CSE 701 SEM: Some Recent Progresses in Machine Learning

Lecture 1: Introduction

02/01/2023

Instructor: Kaiyi Ji

University at Buffalo
The State University of New York

About Me

- Instructor: Kaiyi Ji, kaiyiji@buffalo.edu
- Office location: Davis Hall 338G
- Course website: *https://cse.buffalo.edu/~kaiyiji/cse701.html*
- Piazza: *https://piazza.com/class/ldkjmlkswt3131*
- Office hours: email me for appointments
- Research interests:
  - ✓ Large-scale optimization for machine learning
  - ✓ Efficient continual learning
  - ✓ Federated learning

Course Description

- Prerequisites: CSE 474/574 Introduction to Machine Learning or related courses. Good math.

- What we need to do?
  - ✓ Read papers listed on course website
    - Optimization/generalization/algorithms/applications/learning
  - ✓ Learn from these materials
    - Summary/presentation/implementation (highly recommend)

- Goal of this course
  - ✓ Understand algorithm design and theoretical properties in ML
  - ✓ Learn how to design smart algorithms in DL (beyond off-shelf methods)
  - ✓ Practice skills of paper reading, presentation, and summary

Logistics

- Location: Talbrt 103.

- Time: Wednesday 4:00PM - 6:50PM

- Suggested references:

  - Z. Allen-Zhu, Y. Li, and Z. Song, *"A convergence theory for deep learning via over-parameterization",* ICML 2019.

  - L. Bottou, F.E Curtis and J. Nocedal. "Optimization methods for large-scale machine learning," Siam Review, 2018.

  - I. Goodfellow, Y. Bengio, A. Courville. *"Deep learning,"* MIT press, 2016.

Logistics

- Each lecture will present a topic covering 2~3 papers
  - Paper(s) listed in **course website (see schedule)**
  - Read paper(s) before each lecture
- Write **one-page** summary after each lecture
  - If there are more than one papers, pick **one** up to you
  - Due: the day before next lecture

Logistics

- Each student presents one selected paper
  - Sign up a paper by **Today, 11:59 pm**: <Link in course website>
  - Presentation: 30-50 min, 20-40 slides
- Some bonus:
  - Skip summary of your presentation paper
  - Sign up next lecture (2/8): can skip **two more** summaries

Logistics

- **Grading policy:**
  - 30% for class participation
  - 35% for writing summaries (14 summaries $\times$ 2.5%)
  - 35% for presentation

Academic Integrity

- Writing paper summaries independently
  - Do not copy others's work or solution
  - Any plagiarism will result in a **F** score
- Any reference used in your presentation must be cited
  - Online resources: authors' slides
- Academic integrity policies can be found at
  - *https://engineering.buffalo.edu/computer-science-engineering/informati on-for-students/academics/academic-integrity.html.*

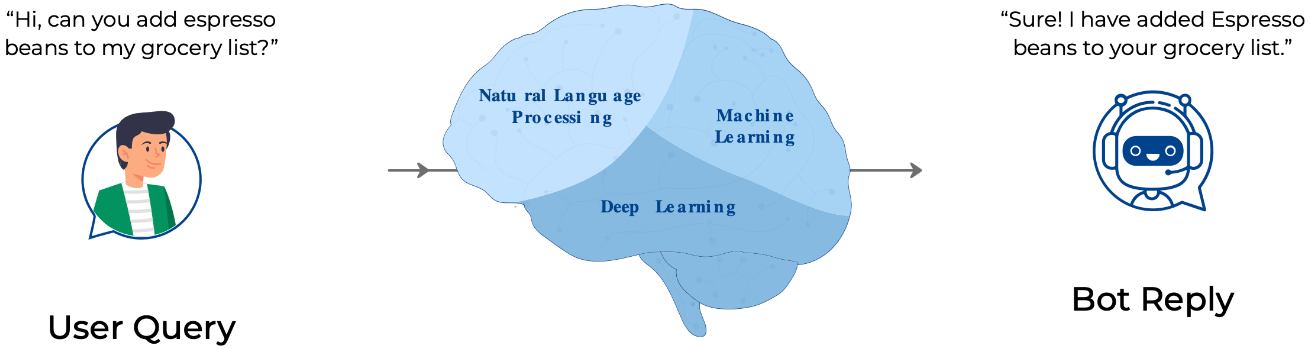
Today's Plan

- A short background introduction to recent process on:
  - Optimization in ML and tools
  - Theoretical analysis of learning neural networks
  - Emerging ML/DL paradigms

- Submit your summary before next Tuesday (2/7, 11:59 pm)
  - Under assignment/summary_1 folder in Piazza
  - Please send a **private** message and attach your **summary pdf**

