Passive and Active Multi-Task Representation Learning

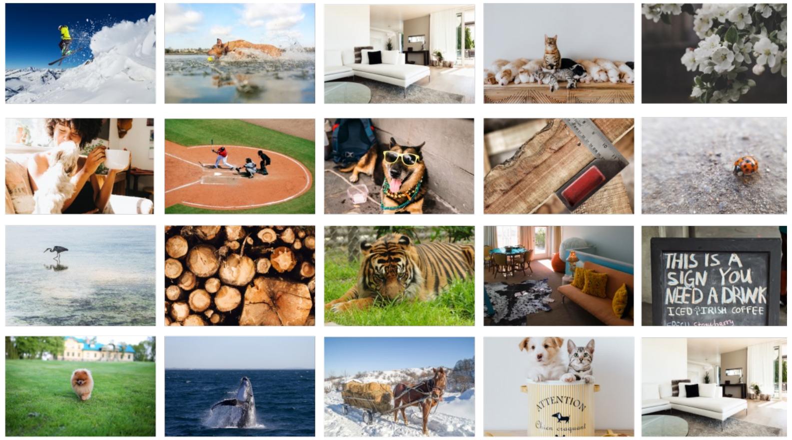
Maxine Salon

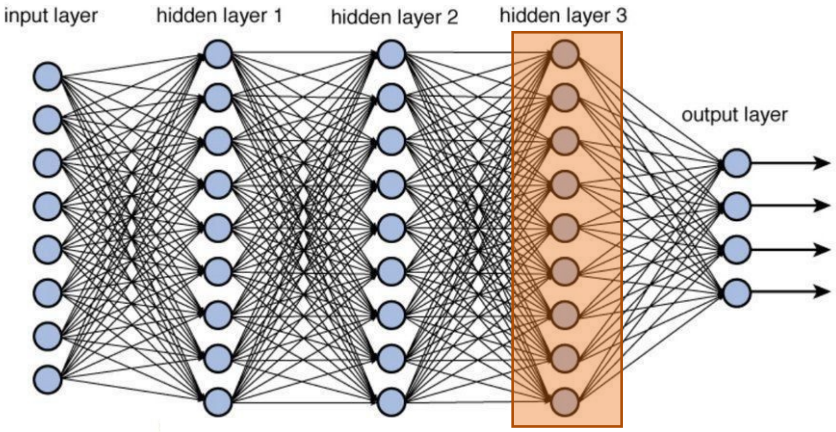
Yifang Chen

02-22-2022

Standard Paradigm in Representation Learning

Source tasks
(for training
representation):
ImageNet





ResNet

Target task:
Few-shot Learning
on VOC07 dataset
(20 classes, 1-8
examples per class)

- Without representation learning: **5% - 10%** (random guess = **5%)**
- With representation learning: **50% - 80%**

Talk Part I

For a good representation learning,

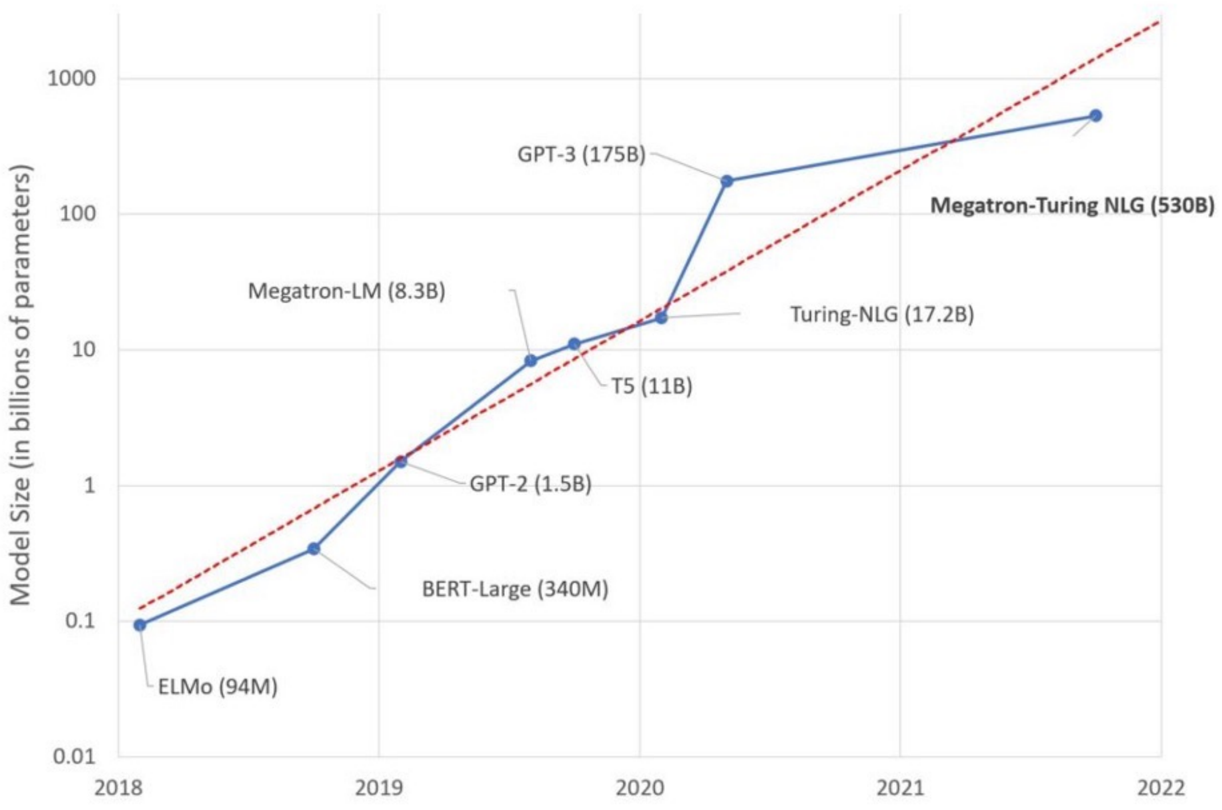
Q1: When?

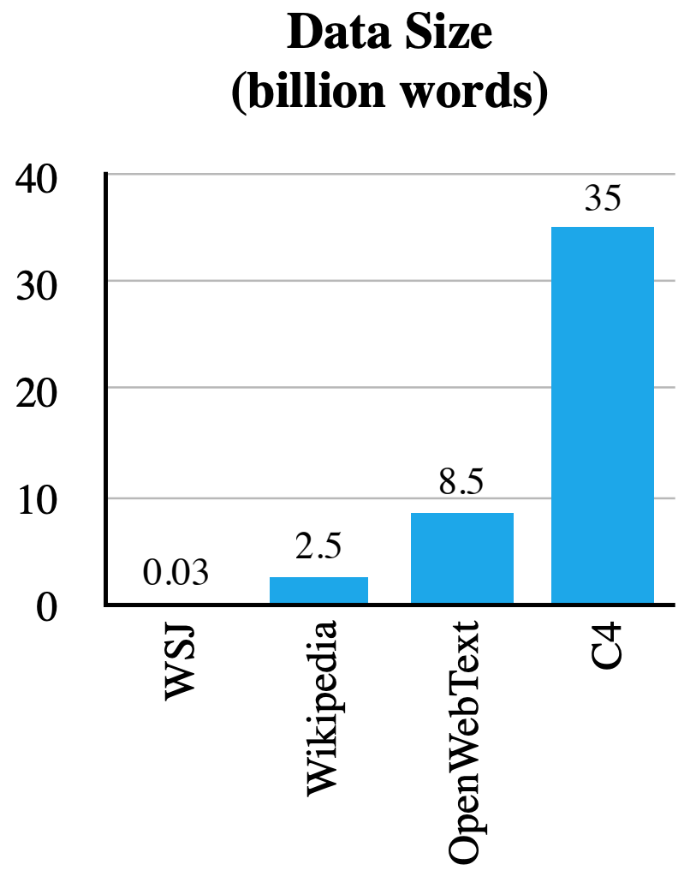
What are the necessary and sufficient conditions?

Q2: How?

What is the practical algorithm?

