COMPSCI 589
Lecture 16: Alternative Learning Problems

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

Machine Learning

Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Substitute "training data D" for "experience E."

The Classifier Learning Problem

Definition: Classifier Learning

Given as input a training data set of example pairs $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i), 1 \leq i \leq N_{tr}\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a feature vector and $y_i \in \mathcal{Y}$ is a class label, output a function $f : \mathbb{R}^D \rightarrow \mathcal{Y}$ (the classifier) that accurately predicts the class label y for any feature vector $\mathbf{x}$.

The Regression Learning Problem

Definition: Regression Learning Problem

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a feature vector and $y_i \in \mathbb{R}$ is the output, learn a function $f : \mathbb{R}^D \to \mathbb{R}$ that accurately predicts y for any feature vector $\mathbf{x}$.

Multi-Task Learning

Definition: Multi-Task Learning

Given T tasks, each with its own training data $\mathcal{D}_{tr}^t = \{(\mathbf{x}_i^t, y_i^t), 1 \leq i \leq N_{tr}^t\}$ for $t = 1, \ldots, T$, learn functions $f_t : \mathbb{R}^D \to \mathcal{Y}_t$ for each task t simultaneously.

If the tasks are related in some way, it should be possible to leverage common structure to improve generalization across all tasks by sharing information across the T training sets.

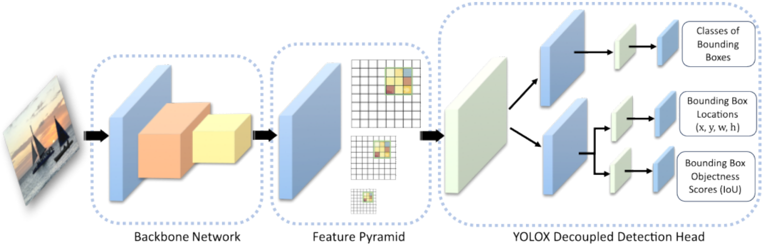
Examples Applications

- Predicting the **class labels** and **locations** of objects in an image.
- Predicting the **class label** and **data quality** of feature vectors.
- Predicting **age** and **affective state** from an image of a person.
- Predicting **disease risk** and **treatment response** from patient records.
- Predicting **heart rate**, **respiratory rate** and **activity type** from actigraphy and photoplethsmography data.

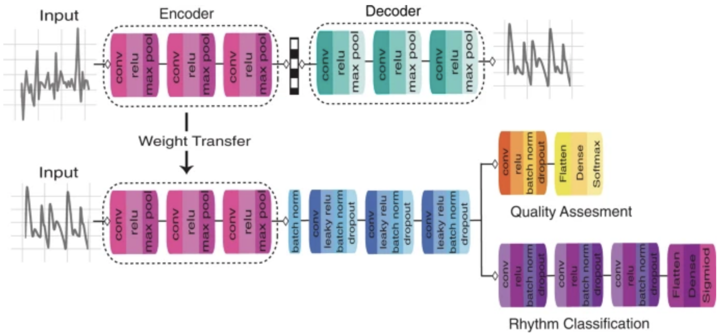
Example Method: Multi-Output Neural Networks

- Useful in cases where we need to produce multiple outputs for the same input feature vector.
- Learn a single neural network model with multiple output heads, one per task.
- The feature extraction portion of the network can learn using examples from all tasks.
- Each task gets a specialized output head.
- Tasks can be a mix of classification and regression problems.

Yolo Object Detector



Deep Beat Heart Rhythm Calassification



The Continual Learning Problem

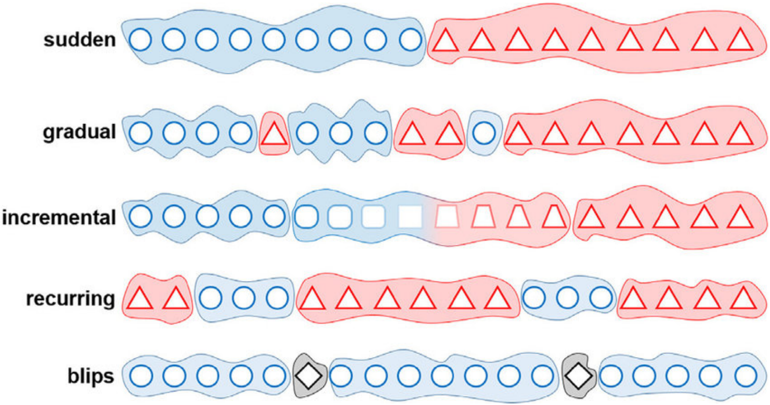
Definition: Continual Learning under Distribution Drift