Machine Learning

CV: objective detection



NLP: conversational user interface

Source: GenieTalk.ai

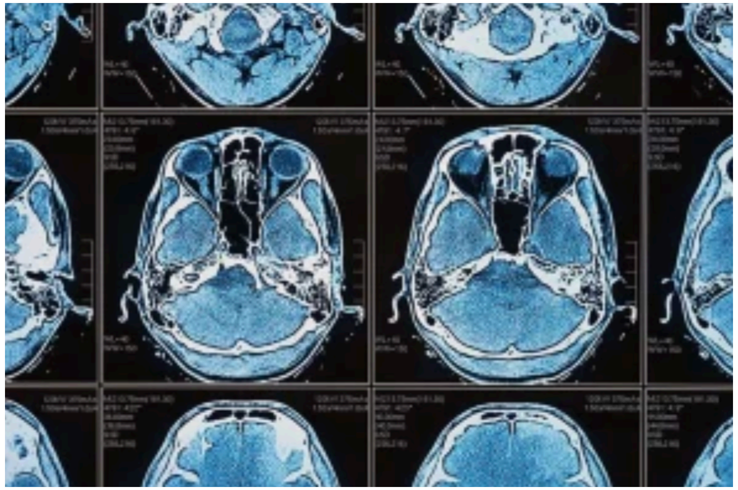


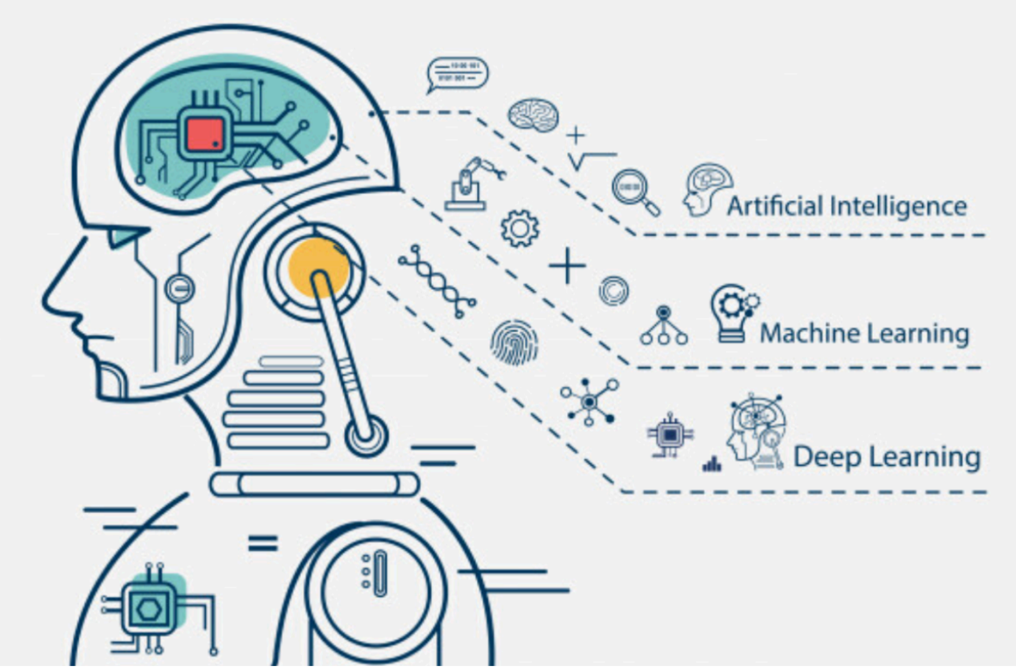
Image processing: disease detection

DL in autonomous driving

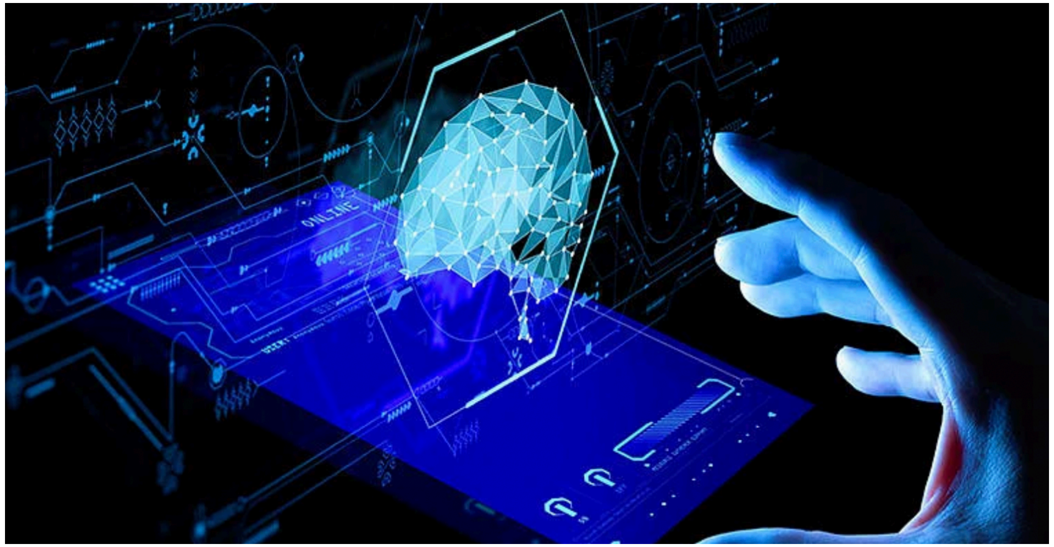
Source: Tian’s blog

Optimization in Machine Learning

- How to train machine?



Source: nearlearn.com



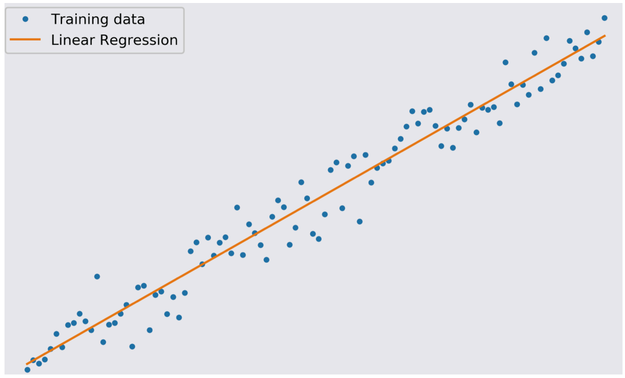
Source: optimization by TensorFlow, Loon's blog

- **Optimization** is **engine** to train intelligence of machine!

