The Meta-Learning Problem & Black-Box Meta-Learning

CS 330
Logistics

Optional Homework 0 due tonight.

Homework 1 posted to be today, due Wednesday, October 6
Azures guide to be posted today

One more CA!

Edwin Pan
Teaching Assistant
OH: Sat 5:00-7:00pm
Plan for Today

**Transfer Learning**
- Problem formulation
- Fine-tuning

**Meta-Learning**
- Problem formulation
- General recipe of meta-learning algorithms
- Black-box adaptation approaches
- Case study of GPT-3 (time-permitting)

Goals for by the end of lecture:
- Differences between multi-task learning, transfer learning, and meta-learning problems
- Basics of transfer learning via fine-tuning
- Training set-up for few-shot meta-learning algorithms
- How to implement black-box meta-learning techniques
Multi-Task Learning vs. Transfer Learning

**Multi-Task Learning**
Solve multiple tasks $\mathcal{T}_1, \cdots, \mathcal{T}_T$ at once.

\[
\min_{\theta} \sum_{i=1}^{T} \mathcal{L}_i(\theta, \mathcal{D}_i)
\]

**Transfer Learning**
Solve target task $\mathcal{T}_b$ after solving source task $\mathcal{T}_a$ by transferring knowledge learned from $\mathcal{T}_a$.

**Key assumption:** Cannot access data $\mathcal{D}_a$ during transfer.

Transfer learning is a valid solution to multi-task learning. (but not vice versa)

**Question:** What are some problems/applications where transfer learning might make sense?

- when $\mathcal{D}_a$ is very large
  (don’t want to retain & retrain on $\mathcal{D}_a$)

- when you don’t care about solving $\mathcal{T}_a$ & $\mathcal{T}_b$ simultaneously
Transfer learning via fine-tuning

\[ \phi \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_{tr}) \]

(typically for many gradient steps)

Where do you get the pre-trained parameters?
- ImageNet classification
- Models trained on large language corpora (BERT, LMs)
- Other unsupervised learning techniques
- Whatever large, diverse dataset you might have

Some common practices
- Fine-tune with a smaller learning rate
- Smaller learning rate for earlier layers
- Freeze earlier layers, gradually unfreeze
- Reinitialize last layer
- Search over hyperparameters via cross-val
- Architecture choices matter (e.g. ResNets)

What makes ImageNet good for transfer learning?
Huh, Agrawal, Efros. ‘16

<table>
<thead>
<tr>
<th>Pre-trained Dataset</th>
<th>PASCAL</th>
<th>SUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>58.3</td>
<td>52.2</td>
</tr>
<tr>
<td>Random</td>
<td>41.3 [21]</td>
<td>35.7 [2]</td>
</tr>
</tbody>
</table>

Pre-trained models often available online.
Universal Language Model Fine-Tuning for Text Classification. Howard, Ruder. ‘18

![Graphs showing validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).]

Figure 3: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).

Fine-tuning doesn’t work well with very small target task datasets

This is where meta-learning can help.
Plan for Today

Transfer Learning
- Problem formulation
- Fine-tuning

Meta-Learning
- **Problem formulation**
- General recipe of meta-learning algorithms
- Black-box adaptation approaches
- Case study of GPT-3 (time-permitting)
The Meta-Learning Problem Statement

(that we will consider in this class)
Two ways to view meta-learning algorithms

Mechanistic view

➢ Deep network that can read in an entire dataset and make predictions for new datapoints
➢ Training this network uses a meta-dataset, which itself consists of many datasets, each for a different task

Probabilistic view

➢ Extract prior knowledge from a set of tasks that allows efficient learning of new tasks
➢ Learning a new task uses this prior and (small) training set to infer most likely posterior parameters

For now: Focus primarily on the mechanistic view.

(Bayes will come back later)
How does meta-learning work? An example.

Given 1 example of 5 classes:

- **training data** $D_{\text{train}}$
- **test set** $X_{\text{test}}$

Classify new examples
How does meta-learning work? An example.

<table>
<thead>
<tr>
<th>meta-training</th>
<th>( \mathcal{T}_1 )</th>
<th>( \mathcal{T}_2 )</th>
<th>...</th>
<th>( \mathcal{T}_n )</th>
<th>training classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td></td>
<td><img src="image3.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td></td>
<td><img src="image6.png" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

Given 1 example of 5 classes:

meta-testing \( \mathcal{T}_{test} \)

<table>
<thead>
<tr>
<th>training data</th>
<th>( D_{train} )</th>
<th>test set</th>
<th>( X_{test} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Can replace image classification with: regression, language generation, skill learning, any ML problem.
The Meta-Learning Problem

Given data from $\mathcal{T}_1, \ldots, \mathcal{T}_n$, quickly solve new task $\mathcal{T}_{\text{test}}$

**Key assumption:** meta-training tasks and meta-test task drawn i.i.d. from same task distribution

$$\mathcal{T}_1, \ldots, \mathcal{T}_n \sim p(\mathcal{T}), \mathcal{T}_j \sim p(\mathcal{T})$$

Like before, tasks must share structure.

What do the tasks correspond to?
- recognizing handwritten digits from different languages (see homework 1!)
- giving feedback to students on different exams
- classifying species in different regions of the world
- a robot performing different tasks

How many tasks do you need? The more the better. (analogous to more data in ML)
Some terminology

- task training set \( D_{i}^{\text{tr}} \) “support set”
- task test dataset \( D_{i}^{\text{test}} \) “query set”

**k-shot learning**: learning with \( k \) examples per class (or \( k \) examples total for regression)

**N-way classification**: choosing between \( N \) classes

**Question**: What are \( k \) and \( N \) for the above example?
Problem Settings Recap

**Multi-Task Learning**
Solve multiple tasks $\mathcal{T}_1, \ldots, \mathcal{T}_T$ at once.

$$\min_{\theta} \sum_{i=1}^{T} \mathcal{L}_i(\theta, \mathcal{D}_i)$$

**Transfer Learning**
Solve target task $\mathcal{T}_b$ after solving source task $\mathcal{T}_a$
by *transferring* knowledge learned from $\mathcal{T}_a$

**The Meta-Learning Problem**
Given data from $\mathcal{T}_1, \ldots, \mathcal{T}_n$, quickly solve new task $\mathcal{T}_{test}$

In *transfer learning* and *meta-learning*:
generally impractical to access prior tasks

In all settings: tasks must share structure.
Plan for Today

Transfer Learning
- Problem formulation
- Fine-tuning

Meta-Learning
- Problem formulation
- **General recipe of meta-learning algorithms**
- Black-box adaptation approaches
- Case study of GPT-3 (time-permitting)
One View on the Meta-Learning Problem

Supervised Learning:

Inputs: $\mathbf{x}$  
Outputs: $\mathbf{y}$  
Data: $\{(\mathbf{x}, \mathbf{y})_i\}$

$\mathbf{y} = g_\phi(\mathbf{x})$

Meta Supervised Learning:

Inputs: $\mathcal{D}_\text{tr}^{\text{tr}}$ $\mathbf{x}_\text{ts}$  
Outputs: $\mathbf{y}_\text{ts}$  
Data: $\mathcal{D}_i$  
$\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$

$\{(\mathbf{x}, \mathbf{y})_1:K\}$  
$\mathbf{y}_\text{ts} = f_\theta(\mathcal{D}_\text{tr}^{\text{tr}}, \mathbf{x}_\text{ts})$

Why is this view useful?
Reduces the meta-learning problem to the design & optimization of $f$.

General recipe

How to design a meta-learning algorithm

1. Choose a form of $f_\theta(D^{tr}, x^{ts})$

2. Choose how to optimize $\theta$ w.r.t. max-likelihood objective using meta-training data

meta-parameters
Plan for Today

Transfer Learning
- Problem formulation
- Fine-tuning

Meta-Learning
- Problem formulation
- General recipe of meta-learning algorithms
- **Black-box adaptation approaches**
- Case study of GPT-3 (time-permitting)
Black-Box Adaptation

**Key idea:** Train a neural network to represent \( \phi_i = f_\theta(D_{i\text{tr}}) \) "learner"

Predict test points with \( y^{ts} = g_{\phi_i}(x^{ts}) \)

\[
\begin{align*}
\max_\theta \sum_{T_i} \sum_{(x,y) \sim D_{i\text{test}}} \log g_{\phi_i}(y|x) \\
\mathcal{L}(\phi_i, D_{i\text{test}}) \\
\max_\theta \sum_{T_i} \mathcal{L}(f_\theta(D_{i\text{tr}}), D_{i\text{test}})
\end{align*}
\]
Black-Box Adaptation

**Key idea:** Train a neural network to represent \( \phi_i = f_\theta(D_i^{tr}) \).

1. Sample task \( \mathcal{T}_i \) (or mini batch of tasks)
2. Sample disjoint datasets \( D_i^{tr}, D_i^{test} \) from \( D_i \)
Black-Box Adaptation

**Key idea:** Train a neural network to represent $\phi_i = f_\theta(D_{i}^{tr})$.

1. Sample task $\mathcal{T}_i$ (or mini batch of tasks)
2. Sample disjoint datasets $D_{i}^{tr}, D_{i}^{test}$ from $D_i$
3. Compute $\phi_i \leftarrow f_\theta(D_{i}^{tr})$
4. Update $\theta$ using $\nabla_\theta \mathcal{L}(\phi_i, D_{i}^{test})$
Black-Box Adaptation

**Key idea:** Train a neural network to represent \( \phi_i = f_\theta(D^{tr}_i) \).

**Challenge**
Outputting all neural net parameters does not seem scalable?

**Idea:** Do not need to output all parameters of neural net, only sufficient statistics

(Santoro et al. MANN, Mishra et al. SNAIL)

low-dimensional vector \( h_i \)

represents contextual task information

\[ \phi_i = \{ h_i, \theta_g \} \]

recall:

**general form:** \( y^{ts} = f_\theta(D^{tr}_i, x^{ts}) \)

What architecture should we use for \( f_\theta \)?

\[ \begin{aligned}
(x_1, y_1) & \quad (x_2, y_2) & \quad (x_3, y_3) \\
\downarrow & \quad \downarrow & \quad \downarrow \\
\{ h_i \} & \quad \{ g_\phi_i \} & \quad y^{ts}
\end{aligned} \]
Black-Box Adaptation Architectures

LSTMs or Neural turing machine (NTM)

Feedforward + average

Other external memory mechanisms

Meta Networks
Munkhdalai, Yu. ICML ’17

Convolutions & attention

Meta-Learning with Memory-Augmented Neural Networks
Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML ‘16

Conditional Neural Processes
Garnelo, Rosenbaum, Maddison, Ramalho, Saxton, Shanahan, Teh, Rezende, Eslami. ICML ’18

Question: Why might feedforward+average be better than a recurrent model?

HW 1:
- implement data processing
- implement simple black-box meta-learner
- train few-shot Omniglot classifier
Let’s run through an example

Omniglot dataset  Lake et al. Science 2015

1623 characters from 50 different alphabets

Hebrew  Bengali  Greek  Futurama

many classes, few examples
the “transpose” of MNIST
statistics more reflective of the real world

20 instances of each character

More few-shot image recognition datasets: tieredImageNet, CIFAR, CUB, CelebA, ORBIT, others

More benchmarks: molecular property prediction (Nguyen et al. ’20), object pose prediction (Yin et al. ICLR ’20), channel coding (Li et al. ’21)
Black-Box Adaptation

**Key idea:** Train a neural network to represent $\phi_i = f_\theta(D_i^{tr})$.

+ expressive
+ easy to combine with variety of learning problems (e.g. SL, RL)
- complex model w/ complex task: challenging optimization problem
- often data-inefficient

How else can we represent $\phi_i = f_\theta(D_i^{tr})$?

**Next time (Wednesday):** What if we treat it as an optimization procedure?
Plan for Today

**Transfer Learning**
- Problem formulation
- Fine-tuning

**Meta-Learning**
- Problem formulation
- General recipe of meta-learning algorithms
- Black-box adaptation approaches
- **Case study of GPT-3 (time-permitting)**
Case Study: GPT-3

Language Models are Few-Shot Learners

Tom B. Brown*  Benjamin Mann*  Nick Ryder*  Melanie Subbiah*
Jared Kaplan†  Prafulla Dhariwal  Arvind Neelakantan  Pranav Shyam  Girish Sastry
Amanda Askell  Sandhini Agarwal  Ariel Herbert-Voss  Gretchen Krueger  Tom Henighan
Rewon Child  Aditya Ramesh  Daniel M. Ziegler  Jeffrey Wu  Clemens Winter
Christopher Hesse  Mark Chen  Eric Sigler  Mateusz Litwin  Scott Gray
Benjamin Chess  Jack Clark  Christopher Berner
Sam McCandlish  Alec Radford  Ilya Sutskever  Dario Amodei

OpenAI

May 2020
What is GPT-3?

a language model

black-box meta-learner trained on language generation tasks

\( \mathcal{D}_i^{tr} \): sequence of characters \( \mathcal{D}_i^{ts} \): the following sequence of characters


architecture: giant “Transformer” network \( 175 \) billion parameters, 96 layers, 3.2M batch size

What do different tasks correspond to?

spelling correction

simple math problems

translating between languages

a variety of other tasks

How can those tasks all be solved by a single architecture?
How can those tasks all be solved by a single architecture? Put them all in the form of text!

Why is that a good idea? Very easy to get a lot of meta-training data.

Learning via SGD during unsupervised pre-training

- **In-context learning**
  - sequence #1: simple math problems
    1. $5 + 8 = 13$
    2. $7 + 2 = 9$
    3. $1 + 0 = 1$
    4. $3 + 4 = 7$
    5. $5 + 9 = 14$
    6. $9 + 8 = 17$
  - sequence #2: spelling correction
    1. goat $=>$ goat
    2. sake $=>$ snake
    3. brid $=>$ bird
    4. fishe $=>$ fish
    5. duck $=>$ duck
    6. cmihp $=>$ chimp
  - sequence #3: translating between languages
    1. thanks $=>$ merci
    2. hello $=>$ bonjour
    3. mint $=>$ menthe
    4. wall $=>$ mur
    5. otter $=>$ loutre
    6. bread $=>$ pain
Some Results

One-shot learning from dictionary definitions:

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:

We screeged at each other for several minutes and then we went outside and ate ice cream.

Few-shot language editing:

Poor English input: I eat the purple berries.
Good English output: I ate the purple berries.
Poor English input: Thank you for picking me as your designer. I'd appreciate it.
Good English output: Thank you for choosing me as your designer. I appreciate it.
Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.
Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.
Poor English input: I'd be more than happy to work with you in another project.
Good English output: I'd be more than happy to work with you on another project.

Non-few-shot learning tasks:

Poor English input: Please provide me with a short brief of the design you’re looking for and that’d be nice if you could share some examples or project you did before.
Good English output: Please provide me with a brief description of the design you’re looking for and that would be nice if you could share some examples or projects you have done before.

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church’s annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist
General Notes & Takeaways

The results are extremely impressive.

The model is far from perfect.

The model fails in unintuitive ways.

Q: How many eyes does a giraffe have?
A: A giraffe has two eyes.

Q: How many eyes does my foot have?
A: Your foot has two eyes.

Q: How many eyes does a spider have?
A: A spider has eight eyes.

Q: How many eyes does the sun have?
A: The sun has one eye.

Source: https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

The choice of $\mathcal{D}^{tr}_i$ at test time is important. (“priming”)

Source: https://github.com/shreyashankar/gpt3-sandbox/blob/master/docs/priming.md
Plan for Today

*Transfer Learning*
- Problem formulation
- Fine-tuning

*Meta-Learning*
- Problem formulation
- General recipe of meta-learning algorithms
- Black-box adaptation approaches
- Case study of GPT-3 (time-permitting)

Goals for by the end of lecture:
- Differences between multi-task learning, transfer learning, and meta-learning problems
- Basics of transfer learning via fine-tuning
- Training set-up for few-shot meta-learning algorithms
- How to implement black-box meta-learning techniques
Reminders

Optional Homework 0 due tonight.

Homework 1 posted to be today, due Wednesday, October 6

Azure guide to be posted today

Next time: Optimization-based meta-learning