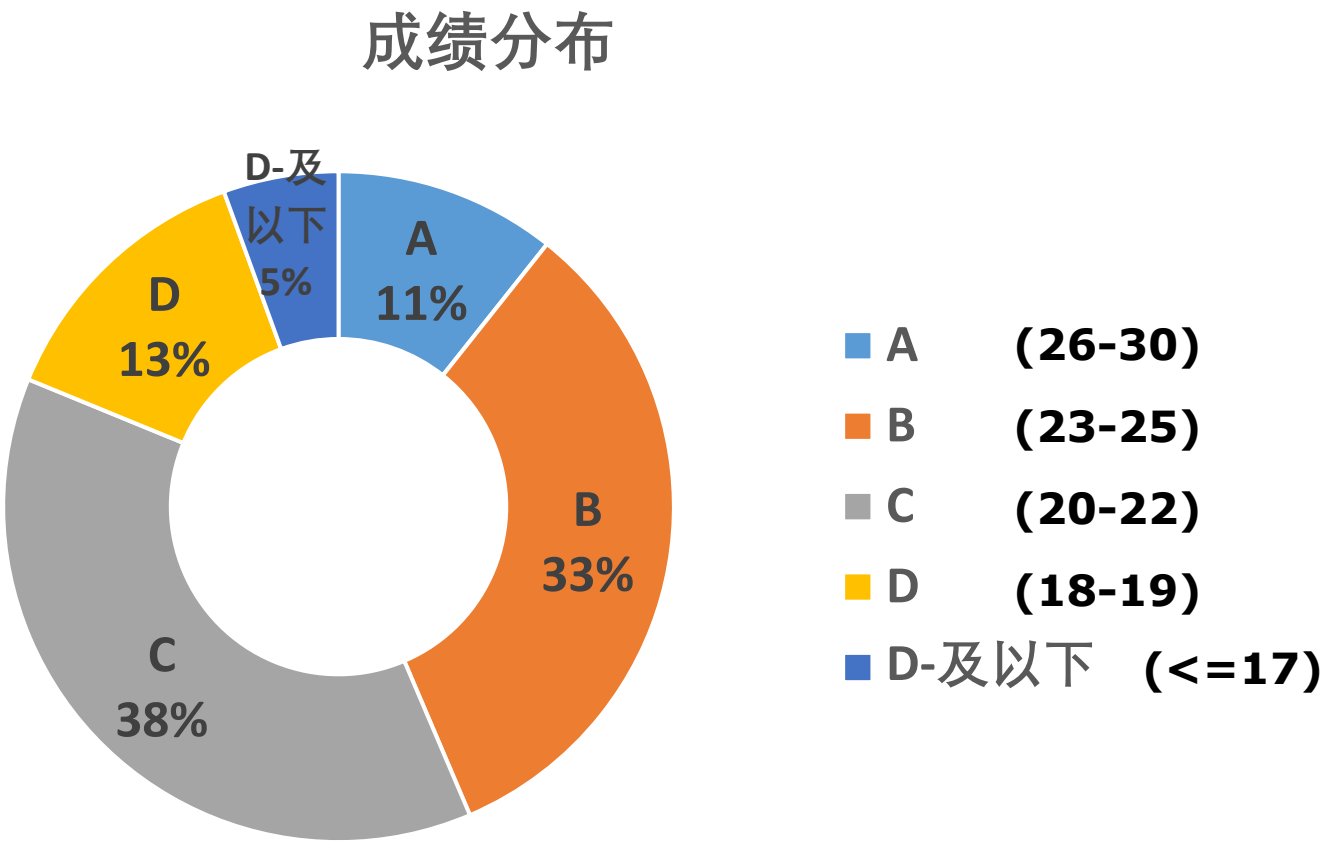
Summary of Lecture Six

- Support vector machine
- Summary of shallow learning
- Introduction to deep learning
- Transfer learning
- Quiz (20:05-20:20)

Outline of Lecture Seven

- Evaluation of Assignment #1
- Transfer learning
- Assignment #2

第一次Quiz结果

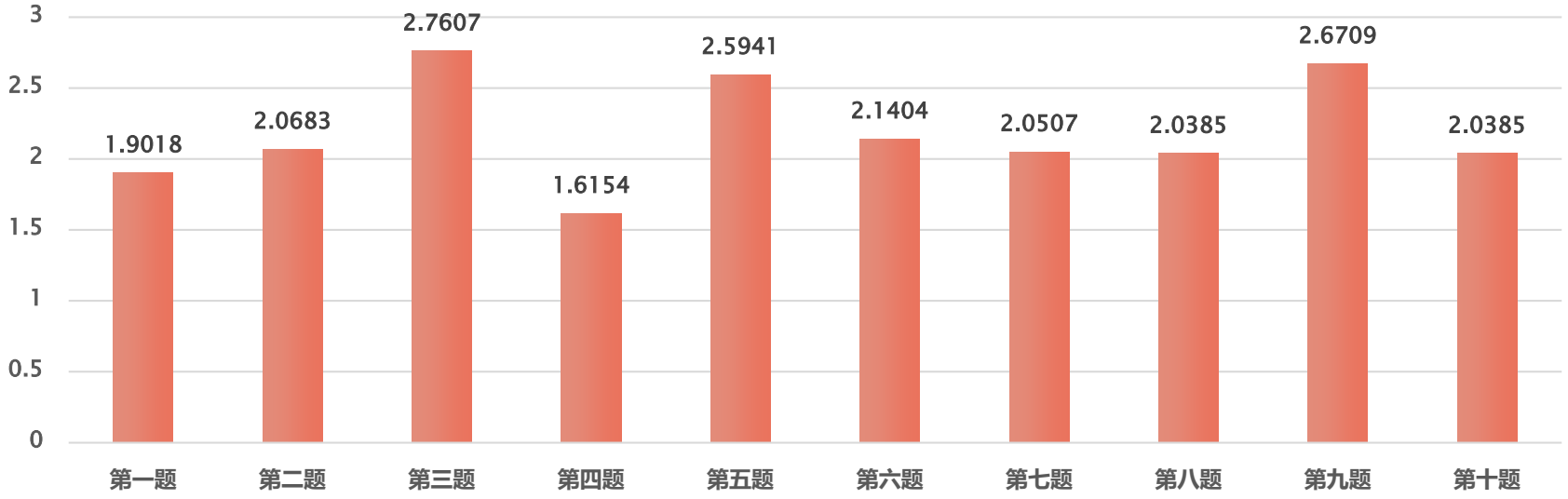


第一次Quiz结果

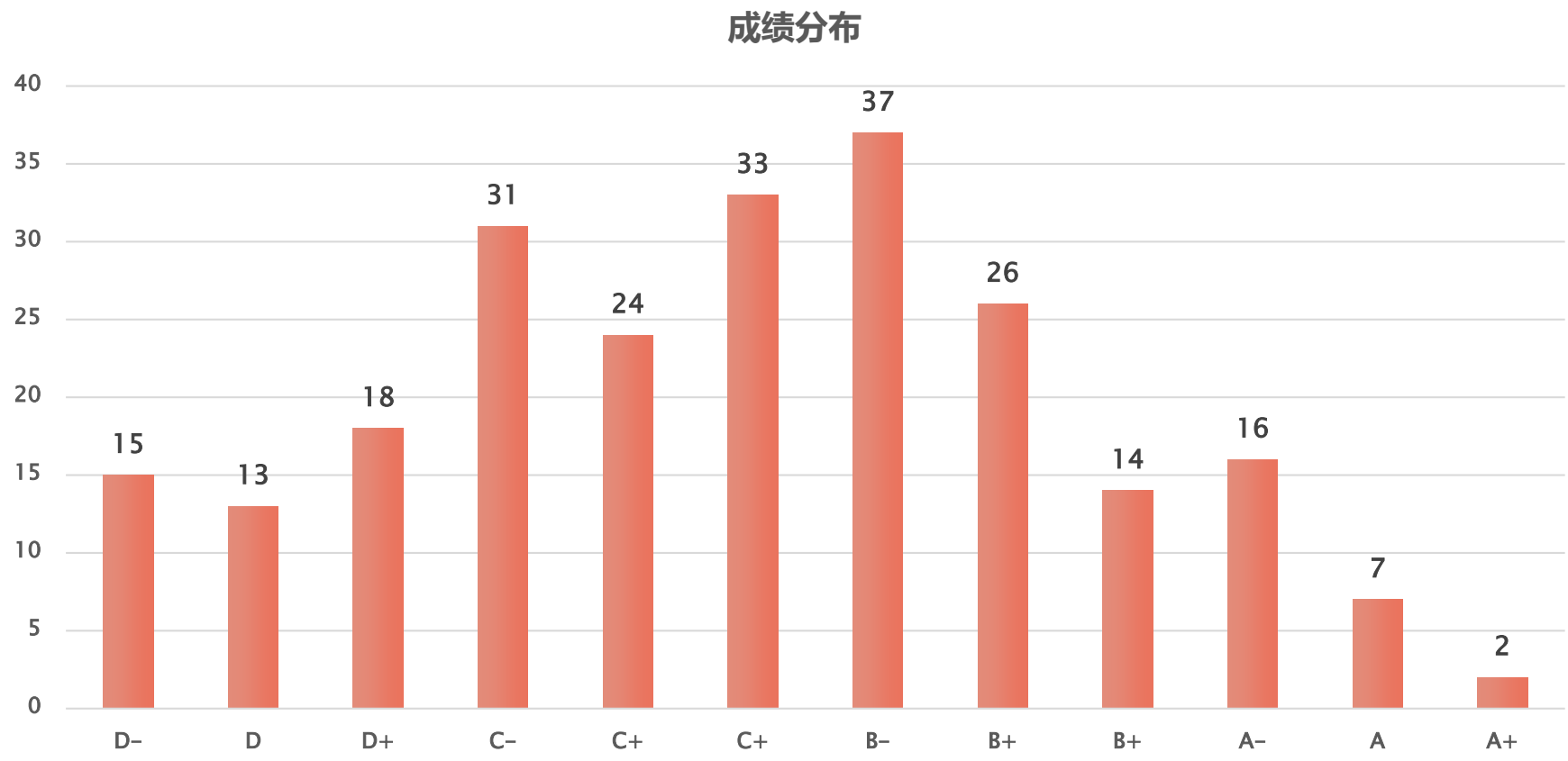
总人数:234人

满分 30

平均分 21.9



第一次Quiz结果

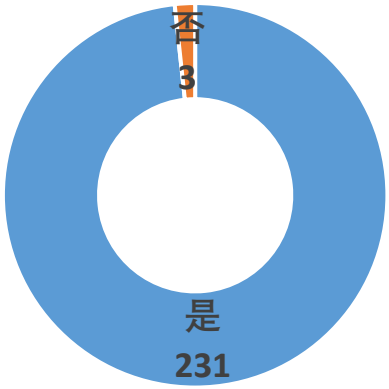

成绩分布
40
35
30
25
20
15
10
5
0
15
13
18
31
24
33
37
26
14
16
7
2
D-
D
D+
C-
C+
C+
B-
B+
B+
A-
A
A+

第一次Quiz结果

1、关于神经元的激活函数，有线性的和非线性两种，试问：

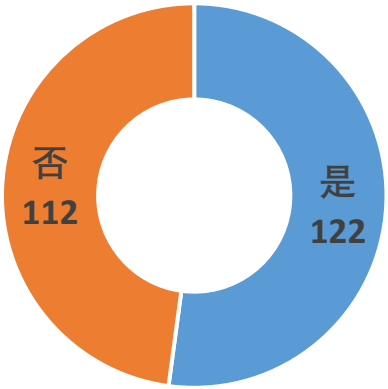
a) 它们对神经网络的学习能力有影响吗？

是



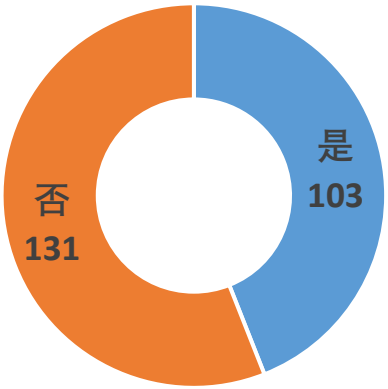
b) 它们对决定神经网络的层数有影响吗？

否

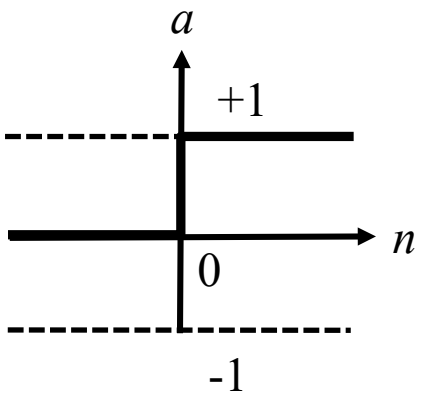


c) 它们对使用什么样的学习算法有影响吗？

是

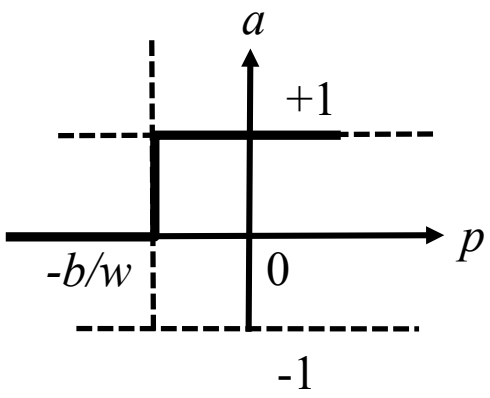


Activation (Transfer) Functions



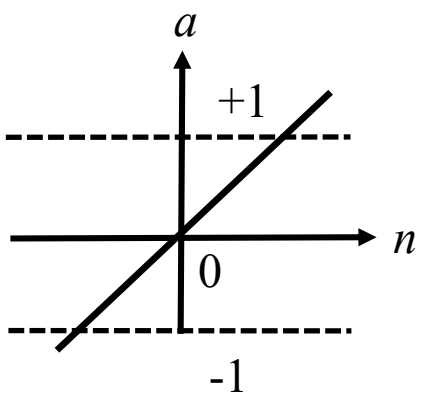
a=hardlim(n)

Hard Limit Transfer Function



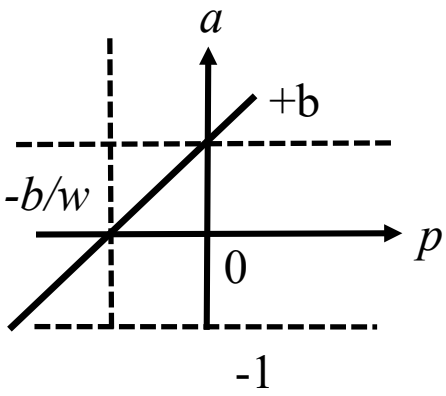
a=hardlim(wp+b)

Single-Input *hardlim* Neuron



a=purelin(n)

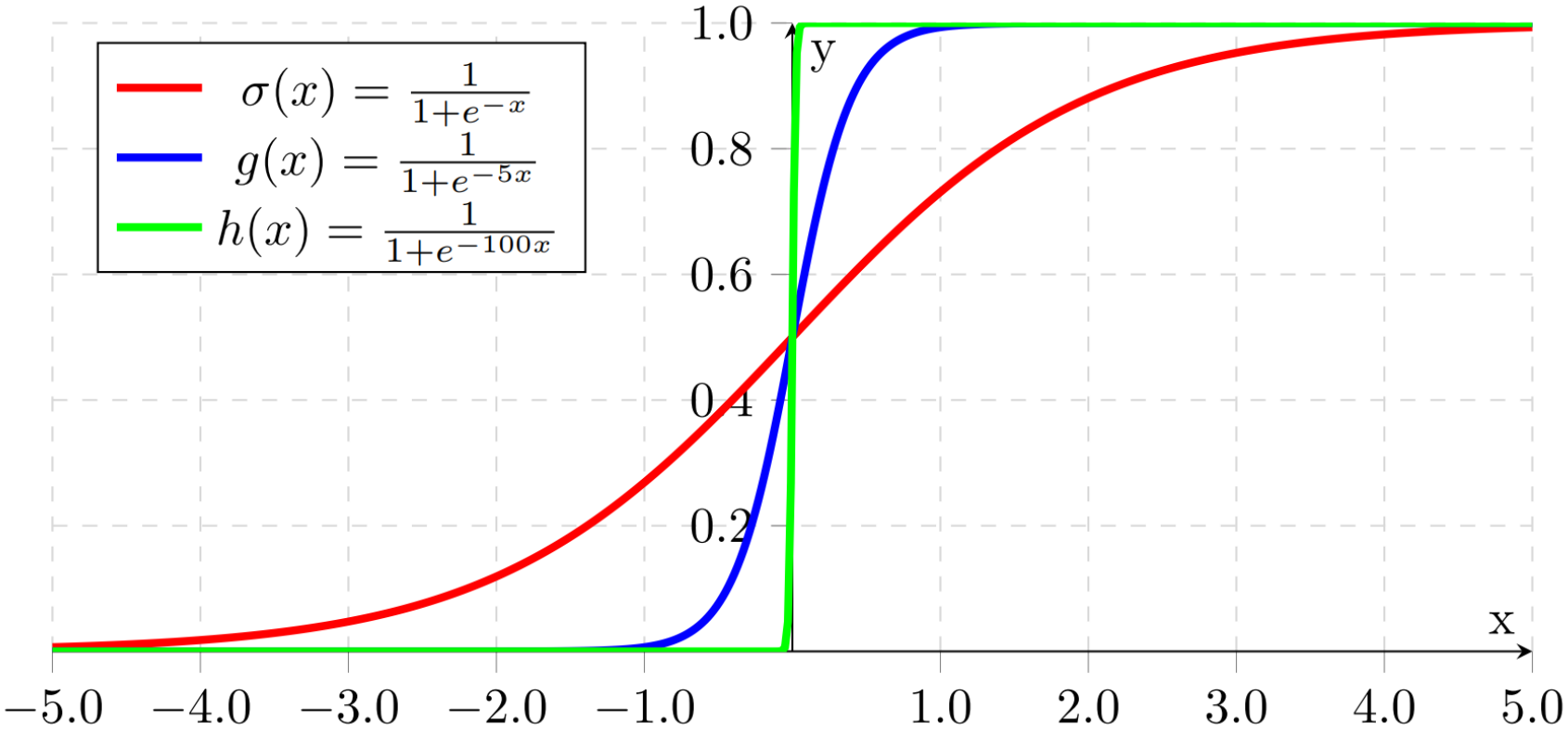
Linear Transfer Function



a=purelin(wp+b)

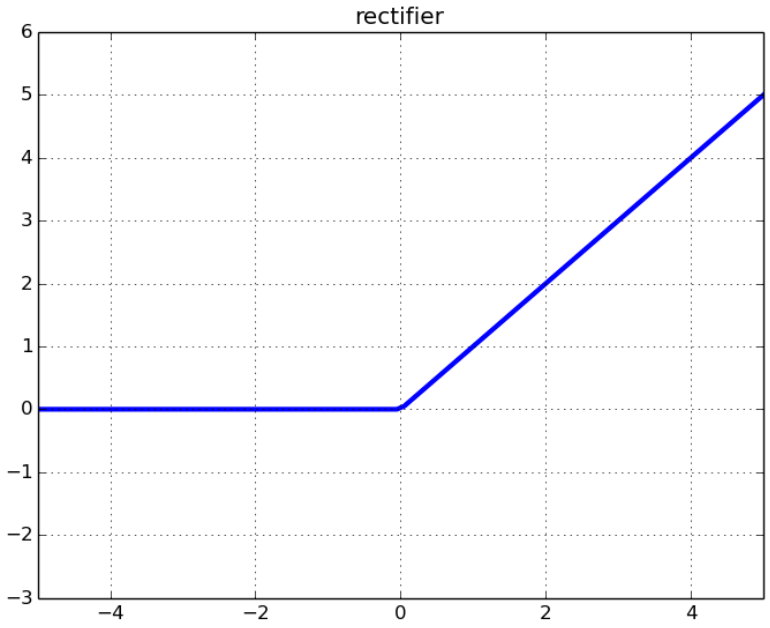
Single-Input *purelin* Neuron

Sigmoidal Activation Function



Rectified Linear Unit: ReLU

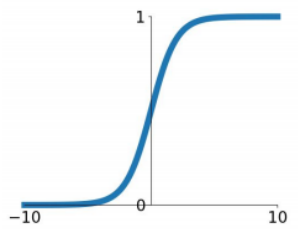
- *ReLU:* **Rectified Linear Unit** (线性整流函数)
- $ReLU = \max(0, x) = \begin{cases} 0, if\ x \leq 0 \\ x, if\ x > 0 \end{cases}$



Various Activation Functions

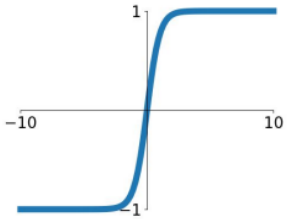
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



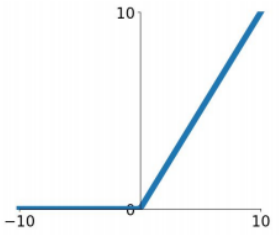
tanh

$$\tanh(x)$$



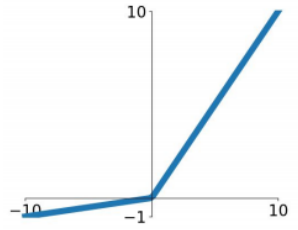
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

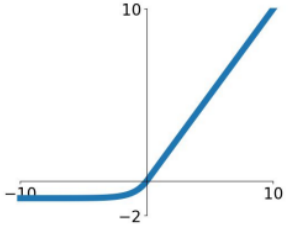


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

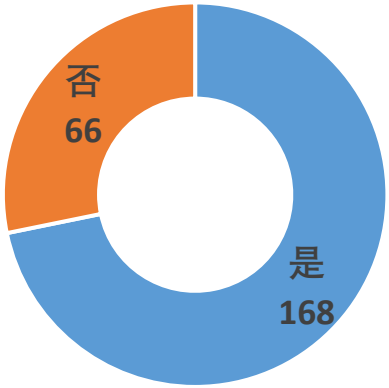


第一次Quiz结果

2、感知机学习算法与最小均方算法的主要区别是：

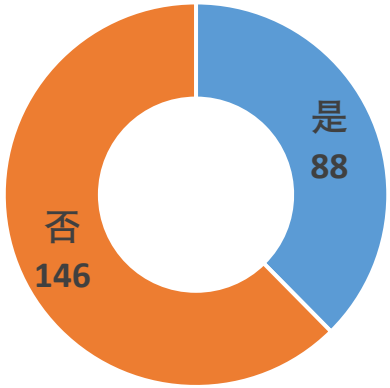
a） 调节权重的机理不同？

是



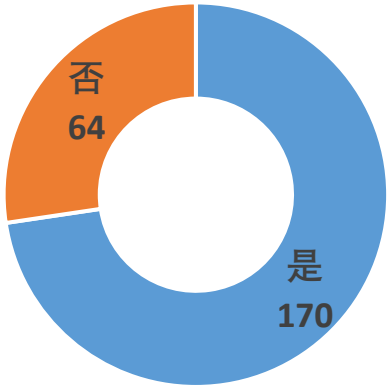
b） 性能评价的标准不同？

否



c） 形成的分界面不同？

是



LMS Versus Peceptron Learning Rule

LMS Learning Rule

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

$${}_1\mathbf{w}(k+1) = {}_1\mathbf{w}(k) + 2\alpha e(k)\mathbf{p}(k)$$

$$b(k+1) = b(k) + 2\alpha e(k)$$

Perceptron Learning Rule

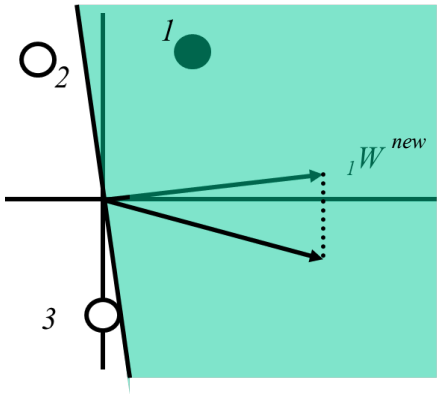
$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + e\mathbf{p} = {}_1\mathbf{w}^{old} + (t-a)\mathbf{p}$$

$$b^{new} = b^{old} + e$$

Difference between LMS and Perceptron Learning Rule

Perceptron rule stops as soon as the patterns are correctly classified, even though some patterns may be ***close to the boundaries.***

The LMS algorithm minimizes the mean square error. Therefore it tries to move the decision boundaries ***as far from the reference patterns as possible***.

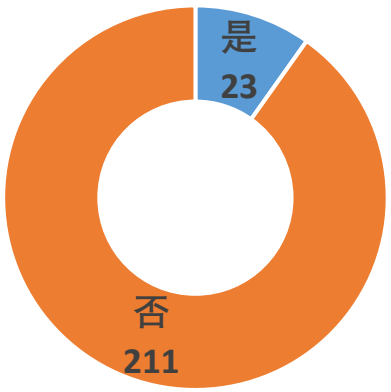


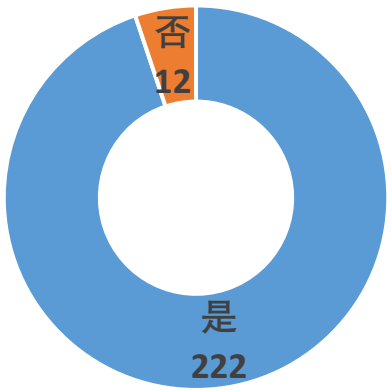
第一次Quiz结果

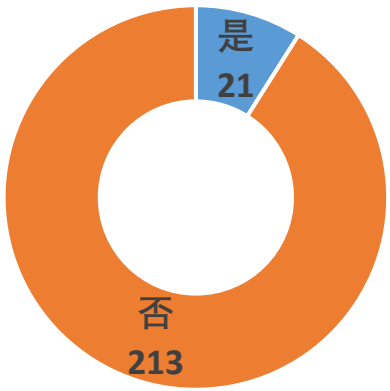
3、从神经网络一般化能力的角度看，应试教育类似于：

a) 欠学习（under-fitting）？ b) 过学习（overfitting）？ c) 好的学习（good fitting）？

否 是 否

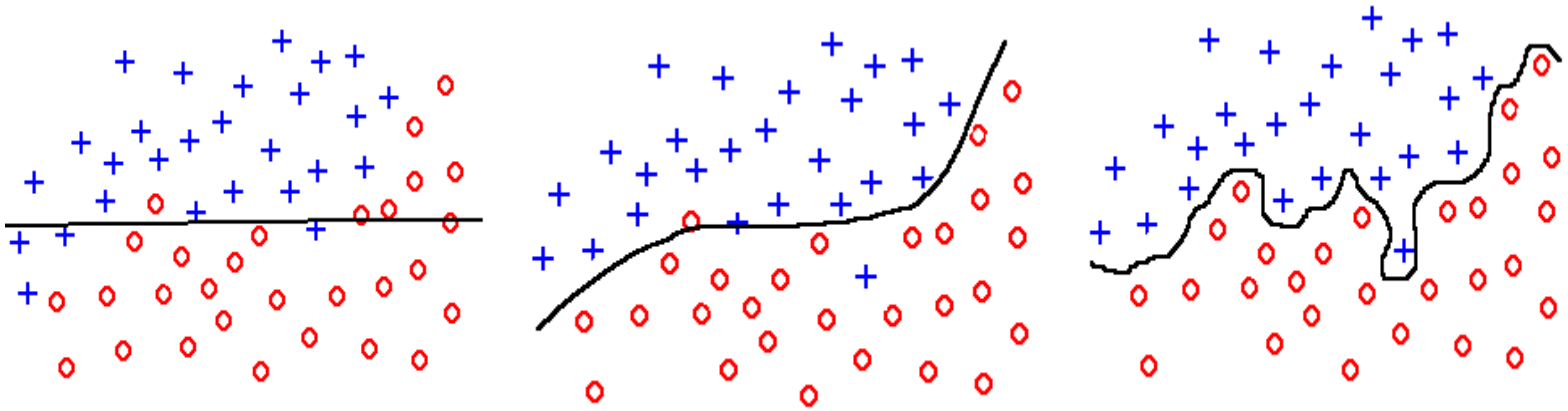






Overfitting and underfitting

- Problem: How rich class of classifications *F(X,W) to use*



underfitting good fit overfitting

- Problem of generalization: A small empirical risk $R_{emp}(W)$ does not imply small true expected risk $R(W)$

Examples of Kernel Functions

- Linear kernel

$$\mathrm{K}(x_i, x_j) = x_i^T x_j$$

- Polynomial kernel

$$\mathrm{K}(x_i, x_j) = (1 + x_i^T x_j)^p$$

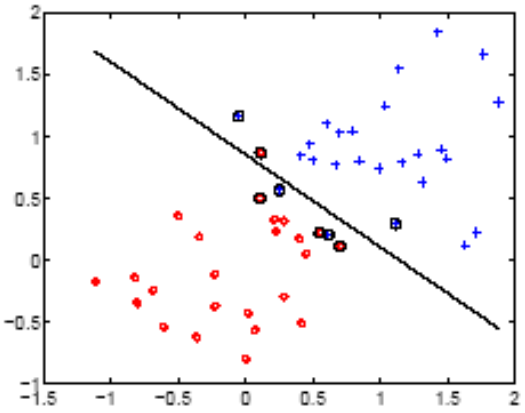
where $p = 2,3, \ldots$. To get the feature vectors we concatenate all *p*-th order polynomial of the components of x (weighted appropriately)

- Radial basis kernel

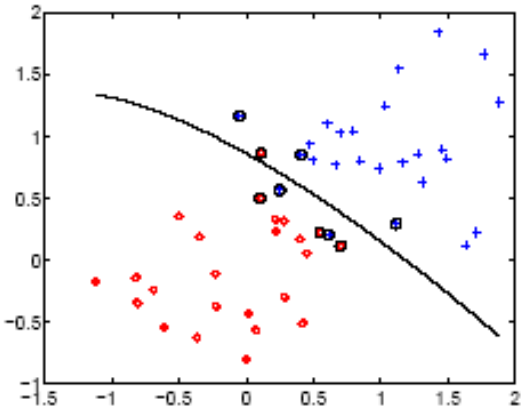
$$\mathrm{K}(x_i, x_j) = \exp(-\frac{1}{2}\|x_i - x_j\|^2)$$

In this case the feature space consists of functions and results in a non-parametric classifier

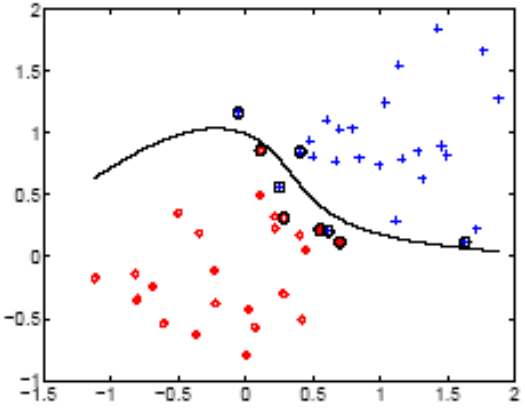
SVM Examples



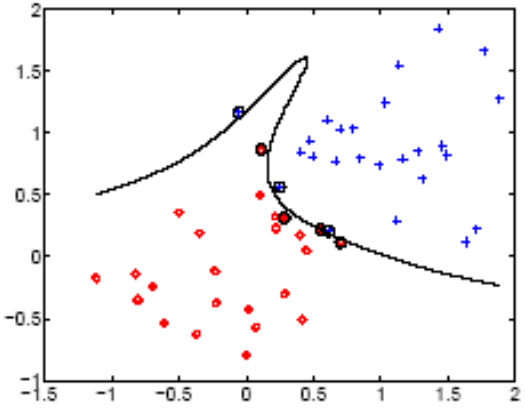
linear



2^{nd} order polynomial



4^{th} order polynomial



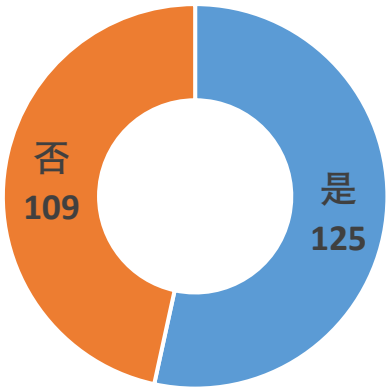
8^{th} order polynomial

第一次Quiz结果

4、用BP算法调节多层感知机的连接权重时，什么场合需要考虑信用赋值问题?

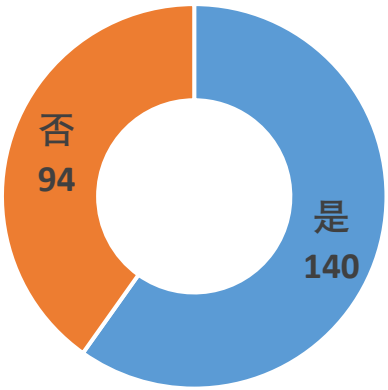
a) 调节输入层到中间层的连接权重？

是



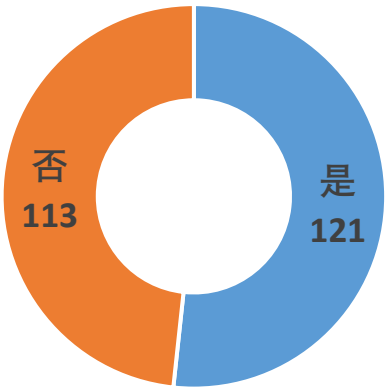
b) 调节中间层到中间层的连接权重？

是



c) 调节中间层到输出层的连接权重？

否



Signal-flow graph of output neuron *k* and hidden neuron *j*

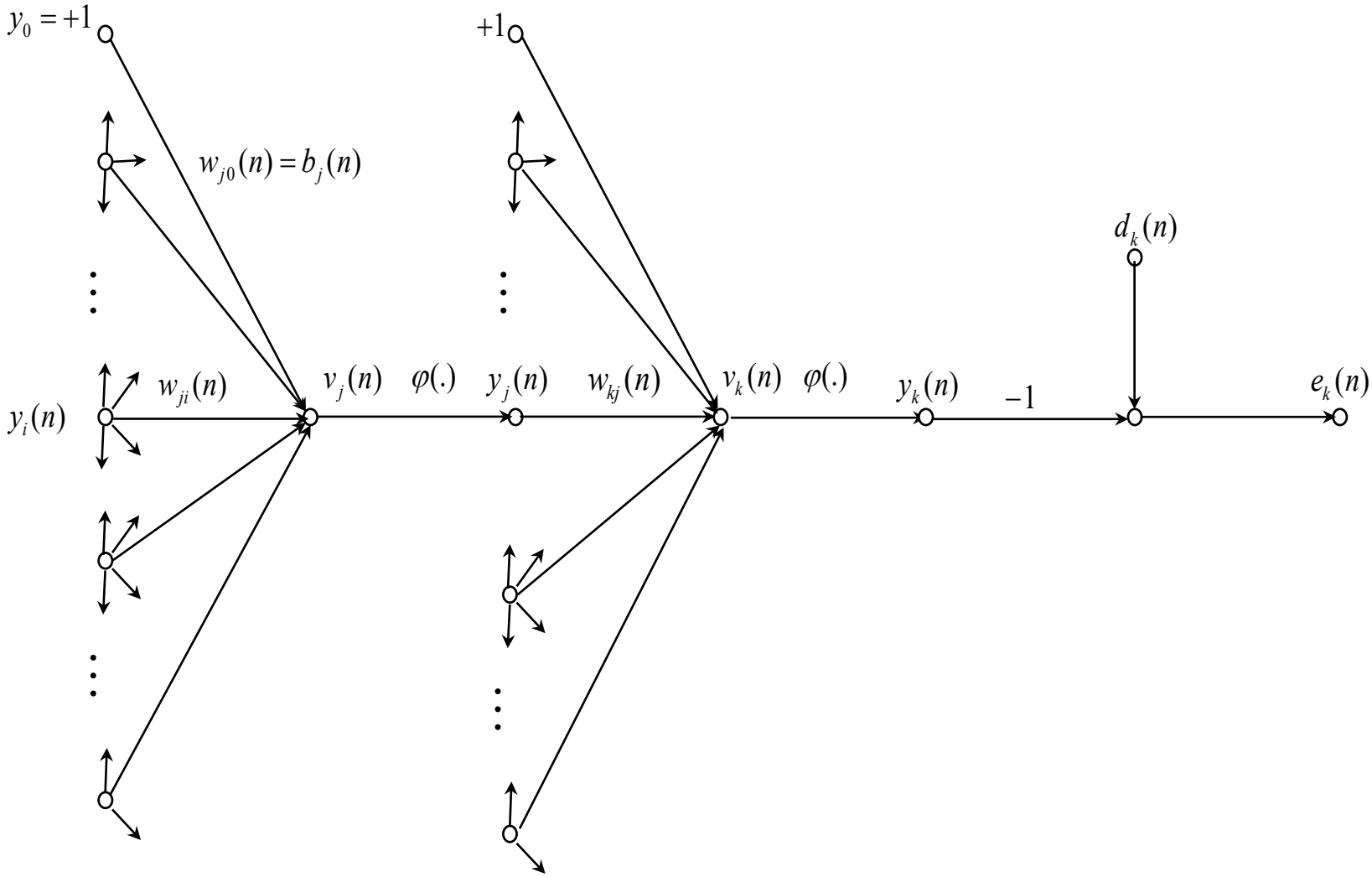


Figure 4.4

Back-Propagation Algorithm -9

Case 1: Neuron j is an Output Node

- Computation of the error $e_j(n)$ is straightforward in this case.
- The desired response $d_j(n)$ for the neuron j is directly available.
- One can use the previous formulas (13) and (14).

