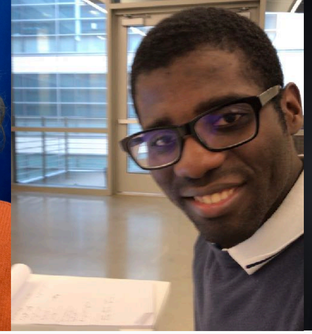

Matteo Rovere


Giovanni Paolini


Glen Mbeng


Stefano Soatto

Information in the Weights and Emergent Properties of Deep Neural Networks

Alessandro Achille
University of California, Los Angeles

Menu

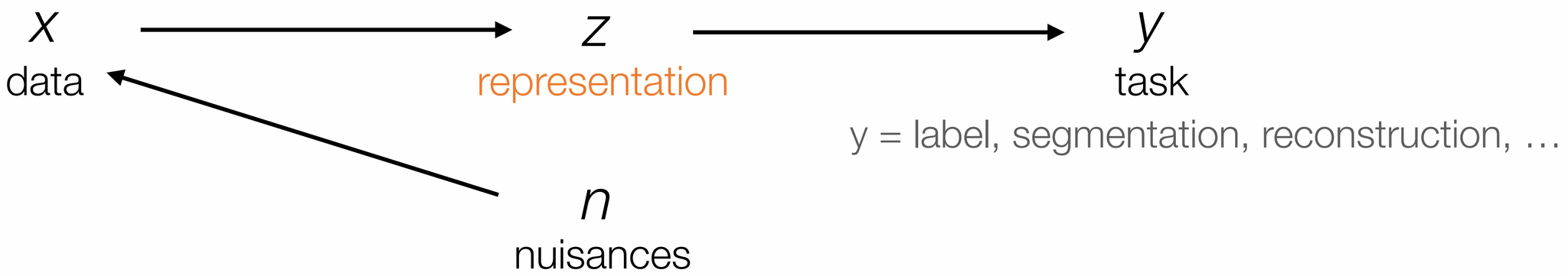
Two representations are important in machine learning:

- The representation of the test image (activations)
- The representation of the train dataset (weights)

In a DNN, the optimality of one implies optimality of the other (Emergence Bound)

Information gives an asymmetric distance distance on the space of learning tasks, which allows prediction of transfer learning.

What is an optimal representation (of test data)?



Sufficient	$I(z; y) = I(x; y)$
Nuisance invariance	$n \perp\!\!\!\perp y \Rightarrow I(n; z) = 0$
Minimal	$I(x; z)$ = minimal
Compositional	Minimal component correlation?

A. and Soatto, *Information Dropout: finding optimal representation through noisy computation*, PAMI 2017

A Variational Principle for representation learning: The Information Bottleneck principle

A minimal sufficient representation ***z*** of the data ***x*** for the task ***y*** is the solution to:

$$\begin{aligned} \text{minimize}_{p(z|x)} \quad & I(z;x) \\ \text{s.t.} \quad & I(z;y) = I(x;y) \end{aligned}$$

Information Bottleneck Lagrangian: (Tishby et al., 1999)

$$\mathscr{L}(p(z\,|\,x)) = H(y\,|\,z) + \beta\, I(z;x)$$

cross-entropy regularizer

Invariant if and only if minimal

Recall. A representation *z* is minimal for the task *y* if it minimizes *I(z; x)* among the sufficient representations.

Theorem (A., Soatto) Let *z* be a sufficient representation and *n* a nuisance. Then,

$$I(z;n) \leq I(z;x) - I(x;y)$$

invariance minimality constant

Moreover, there exists a nuisance *n* for which equality holds.

Corollary: A representation is maximally invariant if and only if it is minimal

A. and Soatto, *Emergence of Invariance and Disentanglement in Deep Representations*, JMLR 2018

The catch

What if we just represent an image by its index in the training set (or by a unique hash)?

x	z	y
24,576 bits	*16 bits*	*4 bits*
	0000000000000000	0100
	0000000000000001	0001
	0000000000000010	0010
	0000000000000011	0101

It is a sufficient representation and it is close to minimal.

