On the Role of Neural Collapse in Transfer and Few-Shot Learning

Tomer Galanti

December 7, 2022

Based on joint work with

András György

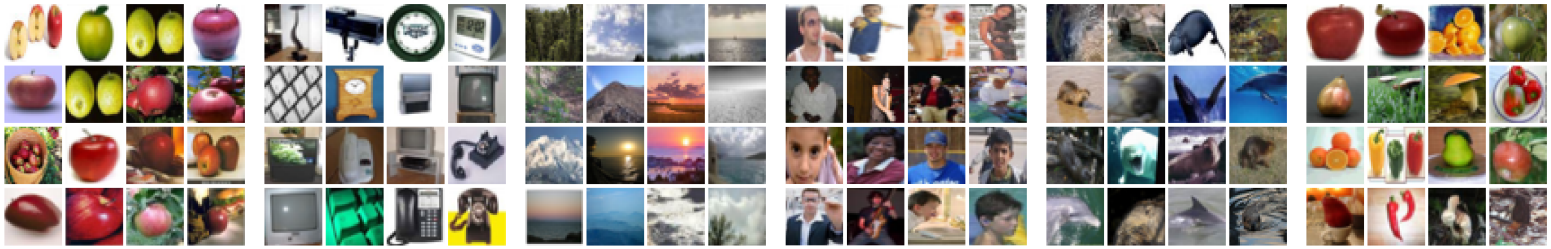
DeepMind

Marcus Hutter

DeepMind

Motivation

- An agent has access to a huge number of data over its lifetime



(Krizhevsky et al. 2009)

- Learn new concepts from few examples

(Bertinetto et al. 2018)

Motivation

Target task: a k-class classi cation problem.

k classes.

Few samples per class.

Motivation

Target task: a k-class classi cation problem.

k classes.

Few samples per class.

Problem: directly training on the data would probably result in over tting.

Motivation

Target data: classi cation task with a few samples per class.

Source data: many classes with lots of data per class.

Goal: train a feature map f on the source data that is “good” for the target task.

Motivation

Approach 1: Few-shot learning

Split the source data into many separate tasks.

(Bertinetto et al. 2018)

Motivation

Approach 1: Few-shot learning

Split the source data into many separate tasks.

Use the current feature map f.

(Bertinetto et al. 2018)

Motivation

Approach 1: Few-shot learning

Split the source data into many separate tasks.

Use the current feature map f.

For each random task and few samples S from the task, train a classifier g_S on top of f.

(Bertinetto et al. 2018)

Motivation

Approach 1: Few-shot learning

Split the source data into many separate tasks.
Use the current feature map f.
For each random task and few samples S from the task, train a classi er g_S on top of f.
Choose f that minimizes the expected error (w.r.t. the task + data) of classi ers g_S f.

(Bertinetto et al. 2018)

Motivation

Approach 1: Few-shot learning

Split the source data into many separate tasks.
Use the current feature map f.
For each random task and few samples S from the task, train a classi er g_S on top of f.
Choose f that minimizes the expected error (w.r.t. the task + data) of classi ers g_S f.

Examples:

Matching Networks (Vinyals et al. 2016).

LSTM Meta-Learner (Ravi et al. 2017).

MAML (Finn et al. 2017).

(Bertinetto et al. 2018)

Motivation

Approach 2: Transfer learning (Caruana 1995; Bengio 2012; Yosinski et al. 2012).

- Treat the source data as one classification task.
- Train one classifier $g \quad f$ on the source task (e.g., ResNet-50 on ImageNet).
- Train a small complexity classifier g (e.g., a linear layer) on top of f using the target data.

Motivation

- Transfer learning works well between tasks of different modalities (Xu et al. 2022).
- Large language models (GPT-3, Bert, etc').
- Transferring between different tasks (e.g., image classification to image segmentation).

Motivation

Surprisingly, transfer learning is competitive with the first approach in few-shot learning (Dhillon et al. 2020; Tian et al. 2020).

Motivation

Surprisingly, transfer learning is competitive with the first approach in few-shot learning (Dhillon et al. 2020; Tian et al. 2020).

Main result: an explanation of this success via neural collapse.

Problem Setup

Target task:

- k classes P_c.
- Few samples (n) per class S_c.

