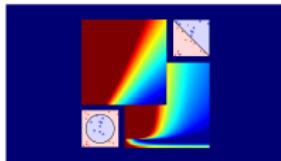


Machine Learning Foundations (機器學習基石)



Lecture 2: The Learning Problems, Extended

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)



Multiclass Classification: Which Fruit?



?

(image by Robert-Owen-Wahl from Pixabay)



apple



orange



strawberry



kiwi

(images by Pexels, PublicDomainPictures, 192635, Rob van der Meijden from Pixabay)

$$\mathcal{Y} = \{\text{apple}, \text{orange}, \text{strawberry}, \text{kiwi}\}$$

Multilabel Classification: Which Fruits?



? : {apple, orange, kiwi}

(image by Michal Jarmoluk from Pixabay)



apple



orange



strawberry



kiwi

(images by Pexels, PublicDomainPictures, 192635, Rob van der Meijden from Pixabay)

multilabel classification:
classify input to **multiple (or no)** categories
 $\mathcal{Y} = 2^{\{\text{apple, orange, strawberry, kiwi}\}}$

What Tags?

The screenshot shows a product page from Amazon. At the top, there's a browser header with back and forward buttons, a search bar containing 'Amazon.com: Learning From D...', and a URL 'http://www.amazon.com/gp/product/1600490069'. Below the header is the product title 'Learning From Data [Hardcover]' and the authors' names: 'Yaser S. Abu-Mostafa (Author), Malik Magdon-Ismail (Author), Hsuan-Tien Lin (Author)'. A 5-star rating with 2 reviews is shown, along with a 'Liked' button. The book cover is visible on the left, featuring a blue and red abstract graphic. To the right of the book cover, it says 'Available from these sellers.' and '1 new from \$28.00'.

?: {~~machine learning, data structure, data mining, object oriented programming, artificial intelligence, compiler, architecture, chemistry, textbook, children book, ... etc.~~} }

another **multilabel** classification problem:
tagging input to multiple categories

Binary Relevance: Multilabel Classification via Yes/No

binary classification

{yes, no}

multilabel w/ L classes: \bigcup yes/no questions

machine learning (Y), data structure (N), data mining (Y), OOP (N), AI (Y), compiler (N), architecture (N), chemistry (N), textbook (Y), children book (N), etc.

- Binary Relevance (BR): reduction (transformation) to **multiple isolated binary classification**
- disadvantages (addressed by more sophisticated models):
 - **isolation**—hidden relations not exploited
(e.g. ML and DM **highly correlated**, ML **subset of AI**, textbook & children book **disjoint**)
 - **imbalanced**—few **yes**, many **no**

BR for multilabel classification:
uses **binary classification** as a core tool

Sophisticated Output: Image Generation Problems

Style Transfer



(Leonardo da Vinci,
in Public Domain)

+



(Van Gogh,
in Public Domain)

⇒



(Pjfinlay,
with CC0)