This Information Bottleneck is wishful thinking

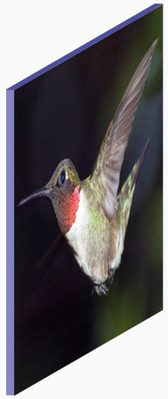
The IB is a statement of desire for future data we do not have:

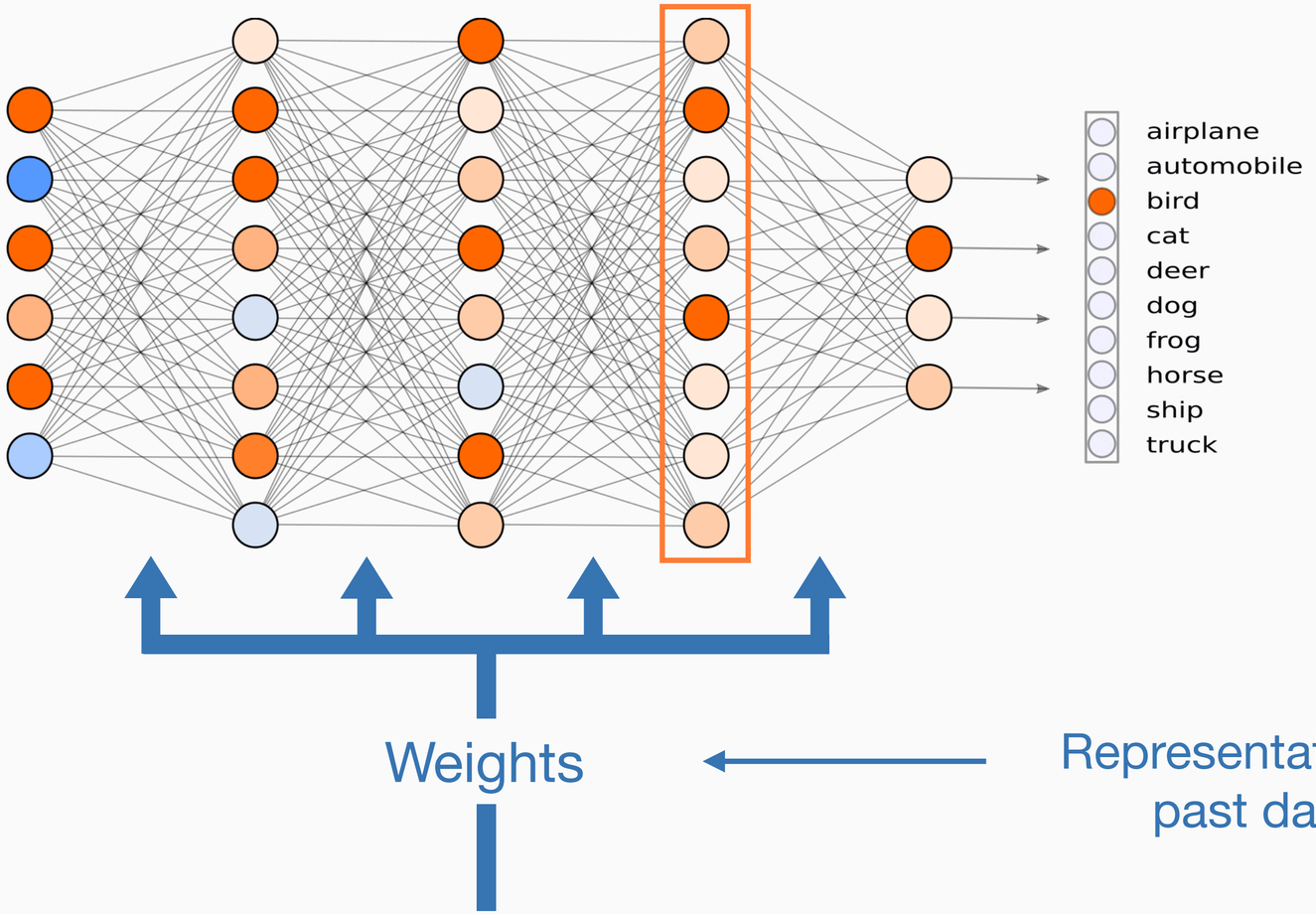
$$\min_{q(z|x)} \mathcal{L} = H_{p,q}(y|z) + \beta\, I(z;x)$$

What we have is the data collected in the past.

What is the best way to use the past data in view of future tasks?

