Learning with Few Labeled Data

Pratik Chaudhari

Electrical and Systems Engineering &
Computer and Information Science

University of Pennsylvania

Menu

– **Few-shot image classification**

– A thermodynamical view of representation learning

Three regimes of image classification

High-shot regime

100–1000 samples/class

Three regimes of image classification

High-shot regime

100–1000 samples/class

Low-shot regime

10 samples/class

Three regimes of image classification

High-shot regime
100–1000 samples/class

Low-shot regime
10 samples/class

Extreme low-shot regime
1 sample/class

Problem formulation

Training set consists of labeled samples from lots of "tasks", e.g., classifying cars, cats, dogs, planes ...

Problem formulation

Training set consists of labeled samples from lots of "tasks", e.g., classifying cars, cats, dogs, planes . . .

Data from the new task, e.g., classifying strawberries has

w "ways": number of classes,

s "shots": number of labeled samples per class.

Problem formulation

Training set consists of labeled samples from lots of "tasks", e.g., classifying cars, cats, dogs, planes ...

Data from the new task, e.g., classifying strawberries has

w "ways": number of classes,

s "shots": number of labeled samples per class.

Few-shot setting considers the case when s is small.

A flavor of current few-shot algorithms

Meta-learning forms the basis for almost all current algorithms. Here's one successful instantiation.

A flavor of current few-shot algorithms

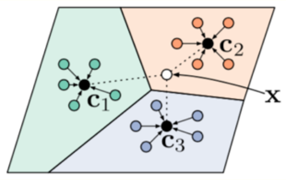
Meta-learning forms the basis for almost all current algorithms. Here's one successful instantiation.

Prototypical Networks [Snell et al., 2017]

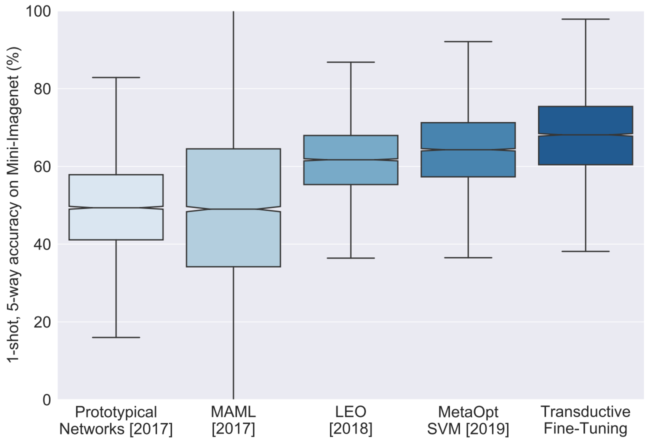
– Collect a meta-training set, this consists of a large number of related tasks

– Train one model on all these tasks to ensure that the clustering of features of this model correctly classifies the task

– If the test task comes from the same distribution as the meta-training tasks, we can use the clustering on the new task to classify new classes



How well does few-shot learning work today?



The key idea

A classifier trained on a dataset D_s is a function F that classifies data x using

$$\widehat{y} = F(x;\ D_s).$$

The key idea

A classifier trained on a dataset D_s is a function F that classifies data x using

$$\widehat{y} = F(x;\ D_s).$$

The parameters $\theta^* = \theta(D_s)$ of the classifier are a statistic of the dataset D_s obtained after training. Maintaining this statistic avoids having to search over functions F at inference time.

The key idea

A classifier trained on a dataset D_s is a function F that classifies data x using

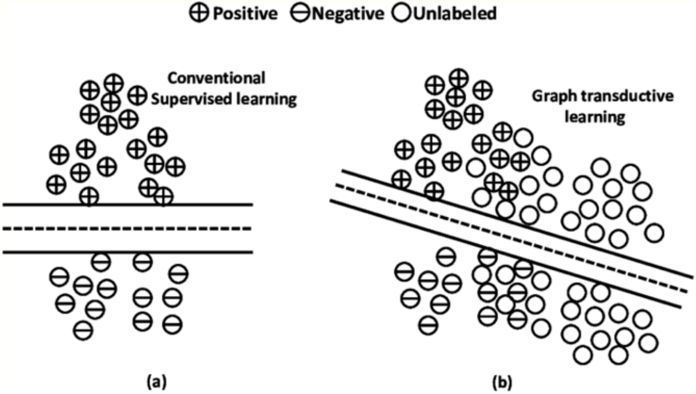
$$\widehat{y} = F(x;\ D_s).$$

The parameters $\theta^* = \theta(D_s)$ of the classifier are a statistic of the dataset D_s obtained after training. Maintaining this statistic avoids having to search over functions F at inference time.

