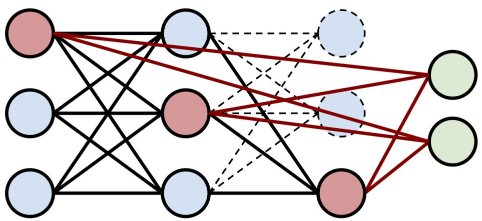
Google Research

Head2Toe

Utilizing Intermediate Representations for Better Transfer Learning

Utku Evci, Vincent Dumoulin, Hugo Larochelle, Michael C. Mozer

ICML 22'



Transfer Learning for SOTA

- Train on large *upstream* data set, fine tune on smaller *downstream* data set,
- Unsupervised / supervised pre-training is a popular recipe.
  - **Language** (BERT, GPT-3), **Vision** (CLIP, VIT), **Speech** (wav2vec), **RL**?
- Why not train downstream data set from scratch?
  - Slower convergence
  - Worse generalization

Common Transfer Learning Recipes

- **LINEAR**: Only train a new classification head
  - Cheap to run and store
  - Suboptimal performance
- **FINE-TUNING**: Pretrained feature extractor is tuned together with the head
  - High cost of running and storing for each task.
    - Mitigation strategies exist [1,2,3].
  - Better performance

Can we have best of both worlds?

1. Parameter-Efficient Transfer Learning with Diff Pruning
2. Parameter-Efficient Transfer Learning for NLP
3. Learning a Universal Template for Few-shot Dataset Generalization

Google Research

Taylor Approximation of Fine-Tuning

Loss function

Input sample

Solution after finetuning

Initial weights

$$\boxed{F}(\boxed{x};\boxed{w}^*) \approx F(x;\boxed{w}) + \sum_{i,j} \frac{\partial F(x;w)}{\partial w_{ij}} \Delta w_{ij}$$

Activation

Pre-activation

$$\approx F(x;w) + \sum_{i,j} \boxed{h_i} \frac{\partial F(x;w)}{\partial \boxed{z_j}} \Delta w_{ij}$$

$$\approx F(x;w) + \sum_{i} h_i \boxed{\sum_{j} \frac{\partial F(x;w)}{\partial z_j} \Delta w_{ij}}$$

$$\approx F(x;w) + \sum_{i} h_i \boxed{c_{i,x}}$$

Google Research

Hypothesis

Fine-tuning performance can be matched

using a linear probe on intermediate activations.

Problems with Extended Feature Set

- **Overfitting**: When #FeatureDim>>#Samples.
  - Previous work* shows that regularization helps few-shot transfer when intermediate features are used.

Method	Aggregation	5-shot	1-shot
Cls	last	76.28 ±0.41	60.09 ±0.61
	concat	75.67 ±0.41	57.15 ±0.61
	SUR	**79.25** ±0.41	**60.79** ±0.62

- **Cost**: O(#FeatureDim * #Classes) both memory and compute.
  - #FeatureDim=1m, #Classes=100: 40GB (float32)

Google Research

*Selecting Relevant Features from a Universal Representation for Few-shot Classification

The Case for Feature Selection

- **Assumption:** A small subset of features is enough to achieve good generalization (and less likely to overfit when trained).
- **Implication:** Inference cost is now O(*#FeatureKeptDim * #Classes*).

Head2Toe (H2T) w/ Group-Lasso

- Given a pretrained NN:

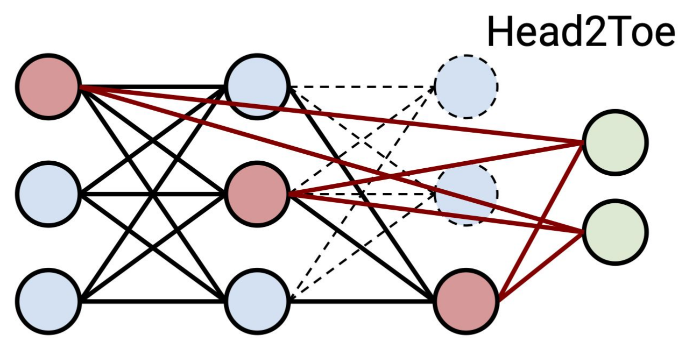
$$\boldsymbol{z}_\ell = \boldsymbol{h}_{\ell-1}\boldsymbol{W}_\ell \quad ; \quad \boldsymbol{h}_\ell = f(\boldsymbol{z}_\ell)$$

$$\boldsymbol{z}'_L = \boldsymbol{h}_{all}\boldsymbol{W}_{all} \quad ; \quad \boldsymbol{h}_{all} = \mathrm{concat}(a_1(\boldsymbol{h}_1), a_2(\boldsymbol{h}_2), ..., a_L(\boldsymbol{h}_L))$$

- Train $\boldsymbol{W}_{all}$ with group-lasso and select features with highest l2-norm.

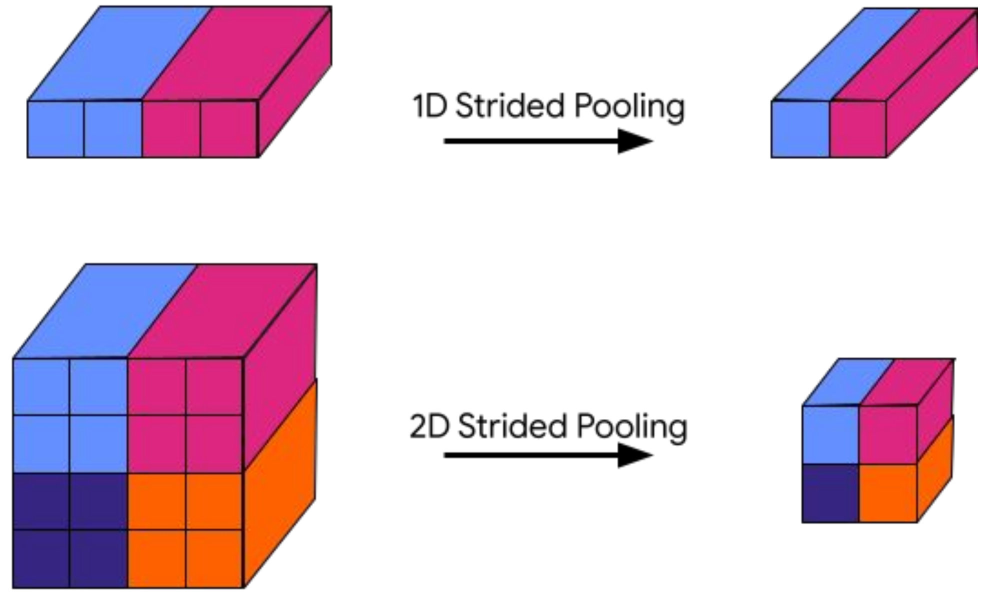
$$|\boldsymbol{W}|_{2,1} = |\boldsymbol{s}|_1 = \sum_i |s_i| \quad ; \quad s_i = \sqrt{\sum_j w_{ij}^2}$$

- After calculating the scores, keep a fraction *f* of features and train a linear classifier on the selected features.



Selection of Intermediate Features

- Strided pooling to aggregate features.
- Pool size is selected per layer s.t. there are ~T features per layer.
- Flatten and normalize features from each layer to unit-norm.



Experimental Setup

- VTAB-1k benchmark: 19 image classification tasks with 1000 training samples each.
  - Natural: natural images
  - Structured: rendered artificial images
  - Specialized: images from non-standard cameras
- Hyper-parameter selection / Validation
  - 5-fold cross validation for <u>each method</u> and <u>transfer task</u> separately.
    - 2 learning rates
    - 2 training steps
    - 3 regularization coefficients (2 for Head2Toe)
    - 3 target feature size
- 3 seeds per task

Results on ResNet-50

- We match/exceed the fine-tuning results reported in the VTAB paper*.