Big Model Trained on Big Data



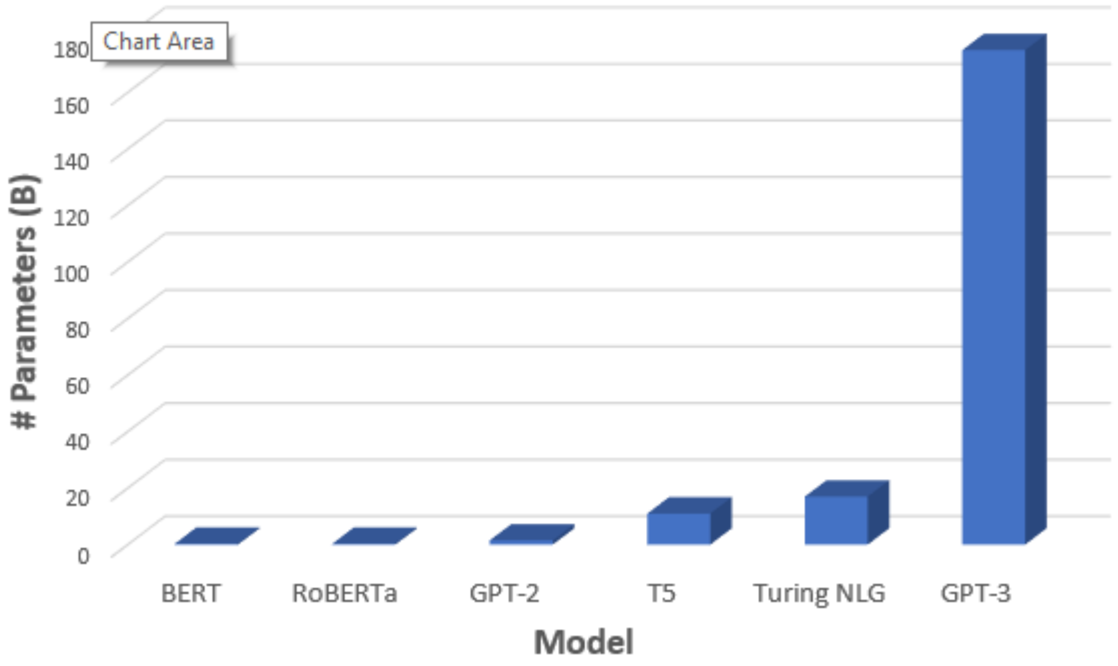


Pre-training data is cheap. Use as much as possible.
BUT…

Cost of Training Big Models

GPT-3:

- 175 Billion parameters
- 45TB data
- 10,000 GPUs
- Estimated cost ~$10M



Practical scenario:

- Limited resources: $, GPU, engineers.
- One or a few target downstream tasks.

Talk Part 2: pre-training data/task selection for representation learning

Motivations:

- Resources needed scale with # of pre-training data used.
- Data/task selection can improve performance [Chen Crammer He Roth Su 2021].

Approach: Active Learning

- Actively select training data instead of using all the data
- Classical active learning: single-task.
- Our work: Task level active learning.

Outline

Supervised Multi-Task Rep Learning

- What leads to good rep and transfer learning ?
- Theory results on classical setting
- Theory results on harder setting
  - High dim rep, overparameterized neural net
  - High task number, low data amount per task

Active Multi-Task Rep Learning

- When can we do better than passive learning ?
- Algorithm and experiment

Outline

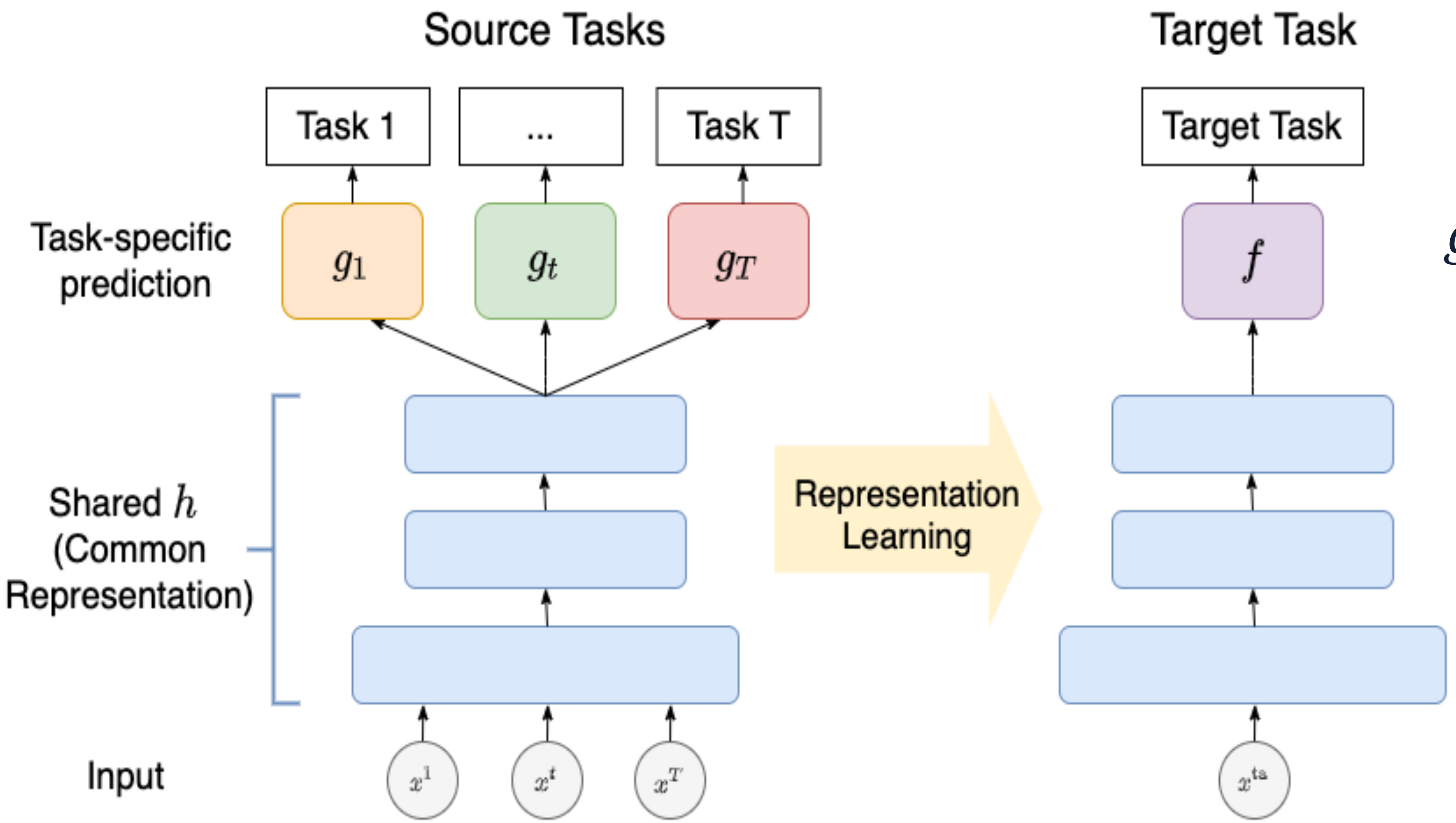
Supervised Multi-Task Rep Learning

- What leads to good rep and transfer learning ?
- Theory results on classical setting
- Theory results on harder setting
  - High dim rep, overparameterized neural net
  - High task number, low data amount per task

Active Multi-Task Rep Learning

- When can we do better than passive learning ?
- Algorithm and experiment

Supervised Multi-Task Representation Learning



$g_1, \dots g_T, f \in \mathcal{G}$: prediction class
(e.g., linear classifier)

$h \in \mathcal{H}$: representation class
(e.g., multi-layer NN)

Formulation

Representation Learning

- T source tasks, **each with $\boldsymbol{n_1}$** data:
  $\left\{\left(x_1^t, y_1^t\right) \ldots \left(x_{n_1}^t, y_{n_1}^t\right)\right\}_{t=1}^{T}$
  (uniform passive sampling)

- Learning representation:

$$\min_h \sum_{t=1}^{T} \min_{g_t} \sum_{i=1}^{n_1} \ell\left(g_t \circ h\left(x_i^t\right), y_i^t\right)$$