Case 2: Neuron j is a Hidden Node

- Now there is no desired response available for neuron j.
- Question: how to compute the responsibility of this neuron for the error made at the output?
- This is the *credit-assignment problem*

Back-Propagation Algorithm -10

- The error signal for a hidden neuron must be determined recursively in terms of the error signals of all neurons connected to it.
- Here the development of the back-propagation algorithm gets complicated.
- Figure 4.4 illustrates the situation where neuron j is a hidden node.
- Using Eq. (14), we may redefine the local gradient $\delta_j(n)$ for hidden neuron j as follows:

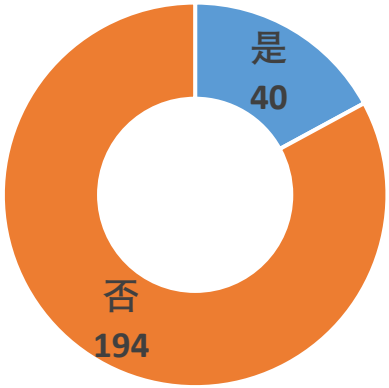
$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \varphi_j'(v_j(n)) \qquad (15)$$

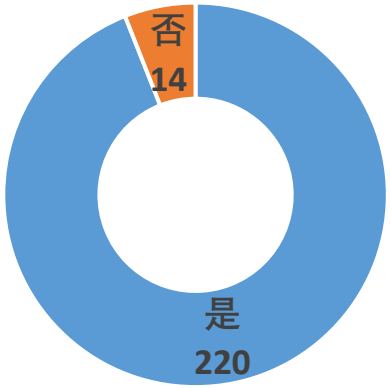
第一次Quiz结果

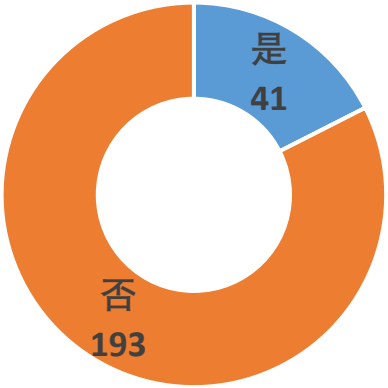
5、三层感知机的一般化能力与下面哪些因素有关?

a）输入层神经元的数目？　b）中间层神经元的数量？　c）调输出层神经元的数量？

否　　是　　否







Sufficient Training Data

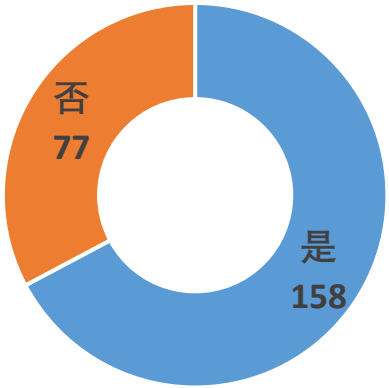
- Three factors affect generalization:
  1. The size and representativeness of the training set.
  2. The architecture of the neural network.
  3. The physical complexity of the problem at hand.
- Only the first two factors can be controlled.
- The issue of generalization may be viewed from two different perpectives:
  - *The architecture of the network is fixed.* Determine the size of the training set for a good generalization.
  - *The size of the training set is fixed.* Determine the best architecture of network for achieving a good generalization.

第一次Quiz结果

6、BP算法训练多层感知机时导致梯度消失的原因是：

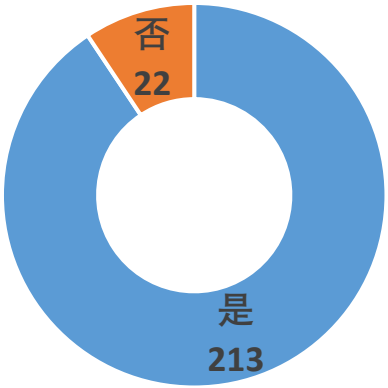
a) 激活函数的非线性特性？

是



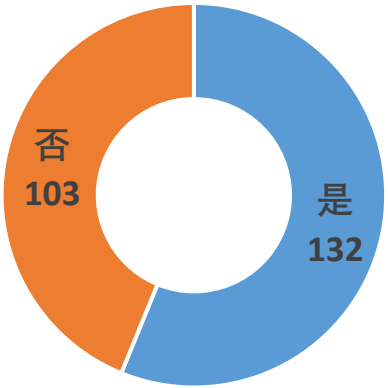
b) 多层感知机中间层的层数？

是

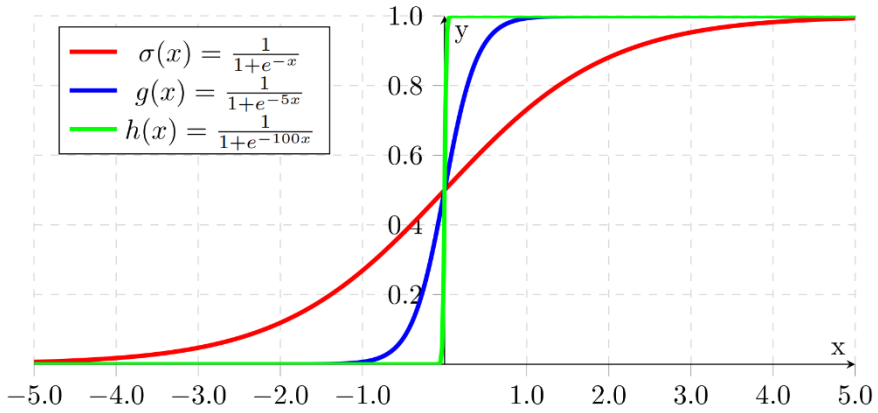


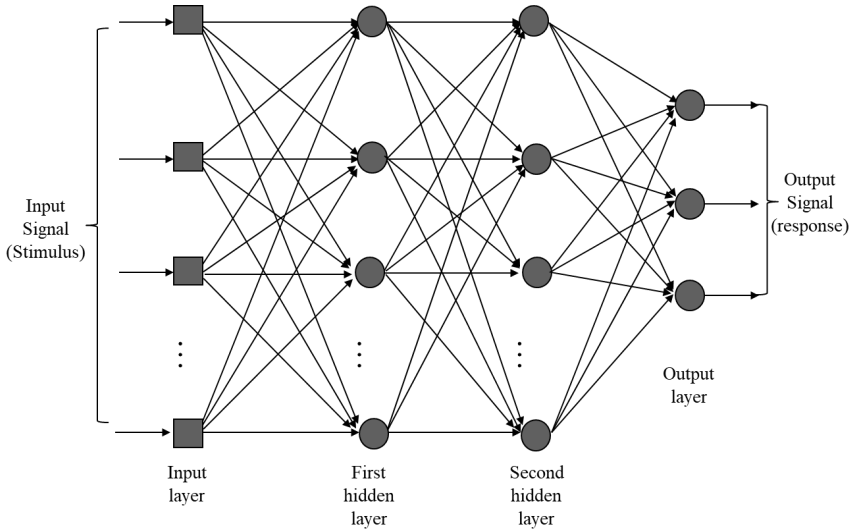
c） 权重的初始化值？

是



Sigmoidal Activation Function





Back-Propagation Algorithm -13

- Inserting these expressions into (3) we get the desired partial derivative

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = -\sum_k e_k(n)\varphi'_k(v_k(n))w_{kj}(n) = -\sum_k \delta_k(n)w_{kj}(n) \qquad (23)$$

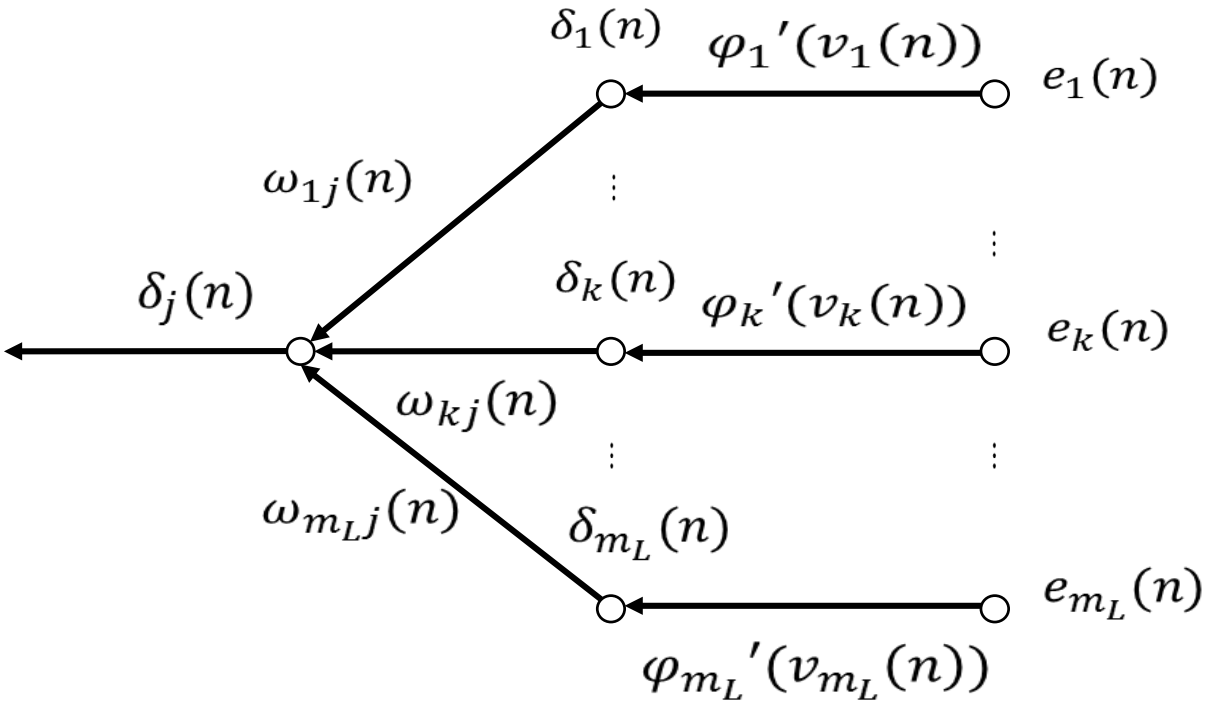
- Here again $\delta_k(n)$ denotes the local gradient for neuron k.

- Finally, inserting (8) into (1) yields the *back-propagation formula* for the local gradient $\delta_j(n)$:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n)w_{kj}(n) \qquad (24)$$

- This holds when neuron j is hidden.

Back-propagation of local gradient



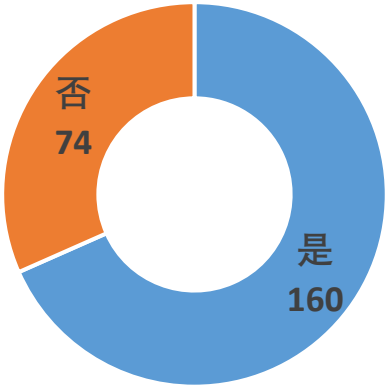
$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

Figure 4.5

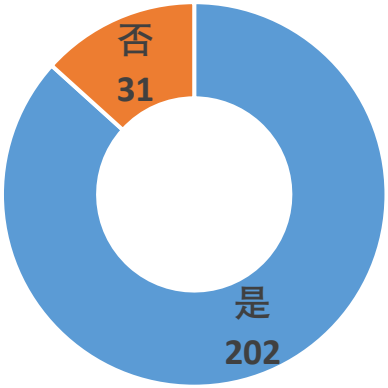
第一次Quiz结果

7、将一个复杂的两类问题分解成更小的两类问题的优点是什么?

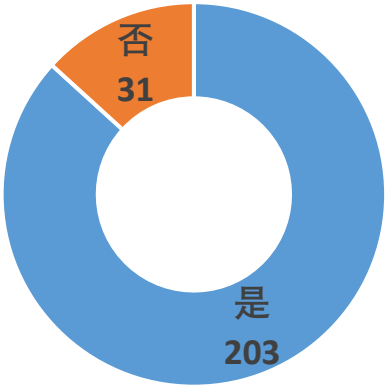
a）可以使用不同的学习算法？ 否



b）可以引入先验知识？ 是



c）可以加快训练速度？ 是



Task Decomposition

- Training data for a K-class problem

$$T = \{ (X_l, Y_l) \}_{l=1}^{L}$$

- Decompose a K-class problem into K(K-1)/2 two-class problems

$$X_i = \left\{ X_l^{(i)} \right\}_{l=1}^{L_i} \quad \text{for } i = 1,2,\ldots,\mathrm{K}$$

$$T_{ij} = \left\{ (X_l^{(i)}, 1-\varepsilon) \right\}_{l=1}^{L_i} \cup \left\{ (X_l^{(j)}, \varepsilon) \right\}_{l=1}^{L_j} \text{ for } i = 1,\ldots K \text{ and } \; j = i+1$$

- Decompose a two-class problem into a number of relatively balanced two-class problems as smaller as needed

$$\text{Partition} \quad \text{of } X_i \text{ into } N_i \text{ subsets} \quad X_{ij} = \left\{ X_l^{(ij)} \right\}_{l=1}^{L_i^{(j)}} \quad \text{for} \quad j = 1,\ldots,N_i$$

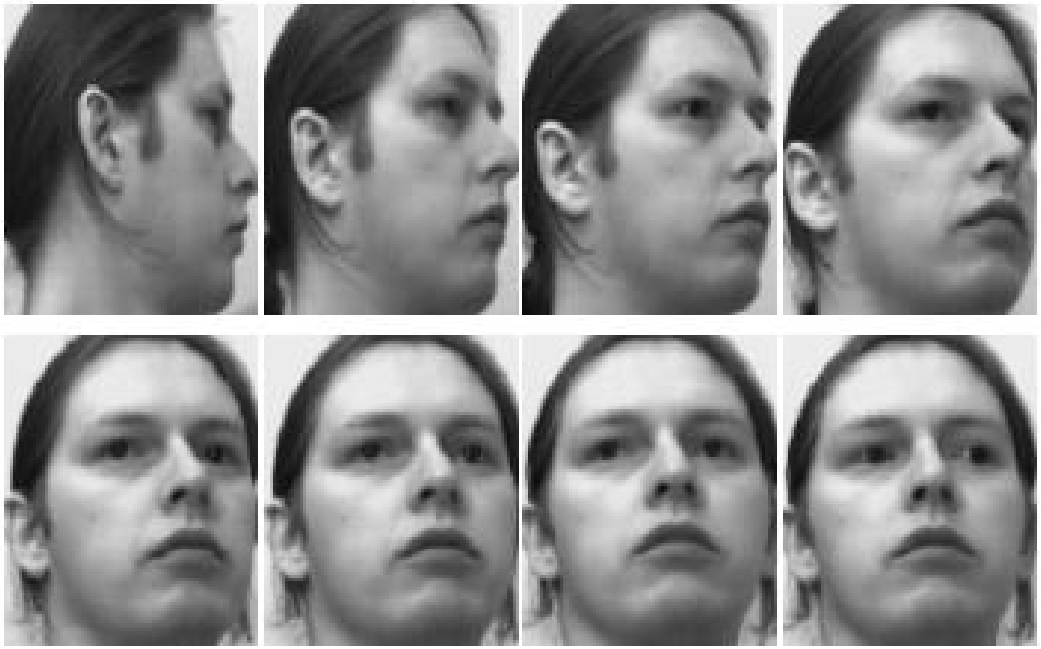
$$T_{ij}^{(u,v)} = \left\{ (X_l^{(iu)}, 1-\varepsilon) \right\}_{l=1}^{L_i^{(u)}} \cup \left\{ (X_l^{(jv)}, \varepsilon) \right\}_{l=1}^{L_j^{(v)}}$$

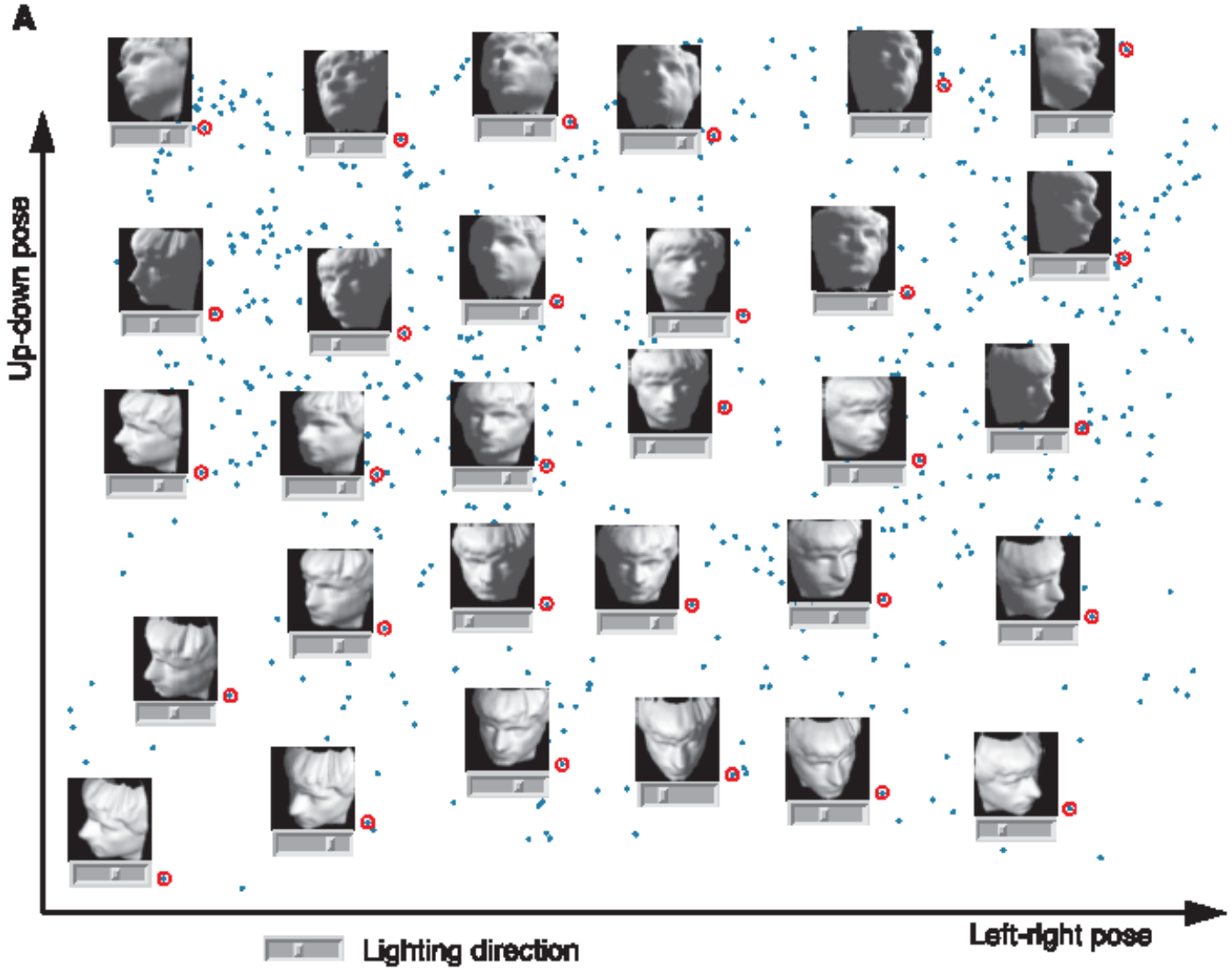
$$\text{for } u = 1,\ldots,N_i, v = 1,\ldots,N_j, \text{ and } \; j \neq i$$

Gender Recognition

(H. C. Lian & B. L. Lu, 2005)

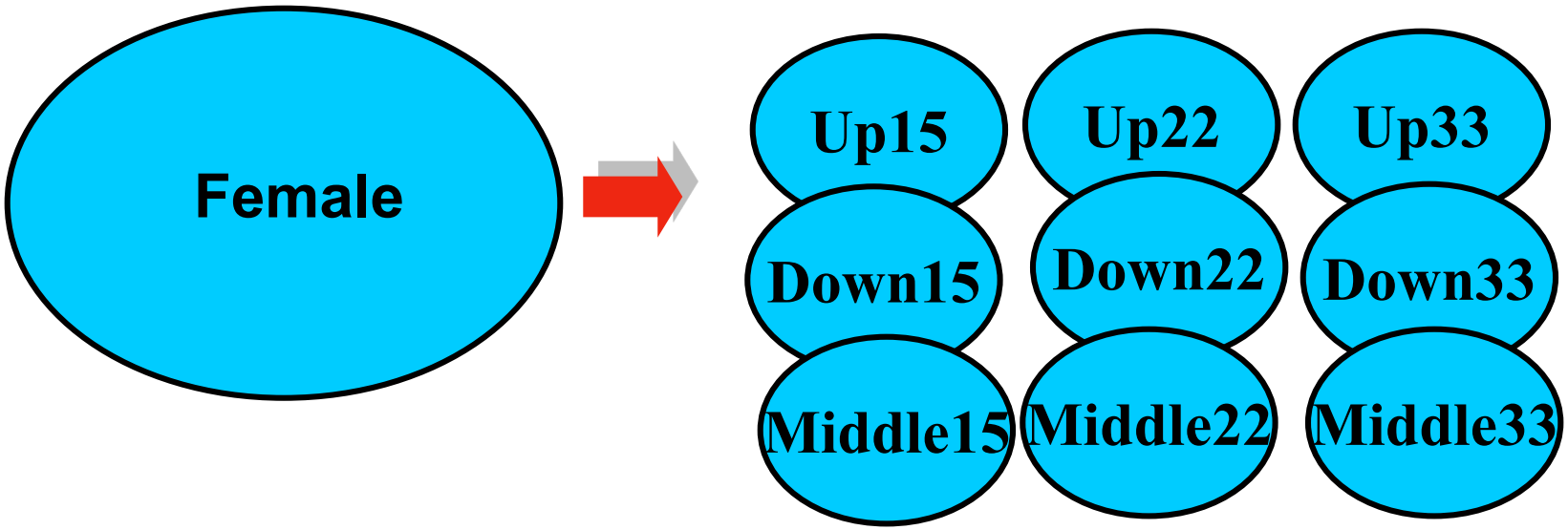
Multi-view Face Recognition



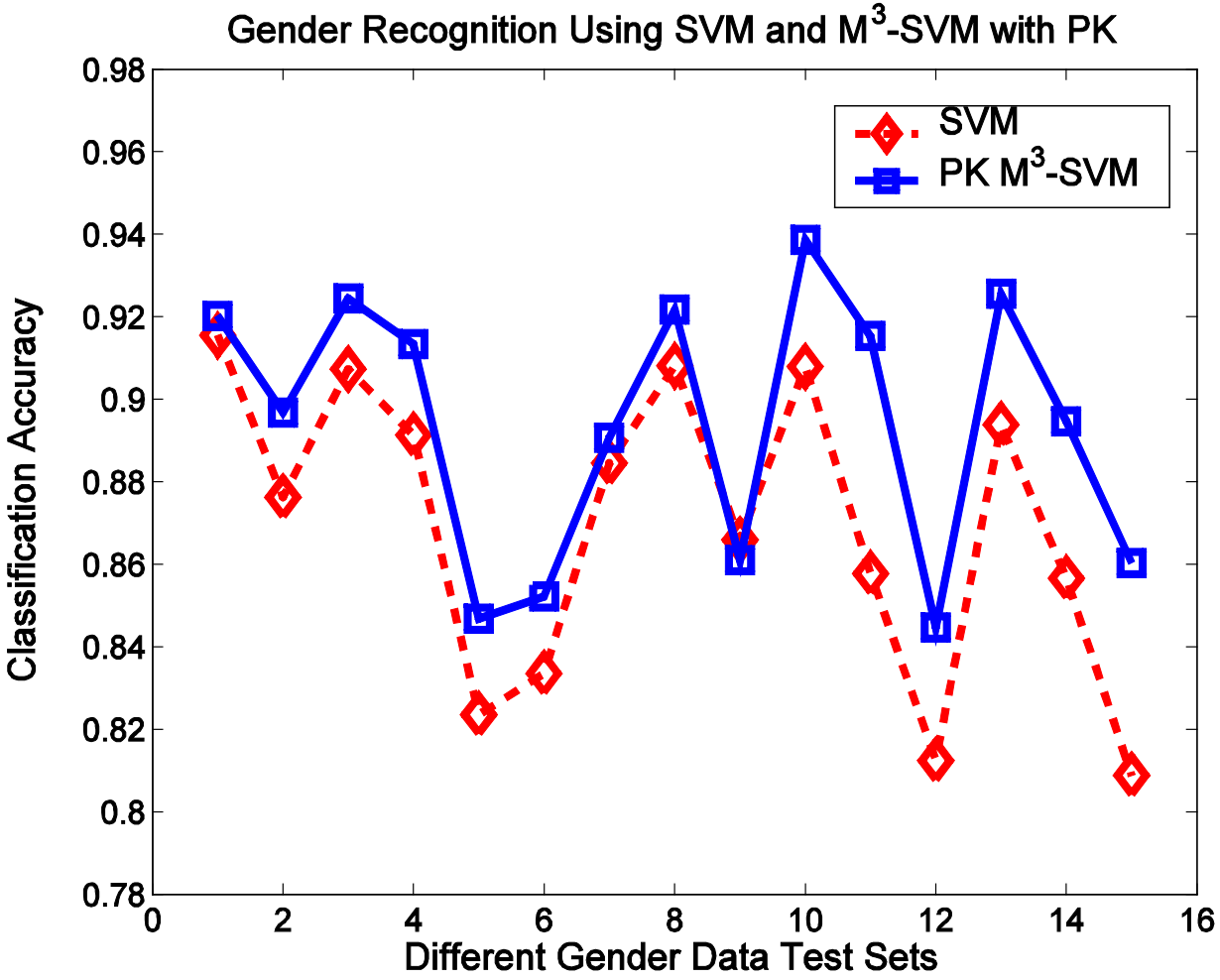

A
Up-down pose
Left-right pose
Lighting direction

Task Decomposition

View information is used for task decomposition



Performance Compaison

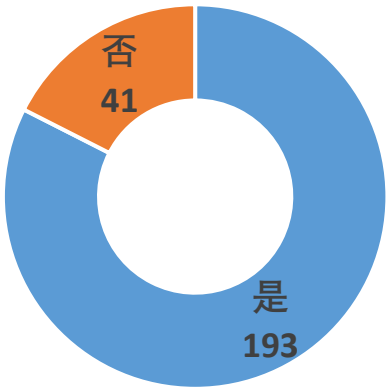


第一次Quiz结果

8、通用逼近定理从理论上告诉我们:

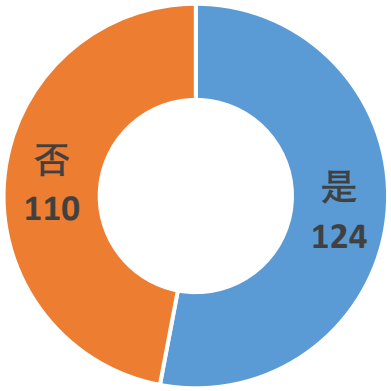
a) 三层感知机具有解决任意线性不可分问题的能力?

是



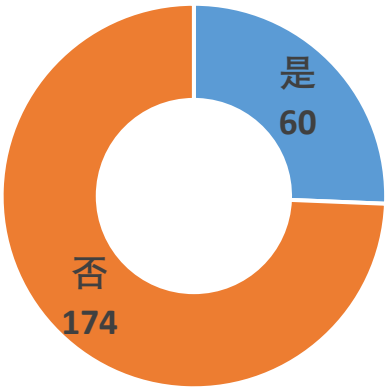
b) 可以用BP算法训练三层感知机解决任意分类问题?

否



c) 可以确保三层感知机具有好的一般化能力?

否



Universal Approximation Theorem -4

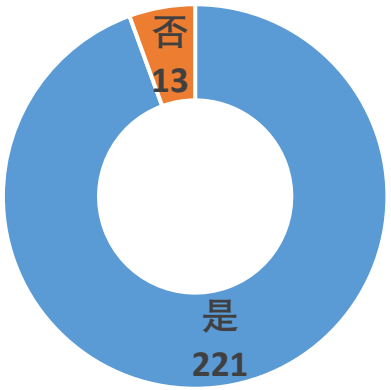
- The universal approximation theorem is an *existence* theorem.
- In effect, the theorem states that *a MLP network with a single hidden layer is sufficient for uniform approximation with accuracy* ε.
- However, the theorem does not say that a single hidden layer is optimal with respect to:
  - learning time
  - ease of implementation
  - generalization ability (most important property).

第一次Quiz结果

9、自组织网络（SOM）与多层感知机的主要区别是:

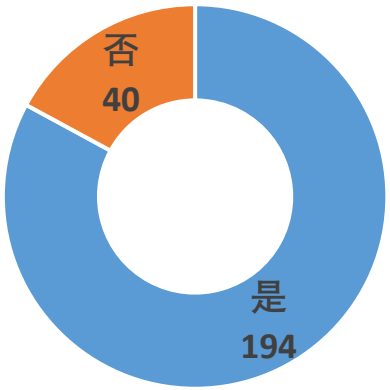
a） 无监督学习？

是



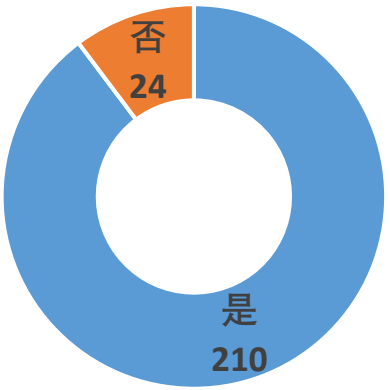
b） 神经元的局部响应特性？

是



c）神经元之间的侧向抑制连接？

是



What is a Self-organizing Map?

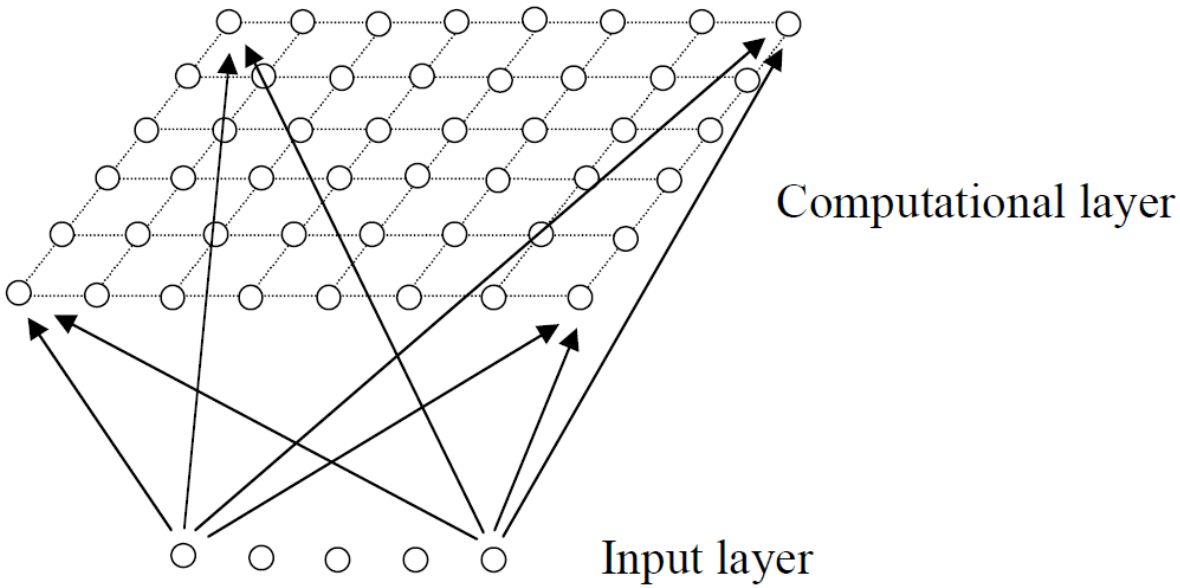
So far we have looked at networks with ***supervised training*** techniques, in which there is a target output for each input pattern, and the network learns to produce the required outputs.

We now turn to ***unsupervised training***, in which the networks learn to form their own classifications of the training data without external help. To do this we have to assume that class membership is broadly defined by the input patterns sharing ***common features***, and that the network will be able to identify those features across the range of input patterns.

One particularly interesting class of unsupervised system is based on ***competitive learning***, in which the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time. This activated neuron is called a ***winner-takes-all neuron*** or simply the ***winning neuron***. Such competition can be induced/implemented by having ***lateral inhibition connections*** (negative feedback paths) between the neurons. The result is that the neurons are forced to organise themselves. For obvious reasons, such a network is called a ***Self Organizing Map (SOM)***.

Kohonen Networks

We shall concentrate on the particular kind of SOM known as a ***Kohonen Network***. This SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer:



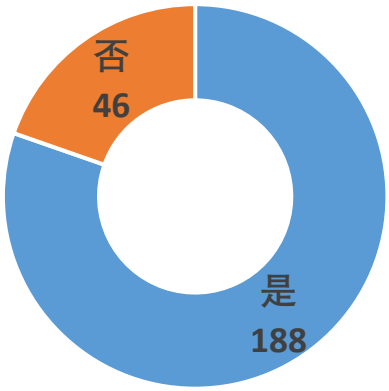
Clearly, a one dimensional map will just have a single row (or a single column) in the computational layer.

第一次Quiz结果

10、多层感知机与径向基函数(RBF)网络的主要区别是：

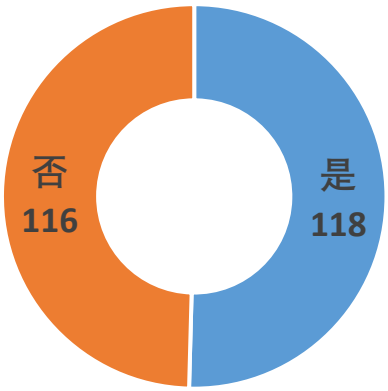
a) 调节网络参数的机制不同？

是



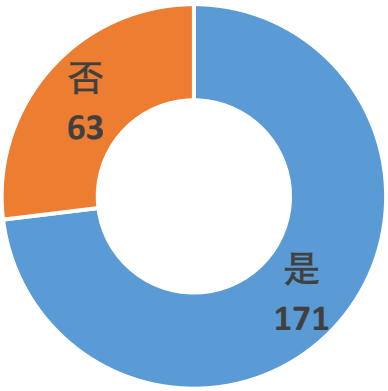
b) 对中间层数目的限制不同？

是



c) 输入层到中间层的权重不同？

是



Introduction to RBF Network

- A basic *radial-basis function (RBF) network* consists of three layers having entirely different roles:

  1. Input layer is made up of source nodes (sensory units).

  2. The hidden layer applies a nonlinear transformation from the input space to the hidden space.

  -RBF networks have only one, often high-dimensional hidden layer.

  3. A linear output layer.

- The hidden space is usually chosen high-dimensional because of two reasons:

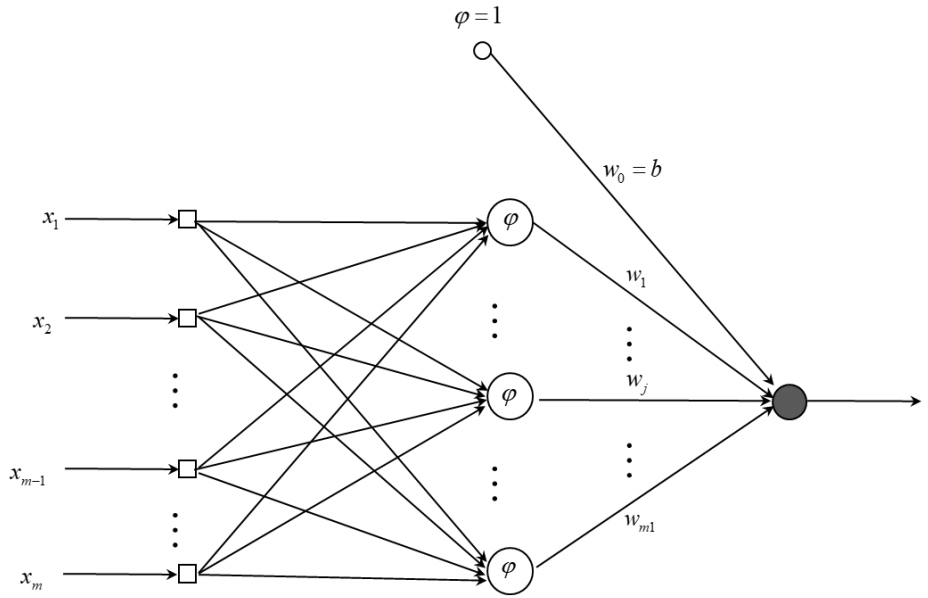
  1. Pattern vectors are more likely to be linearly separable in a high-dimensional space.