	Natural							Specialized				Structured								
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	Mean
Linear	48.5	86.0	67.8	84.8	87.4	47.5	34.4	**83.2**	92.4	73.3	73.6	39.7	39.9	36.0	66.4	40.4	37.0	19.6	25.5	57.0
+All-ℓ_2	44.7	87.0	67.8	84.2	86.1	81.1	31.9	82.6	**95.0**	**76.5**	74.5	50.0	56.3	38.3	65.5	59.7	44.5	37.5	40.0	63.3
+All-ℓ_1	**50.8**	88.6	67.4	84.2	87.7	84.2	34.6	80.9	94.9	75.6	**74.7**	49.9	57.0	41.8	72.9	59.0	44.8	37.5	40.8	64.6
+All-$\ell_{2,1}$	49.1	86.7	**68.5**	84.2	**88.0**	**84.4**	**34.8**	81.5	94.9	75.7	74.3	48.3	58.4	42.0	**74.4**	58.8	45.2	37.8	34.4	64.3
Head2Toe	47.1	**88.8**	67.6	**85.6**	87.6	84.1	32.9	82.1	94.3	76.0	74.1	**55.3**	**59.5**	**43.9**	72.3	**64.9**	**51.1**	**39.6**	**43.1**	**65.8**
Scratch*	11.0	37.7	23.0	40.2	13.3	59.3	3.9	73.5	84.8	41.6	63.1	38.5	54.8	35.8	36.9	87.9	37.3	20.9	36.9	42.1
Fine-tuning	33.2	84.6	54.5	**85.2**	79.1	**87.8**	16.6	82.0	92.5	73.3	73.5	54.6	63.7	**46.3**	72.1	**94.8**	47.1	35.0	33.3	63.6

Results on ViT-B/16

- Similarly, Head2Toe matches fine-tuning. +5% if the backbone has option to be tuned.

	Natural							Specialized				Structured								
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	Mean
Linear	55.0	81.0	53.6	72.1	85.3	38.7	32.3	80.1	90.8	67.2	74.0	38.5	36.2	33.5	55.7	34.0	31.3	18.2	26.3	52.8
+All-ℓ_2	57.3	87.0	64.3	82.8	84.0	75.7	32.4	82.0	94.7	79.7	**74.8**	47.4	57.8	41.4	62.8	46.6	33.3	31.0	38.8	61.8
+All-ℓ_1	58.4	**87.3**	**64.9**	83.3	84.6	80.0	34.4	**82.3**	**95.6**	79.6	73.6	47.9	57.7	**42.2**	**65.1**	44.5	33.4	32.4	38.4	62.4
+All (Group)	**59.6**	87.1	64.9	85.2	**85.4**	79.5	**35.3**	82.0	95.3	**80.6**	74.2	47.9	57.8	40.7	64.9	46.7	**33.6**	31.9	39.0	62.7
Head2Toe	58.2	87.3	64.5	**85.9**	85.4	**82.9**	35.1	81.2	95.0	79.9	74.1	**49.3**	**58.4**	41.6	64.4	**53.3**	32.9	**33.5**	**39.4**	**63.3**
Scratch	7.6	19.1	13.1	29.6	6.7	19.4	2.3	71.0	71.0	29.3	72.0	31.6	52.5	27.2	39.1	66.1	29.7	11.7	24.1	32.8
Fine-tuning	44.3	84.5	54.1	84.7	74.7	**87.2**	26.9	**85.3**	95.0	76.0	70.4	**71.5**	60.5	46.9	**72.9**	74.5	38.7	28.5	23.8	63.2
Head2Toe-FT	43.9	82.3	53.5	**84.9**	76.7	86.5	24.5	79.9	95.9	77.5	**74.3**	68.0	70.9	**48.2**	72.4	76.1	44.8	32.1	42.5	65.0
Head2Toe-FT+	**57.3**	**87.1**	**63.8**	83.7	**84.8**	86.8	**35.1**	80.2	**96.1**	**79.9**	74.1	69.9	**71.2**	47.8	72.8	**77.4**	**45.9**	**33.9**	**43.0**	**67.9**