We cannot learn a good (sufficient) statistic using few samples. So we will search over functions at test-time more explicitly

$$\widehat{y} = \underset{y_{N_s+1}}{\operatorname{argmin}} \ \min_{\theta} \ \frac{1}{N_s+1} \sum_{i=1}^{N_s+1} -\log p_\theta(y_i \mid x_i) + \frac{1}{2\lambda} \left\| \theta - \theta^*(D_s) \right\|^2.$$

Transductive Learning



A very simple baseline

1. Train a large deep network on the meta-training dataset with the standard classification loss

A very simple baseline

1. Train a large deep network on the meta-training dataset with the standard classification loss
2. Initialize a new "classifier head" on top of the logits to handle new classes

A very simple baseline

1. Train a large deep network on the meta-training dataset with the standard classification loss
2. Initialize a new "classifier head" on top of the logits to handle new classes
3. Fine-tune with the few labeled data from the new task

A very simple baseline

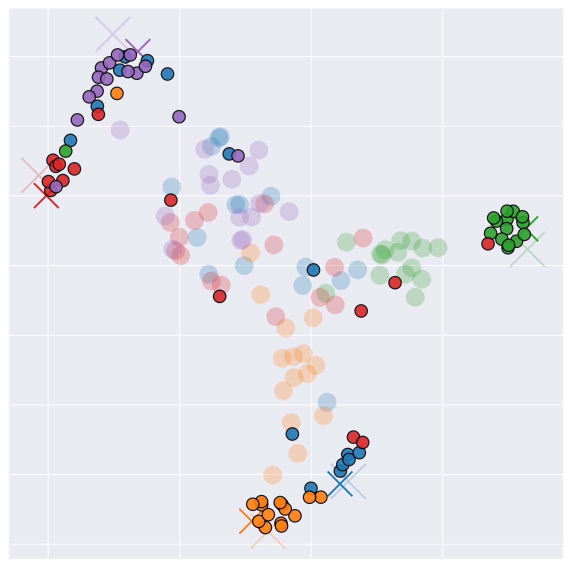
1. Train a large deep network on the meta-training dataset with the standard classification loss
2. Initialize a new "classifier head" on top of the logits to handle new classes
3. Fine-tune with the few labeled data from the new task
4. Perform transductive learning using the unlabeled test data

A very simple baseline

1. Train a large deep network on the meta-training dataset with the standard classification loss
2. Initialize a new "classifier head" on top of the logits to handle new classes
3. Fine-tune with the few labeled data from the new task
4. Perform transductive learning using the unlabeled test data

with a few practical tricks like cosine annealing of step-sizes, mixup regularization, 16-bit training, very heavy data augmentation, and label smoothing cross-entropy