  2.The ability of the network is the better the more there are hidden units.

Radial Basis Function

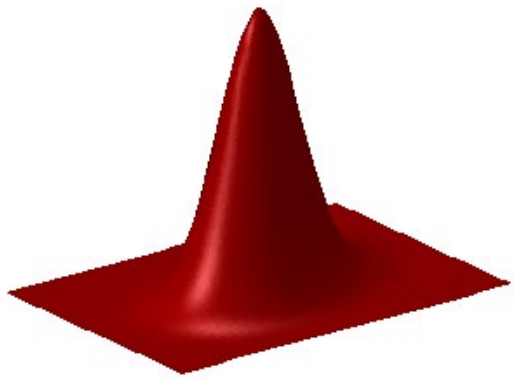
$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x})$$

Three parameters for a radial basis function:



$$\phi_i(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_i\|)$$

- Center: $\mathbf{x}_i$
- Distance Measure: $r = \|\mathbf{x} - \mathbf{x}_i\|$
- Shape: ϕ



Comparison of RBF with MLP

- Both RBF and MLP networks are nonlinear layered networks having universal approximation properties.
- The most important differences between them are:

  1. An RBF network has a single hidden layer, while an MLP can have several hidden layers.

  2. The computational nodes in the MLP network are similar in various layers, while in the RBF network they are quite different in the output and hidden layers.

Comparison of RBF with MLP -2

3. In the RBF network, the output layer is linear, while it is usually nonlinear in an MLP network.

4. In each hidden node, the activation function of RBF network computes an *Euclidean distance*, while in MLP networks an *inner product* between the input and the weight vector is computed.

5. MLPs construct *global* approximations, while RBF networks approximate *locally* nonlinear input-output mappings.

- MLP may require less parameters than the RBF network for achieving the same accuracy.

Outline of Lecture Seven

- Evaluation of Assignment #1
- Transfer learning
- Assignment #2

Notations

Domain:

It consists of two components: A feature space $\mathcal{X}$, a marginal distribution

$$\mathcal{P}(X), \text{where } X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$$

In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

Given a specific domain and label space $\mathcal{Y}$, for each x_i in the domain, to predict its corresponding label y_i, where $y_i \in \mathcal{Y}$

In general, if two tasks are different, then they may have different label spaces or different conditional distributions

$$\mathcal{P}(Y|X), \text{where } Y = \{y_1, ..., y_n\} \text{ and } y_i \in \mathcal{Y}$$

Notations (2)

For simplicity, we only consider at most two domains and two tasks.

Source domain:

$\mathcal{P}(X_S)$, where $X_S = \{x_{S_1}, x_{S_2}, ..., x_{S_{n_S}}\} \in \mathcal{X}_S$

Task in the source domain:

$\mathcal{P}(Y_S|X_S)$, where $Y_S = \{y_{S_1}, y_{S_2}, ..., y_{S_{n_S}}\}$ and $y_{S_i} \in \mathcal{Y}_S$

Target domain:

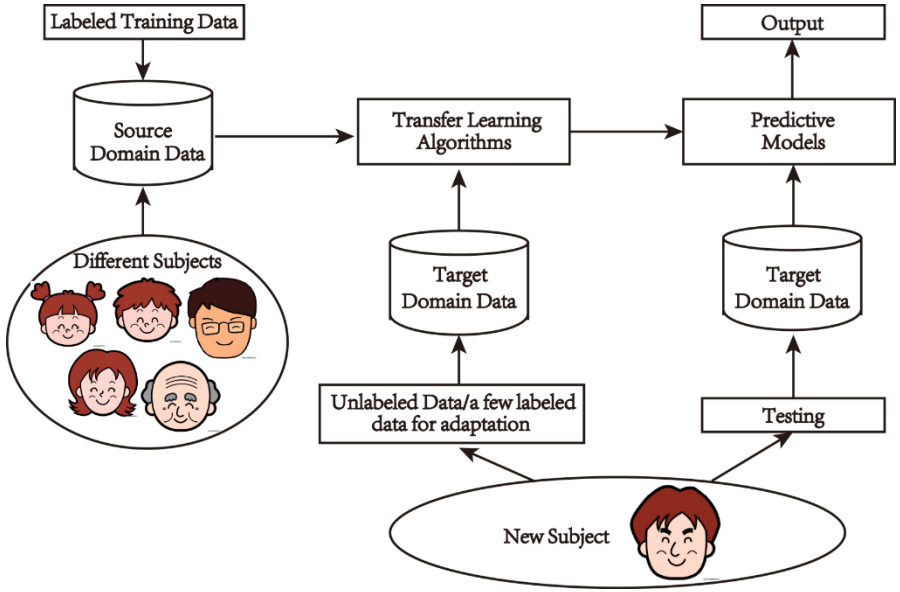
$\mathcal{P}(X_T)$, where $X_T = \{x_{T_1}, x_{T_2}, ..., x_{T_{n_T}}\} \in \mathcal{X}_T$

Task in the target domain

$\mathcal{P}(Y_T|X_T)$, where $Y_T = \{y_{T_1}, y_{T_2}, ..., y_{T_{n_T}}\}$ and $y_{T_i} \in \mathcal{Y}_T$

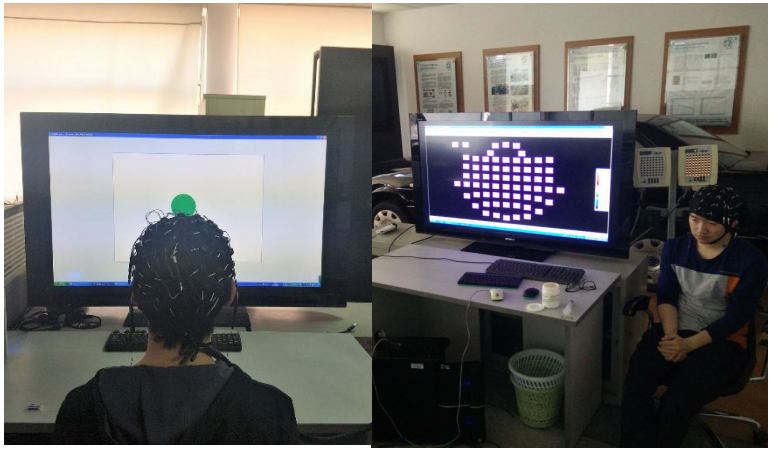
Why Transfer Learning?

- In some domains, labeled data are in short supply.
- In some domains, the calibration effort is very expensive.
- In some domains, the learning process is time consuming.
- How to extract knowledge learnt from related domains to help learning in a target domain with a few labeled data?
- How to extract knowledge learnt from related domains to speed up learning in a target domain?



Motivation

- The data collection and annotation in real scenarios are usually costly and difficult
- Transferring knowledge from laboratory to real scenario
  - Laboratory: Students with controlled sleep time
  - Real Scenario: CRH railway drivers before work
- Cross-scenario & Cross-subject tasks



Modeling
Predicting

In the Laboratory

- We take 4-hour sleep, 6-hour sleep and 8-hour sleep as poor, normal and good in terms of the sleep quality in our study
- The sleep time and wake up time for three experiments are 3:00-7:00, 1:00-7:00 and 23:00-7:00, respectively.
- Total sleep time and deep sleep time of all subjects are recorded by smart bands
- High quality EEG acquisition: 62-channel wet electrode cap
- Different state for same subjects: 3 times, 30 min each experiments

4h
6h
8h

In the Real Scenario (CRH)

- To keep the drivers' daily routine, the experiment settings have to be adjusted to meet the need of real-scenario application.
  - No control on sleep time: range of 2~10 h
  - Easy to set up devices: 18-channel dry electrode cap (DSI-24)
  - One experiment for each subjects: 1 time, 6 min each

Sleep:
2~10h

(CRH: China Railway High-speed)

Summary of Domain Differences

Categories	Specifications	Laboratory	Real-scenario
Subjects	Age	21-26	25-49
	Sex-distribution	4 males, 6females	70 males
	Occupation	Students	High-speed train drivers
	Experiments per subject	3	1
Controlled conditions	Sleep time	4,6,8 hours	Not controlled
	Monitoring sleep	Yes	No
	Head Cleaning	Yes	No
	Experiment Starts	$\leq$1 hour after wake up	1-10 hour after wake up
Experiment settings	Experiment duration	30 min	6 min
	Task	Eyes open	Eyes open and close
	Environment	Quite and isolated	Noisy
Devices	Electrods type	Wet electrodes	Dry electrodes
	Resolution	62 channels	18 channels
	Reference	REF(Between CPZ and PZ)	A1 and A2 (On the ears)
	Common Mode Follower	0	1 (CMF)
	Sample rate	1000 Hz	300 Hz

Settings of Transfer Learning

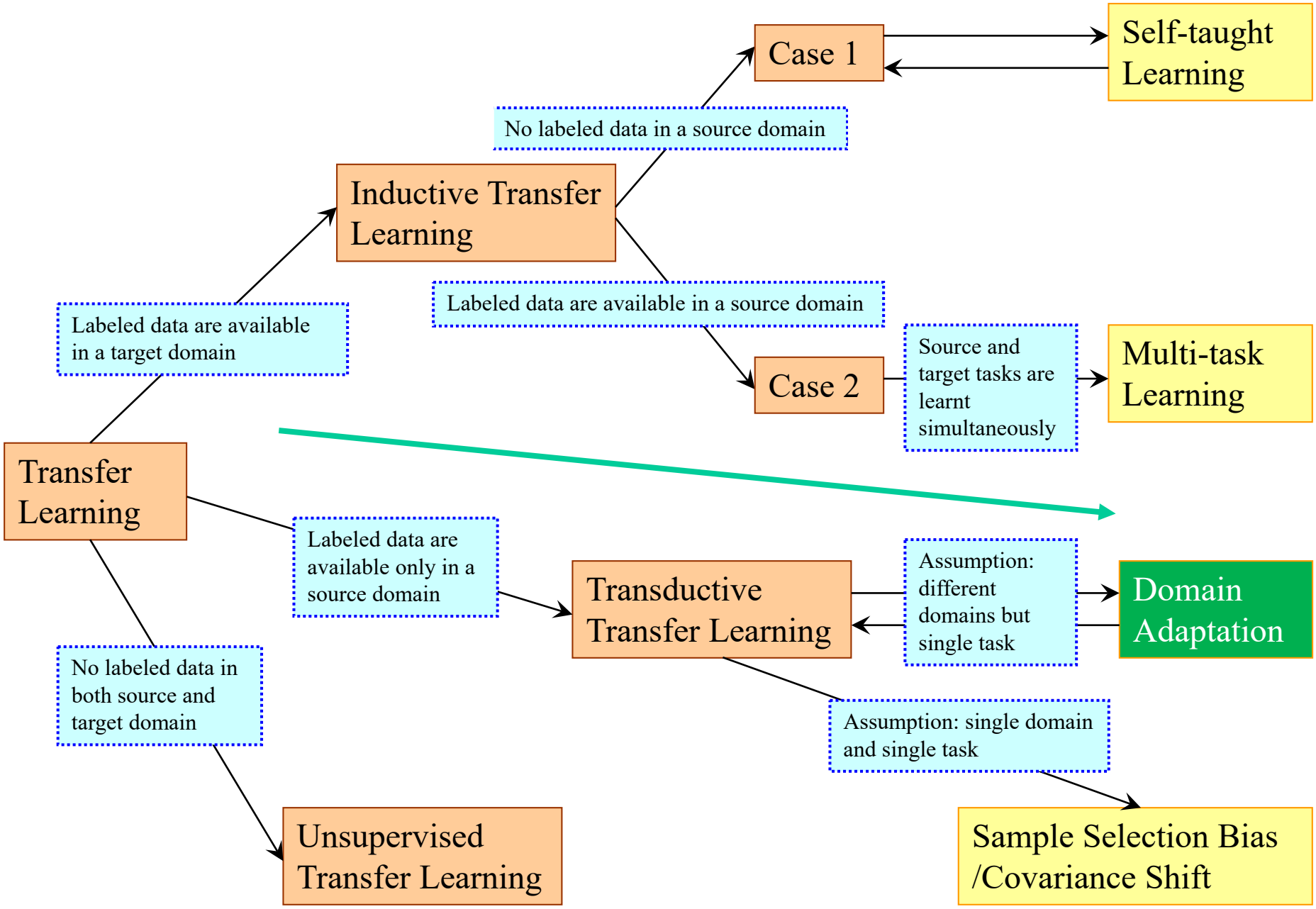
Relationship between Traditional Machine Learning and Various Transfer Learning Settings

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	*Inductive Transfer Learning* / *Unsupervised Transfer Learning*	the same	different but related
		different but related	different but related
	Transductive Transfer Learning	different but related	the same

Different Settings of Transfer Learning

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Pan S J, Yang Q. A survey on transfer learning , IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.



Pan S J, Yang Q. A survey on transfer learning [J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.

Three Research Issues

- "What to transfer?"
  - Which part of knowledge can be transferred across domain? e.g. feature representation, parameter settings, latent feature distribution, etc.
- "How to transfer?"
  - Specific learning algorithms to transfer the knowledge.
- "When to transfer?"
  - Asks in which situations, transferring skills should be done. Likewise, we are interested in knowing in which situations, knowledge should not be transferred.

Approaches to Transfer Learning

Transfer learning approaches	Description
Instance-transfer	*To re-weight some labeled data in a source domain for use in the target domain*
Feature-representation-transfer	Find a "good" feature representation that reduces difference between a source and a target domain or minimizes error of models
Model-transfer	Discover shared parameters or priors of models between a source domain and a target domain
Relational-knowledge-transfer	Build mapping of relational knowledge between a source domain and a target domain.

Approaches to Transfer Learning

归纳式迁移学习　　直推式迁移学习

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance-transfer	√	√	
Feature-representation-transfer	√	√	√
Model-transfer	√		
Relational-knowledge-transfer	√		

Inductive Transfer Learning

Instance-transfer Approaches

- Assumption: **the source domain and target domain data use exactly the same features and labels.**
- Motivation: **Although the source domain data can not be reused directly, there are some parts of the data that can still be reused by re-weighting.**
- Main Idea: **Discriminatively adjust weighs of data in the source domain for use in the target domain.**

Feature Representation Transfer Approach

- Assumption: If *t* tasks are related to each other, then they may share some common features which can benefit for all tasks.
- Input: *t* tasks, each of them has its own training data.
- Output: Common features learnt across *t* tasks and *t* models for *t* tasks, respectively.

Relational knowledge transfer Approach

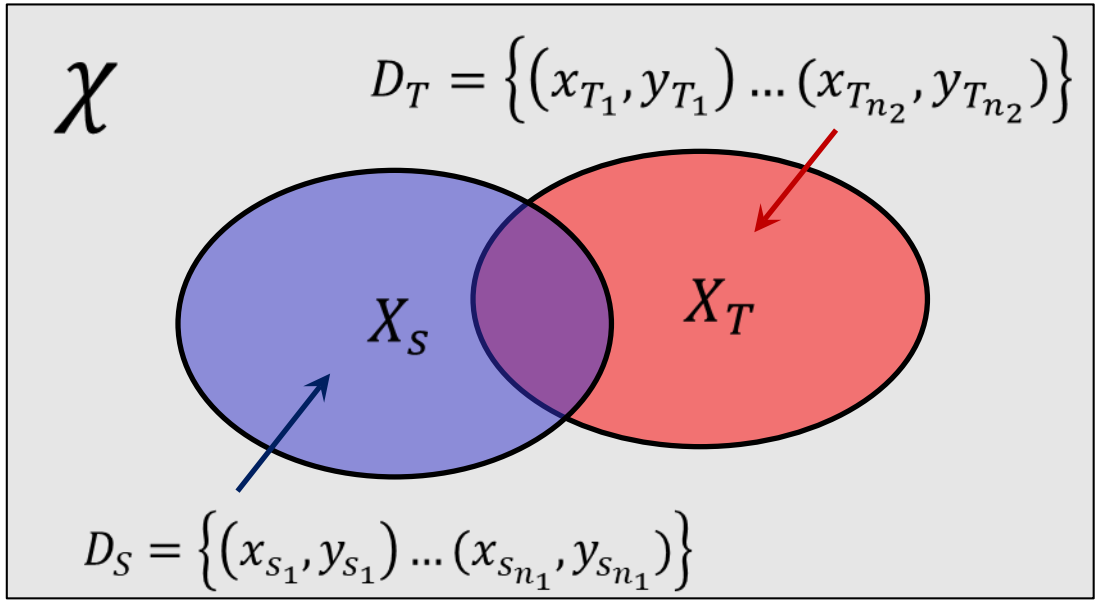
- Assumption: **If the target domain and source domain are related, then there may be some relationship between domains being similar, which can be used for transfer learning.**
- Input:
  - **Relational data in the source domain and a statistical relational model, Markov Logic Network (MLN), which has been learnt in the source domain.**
  - **Relational data in the target domain.**
- Output: **A new statistical relational model, MLN, in the target domain.**
- Goal: **To learn a MLN in the target domain more efficiently and effectively.**

Negative Transfer

- Most approaches to transfer learning assume transferring knowledge across domains be always positive.
- However, in some cases, when two tasks are too dissimilar, brute-force transfer may even hurt the performance of the target task, which is called **negative transfer**.
- Some researchers have studied how to measure relatedness among tasks.
- How to design a mechanism to avoid negative transfer needs to be studied theoretically.

Transfer Component Analysis

Let the source and target domain D_S and D_T be in the same space χ, but X_S and X_T have different marginal distributions $P(X_S) \neq Q(X_T)$ (or $P \neq Q$ for short).
The key assumption: $P(Y_S|X_S) = P(Y_T|X_T)$.



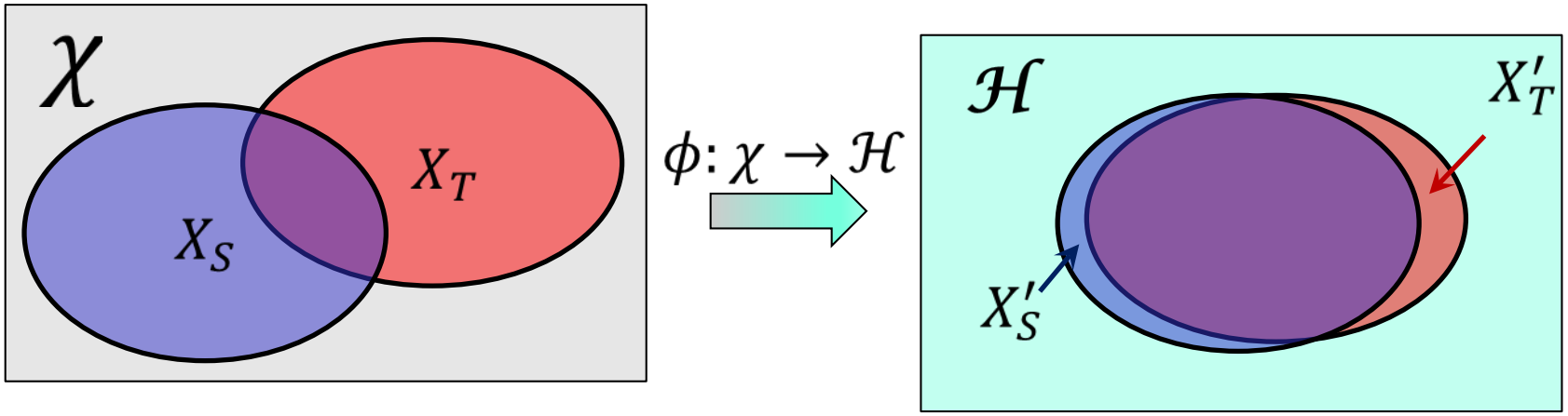
Pan, Sinno Jialin, et al. Domain adaptation via transfer component analysis, IEEE Transactions on Neural Networks 22.2 (2011): 199-210.

Transfer Component Analysis

To find a common latent representation for both X_S and X_T, a nonlinear transformation $\phi: \chi \to \mathcal{H}$ ($\mathcal{H}$ is an RKHS) is introduced.

We desire that $\mathrm{P}'(\mathrm{X}'_S) \approx Q'(X'_T)$, which is complemented by minimizing MMD (*Maximum Mean Discrepancy*): 最大均值差异

$$\mathrm{Dist}(X'_S, X'_T) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_{S_i}) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(x_{T_i}) \right\|_{\mathcal{H}}^2$$



（RKHS：再生核希尔伯特空间）

Transfer Component Analysis

By kernel trick: $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, the distance can be written as:

$$Dist(X'_S, X'_T) = tr(KL),$$

Where

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix}$$

K is a $(n_1 + n_2) \times (n_1 + n_2)$ kernel matrix, the subscripts represents where the x_i, x_j come from in $k(x_i, x_j)$.

Matrix K can be decomposed as $\mathrm{K} = (KK^{-1/2})(K^{-1/2}K)$. Use a $(n_1 + n_2) \times m$ matrix $\widetilde{W}$ to transform the feature vectors in K's decomposed components

$$\widetilde{K} = (KK^{-1/2}\widetilde{W})(\widetilde{W}^T K^{-1/2} K) = KWW^T K$$

Where $W = K^{-1/2}\widetilde{W}$.

Transfer Component Analysis

Using the definition of $\widetilde{K}$, we have

$$Dist(X_S', X_S') = tr\left(\left(KWW^TK\right)L\right) = tr(KWLW^TK)$$

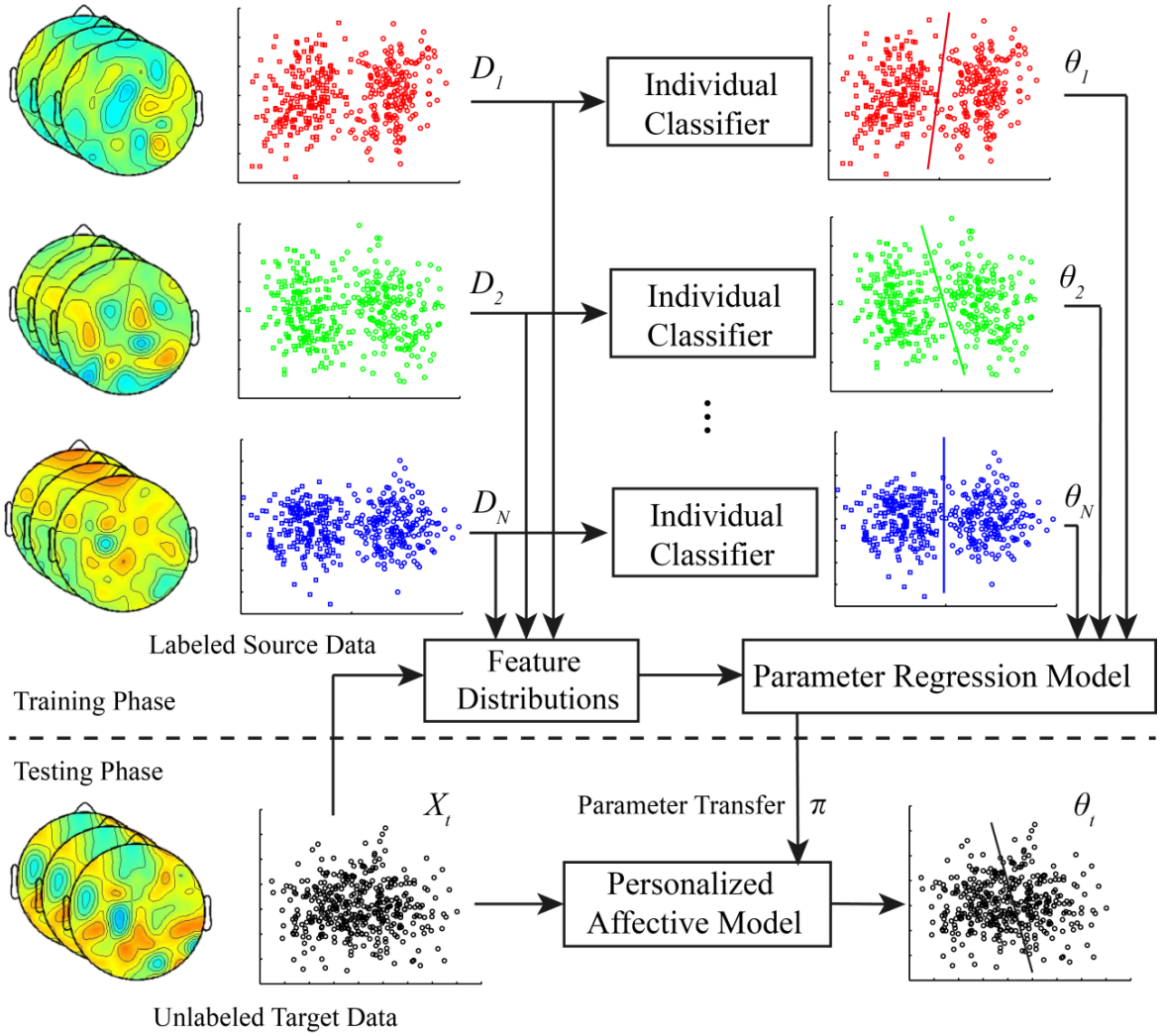
The kernel learning problem is then reduced to

$$\min_{W} \ \mu\, tr\left(KWLW^TK\right) + tr(W^TW)$$
$$s.t. \quad W^TKHKW = I$$

Where $tr\left(W^TW\right)$ is a regularization term . Restriction term is added to avoid the trivial solution ($W = 0$), where H is the centering matrix.

The solution to W is then the eigenvectors corresponding to the ***m* leading eigenvalues** of $(I + \mu KLK)^{-1}$KHK.

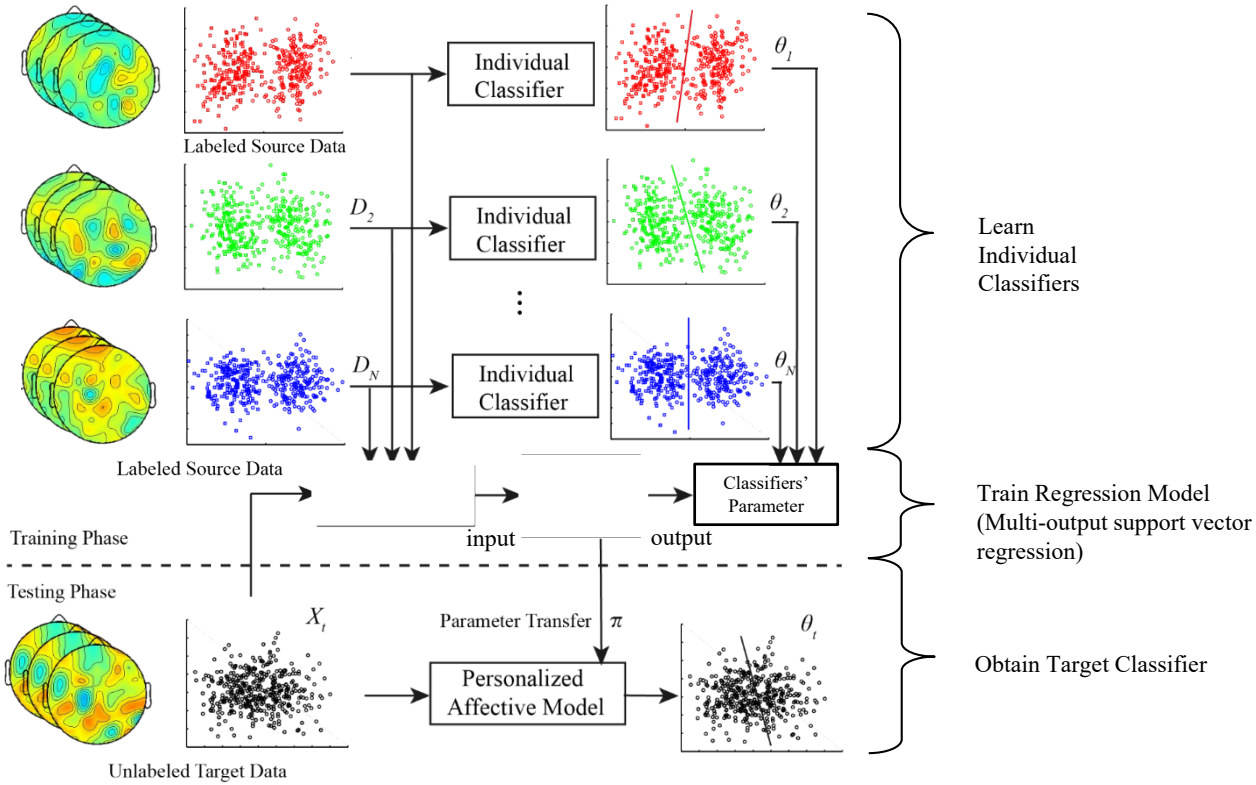
Transductive Parameter Transfer



直推参数迁移

Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM International Conference on Multimedia, pages 357–366. ACM, 2014.

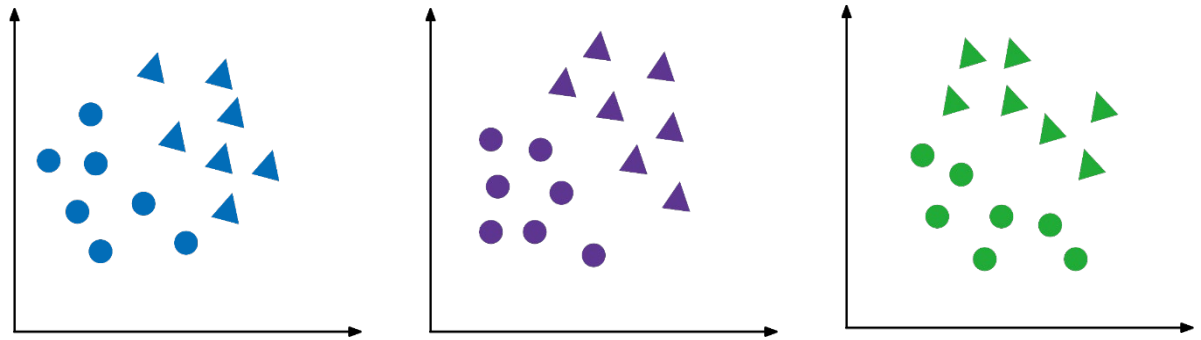
Transductive Parameter Transfer



Transductive Parameter Transfer

An introductory example

Suppose we have labeled data sets from three domains (source domains) and an unlabeled target domain data set.



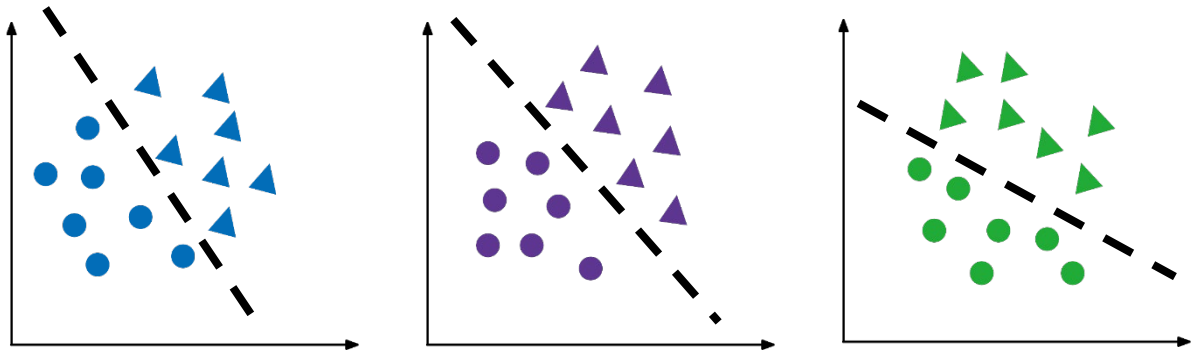
E. Sangineto, G. Zen, E. Ricci, and N. Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM Multimedia, 2014.

Transductive Parameter Transfer

An introductory example

Suppose we have labeled data sets from three domains (source domains) and an unlabeled target domain data set.

For the source domains, we can easily train SVMs on the labeled data.



E. Sangineto, G. Zen, E. Ricci, and N. Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM Multimedia, 2014.

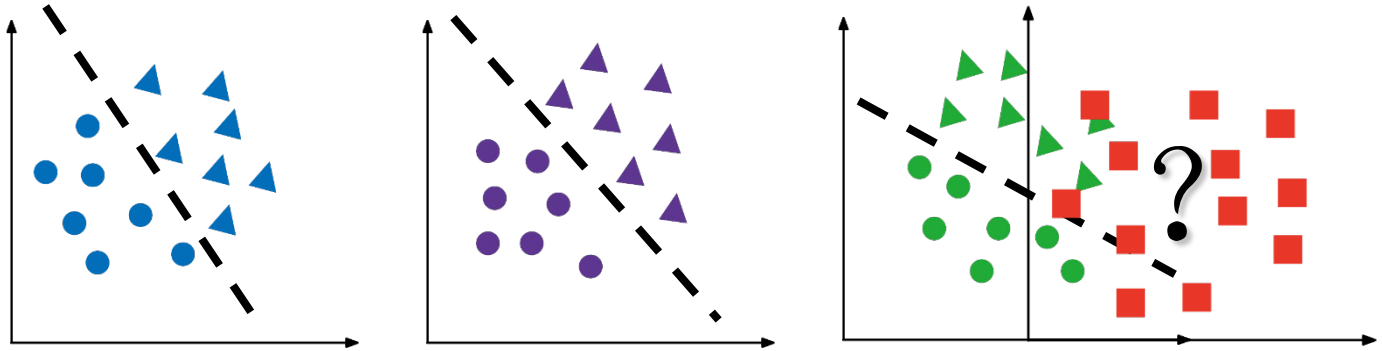
Transductive Parameter Transfer

An introductory example

Suppose we have labeled data sets from three domains (source domains) and an unlabeled target domain data set.

For the source domains, we can easily train SVMs on the labeled data.

Now the problem is how can we transfer the parameters in the trained SVMs to the target domain data set.



E. Sangineto, G. Zen, E. Ricci, and N. Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM Multimedia, 2014.

Transductive Parameter Transfer

Demonstration of the objective

We denote the source domain data sets as

$$D_1^s, \ldots, D_N^s, D_i^s = \{\mathbf{x}_j^s, y_j^s\}_{j=1}^{n_i^s}$$

Where D_i^s indicates the *i*th data set. Also let us use X_i^s to indicate the data with label. Then we assume that X_i^s is drawn from the marginal distribution $P_{X_i^s}$: $X_i^s \sim P_{X_i^s}$.

For each data set, we can easily train an SVM parameterized with $\boldsymbol{\theta}_i^s$.

The question is: for target domain data $X^t \sim P_{X^t}$, how can we obtain the corresponding SVM parameter $\boldsymbol{\theta}^t$.

Transductive Parameter Transfer

Analysis

By observing the problem setting, we assume there is a function from marginal distribution to the SVM parameter vector:

$$f : \mathcal{P} \rightarrow \Theta$$

where $\mathcal{P}$ and Θ indicate the space of the marginal distribution and the SVM parameter vectors respectively.

So we have

$$\boldsymbol{\theta}_i^s = f(P_{X_i^s})$$

If the function can be obtained, the target domain SVM parameter vector is simply

$$\boldsymbol{\theta}^t = f(P_{X^s}).$$

Transductive Parameter Transfer

Analysis (Cont.)

However, the marginal distributions $P_{X_i^s}$ can not be easily depicted.

Because we know $X_i^s \sim P_{X_i^s}$, we can use X_i^s instead of $P_{X_i^s}$.

We then assume another function:

$$\hat{f} : 2^{\mathcal{X}} \rightarrow \Theta.$$

$\hat{f}$ is a function from the feature set space $2^{\mathcal{X}}$ to the parameter space.

Similarly, we have $\boldsymbol{\theta}_i^s = \hat{f}(X_i^s)$ and $\boldsymbol{\theta}^t = \hat{f}(X^t)$.

Now the problem becomes how to learn the function $\hat{f}$.

Transductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR

The training set is $\mathcal{T} = \{X_i^s, \boldsymbol{\theta}_i^s\}_{i=1}^N$.

It is a simple multivariate regression problem. Various methods can be used to solve it. In the original paper, M-SVR (Tuia et al.) was applied.

D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, andG. Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameterestimation. IEEE Geoscience and Remote SensingLetters, 8(4):804–808, July 2011.

Tranductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR Cont.

In the M-SVR framework, $\hat{f}$ can be defined by

$$\hat{f}(X) = \phi(X)'B + \boldsymbol{c}'$$

where B is a matrix and $\boldsymbol{c}$ is a vector.$\phi(X)$ is a nonlinear mapping from the matrix space to a vector space.

The corresponding loss is defined as

$$L(B, \boldsymbol{c}; \mathcal{T}) = \frac{1}{2}||B||_F^2 + \lambda_E \sum_{i=1}^{N} E(||\boldsymbol{\theta}_i^{s\prime} - \hat{f}(X_i^s)||)$$

where $||\cdot||_F$ indicates the Frobenius norm: $||A||_F = \sqrt{\text{trace}(A'A)}$, the λ_E is a tradeoff hyperparameter, and $E(\cdot)$ is a loss function parameterized by ϵ :

$$E(u) = \begin{cases} 0 & u < \epsilon \\ u^2 - 2u\epsilon + \epsilon^2 & u \geq \epsilon \end{cases}$$

Tranductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR Cont.