Cost of Head2Toe

- **FLOPs cost of H2T consists of three parts:**
  a. Calculating the representations for all data (fixed)
  b. Training W_{all} (~#FeatureDim * #Classes)
  c. Validating different fractions: ~18% of **(b)**.
- **Storage size of H2T depends on #FeaturesSelected and the bitmap.**

Dataset	F	N	C	FLOPs (vs FINETUNING)	Size (vs FINETUNING)	Size (vs LINEAR)
Caltech101	0.010	467688	102	0.009675	0.020750	2.353167
CIFAR-100	0.200	30440	100	0.005792	0.025743	2.977301
Clevr-Dist	0.001	467688	6	0.005747	0.000741	1.417419
Clevr-Count	0.005	30440	8	0.000568	0.000092	0.132278
Retinopathy	0.200	467688	5	0.005657	0.020531	47.099634
DMLab	0.020	467688	6	0.005747	0.003011	5.756287
dSpr-Orient	0.200	30440	16	0.005302	0.004183	3.001686
dSpr-Loc	0.005	467688	16	0.006644	0.002212	[illegible]
DTD	0.005	1696552	47	0.015823	0.019157	[illegible]
EuroSAT	0.100	30440	10	0.005267	0.001336	[illegible]
KITTI-Dist	0.020	467688	4	0.005567	0.002215	[illegible]
Flowers102	0.100	30440	102	0.001117	0.013146	[illegible]
Pets	0.002	467688	37	0.003842	[illegible]	[illegible]
Camelyon	0.020	30440	2	0.005220	[illegible]	[illegible]
Resisc45	0.020	467688	45	0.009247	[illegible]	[illegible]
sNORB-Azim	0.002	1696552	18	0.011069	[illegible]	[illegible]
sNORB-Elev	0.050	467688	9	0.006016	[illegible]	[illegible]
Sun397	0.100	30440	397	[illegible]	[illegible]	[illegible]
SVHN	0.005	1696552	10	[illegible]	[illegible]	[illegible]
Average				0.006295	0.010729	5.674742

FLOPs (vs FINETUNING)	Size (vs FINETUNING)	Size (vs LINEAR)
0.006295	0.010729	5.674742

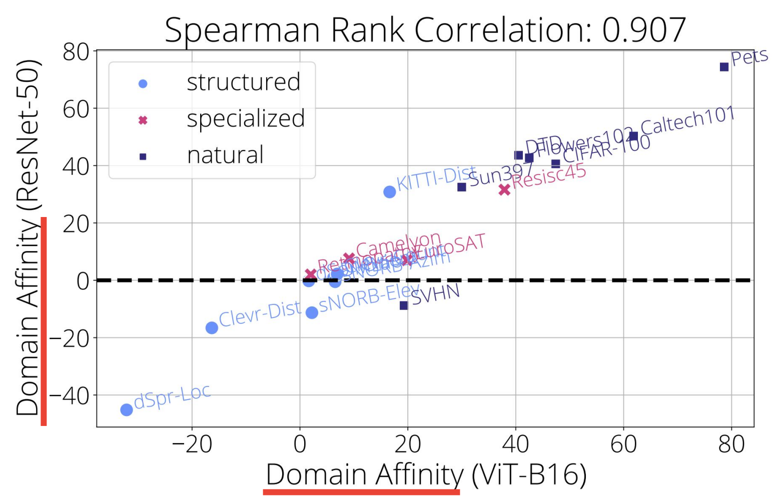
Google Research

Defining a Metric for Task/Domain Affinity

- **Assumption:** If a downstream task is similar to the upstream dataset, it will achieve better linear performance in a data-limited setting.