Predictor Learning

- 1 target task, with $\boldsymbol{n_{T+1} \ll n_1}$ data:
  $\left(x_1^{T+1}, y_1^{T+1}\right) \ldots \left(x_{n_2}^{T+1}, y_{n_2}^{T+1}\right) \sim \mu$

- Training for the target task:

$$\min_{f_{T+1}} \sum_{i=1}^{n_{T+1}} \ell\left(f_{T+1} \circ \boldsymbol{h}\left(x_i^{T+1}\right), y_i^{T+1}\right)$$

Representation $\boldsymbol{h}(\cdot)$ is fixed

Formulation

Representation Learning

- T source tasks, **each with n_1** data:
  $\left\{\left(x_1^t, y_1^t\right) \ldots \left(x_{n_1}^t, y_{n_1}^t\right)\right\}_{t=1}^{T}$
  (uniform passive sampling)

- Learning representation:
  $$\min_h \sum_{t=1}^{T} \min_{w_t} \sum_{i=1}^{n_1} \ell\left(\langle w_t, h\left(x_i^t\right)\rangle, y_i^t\right)$$

Predictor Learning

- 1 target task, with $n_{T+1} \ll n_1$ data:
  $\left(x_1^{T+1}, y_1^{T+1}\right) \ldots \left(x_{n_2}^{T+1}, y_{n_2}^{T+1}\right) \sim \mu$

- Training for the target task:
  $$\min_{w_{T+1}} \sum_{i=1}^{n_{T+1}} \ell\left(\langle w_{T+1}, h\left(x_i^{T+1}\right)\rangle, y_i^{T+1}\right)$$
  Representation $h(\cdot)$ is fixed

* In this lecture, we stick with the linear predictor w_t. The other choice of f can be, for example, monotonic Lipschitz function for multi-task index model. (See [T. Jordan Jin 2020b] for more examples on general choices of rep and predictor)

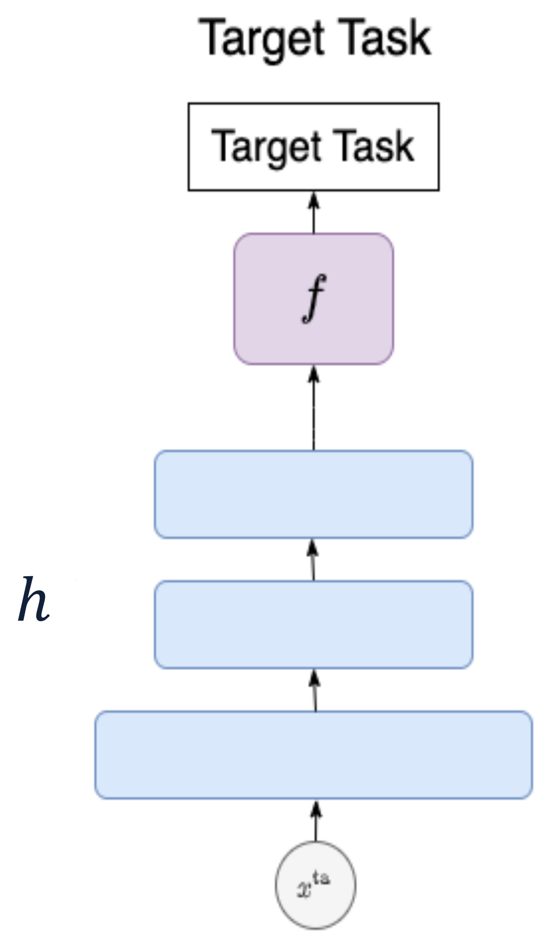
Standard Statistical Learning Theory

Training with data only from the target domain:

$$\min_{h\in\mathcal{H}, w_{T+1}\in\mathbb{R}^k} \sum_{i=1}^{n_{T+1}} \ell(\langle w_{T+1}, h(x_i^{T+1})\rangle, y_i^{T+1})$$

Theorem （Example）
Target task loss = $O(\frac{\mathcal{C}(\mathcal{H})+k}{n_{T+1}})$

$\mathcal{C}(\mathcal{H})$: complexity measure of the representation class $\mathcal{H}$. E.g., # of variables (linear function class), VC-dimension, Rademacher complexity, Gaussian width, etc

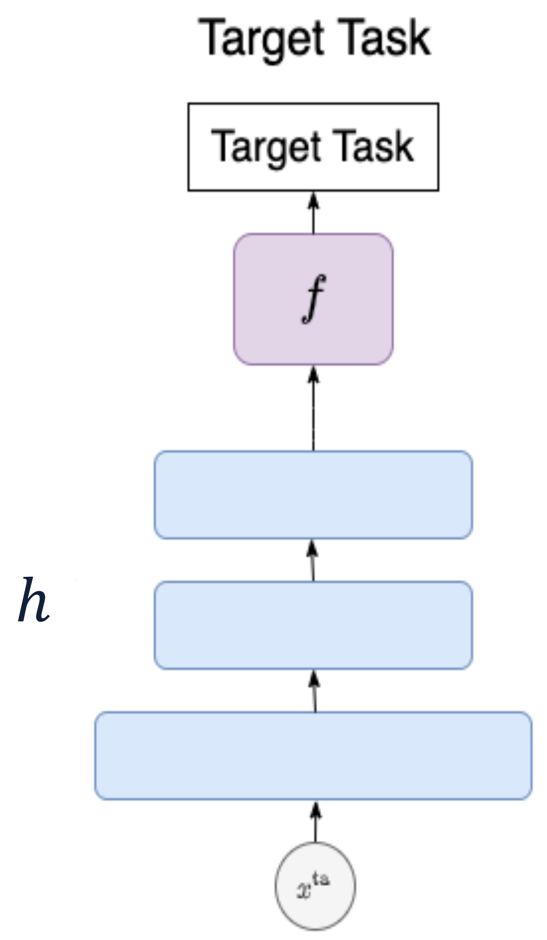


Standard Statistical Learning Theory

Theorem （Example）
Target task loss = $O(\frac{\mathcal{C}(\mathcal{H})+k}{n_{T+1}})$

In most cases, $\mathcal{C}(\mathcal{H}) \gg k$. E.g. $\mathcal{H}$ is a large neural network except the last layer.

Q: **Can we learn $\mathcal{H}$ from other tasks so n_{T+1} only need to scale with k ?**



Existence of a Good Representation

Assumption 1: Existence of a Good Representation
There exist a representation $\boldsymbol{h}^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$: $\mathbb{E}_{(x^t, y^t) \sim \mu_t}[\ell(\langle w_t^*, \boldsymbol{h}^*(x^t)\rangle, y^t)] = 0 \ \forall t = 1, \ldots, T$ $\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, \boldsymbol{h}^*(x_{T+1})\rangle, y_{T+1})] = 0$

A **shared** good representation for all source tasks and the target task:

This is why we use representation learning.

(Without this assumption, we should not use representation learning)

Existence of Good Rep is NOT Enough

Input: 1000 dimensional 0/1 vector, $\{0,1\}^{1000}$

Good representation: first 100 dimension
- All tasks (source and target) only need first 100 digits for accurate prediction.
- Predicting whether the 10^{th}-digit is 1, predicting the sum of first 100 digits, etc.

"Worst-case" target task

Bad scenario:
- Source tasks only need to use first 50 digits: e.g., whether the 10^{th}-digit is 1
- Target tasks need to use **all** first 100 digits: e.g., predicts the sum of first 100 digits

Source tasks cannot give the **full information** about the good representation!

Assumption 2: Diversity of Source Tasks

Representation learning is useful only if source tasks can give the full information about the good representation, a.k.a., **diversity of the source tasks.**

Q: **What is the definition of diversity?**

Diversity for Linear Predictors

Assumption 1: Existence of a Good Representation

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x^t, y^t) \sim \mu_t}[\ell(\langle w_t^*, h^*(x^t)\rangle, y^t)] = 0 \; \forall t = 1, \ldots, T$$

$$\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, h^*(x_{T+1})\rangle, y_{T+1})] = 0$$

Assumption 2: Diversity of Source Tasks for Linear Predictor

$W^* = [w_1^*, w_2^*, \ldots, w_T^*] \in \mathbb{R}^{k \times T}$ is full rank (=k).