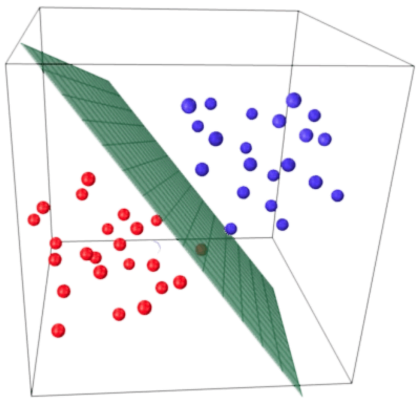
Convex Optimization in ML

- Linear regression and classification
  - Well-shaped convex loss function averaged over data samples (feature-label pairs)

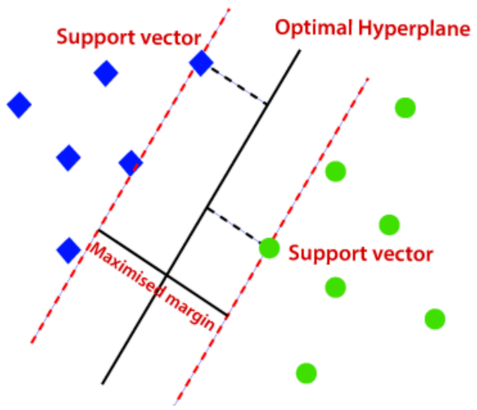
- Support vector machine (SVM)
  - Margin based classification, **strong theoretical guarantee**



Linear regression



Linear classification



SVM in classification

Optimization in ML

- Linear classification, SVM, logistic regression, etc.
  - A averaged loss measures the classifier quality
- Find **lowest** loss —> find best classifier

Random search lower loss?

```
for num in xrange(1000):
  W = np.random.randn(10, 3073) * 0.0001 # generate random parameters
  loss = L(X_train, Y_train, W) # get the loss over the entire training set
  if loss < bestloss: # keep track of the best solution
    bestloss = loss
    bestW = W
```

0.15 accuracy << 0.95 SOTA

Picture and example from
stanford cs231n

Optimization in ML

- Use slope information

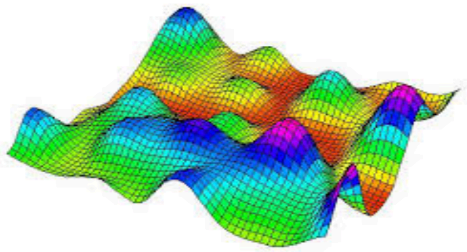
- Gradient: direction to decrease loss!
  - $\nabla L(W) = \dfrac{dL(W)}{dW}$: compute **derivative** of $L(W)$ w.r.t. W
  - Deep learning with NNs: use automatic differentiation (e.g., backpropagation)

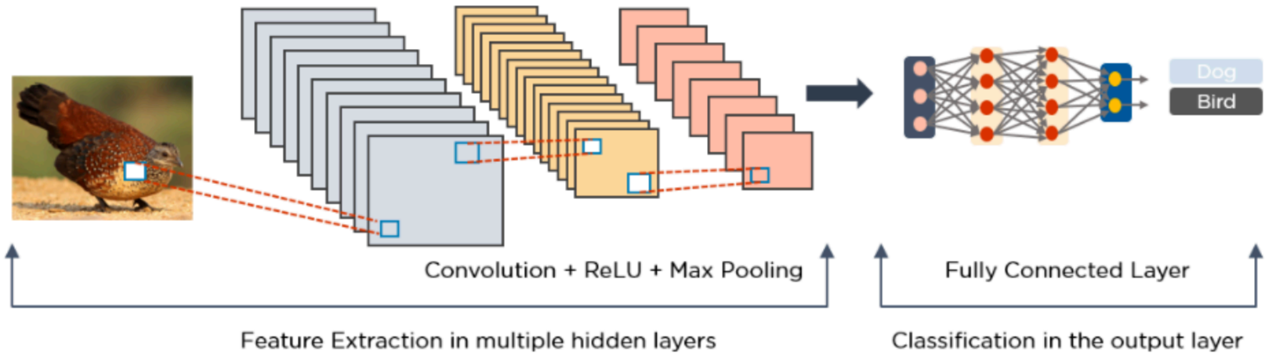
```
from torch.autograd import grad
```

```
grads = torch.autograd.grad(output, inputs, grad_outputs=grad_outputs, allow_unused=True,
                            retain_graph=retain_graph, create_graph=create_graph)
```

Nonconvex Optimization in ML

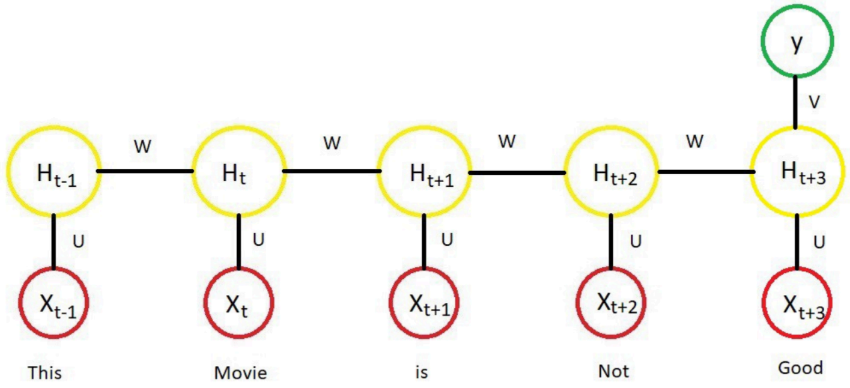
- Training with deep neural networks (highly nonconvex landscape)
  - Multi-layer perceptron (MLP) in classification/regression
  - Convolutional neural networks (CNN) in image processing, vision, etc
  - Recurrent neural networks (RNN), transformer in natural language processing (NLP)