Test Image





Weights

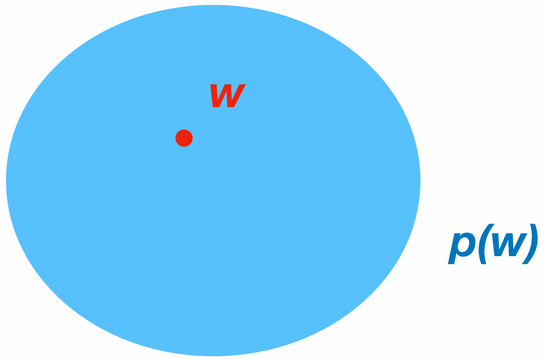
Representation of past data

Training Set

{  , (car, horse, deer, …) }

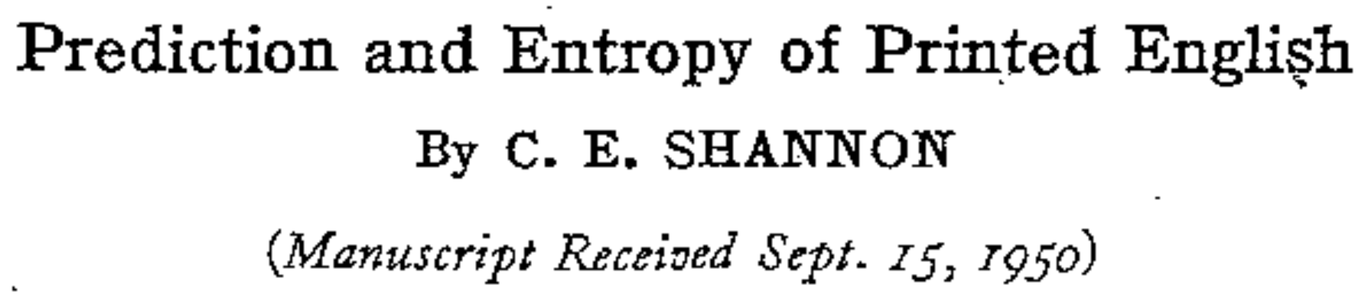
How do we measure the information in the weights of a DNN?

Problem. Assume some prior distribution $p(w)$ over the weights. Codifying a particular set of weights as real numbers requires *infinite information*.



Example: Measuring Information by Adding Noise

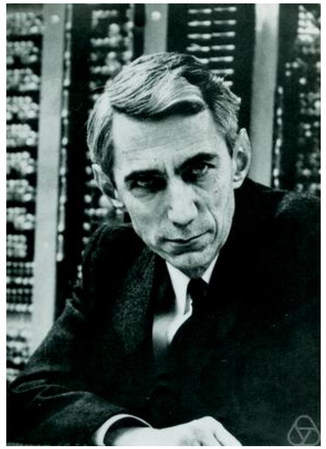
Estimate the amount of information by adding noise and measuring the decrease in performance.



Prediction and Entropy of Printed English

By C. E. SHANNON

(*Manuscript Received Sept. 15, 1950*)



Example: Shannon (1951) estimates the information content of the English language by corrupting random letters and measuring the reconstruction error of English speakers.

"Thif is a vevy moisy party" → "This is a very noisy party"

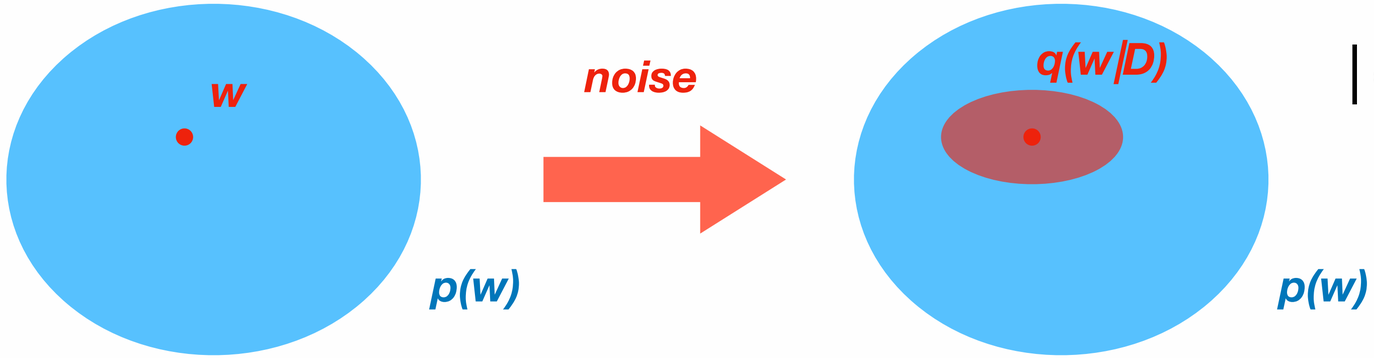
Shannon, *Prediction and Entropy of Printed English*, Bell System Technical Journal*, 1951*