Problem Setup

Target task:

k classes P_c.

Few samples (n) per class S_c.

Source task:

l classes $\tilde{P}_c$.

Many samples (m) per class $\tilde{S}_c$.

Problem Setup

Target task:

k classes P_c.

Few samples (n) per class S_c.

Source task:

l classes $\tilde{P}_c$.

Many samples (m) per class $\tilde{S}_c$.

Algorithm:

We train on a model $g \circ f$ to classify $\tilde{S}$.

We train g to classify $f(S)$.

Illustration

Plan for the analysis

Neural collapse – on the training set.

Neural collapse generalizes to new samples.

Neural collapse generalizes to new classes.

Neural collapse (in new classes) implies few-shot learnability.

Neural Collapse

According to Papyan et al. (2020)

- Train a large overparameterized NN for classi cation.
- The feature embeddings belonging to training samples of the same class concentrate around their class means.

Reformulation of neural collapse

Class-distance-normalized variance: for two class distributions Q_1 and Q_2 with feature embedding f:

$$V_f(Q_1; Q_2) = \frac{\mathrm{Var}_f(Q_1) + \mathrm{Var}_f(Q_2)}{2k\ _f(Q_1) \quad _f(Q_2)k^2}$$

where

$_f(Q) = E_{x\ Q}[f(x)]$ - feature mean for Q;

$\mathrm{Var}_f(Q) = E_{x\ Q}[kf(x) \quad _f(Q)k^2]$ - feature variance for Q.

Reformulation of neural collapse

Class-distance-normalized variance: for two class distributions Q_1 and Q_2 with feature embedding f:

$$V_f(Q_1; Q_2) = \frac{\mathrm{Var}_f(Q_1) + \mathrm{Var}_f(Q_2)}{2k\ _f(Q_1) \quad _f(Q_2)k^2}$$

where

$_f(Q) = E_{x\ Q}[f(x)]$ - feature mean for Q;

$\mathrm{Var}_f(Q) = E_{x\ Q}[kf(x) \quad _f(Q)k^2]$ - feature variance for Q.

Neural collapse: For empirical distributions $\hat{S}_i$ and $\hat{S}_j$ for classes $i; j$ in the training data

$$\lim_{t!1} V_f(\hat{S}_i; \hat{S}_j) = 0$$

Experiments

3 layers

5 layers

10 layers

14 layers

18 layers

20 layers

Figure: Normalized variance for Convolutional network of varying depth trained on CIFAR10.

Re ned plan for the analysis

Assumption: The embeddings of training samples are clustered.

Step 1: The embeddings of test samples are clustered.

Step 2: The embeddings of samples from new classes are clustered.

Step 3: if Step 2 holds, we can ef ciently learn to classify with very few samples.

Assumptions

What is the relationship between the source and target tasks?

Assumptions

What is the relationship between the source and target tasks?

What if the two tasks are arbitrary?

Assumptions

What is the relationship between the source and target tasks?

What if the two tasks are arbitrary?

Then, we should not expect the model to transfer very well...

Assumptions

For simplicity, we think of the classes as random classes from ImageNet.

Assumptions

For simplicity, we think of the classes as random classes from ImageNet.

The source classes $\tilde{P}_1, \ldots, \tilde{P}_l$ and target classes $P_1, \ldots, P_k$ are i.i.d. samples from the same distribution of classes D.

Objective

Suppose we have a downstream task P with classes $P_1; : : : ; P_k$ D , we want to minimize the expected error of random classi ers $h_{S;f} = g_S$ f for random sets $S = [_{i=1}^{k} S_i$ (each S_i is of size n).

Objective

Suppose we have a downstream task P with classes $P_1, \ldots, P_k \sim D$, we want to minimize the expected error of random classifiers $h_{S,f} = g_S \circ f$ for random sets $S = \bigcup_{i=1}^{k} S_i$ (each S_i is of size n).

$$\mathbb{E}_P \mathrm{Err}_P(f) := \underbrace{\mathbb{E}_{P_1,\ldots,P_k \sim D}}_{\text{random target task } P} \underbrace{\mathbb{E}_{S_1,\ldots,S_k}}_{\text{few samples per class}} \underbrace{\mathbb{E}_{(x,y)\sim P} \mathbb{I}[h_{S,f}(x) \neq y]}_{\text{error on task } P}$$