Given a stream $\{\mathbf{x}_t, y_t\}_{t=1}^{\infty}$ where the underlying joint distribution $P_t(\mathbf{X}, Y)$ changes over time, learn a sequence of predictive functions $f_t(\mathbf{x})$ that adapts to the shifts while maintaining accuracy over the recent data.

Examples of Continual Learning under Concept Drift

- Predicting stock prices where market conditions change over time.
- Predicting temperature and rainfall under climate change.
- Predicting user ratings in recommendation systems as preferences evolve.
- Detecting email spam as spammers change content patterns over time.
- Detecting intrusions in cybersecurity as attackers change methods.

Concept Drift Dynamics



Methods for Continual Learning under Concept Drift

- **Sliding window models:** Train on most recent W observations to adapt to current distribution.
- **Exponential forgetting:** Weight recent examples higher to track changes.
- **Ensemble methods:** Maintain multiple models and re-weight them as drift occurs.
- **Drift detection:** Monitor prediction errors to detect distribution change.
- **Adaptive learning rates:** Increase updates when drift is detected.

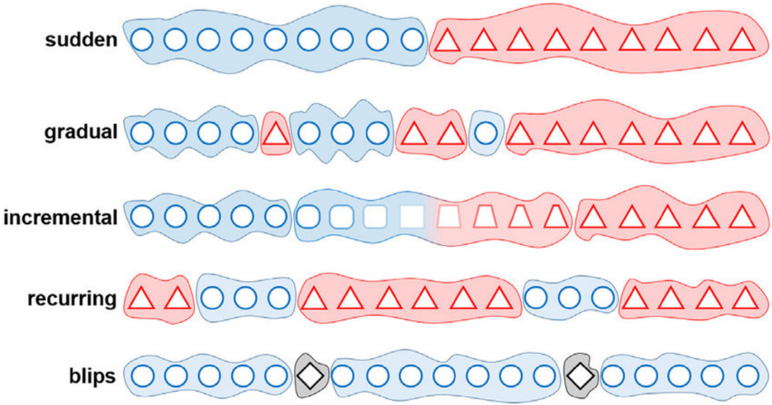
The Transfer Learning Problem

Definition: Transfer Learning

Given a supervised source task $\mathcal{T}_S$, a large source dataset $\mathcal{D}_S$, a related supervised target task $\mathcal{T}_T$, and a small target data set $\mathcal{D}_T$, output a prediction function $f_T(\mathbf{x})$ that performs as well as possible on $\mathcal{T}_T$.

A common scenario is for the generative processes for $\mathcal{D}_S$ and $\mathcal{D}_T$ to be different, resulting in $\mathcal{D}_T$ being OOD relative to $\mathcal{D}_S$.

Types of Source-Target Dataset Shifts



Examples of Transfer Learning

- Pretraining a CNN on ImageNet and fine-tuning for medical image diagnosis.
- Adapting a speech recognizer trained on English to Spanish.
- Applying a sentiment model trained on product reviews to tweets.
- Transferring control policies learned on one type of robot to a different type of robot.