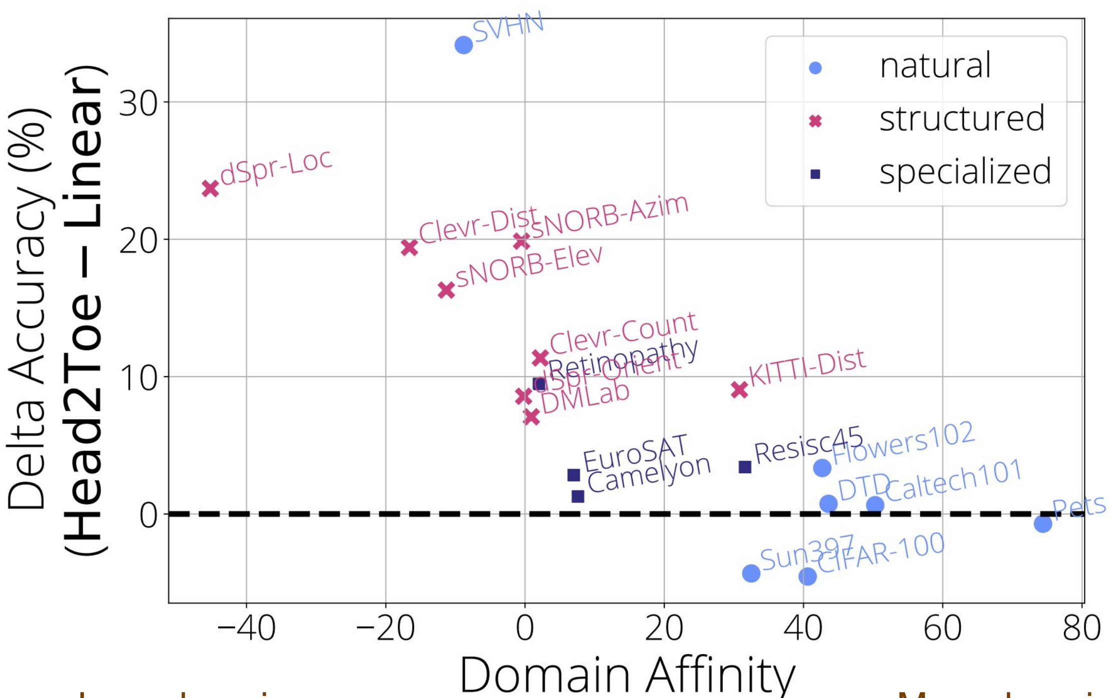
$$\mathit{DomainAffinity} = \mathrm{Acc}_{\mathrm{LINEAR}} - \mathrm{Acc}_{\mathrm{SCRATCH}}$$

- This metric is robust to different backbones and algorithms used to train it.



Google Research

Head2Toe Improves OOD Generalization



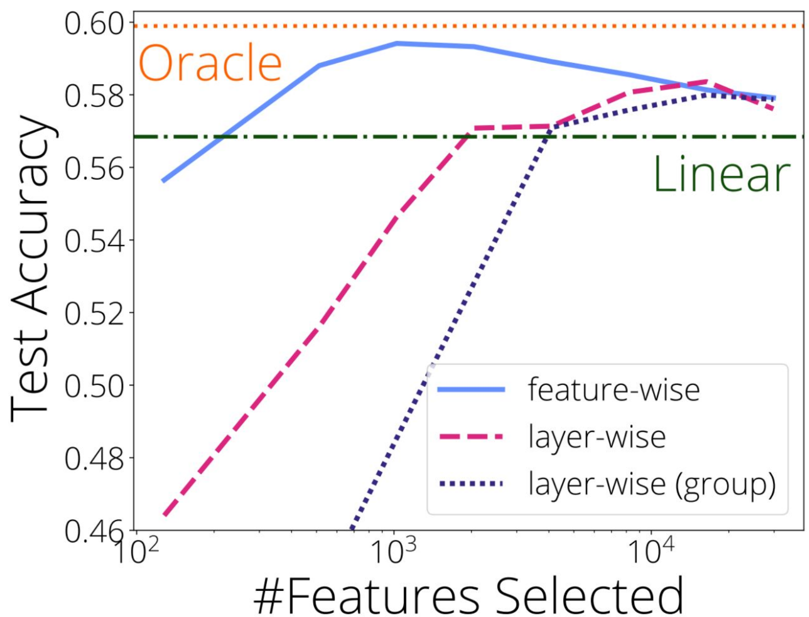
Less domain similiarity

More domain similarity

Google Research

Head2Toe - Layers vs. Features

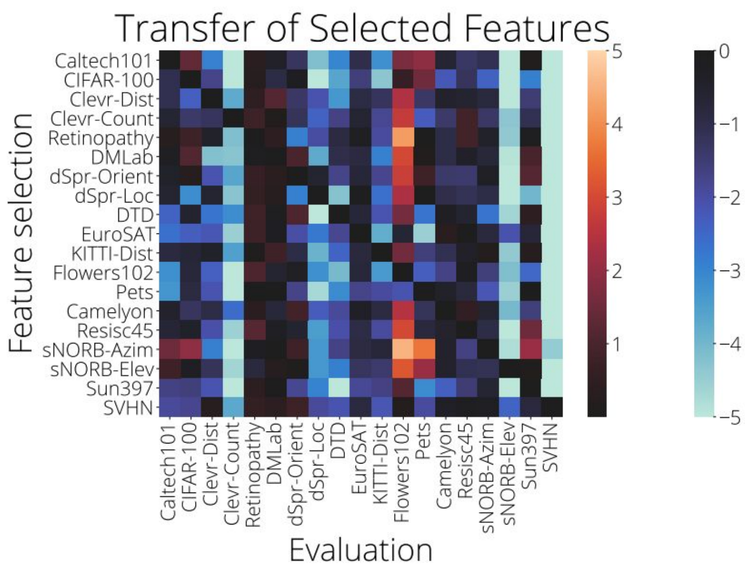
- What if we select layers instead of individual features?
  - Feature selection works better.