all images are downloaded from Wikipedia

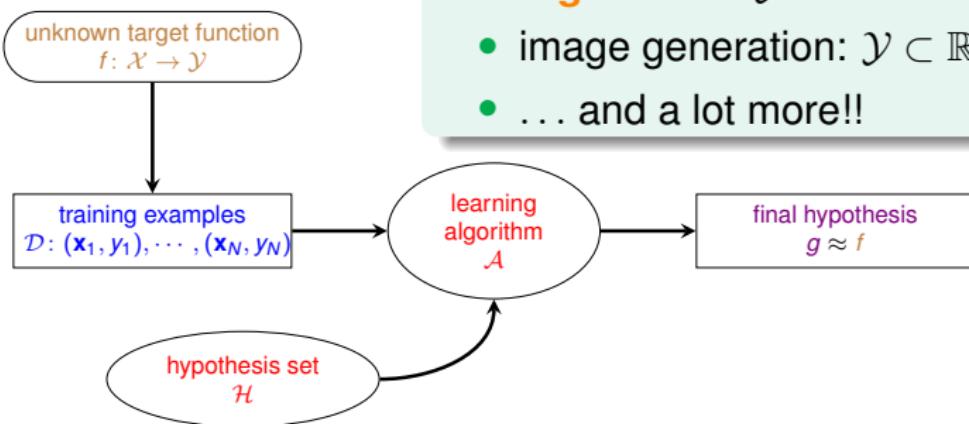
Other Image Generation Problems

- noisy image ⇒ clean image
- low-resolution image ⇒ high-resolution image

\mathcal{Y} : a ‘manifold’ $\subset \mathbb{R}^{w \times h \times c}$,
arguably **not just multi-pixel regression**

Learning with Different Output Space \mathcal{Y}

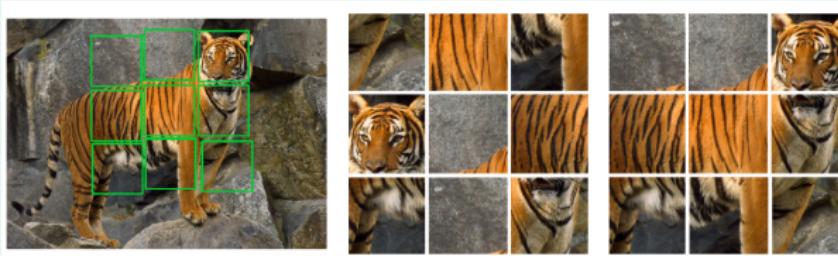
- **binary classification:** $\mathcal{Y} = \{-1, +1\}$
- multiclass classification: $\mathcal{Y} = \{1, 2, \dots, K\}$
- multilabel classification: $\mathcal{Y} = 2^{\{1, 2, \dots, K\}}$
- **regression:** $\mathcal{Y} = \mathbb{R}$
- image generation: $\mathcal{Y} \subset \mathbb{R}^{w \times h \times c}$
- ... and a lot more!!



core tools: binary classification and regression

Self-supervised: Unsupervised + Self-defined Goal(s)

jigsaw puzzle: pieces → full picture



(Figure 1 of Noroozi and Favaro,

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV 2016)

Other Popular Goals

- colorization: grayscale image → colored image
- center word prediction: chunk of text → center word
- next sentence prediction: sentence A → is sentence B next?

self-supervised learning: recipe to learn
'physical knowledge' before actual task

Weakly-supervised: Learning without True y_n

complementary label: \bar{y}_n ('not' label) instead of y_n



(Figure 1 of Yu et al., Learning with Biased Complementary Labels, ECCV 2018)

Other Weak Supervisions

- partial label: a set Y_n that contains true y_n
- noisy label: y'_n , a noisy version of true y_n
- proportion label: aggregated statistics of a set of y_n

weakly-supervised learning: another
realistic (?) family to reduce labeling burden

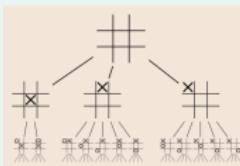
THE Most Well-known Reinforcement Learning Agent



(Public Domain, from Wikipedia; used here for education purpose; all other rights still belong to Google DeepMind)

Non-ML Techniques

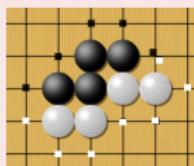
Monte C. Tree Search
≈ **move simulation** in
brain



(CC-BY-SA 3.0 by Stannered on
Wikipedia)

ML Techniques

Deep Learning
≈ **board analysis** in
human brain



(CC-BY-SA 2.0 by Frej Björn on
Wikipedia)

Reinforcement Learn.
≈ **(self)-practice** in
human training



(Public Domain, from Wikipedia)

good AI: important to use the **right**
techniques—ML & others, including human

The LATEST Well-known RL Agent



(Public Domain, from Wikipedia; used here for education purpose; all other rights still belong to OpenAI)