Need $T \geq k$: cover the **span** of the good representation.

Also see [Tripuraneni Jordan Jin 2020]

Linear Representation (Subspace Learning)

Input: $x \in \mathbb{R}^d$. Linear representation class $\mathcal{H}$: matrices of size $k \times d$ $(k \ll d)$.

Assumption 1: Existence of a Good Representation

There exists a linear representation $B^* \in \mathbb{R}^{k \times d}$, and $w_1^*, w_2^*, \dots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x^t, x^t) \sim \mu_t}[\ell(\langle w_t^*, B^* x^t \rangle, y^t)] = 0 \ \forall t = 1, \dots, T$$

$$\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, B^* x_{T+1} \rangle, y_{ta})] = 0$$

Theorem [Du Hu Kakade Lee Lei, 2020]

Under Assmp. 1 &2, we have the target task loss $= O\left(\frac{dk + Tk}{n_1 \sigma_{min}^2(W^*)} + \frac{k}{n_{T+1}}\right)$.

When source tasks are uniformly spread, $\boldsymbol{\sigma_{min}(W^*) = \Theta(\sqrt{T/k})}$.

Without representation learning, directly learning a linear predictor on $\mathbb{R}^d$: $O(\frac{d}{n_{T+1}})$.

Main Result for General Representation Class

Assumption 1: Existence of a Good Representation

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \dots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x^t, y^t) \sim \mu_t}[\ell(\langle w_t^*, h^*(x^t)\rangle, y^t)] = 0 \ \forall t = 1, \dots, T$$

$$\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, h^*(x_{T+1})\rangle, y_{T+1})] = 0$$

Theorem [Du Hu Kakade Lee Lei, 2020]

Under Assmp. 1 &2, we have the target task loss $= O\left(\frac{\mathcal{C}\left(\mathcal{H}, \{x_i^t\}_{i,t}\right)^2}{\boldsymbol{n_1 \sigma_{min}^2(W^*)}} + \frac{k}{n_{T+1}}\right)$.

$\mathcal{C}\left(\mathcal{H}, \{x_i^t\}_{i,t}\right)$: Gaussian width of the representation class $\mathcal{H}$ projected on all the input data.

- Measures how well the function in the class can fit the noise.
- Can use existing theory for neural networks for $\mathcal{C}(\mathcal{H}, \cdot)$.

Key Message

Existence of a good representation and **diversity of tasks** are key conditions that enable **representation learning** to improve sample efficiency.

Beyond the standard results

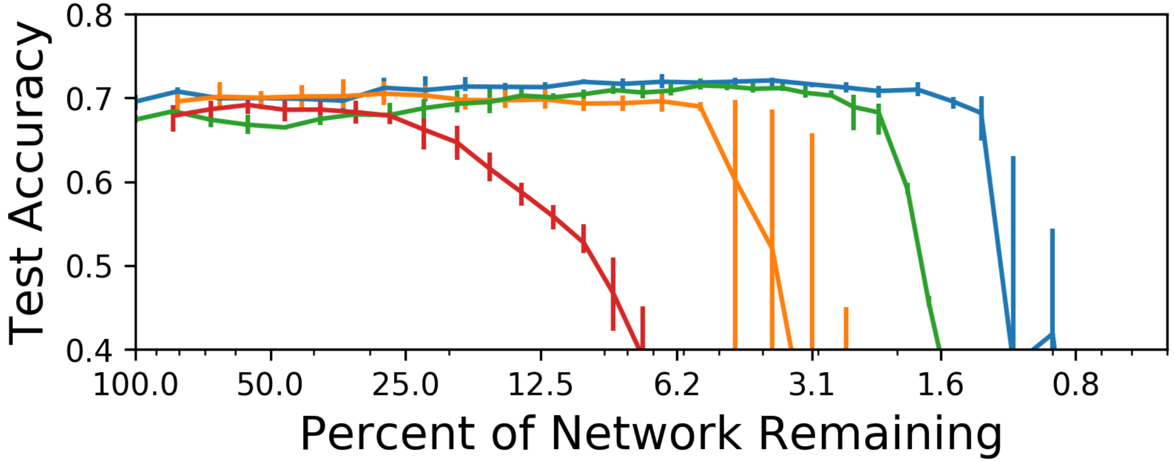
The current results we presented here has two intrinsic assumptions:

1. The exact dimension/complexity of the representation space $\{h(x) | \forall\, h \in \mathcal{H}\}$ is known to the learner. e.g., $\phi(x) = x^\top B$ where $B \in \mathbb{R}^{k \times d}$, k is known to the learner.

Q: **Can we achieve good guarantees when the exact low dim of $\phi(x)$ is unknown ?**

Example

Test acc after prune neural net: each curve corresponding to different architecture or pruning methods



- Neural net is inherently sparse or has intrinsic low rank

- But usually, we don’t have prior knowledge on this low rank. Complicated pruning methods are needed to learn the true underlying low dim subspace.

Main result for implicit low dim representation

Assumption 1: high dimension linear representation

There exist a good representation $\phi^*(x) = B^* x$ where $B^* \in \mathbb{R}^{T \times d}$, and $w_1^*, w_2^*, \dots, w_T^*, w_{T+1}^* \in \mathbb{R}^T$.

But B^* has intrinsic unknown low intrinsic rank $R = \frac{||\Theta^*||_*}{||\Theta^*||_F}$,

where $\Theta^* = (B^*)^\top [w_1^*, w_2^*, \dots, w_T^*]$

Add regularization term to ERM

$$\hat{B} = \underset{B}{\mathrm{argmin}} \sum_{t=1}^{T} \min_{w_t} \sum_{i=1}^{n_1} \ell(\langle B x_i^t, w_t \rangle, y_i^t) + \lambda ||B|| + \lambda \sum_{t=1}^{T} ||w_t||$$

$$\widehat{w_{T+1}} = \underset{||w|| \leq \frac{\sqrt{||\Theta^*||_*}}{T}}{\mathrm{argmin}} \sum_{i=1}^{n_{T+1}} \ell(\langle B x_i^{T+1}, w \rangle, y_i^{T+1})$$

Main result for implicit low dim representation

Assumption 1: high dimension linear representation

There exist a good representation $\phi^*(x) = B^*x$ where $\mathrm{B}^* \in \mathbb{R}^{T\times d}$, and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^T$.

But B^* has intrinsic unknown low intrinsic rank $R = \frac{||\Theta^*||_*}{||\Theta^*||_F}$,

where $\Theta^* = (B^*)^\top [w_1^*, w_2^*, \ldots, w_T^*]$

Theorem [Du Hu Kakade Lee Lei, 2020] (Informal)

Under Assmp. 1, in a common case $\mathrm{T} = \mathrm{d}, w_{T+1} \sim \mathcal{N}(0, \Theta^*(\Theta^*)^\top/T)$,

we have the target task loss $= O\left(\frac{R||\Theta^*||_F}{\sqrt{T}}\sqrt{\frac{d}{n_1 T}} + \frac{R||\Theta^*||_F}{\sqrt{T}}\sqrt{\frac{1}{n_{T+1}}})\right)$,

Without regularization on ERM, last term will scale with T=d : $O\left(\frac{d}{n_{T+1}}\right)$.

Main result for implicit low dim representation

Assumption 1: Overparameterized 2-layer neural network

There exist a good representation $\phi^*(x) = \max(B^* x, 0)$ $(relu)$ where $B^* \in \mathbb{R}^{d \times d_0}$, and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^d$.

But B^* has intrinsic unknown low intrinsic rank $R = ||W^*||_F^2 + ||B^*||_F^2$,

Solution: Add regularization term to ERM

$$\hat{B} = \underset{B}{\operatorname{argmin}} \sum_{t=1}^{T} \min_{w_t} \sum_{i=1}^{n_1} \ell(\langle \max(Bx_i^t, 0), w_t \rangle, y_i^t) + \lambda ||B||_F + \lambda \sum_{t=1}^{T} ||w_t||$$

$$\widehat{w_{T+1}} = \operatorname{argmin}_{w \in some\ regu\ \text{constraints}} \sum_{i=1}^{n_{T+1}} \ell(\langle Bx_i^{T+1}, w \rangle, y_i^{T+1})$$

Main result for implicit low dim representation

Assumption 1: Overparameterized 2-layer neural network

There exist a good representation $\phi^*(x) = \max(B^*x, 0)$ $(relu)$ where $\mathrm{B}^* \in \mathbb{R}^{d \times d_0}$, and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^d$.