By applying the kernel trick, the prediction function can be rewritten as

$$\hat{f}(X) = [\kappa(X_1^s, X), \ldots, \kappa(X_N^s, X)] \cdot V + \boldsymbol{c}'$$

where V is obtained by solving the dual problem.

To optimize V and $\boldsymbol{c}$, we adopt iterative procedure. In iteration k, suppose we have the temporal values of B and $\boldsymbol{c}$ being B^k and $\boldsymbol{c}^k$. Let us denote the error $\boldsymbol{e} = \hat{f}(X)' - \boldsymbol{\theta}$, then the loss function can be expand with Taylor expansion:

$$L^*(B, \boldsymbol{c}) = \frac{1}{2}||B||_F^2 + \lambda_E(\sum_{i=1}^{N} E(e_i^k) + \frac{dE(u)}{du}\Big|_{u_i^k} \frac{(e_i^k)'}{u_i^k}[e_i - e_i^k])$$

where $u = ||e||$. The superscripts ‘k’s indicate the values at the kth iteration.

Transductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR Cont.

Then we can further approximate the loss function by using a quadratic approximation:

$$L^{**}(B, \boldsymbol{c}) = \frac{1}{2}||B||_F^2 + \lambda_E \frac{1}{2} \sum_{i=1}^{N} a_i u_i^2 + \tau$$

where

$$a_i = \frac{\lambda_E}{u_i^k} \frac{dE(u)}{du}\bigg|_{u_i^k}$$

and τ is a value that does not depend on B and $\boldsymbol{c}$.

It can be easily observed that the $L^{**}(\cdot)$ is simply a quadratic function of B and c, so the solution B^{sol} and $\boldsymbol{c}^{sol}$ can be obtained.

Transductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR Cont.

The update rule of B and $\boldsymbol{c}$ is

$$\begin{bmatrix} B^{k+1} \\ (\boldsymbol{c}^{k+1})' \end{bmatrix} = \begin{bmatrix} B^{k} \\ (\boldsymbol{c}^{k})' \end{bmatrix} + \eta_k \begin{bmatrix} B^{sol} - B^{k} \\ (\boldsymbol{c}^{sol})' - (\boldsymbol{c}^{k})' \end{bmatrix}$$

where the step size η_k is decided by back tracking algorithm. Specifically, we initially set it to 1, and check whether the loss function drops. If not, the step size is multiplied by a value less than 1 and retry.

Transductive Parameter Transfer

Learning problem of $\hat{f}$ with M-SVR Cont.

Solving B is not convenient because map function $\phi(X)$ is usually not analytical, so we substitute V into the previous equations by utilizing $B_{\cdot,j} = KV_{\cdot,j}$ where K is the kernel matrix defined by

$$K_{ij} = \kappa(X_i, X_j)$$

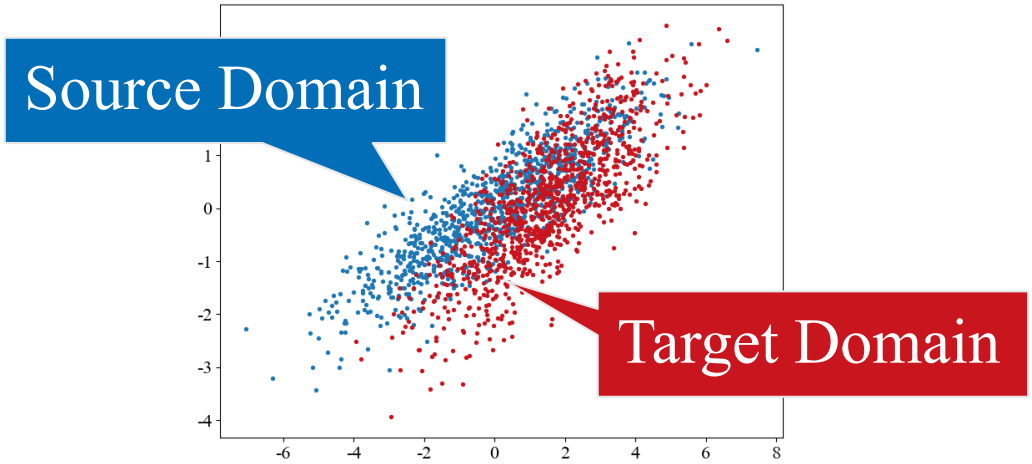
Tansductive Parameter Transfer

Some points

1.Other types of parameterized models (e.g., LDA) should also work well with TPT.
2.When learning , the features are matrices. So some kind of mapping should be applied to project the features to vector spaces. For M-SVR, the mapping corresponds to specially designed kernels.
3.The regression method used in TPT is also free to choose. But the authors found M-SVR performed better on their tasks.

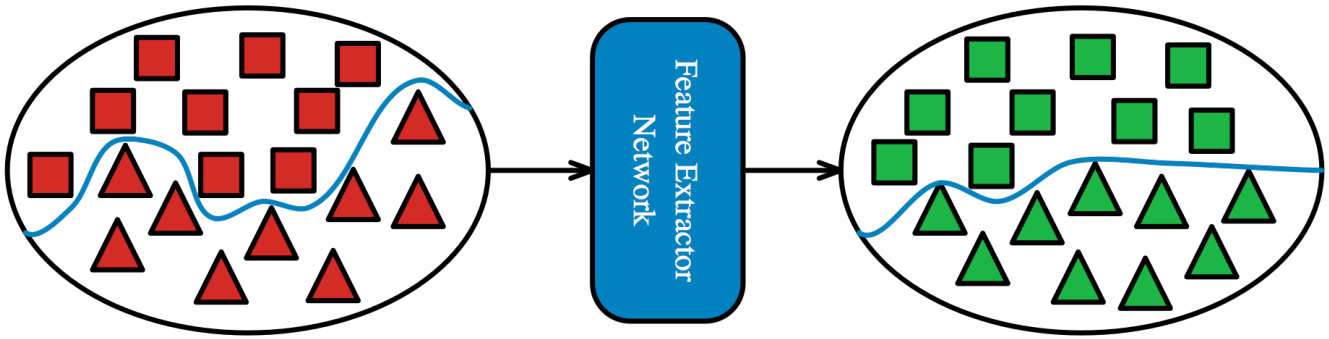
Domain Shift and Domain Adaptation

- Training data are drawn from source domain $\{X_s, \varphi_s\}$, and test data are drawn from target domain $\{X_t, \varphi_t\}$.
- Both domain have the same set of features ($X_s, X_t \in R^m$).
- Domain shift ($P(X_s) \neq P(X_t)$) makes ordinary learning methods degenerate.
- We can use domain adaptation methods to eliminate or reduce the domain shift.



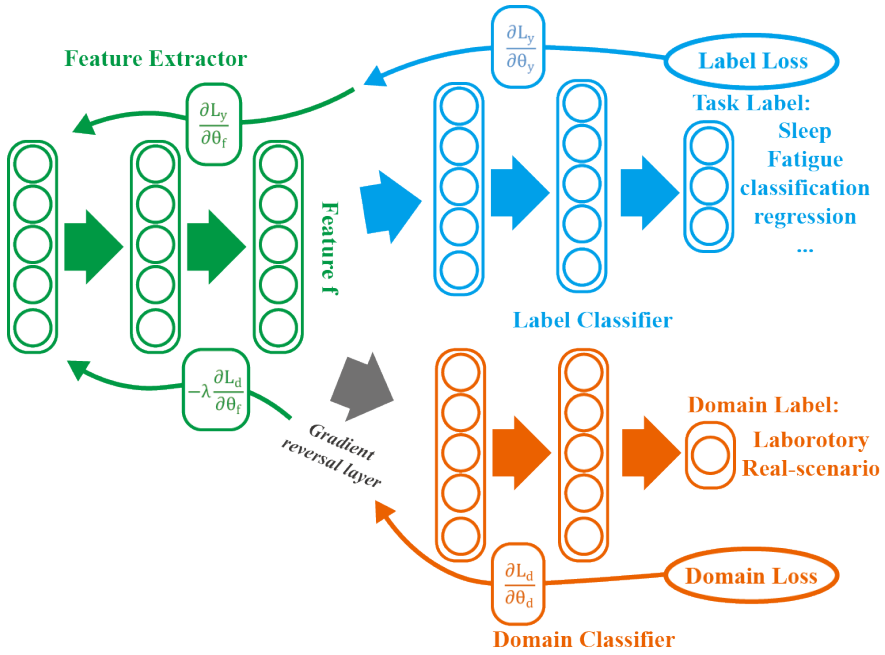
Domain Adaptation with Adversarial Networks

- To reduce the domain shift, neural networks can be used to transform the original feature into a new feature space where two conditions must be satisfied:
  - The new features keep essential information for label prediction.
  - Domain shift is reduced (features from both domains share similar distributions in the new feature space).



Domain Adversarial Neural Network

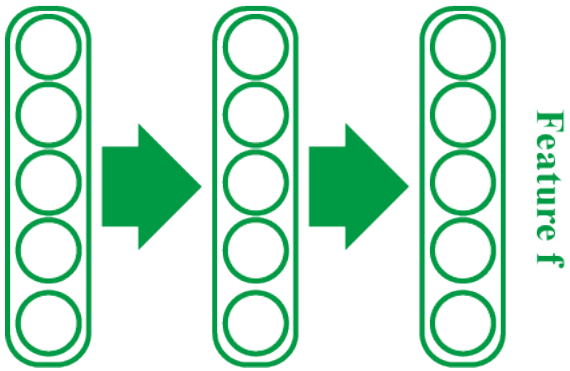
DANN is a domain adaptation approach with deep architectures that can be trained with labeled source domain data and unlabeled target domain data. Its adaptation behavior is achieved by augmenting a normal feed-forward model with an adversarial manner.



Paper: Y. Ganin and V. Lempitsky, Unsupervised domain adaptation by backpropagation, ICML2015, pp. 1180–1189.

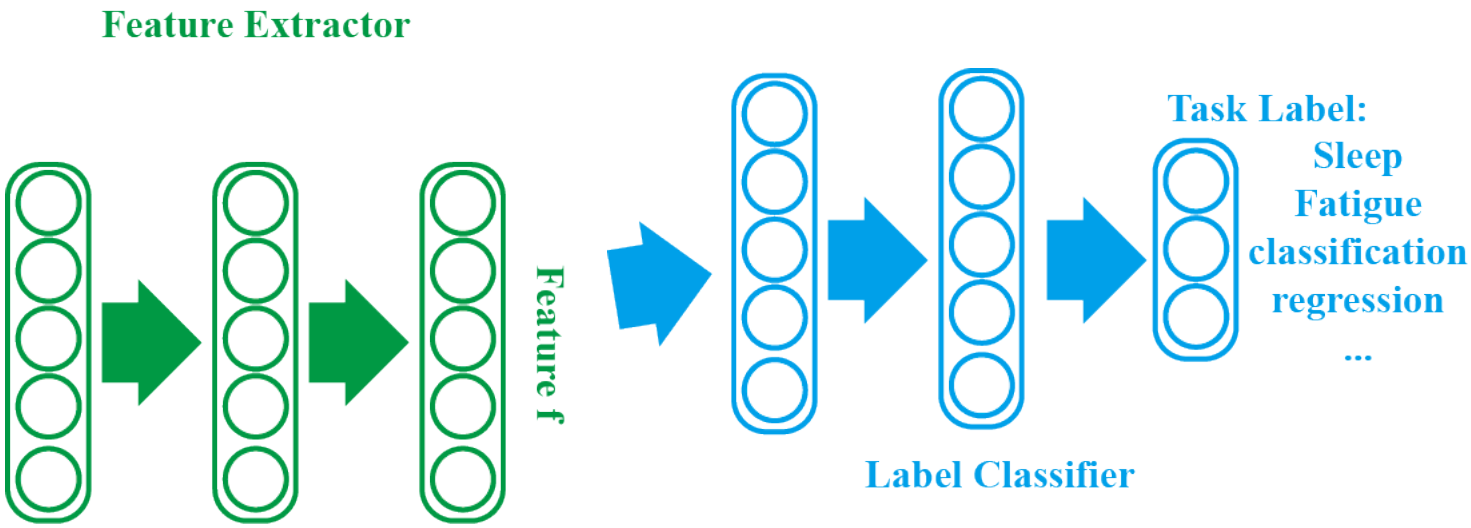
Domain Adversarial Neural Network

Feature Extractor



The first layers works as a feature extractor, it maps the low level features to a non-linear transformed high level feature f

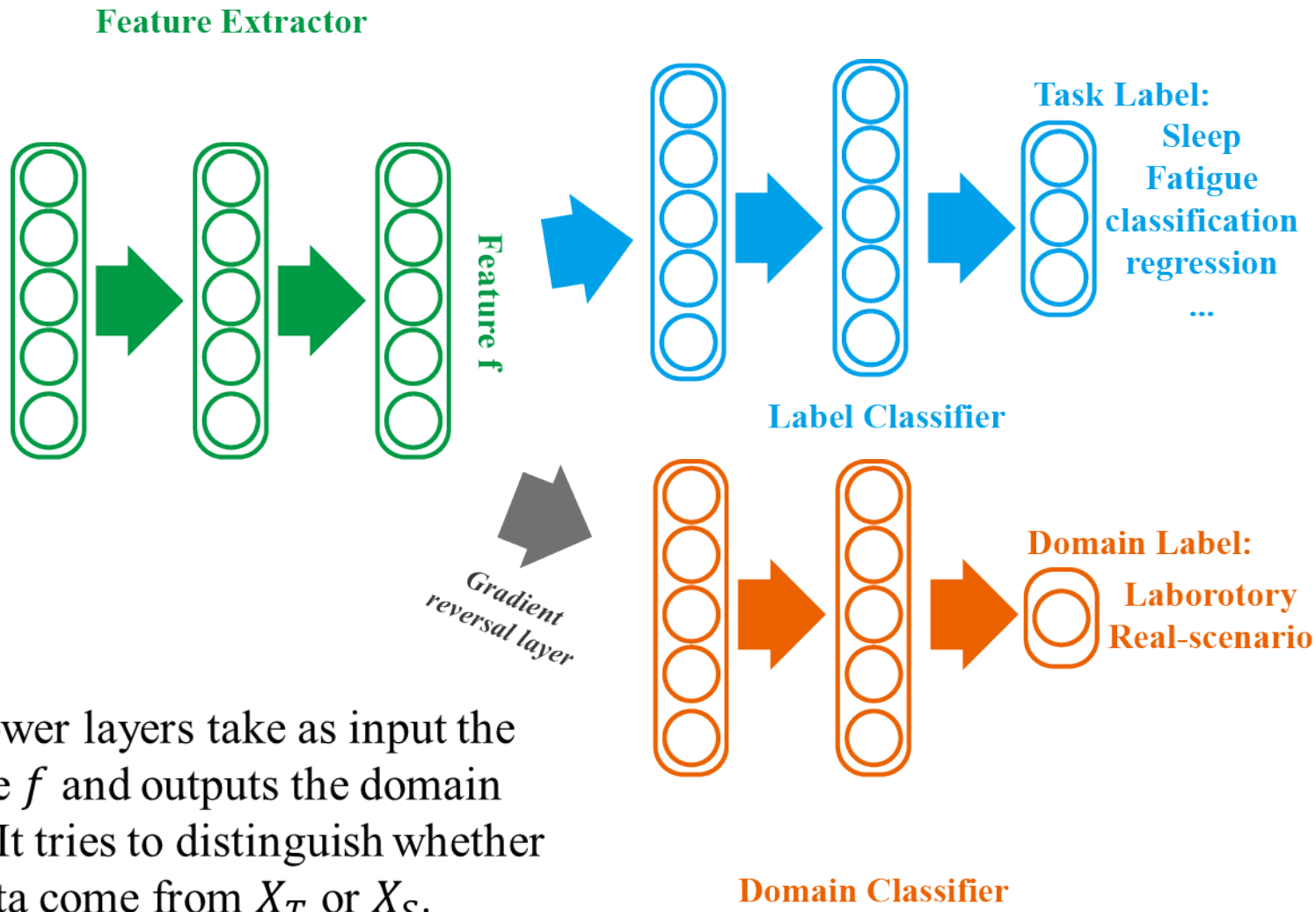
Domain Adversarial Neural Network



The upper layers take as input the feature f and outputs the label prediction for the required task

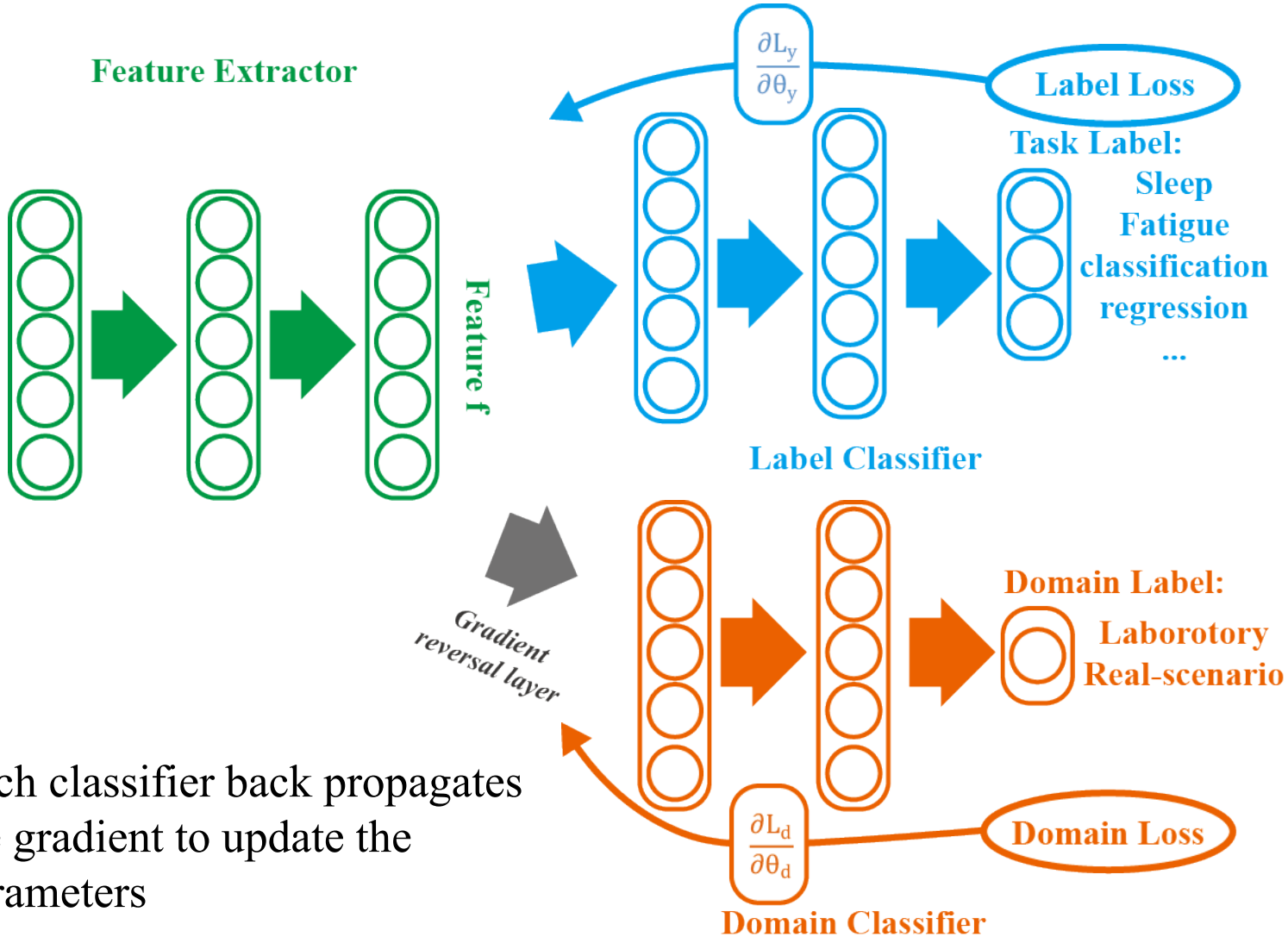
These two parts works like a normal multilayer network.

Domain Adversarial Neural Network



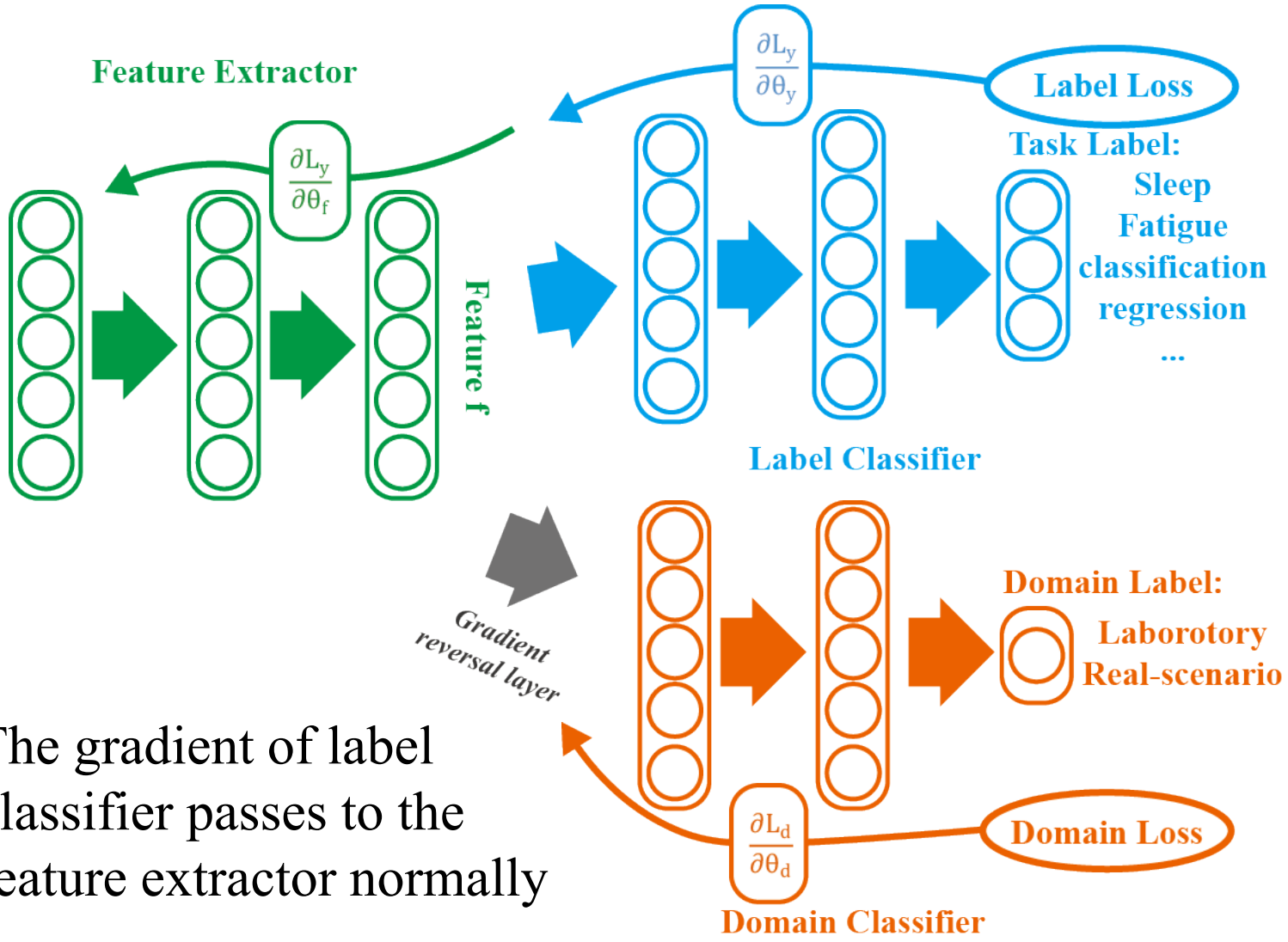
The lower layers take as input the feature f and outputs the domain label. It tries to distinguish whether the data come from X_T or X_S.

Domain Adversarial Neural Network



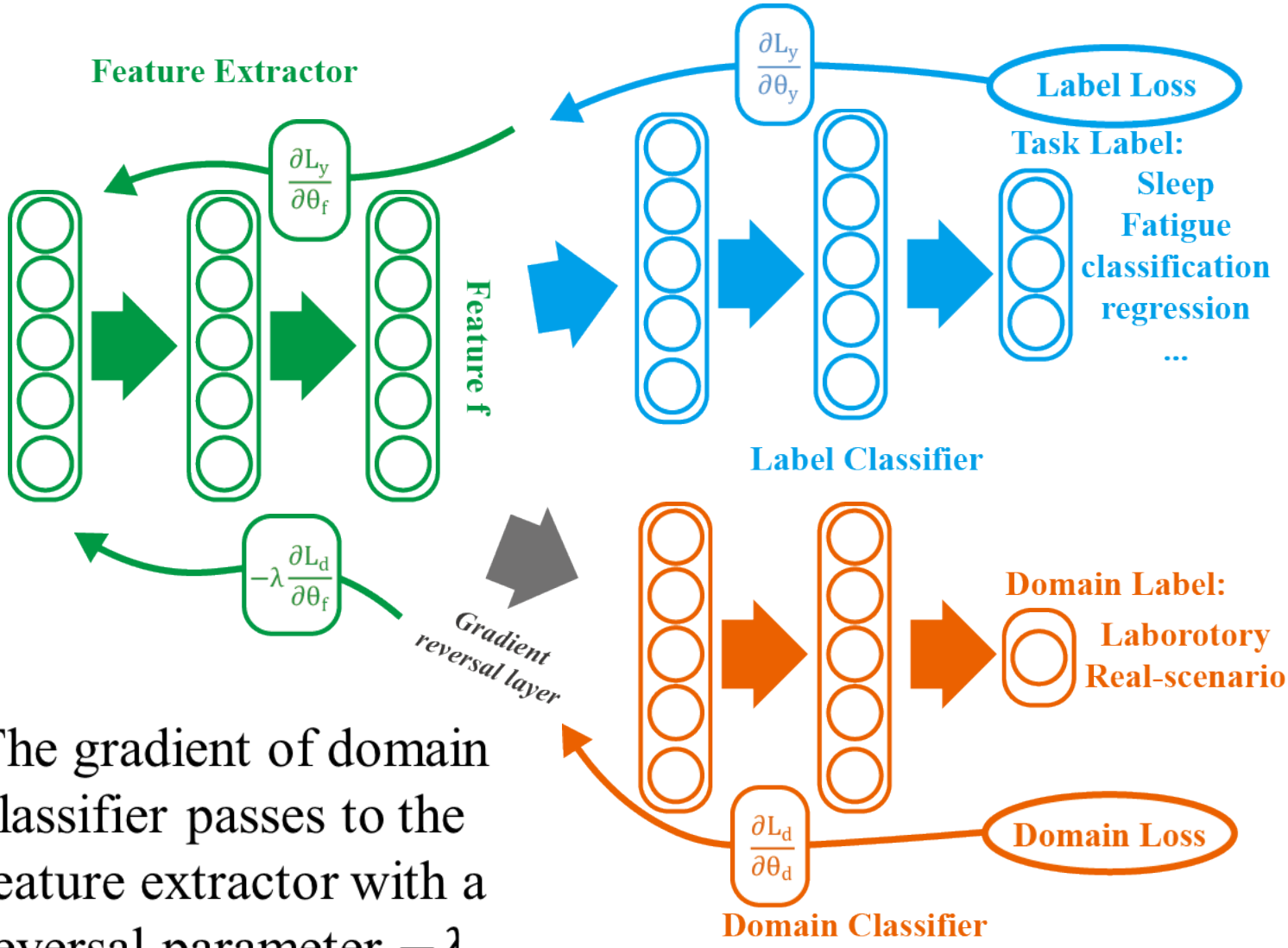
Each classifier back propagates the gradient to update the parameters

Domain Adversarial Neural Network



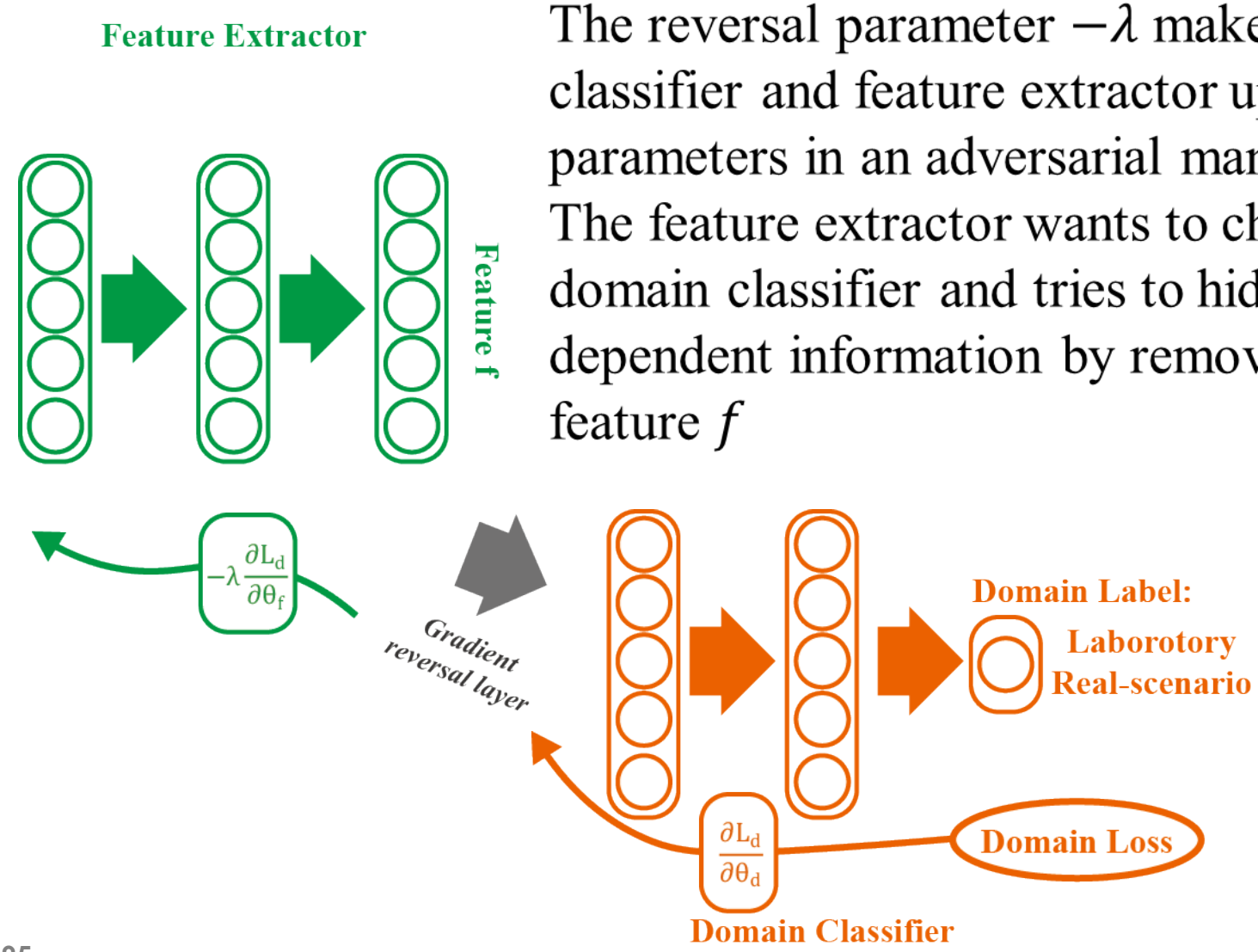
The gradient of label classifier passes to the feature extractor normally

Domain Adversarial Neural Network



The gradient of domain classifier passes to the feature extractor with a reversal parameter $-\lambda$

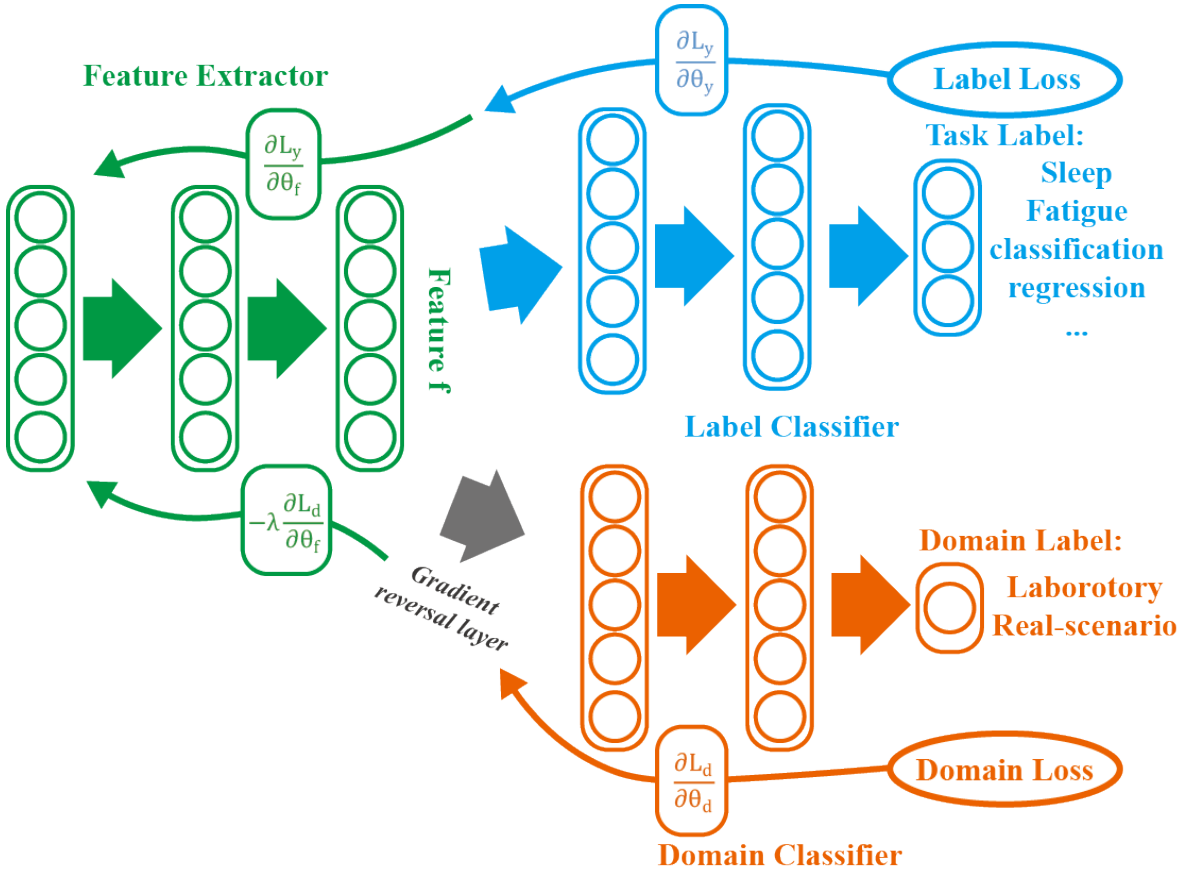
Domain Adversarial Neural Network



The reversal parameter $-\lambda$ makes the domain classifier and feature extractor updates their parameters in an adversarial manner.
The feature extractor wants to cheat the domain classifier and tries to hide any domain dependent information by removing it from feature f

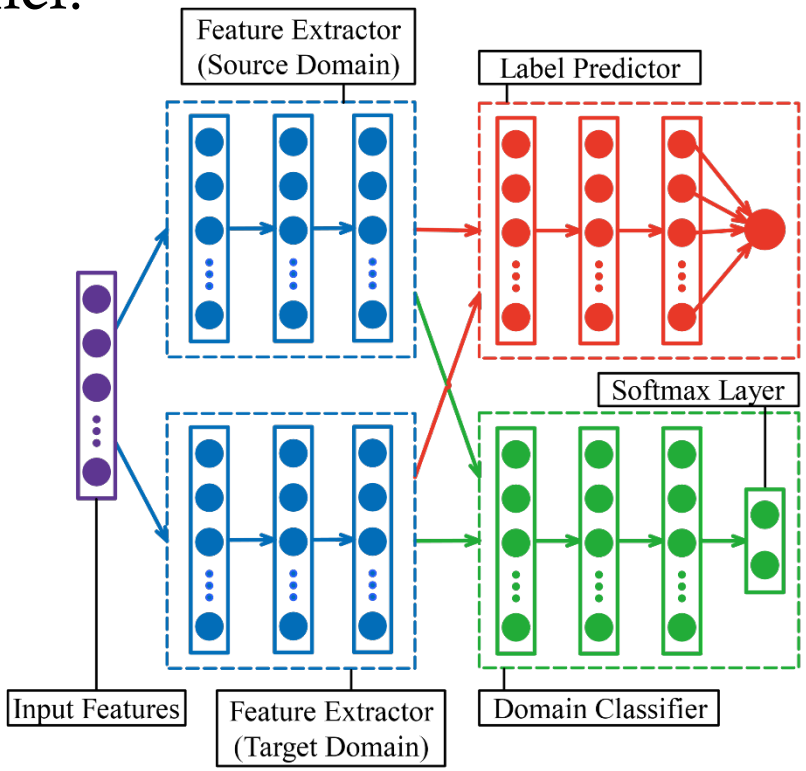
Domain Adversarial Neural Network

When the model converges, the feature f would be containing no domain related information. Thus it works as function of domain adaptation.