GPT-3

Self-Supervised

- mainly **next-token prediction** from 2048 tokens
- **175 billion** parameters trained with **500 billion** tokens

chatGPT

Supervised (Few-Shot) + Supervised (Ranking) + Reinforcement

Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

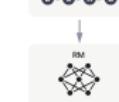
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



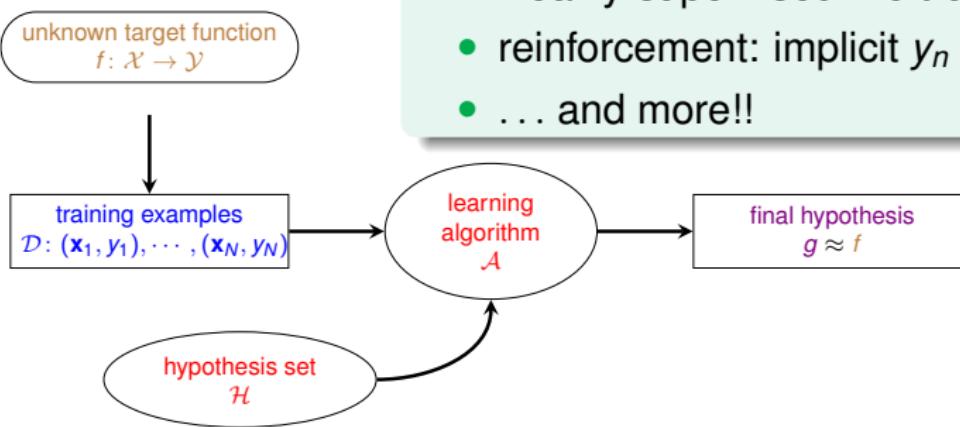
The reward is used to update the policy using PPO.



staged-ML important for building huge ML systems

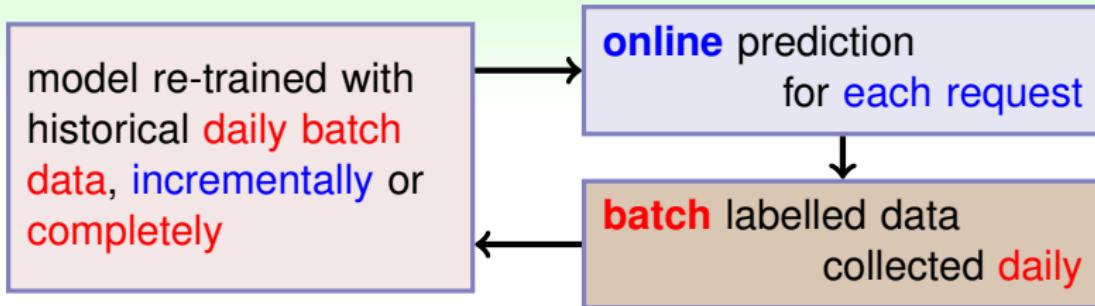
Learning with Different Data Label y_n

- **supervised:** all y_n
- unsupervised: no y_n
- self-supervised: self-defined y'_n from \mathbf{x}_n
- semi-supervised: some y_n
- weakly-supervised: no true y_n
- reinforcement: implicit y_n by goodness(\tilde{y}_n)
- ... and more!!



core tool: supervised learning

Online + Batch for Real-World Applications



purely online

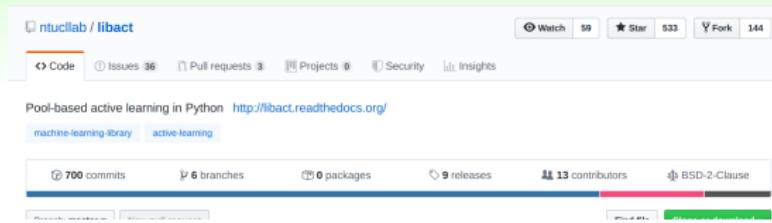
- **incremental update** costly online
- **delayed labels** hard to handle properly

purely batch

- cannot capture **drifts/trends** well
- **complete re-training** possibly costly

real-world ML system
different from **textbook settings**

Making Active Learning More Realistic



open-source tool libact developed by NTU CLLab (Yang, 2017)

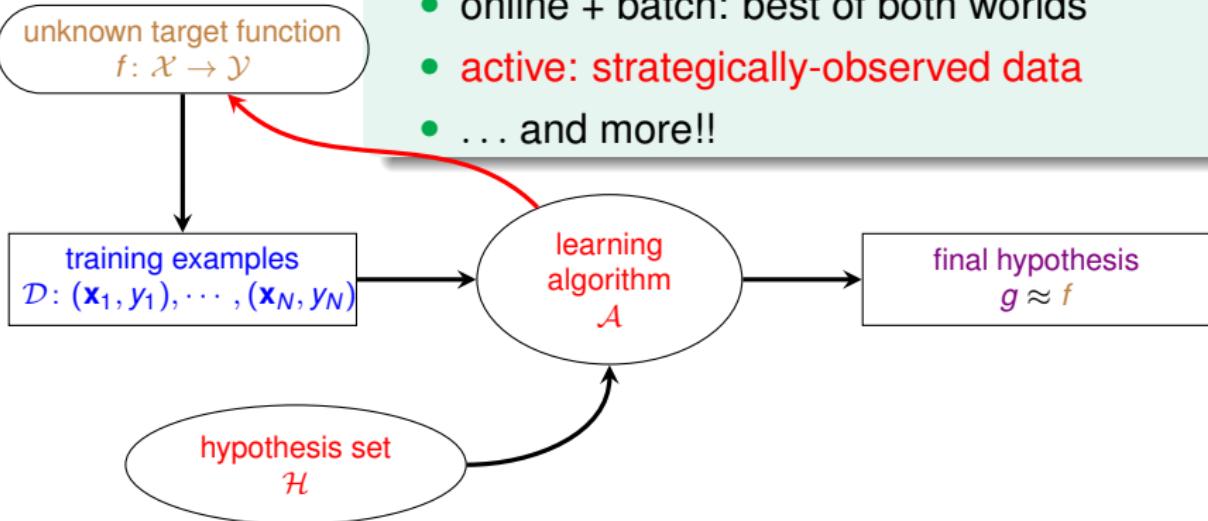
<https://github.com/ntucllab/libact>

- including many popular strategies
- received > 500 stars and continuous issues

“libact is a Python package designed to **make active learning easier** for real-world users”

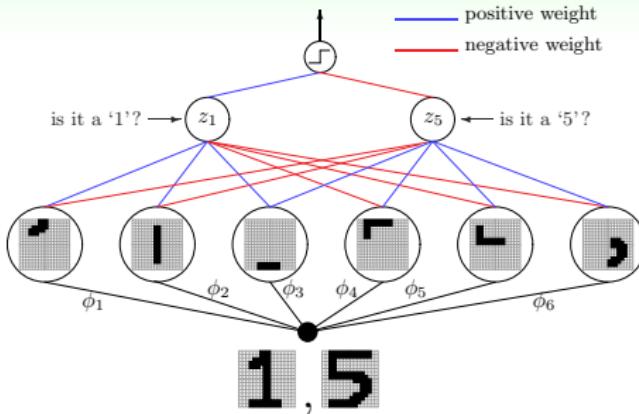
Learning with Different Protocol $f \Rightarrow (\mathbf{x}_n, y_n)$

- **batch**: all known data
- online: sequential (passive) data
- online + batch: best of both worlds
- **active**: strategically-observed data
- ... and more!!



core protocol: batch

Deep Learning: ‘Automatic’ Conversion from Raw to Concrete



- layered extraction: **simple** to **complex** features
- natural for **difficult** learning task with **raw features**, like **vision**

deep learning: currently popular in
vision/speech/...