Objective

Suppose we have a downstream task P with classes $P_1, \ldots, P_k \sim D$, we want to minimize the expected error of random classifiers $h_{S,f} = g_S \circ f$ for random sets $S = \bigcup_{i=1}^{k} S_i$ (each S_i is of size n).

$$\mathbb{E}_P \mathrm{Err}_P(f) := \underbrace{\mathbb{E}_{P_1,\ldots,P_k \sim D}}_{\text{random target task } P} \underbrace{\mathbb{E}_{S_1,\ldots,S_k}}_{\text{few samples per class}} \underbrace{\mathbb{E}_{(x,y)\sim P} \mathbb{I}[h_{S,f}(x) \neq y]}_{\text{error on task } P}$$

$$h_{S,f}(x) := \arg\min_{c \in [k]} \| f(x) - f(S_c) \|$$

Objective

Suppose we have a downstream task P with classes $P_1, \ldots, P_k \quad D$, we want to minimize the expected error of random classifiers $h_{S,f} = g_S \quad f$ for random sets $S = \bigcup_{i=1}^{k} S_i$ (each S_i is of size n).

$$E_P \mathrm{Err}_P(f) := \underbrace{E_{P_1,\ldots,P_k \ D}}_{\text{random target task } P} \ \underbrace{E_{S_1,\ldots,S_k}}_{\text{few samples per class}} \ \underbrace{E_{(x,y) \ P} \mathbb{I}[h_{S,f}(x), y]}_{\text{error on task } P}$$

$$h_{S,f}(x) := \arg\min_{c \in [k]} \| f(x) \quad f(S_c) \|$$

We think of this objective as an expected error on a downstream task.

First Attempt

Assume neural collapse happens at training: $V_f(S_i; S_j) \not\to 0$.

First Attempt

Assume neural collapse happens at training: $V_f(\tilde{S}_i, \tilde{S}_j) \not\to 0$.

Neural collapse generalizes to new samples:

$$V_f(\tilde{P}_i, \tilde{P}_j) \approx V_f(\tilde{S}_i, \tilde{S}_j) + o_m(1)$$

where $\tilde{P}_i$ and $\tilde{P}_j$ are the corresponding class distributions.

First Attempt

Assume neural collapse happens at training: $V_f(S_i; S_j) \not\to 0$.

Neural collapse generalizes to new samples:

$$V_f(\hat{P}_i; \hat{P}_j) \approx V_f(S_i; S_j) + o_m(1)$$

where $\hat{P}_i$ and $\hat{P}_j$ are the corresponding class distributions.

Neural collapse generalizes to new classes:

$$\mathbb{E}_{P_1;P_2 \sim P}[V_f(P_1; P_2)] \approx \mathrm{Avg}_{i,j}[V_f(\hat{P}_i; \hat{P}_j)] + o_l(1)$$

First Attempt

Assume neural collapse happens at training: $V_f(S_i; S_j) \;\; ! \;\; 0.$

- Neural collapse generalizes to new samples:

$$V_f(P_i; P_j) \quad . \quad V_f(S_i; S_j) + o_m(1)$$

  where P_i and P_j are the corresponding class distributions.

- Neural collapse generalizes to new classes:

$$\mathrm{E}_{P_1;P_2 \;\; \mathrm{P}}[V_f(P_1; P_2)] \quad . \quad \mathrm{Avg}_{i,j}[V_f(P_i; P_j)] + o_l(1)$$

- Neural collapse implies few-shot learnability: for a linear classifier trained with n samples over 2 classes,

$$Err_{P_{ij}}(f) \qquad 1 + \frac{1}{n} \quad V_f(P_1; P_2)$$

Problems

Problems:

- If f $_f(P)$g$_P$ is bounded and the support of D is infinite, then E$_{P_c, P_{c^0}}$ $[V_f(P_c; P_{c^0})] = 1$.
- Generalization bounds typically depend on sup$_{f2F}$ `(f).