CNN for image processing

Resource: Avijeet Biswal's lesson

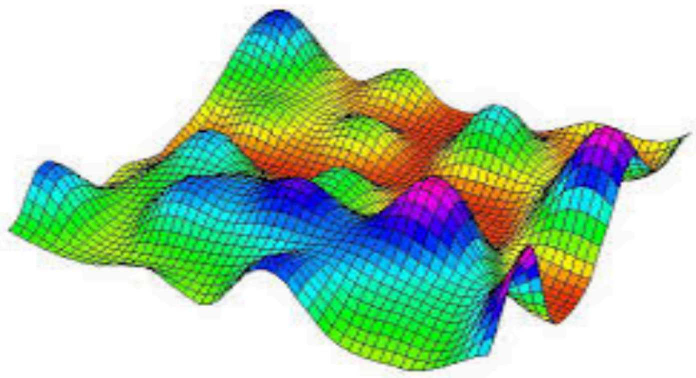


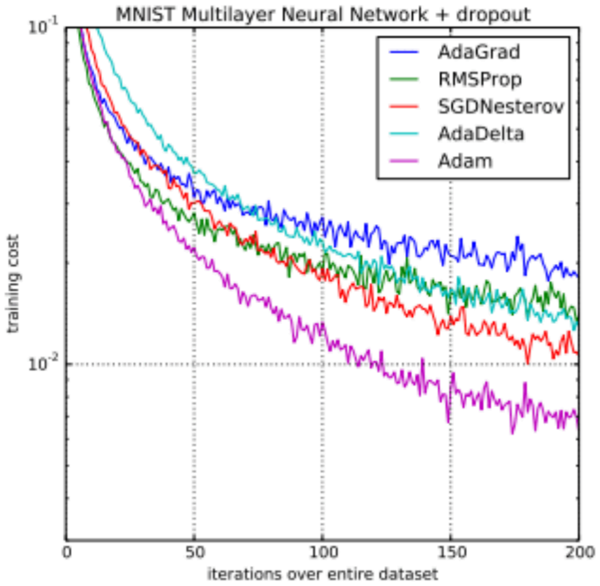
LSTM in NLP

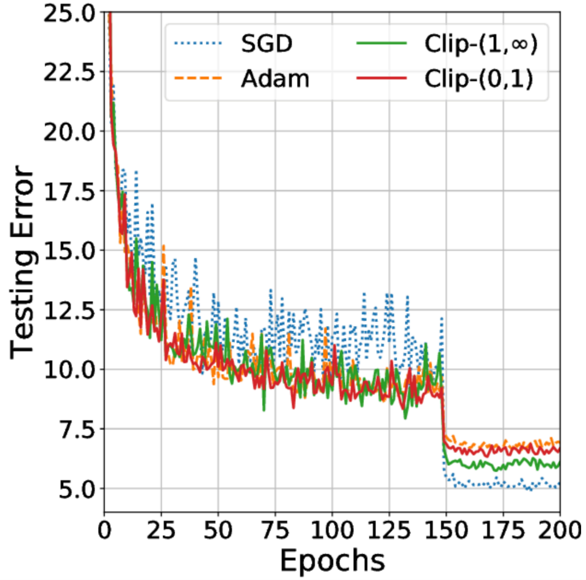
Resource: Vibhor Nigam's blog

Nonconvex Optimization in ML

- Popular solutions:
  - Stochastic gradient descent (SGD)
  - SGD with Nesterov Momentum
  - Adaptive optimizers: Adam and its variants

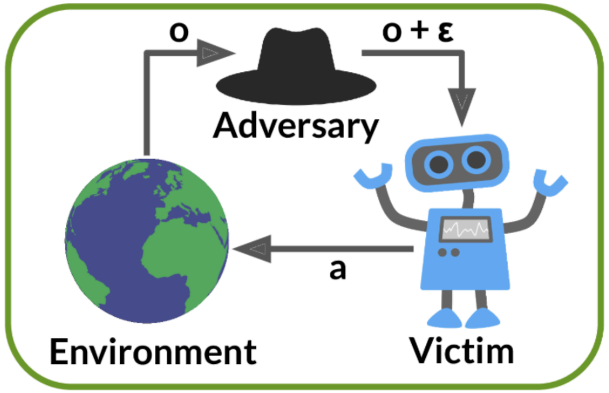






Blackbox Optimization in ML

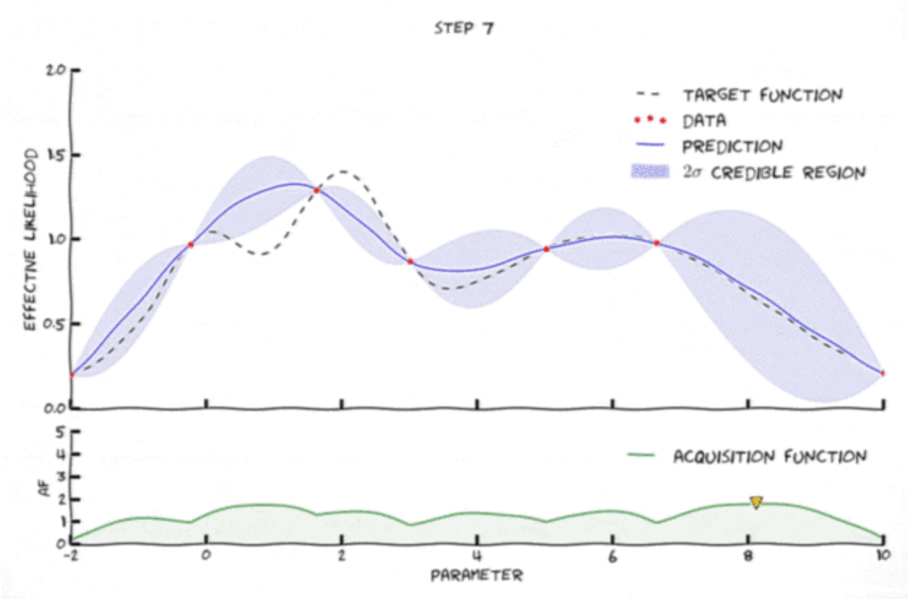
- Emerging applications in ML
  - Adversarial attack: (attacker does not know detailed information but direct feedback)
  - Practical systems: too complex to capture underlying structure
  - Gradient information is expensive to get (non-smooth or non-differentiable)

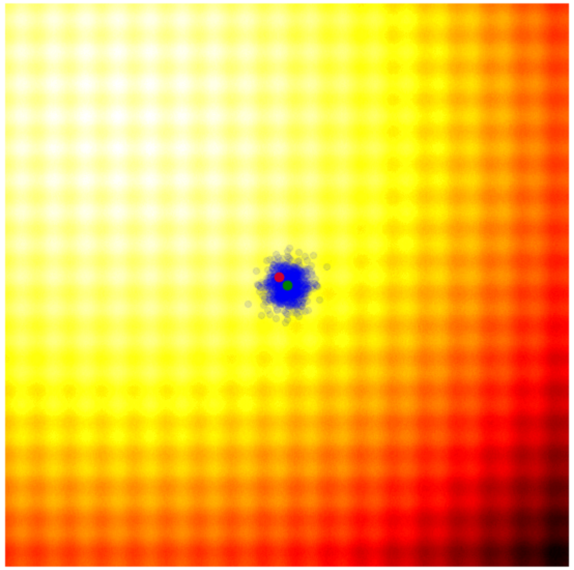


How and what we can do??

Blackbox Optimization in ML

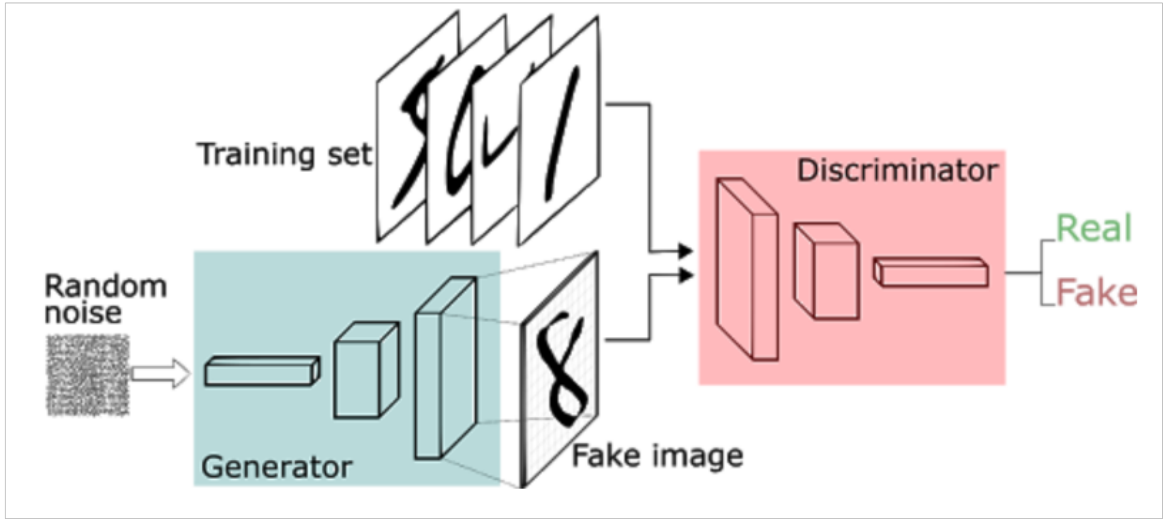
- Tools we can use:
  - Zeroth-order optimization -> function values to estimate gradients
  - Bayesian optimization -> approximate function values via Gaussian process
  - Evolution strategies -> random exploration + find local lowest point



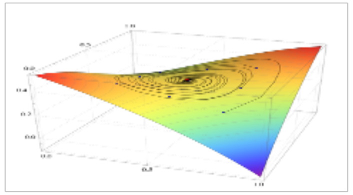


Minimax Optimization in ML

- Generative adversarial networks (GANs): data generation



$$\min_{\theta\in\Theta}\max_{w\in\mathcal{W}} \quad L(\theta, w)$$

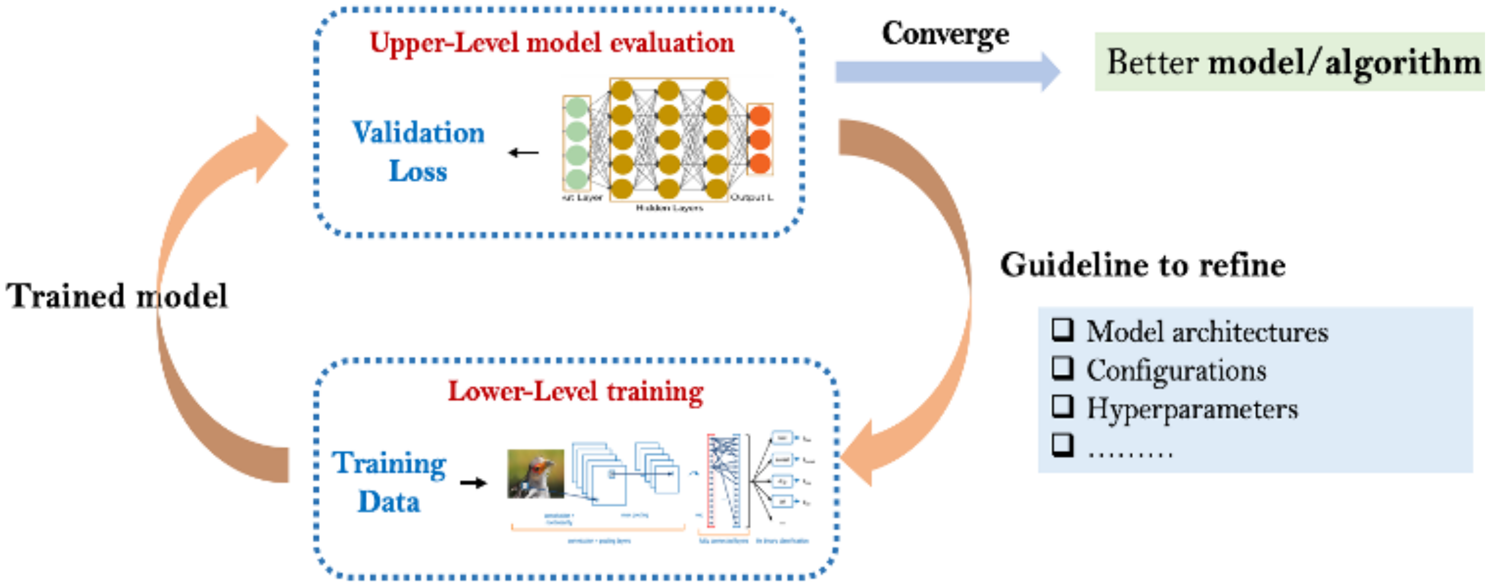


Minimax optimization

- Imbalanced data classification (deep AUC maximization)

