Transfer

A Survey on Deep Transfer Learning

ONE Deep Transfer Learning

TWO Categories

THREE Conclusion

PART 1

Deep Transfer Learning

- Introduction
- Traditional ML vs. TL
- Notations

Introduction

Transfer learning:

- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks (in new d omains)

Deep learning :

- Attempt to learn high-level features from mass data

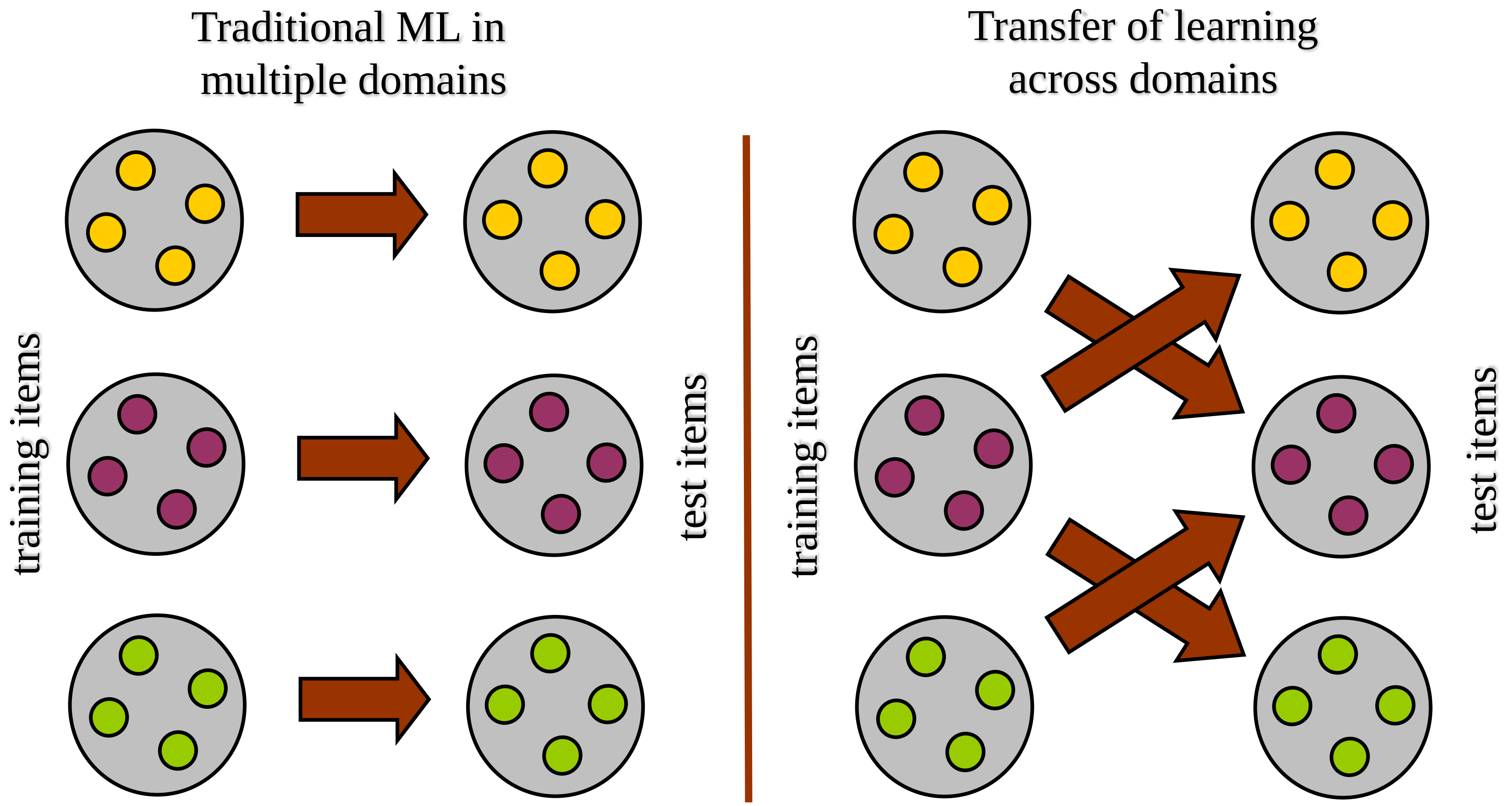
Why transfer:

- Deep learning has a very strong dependence on massive training data
- Insufficient training data is a inescapable problem
- In some domains, the learning process is time consuming

Deep transfer learning:

- relaxes the hypothesis that the training data must be independent and identically distributed (i.i.d.) with the test data