But B^* has intrinsic unknown low intrinsic rank $R' = ||W^*||_F^2 + ||B^*||_F^2$,

.

Theorem [Du Hu Kakade Lee Lei, 2020] (Informal)

Under Assmp. 1, when w_{T+1} is in some benign setting (skip here), we have the target

$$\text{task loss} = O\left(\frac{R'}{\sqrt{T}}\sqrt{\frac{d}{n_1 T}} + \frac{R'}{\sqrt{T}}\sqrt{\frac{1}{n_{T+1}}})\right),$$

Beyond the standard results

The current results we presented here has two intrinsic assumptions:

2. The number of task is not huge $T \leq O(d)$ and each task collects a proper amount of data $n_1 \geq \Omega(d)$.

Q:

Can we achieve good guarantees when we have huge number of tasks, but each task has very limited data ?

Example

- Suppose there exists T diverse tasks each has $dlog(T)$ number of samples and satisfies the diverse requirement.

- Now uniformly divide each task into to d subtasks, shuffle all the subtask and present to the leaner. So, the learner saw dT sub-tasks, but not know which are belong to the same task.

- With the exact same data, the test loss for target on learning these sub-tasks should be same as learning directly on T tasks.

- But by using naïve ERM $\min_h \sum_{t=1}^{Td} \min_{w_t} \sum_{i=1}^{n_1} \ell(\langle w_t, h(x_i^t)\rangle, y_i^t)$, the learner will have worse guarantees

Main result for a large number source tasks

Assumption 1: Existence of a Good Representation

There exists a linear representation $B^* \in \mathbb{R}^{k\times d}$, and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$.

Assumption 2: small number of sample per source task

There exist a large number of source tasks $\mathrm{T} \geq d$, but each source task is only guaranteed to provide $n_1 \geq \Omega(\log(T))$ amount of data.

Solution: Alternatively minimize $\widehat{w}_t$ and $\hat{B}$.

- Random shuffle the task and iteratively training on each task
- In each iteration,
  - first fix the current $\hat{B}$ and minimize on $\widehat{w}_t$
  - then fix the current $\widehat{w}_t$ and minimize on $\hat{B}$

Main result for a large number source tasks

Assumption 1: Existence of a Good Representation

There exists a linear representation $B^* \in \mathbb{R}^{k\times d}$, and $w_1^*, w_2^*, \dots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$.

Assumption 2: small number of sample per source task

There exist a large number of source tasks $\mathrm{T} \geq d$, but each source task is only guaranteed to provide $n_1 \geq \Omega(\log(T))$ amount of data.

Theorem [Thekumparampil Jain Netrapalli Oh, 2020] (Informal)

Under Assmp. 1 and 2, we have the target task loss $= O\left(\frac{dk}{n_1 \sigma_{min}^2(W^*)} + \frac{k}{n_{T+1}}\right)$,

If directly using ERM, we will have an extra $O\left(\frac{Tk}{n_1 \sigma_{min}^2(W^*)}\right)$ term and the guarantees can even be impossible when $n_1 \geq \mathrm{O}(d)$

Key Message

Replace bi-level ERM oracle with more advanced methods (e.g., add regularizer, use alternative minimization) gives multi-task rep learning more robustness and adaptivity

Outline

Supervised Multi-Task Rep Learning

- What leads to good rep and transfer learning ?
- Results on benign setting
- Results beyond benign setting
  - High dim rep, overparameterized neural net
  - High task number, low data amount per task

Active Multi-Task Rep Learning

- When can we do better than passive learning ?
- Algorithm and experiment

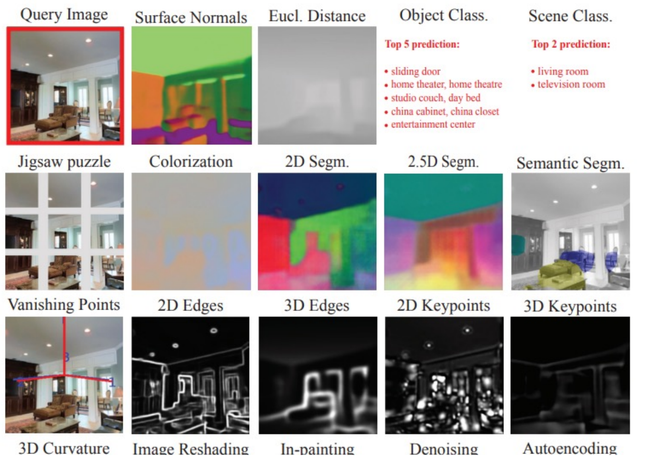
Limitation for passive learning

Passive learning : Train on all available source tasks. Usually, tasks are uniformly collected from real-world environment.

Limitation:

- There exists a large number of tasks (different domain, different metric)
- Processing data can be expensive
- Not all the rep feature are useful for target task

CV:



NLP:

Task	Domain	Metric
natural language inference (NLI)	various	accuracy
sentiment analysis	Movie Reviews	accuracy
paraphrase detection	social QA questions (Quora)	accuracy & F1
NLI	Wikipedia	accuracy
extractive QA	Wikipedia	F1 & EM
cloze-style QA	news (CNN, Daily Mail)	F1 & EM

Task Relevance

Active learning goal: select the most relevant source tasks for the target task.

Q: **How to characterize the relevance?**

Example

Input: 1000 dimensional 0/1 vector, $\{0,1\}^{1000}$

Larger than necessary.

Good representation: first 100 dimension

Bad scenario:

- Source tasks only need to use first 50 digits: e.g., whether the 10^{th}-digit is 1
- Target tasks need to use all first 100 digits: e.g., predicts the sum of first 100 digits

OK scenario:

- Source tasks only need to use first 50 digits: e.g., whether the 10^{th}-digit is 1
- The target task **also only uses the first** 50 digits: e.g., predicts the sum of the first 50 digits.

Which scenario you are in ? (hard to know in advance in practice)

Task Relevance Definition

Assumption 1: Existence of a Good Representation

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x^t, y^t) \sim \mu_t}[\ell(\langle w_t^*, h^*(x^t)\rangle, y^t)] = 0 \ \forall t = 1, \ldots, T$$

$$\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, h^*(x_{T+1})\rangle, y_{T+1})] = 0$$

Assumption 2: Task Relevance

$w_{T+1}^* \in$ Span (W^*) where $W^* = [w_1^*, \ldots, w_T^*] \in \mathbb{R}^{k \times T}$

Definition: $\boldsymbol{\nu}^* = \mathbf{argmin}_{\boldsymbol{\nu} \in \mathbb{R}^T} ||\boldsymbol{\nu}||_2$

s.t. $\boldsymbol{w}_{T+1}^* = \boldsymbol{W}^* \boldsymbol{\nu}$

- Minimize norm in order to have a unique ν^*.
- Assume $||w_t^*||_2 = 1$ for normalization. Then $\frac{1}{T} \leq ||\nu^*||_2^2 \leq 1/\sigma_{\min}^2(W^*)$.
- $\nu^* = [1,0,0, \ldots]$: one source task equals to the target task, others are orthogonal.

Recall Linear Representation (Subspace Learning)

Input: $x \in \mathbb{R}^d$. Linear representation class $\mathcal{H}$: matrices of size $k \times d$ $(k \ll d)$.

Assumption 1: Existence of a Good Representation

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \ldots, w_T^*, w_{T+1}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x^t, y^t) \sim \mu_t}[\ell(\langle w_t^*, h^*(x^t)\rangle, y^t)] = 0 \ \forall t = 1, \ldots, T$$

$$\mathbb{E}_{(x^{T+1}, y^{T+1}) \sim \mu}[\ell(\langle w_{T+1}^*, h^*(x_{T+1})\rangle, y_{T+1})] = 0$$

Theorem [Du Hu Kakade Lee Lei, 2020]

Under Assumption 1 &2, when using passive learning,

we have the target task loss $= O(\frac{dk\ ||\boldsymbol{\nu}^*||_2^2}{\boldsymbol{n_1}} + \frac{k}{n_{T+1}})$.