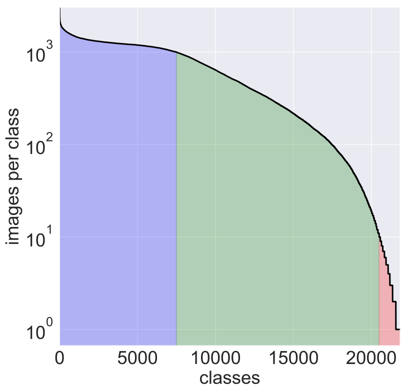
An example



Results on benchmark datasets

Algorithm	Architecture	Mini-ImageNet 1-shot (%)	Mini-ImageNet 5-shot (%)	Tiered-ImageNet 1-shot (%)	Tiered-ImageNet 5-shot (%)	CIFAR-FS 1-shot (%)	CIFAR-FS 5-shot (%)	FC-100 1-shot (%)	FC-100 5-shot (%)
Matching networks (Vinyals et al., 2016)	conv $(64)_{\times 4}$	46.6	60						
LSTM meta-learner (Ravi & Larochelle, 2016)	conv $(64)_{\times 4}$	43.44 ± 0.77	60.60 ± 0.71						
Prototypical Networks (Snell et al., 2017)	conv $(64)_{\times 4}$	49.42 ± 0.78	68.20 ± 0.66						
MAML (Finn et al., 2017)	conv $(32)_{\times 4}$	48.70 ± 1.84	63.11 ± 0.92						
R2D2 (Bertinetto et al., 2018)	conv $(96^k)_{\times 4}$	51.8 ± 0.2	68.4 ± 0.2			65.4 ± 0.2	79.4 ± 0.2		
TADAM (Oreshkin et al., 2018)	ResNet-12	58.5 ± 0.3	76.7 ± 0.3					40.1 ± 0.4	56.1 ± 0.4
Transductive Propagation (Liu et al., 2018b)	conv $(64)_{\times 4}$	55.51 ± 0.86	69.86 ± 0.65	59.91 ± 0.94	73.30 ± 0.75				
Transductive Propagation (Liu et al., 2018b)	ResNet-12	59.46	75.64						
MetaOpt SVM (Lee et al., 2019)	ResNet-12 *	62.64 ± 0.61	$\mathbf{78.63 \pm 0.46}$	65.99 ± 0.72	81.56 ± 0.53	72.0 ± 0.7	84.2 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
Support-based initialization (train)	WRN-28-10	56.17 ± 0.64	73.31 ± 0.53	$67.45 \pm 0.70^\dagger$	$82.88 \pm 0.53^\dagger$	70.26 ± 0.70	$83.82 \pm 0.49^\dagger$	36.82 ± 0.51	49.72 ± 0.55
Fine-tuning (train)	WRN-28-10	57.73 ± 0.62	$\mathbf{78.17 \pm 0.49}$	66.58 ± 0.70	$\mathbf{85.55 \pm 0.48}$	68.72 ± 0.67	$\mathbf{86.11 \pm 0.47}$	38.25 ± 0.52	$\mathbf{57.19 \pm 0.57}$
Transductive fine-tuning (train)	WRN-28-10	$\mathbf{65.73 \pm 0.68}$	$\mathbf{78.40 \pm 0.52}$	$\mathbf{73.34 \pm 0.71}$	$\mathbf{85.50 \pm 0.50}$	$\mathbf{76.58 \pm 0.68}$	$\mathbf{85.79 \pm 0.50}$	$\mathbf{43.16 \pm 0.59}$	$\mathbf{57.57 \pm 0.55}$
Activation to Parameter (Qiao et al., 2018) (train + val)	WRN-28-10	59.60 ± 0.41	73.74 ± 0.19						
LEO (Rusu et al., 2018) (train + val)	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09				
MetaOpt SVM (Lee et al., 2019) (train + val)	ResNet-12 *	64.09 ± 0.62	$\mathbf{80.00 \pm 0.45}$	65.81 ± 0.74	81.75 ± 0.53	72.8 ± 0.7	85.0 ± 0.5	47.2 ± 0.6	62.5 ± 0.6
Support-based initialization (train + val)	WRN-28-10	58.47 ± 0.66	75.56 ± 0.52	$67.34 \pm 0.69^\dagger$	$83.32 \pm 0.51^\dagger$	$72.14 \pm 0.69^\dagger$	$85.21 \pm 0.49^\dagger$	45.08 ± 0.61	60.05 ± 0.60
Fine-tuning (train + val)	WRN-28-10	59.62 ± 0.66	$\mathbf{79.93 \pm 0.47}$	66.23 ± 0.68	$\mathbf{86.08 \pm 0.47}$	70.07 ± 0.67	$\mathbf{87.26 \pm 0.45}$	43.80 ± 0.58	64.40 ± 0.58
Transductive fine-tuning (train + val)	WRN-28-10	$\mathbf{68.11 \pm 0.69}$	$\mathbf{80.36 \pm 0.50}$	$\mathbf{72.87 \pm 0.71}$	$\mathbf{86.15 \pm 0.50}$	$\mathbf{78.36 \pm 0.70}$	$\mathbf{87.54 \pm 0.49}$	$\mathbf{50.44 \pm 0.68}$	$\mathbf{65.74 \pm 0.60}$

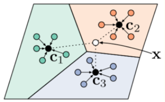
The ImageNet-21k dataset



1-shot, 5-way accuracies are as high as 89%, 1-shot 20-way accuracies are about 70%.