Methods for Transfer Learning

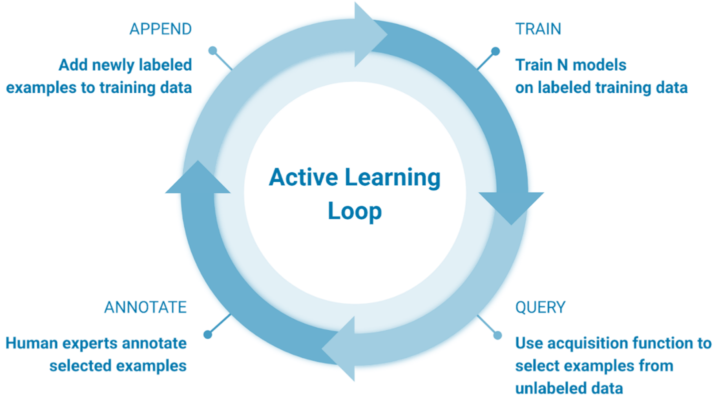
- **Feature-based transfer:** Train a neural network model on $\mathcal{D}_S$, use it extract features for data cases from $\mathcal{D}_T$, learn a basic model in the new feature space.
- **Fine-tuning:** Train a neural network model on $\mathcal{D}_S$. Fine-tune the model using a small amount of training iterations on $\mathcal{T}_T$.
- **Freezing layers:** Train a neural network model on $\mathcal{D}_S$. Freeze all but the last few layers, fine-tune the last few layers using $\mathcal{T}_T$.
- **Re-Weighting:** Estimate density ratio $\frac{p_T(\mathbf{x})}{p_S(\mathbf{x})}$, re-weight the source data using density ratios, learn jointly on re-wighted source data from $\mathcal{D}_S$ and target data $\mathcal{T}_T$.

The Active Learning Problem

Definition: Active Learning

Given a large unlabeled pool $\mathcal{D}_U = \{\mathbf{x}_i | 1 \leq i \leq N\}$ and a limited labeling budget B, iteratively select the most informative examples from $\mathcal{D}_U$ to label, resulting in a labeled data set $\mathcal{D}_L = \{(\mathbf{x}_j, y_j) | 1 \leq j \leq B\}$. Use the labeled data $\mathcal{D}_L$ to learn a predictive model $f(\mathbf{x})$ that performs well on future data.

Active Learning Loop



Examples of Active Learning

- Labeling the most uncertain medical images for diagnosis.
- Selecting ambiguous text samples for sentiment analysis.
- Choosing diverse images for object detection annotation.

Methods for Active Learning

- **Random sampling:** Randomly sample B instances and label them.
- **Uncertainty sampling:** Query examples with high predictive entropy or variance.
- **Query-by-committee:** Select cases with the greatest disagreement among models in an ensemble.
- **Expected model change:** Pick samples that most affect parameter updates.

The Semi-Supervised Learning Problem

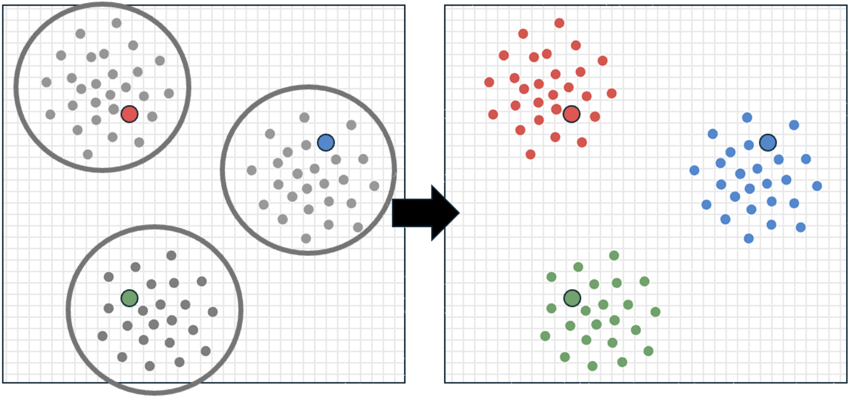
Definition: Semi-Supervised Learning

Given a small labeled dataset $\mathcal{D}_L = \{(\mathbf{x}_i, y_i)\}$ and a large unlabeled dataset $\mathcal{D}_U = \{\mathbf{x}_j\}$, learn a prediction function $f(\mathbf{x})$ that will perform well on future data.

Methods for Semi-Supervised Learning

- **Pseudo-labeling:** Iteratively select the instances where the model makes the most confident predictions, treat these predictions as true labels, re-fit the model.
- **Graph-based methods:** Propagate label information over similarity graphs constructed from both labeled and unlabeled instances. Learn models on propagated labels.
- **Entropy regularization:** Encourage confident predictions on unlabeled data.
- **Feature representation:** Use the unlabeled data to learn feature representations using unsupervised deep learning methods. Use the labeled data to learn shallow models in learned feature space.

Cluster-Based Semi-Supervised Learning



Graph-Based Semi-Supervised Learning