Reducing the description length by adding noise

Keeping Neural Networks Simple by Minimizing the Description Length of the Weights

Geoffrey E. Hinton and Drew van Camp
Department of Computer Science
University of Toronto

Idea: Add noise to the weights to encode with a finite amount of information



$$|\,\text{encoding}\,| = \mathrm{KL}(\, q(w \mid \mathscr{D}) \,\|\, p(w)\,)$$

The Information in the Weights

We want to measure the trade-off between the amount KL(*q(w | D)* || *p(w)*) of noise added and the accuracy of the network.

$$S(t) = \min \; \mathrm{KL}\left(q(w \mid \mathcal{D}) \,\|\, p(w) \right)$$

Minimum information in weights…

$$\text{s.t.} \;\; \mathbb{E}_{w \sim q(w|\mathcal{D})}[L_{\mathcal{D}}(w)] < t$$

… such that the expected training loss is less than t

Or, equivalently the Lagrangian:

Expected loss — Information in the Weights

$$\mathcal{L} = \mathbb{E}_{w \sim q(w|\mathcal{D})}[L_{\mathcal{D}}(w)] + \beta \, \mathrm{KL}(\underbrace{q(w|\mathcal{D})}_{\text{optimal noise}} \,\|\, \underbrace{p(w)}_{\text{fixed prior}})$$

A., Paolini, Soatto, Information Complexity of Learning Tasks, 2019
A., and Soatto, Where is the Information in a DNN?, 2019

Grounding: The PAC-Bayes generalization bound

PAC-Bayes bound on the test error: (Catoni, 2007; McAllester 2013)

$$L_{\text{test}} \leq \frac{1}{1-\frac{1}{2\beta}}\left[\mathbb{E}_w[L_{\mathcal{D}}(w)] + \beta\,\mathrm{KL}(q(w|\mathcal{D})\,\|\,p(w))\right]$$

What matters for generalization is not the number of weights, but the information they contain.

This gives non-vacuous generalization bounds.

A. and Soatto, Emergence of Invariance and Disentanglement in Deep Representations, JMLR 2018
Dziugaite and Roy, Computing non-vacuous generalization bounds for deep neural networks, UAI 2017

The sharpest PAC-Bayes bound is the IB Lagrangian

Which encoding prior p(w) gives the tightest PAC-Bayes bound?

Proposition. On expectation over the sampling of the dataset $\mathscr{D}$, the sharpest PAC-Bayes bound is obtained when $p(w) = \mathbb{E}_{\mathscr{D}}[q(w \mid \mathscr{D})]$, in which case $\mathbb{E}_{\mathscr{D}}[\mathrm{KL}(q(w \mid \mathscr{D}) \| p(w))] = I(w; \mathscr{D})$.

The Weight Lagrangian then becomes:

$$\mathscr{L}(q(w \mid \mathscr{D}) = \mathbb{E}_{\mathscr{D}}[H(\mathscr{D} \mid w)] + \beta I(w; \mathscr{D})$$

IB Lagrangian for the weights

A. and Soatto, Emergence of Invariance and Disentanglement in Deep Representations, JMLR 2018
McAllester, A PAC-Bayes Tutorial, 2013

The Information in a Deep Neural Network

$$L(w) = H_{p,q}(\mathcal{D}|w) + \beta\,\mathrm{KL}(\underbrace{q(w|\mathcal{D})}_{\text{output of training}} \,\|\, \underbrace{p(w)}_{\text{encoding prior}})$$

Shannon Information: using the encoding prior $p(w) := \mathbb{E}_{\mathscr{D}}[q(w \mid \mathscr{D})]$