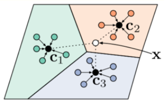
A proposal for systematic evaluation

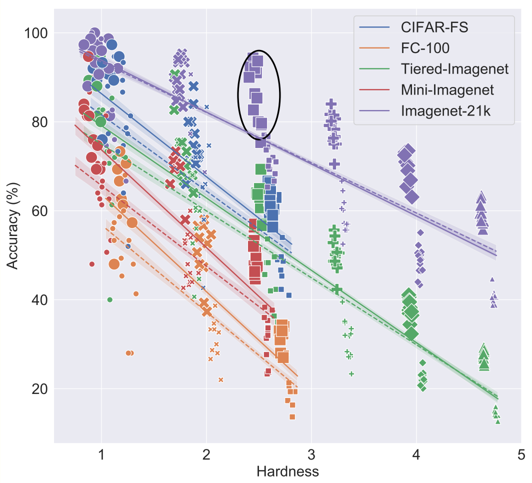
Hardness measures how difficult it is to classify a test task



A proposal for systematic evaluation

Hardness measures how difficult it is to classify a test task





Menu

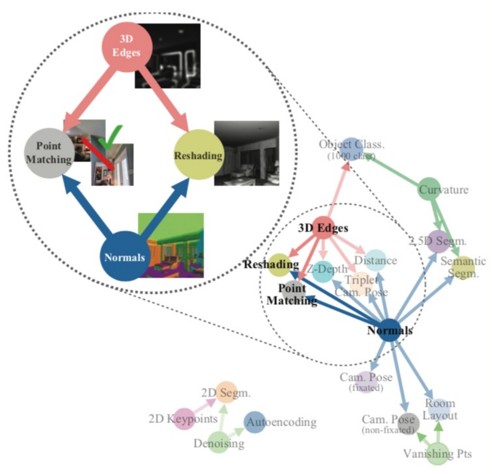
– Few-shot image classification

– **A thermodynamical view of representation learning**

Transfer learning

Transfer learning

Let's take an example from computer vision[1]



[1] Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., & Savarese, S. Taskonomy: Disentangling task transfer learning. CVPR 2018.

Information Bottleneck Principle

A generalization of rate-distortion theory for learning relevant representations of data [Tishby et al., 2000]

$$X \to Z \to Y$$

Z is a representation of the data X. We want

– Z to be sufficient to predict the target Y, and

– Z to be small in size, e.g., few number of bits.

$$\min_{Z|X,\ Y|Z} \{I(X;Z) - I(Z;Y)\}.$$

Doing well on one task requires throwing away nuisance information [Achille & Soatto, 2017].

The key idea

The IB Lagrangian simply minimizes $I(X; Z)$, it does not let us measure what was thrown away.

Choose a canonical task to measure discarded information. Setting

$$Y := X,$$

i.e., reconstruction of data, gives a special task. It is the superset of all tasks and forces the model to learn lossless representations.

The key idea

The IB Lagrangian simply minimizes $I(X;Z)$, it does not let us measure what was thrown away.

Choose a canonical task to measure discarded information. Setting

$$Y := X,$$

i.e., reconstruction of data, gives a special task. It is the superset of all tasks and forces the model to learn lossless representations.