Adversarial Discriminative Domain Adaptation

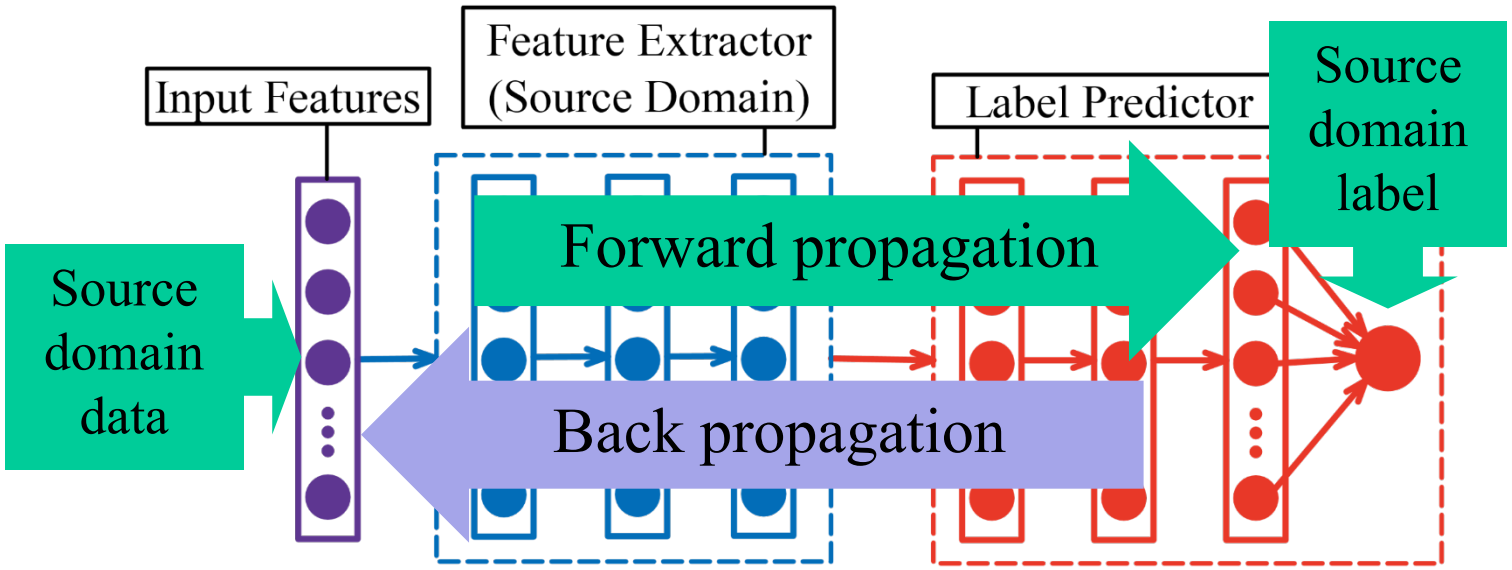
- ADDA is based on deep adversarial network. It aligns the distributions of source domain data and target domain data by training a feature extractor and a domain classifier in an adversarial manner.



E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in IEEE Conference on Computer Vision and Pattern Recognition, July 2017, pp. 2962–2971.

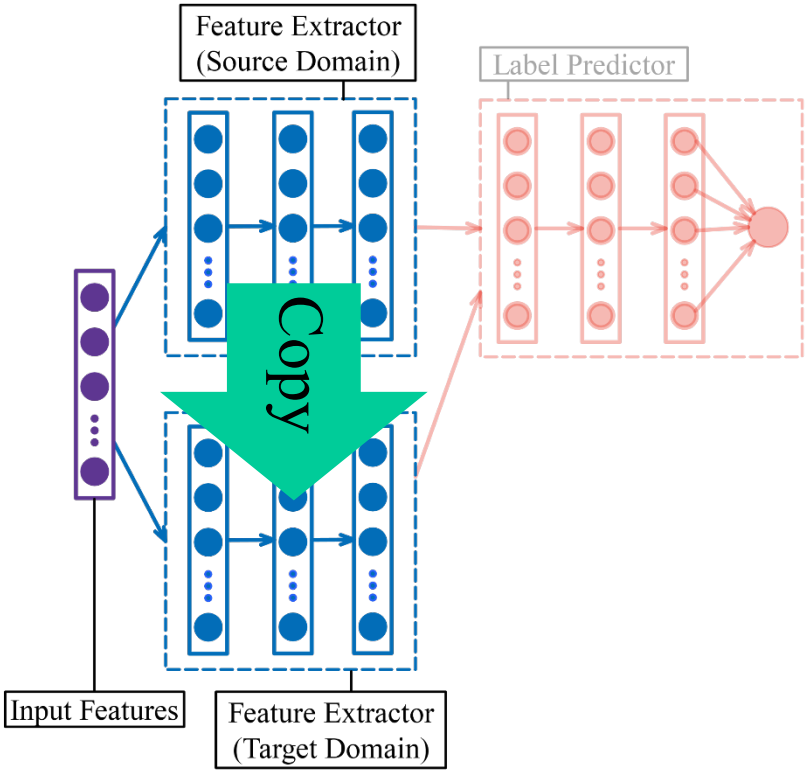
Training of ADDA (step 1)

- First, we train a simple multilayer perceptron on the source domain data with source domain data as input. The network is divided into 3 parts: input layer, feature extractor and label predictor. We can follow the general training procedure for this step (e.g., BP algorithm).



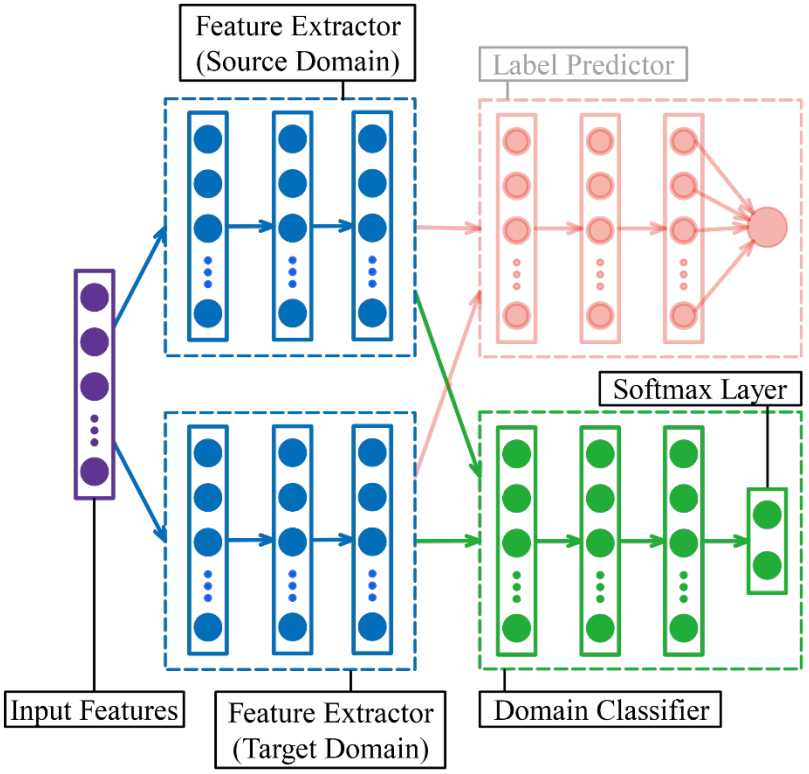
Training of ADDA (step 2)

- After the network converge, another feature extractor is constructed. The new feature extractor only accepts input from the target domain. Its parameters are initialized by copying the ones from the original feature extractor.



Training of ADDA (step 3)

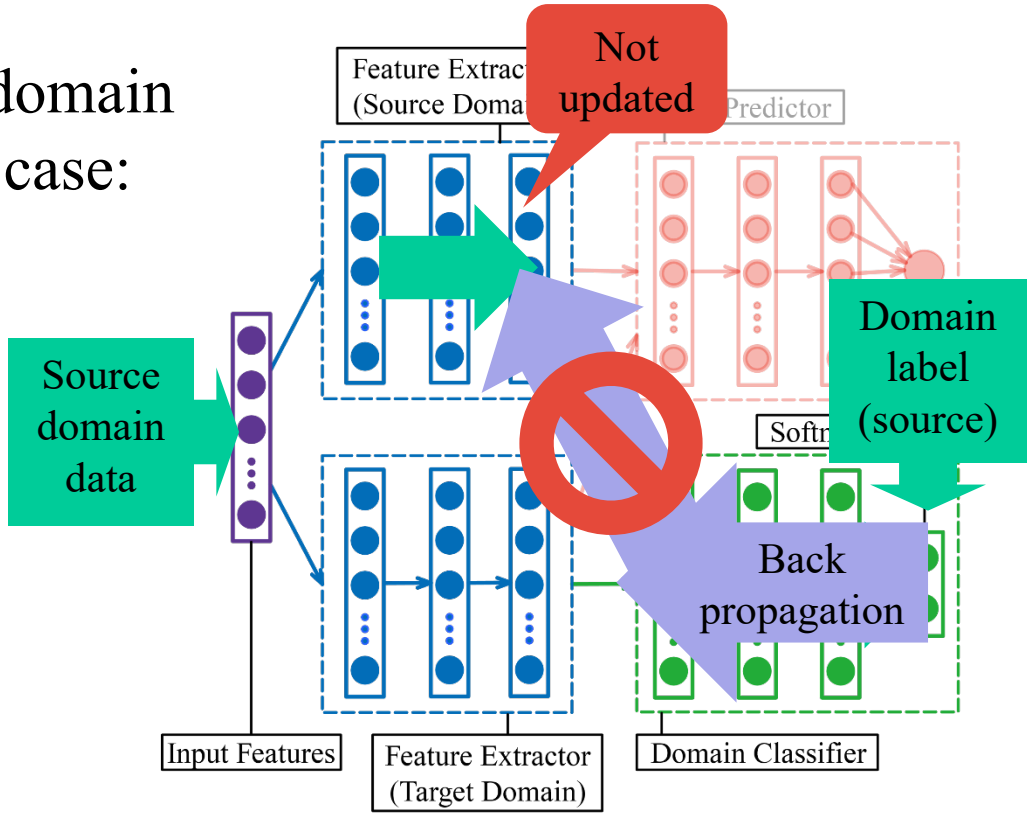
- In this step, a domain classifier sub-network is trained when the label predictor and the source domain feature extractor are not updated. The domain classifier is trained to discriminate which domain (source or target?) the input is from.



Training of ADDA (step 3 Cont.)

- However, the target domain feature extractor is trained to deceive the domain classifier. The domain classifier and the target domain feature extractor compete with each other (i.e., adversarial training).

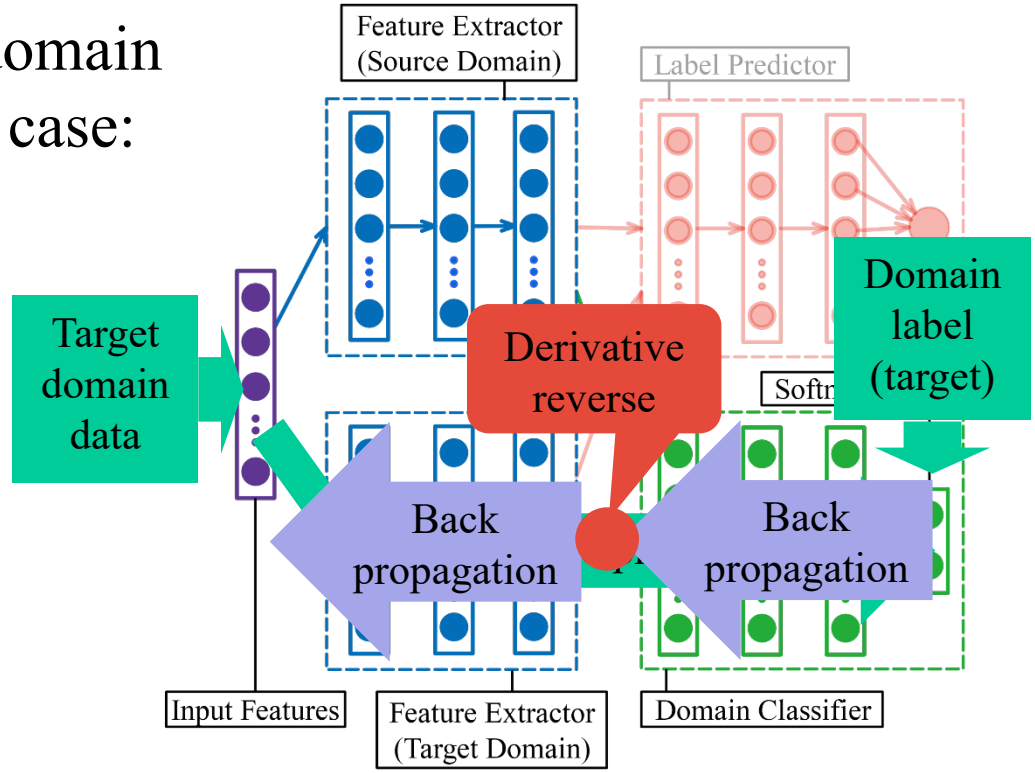
Source domain training case:



Training of ADDA (step 3 Cont.)

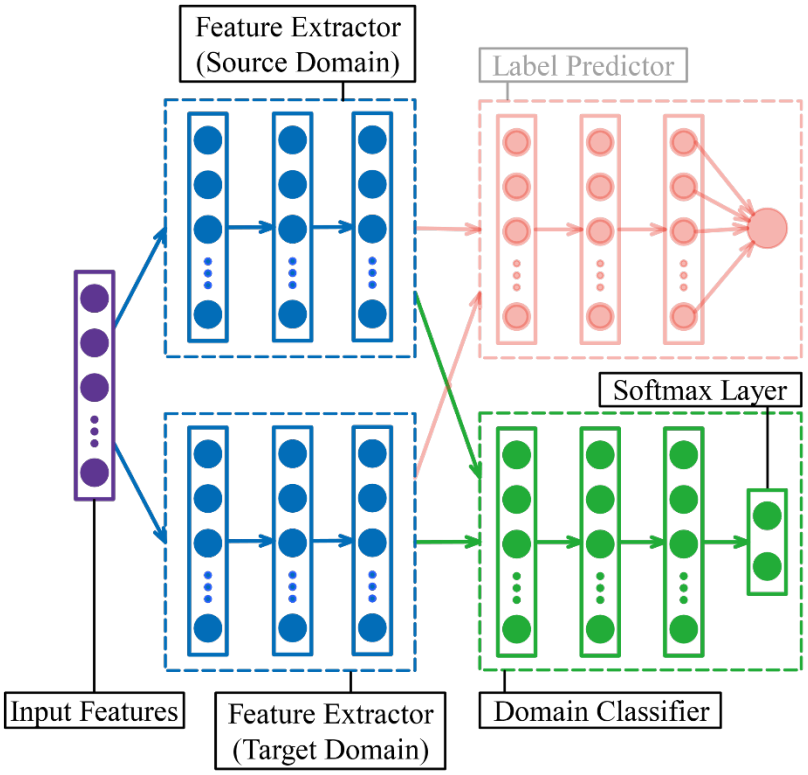
- However, the target domain feature extractor is trained to deceive the domain classifier. The domain classifier and the target domain feature extractor compete with each other (i.e., adversarial training).

Target domain training case:



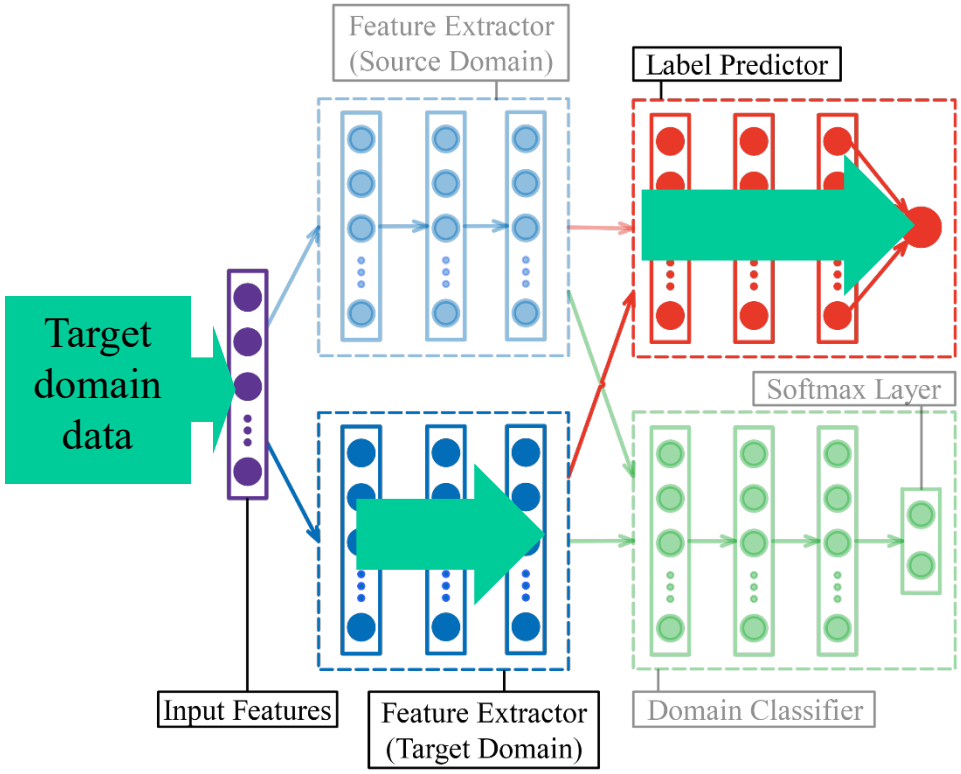
Training of ADDA (step 3 Cont.)

- As the training procedure progresses, the target feature extractor becomes better and better in deceiving the domain classifier. In another word, the target feature extractor extracts features that are similar to the source ones. In this way, the domain discrepancy is reduced.



Training of ADDA (step 4)

- After the adversarial training converges, we can predict the labels of the target domain data by using the pass as shown in the figure.



Personalizing EEG-based Affective Models with Transfer Learning

Wei-Long Zheng and Bao-Liang Lu, Personalizing EEG-based Affective Models with Transfer Learning, Proc. of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16).

Feature Reduction based Subject Transfer

Although the distributions of source domain and target domain in high dimensional space are different, we may find a low dimensional manifold space where the distributions of both domains are similar.

TCA and KPCA try to learn a set of common transfer components underlying both the source domain and the target domain. When projected to this subspace, the difference of feature distributions of both domains can be reduced.

$$P(\phi(X_S)) \approx P(\phi(X_T))$$
$$P(Y_S|\phi(X_S)) \approx P(Y_T|\phi(X_T))$$

TCA-based Subject Transfer

TCA algorithm is firstly proposed by Pan et al. for cross-domain indoor WiFi localization and cross-domain text classification.

Algorithm 1 TCA-based Subject Transfer

input : Source domain data set $\mathcal{D}_S = \{(x_{S_i}, y_{srci})\}_{i=1}^{n_1}$, and target domain data set $\mathcal{D}_T = \{x_{T_j}\}_{j=1}^{n_2}$.

output : Transformation matrix W.

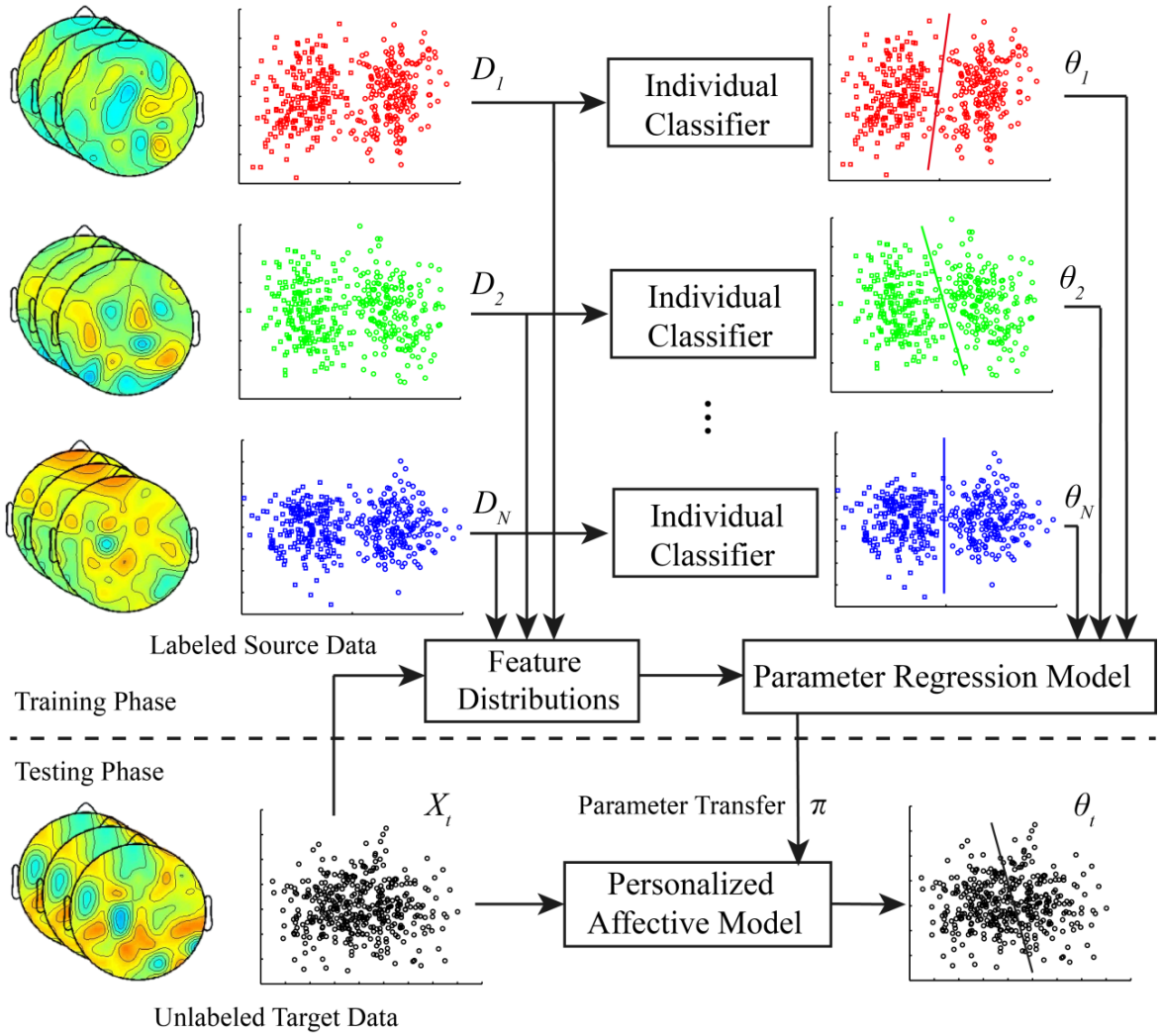
1: Compute kernel matrix K from $\{x_{S_i}\}_{i=1}^{n1}$ and $\{x_{T_j}\}_{j=1}^{n2}$, matrix L, and the centering matrix H.

2: Eigendecompose the matrix $(KLK + \mu I)^{-1}KLK$ and select the m leading eigenvectors to construct the transformation matrix W.

3: **return** tranformation matrix W.

Pan S J, Tsang I W, Kwok J T, et al. Domain adaptation via transfer component analysis [J]. IEEE Transactions on Neural Networks, 2011, 22(2): 199-210.

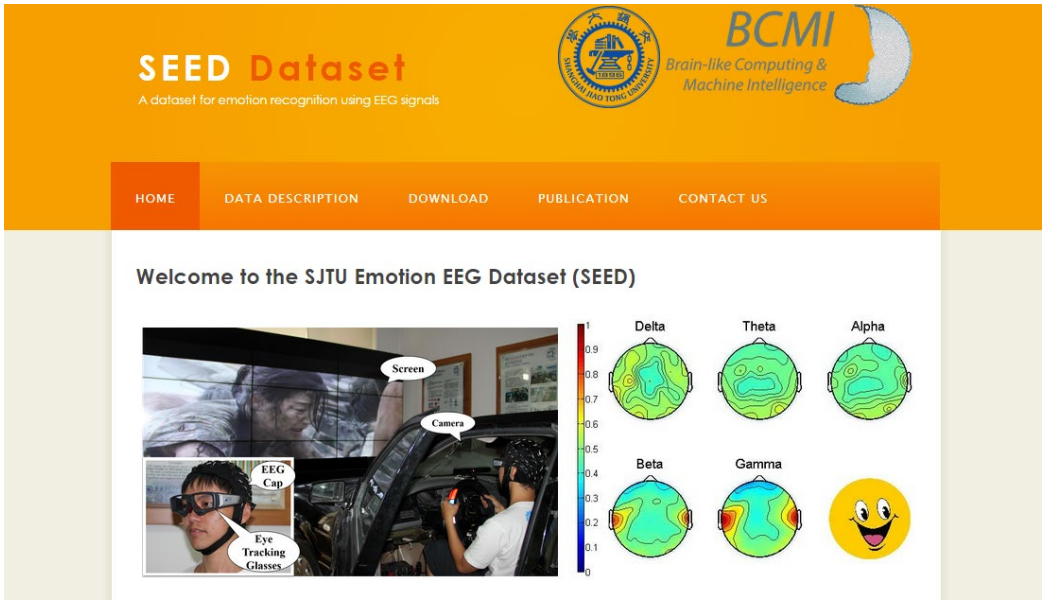
Transductive Parameter Transfer



Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM International Conference on Multimedia, pages 357–366. ACM, 2014.

SJTU Emotion EEG Dataset (SEED)

- No. subjects: 15, No. EEG Channels: 62
- EEG Features: differential entropy in five frequency bands.
- Three emotions: positive, neutral and negative



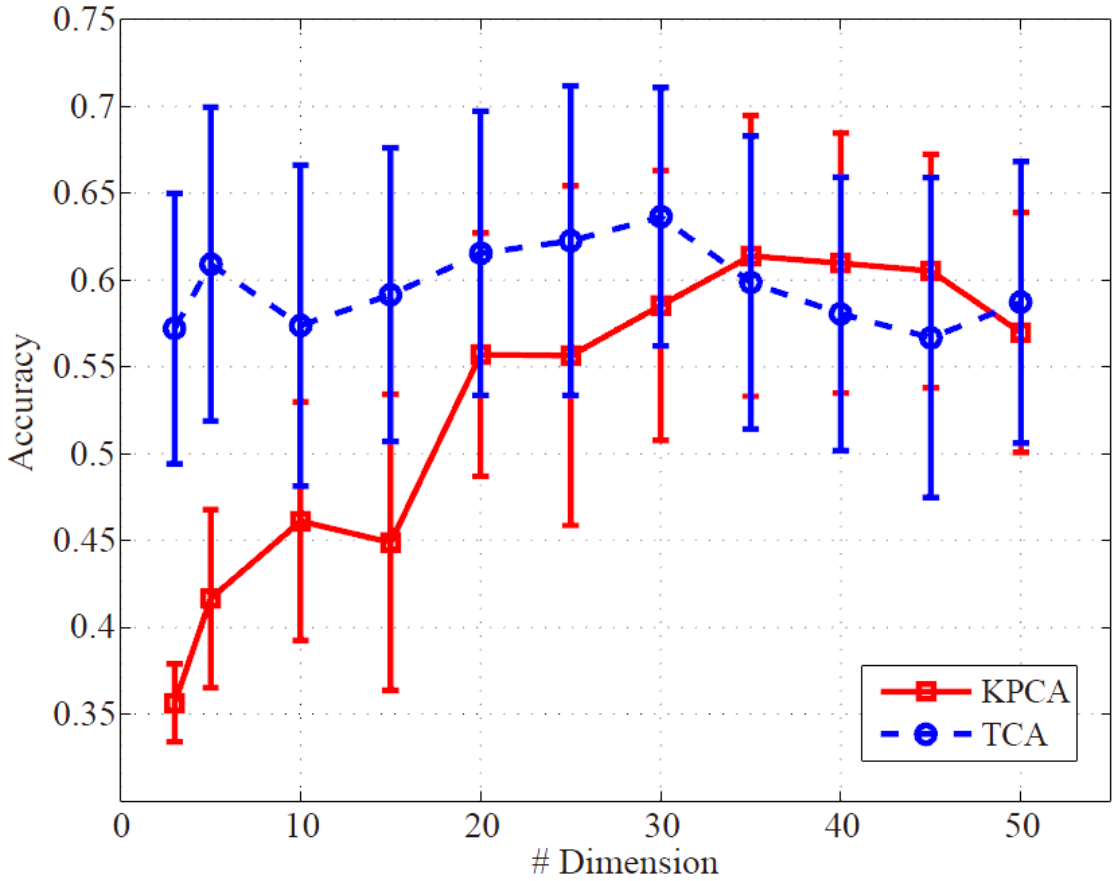
http://bcmi.sjtu.edu.cn/~seed/

"Investigating Critical Frequency Bands and Channels for EEG-based Emotion Recognition with Deep Neural Networks", Wei-Long Zheng, and Bao-Liang Lu, IEEE Transactions on Autonomous Mental Development (IEEE TAMD), 2015.

Evaluation Details

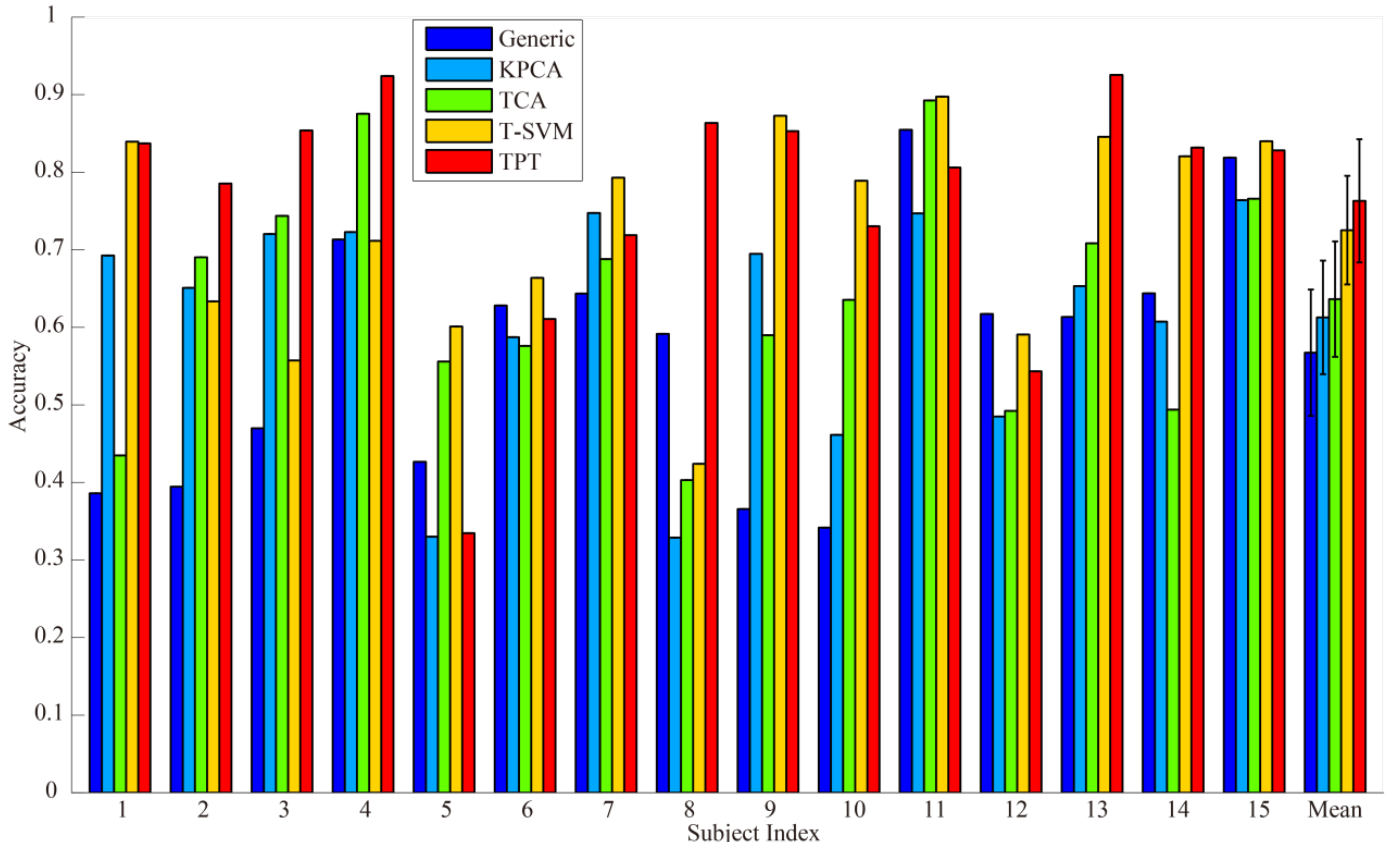
- Leave-one-subject-out cross validation method
- Baseline method
  - We concatenate data from all available subjects as training data and train a generic classifier with linear SVM.
- Transductive SVM (T-SVM)
  - It is developed to learn a decision boundary and maximize the margin with unlabeled data.
- One vs. one strategy for multi-class classification
- All the algorithms are implemented in MATLAB.

Experimental Results



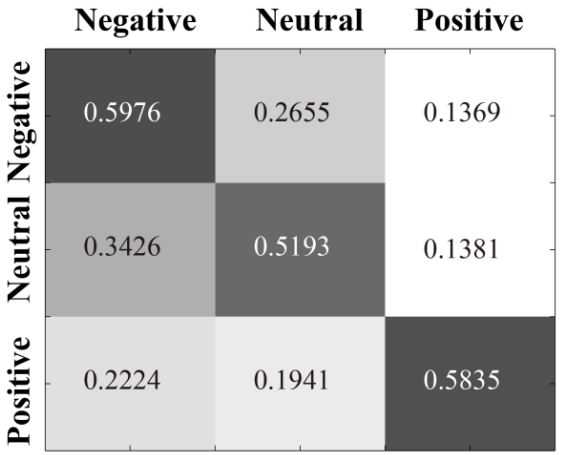
Comparison of KPCA and TCA approaches for different dimensionality of the subspace.