Generalization to New Classes

Theorem (Informal)

Let F *be a class of q depth ReLU neural networks f* : $\mathbb{R}^d$! $\mathbb{R}^p$. *Let* D *be a distribution over classes and let* fP_cg$_{c=1}^{l}$ D^l. *Then, with a high probability over the selection of the classes, for every f* 2 F *and* > 0, *we have*

$$\mathrm{E}_P Err_P(f) \quad (k \quad 1)\ \mathrm{Avg}_{i,j}\, Err_{P_{ij}}(f) + \mathrm{O}\left(\frac{(k \quad 1)\ p\ \mathrm{C}(f)\ \sqrt{\frac{p}{q \log(l)}}}{\bar{p}_l}\right);$$

Generalization to New Samples

Theorem (Informal)

Let $\mathcal{F}$ *be a class of ReLU neural networks of depth* q*. Let* P_i *and* P_j *be two class-conditional distributions. Then, with high probability over the selection of* $S_i \sim P_i^m$ *and* $S_j \sim P_j^m$*, for any* $f \in \mathcal{F}$ *and* $\epsilon > 0$*, we have*

$$Err_{P_{ij}}(f) \leq \frac{m}{m+n} Err_{S_{ij}}(f) + O\left(\frac{C(f)}{\sqrt{m}} \sqrt{np + \frac{p}{q}}\right)$$

Few-Shot Learning and Normalized Variance

- **Two classes:** $Q_i; Q_j$.
- **Dataset:** datasets S_c Q_c^n.
- **Feature map:** $f : \mathrm{R}^d$! R^p (e.g., pretrained).
- = O(k $_f(Q_i)$ $_f(Q_j)$k).

$$Err_{Q_{ij}}(f) \quad . \quad (1 + \tfrac{1}{n}) \quad V_f(Q_i; Q_j)$$

Few-Shot Learning and Normalized Variance

If f Q_i and f Q_j are also spherically symmetric,

$$Err_{Q_{ij}}(f) \quad . \quad (\tfrac{1}{p} + \tfrac{1}{n}) \quad V_f(Q_i; Q_j)$$

Few-Shot Learning and Normalized Variance

Theorem (Informal)

Let F *be a class of q depth ReLU neural networks* $f : \mathbb{R}^d \to \mathbb{R}^p$. *With a high probability over the selection of the training data* $\{S_c\}_{c=1}^l$, *for every* $f \in F$, *we have*

$$E_P Err_P(f) \quad (k \quad 1)\frac{m}{m\ n}\left(1 + \frac{1}{n}\right) \quad \mathrm{Avg}_{i,j} V_f(S_i; S_j)$$

$$+ O\left(\sqrt{\frac{(k \quad 1) \quad p \quad C(f)}{l \quad \min_{i,j} \| \quad f(S_i) \quad f(S_j)\|}\sqrt{q\log(l)}} + \sqrt{\frac{C(f) \quad np}{m \quad \min_{i,j} \| \quad f(S_i) \quad f(S_j)\|}}\sqrt{q}\right)$$

Experimental Setup

Phase 1 (train)

- Train $h = g \circ f$ to minimize cross-entropy classification loss on the source classes.

Phase 2 (eval)

- Few-shot task: 5 classes, $n = 1; 5$ samples per-class.
- Train ridge regression on top of f using the $5 \times n$ dataset with one-hot labels.
- Evaluate on test samples from each class.
- Average over many tasks.