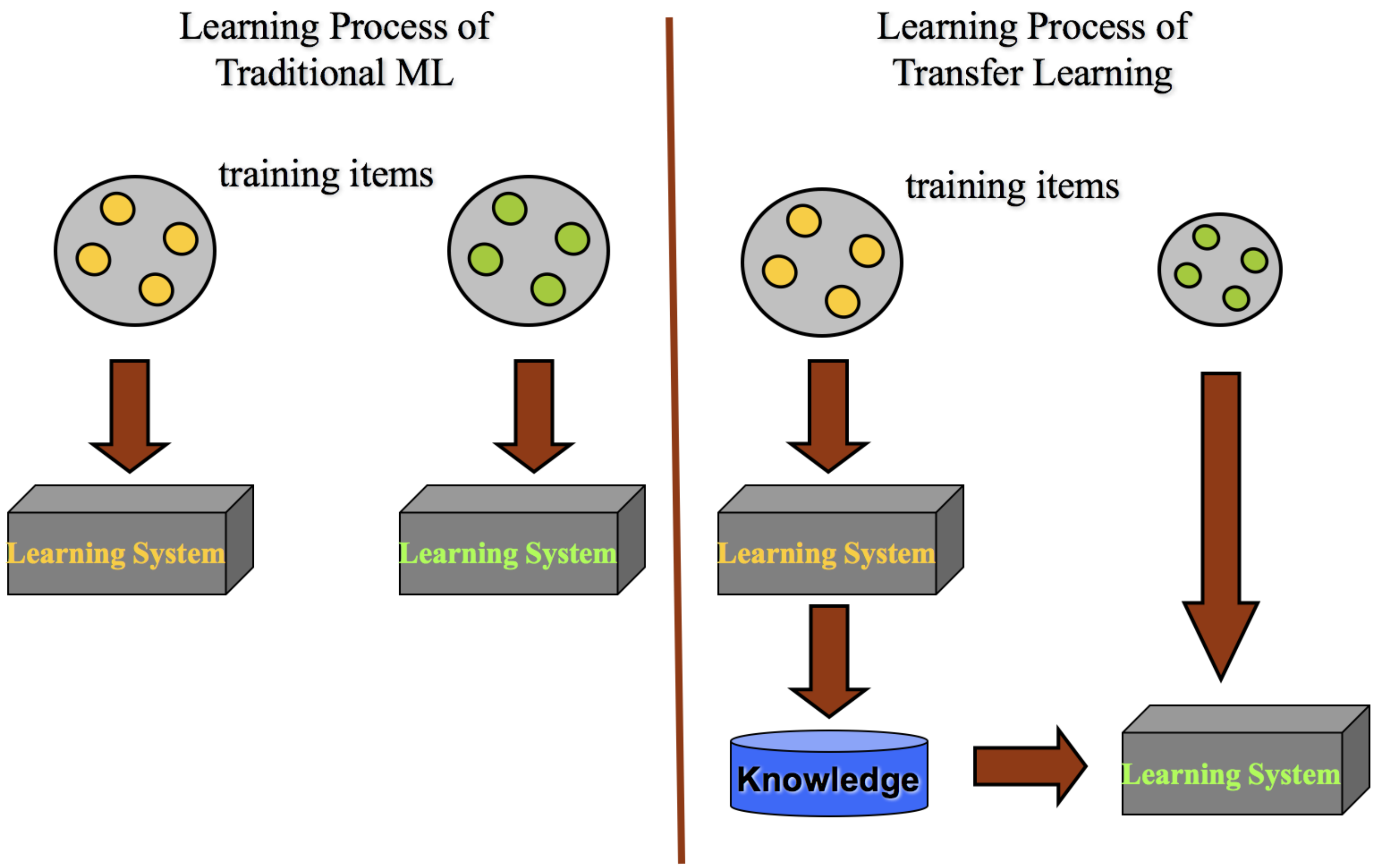
Traditional ML vs. TL



Humans can learn in many domains.

Humans can also transfer from one domain to other domains.

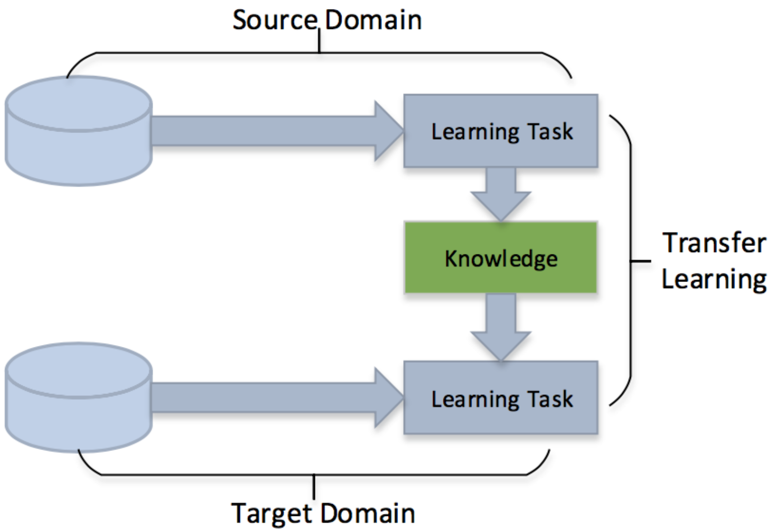
Traditional ML vs. TL



Notations

Domain: $\mathcal{D} = \{\chi, P(X)\}$

- A feature space χ
- edge probability distribution $\mathcal{P}(X)$, 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 edge probability distributions.

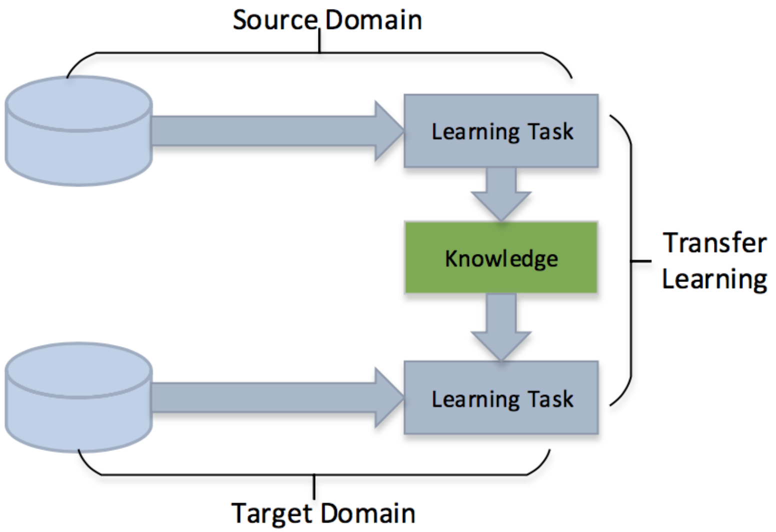


Task: $\mathcal{T} = \{y, f(x)\}$

- label space : y
- target prediction function / conditional probability function : $f(x)$ / $P(y|x)$
- In general, if two tasks are different, then they may have different label spac es or different conditional distributions

Notations

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



- Source domain:

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

- Task in the source domain:

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

- Target domain:

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

- Task in the target domain

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

Definition

Definition 1. ***(Transfer Learning).*** *Given a learning task $\mathcal{T}_t$ based on $\mathcal{D}_t$, and we can get the help from $\mathcal{D}_s$ for the learning task $\mathcal{T}_s$. Transfer learning aims to improve the performance of predictive function $f_{\mathcal{T}}(\cdot)$ for learning task $\mathcal{T}_t$ by discover and transfer latent knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$, where $\mathcal{D}_s \neq \mathcal{D}_t$ and/or $\mathcal{T}_s \neq \mathcal{T}_t$. In addition, in the most case, the size of $\mathcal{D}_s$ is much larger than the size of $\mathcal{D}_t$, $N_s \gg N_t$.*

Definition 2. ***(Deep Transfer Learning).*** *Given a transfer learning task defined by $\langle \mathcal{D}_s, \mathcal{T}_s, \mathcal{D}_t, \mathcal{T}_t, f_{\mathcal{T}}(\cdot) \rangle$. It is a deep transfer learning task where $f_{\mathcal{T}}(\cdot)$ is a non-linear function that reflected a deep neural network.*

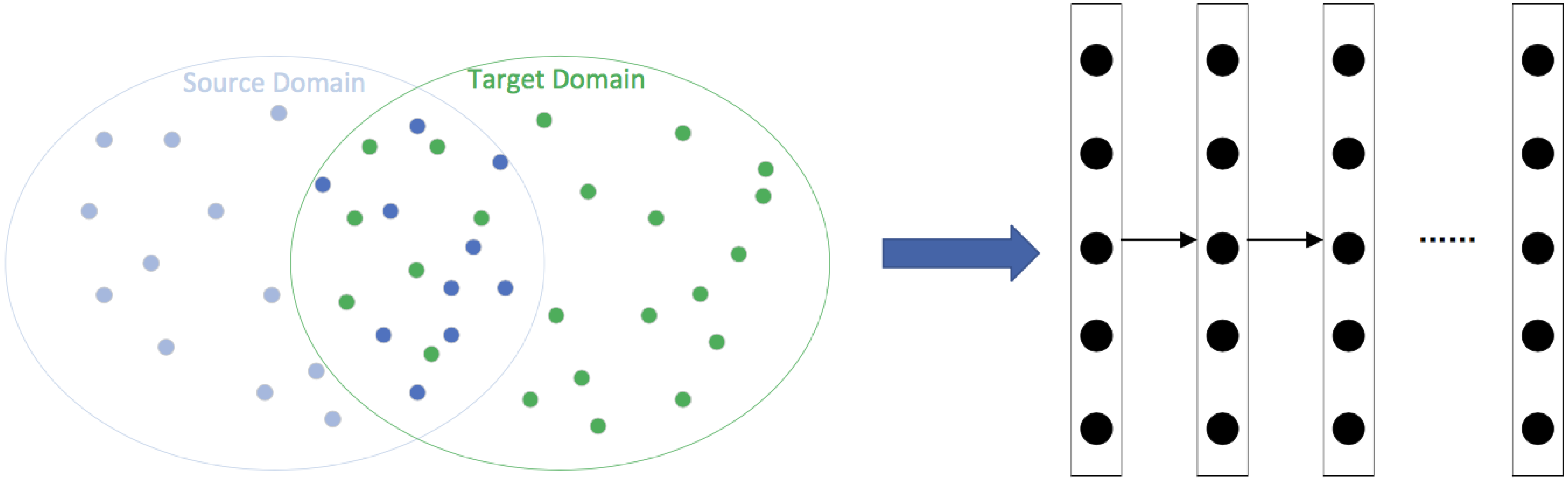
PART 2

Categories

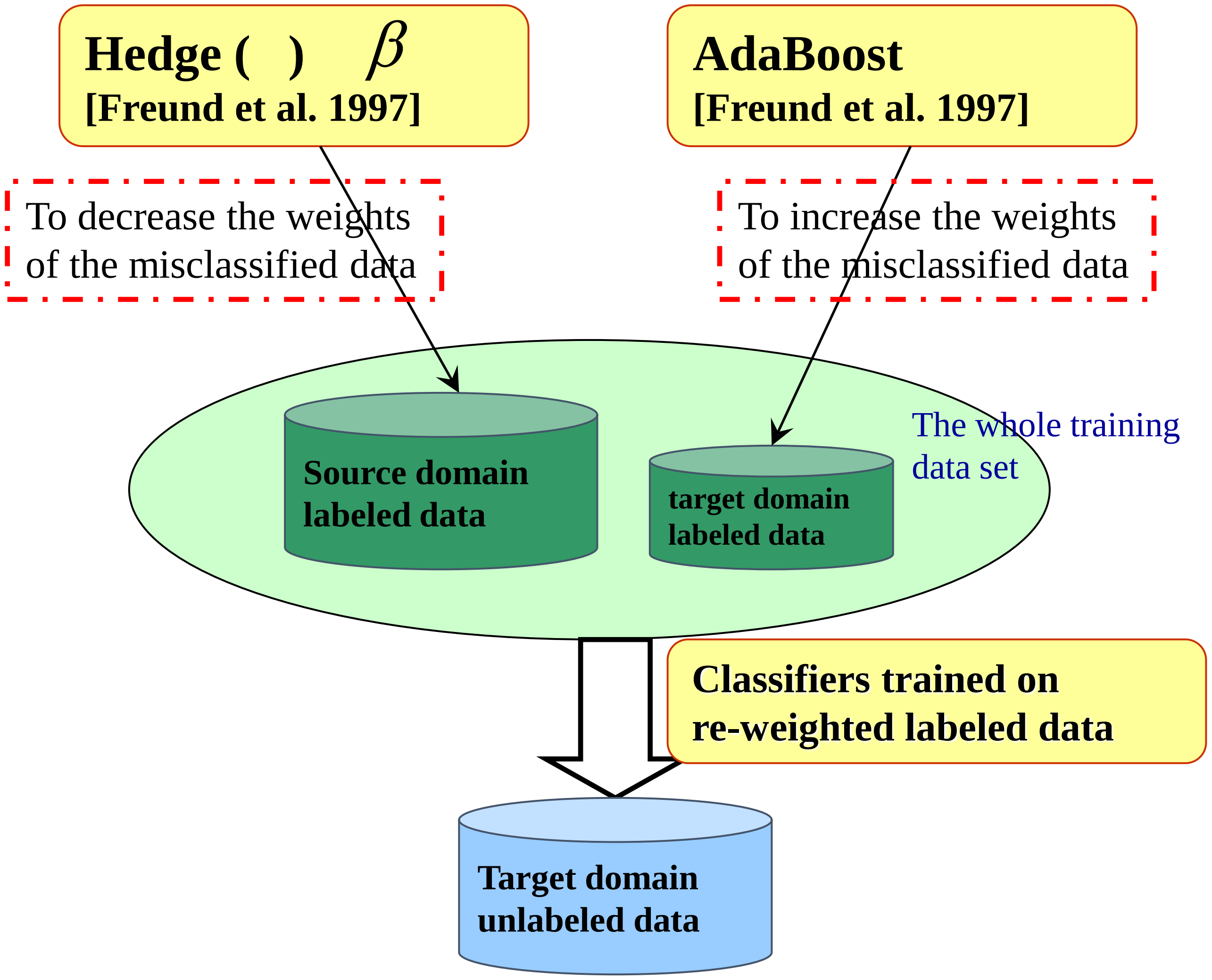
- instances-based
- mapping-based
- network-based
- adversarial- based

Instances-based deep transfer learning

- **Motivation** : the source domain and target domain data use exactly the same features and labels.
- **Assumption** : Although the source domain data can not be reused directly, there are some parts of t
  he 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.



TrAdaBoost

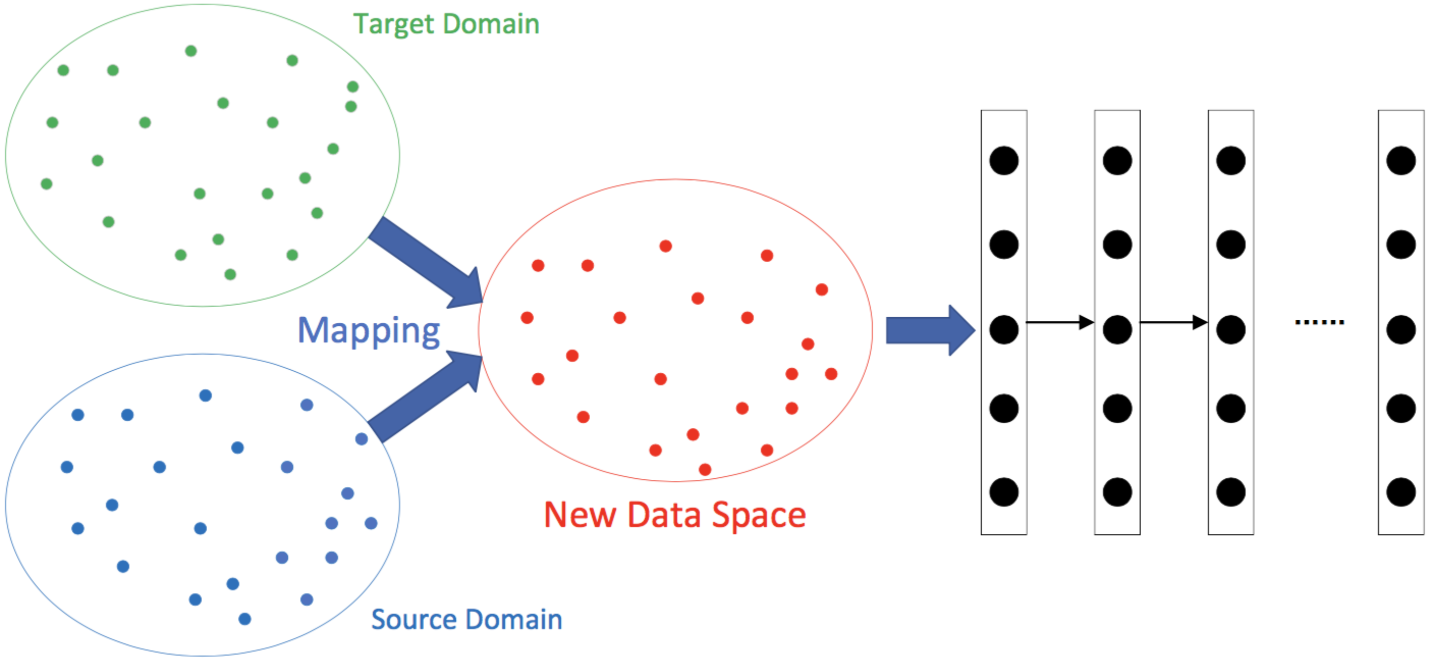


Main idea:

- use AdaBoost-based technology to filter out instances that are dissimilar to the target domain in source domains.
- Re-weighted instances in source domain to compose a distribution similar to target domain.

Mapping-based deep transfer learning

- **Motivation** : In this new data space, instances are similarly and suitable for deep neural network.
- **Assumption** : Although there are different between two origin domains, they can be more similarly i n an elaborate new data space.
- **Main Idea**: mapping instances from the source domain and target domain into a new data space



TCA-based method

transfer component analysis (TCA):

- TCA tries to learn some transfer components across domains in a reproducing kernel Hilbert space using maximum mean miscrepancy.
- In the subspace spanned by these transfer components, data properties are preserved and data distributions in different domains are close to each other.

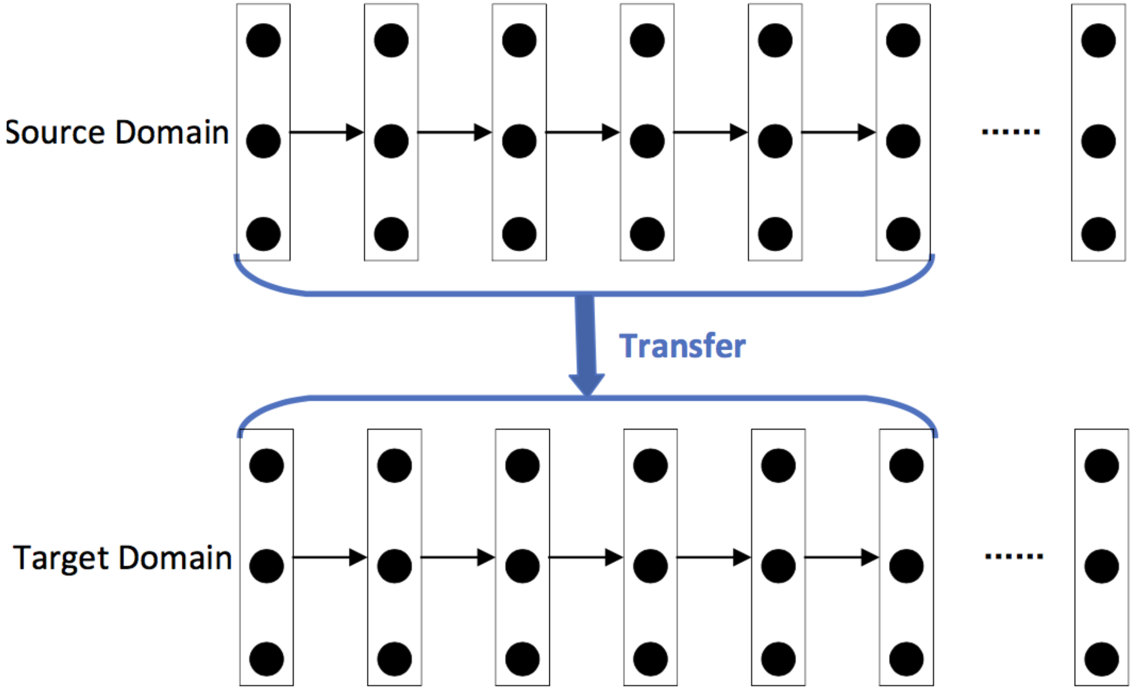
Improve work:

- replace MMD distance with multiple kernel variant MMD (MK-MMD) distance

$$D_{\mathcal{MMD}}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|$$

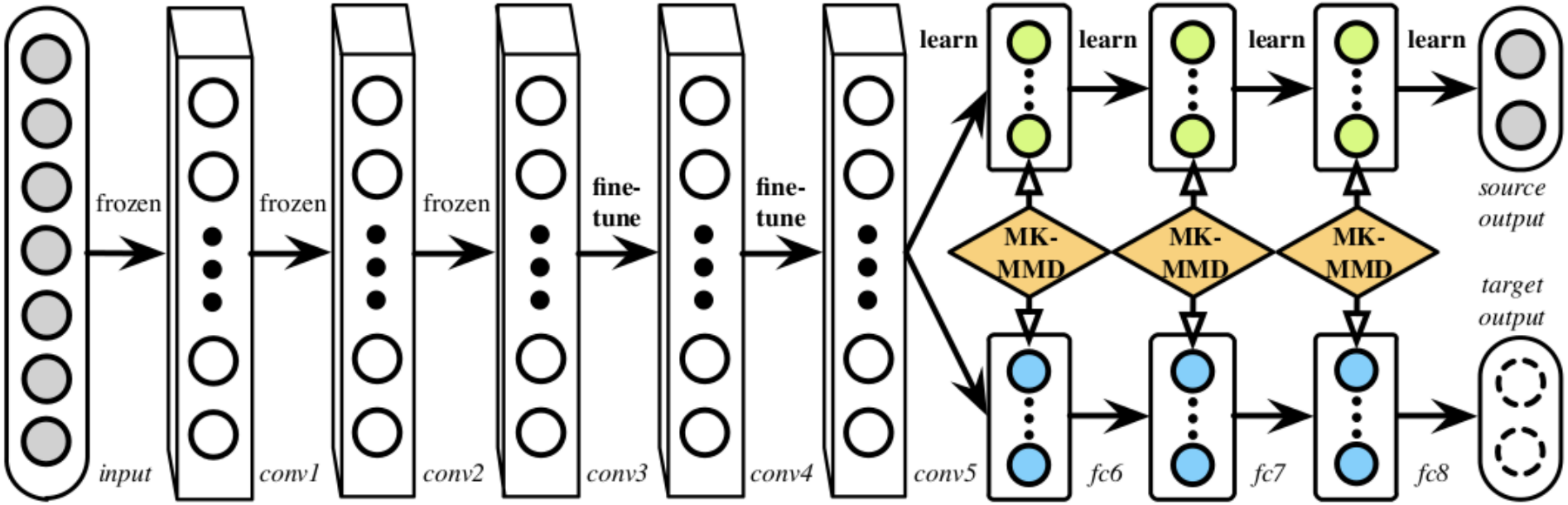
Network-based deep transfer learning

- **Assumption**: The front-layers of the network can be treated as a feature extractor, and the extracted feat ures are versatile.
- **Main Idea**: reuse the partial network that pre-trained in the source domain, including its network structure and connection parameters, transfer it to be a part of deep neural network which used in target domain



Deep Adaptation Netowrk， DAN

- hidden representations of all task-specific layers are embedded in a reproducing kernel Hilbert space where the mean embeddings of different domain distributions can be explicitly matched.
- The domain discrepancy is further reduced using an optimal multi-kernel selection method for mean embedding matching.



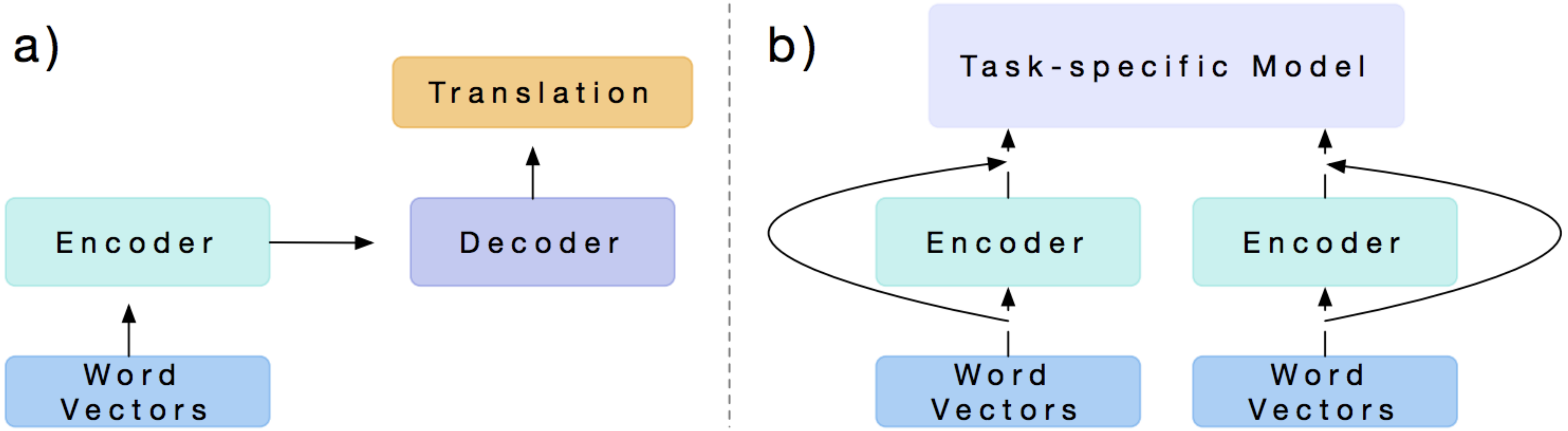
Context vectors（CoVe）

CV:

- LeNet, AlexNet, VGG, Inception, ResNet

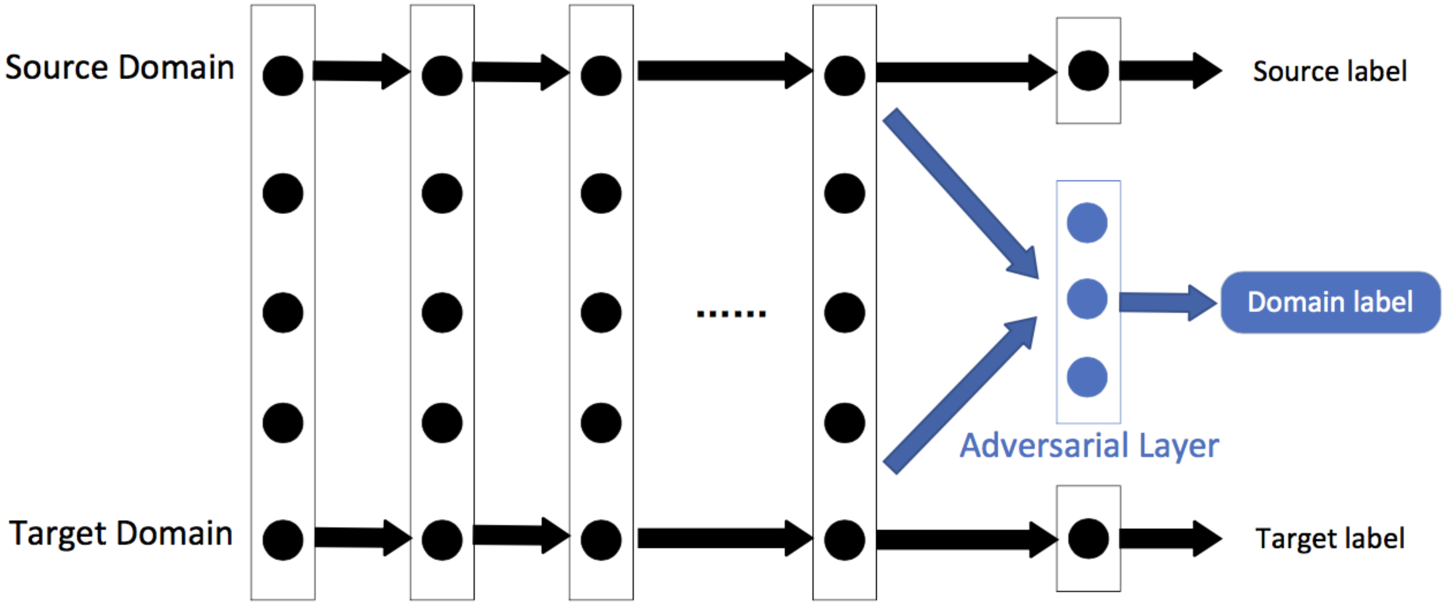
NLP:

