



迁移学习在神经网络中的应用

主讲人：陈晓敏

中国 北京 2017/06/16



01 迁移学习简介

迁移学习是运用已存有的知识对不同但相关领域问题进行求解的一种新的机器学习方法



训练高可靠性和准确性分类模型，数据应满足两个基本假设

独立同分布

用于学习的训练样本与新的测试样本需满足独立同分布的条件

01



02

足够训练样本

必须有足够可利用的训练样本才能学习得到一个好的分类模型



现实应用中往往无法满足这两个假设

样本数据可能失效

原先可利用的有标签的样本数据可能变得不可用，与新来的测试样本的分布产生语义、分布上的缺口，如股票数据

01



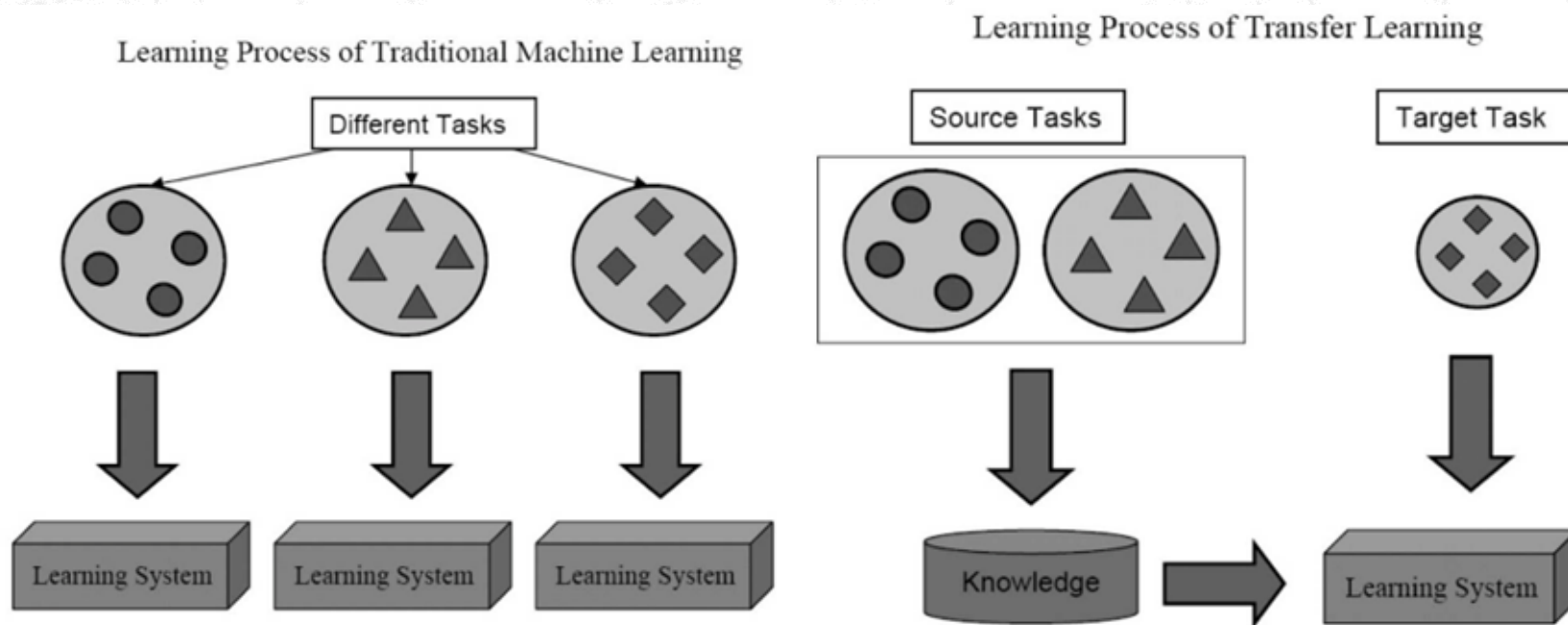
02

有标签样本匮乏

通过数据挖掘技术，新数据不断涌现，标注大量样本非常费时费力，而且人工标注容易出错



迁移学习定义



迁移学习是运用已存有的知识对不同但相关领域问题进行求解的一种新的机器学习方法。它放宽了传统机器学习中的两个基本假设，目的是迁移已有的知识来解决目标领域中仅有少量有标签样本数据甚至没有的学习问题。