Empirical Results

Method	Architecture	Mini-ImageNet				CIFAR-FS				FC-100			
		1-shot		5-shot		1-shot		5-shot		1-shot		5-shot	
Matching Networks [VBL+ 16]	64-64-64-64	43:56	0:84	55:31	0:73	-		-		-		-	
LSTM Meta-Learner [RL17]	64-64-64-64	43:44	0:77	60:60	0:71	-		-		-		-	
MAML [FAL17]	32-32-32-32	48:70	1:84	63:11	0:92	58:9	1:9	71:5	1:0	-		-	
Prototypical Networks [SSZ17]	64-64-64-64	49:42	0:78[y]	68:20	0:66[y]	55:5	0:7	72:0	0:6	35:3	0:6	48:6	0:6
Relation Networks [SYZ+ 18]	64-96-128-256	50:44	0:82	65:32	0:7	55:0	1:0	69:3	0:8	-		-	
SNAIL [MRCA18]	ResNet-12	55:71	0:99	68:88	0:92	-		-		-		-	
TADAM [ORLL18]	ResNet-12	58:50	0:30	76:7	0:3	-		-		40:1	0:4	56:1	0:4
AdaResNet [MYMT18]	ResNet-12	56:88	0:62	71:94	0:57	-		-		-		-	
Dynamics Few-Shot [GK18]	64-64-128-128	56:20	0:86	73:0	0:64	-		-		-		-	
Activation to Parameter [QLSY18]	WRN-28-10	59:60	0:41[y]	73:74	0:19[y]	-		-		-		-	
R2D2 [BHTV19]	96-192-384-512	51:2	0:6	68:8	0:1	65:3	0:2	79:4	0:1	-		-	
Shot-Free [RBS19]	ResNet-12	59:04	n=a	77:64	n=a	69:2	n=a	84:7	n=a	-		-	
TEWAM [QSL+ 19]	ResNet-12	60:07	n=a	75:90	n=a	70:4	n=a	81:3	n=a	-		-	
TPN [LLP+ 19]	ResNet-12	55:51	0:86	75:64	n=a	-		-		-		-	
LEO [RRS+ 19]	WRN-28-10	61:76	0:08[y]	77:59	0:12[y]	-		-		-		-	
MTL [SLCS19]	ResNet-12	61:20	1:80	75:50	0:80	-		-		-		-	
OptNet-RR [LMRS19]	ResNet-12	61:41	0:61	77:88	0:46	72:6	0:7	84:3	0:5	40:5	0:6	57:6	0:9
MetaOptNet [LMRS19]	ResNet-12	62:64	0:61	78:63	0:46	72:0	0:7	84:2	0:5	41:1	0:6	55:3	0:6
Transductive Fine-Tuning [DCRS20]	WRN-28-10	65:73	0:68	78:40	0:52	76:58	0:68	85:79	0:5	43:16	0:59	57:57	0:55
Distill-simple [TWK+ 20]	ResNet-12	62:02	0:63	79:64	0:44	71:5	0:8	86:0	0:5	42:6	0:7	59:1	0:6
Distill [TWK+ 20]	ResNet-12	64:82	0:60	82:14	0:43	73:9	0:8	86:9	0:5	44:6	0:7	60:9	0:6
Ours (simple)	WRN-28-4	58:12	1:19	72:0	0:99	68:81	1:20	81:49	0:98	44:96	1:14	57:21	10:89
Ours (lr scheduling)	WRN-28-4	60:37	1:25	72:35	0:99	70:0	1:29	81:39	0:96	43:42	1:0	54:14	1:1
Ours (lr scheduling + model selection)	WRN-28-4	61:27	1:14	74:74	0:76	72:37	1:12	82:94	0:89	45:81	1:27	56:85	1:30

Table: 1-shot and 5-shot classi cation performance on Mini-ImageNet, CIFAR-FS, and FC-100.

Experimental evidence for NC2

$= 2^4$ $= 2^5$ $= 2^6$ $= 2^7$

We plot $\min_{i,j} \| f(S_i) - f(S_j) \|$ when training WRN-28-4 on CIFAR-FS.

Empirical Results

(a) source train CDNV (b) source test CDNV

(c) target CDNV (d) 5-shot target accuracy

Figure: Within-class variation collapse of wide ResNet on CIFAR-FS with varying number of source classes.

Conclusions

First theoretical proof that learning embeddings with foundation models works.

Small normalized variance implies good few-shot performance.

Theoretical and empirical evidence on the relations between NC and transferability.

Conclusions

First theoretical proof that learning embeddings with foundation models works.

Small normalized variance implies good few-shot performance.

Theoretical and empirical evidence on the relations between NC and transferability.

Some open questions

Can we get better transferability by explicitly enforcing neural collapse?

Are there other structures that are favorable for adaptivity and transferability?

Can the analysis be extended beyond classi cation?

What about transfer between different modalities? Tasks?

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi.
Meta-learning with differentiable closed-form solvers.
In *International Conference on Learning Representations*, 2019.

Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto.
A baseline for few-shot image classification.
In *International Conference on Learning Representations*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine.
Model-agnostic meta-learning for fast adaptation of deep networks.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

Spyros Gidaris and Nikos Komodakis.
Dynamic few-shot visual learning without forgetting.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang,