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

CS 330

Logistics

Homework 1 posted today, due **Wednesday, September 30**

Project guidelines will be posted by tomorrow.

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)

} Topic of Homework 1!

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?
(answer in chat or raise hand)

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

Parameters pre-trained on $\mathcal{D}_a$

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_b^{\text{tr}})$$

training data for new task $\mathcal{T}_b$

(typically for many gradient steps)

Pre-trained Dataset	PASCAL	SUN
Original (ImageNet)	58.3	52.2
Random	41.3 [21]	35.7 [2]

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

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

Pre-trained models often available online.

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 (Bayesian Optimization, Grid-Search etc.)
- Architecture choices matter (e.g. ResNets)

Universal Language Model Fine-Tuning for Text Classification. Howard, Ruder. ‘18

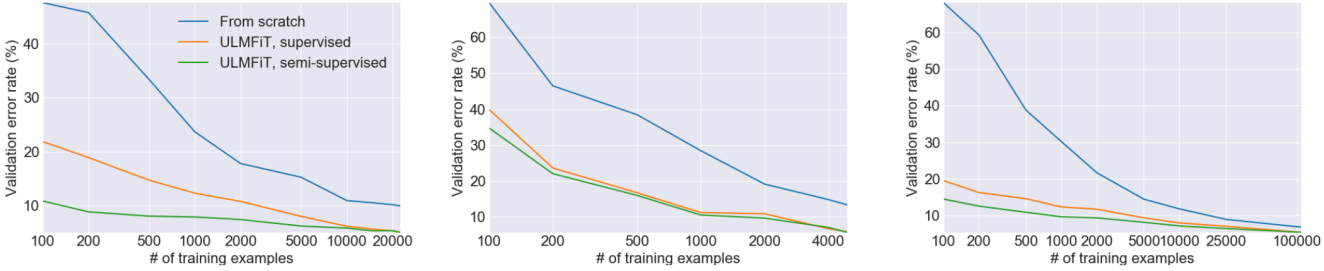


Figure 3: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).

Fine-tuning doesn’t work well with 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

Today: Focus primarily on the mechanistic view.

(Bayes will come back later)

How does meta-learning work? An example.

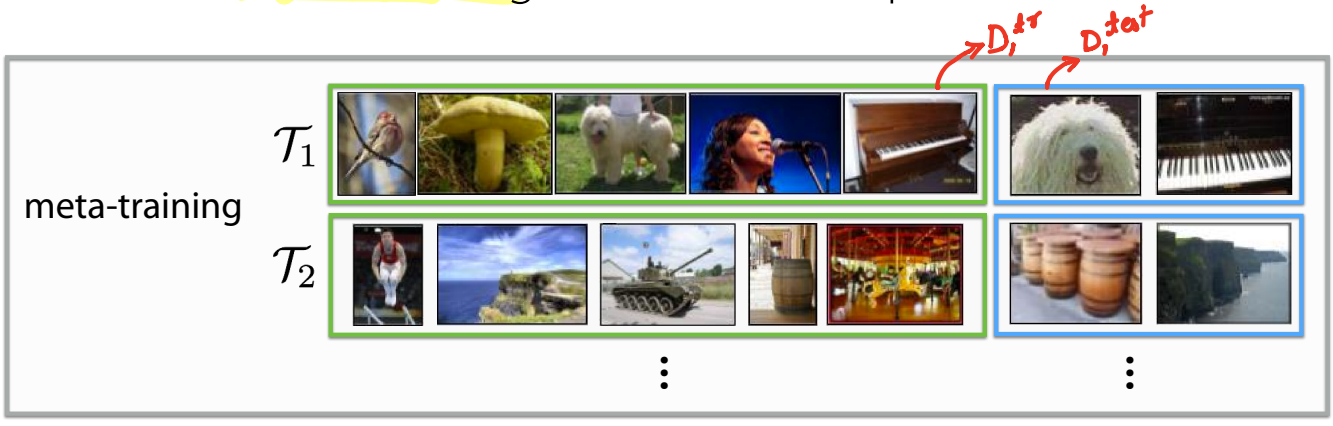
Given 1 example of 5 classes:

training data $\mathcal{D}_{\text{train}}$

Classify new examples

test set $\mathbf{x}_{\text{test}}$

How does meta-learning work? An example.



Given 1 example of 5 classes:

Classify new examples

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 $\mathcal{T}_{\text{test}}$

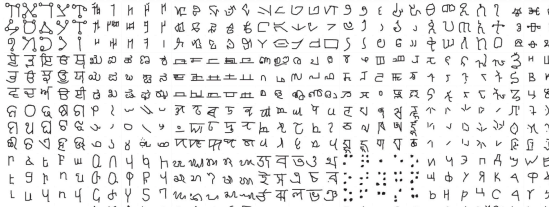
Key assumption: meta-training tasks and meta-test task drawn i.i.d. from same task distribution

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

Like before, tasks must share structure.

What do the tasks correspond to?

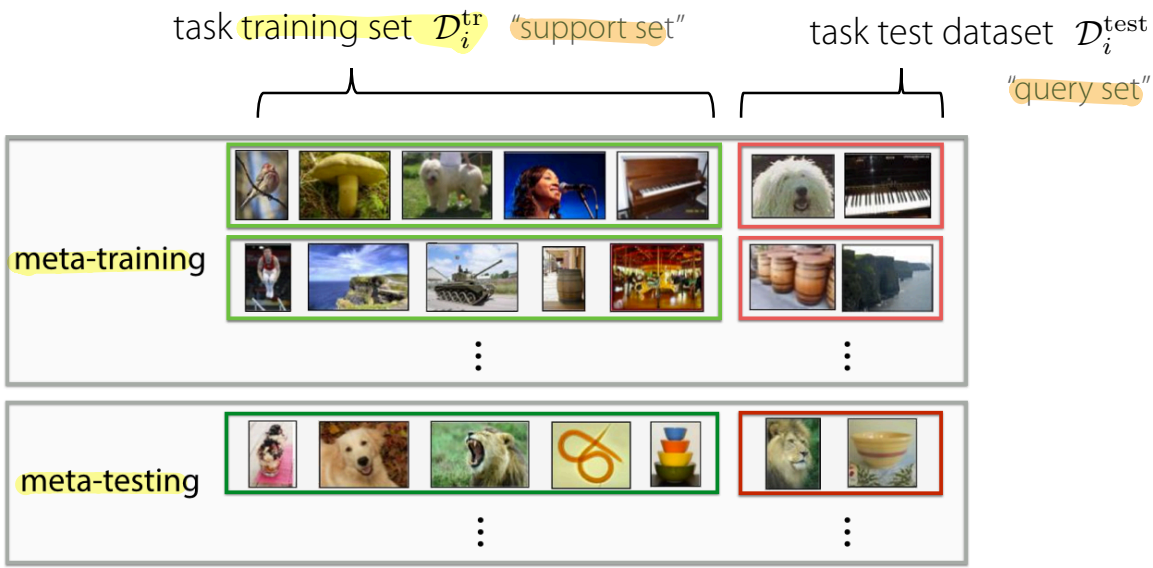
- recognizing handwritten digits from different languages (see homework 1!)
- spam filter for different users
- classifying species in different regions of the world
- a robot performing different tasks



How many tasks do you need?

The more the better. (analogous to more data in ML)

Some terminology



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? (answer in chat) K=1, N=5 each support set

Problem Settings Recap

contain 1 example from each class. Total 5 classes

Multi-Task Learning

Solve multiple tasks $\mathcal{T}_1, \cdots, \mathcal{T}_T$ at once.

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

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 $\mathcal{T}_{\text{test}}$

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

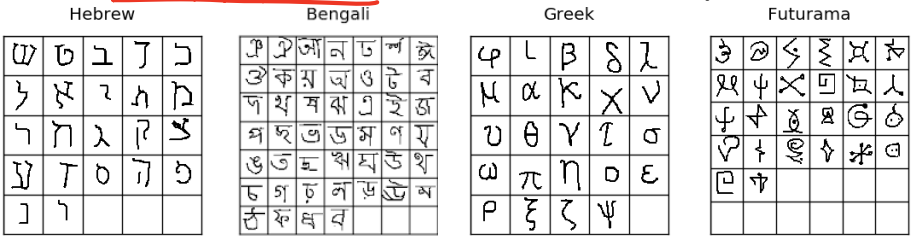
Meta-Learning
- Problem formulation
- **General recipe of meta-learning algorithms**
- Black-box adaptation approaches
- Case study of GPT-3 (time-permitting)

General recipe

How to *evaluate* a meta-learning algorithm

the Omniglot dataset Lake et al. Science 2015

1623 characters from 50 different alphabets



...

many classes, few examples

the "transpose" of MNIST

statistics more reflective of the real world

20 instances of each character

Proposes both few-shot discriminative & few-shot generative problems

Initial few-shot learning approaches w/ Bayesian models, non-parametrics

Fei-Fei et al. '03 Lake et al. '11 Salakhutdinov et al. '12 Lake et al. '13

Other datasets used for few-shot image recognition: tieredImageNet, CIFAR, CUB, CelebA, others

Other benchmarks: molecular property prediction (Ngyugen et al. '20), object pose prediction (Yin et al. ICLR '20)

Another View on the Meta-Learning Problem

Supervised Learning:

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

$$\mathbf{y} = f(\mathbf{x}; \theta)$$

Meta Supervised Learning:

Inputs: $\mathcal{D}^{\text{tr}}$ $\mathbf{x}_{\text{test}}$ Outputs: $\mathbf{y}_{\text{test}}$ Data: $\{\mathcal{D}_i\}$

$$\{(\mathbf{x}, \mathbf{y})_{1:K}\} \quad \mathbf{y}_{\text{test}} = h(\mathcal{D}^{\text{tr}}, \mathbf{x}_{\text{test}}; \theta)$$

$$\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$$

Why is this view useful?

Reduces the meta-learning problem to the design & optimization of h.

Finn. *Learning to Learn with Gradients*. PhD Thesis 2018

General recipe

How to *design* a meta-learning algorithm

1. Choose a form of $h(\mathcal{D}^{\text{tr}}, \mathbf{x}_{\text{test}}; \theta)$ "learner"

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

meta-parameters

Plan for Today

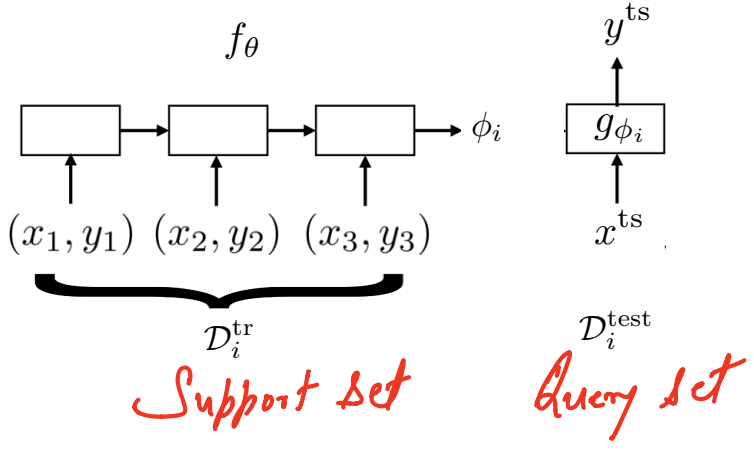
Transfer Learning
- Problem formulation
- Fine-tuning

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 $\phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$ "learner"

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



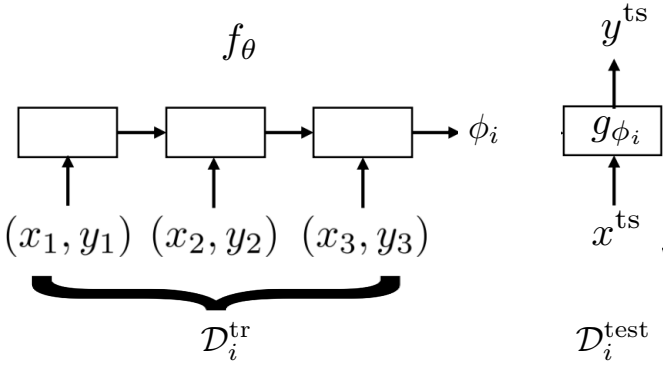
Train with standard supervised learning!

$$\max_\theta \sum_{\mathcal{T}_i} \underbrace{\sum_{(x,y)\sim\mathcal{D}_i^{\text{test}}} \log g_{\phi_i}(y|x)}_{\mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})}$$

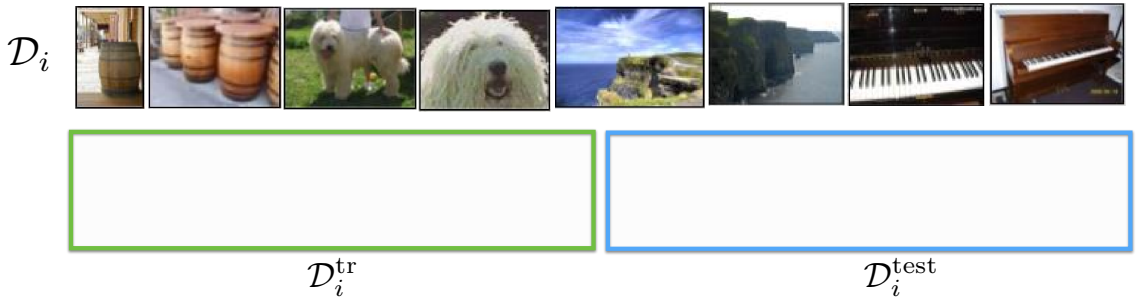
$$\max_\theta \sum_{\mathcal{T}_i} \mathcal{L}(f_\theta(\mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{test}})$$

Black-Box Adaptation

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

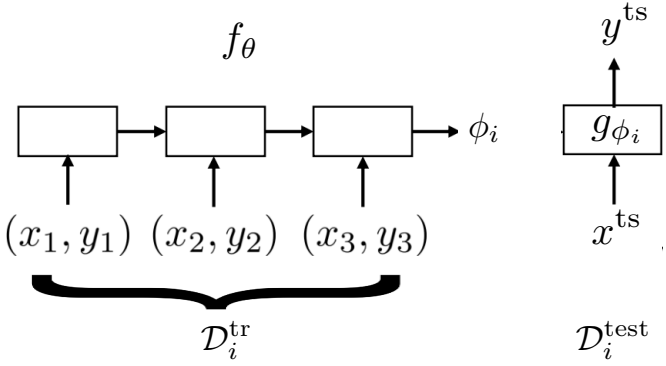


1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$

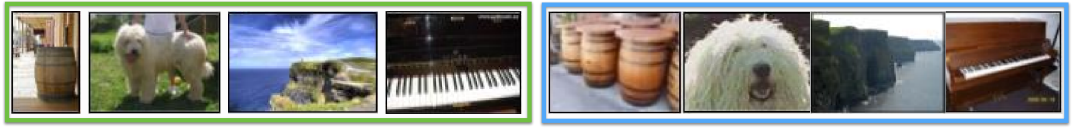


Black-Box Adaptation

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



1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Compute $\phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$



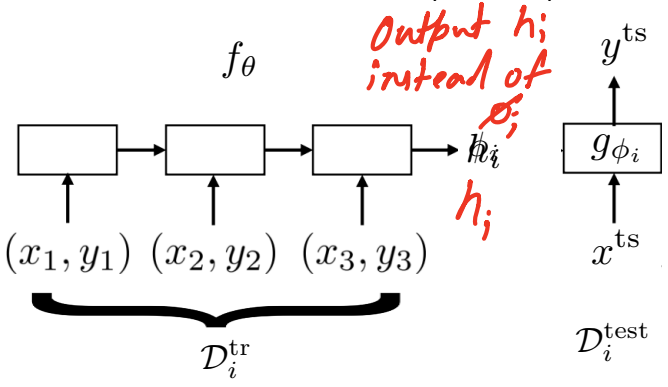
Black-Box Adaptation

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

Challenge

Outputting all neural net parameters does not seem scalable?

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



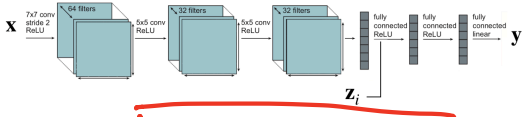
(Santoro et al. MANN, Mishra et al. SNAIL)

low-dimensional vector h_i

represents contextual task information

$$\phi_i = \{h_i, \theta_g\}$$

recall:

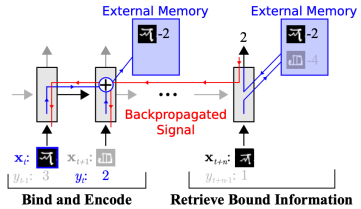


general form: $y^{\text{ts}} = f_\theta(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$

What architecture should we use for f_θ?

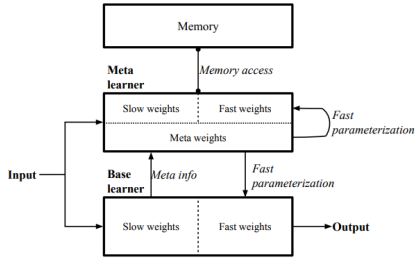
Black-Box Adaptation

LSTMs or Neural turing machine (NTM)



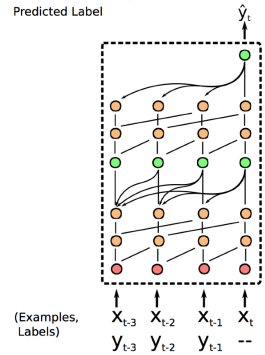
Meta-Learning with Memory-Augmented Neural Networks
Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML '16

Other external memory mechanisms



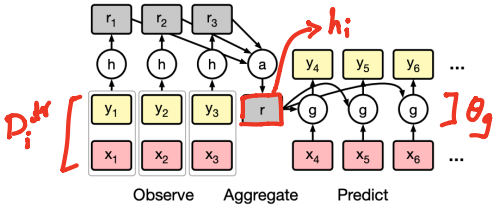
Meta Networks
Munkhdalai, Yu. ICML '17

Convolutions & attention

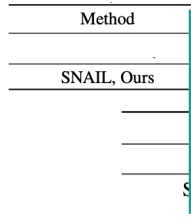


A Simple Neural Attentive Meta-Learner
Mishra, Rohaninejad, Chen, Abbeel. ICLR '18

Feedforward + average



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



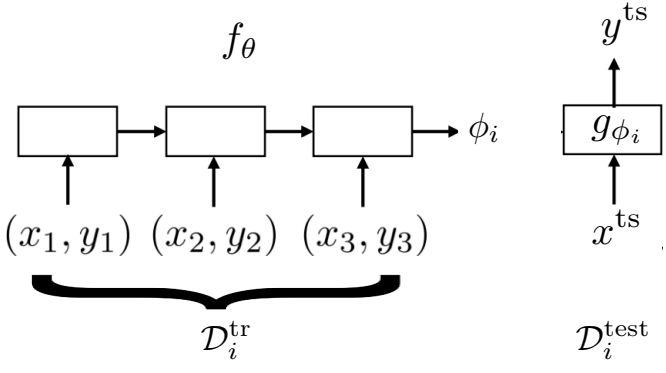
HW 1:
- implement data processing
- implement simple black-box meta-learner
- train few-shot Omniglot classifier

Question: Why might feedforward+average be better than a recurrent model? (raise your hand)

Because these data do not contain any temporal structure.

Black-Box Adaptation

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



+ **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 $\phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$?

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*

$\mathscr{D}_i^{\mathrm{tr}}$: sequence of characters $\mathscr{D}_i^{\mathrm{ts}}$: the following sequence of characters

[meta-training] dataset: crawled data from the internet, English-language Wikipedia, two books corpora

architecture: giant "Transformer" network 175 billion parameters, 96 layers, 3.2M batch size

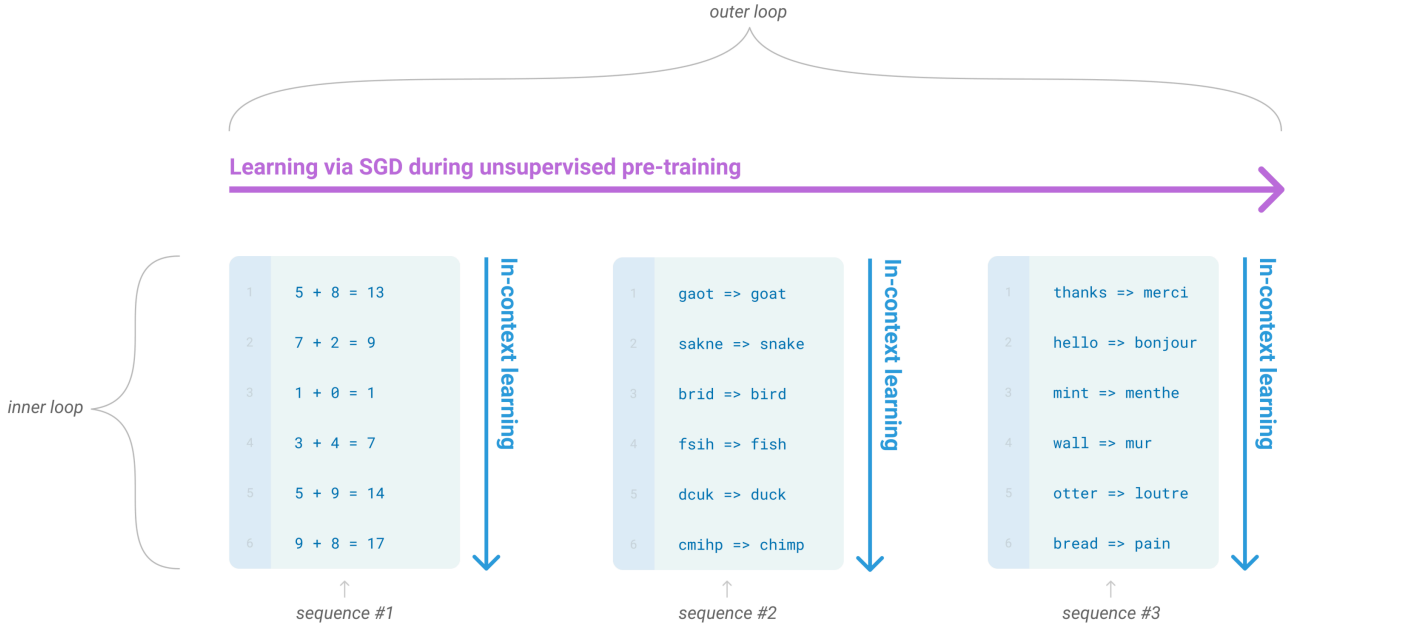
What do different tasks correspond to?

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.



simple math problems spelling correction translating between languages

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

Few-shot language editing:

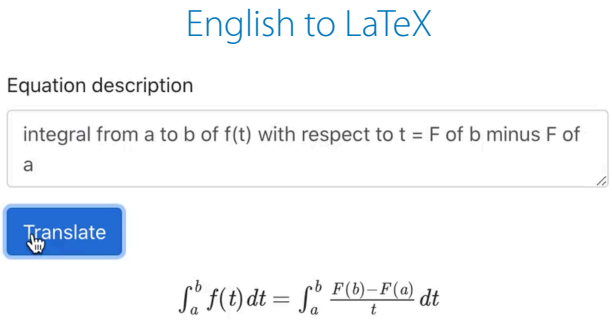
Poor English input: I eated the purple berries.
Good English output: I ate the purple berries.
Poor English input: Thank you for picking me as your designer. I'd appreciate it.
Good English output: Thank you for choosing me as your designer. I appreciate it.
Poor English input: The mentioned changes 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.

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.

Non-few-shot learning tasks:

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**

Other Cool Use-Cases

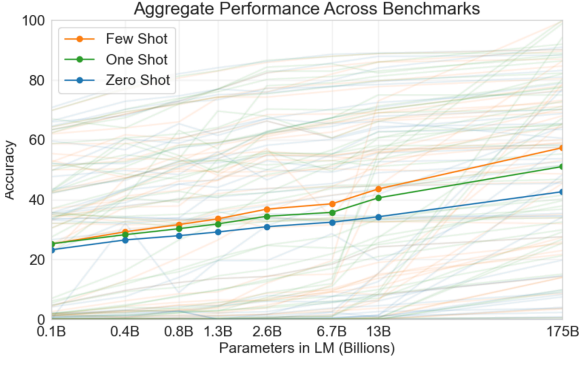


Source: https://twitter.com/sh_reya/status/1284746918959239168

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 $\mathscr{D}_i^{tr}$ at test time is important.

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)

} Topic of Homework 1!

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

Homework 1 posted today, due **Wednesday, September 30**

Project guidelines will be posted by tomorrow.

Next time: Optimization-based meta-learning