The architecture we will focus on is

$$\begin{array}{ccccc} X & \xrightarrow{\text{Encoder } e_\theta(z|x)} & Z & \xrightarrow{\text{Classifier } c_\theta(y|z)} & Y \\ & & \Big\downarrow{\scriptstyle \text{Decoder } d_\theta(x|z)} & & \\ & & \widehat{X} & & \end{array}$$

An auto-encoder

Shanon entropy measures the complexity of data

$$H = \mathop{\mathbb{E}}_{x \sim p(x)}[-\log p(x)].$$

An auto-encoder

Shanon entropy measures the complexity of data

$$H = \mathop{\mathbb{E}}_{x \sim p(x)} [-\log p(x)].$$

Distortion D measures the quality of reconstruction

$$D = \mathop{\mathbb{E}}_{x \sim p(x)} \Big[- \int \mathrm{d}z \; e(z|x) \log d(x|z) \Big].$$

An auto-encoder

Shanon entropy measures the complexity of data

$$H = \mathop{\mathbb{E}}_{x \sim p(x)} [-\log p(x)].$$

Distortion D measures the quality of reconstruction

$$D = \mathop{\mathbb{E}}_{x \sim p(x)} \left[-\int \mathrm{d}z \; e(z|x) \log d(x|z) \right].$$

Rate R measures the average excess bits used to encode the representation

$$R = \mathop{\mathbb{E}}_{x \sim p(x)} \left[\int \mathrm{d}z \; e(z|x) \log \frac{e(z|x)}{m(z)} \right].$$

Rate-Distortion curve

We know that [Alemi et al., 2017]

$$H - D \leq \mathrm{KL}(e(z|x) \;||\; p(z|x)) \leq R, \tag{1}$$

this is the well-known ELBO (evidence lower-bound).

Let

$$F(\lambda) = \min_{e_\theta(z|x), m_\theta(z), d_\theta(x|z)} \{R + \lambda D\}.$$

Rate-Distortion curve

We know that [Alemi et al., 2017]

$$H - D \leq \mathrm{KL}(e(z|x) \mid\mid p(z|x)) \leq R, \tag{1}$$

this is the well-known ELBO (evidence lower-bound).

Let

$$F(\lambda) = \min_{e_\theta(z|x), m_\theta(z), d_\theta(x|z)} \{R + \lambda D\}.$$

This is a Lagrange relaxation of the fact that given a variational family and data there is an optimal value $R = \mathrm{func}(D)$ that best sandwiches (1).

Rate-Distortion curve

We know that [Alemi et al., 2017]

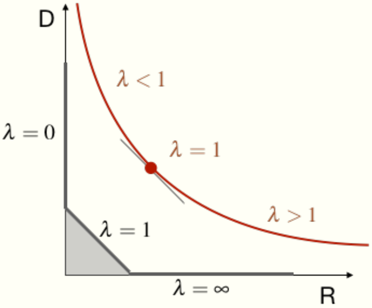
$$H - D \leq \text{KL}(e(z|x) \,||\, p(z|x)) \leq R, \tag{1}$$

this is the well-known ELBO (evidence lower-bound).

Let

$$F(\lambda) = \min_{e_\theta(z|x), m_\theta(z), d_\theta(x|z)} \{R + \lambda D\}.$$

This is a Lagrange relaxation of the fact that given a variational family and data there is an optimal value $R = \text{func}(D)$ that best sandwiches (1).



Rate-Distortion-Classification (RDC) surface

Let us extend the Lagrangian to

$$F(\lambda, \gamma) = \min_{e_\theta(z|x), m_\theta(z), d_\theta(x|z)} \{R + \lambda D + \gamma C\}$$

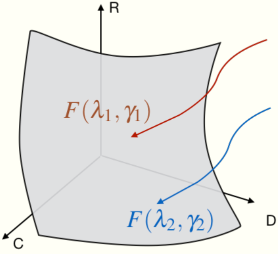
where the classification loss is

$$C = \mathbb{E}_{x \sim p(x), y \sim p(y|x)} \left[- \int \mathrm{d}z \; e(z|x) \log c(y|z) \right]$$

Can also include other quantities like the entropy S of the model parameters

$$S = \mathbb{E}_{x \sim p(x),\; y \sim p(y|x)} \left[\log \frac{p(\theta|\{x, y\})}{m(\theta)} \right]$$

Rate-Distortion-Classification (RDC) surface



The existence of a convex surface func$(R, D, C, S) = 0$ tying together these functionals allows a formal connection to thermodynamics [Alemi and Fischer 2018]

$$dR = -\lambda\, dD - \gamma\, dC - \sigma\, dS.$$

Just like energy is conserved in physical processes, information is conserved in the model, either it is in the encoder-classifier pair or it is in the decoder.