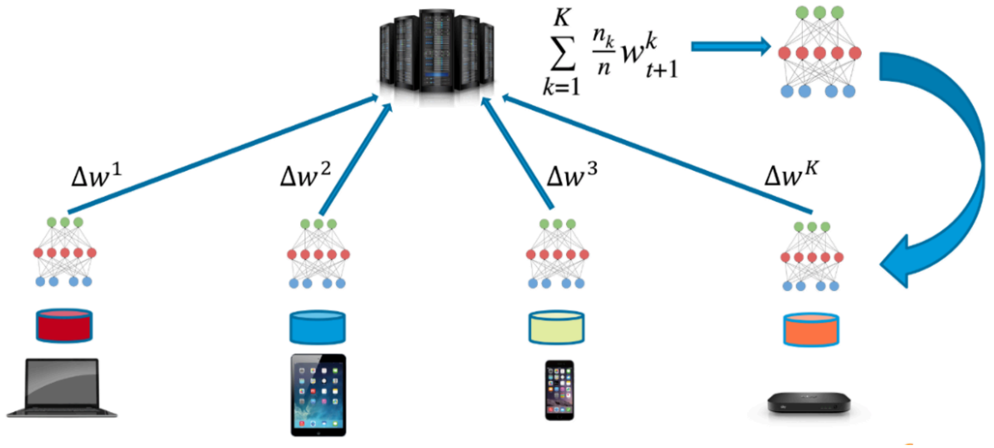
Bilevel Optimization in ML

- Make ML learn better and faster
  - Model pruning
  - Hyperparameter optimization
  - Neural architecture search
  - Fair ML
  - Federated learning

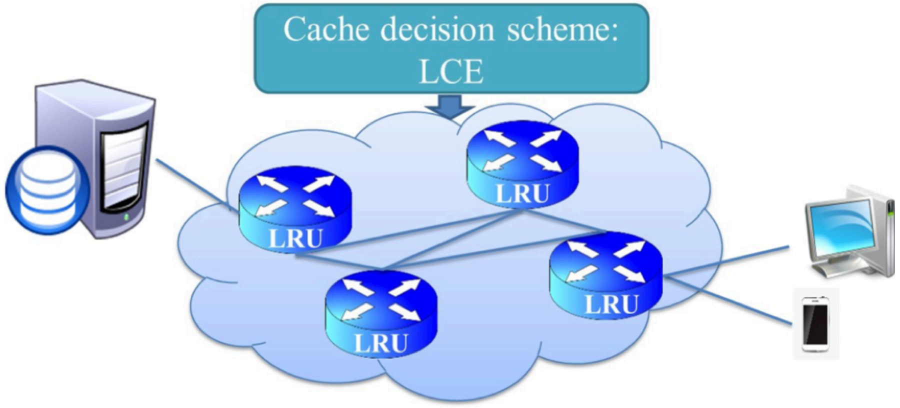


Distributed Optimization in ML

- Decentralized/federated/distributed learning over networks
  - Improve scalability over big data and huge models



Federated learning with edge devices



Distributed protocol in Internet!

Second Part: Theory in Deep Learning

- Theory on learning neural networks
  - Why SGD finds good solution in complex neural networks?
  - What is mechanism hidden in training process?

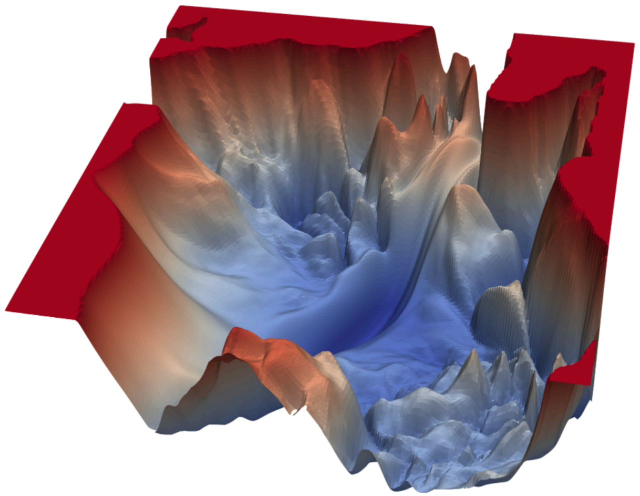
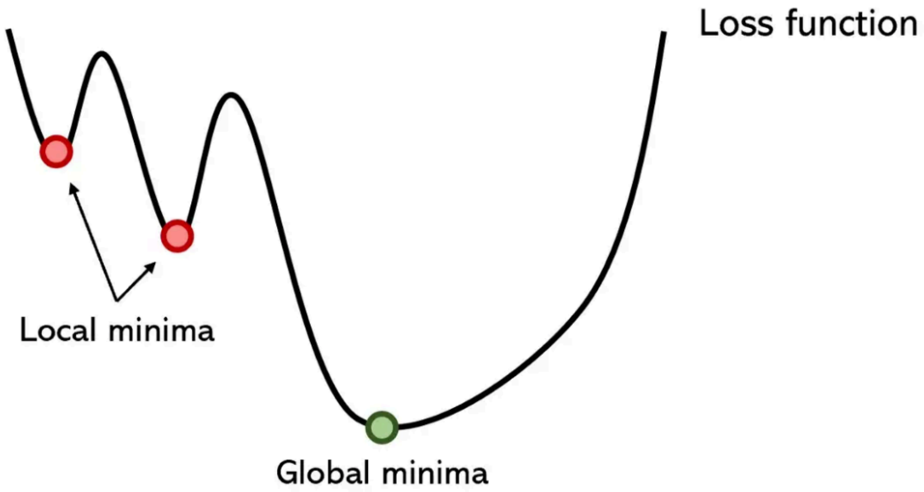
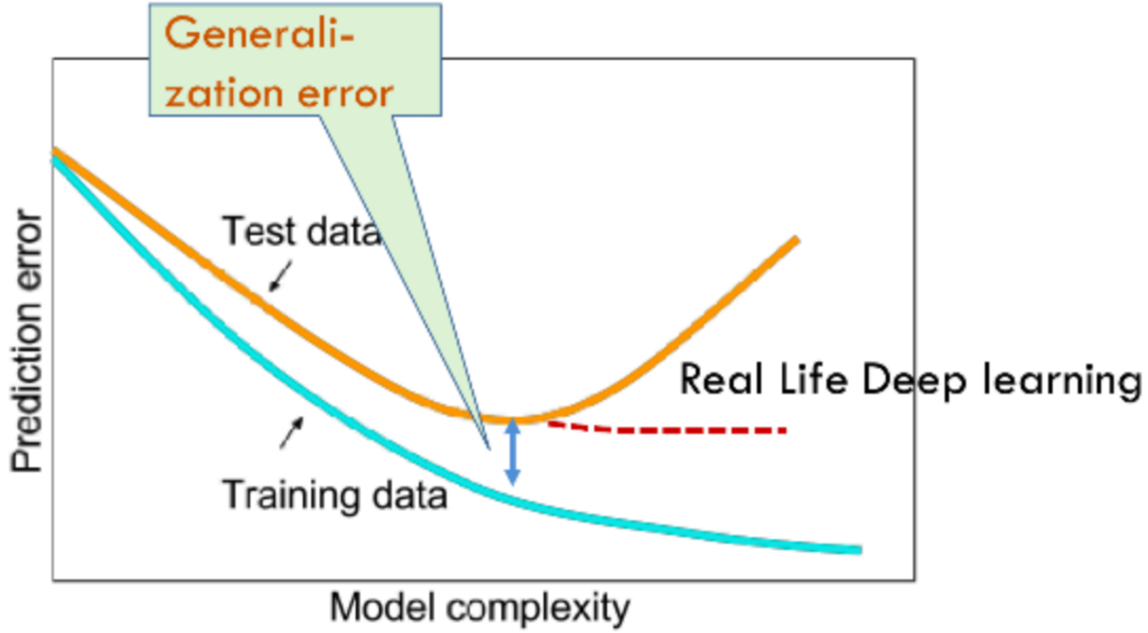


Fig. 2.37 Visualizing the loss landscape of neural nets [Li et al., 2018].



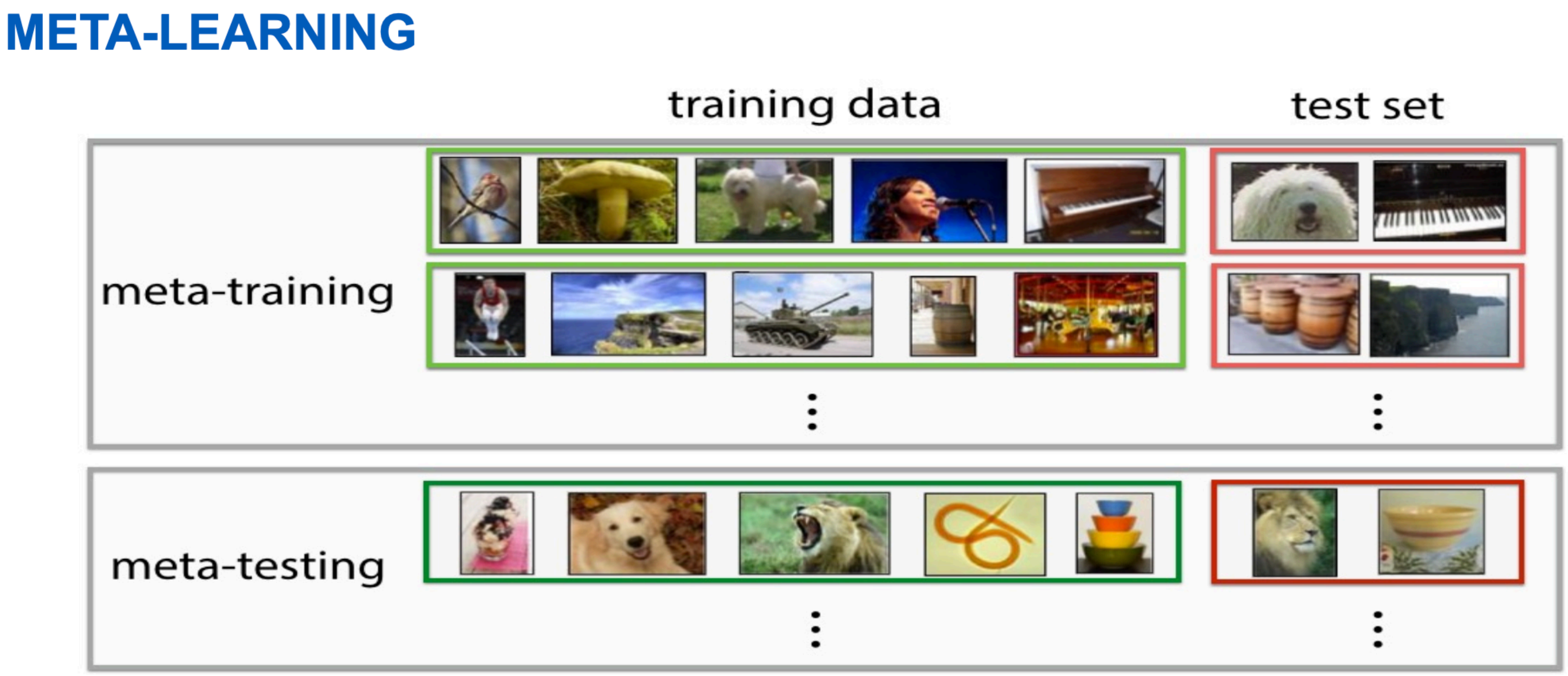
Second Part: Theory in Deep Learning

- Generalization analysis of ML models
  - Why learning on training data implies good performance on test data?
  - What if training and test data have distribution shift?
  - What does overfitting implies?
  - Overfitting always bad?



