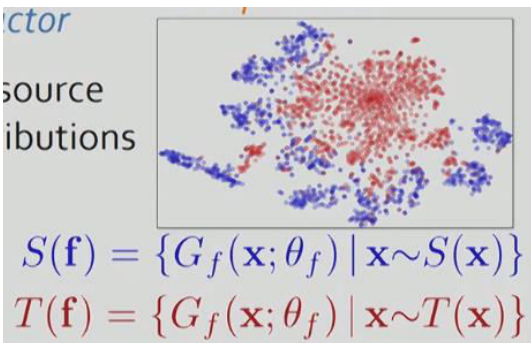
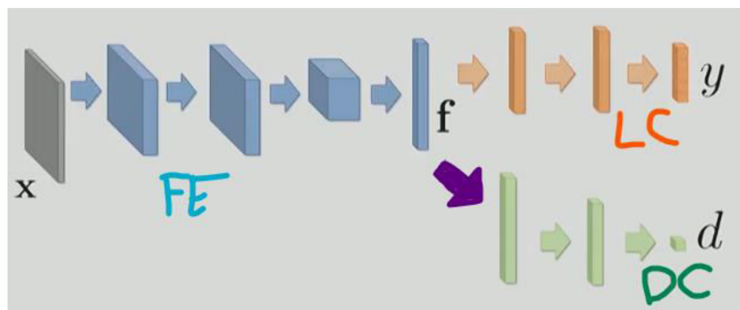
https://arxiv.org/pdf/1505.07818.pdf
https://www.youtube.com/watch?v=uUUvieVxCMs

- There are often modalities that don't have large labeled data sets - e.g. biomedical
- There are usually, however, sources of **surrogate data**
    - **Surrogate Data -** analogous data that is similar in some ways that is plentiful and has labels
    - We can borrow from those adjacent modalities
- We can also apply **data augmentation** to amplify our data and make it more effective (e.g. transformations)
- If our training data is shifted, our neural network will overfit for the patterns in the training data that are not present in the test data
- **Domain Adaptation - shifting** the training data in some way to make it more like our test data
    - We may need domain adaptation if we have (semi-)synthetic data that looks realistic to us

Large-Scale Deep Unsupervised Domain Adaptation - a process that makes some assumptions
- Lots of **labeled** data in **source domain**
- Lots of **unlabeled** data in **target domain**
- **Goal:** train a neural network with source domain data that does well on target domain





- We can use something like **t-SNE** to visualise the features of the two distributions
    - This can help us see how different our source and target distributions are
    - **Goal:** we want to align the two distributions in high dimensions so they overlap when we visualise them in something like t-SNE

Approach: Define a loss that measures the mismatch between the distributions
- While minimising the loss, you adjust the feature vector so the features of the two distributions overlap
- 2016 Sun and Saenko: Use **second-order moments** mismatch
    - This is a **parametric approach**
- 2015 Long and Wang: Use **maximum mean discrepancy (MMD)**
    - This is a **non-parametric approach**
- GAN: Alignment through **adversarial learning**, e.g. with GANs

Adversarial Learning
- We can train a **binary classifier** that predicts which distribution do input points in the feature space come from
- In this case, we want the **feature extractor** to **fool** the binary classifier to misclassify the points
- We Make a **Three-Partite neural network**
    - **Feature Extractor -** takes features out of input sample
    - **Label/Class Classifier -** classifier that predicts what class the data is
        - This part we train on the source data, as we need the labels
    - **Domain Classifier -** classifier that predicts which distribution the data is from
        - This part we train on all data, as we only need to know which distribution they come from
- At test time, we use the **feature extractor + class predictor**

Emerging Features in adversarial learning

- **Discriminative features -** good at predicting ***y***
- **Domain-Discriminative -** good at predicting ***d***
    - *This, we don't actually want, as we want features from source that are the same as in target distribution*

Simulated Annealing - A **Heuristic process**: simple, effective, or **evolutionary** process to promote **exploring**

- We don't always improve to find the local minimum, sometimes we encourage **searching instead**
    - **Exploiting -** always improving, which is susceptible to getting stuck in a local minimum
        - Kind of like over-fitting to some portion of your data
    - **Exploring -** visiting more of the search space with the hope of finding a better minimum
- Simulated annealing introduces a **trade-off** between the two

Gradient Reversal Layer - on backpropagation, we multiply the gradients by a **negative parameter**, to make the features that result from the learning process **be bad at predicting *d***

- The reversal parameter is **annealed from zero** during training
    - *In the beginning, the adversarial term confuses the process*

Conclusion

- Borrows ideas from GANs
- Easier than GANs, as it has external stabilising of losses
- You can use semi-supervised learning, reduces requirement for labels
- Results for some domains (e.g. **segmentation**) is not great