- train a two-layer, bidirectional LSTM as the encoder of an attentional sequence-tosequence model for machine translation and use it to provide context for other NLP models.



Adversarial-based deep transfer learning

- **Motivation** : generative adversarial nets (GAN)
- **Assumption** : For effective transfer, good representation should be discriminative for the main learning task and indiscriminate between the source domain and target domain.
- **Main Idea**: find transferable representations that is applicable to both the source domain and the target domain.



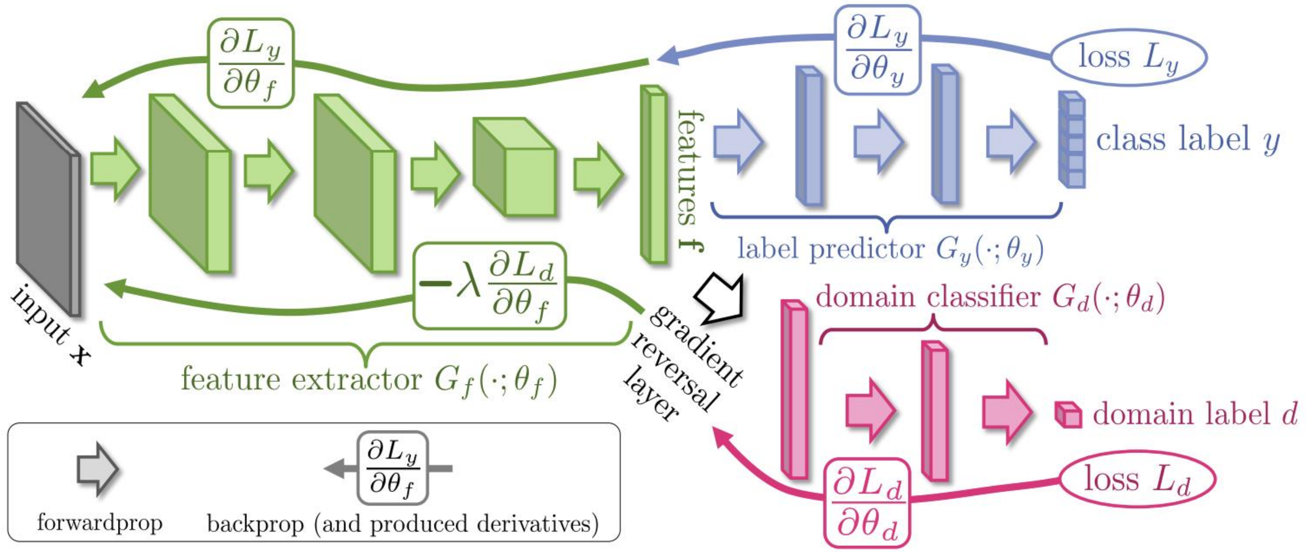
Unsupervised domain adaptation by backpropagation

Architecture :

- a deep feature extractor 、a deep label predictor (blue)、a domain classifier (red)

model:

- domain classifier via a gradient reversal layer that multiplies the gradient by a certain negative constant during the backpropagation-based training.
- Gradient reversal ensures that the feature distributions over the two domains are made similar



PART 3

Conclusion

Categorizing of deep transfer learning

Approach category	Brief description	Some related works
Instances-based	Utilize instances in source domain by appropriate weight.	[4], [27], [20], [24], [10], [26], [11]
Mapping-based	Mapping instances from two domains into a new data space with better similarity.	[23], [12], [8], [14], [2]
Network-based	Reuse the partial of network pre-trained in the source domain.	[9], [17], [15], [30], [3], [6], [28]
Adversarial-based	Use adversarial technology to find transferable features that both suitable for two domains.	[1], [5], [21], [22], [13], [16]

END

THANK YOU