Third Part: Recent Popular Learning Paradigm

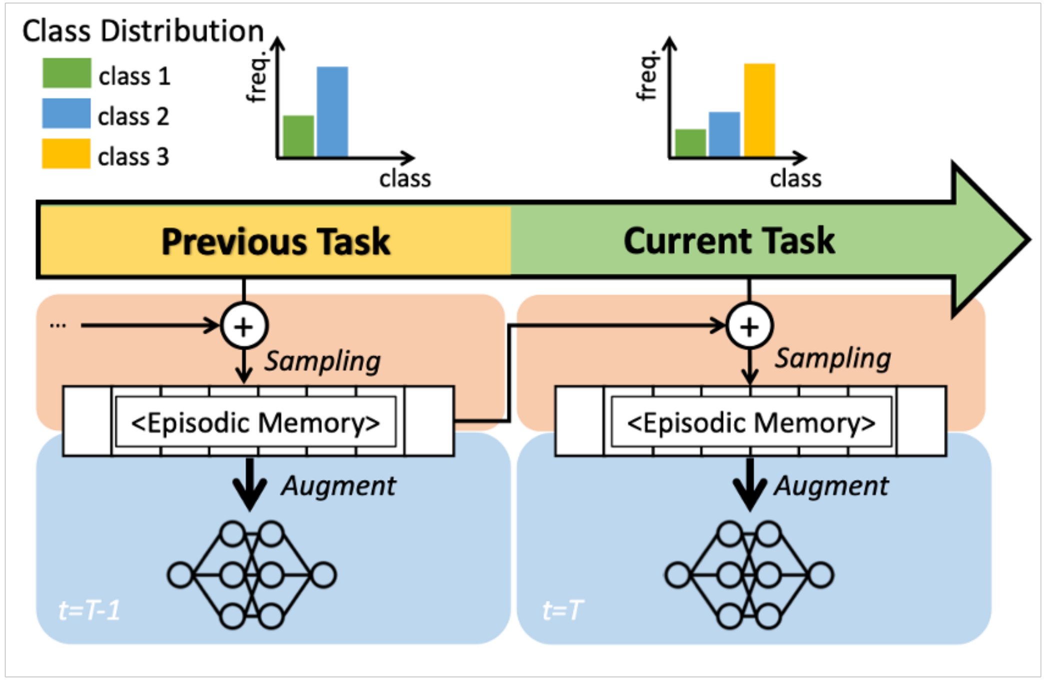
- Meta-learning or few-shot learning
  - Extract useful prior information from past tasks (learner)
  - Use this information to improve and accelerate training (meta-learner)



Third Part: Recent Popular Learning Paradigm

- Continual learning
  - Deal with catastrophic forgetting
  - Neural structure, memory replay, gradient align based methods

❖ Learning over a long-time horizon



❖ Replay memory (RM)

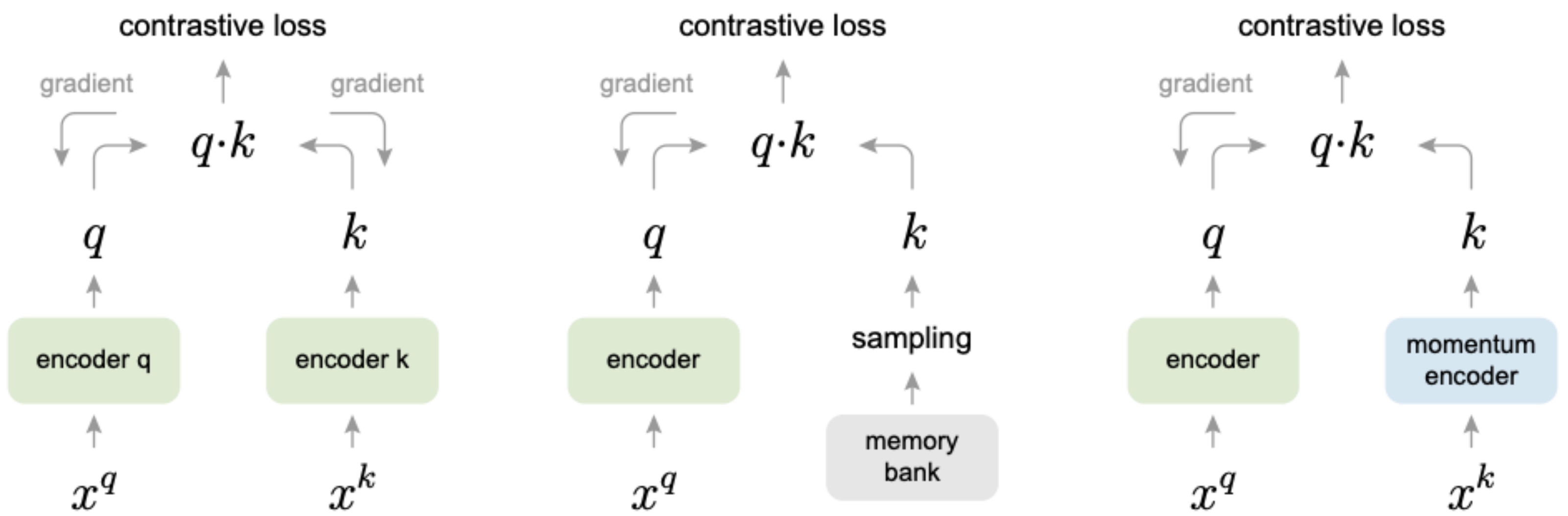
- ❑ Finite memory:
  - Cannot accommodate all previous data
- ❑ Store subset of previous samples in RM
  - Partial previous knowledge revisit

❖ Data summaries for replay memory

- ❑ Previous samples **not equally** important
- ❑ How to select most representative ones?

Third Part: Recent Popular Learning Paradigm

- Contrastive learning
  - Unsupervised learning; data has no labels
  - Representation based approaches; data augmentation (resize, rotate, noise, flip,….)



(a) end-to-end (b) memory bank (c) MoCo

Works terribly with small batchsizes, why and how to resolve?