Algorithm with Known ν^*

Representation Learning

- Total budget: $n_1 T$ data.
- Sample $\boldsymbol{n_t} \propto (\boldsymbol{\nu_t^*})^2$ from the t-th task: $\left\{(x_1^t, y_1^t) \ldots (x_{n_i}^t, y_{n_i}^t)\right\}_{t=1}^T$
- Learning representation:

$$\min_h \sum_{t=1}^{T} \min_{w_t} \sum_{i=1}^{\boldsymbol{n_t}} \ell(\langle w_t, h(x_i^t)\rangle, y_i^t)$$

ℓ: quadratic loss

Predictor Learning

- 1 target task, with $n_{T+1} \ll n_1$ data: $(x_1^{T+1}, y_1^{T+1}) \ldots (x_{n_{T+1}}^{T+1}, y_{n_{T+1}}^{T+1}) \sim \mu$
- Training for the target task:

$$\min_{w_{T+1}} \sum_{i=1}^{n_{T+1}} \ell(\langle w_{T+1}, h(x_i^{T+1})\rangle, y_i^{T+1})$$

Representation $\boldsymbol{h}(\cdot)$ is fixed

Theoretical Result with Known ν^*

Theorem [C. Du Jamieson, 2022]

If we sample $\boldsymbol{n_t \propto (\nu_t^*)^2}$ from the t-th task with total budget $n_1 T$, we have

$$\text{the target task loss} = O\left(\frac{dk\, \boldsymbol{s^*||\nu^*||^2}}{\boldsymbol{n_1 T}} + \frac{k}{n_{T+1}}\right),$$

where $s^* = \min_{\gamma\in[0,1]} (1-\gamma)||\nu^*||_{0,\gamma} + \gamma T$ and $||\nu^*||_{0,\gamma} = |\{|\nu_t^* \geq \sqrt{\frac{\gamma}{N_1 T}}|\}|$.

s^*: approximate sparsity. $1 \leq s^* \leq T$

- Passive uniform sampling: $O(\frac{dk\, ||\boldsymbol{\nu^*}||_2^2}{\boldsymbol{n_1}} + \frac{k}{n_{T+1}})$.
- Bound never worse than passive sampling.

Example: one source task equals target task, but others are orthogonal:

- $s^* = 1, \nu^* = 1 \ \Rightarrow \frac{1}{T}$ improvement over passive sampling
- Intuition: should just sample from this particular source task!

Algorithm with Unknown ν^*

Main ideas: 1) estimate ν^* iteratively, 2) doubling schedule.

- Initialize $\hat{\nu}_t = 1$ for t= 1,…,T.
- For j =1, 2,…
  - Sample $\boldsymbol{n_t} \propto (\hat{\boldsymbol{\nu}}_t)^2\, \boldsymbol{2^j}$ from the t-th task: $\left\{\left(x_1^t, y_1^t\right) \ldots \left(x_{n_1}^t, y_{\boldsymbol{n_t}}^t\right)\right\}_{t=1}^{T}$
  - Learn representation:

$$\hat{h}, \widehat{W} = \operatorname{argmin}_h \sum_{t=1}^{T} \underset{w_t}{\operatorname{argmin}} \sum_{i=1}^{\boldsymbol{n_t}} \ell\left(\langle w_t, h\left(x_i^t\right)\rangle, y_i^t\right)$$

  - Learn the target task:

$$\widehat{w}_{T+1} = \operatorname{argmin}_{w_{T+1}} \sum_{i=1}^{n_{T+1}} \ell\left(\langle w_{T+1}, \hat{h}\left(x_i^{T+1}\right)\rangle, y_i^{T+1}\right)$$

  - Estimate task relevance: $\hat{\nu} = \operatorname{argmin}_\nu \|\nu\|_2$ s.t. $\widehat{W}\nu = \widehat{w}_{T+1}$

Theoretical Result with Known ν^*

Theorem [C. Du Jamieson, 2022]

With total budget $n_1 T$, we have

$$\text{the target task loss} = O\left(\frac{dk\, \boldsymbol{s^*||\nu^*||^2}}{\boldsymbol{n_1 T}} + \frac{k}{n_{T+1}} + \text{lower order terms}\right)$$

where $s^* = \min_{\gamma \in [0,1]} (1-\gamma)||\nu^*||_{0,\gamma} + \gamma T$ and $||\nu^*||_{0,\gamma} = |\{|v_t^2 \geq \frac{\gamma}{N_1 T}|\}|$.

Lower order terms account for estimating ν^*.

Experiments

Dataset: MNIST-C(orruption)

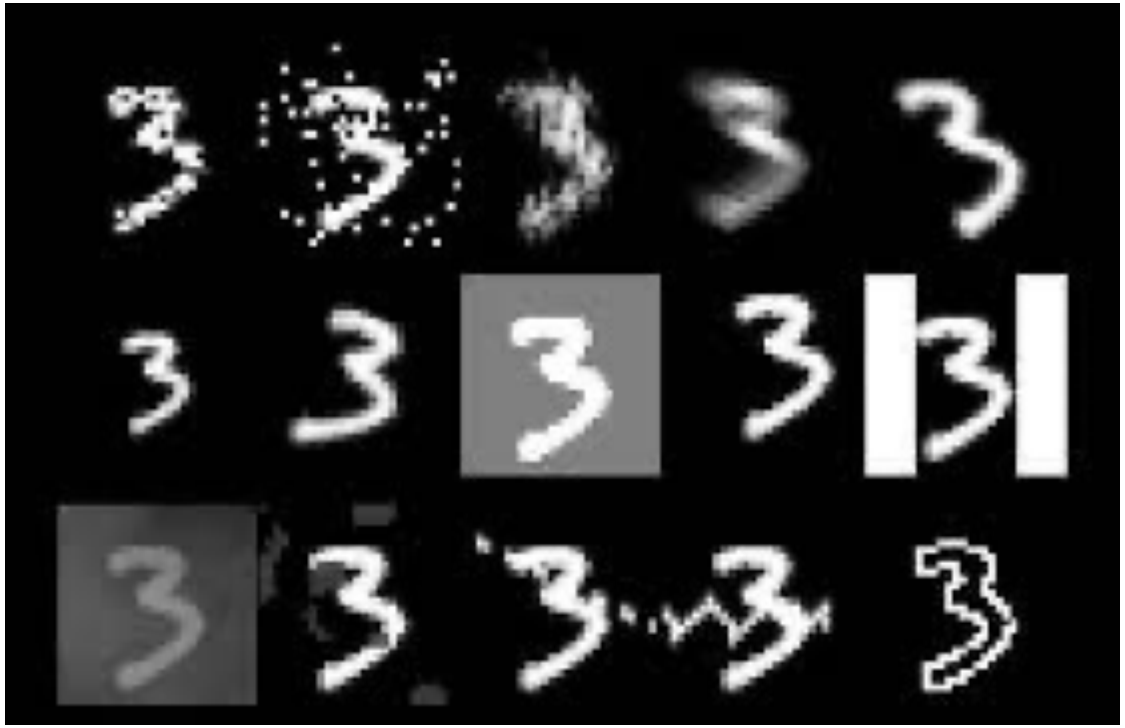
- 16 types of corruptions

Multi-task formulation:

- 10 digits x 16 types of corruptions = 160 binary tasks
- Each target task has 150 source tasks (10 digits x 15 other types of corruptions)