Experimental Results (2)

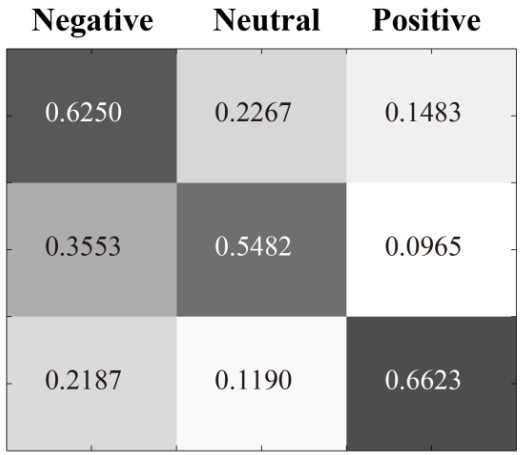


Stats.	Generic	KPCA	TCA	T-SVM	TPT
Mean	0.5673	0.6128	0.6364	0.7253	**0.7631**
Std.	0.1629	0.1462	0.1488	**0.1400**	0.1589

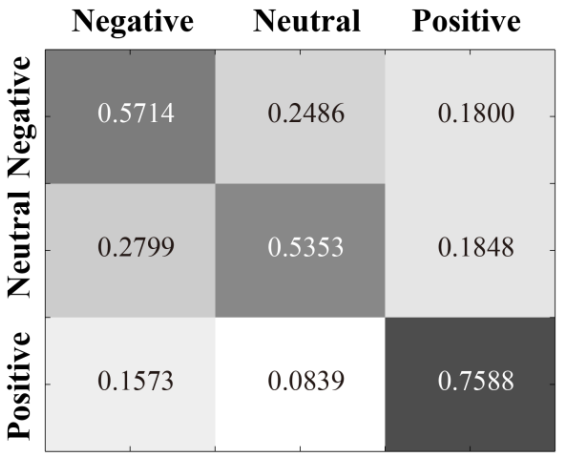
Generalization Performance



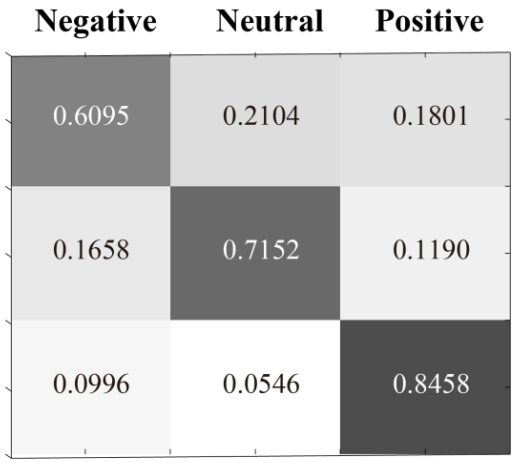
(a) Generic



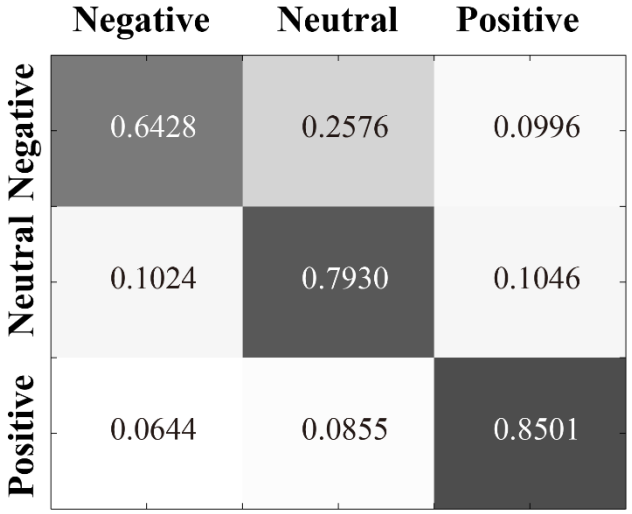
(b) KPCA



(c) TCA



(d) T-SVM



(e) TPT

Sleep Quality Estimation: From Laboratory to Real Scenario

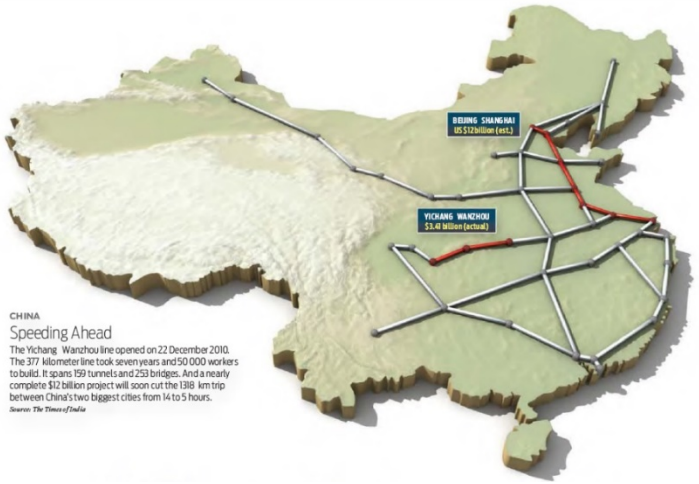
Jia-Jun Tong, Yun Luo, Bo-Qun Ma, Wei-Long Zheng, Bao-Liang Lu, Xiao-Qi Song and Shi-Wei Ma， Proc. IJCNN2018

Introduction

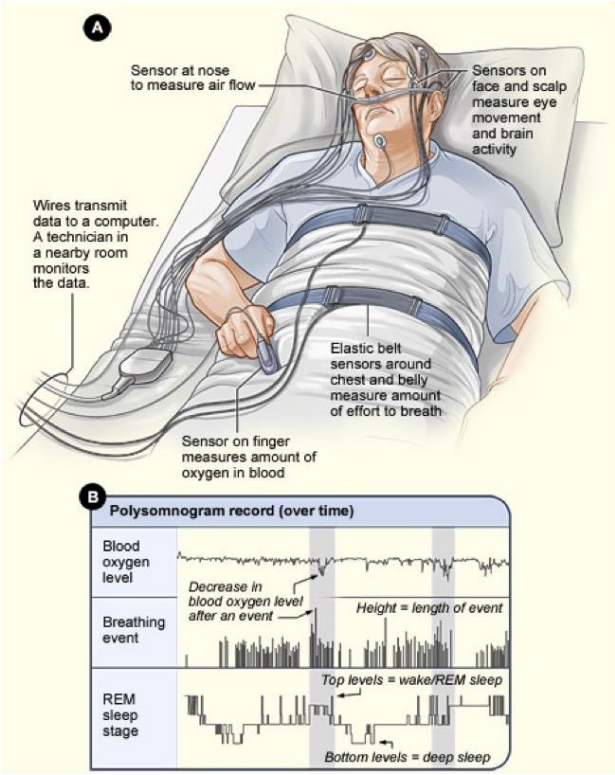
- Sleep quality evaluation has remarkable value.
- An objective and effective measurement of sleep quality is quite valuable in transportation, medicine, health care, and neuroscience.

Background: High-speed railways in China

- 20380 kms in use. It is about half the world's supply
- 16775 kms are developing
- 30000 kms will be finished by 2020
- Active safety and mental state monitoring technologies become very important
  - **Check sleep quality before work**
  - Predict drowsy driving during work
  - Monitor mental state before work, during work, and after work



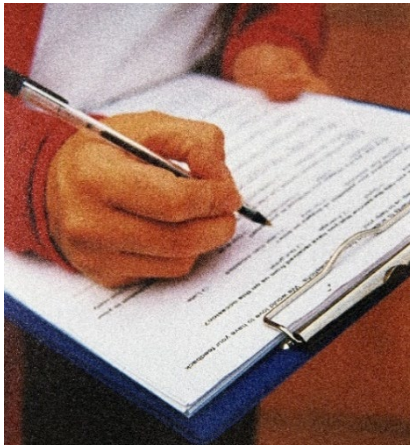
Monitoring Whole Sleep Procedure



- Polysomnography (PSG) and smart bands require the subjects to wear equipments such as EEG cap, eye sensors, nose sensors, elastic belt sensors during the **whole sleep procedure**

Our Research Goal

- At present, sleep quality assessment is performed by using questionnaire at CRH (China Railway High-speed）
  - Subjective
  - inaccurate
- An objective and convenient method as alcometer is perfect



Our previous work

We proposed an approach which is characterized by

- EEG-based Evaluation
- Data acquisition in 30 minutes, instead of whole-process physiological signal acquisition
- Assessment is data driven
- Subject-independent (transfer learning)

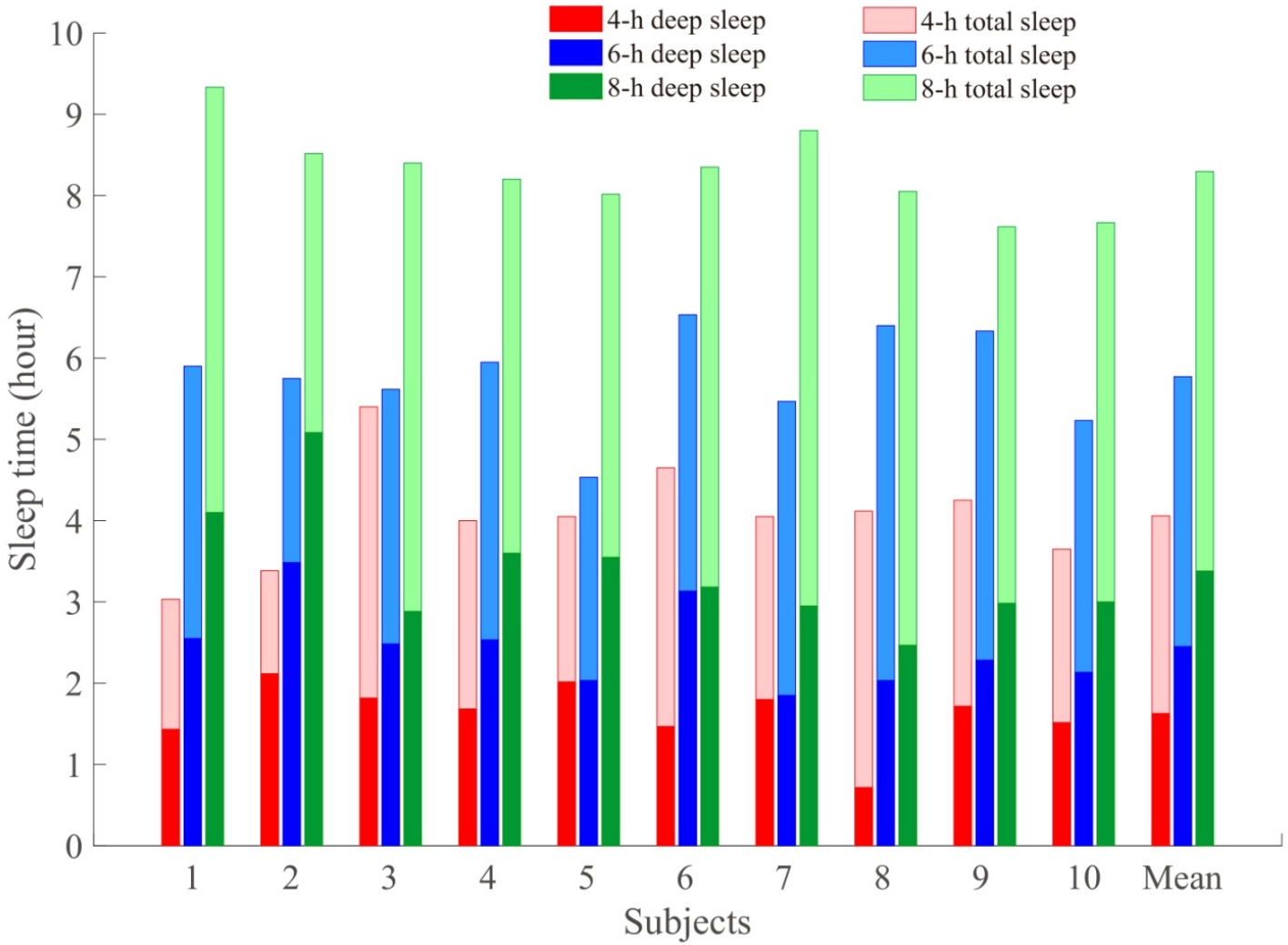
Xing-Zan Zhang, Wei-Long Zheng, and Bao-Liang Lu, EEG-based sleep quality evaluation with deep transfer learning, **Proc. ICONIP** 2017.

Li-Li Wang, Wei-Long Zheng, Hai-Wei Ma and Bao-Liang Lu, Measuring Sleep Quality from EEG with Machine Learning Approaches, **Proc. IJCNN 2016**

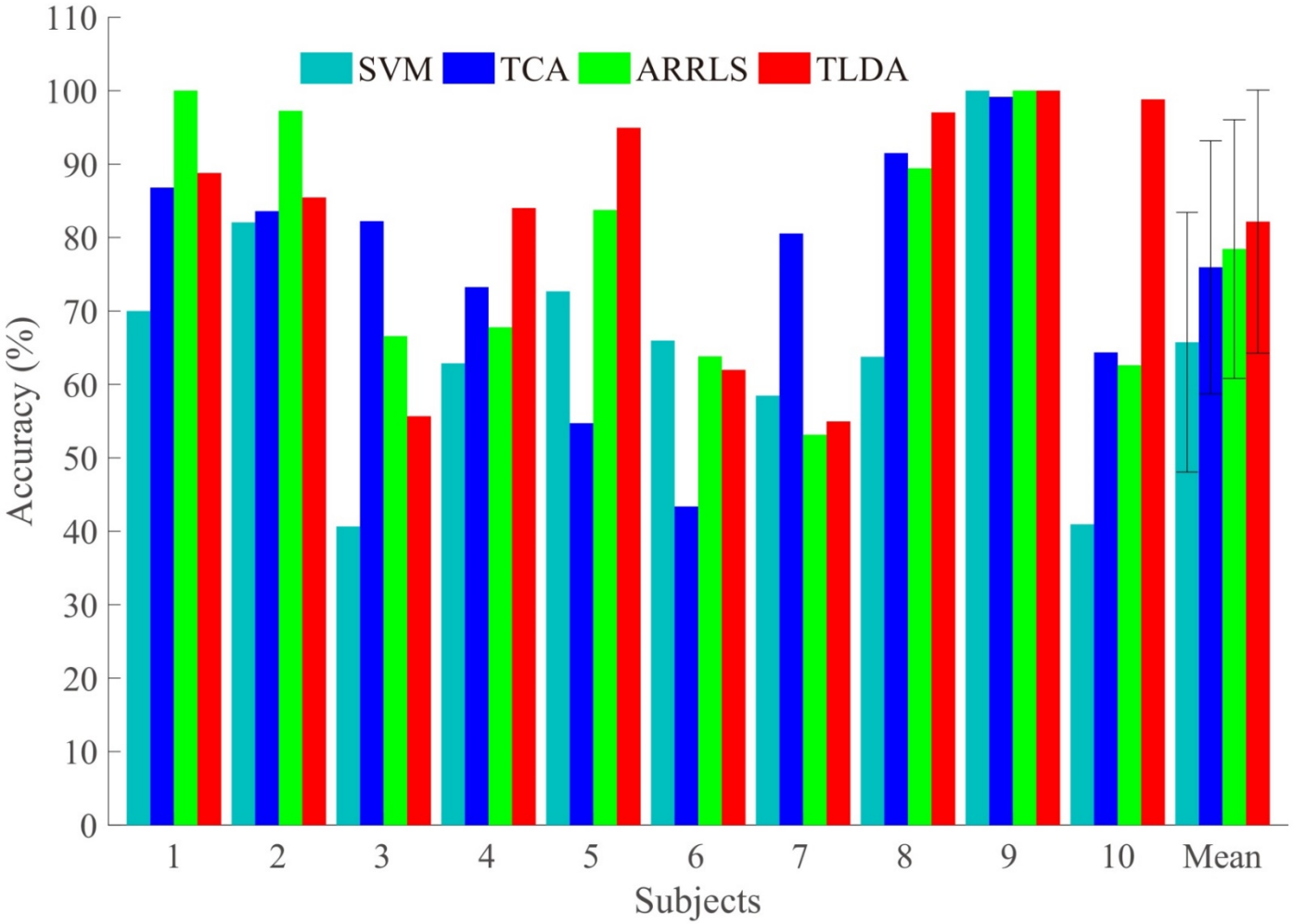
Experiment Settings

- Deep sleep is the main factor that counts for sleep quality according to National Sleep Foundation (NSF)
- 4-h (poor), 6-h (normal), 8-h (good)
- 3:00-7:00, 1:00-7:00, 23:00-7:00,
- Subjects: six males and four females
- Age range: 21-26, mean: 23.57, std: 1.62

Total and Deep Sleep Time

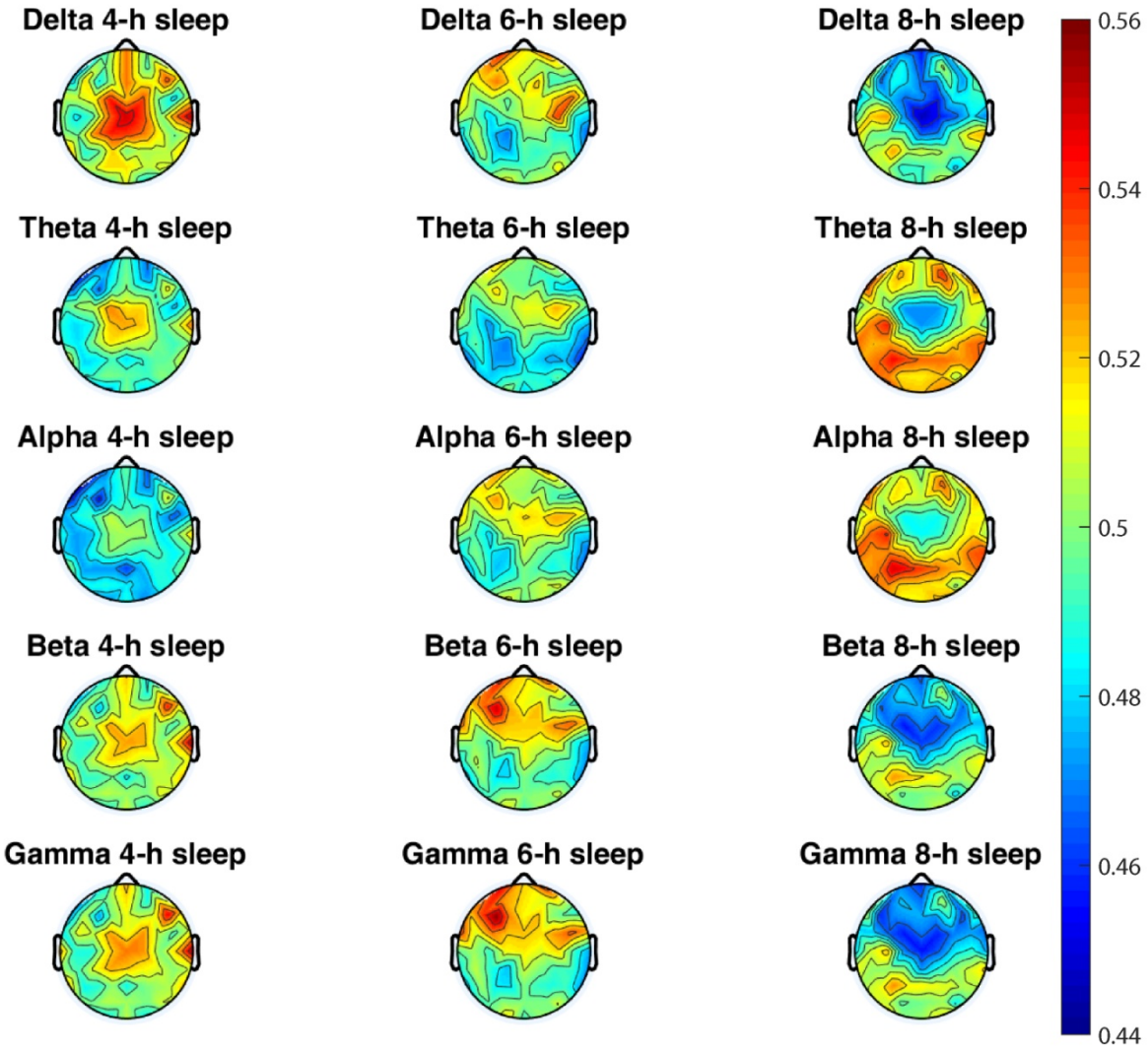


Results and discussion



Accuracy comparison of SVM, TCA, ARRLS and TLDA for each subject

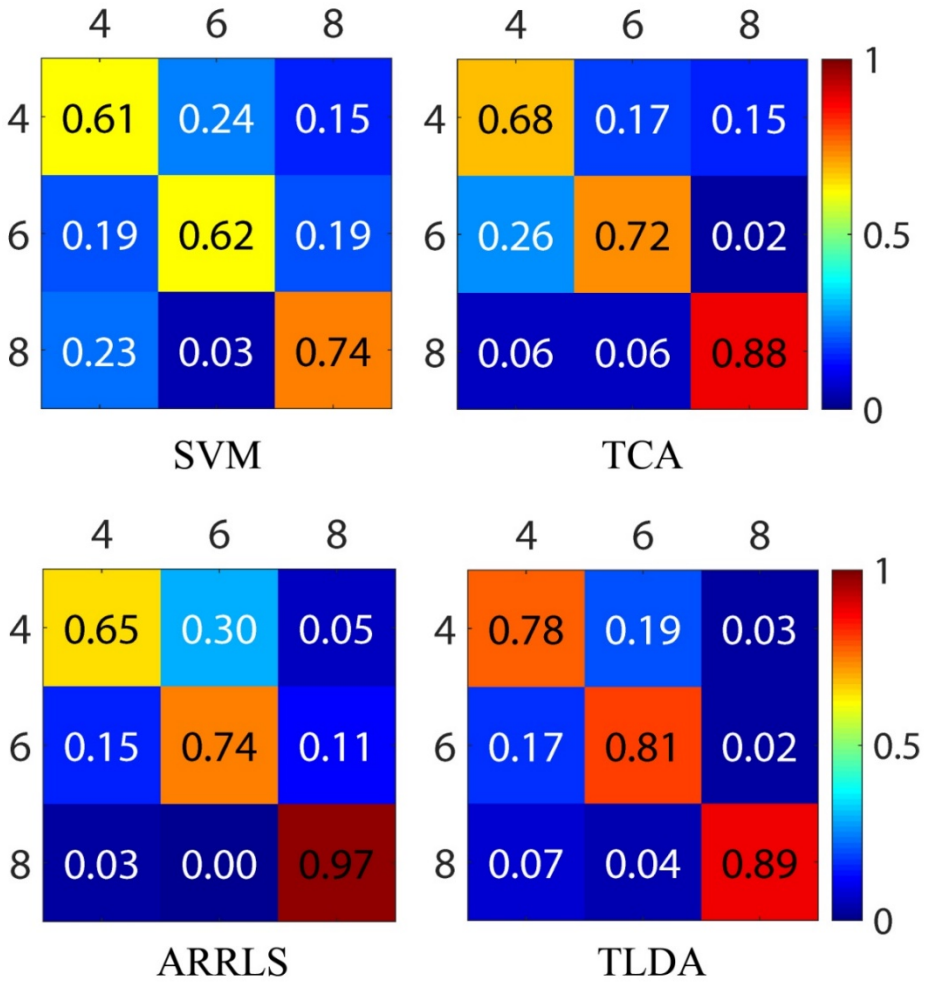
Results and discussion-DE pattern



Neural patterns of three kinds of sleep quality

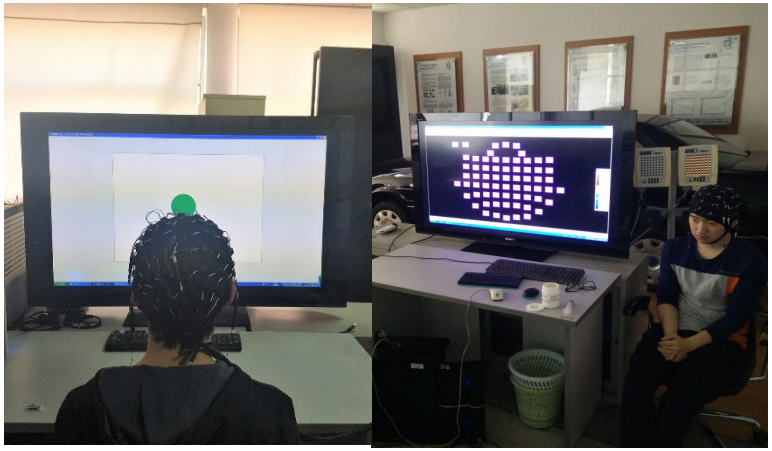
Results and discussion-confusion matrices

- 8-hour sleep can be more effectively recognized.
- 4-hour and 6-hour sleep are more likely to be misclassified with each other.
- DE patterns of 4-hour sleep and 6-hour sleep are much closer than the patterns of 8-hour sleep.



Motivation

- The data collection and annotation in real scenarios are usually costly and difficult
- Transferring knowledge from laboratory to real scenario
  - Laboratory: Students with controlled sleep time
  - Real Scenario: CRH railway drivers before work
- Cross-scenario & Cross-subject tasks



Modeling
Predicting

In the Laboratory

- We take 4-hour sleep, 6-hour sleep and 8-hour sleep as poor, normal and good in terms of the sleep quality in our study
- The sleep time and wake up time for three experiments are 3:00-7:00, 1:00-7:00 and 23:00-7:00, respectively.
- Total sleep time and deep sleep time of all subjects are recorded by smart bands
- High quality EEG acquisition: 62-channel wet electrode cap
- Different state for same subjects: 3 times, 30 min each experiments

4h
6h
8h

In the Real Scenario (CRH)

- To keep the drivers' daily routine, the experiment settings have to be adjusted to meet the need of real-scenario application.
  - No control on sleep time: range of 2~10 h
  - Easy to set up devices: 18-channel dry electrode cap (DSI-24)
  - One experiment for each subjects: 1 time, 6 min each

Sleep:
2~10h

(CRH: China Railway High-speed)

Summary of Domain Differences

Categories	Specifications	Laboratory	Real-scenario
Subjects	Age	21-26	25-49
	Sex-distribution	4 males, 6females	70 males
	Occupation	Students	High-speed train drivers
	Experiments per subject	3	1
Controlled conditions	Sleep time	4,6,8 hours	Not controlled
	Monitoring sleep	Yes	No
	Head Cleaning	Yes	No
	Experiment Starts	$\leq$1 hour after wake up	1-10 hour after wake up
Experiment settings	Experiment duration	30 min	6 min
	Task	Eyes open	Eyes open and close
	Environment	Quite and isolated	Noisy
Devices	Electrods type	Wet electrodes	Dry electrodes
	Resolution	62 channels	18 channels
	Reference	REF(Between CPZ and PZ)	A1 and A2 (On the ears)
	Common Mode Follower	0	1 (CMF)
	Sample rate	1000 Hz	300 Hz

Eight Domain Adaptation Methods

- Baseline methods
  - TCA: Transfer Component Analysis
  - ITL: Information-Theoretical Learning
  - GFK: Geodesic Flow Kernel
  - JDA: Joint Distribution Adaptation
  - SA: Subspace Alignment
  - TJM: Transfer Joint Matching
  - MIDA: Maximum Independence Domain Adaptation
- Adversarial Networks
  - Domain Adversarial Neural Network

Cross-subject and Cross-scenario Tasks

- After the preprocessing that unifies the format of data, the two domains still possess a lot of discrepancies
- View the Laboratory data as source domain, we can then use domain adaptation methods to transfer knowledge to the real scenario
- Cross-subject task: training and testing on real-world data
- Cross-scenario task: training on laboratory data but testing on real-world data

Results: Cross-subject Task

ACCURACY (%) OF CROSS-SUBJECT TASK

Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean±Std
SVM	59.57	48.41	57.42	55.71	49.74	54.17±4.87
TCA	58.37	72.98	63.77	69.48	53.85	63.69±7.82
ITL	66.07	66.79	57.14	76.43	56.71	64.63±8.13
GFK	59.41	66.79	53.73	58.45	58.49	59.37±4.70
JDA	59.41	59.52	**76.11**	60.44	55.91	62.28±7.92
SA	**77.42**	67.38	60.36	70.68	46.47	64.46±11.7
TJM	65.91	**74.72**	58.73	62.22	59.37	64.19±6.53
MIDA	71.35	69.76	66.91	69.21	56.51	66.75±5.94
DANN	67.50	74.64	71.23	**76.98**	**78.25**	**73.72**±4.38

The model performance is evaluated in a 5-fold cross validation manner. The real-world data is split into 5 parts, each containing 14 subjects and no subject's data should appear in multiple validation parts.

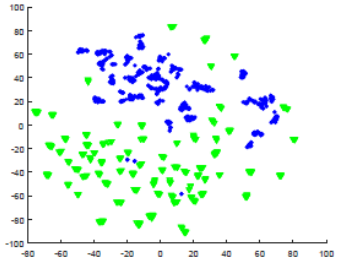
Results: Cross-scenario Task

ACCURACY (%) OF CROSS-SCENARIO TASK

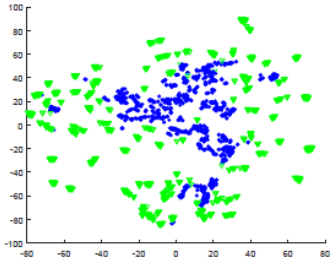
Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean±Std
SVM	43.86	40.35	42.18	38.98	43.59	41.79±2.10
TCA	54.60	61.19	38.89	67.34	56.87	55.60±10.6
ITL	53.69	63.10	24.68	60.71	49.41	50.32±13.7
GFK	49.64	62.46	40.99	**70.95**	49.25	54.66±11.9
JDA	47.70	61.43	35.71	66.11	66.94	55.58±13.5
SA	42.74	62.54	37.50	61.23	54.17	51.64±11.1
TJM	41.67	60.28	38.97	59.88	60.60	52.28±10.9
MIDA	56.47	**66.35**	38.69	68.93	56.87	57.46±11.9
DANN	**69.10**	64.68	**60.79**	62.6296	**69.25**	**65.29**±3.80

Experimental Results (cont.)

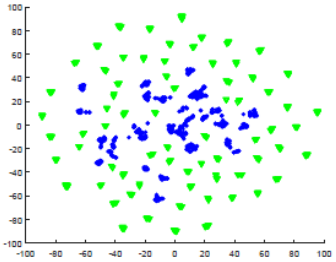
The effects of adaptation on cross-scenario task



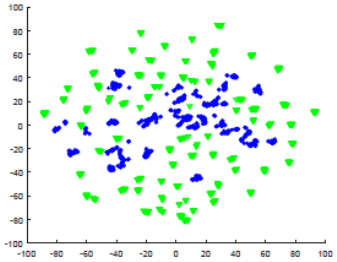
(a) None-adapted



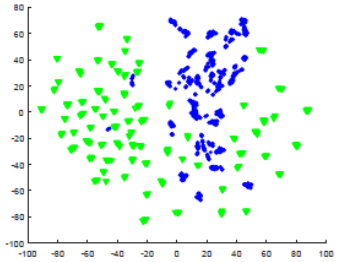
(b) TCA



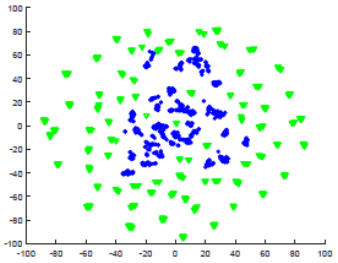
(c) ITL



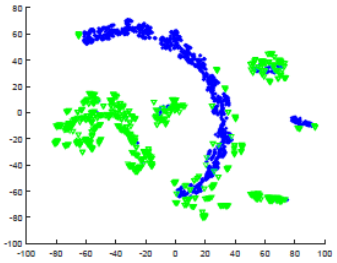
(d) GFK



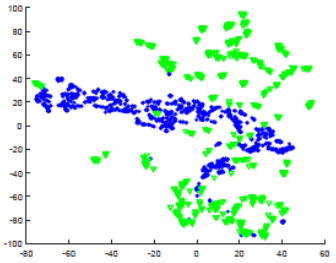
(e) JDA



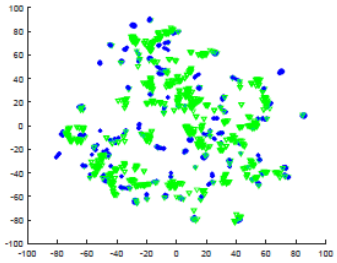
(f) SA



(g) TJM



(h) MIDA



(i) DANN

Conclusions

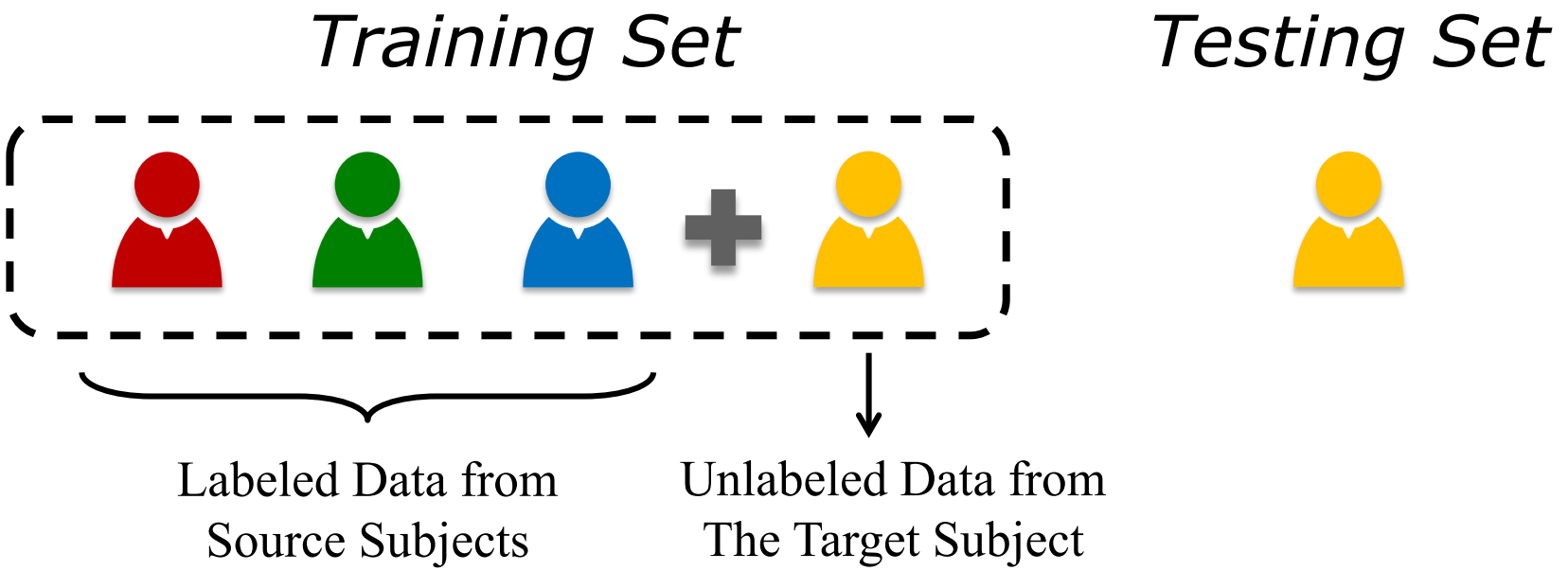
- To estimate the sleep quality of high-speed train drivers, we have evaluated two tasks: the cross-subject and the cross-scenario.
- Eight domain adaption methods have been introduced to both tasks to reduce the domain discrepancy between source and target domains.
- The experimental results have shown that Domain Adversarial Neural Network (DANN) approach outperforms other seven DA models significantly on both tasks in terms of classification accuracy.
- The DANN approach is also stable and robust on the cross-scenario task.

Reducing EEG Subject Variability by Domain Generalization with Deep Adversarial Network

Bo-Qun Ma, He Li, Wei-Long Zheng, and Bao-Liang Lu，Proc. ICONIP2019

Reducing the Subject Variability

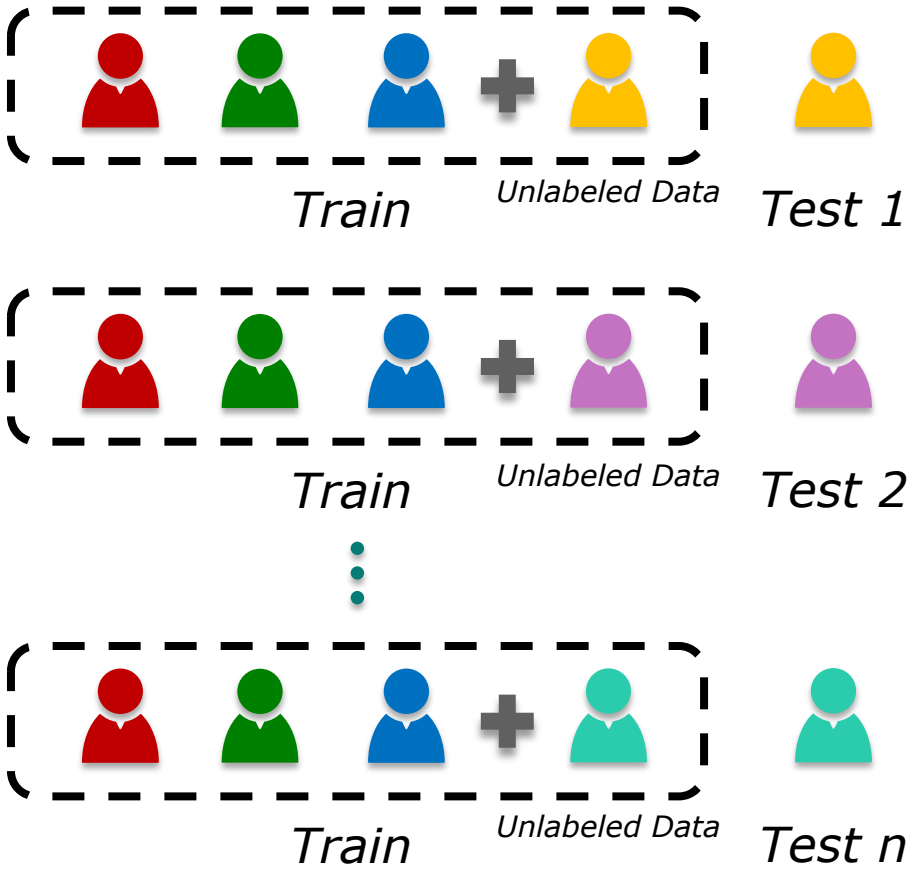
- Subject-dependent (Domain Adaptation)
  - Utilize a small amount of individualized training data
  - Require an entire training session for each test subject



Wei C S, Lin Y P, Wang Y T, et al. A subject-transfer framework for obviating inter-and intra-subject variability in EEG-based drowsiness detection. NeuroImage, 2018, 174: 407-419.