Equilibrium surface of optimal free-energy

The RDC surface determines all possible representations that can be learnt from given data. Can solve the variational problem for $F(\lambda, \gamma)$ to get

$$Z_{\theta,x} = \int \mathrm{d}z\ m_\theta(z)\ d_\theta(x|z)^\lambda\ c_\theta(y_x|z)^\gamma$$

and

$$F(\lambda, \gamma) = \min_{\theta \in \Theta} \mathop{\mathbb{E}}_{x \sim p(x)} [-\log Z_{\theta,x}] := J(\theta, \lambda, \gamma)$$

This is called the "equilibrium surface" because training converges to some point on this surface. We now construct ways to travel on the surface

$$\Theta_{\lambda,\gamma} = \{\theta \in \Theta : \mathop{\mathbb{E}}_{x \sim p(x)} [-\log Z_{\theta,x}] = F(\lambda, \gamma)\}.$$

Equilibrium surface of optimal free-energy

The RDC surface determines all possible representations that can be learnt from given data. Can solve the variational problem for $F(\lambda, \gamma)$ to get

$$Z_{\theta,x} = \int \mathrm{d}z \; m_\theta(z) \; d_\theta(x|z)^\lambda \; c_\theta(y_x|z)^\gamma$$

and

$$F(\lambda, \gamma) = \min_{\theta \in \Theta} \mathop{\mathbb{E}}_{x \sim p(x)} [-\log Z_{\theta,x}] := J(\theta, \lambda, \gamma)$$

This is called the “equilibrium surface” because training converges to some point on this surface. We now construct ways to travel on the surface

$$\Theta_{\lambda,\gamma} = \{\theta \in \Theta : \mathop{\mathbb{E}}_{x \sim p(x)} [-\log Z_{\theta,x}] = F(\lambda, \gamma)\}.$$

The surface depends on data $p(x, y)$.

An iso-classification loss process

A quasi-static process happens slowly enough for the system to remain in equilibrium with its surroundings, e.g., reversible expansion of an ideal gas.

We will create a quasi-static process to travel on the RDC surface. This constraint is

$$\nabla_\theta J(\theta, \lambda, \gamma) = 0 \text{ for all } \theta \in \Theta_{\lambda,\gamma}.$$

An iso-classification loss process

A quasi-static process happens slowly enough for the system to remain in equilibrium with its surroundings, e.g., reversible expansion of an ideal gas.

We will create a quasi-static process to travel on the RDC surface. This constraint is

$$\nabla_\theta J(\theta, \lambda, \gamma) = 0 \text{ for all } \theta \in \Theta_{\lambda,\gamma}.$$

e.g., if we want classification loss to be constant in time, we need

$$\frac{\mathrm{d}}{\mathrm{d}t} \nabla_\theta J = 0 \qquad \text{(Quasi-Static Condition)}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} C = 0 \qquad \text{(Iso-classification Condition).}$$

An iso-classification loss process

A quasi-static process happens slowly enough for the system to remain in equilibrium with its surroundings, e.g., reversible expansion of an ideal gas.

We will create a quasi-static process to travel on the RDC surface. This constraint is

$$\nabla_\theta J(\theta, \lambda, \gamma) = 0 \text{ for all } \theta \in \Theta_{\lambda,\gamma}.$$

e.g., if we want classification loss to be constant in time, we need

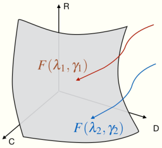
$$\frac{\mathrm{d}}{\mathrm{d}t} \nabla_\theta J = 0 \qquad \text{(Quasi-Static Condition)}$$
$$\frac{\mathrm{d}}{\mathrm{d}t} C = 0 \qquad \text{(Iso-classification Condition).}$$

Can also impose other constraints, e.g.,

$$\frac{\mathrm{d}}{\mathrm{d}t}\{C + \gamma^{-1} R\} = 0$$

which is the objective for learning Bayesian neural networks.

Implementing processes on the RDC surface



Could pick particular values of $(\dot{\lambda}, \dot{\gamma})$ to get

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\theta J = \nabla^2_\theta J \, \dot{\theta} + \dot{\lambda} \frac{\partial}{\partial \lambda} \nabla_\theta J + \dot{\gamma} \frac{\partial}{\partial \gamma} \nabla_\theta J$$

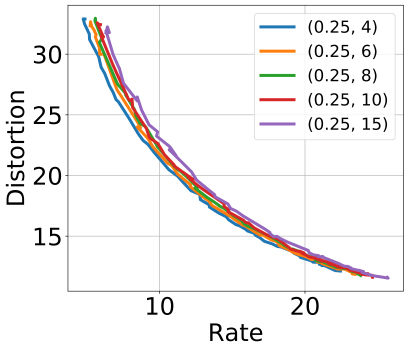
this requires inverting $\nabla^2_\theta J$.

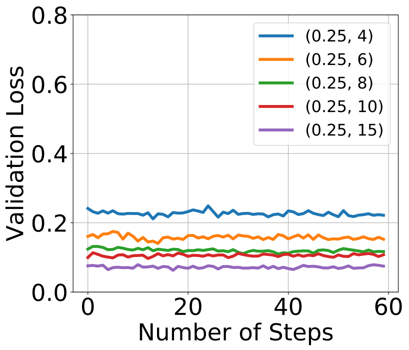
We exploit constraints like $0 = C_\lambda \dot{\lambda} + C_\gamma \dot{\gamma}$ to get

$$\dot{\lambda} = -\alpha \frac{\partial}{\partial C} \gamma = -\alpha \frac{\partial^2}{\partial F^2} \gamma$$

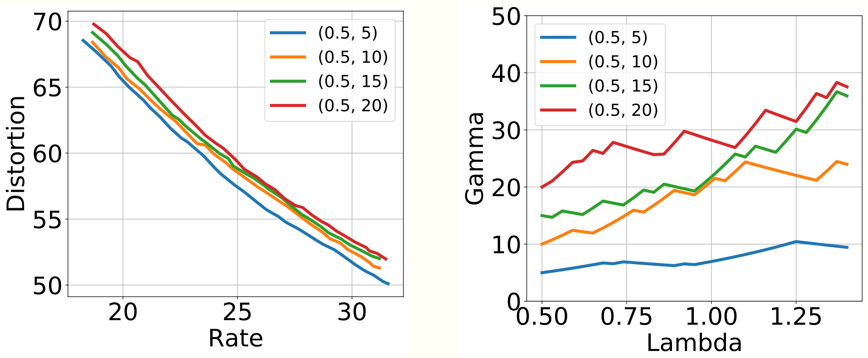
$$\dot{\gamma} = \alpha \frac{\partial}{\partial C} \lambda = \alpha \frac{\partial^2}{F} \partial \lambda \partial \gamma$$

Iso-C process for different initial (λ, γ): MNIST

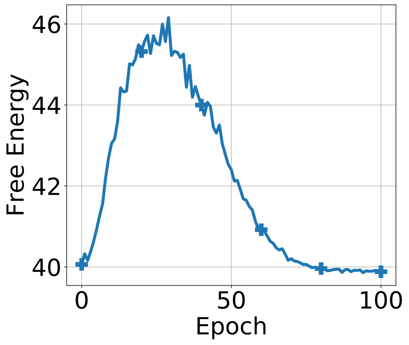


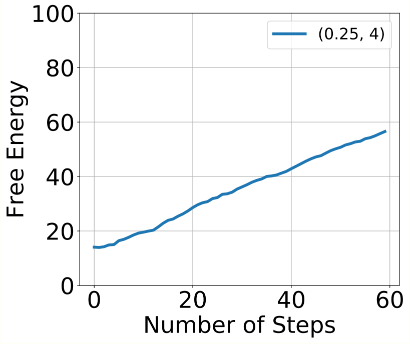


Iso-C process for different initial (λ, γ): CIFAR-10



Iso-C process: Variation of $F(\lambda, \gamma)$ during equilibration





Transferring to new tasks

The RDC surface depends on data $p(x, y)$. We now move the data distribution from the source task to the target task, e.g., interpolate it as

$$p(x, y, t) = (1 - t)\ p^s(x, y) + t\ p^t(x, y).$$

The quasi-static iso-classification process

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\theta J = \frac{\mathrm{d}}{\mathrm{d}t} C$$

can be executed on this changing data distribution.