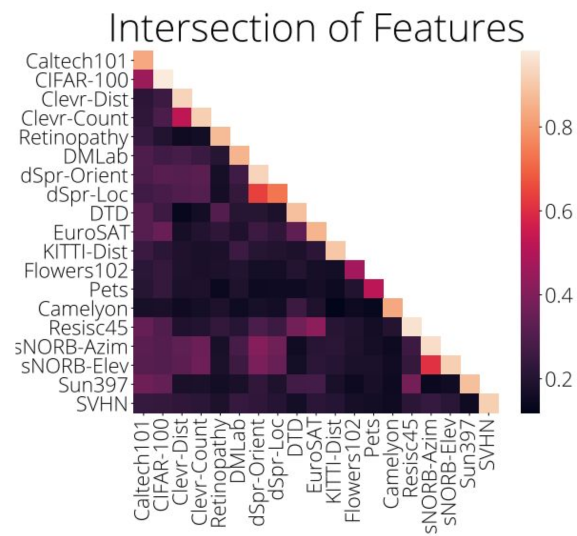


Google Research

Importance of Dynamic Adaptation

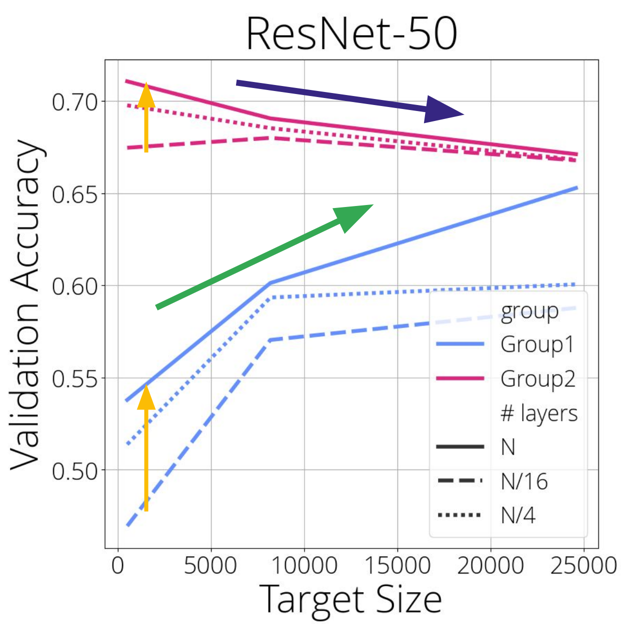
- No single set of features perform best over all tasks.
- Features have <20% intersection.

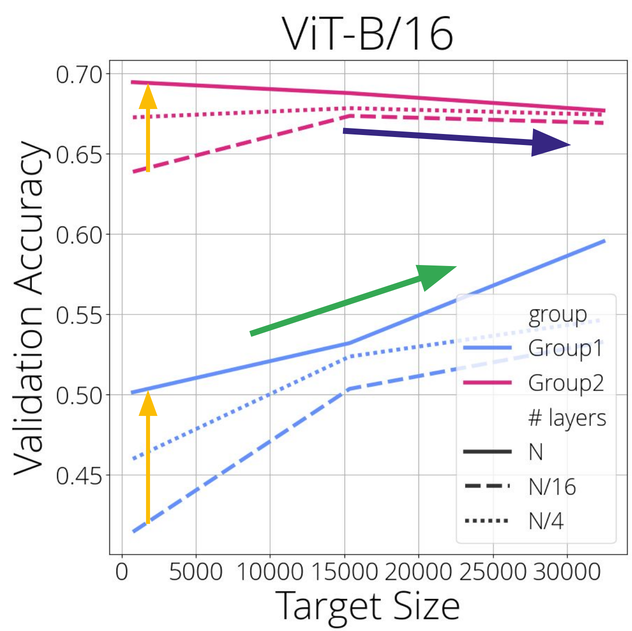




Bitter Lesson*

- Utilizing more layers always improves performance.
- Using more features per layer (smaller pooling size) is useful only a subset of tasks (Group-1).





Group1	Group2
DMLab	CIFAR-100
DTD	Clevr-Count
sNORB-Azim	dSpr-Orient
SVHN	Retinopathy
dSpr-Loc	Resisc45
Pets	EuroSAT
sNORB-Elev	Flowers102
	Camelyon
	Caltech101
	Clevr-Dist
	KITTI-Dist

*incompleteideas.net/IncIdeas/BitterLesson.html

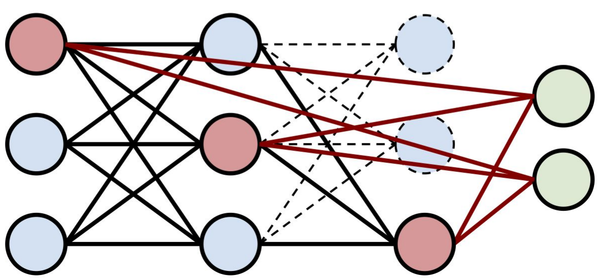
Google Research

Future Research

- Scaling # candidate features further up.
  - Bigger and multiple backbones.
  - Better/cheaper/simpler feature selection algorithms.
  - Better/simpler feature aggregation functions.
- Applying Head2Toe to different domains.

Head2Toe Summary

- Finetuning performance can be matched or exceeded with a special linear probe on intermediate features.
- This strategy helps most on far transfer tasks.
- Extracting features from more layers and features help.
- Select features for each task separately.



Thank you for listening!

Google Research