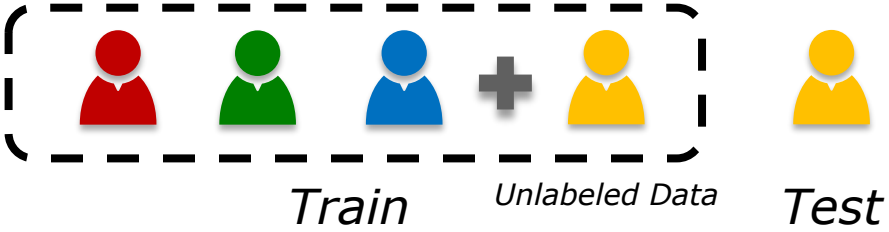
Limitations

- More new subjects means more models to train
- Subject domain information loss in traditional transfer learning problem
- Previous training provides little intuition for the new domain feature extractor

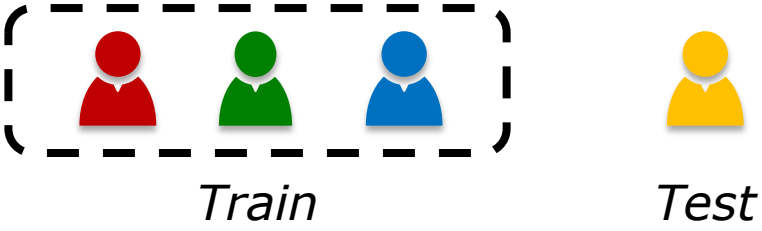


Reducing the Subject Variability

- Subject-dependent (Domain Adaptation)
  - Utilize a small amount of individualized training data
  - Require an entire training session for each test subject



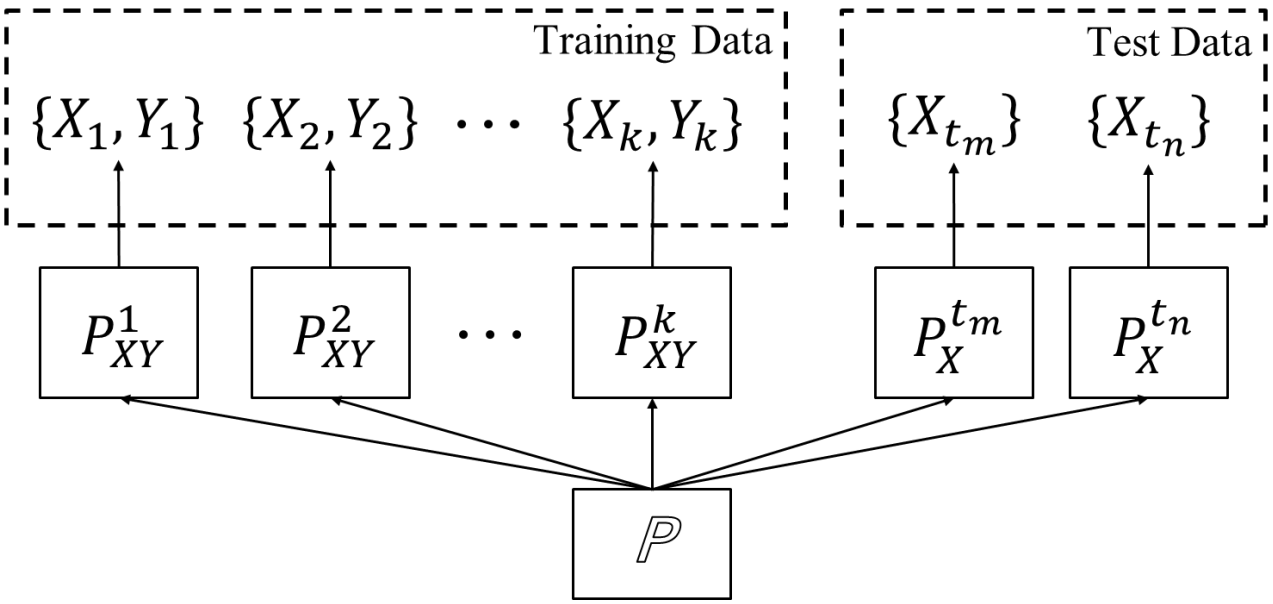
- Subject-independent (Domain Generalization)
  - No data from the new user
  - A robust feature-extraction method



Wei C S, Lin Y P, Wang Y T, et al. A subject-transfer framework for obviating inter-and intra-subject variability in EEG-based drowsiness detection. NeuroImage, 2018, 174: 407-419.

Domain Generalization

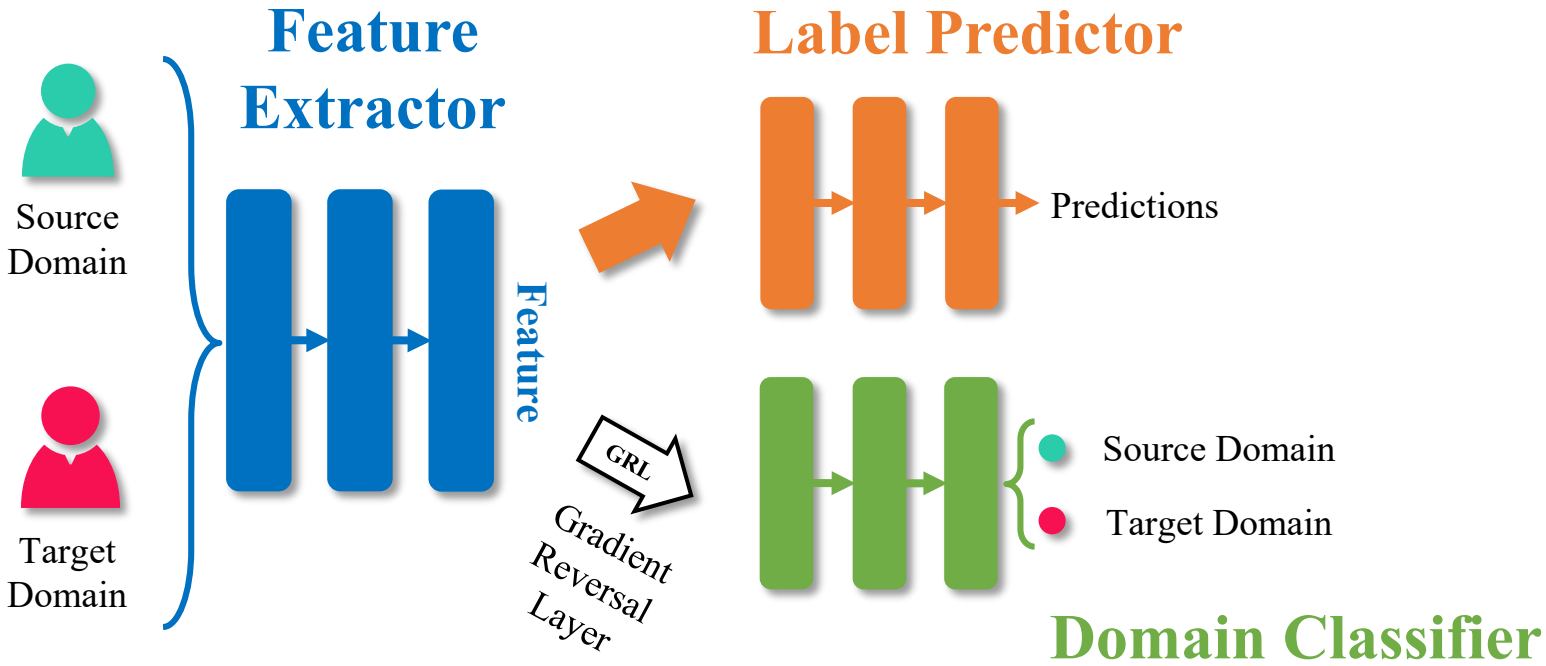
- Domain generalization considers how to take knowledge acquired from an arbitrary number of related domains, and apply it to previously unseen domains
- Estimate a functional relationship that handles changes in the marginal P(X) or conditional P(Y |X) well



Muandet K, Balduzzi D, Schölkopf B. Domain generalization via invariant feature representation, Proc. ICML, 2013

Domain Generalization with Deep Adversarial Network

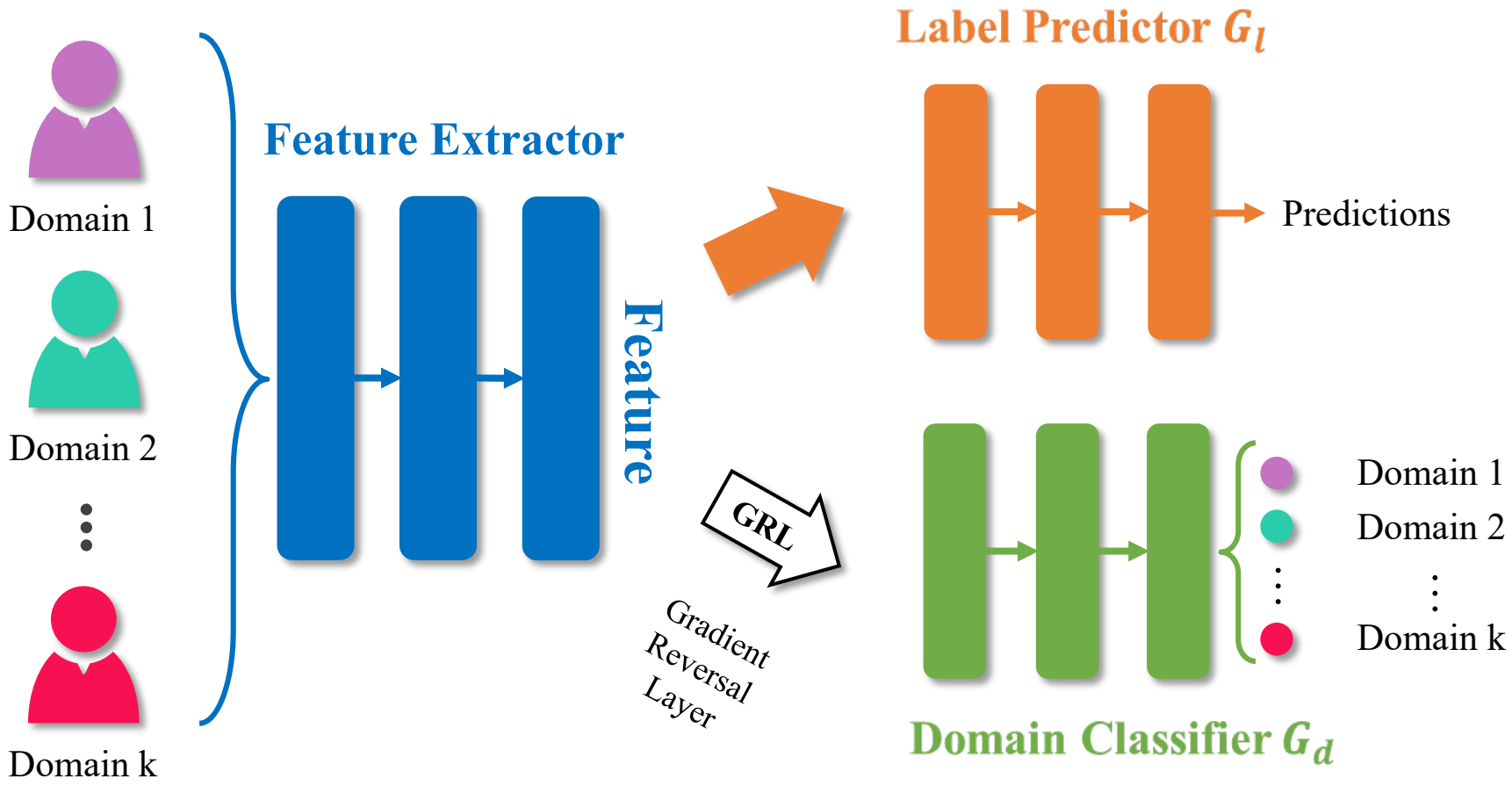
- Our goal is to train a robust feature extractor (without the data from the new subject) which can :
  1. Keep essential information for label prediction
  2. Reduce the domain shift (features from both domains share similar distributions in the new feature space)



Ganin Y, Ustinova E, Ajakan H, et al. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 2016, 17(1): 2096-2030.

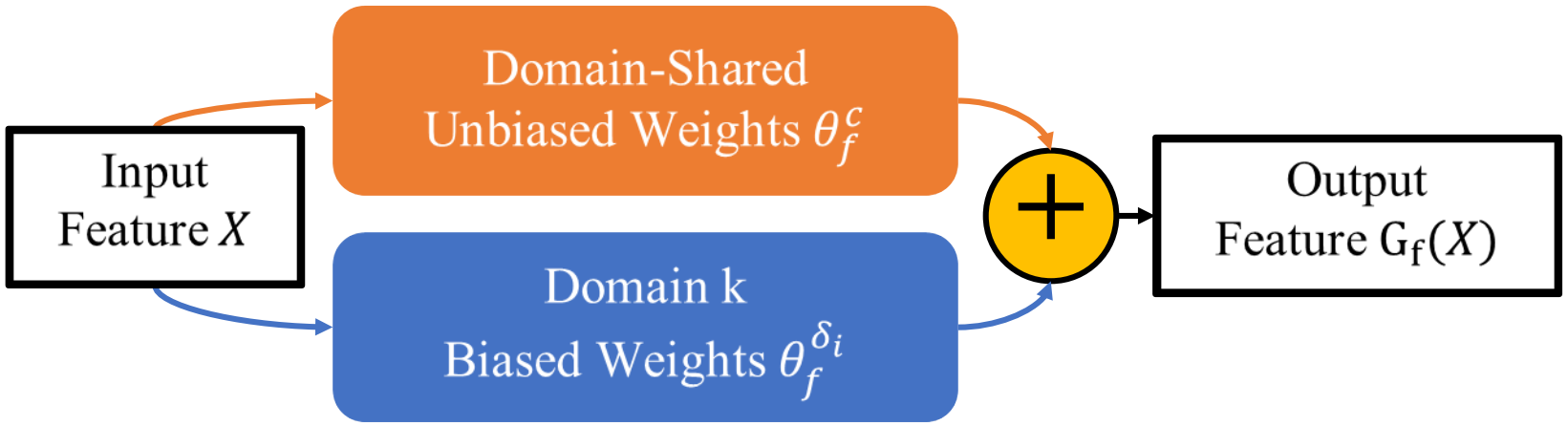
Subject-independent Approach: DG-DANN

- One way is to find the domain-invariant feature space with the information of multiple source domains

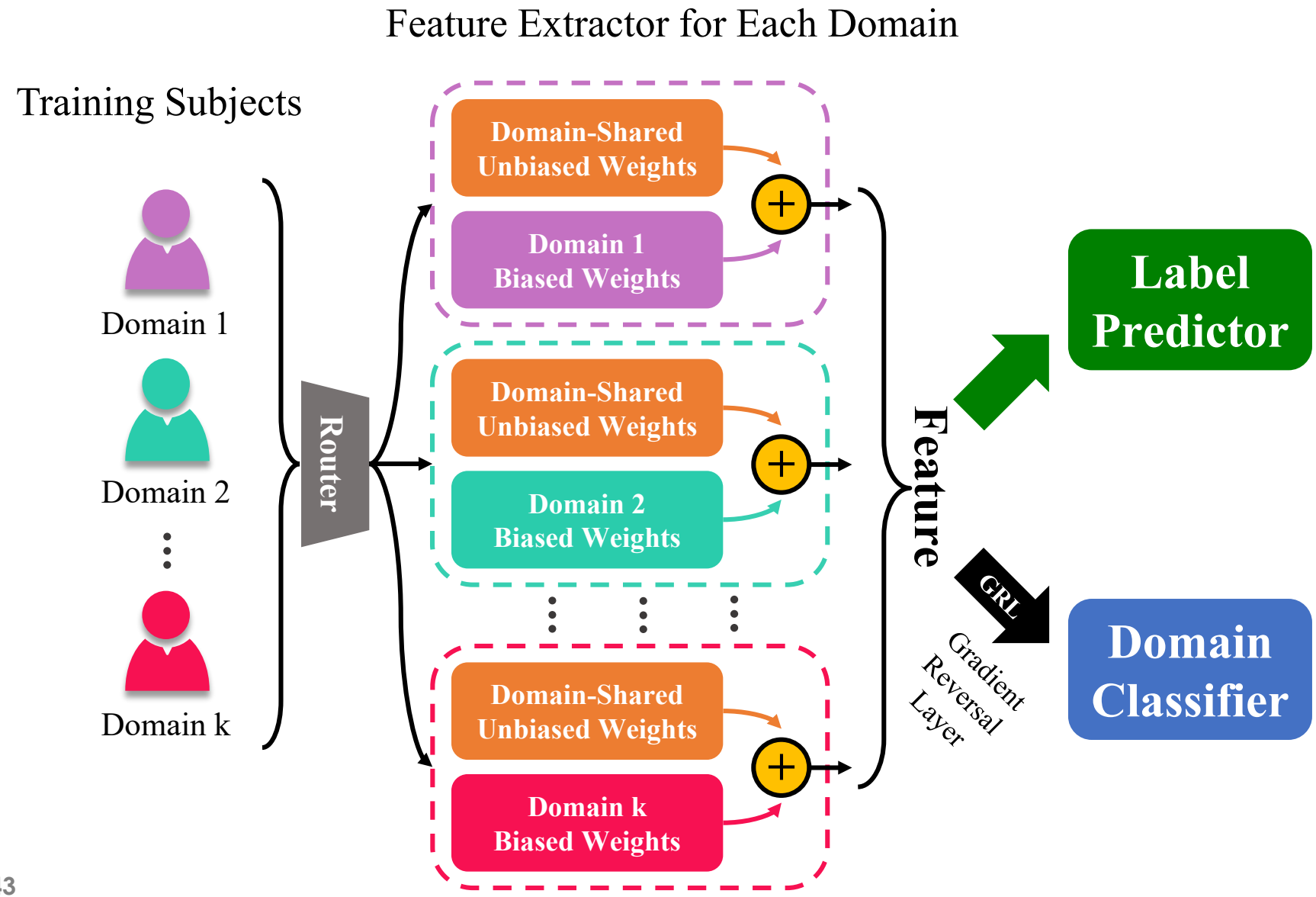


Modifying the Feature Extractor

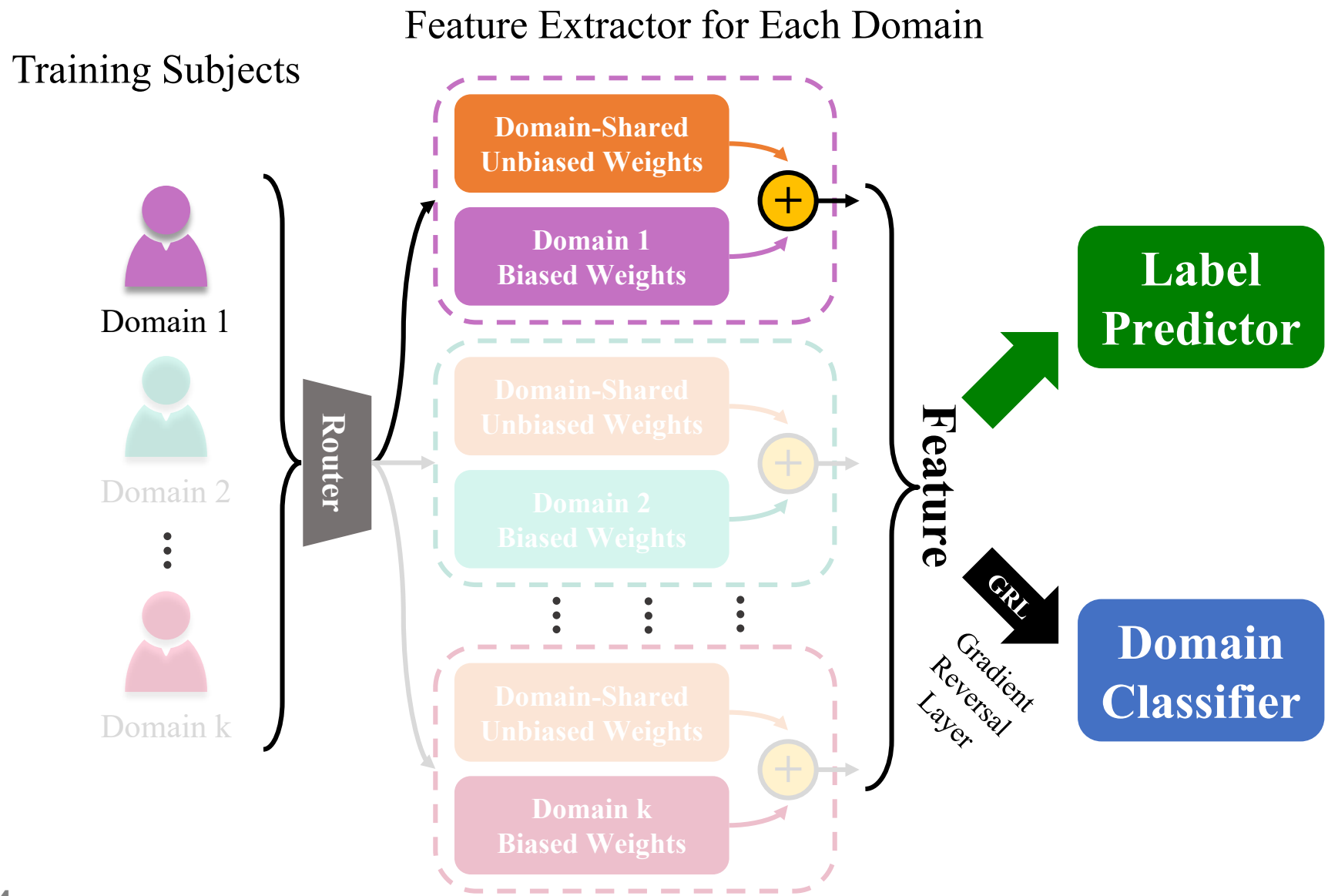
- Another way is to regulate the model parameters
- We assume that the features consist of two parts :
  1. Common space information, which is learnt by the shared unbiased weights θ_f^c
  2. Domain-specific information, which is explicitly defined



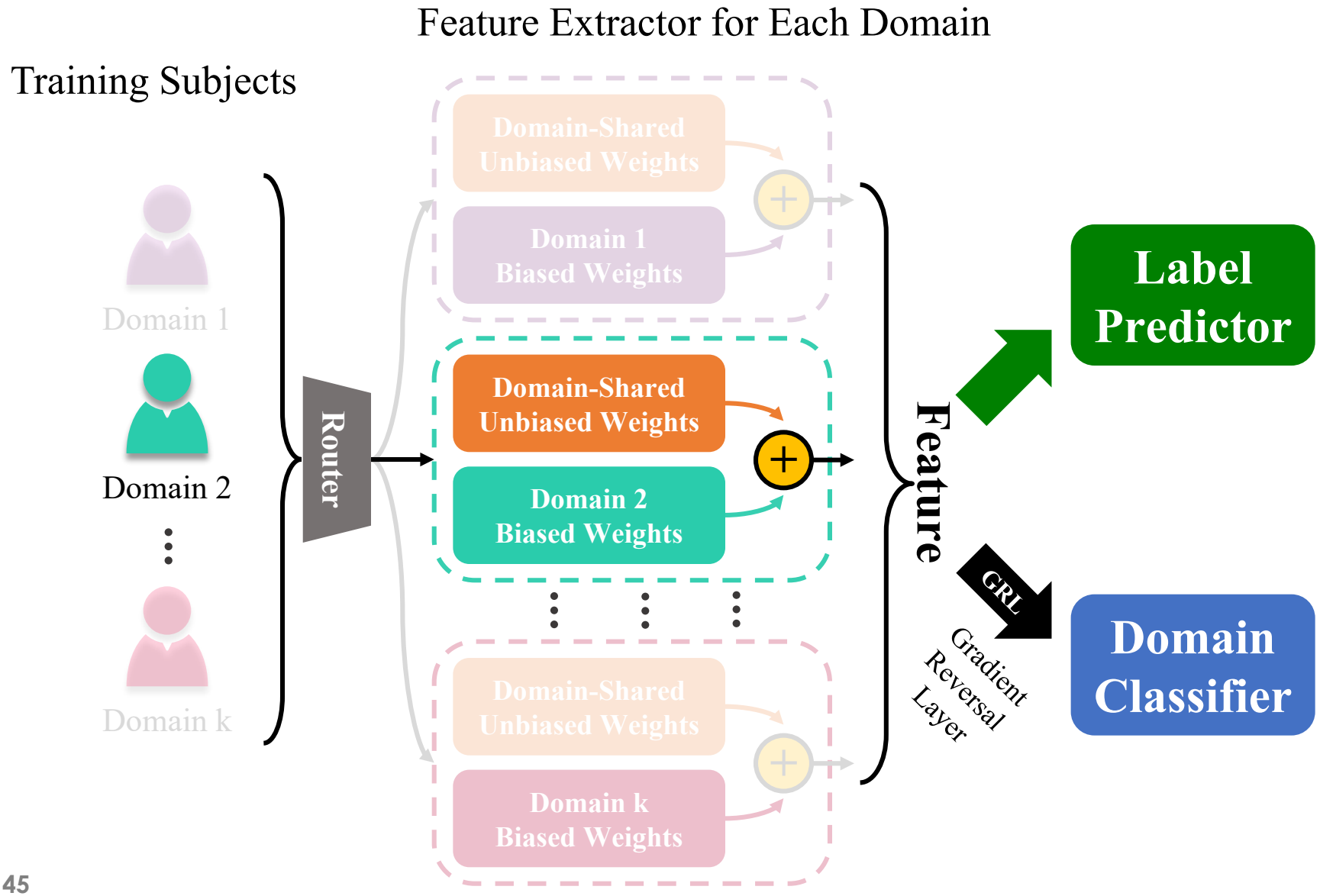
Domain Residual Network: Training



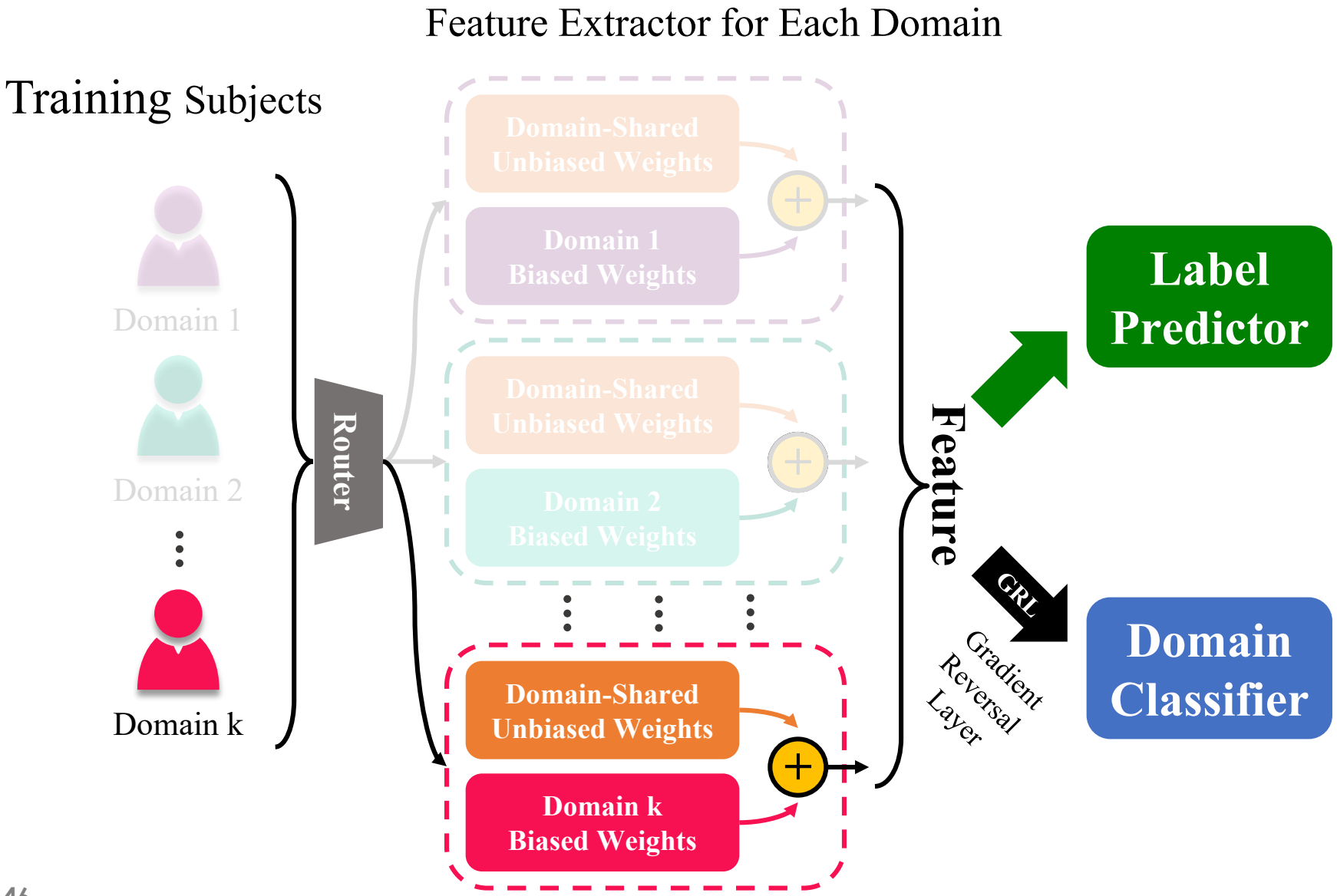
Domain Residual Network: Training



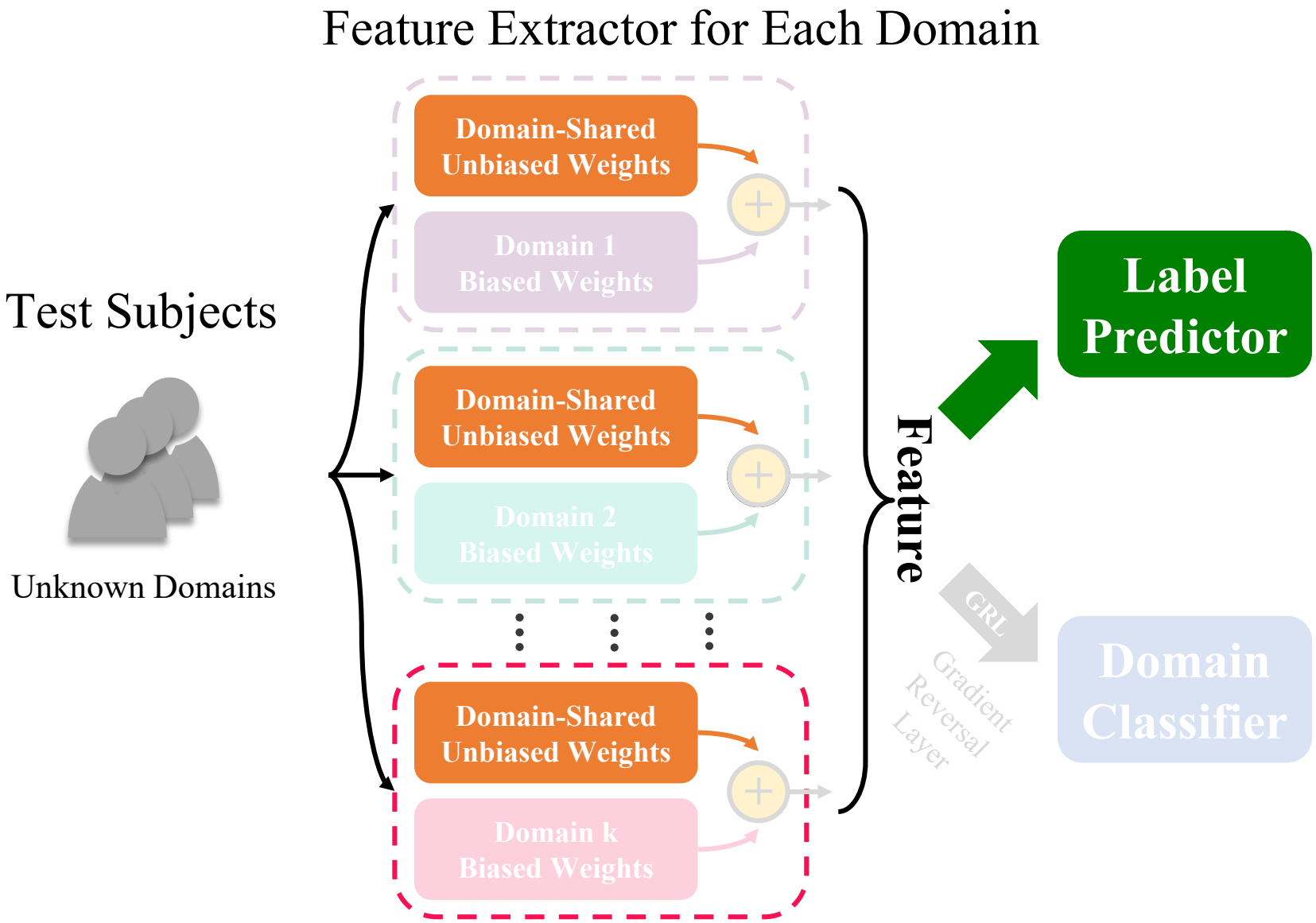
Domain Residual Network: Training



Domain Residual Network: Training

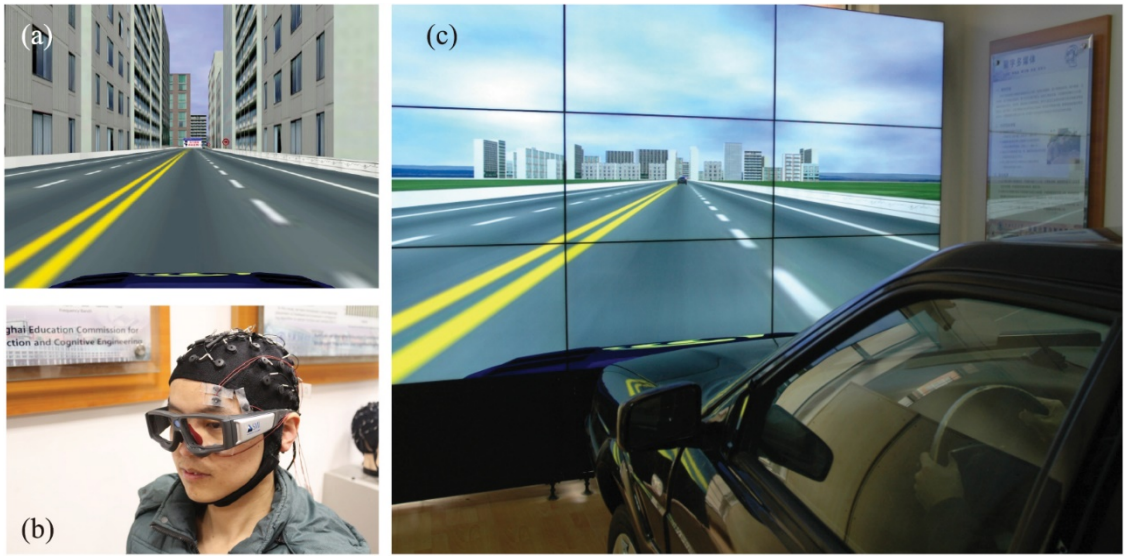


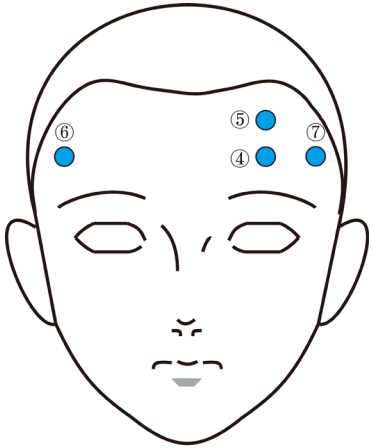
Domain Residual Network: Testing



Experiment: Vigilance Estimation

- The SEED-VIG dataset is a publicly available dataset. It contains 23-subject EEG and EOG signals recorded during simulated driving procedures.
- The vigilance levels are labeled in form of PERCLOS indices
- 885*136 features for each subject





(b) Forehead EOG

Wei-Long Zheng and Bao-Liang Lu, A multimodal approach to estimating vigilance using EEG and forehead EOG. Journal of Neural Engineering, 14(2): 026017, 2017.