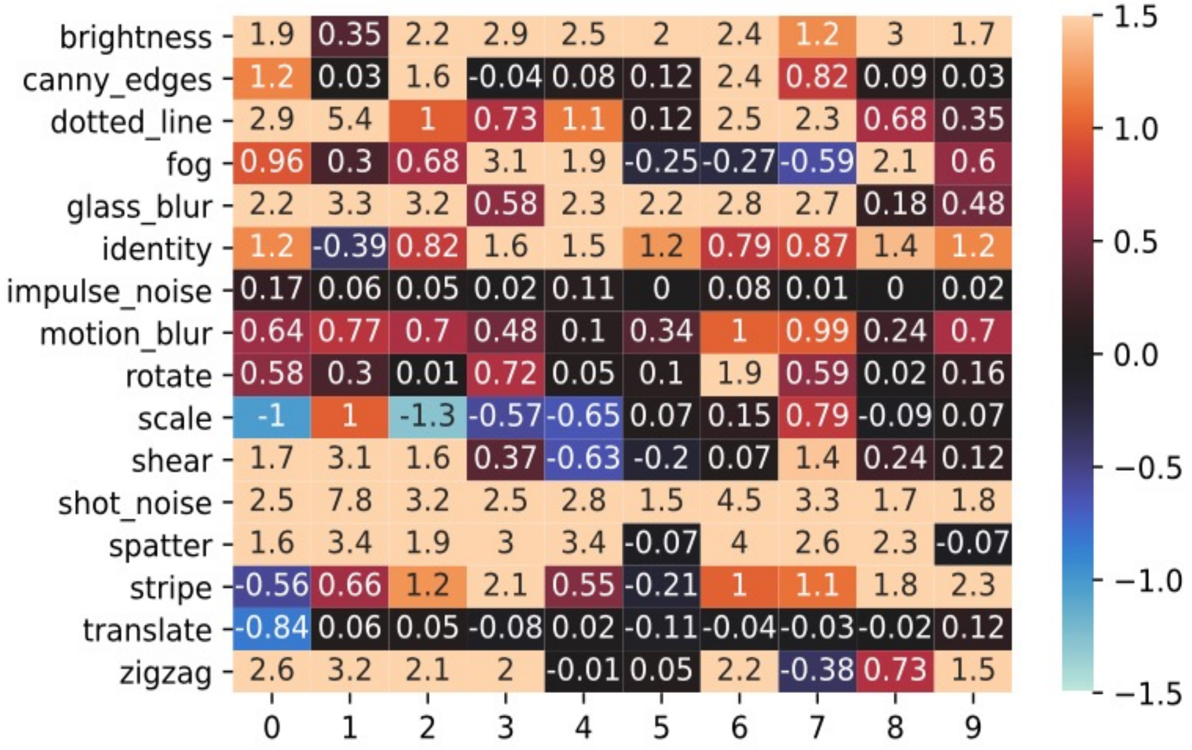
Representation function:

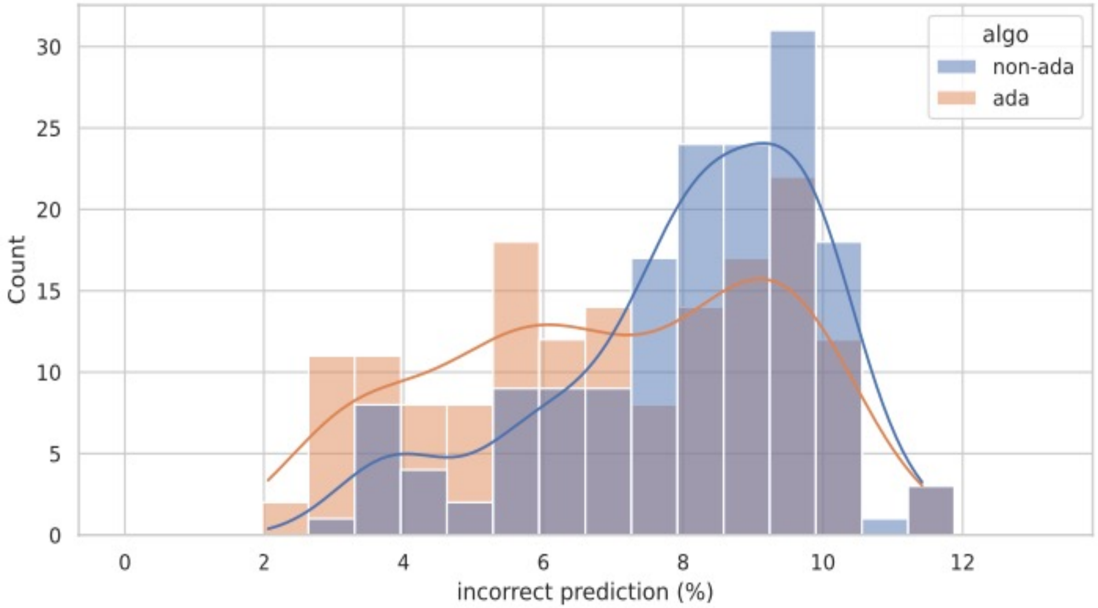
- Linear representation
- 2-layer CNN



[Mu & Gilmer 2019]

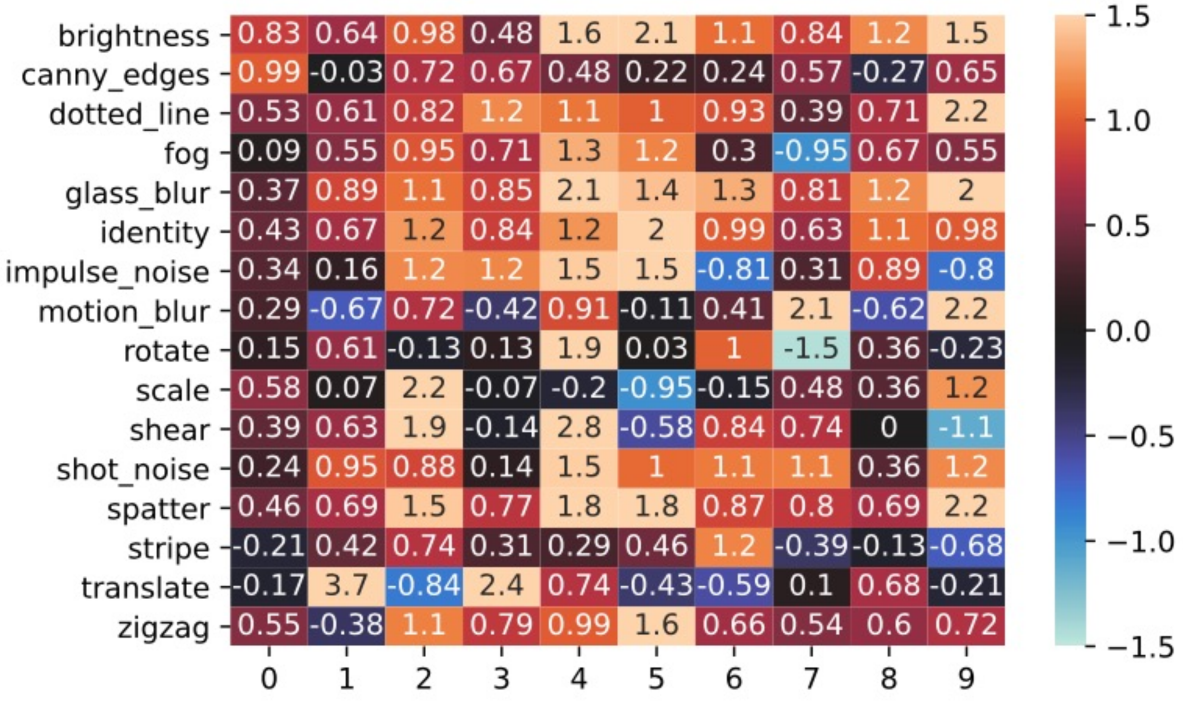
Experiments with Linear Representation

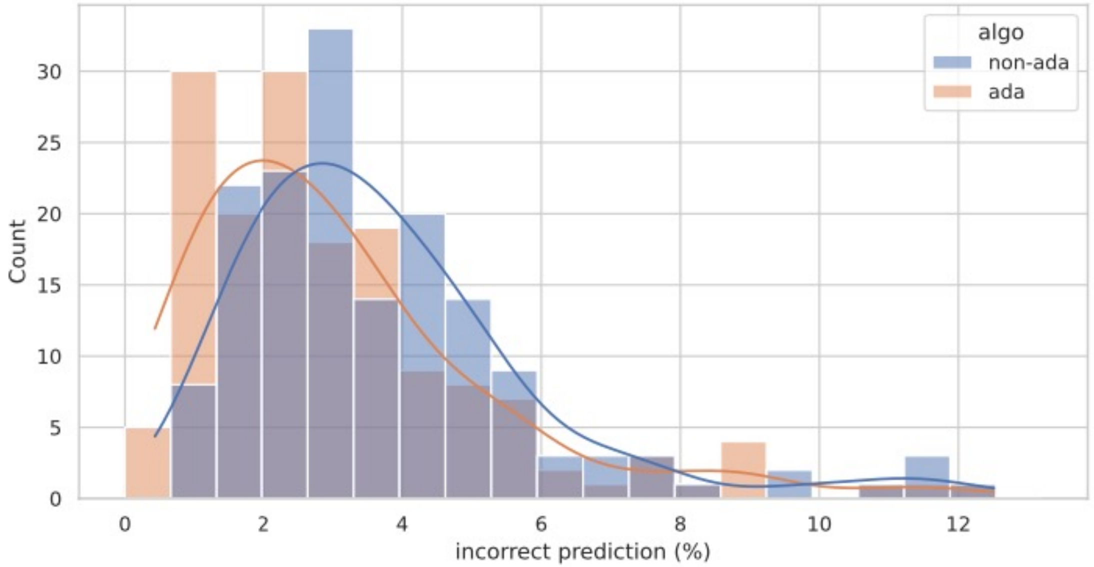




- Row: corruption. Number: improvement over uniform sampling.
  - Average improvement: 1.1% (baseline error ~8%).
  - Positive improvement on 136/160 tasks.
- Right: histogram summary of incorrect predictions.

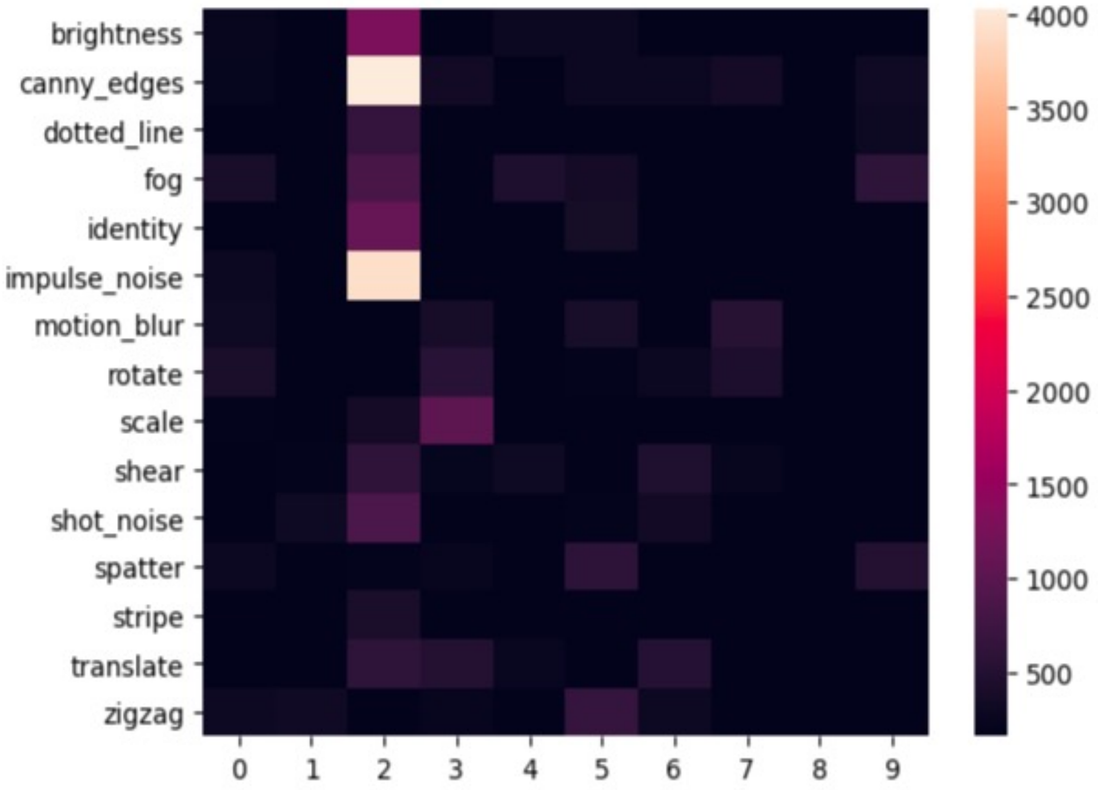
Experiments with ConvNet Representation



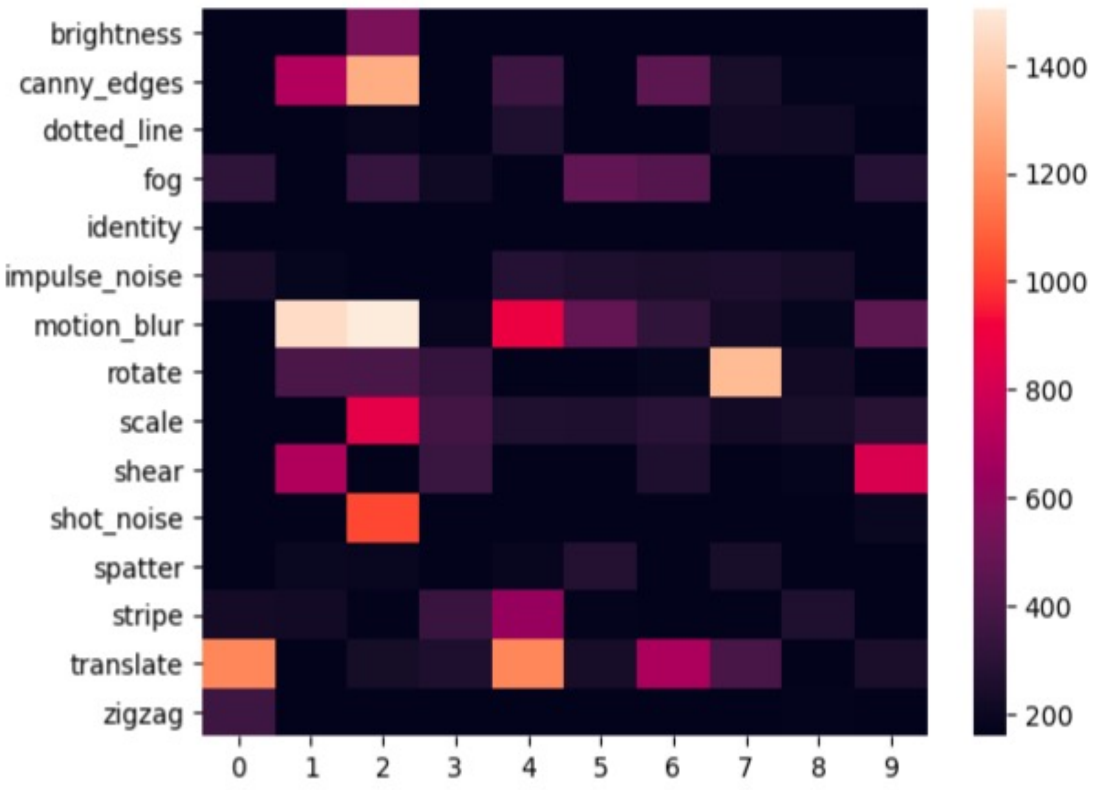


- Average improvement: 0.68% (baseline error ~6%).
- Positive improvement on 133/160 tasks.

Learned Task Relevance ν^*



Linear



ConvNet

- Target task: **digit 2** corrupted by **glass blur**.
- ν_t^* is large on tasks for digit 2.

Summary

Active learning is useful for representation learning:

- A formal definition of task relevance.
- Stronger than passive learning in theory and practice.
- Interpretability.

Future Work:

- Leverage active learning techniques for representation learning
- Other definitions of task relevance?
- Continuous source task space with infinite
- Active learning on finetune/ active prompt-based learning/ self-supervised learning

Thank You