Transferring to new tasks

The RDC surface depends on data $p(x, y)$. We now move the data distribution from the source task to the target task, e.g., interpolate it as

$$p(x, y, t) = (1 - t)\ p^s(x, y) + t\ p^t(x, y).$$

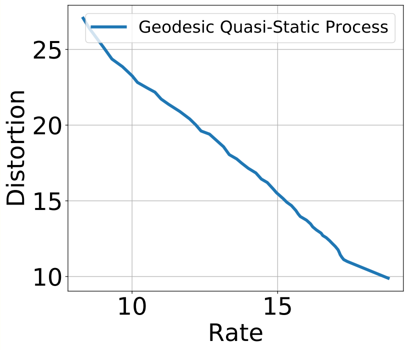
The quasi-static iso-classification process

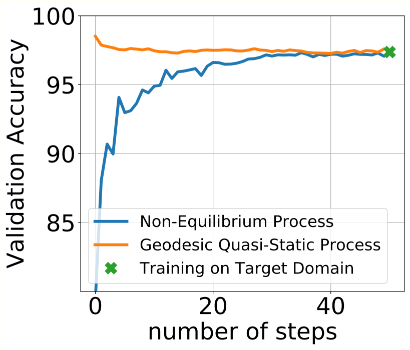
$$0 = \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\theta J = \frac{\mathrm{d}}{\mathrm{d}t} C$$

can be executed on this changing data distribution.

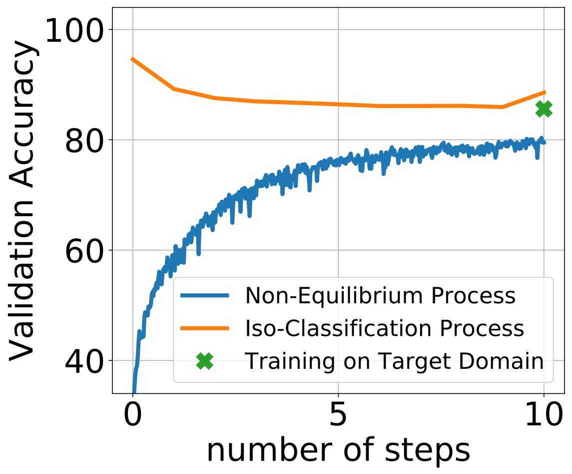
This is a completely controlled mechanism to transfer representations, the classification loss is unchanged upon going to the target dataset.

Iso-C process: MNIST 0–4 to 5–9





Iso-C process: CIFAR-10 Vehicles to Animals



Summary

Simple methods such as transductive fine-tuning work extremely well for few-shot learning. This is really because of powerful function approximators such as neural networks.

The RDC surface is a fundamental quantity and enables principled methods for transfer learning. Also unlocks new paths to understanding regularization and properties of neural architecture for classical supervised learning.

We did well in the era of big data without understanding much about data; this is unlikely to work in the age of little data.

Email questions to pratikac@seas.upenn.edu

Read more at

1. Dhillon, G., Chaudhari, P., Ravichandran, A., and Soatto, S. (2019). A baseline for few-shot image classification. arXiv:1909.02729. ICLR 2020.
2. Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., & Soatto, S. (2020). Rethinking the Hyperparameters for Fine-tuning. arXiv:2002.11770. ICLR 2020.
3. Fakoor, R., Chaudhari, P., Soatto, S., & Smola, A. J. (2019). Meta-Q-Learning. arXiv:1910.00125. ICLR 2020.
4. Gao, Y., and Chaudhari, P. (2020). A free-energy principle for representation learning. arXiv:2002.12406.