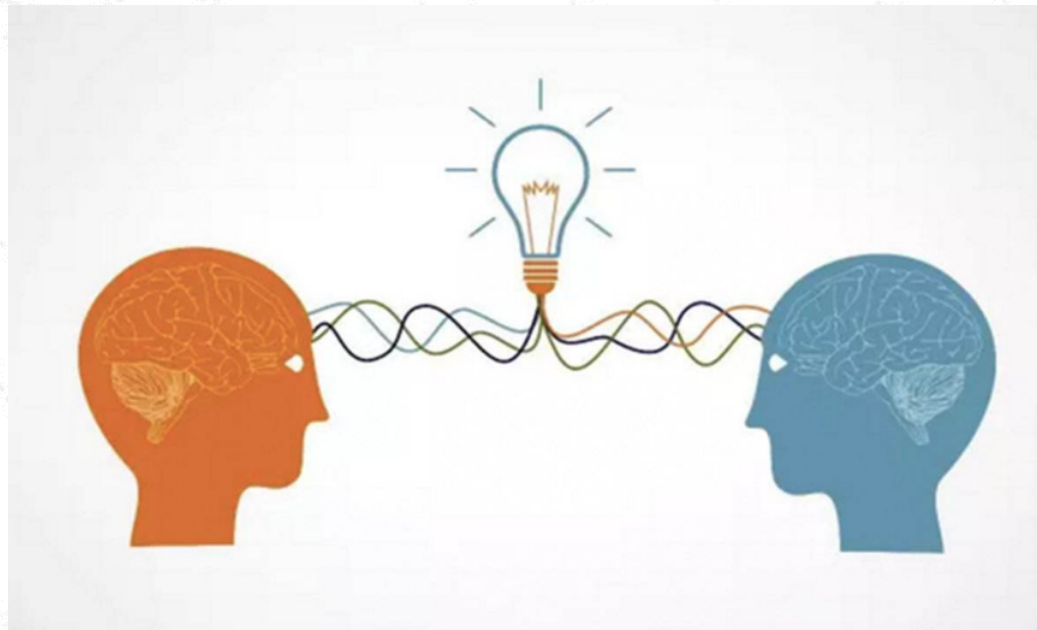


迁移学习限制

两个不同的领域共享的因素越多，迁移学习就越容易，
否则就越困难，甚至出现“负迁移”的情况

71

比如：一个人要是学会了骑自行车，那他就很容易学会开摩托车；但是有时候看起来很相似的事情，却有可能产生“负迁移”现象。比如，学会骑自行车的人来学习三轮车反而不适应，因为这两种车型的重心位置不同





迁移学习分类

基于特征选择

01

基于特征选择的迁移学习方法是识别出源领域与目标领域中共有的特征表示，然后利用这些特征进行知识迁移，与目标域样本类别高度相关的那些特征应该在训练得到的模型中被赋予更高的权重

基于特征映射

02

基于特征映射的迁移学习方法是把各个领域的数据从原始高维特征空间映射到低维特征空间，在该低维空间下，源领域数据与目标领域数据拥有相同的分布，这些映射得到的特征不在原始的特征当中，是全新的特征

基于样本权重

03

基于实例的迁移学习通过度量有标签的训练样本与无标签的测试样本之间的相似度来重新分配源领域中样本的采样权重。相似度大的，即对训练目标模型有利的训练样本被加大权重，否则权重被削弱

02 在神经网络上应用

通过神经网络将源领域的判别知识转移到目标领域，完成分类识别等任务



Transfer Learning of Deep Neural Network for Speech Emotion Recognition

会议: Chinese Conference on Pattern Recognition , 2016

机构: 爱奇艺北京智能媒体部

作者: Ying Huang 等

主要内容:

“

通过内部语音数据集训练一个话者识别任务和性别识别网络，再用情感语音数据的训练集进行微调用于语音情感识别

CCPR2016 语音情感识别比赛冠军



Transfer Learning of Deep Neural Network for Speech Emotion Recognition

源领域数据集： 爱奇艺内部数据， 1105 个话者， 每个话者约 400 句发音， 性别分布均匀

目标域数据集： CHEAVD 数据集， 140 分钟发音， Train: 1981 Test: 243 Val:628

特征： Interspeech 2012 challenge feature set

网络结构： 前馈神经网络 (FFN)

$$f^N(\dots f^3(f^2(f^1(X)))) \quad f(X; \mathbf{w}, \mathbf{b}) = \text{sigmoid}(\mathbf{X}^T \mathbf{w} + \mathbf{b}) = \frac{1}{1 + \exp(-(\mathbf{X}^T \mathbf{w} + \mathbf{b}))}$$

$$\hat{y}_i = P(y = i/X) = \text{softmax}(\mathbf{X}^T \mathbf{w}_i + \mathbf{b}_i) = \frac{\exp(\mathbf{X}^T \mathbf{w}_i + \mathbf{b}_i)}{\sum_j \exp(\mathbf{X}^T \mathbf{w}_j + \mathbf{b}_j)}$$

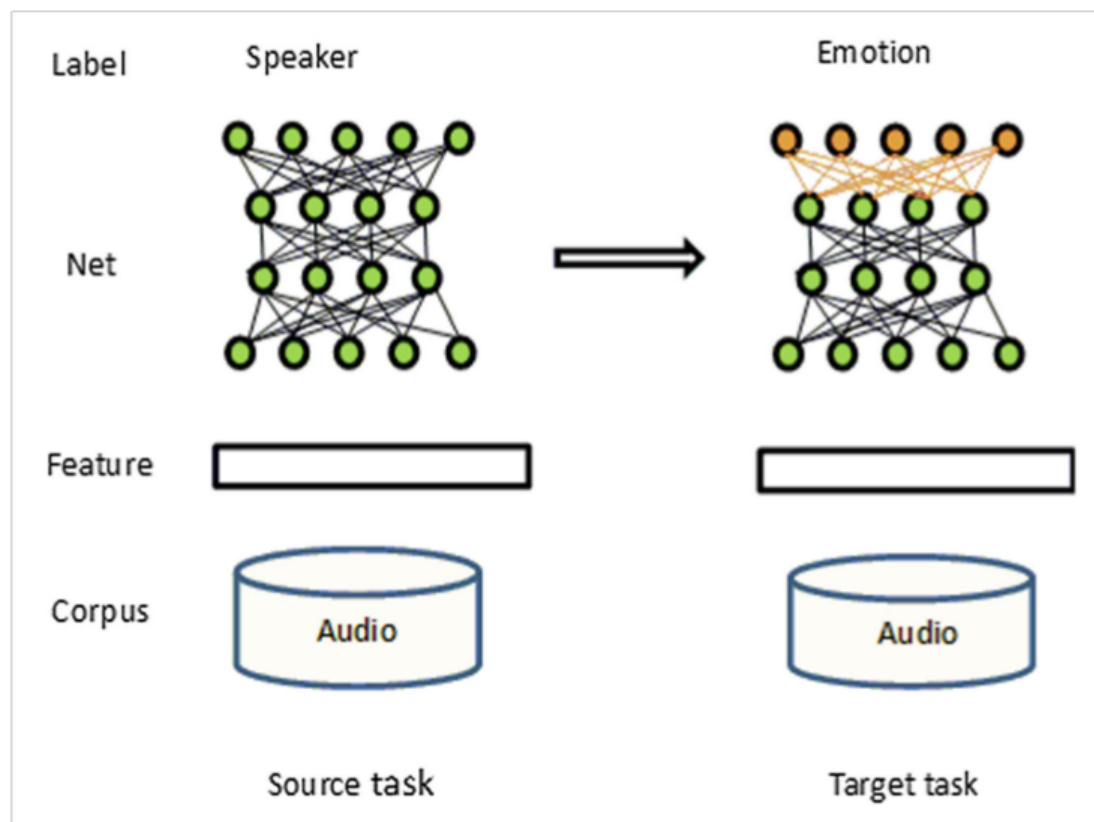
$$J = - \sum_j y_j \log(\hat{y}_j)$$



Transfer Learning of Deep Neural Network for Speech Emotion Recognition

迁移学习:

去除最后一个 output 层
，加一个新 output 层用
目标训练集训练





Transfer Learning of Deep Neural Network for Speech Emotion Recognition

多网络融合:

两个网络: 隐层 1024x1024, 2048x2048

最终输出:

$$l = \begin{cases} i(y_{1,i}^{1024} - y_{2,j}^{1024}) \geq (y_{1,m}^{2048} - y_{2,n}^{2048}) \\ m(y_{1,i}^{1024} - y_{2,j}^{1024}) < (y_{1,m}^{2048} - y_{2,n}^{2048}) \end{cases}$$



Transfer Learning of Deep Neural Network for Speech Emotion Recognition

实验结果:

源任务:

Network structure (hidden layer)	Speaker task Error rate (%)	Gender task Error rate (%)
1024:1024	1.84	0.27
2048:2048	1.23	0.29



Transfer Learning of Deep Neural Network for Speech Emotion Recognition

实验结果:

目标任务:

验证集

Method	ACC (%)	MAP (%)
Baseline	41.98	30.02
DFFN (1024:1024)	46.93	34.16
DFFN (2048:2048)	50.01	35.00
DFFN (Ensemble)	49.46	35.72

测试集

Method	ACC (%)	MAP (%)
Baseline	41.98	30.02
DFFN (1024:1024)	43.17	19.03
DFFN (2048:2048)	44.81	20.90
DFFN (Ensemble)	45.24	20.81

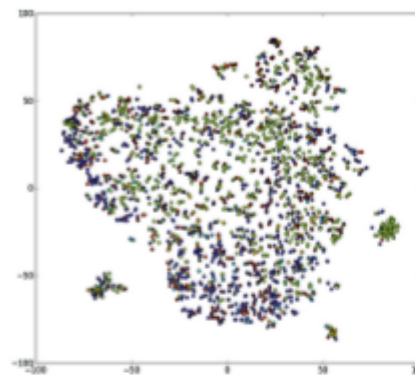


Transfer Learning of Deep Neural Network for Speech Emotion Recognition

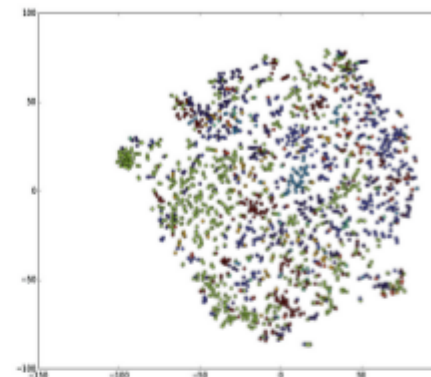
分析:

1024x1024 最后一隐层输出

源任务类别越多对目标任务分类有利

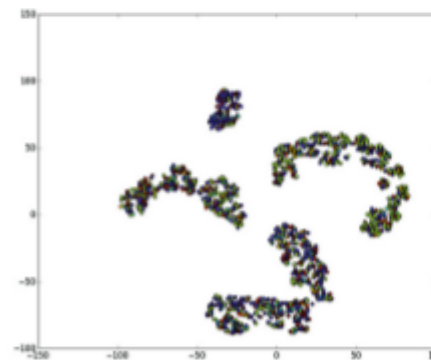


(a)

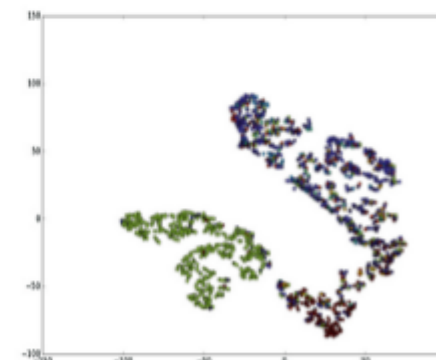


(b)

Fine-tuning 前



(c)



(d)

Fine-tuning 后



Deep Transfer Metric Learning

会议: Computer Vision and Pattern Recognition, 2015

机构: 新加坡南洋理工大学电气与电子工程学院

作者: Junlin Hu 等

主要内容:

“

通过设计神经网络的优化目标函数，使通过网络输出的子空间特征，样本的组内距离达到最小，类间距离达到最大，同时通过源数据集与目标数据集具有相同的分布



Deep Transfer Metric Learning

源领域数据集： WDRef 图像数据集，15000 张人脸图片，1500 个人，每个人 10 张图片

目标域数据集： LFW 图像数据集，13233 张人脸图片，2995 个人

特征： LBP 特征，通过 PCA 降到 500 维

网络结构： 前馈神经网络 (FFN)



Deep Transfer Metric Learning

目标函数:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha S_b^{(M)} \\ + \gamma \sum_{m=1}^M \left(\|\mathbf{w}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right)$$



Deep Transfer Metric Learning

目标函数:

类内散度:

$$S_c^{(m)} = \frac{1}{Nk_1} \sum_{i=1}^N \sum_{j=1}^N P_{ij} d_{f^{(m)}}^2(\mathbf{x}_i, \mathbf{x}_j)$$

类间散度:

$$S_b^{(m)} = \frac{1}{Nk_2} \sum_{i=1}^N \sum_{j=1}^N Q_{ij} d_{f^{(m)}}^2(\mathbf{x}_i, \mathbf{x}_j)$$



Deep Transfer Metric Learning

目标函数:

源集和目标集的距离度量

$$D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) = \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} f^{(m)}(\mathbf{x}_{ti}) - \frac{1}{N_s} \sum_{i=1}^{N_s} f^{(m)}(\mathbf{x}_{si}) \right\|_2^2$$



Deep Transfer Metric Learning

目标函数:

DTML 目标函数

$$\begin{aligned} \min_{f^{(M)}} J = & S_c^{(M)} - \alpha S_b^{(M)} + \beta D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s) \\ & + \gamma \sum_{m=1}^M \left(\|\mathbf{w}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right), \end{aligned}$$



Deep Transfer Metric Learning

目标函数:

DSTML 目标函数

$$\min_{f^{(M)}} J = J^{(M)} + \sum_{m=1}^{M-1} \omega^{(m)} h(J^{(m)} - \tau^{(m)})$$

$$\begin{aligned} J^{(m)} = & S_c^{(m)} - \alpha S_b^{(m)} + \beta D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) \\ & + \gamma \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right), \end{aligned}$$



Deep Transfer Metric Learning

实验结果:

LFW 集

Method	Transfer	Accuracy (%)
DDML [16]	<i>no</i>	83.16 ± 0.80
STML	<i>yes</i>	83.60 ± 0.75
STML ($\beta = 0$)	<i>no</i>	82.57 ± 0.81
DTML	<i>yes</i>	85.58 ± 0.61
DTML ($\beta = 0$)	<i>no</i>	83.80 ± 0.55
DSTML	<i>yes</i>	87.32 ± 0.67



Table 3. Top r ranked matching accuracy (%) on the VIPeR dataset with $\#test = 316$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	3.99	8.73	12.59	25.32
L_2	-	4.24	8.92	12.66	25.35
DDML [16]	i-LIDS	5.63	12.91	21.71	41.80
	CAVIAR	5.91	13.53	19.86	37.92
	3DPeS	6.67	17.16	23.87	41.65
DTML ($\beta = 0$)	i-LIDS	5.88	13.72	21.03	41.49
	CAVIAR	6.02	13.81	20.33	38.46
	3DPeS	7.20	18.04	25.96	43.80
DTML	i-LIDS	6.68	15.73	23.20	46.42
	CAVIAR	6.17	13.10	19.65	37.78
	3DPeS	8.51	19.40	27.59	47.91
DSTML	i-LIDS	6.11	16.01	23.51	45.35
	CAVIAR	6.61	16.93	24.40	41.55
	3DPeS	8.58	19.02	26.49	46.77

Table 4. Top r ranked matching accuracy (%) on the i-LIDS dataset with $\#test = 60$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	16.51	28.41	38.28	69.32
L_2	-	16.30	28.25	38.40	69.77
DDML [16]	VIPeR	25.32	45.61	60.27	83.31
	CAVIAR	25.67	45.03	61.38	82.56
	3DPeS	28.71	48.55	62.53	83.15
DTML ($\beta = 0$)	VIPeR	26.27	47.59	62.62	85.07
	CAVIAR	26.15	46.87	62.08	84.78
	3DPeS	30.23	51.60	65.21	85.53
DTML	VIPeR	28.90	51.43	65.47	87.23
	CAVIAR	26.23	49.31	63.99	87.76
	3DPeS	31.01	54.51	65.96	88.66
DSTML	VIPeR	28.35	50.81	61.58	84.72
	CAVIAR	28.37	49.68	64.59	88.68
	3DPeS	33.37	54.56	68.27	89.32

Table 5. Top r ranked matching accuracy (%) on the CAVIAR dataset with $\#test = 36$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	20.65	36.44	48.52	88.34
L_2	-	20.19	36.43	48.55	87.69
DDML [16]	VIPeR	23.80	42.15	55.61	90.73
	i-LIDS	22.72	41.36	56.92	90.06
	3DPeS	23.85	44.30	57.81	90.27
DTML ($\beta = 0$)	VIPeR	23.71	42.57	56.15	90.55
	i-LIDS	23.09	42.81	58.43	90.41
	3DPeS	25.11	46.71	59.69	91.99
DTML	VIPeR	23.88	42.36	55.60	92.12
	i-LIDS	26.06	47.37	61.70	94.23
	3DPeS	26.10	47.80	61.31	93.02
DSTML	VIPeR	26.05	44.33	57.02	92.80
	i-LIDS	25.91	44.47	58.88	93.33
	3DPeS	28.18	49.96	63.67	94.13

Table 6. Top r ranked matching accuracy (%) on the 3DPeS dataset with $\#test = 95$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	26.93	42.21	51.56	69.57
L_2	-	26.95	42.57	51.46	69.84
DDML [16]	VIPeR	29.56	51.03	61.71	78.62
	i-LIDS	27.81	50.29	58.33	77.05
	CAVIAR	30.32	49.36	58.92	79.61
DTML ($\beta = 0$)	VIPeR	30.33	52.18	62.24	82.58
	i-LIDS	29.12	50.07	59.99	78.59
	CAVIAR	31.23	51.88	60.87	81.30
DTML	VIPeR	32.12	54.36	65.92	84.65
	i-LIDS	32.11	52.08	61.63	79.45
	CAVIAR	31.79	51.92	62.78	81.98
DSTML	VIPeR	32.51	52.97	63.12	83.08
	i-LIDS	31.57	52.54	63.50	84.02
	CAVIAR	32.53	54.29	65.28	84.72



感谢聆听

Thanks For
Listening

