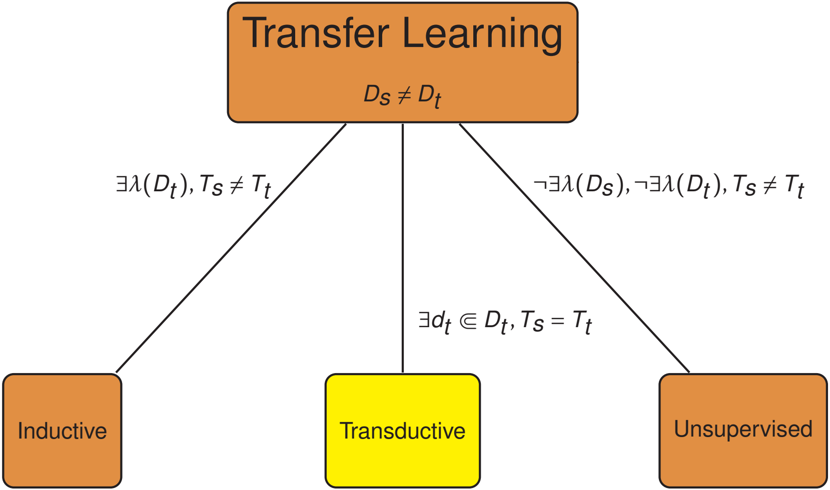
Transfer Learning Taxonomy



Transductive Work

Source and target tasks are the same but the domains are different

- Recognizing spoiled pears after training on apples
- Classifying letters after training on numbers

This is not exactly what we do; we are reusing elements of T_s in T_t, but not the entire task (which would be the entire network with no change).

- *Arnold*; entropy-based IFT; as fast as SVM, not as adaptable
- *Joachims*; TSVMs, better performance, especially on small sets
- *Huang*; non-parametric weight re-sampling between D_s and D_t
- *Sugiyama*; non-density estimation based shift management, better performance
- *Bickel*; Kernel LR classifier with no modeling
- *Dai*; Naive Bayes, good performance, revising model

Transductive Work

- *Blitzer*; SCL Algorithm, feature extraction from D_t; uses pivots, which require domain expertise; may be able to use mutual information for selection though (MI-SCL)
- *Daumé*; Kernel mapping NLP into high-dimensional feature space; train classifiers here; kernel function is domain driven
- *Dai*; co-clustering, good classification performance
- *Xing*; bridged refinement uses a mixture of training and test data to bridge a classifier from $D_s \rightarrow D_t$
- *Ling*; Spectral classifier objective function seeks consistency between D_s and D_t
- *Xue*; Text classifier extends PLSA integrating labeled and unlabeled data from different domains
- *Pan*; uses MMDE in low-dimensional spaces, but computationally intense
- *Pan*; TCA works in same domain as MMDE but better claimed performance

More Neurally Focused

- *Hubel*; multi-starge Hubel-Wiesel architectures, alternating layers of convolutions and max pooling with multiple tasks, which embody transfer learning
- *Huang*; extended Hubel's work by sharing hidden layers, improving recognition accuracy of new languages against fresh DNNs; this extends across language families as well
- *Cireşan*; Pretraining on chinese characters yields much shorter training times for latin character recognition in an entire network
- *Collobert*; A single convolutional network trained jointly on all tasks using weight sharing (); semi-supervised learning as well (note: this is an inductive learner as we have multiple tasks).
- *Kandaswamy*; They transfer features from various layers; I believe they are transferring entire layer parameters here.
- *Swietojanski*; Hybrid machines (RBMs); used in tandem as features for a hidden Markov model. Unsupervised pre-training importnant, and can be language-independent.

More Neurally Focused

- *Yosinski*; reusing entire high-level layers to increase DNN performance; transferring any features better than none at all
- *Morelli*; essentially retraining new classifiers here over time as well.
- *Calandra*; constant re-training of established DNN; essentially, they dynamically create NN on the fly as new data appears. They do not use apoptosis, they depend on wieght shifts as new data appears in freshly trained systems.

Overall, neurogenesis is not being examined at this point. Researchers seem to prefer horizontal reuse over vertical adaptation.