Result : Vigilance Estimation

- Leave-one-subject-out cross validation:
  - 22 subjects for training
  - 1 subject for testing

		Baseline	Domain Adaptation				Domain Generalization		
		SVR	**TCA**	**GFK**	**DANN**	**ADDA**	**DICA**	**DG-DANN**	**DResNet**
PCC	Avg	0.7606	0.7786	0.7907	0.8402	**0.8442**	0.7733	0.8320	0.8440
	Std	0.2314	0.2152	0.1260	0.1535	0.1336	0.1382	0.1000	**0.0935**
RMSE	Avg	0.1689	0.1596	0.1910	0.1427	**0.1405**	0.2007	0.1470	0.1420
	Std	0.0673	0.0544	0.0636	0.0588	0.0514	0.0674	0.0444	**0.0402**

He Li, Wei-Long Zheng, Bao-Liang Lu, Multimodal Vigilance Estimation with Adversarial Domain Adaptation Networks, Proc. IEEE IJCNN2018

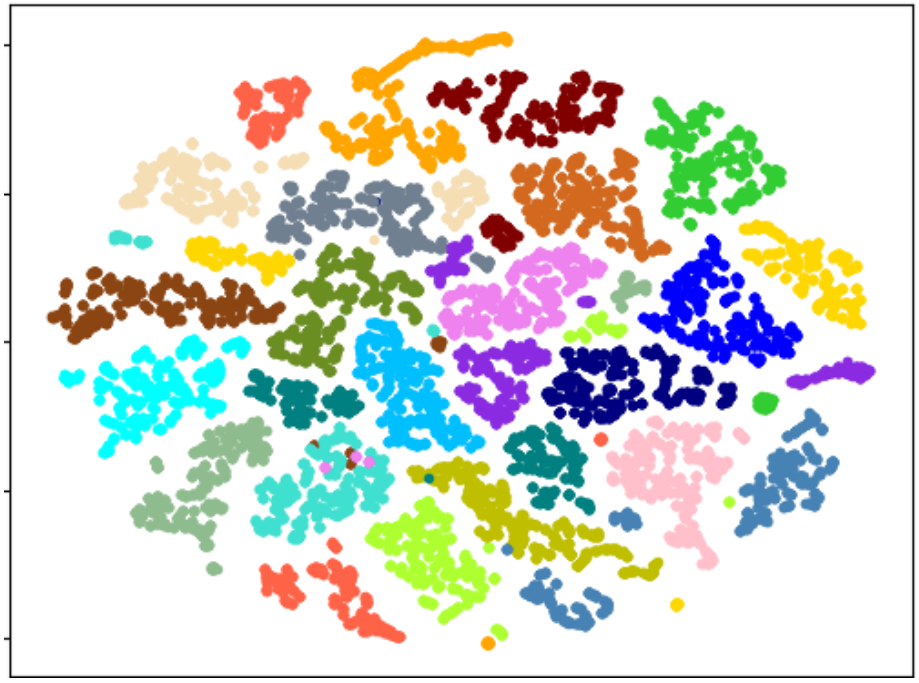
Result : Vigilance Estimation

- Leave-multiple-random-subjects-out evaluation:
  - 2/3 of the subjects (15) for training
  - 1/3 of the subjects (8) for testing

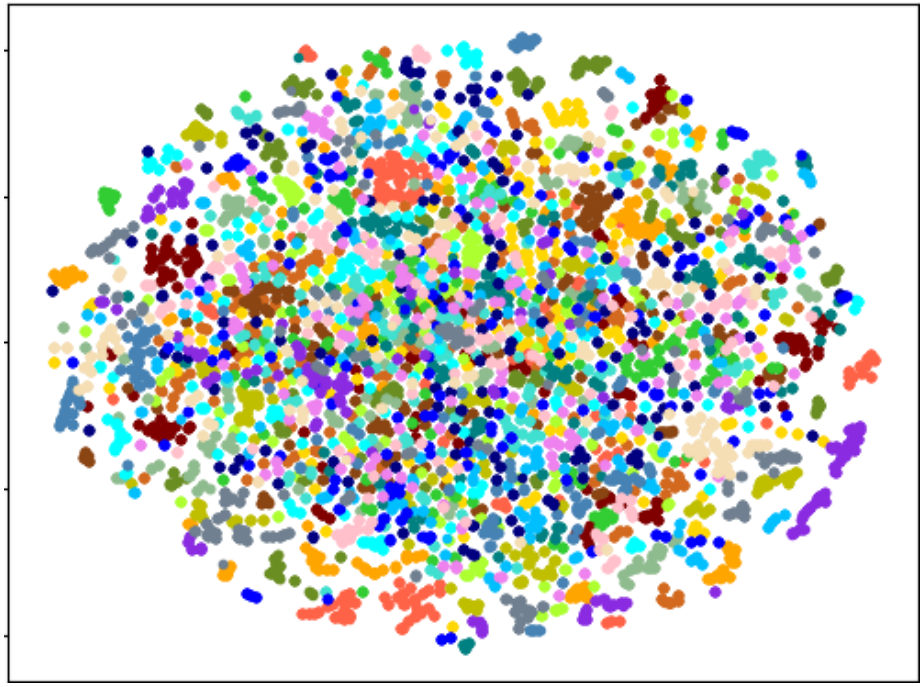
		Baseline	Domain Generalization		
		SVR	**DICA**	**DG-DANN**	**DResNet**
PCC	Avg	0.7499	0.7719	0.8294	**0.8386**
	Std	0.1980	0.1841	0.1541	**0.1532**
RMSE	Avg	0.2068	0.1735	0.1604	**0.1569**
	Std	0.0587	**0.0468**	0.0782	0.0735

Visualized Feature for SEED-VIG

Raw Feature

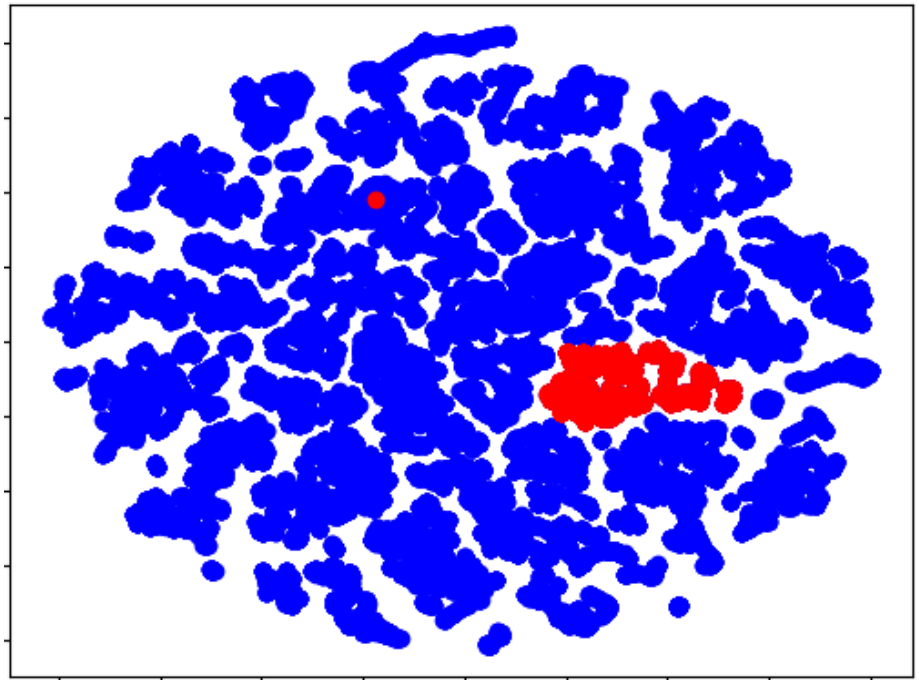


DResNet Feature

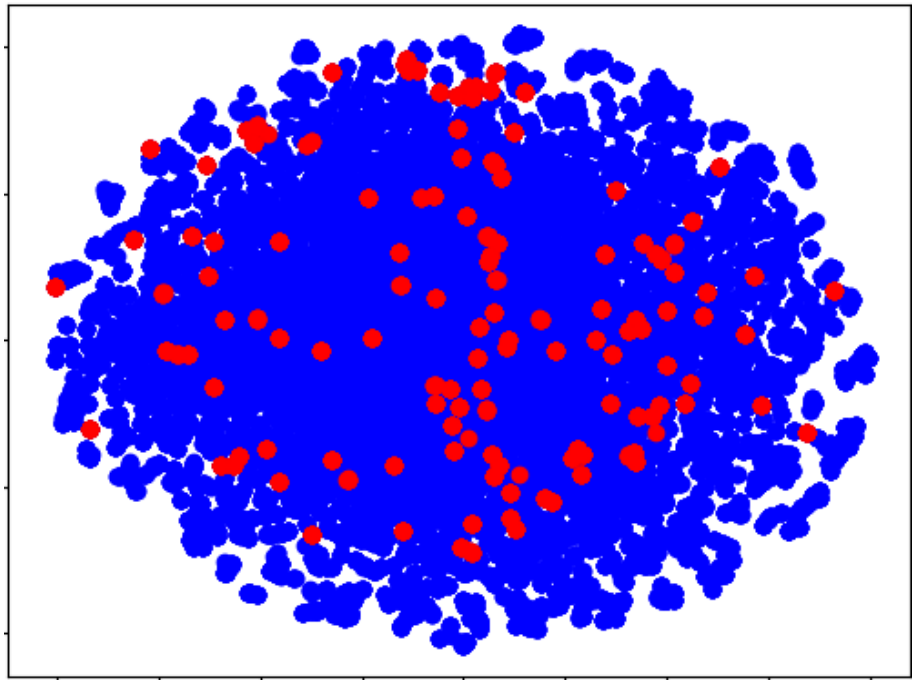


Visualized Feature for SEED-VIG

Raw Feature

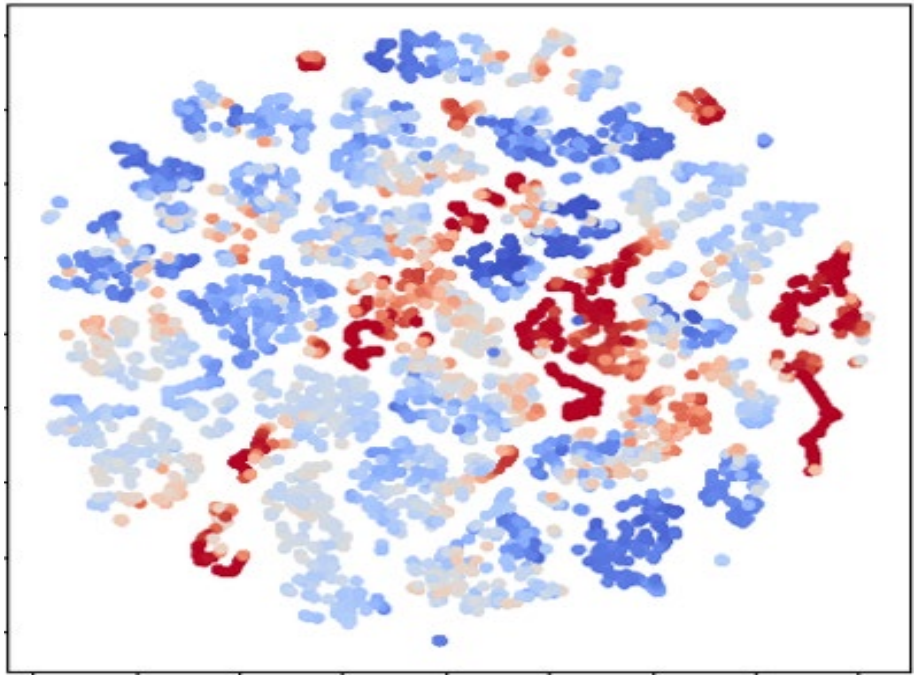


DResNet Feature

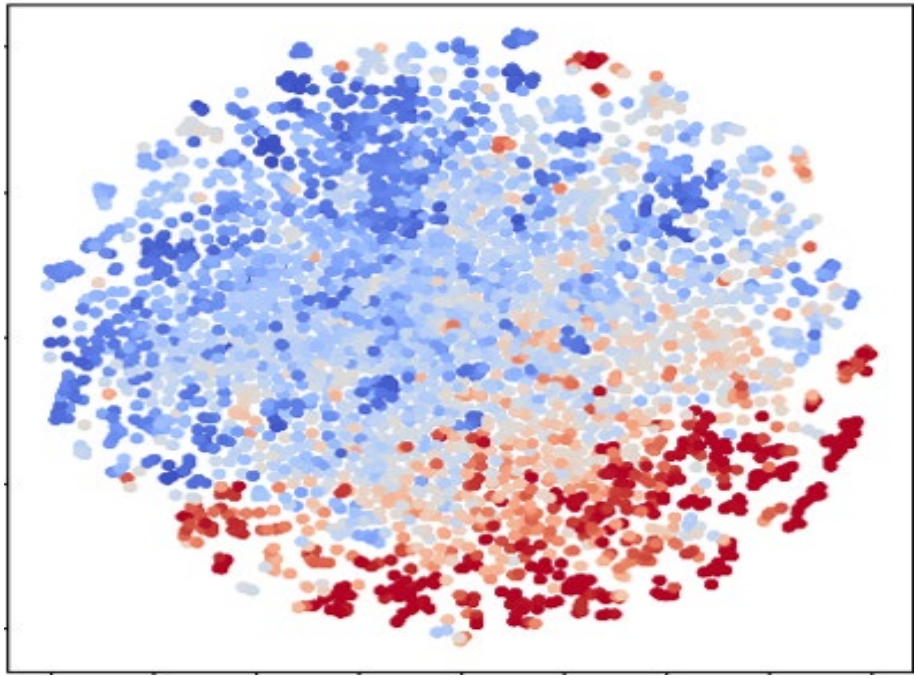


Visualized Feature for SEED-VIG

Raw Feature

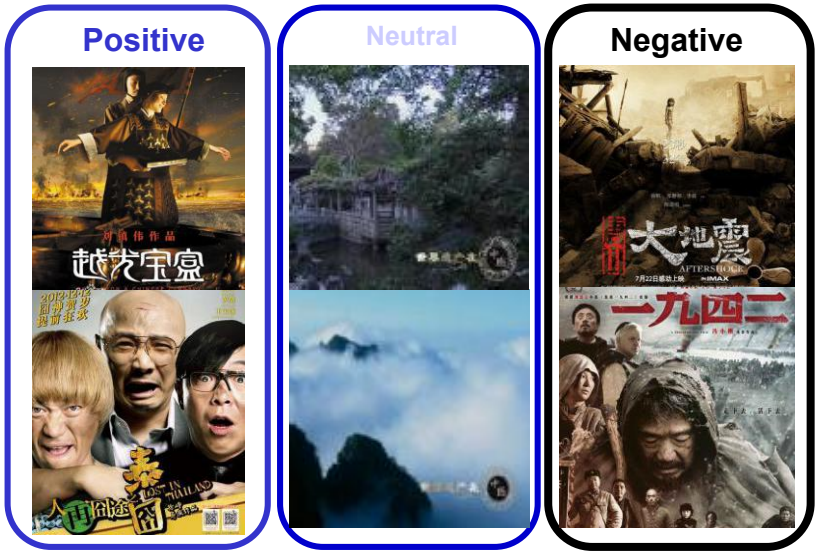


DResNet Feature



Experiment: Emotion Recognition

- 15 Chinese movie clips were used in each experiment, 5 for each emotion (positive, negative, neutral)
- 15 subjects (7 females), each participates in the experiment for three times
- 3394*310 features for each subject



Zheng, W.L., Lu, B.L.: Investigating Critical Frequency Bands and Channels for EEG-based Emotion Recognition with Deep Neural Networks. IEEE Transactions on Autonomous Mental Development 7(3), 162{175 (2015)

Result: Emotion Recognition

- Leave-one-subject-out cross validation:
  - 14 subjects for training
  - 1 subject for testing

	Base line	Domain Adaptation					Domain Generalization			
	SVM	**TCA**	**TPT**	**DANN**	**DAN**	**WGA NDA**	**DICA**	**SCA**	**DG-DANN**	**DResNet**
Avg	0.5818	0.6400	0.7517	0.7919	0.8381	**0.8707**	0.6941	0.6633	0.8430	0.8530
Std	0.1385	0.1466	0.1283	0.1314	0.0856	**0.0714**	0.0779	0.1060	0.0832	0.0797

Luo Y, Zhang S Y, Zheng W L, et al. WGAN Domain Adaptation for EEG-Based Emotion Recognition[C]//International Conference on Neural Information Processing. Springer, Cham, 2018: 275-286.
Li H, Jin Y M, Zheng W L, et al. Cross-Subject Emotion Recognition Using Deep Adaptation Networks[C]//International Conference on Neural Information Processing. Springer, Cham, 2018: 403-413.

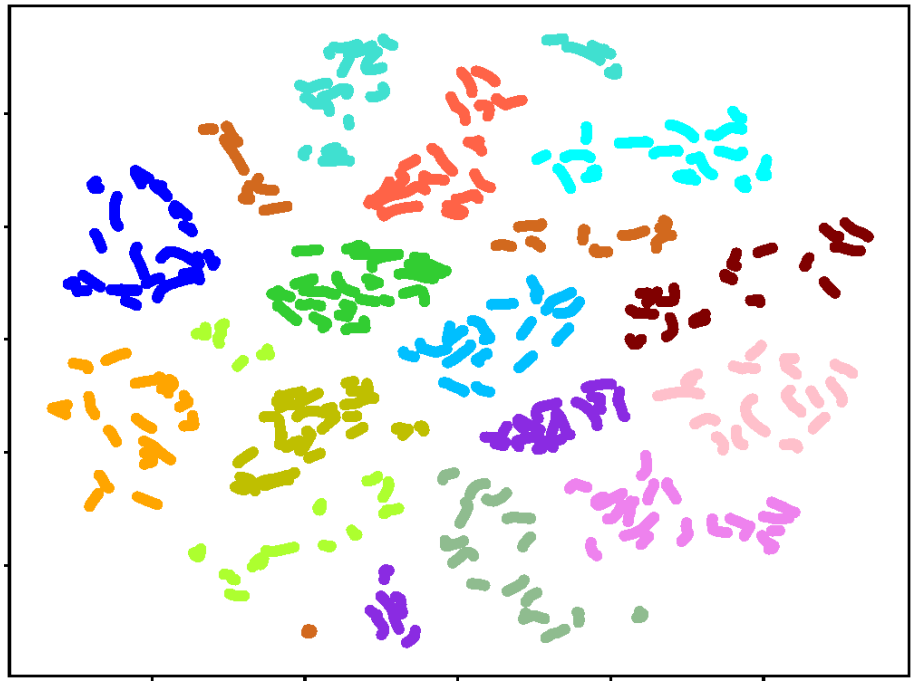
Result: Emotion Recognition

- Leave-multiple-subject-out evaluation:
  - Two-thirds of the subjects (10) for training
  - One-third of the subjects (5) for testing

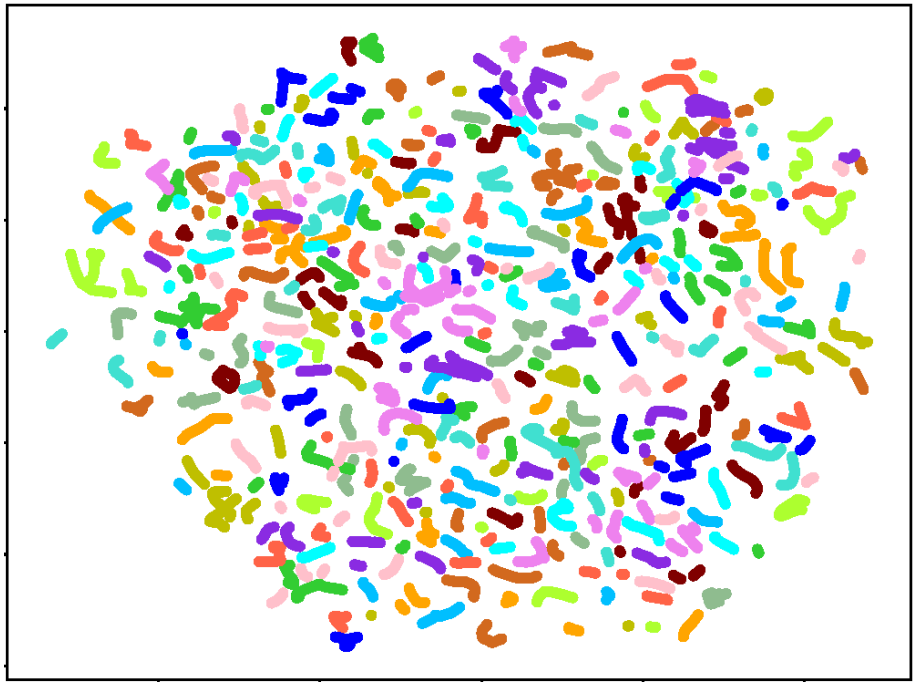
	Baseline	Domain Generalization			
	SVM	**DICA**	**SCA**	**DG-DANN**	**DResNet**
Avg	0.5413	0.6435	0.6083	0.8146	**0.8170**
Std	0.1348	0.0896	**0.0505**	0.0788	0.0737

Visualized Feature for SEED

Raw Feature

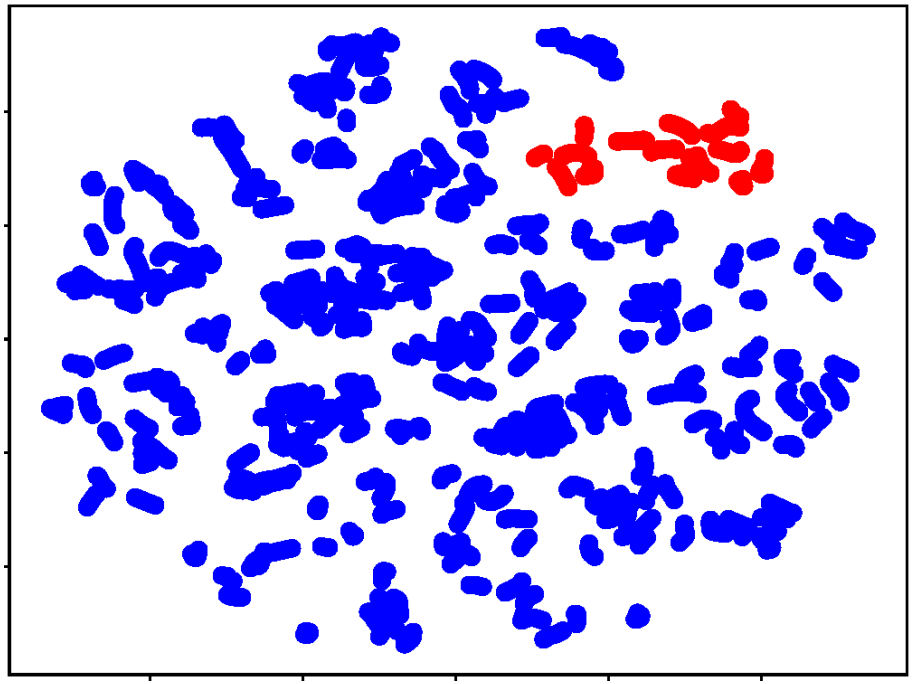


DResNet Feature

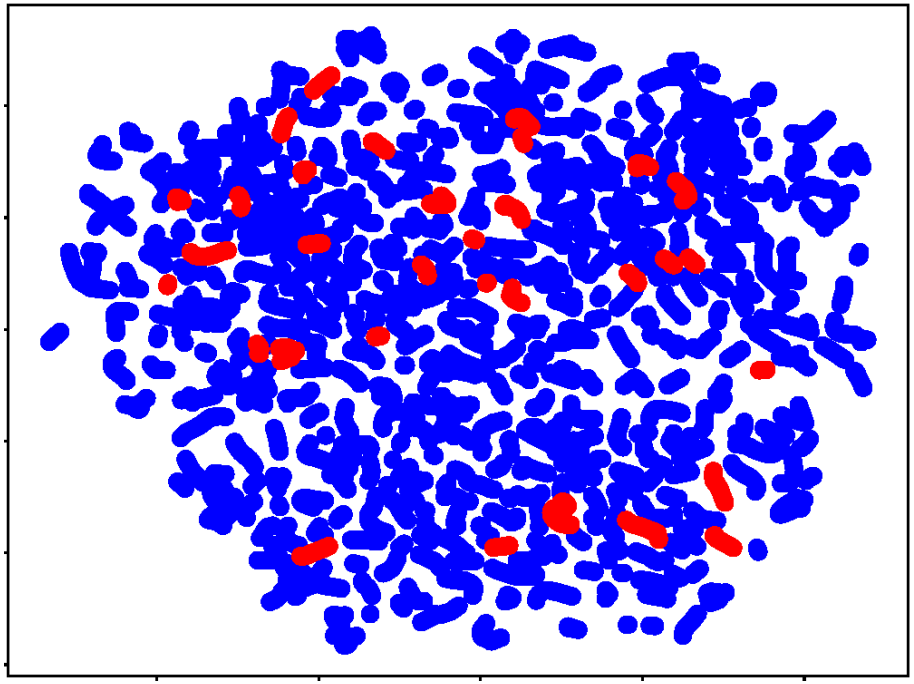


Visualized Feature for SEED

Raw Feature

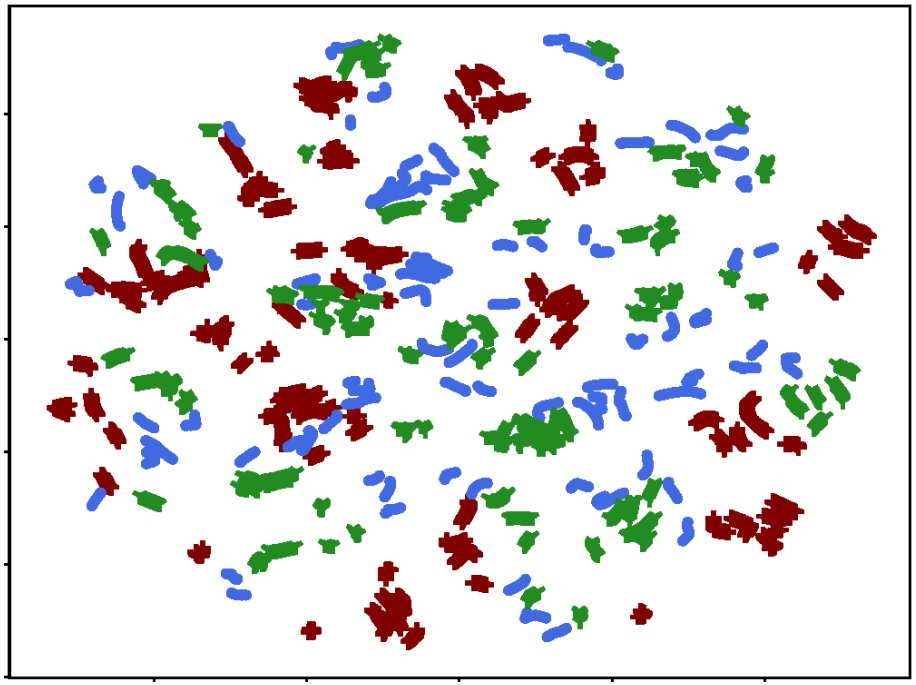


DResNet Feature

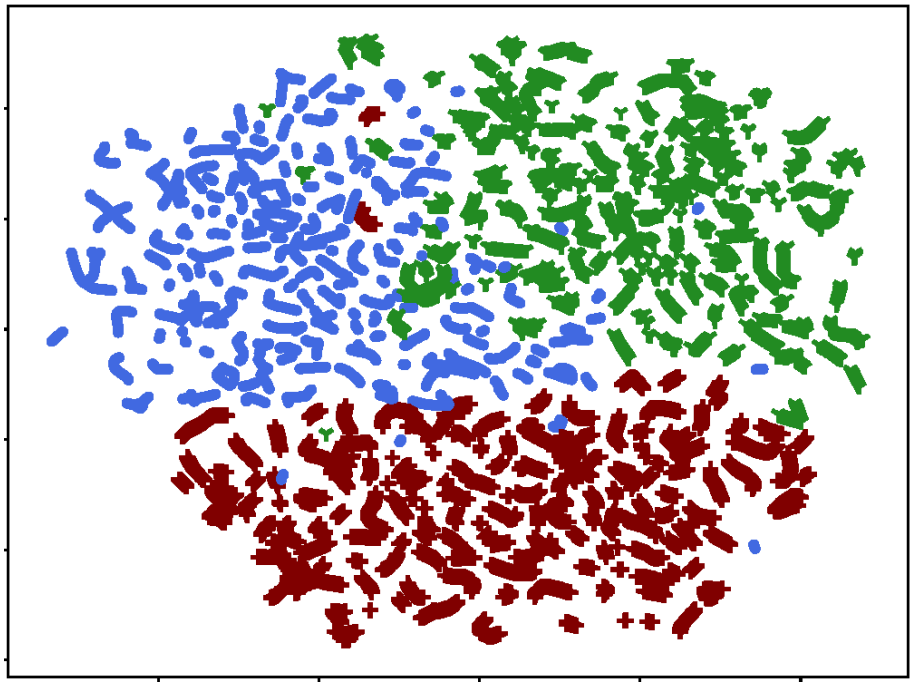


Visualized Feature for SEED

Raw Feature



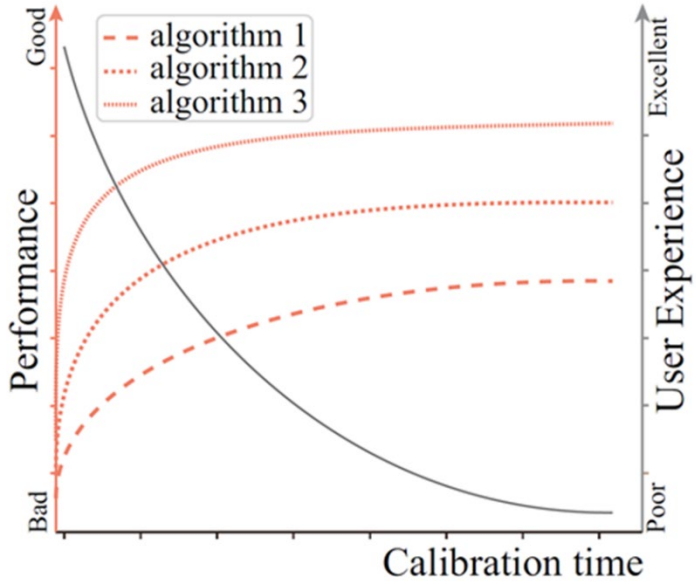
DResNet Feature

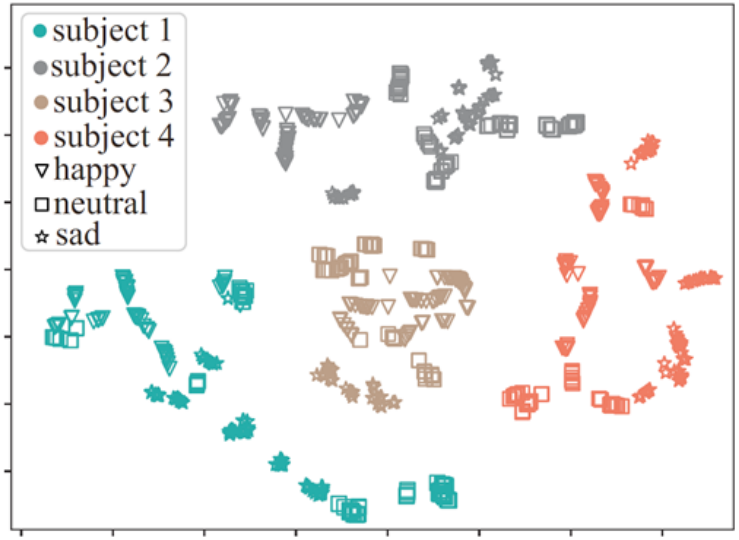


Conclusion

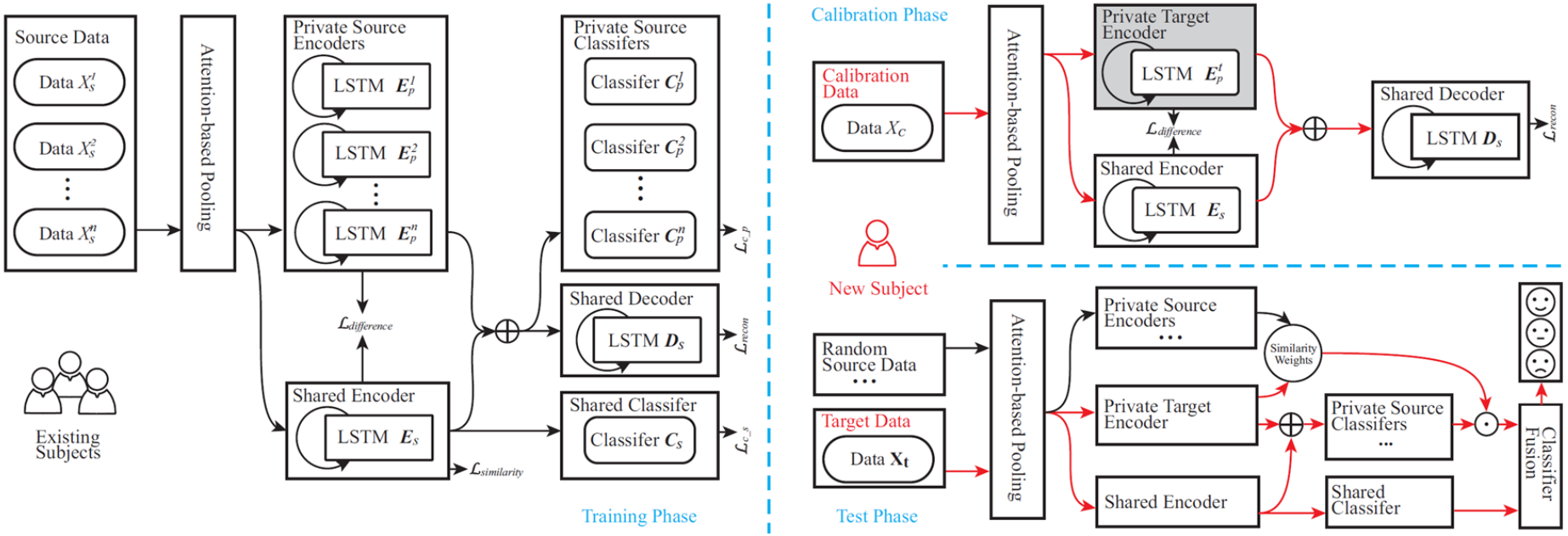
- We focused on reducing the influence of EEG subject variability on BCI systems for unknown subjects by introducing the idea of domain generalization
- Following different approaches to domain generalization, two novel deep adversarial DG models have been proposed
- Evaluations under two settings on public datasets of different topics have indicated that our proposed DG methods are effective for reducing subject variability on depersonalized cross-subject vigilance estimation and emotion recognition problem

跨被试脑电情绪识别模型性能与系统校准时间的平衡





即插即用域适应(Plug-and-Play Domain Adaptation)



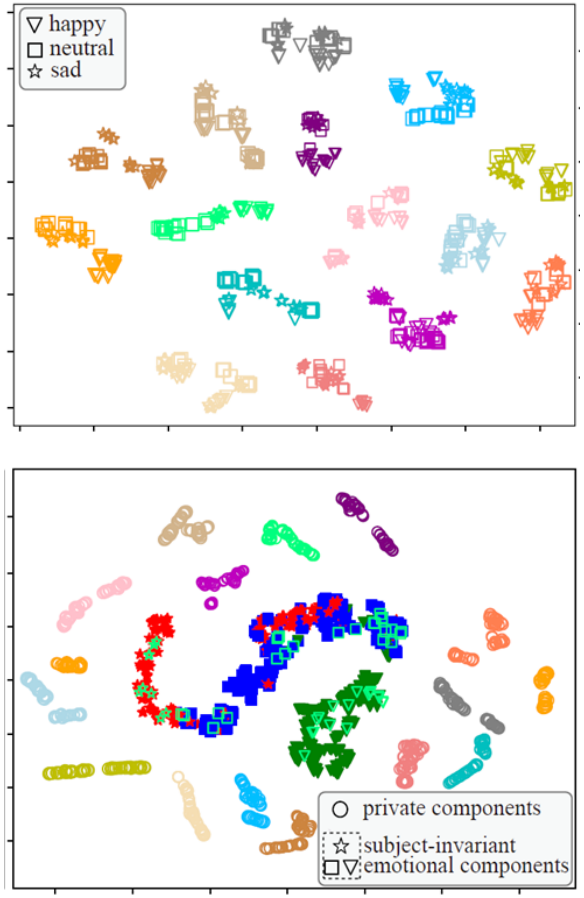
1) 模型分为三个部分：训练、校准与测试。
2) 注意力机制用于寻找与情绪识别相关的关键脑区与关键频段。

Li-Min Zhao, Xu Yan, and Bao-Liang Lu, Plug-and-Play Domain Adaptation for Cross-Subject EEG-based Emotion Recognition accepted by ***AAAI 2021***

即插即用域适应PPDA的实验结果

Methods	#ATD	Avg.	Std.
Baseline SVM (Zheng and Lu 2016)	None	0.567	0.163
DICA (Ma et al. 2019)	None	0.694	0.078
DResNet (Ma et al. 2019)		0.853	0.080
TCA (Zheng and Lu 2016)	All	0.640	0.146
TPT (Zheng and Lu 2016)		0.752	0.128
DANN (Li et al. 2018)		0.792	0.131
DAN (Li et al. 2018)		0.838	0.086
WGANDA (Luo et al. 2018)		0.871	0.071
PPDA_NC	None	0.854	0.071
PPDA	Few	0.867	0.071

注：在本实验中ALL表示使用53分钟的目标数据，Few表示使用45秒的目标。



Useful Sites on Transfer Learning

- Resources:
- Open source program: http://www.cse.ust.hk/TL/
- Qiang Yang: http://www.cs.ust.hk/~qyang/
- Sinno Jialin Pan: http://www.ntu.edu.sg/home/sinnopan/
- Wenyuan Dai: http://www.4paradigm.com/homepage.html

- Survey:
- A survey on Transfer Learning.
- Transfer learning for activity recognition: A survey.
- Transitive Transfer Learning.
- Fuzzy Transfer Learning: Methodology and application.

谢谢！ 下周见！