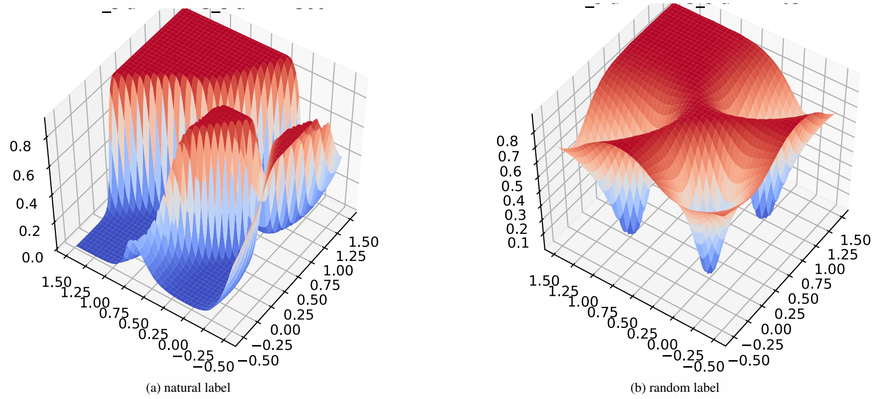
$$\mathbb{E}_{\mathscr{D}}[\mathrm{KL}] = I(w; \mathscr{D})$$

⇒ Minimizes the PAC-Bayes generalization bound

Fisher Information: *p(w)* = uniform encoding prior

$$\mathrm{KL} = -\log|F|$$

⇒ Implicitly minimized by SGD



A., Paolini, Soatto, *The Information Complexity of Learning Tasks, their Structure and their Distance,* ArXiv 2019
Hochreiter and Schmidhuber, *Flat Minima*, Neural Computation 1997
Pictures from Zhang et al, Theory of Deep Learning III: Generalization Properties of SGD, 2017

SGD connects Fisher and Shannon

SGD minimizes the Fisher Information of the Weights. However, generalization is governed by the Shannon Information.

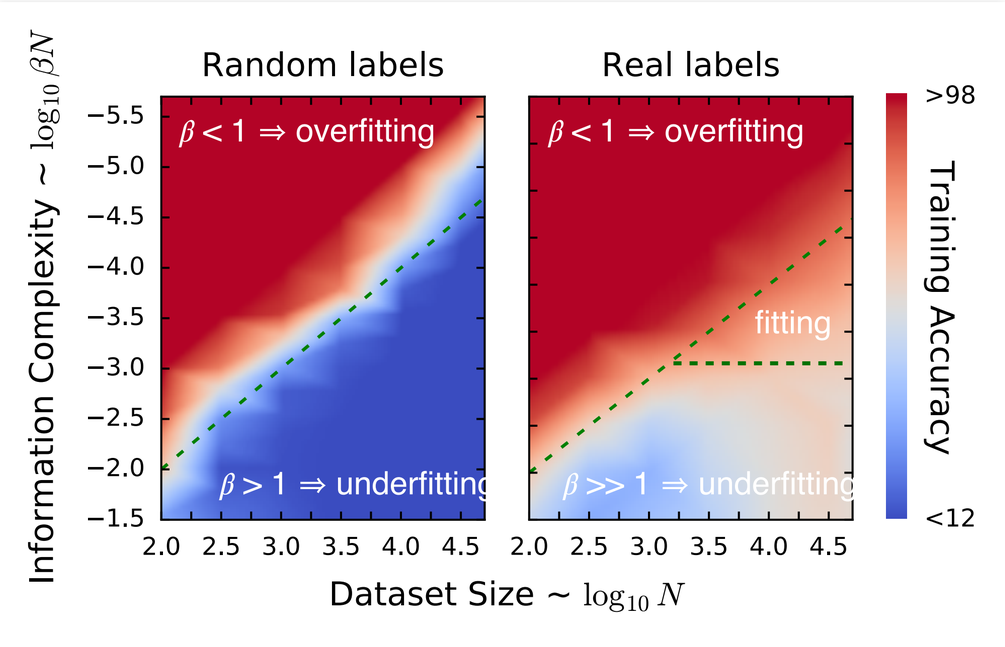
Proposition. Assuming the dataset is parametrized in a differentiable way, we have:

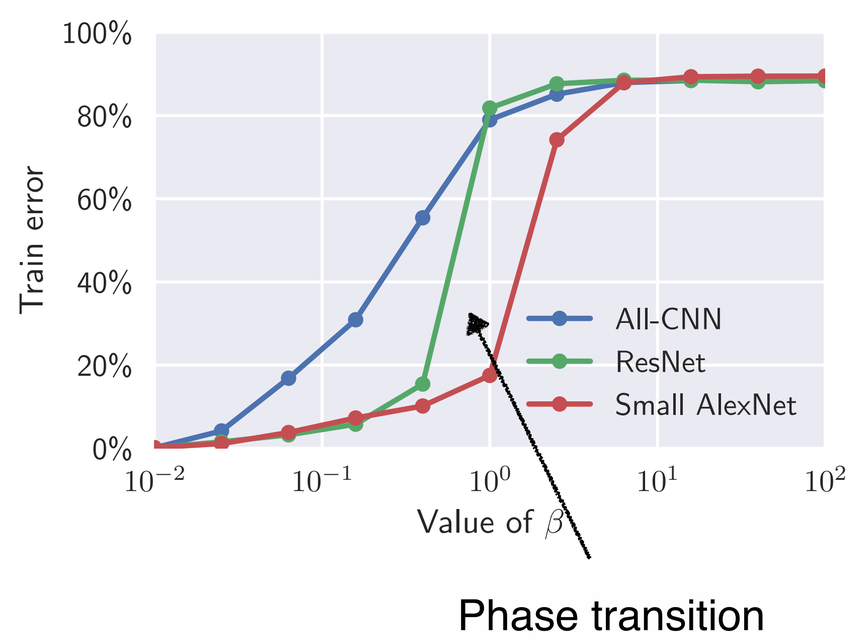
$$\underbrace{I(w;\mathscr{D})}_{\text{Shannon Information}} \approx H(\mathscr{D}) - \mathbb{E}\left[\log\left(\frac{(2\pi e)^k}{|\nabla_{\mathscr{D}} w^* \underbrace{F(w^*)}_{\text{Fisher Information}} \nabla_{\mathscr{D}} w^{*T}|}\right)\right]$$

Where $w^* = w^*(D)$ is the result of running SGD on dataset D and F(w) is the Fisher Information Matrix in w.

A. and Soatto, Where is the Information in a Deep Network?, 2019
Brunel and Nadal. Mutual information, fisher information, and population coding, 1998

Empirical verification: Phase transition





Phase transition

Using the regularized loss:

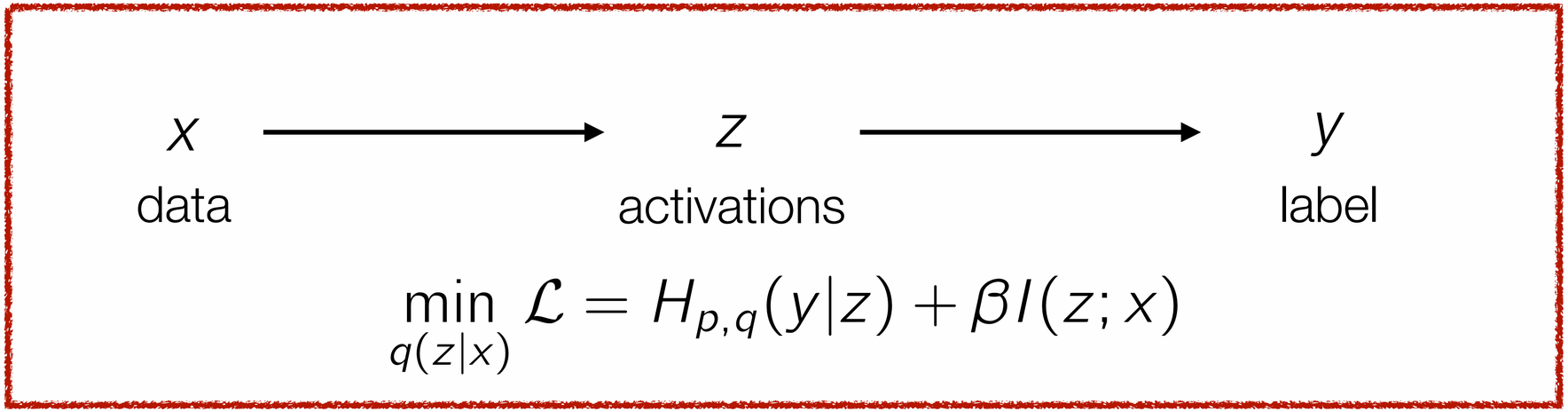
$$L(w) = H_{p,q}(\mathcal{D}|w) + \beta\, KL(q(w|\mathcal{D})||p(w))$$

For random labels there is a transition between over- and under-fitting at $\beta = 1$.

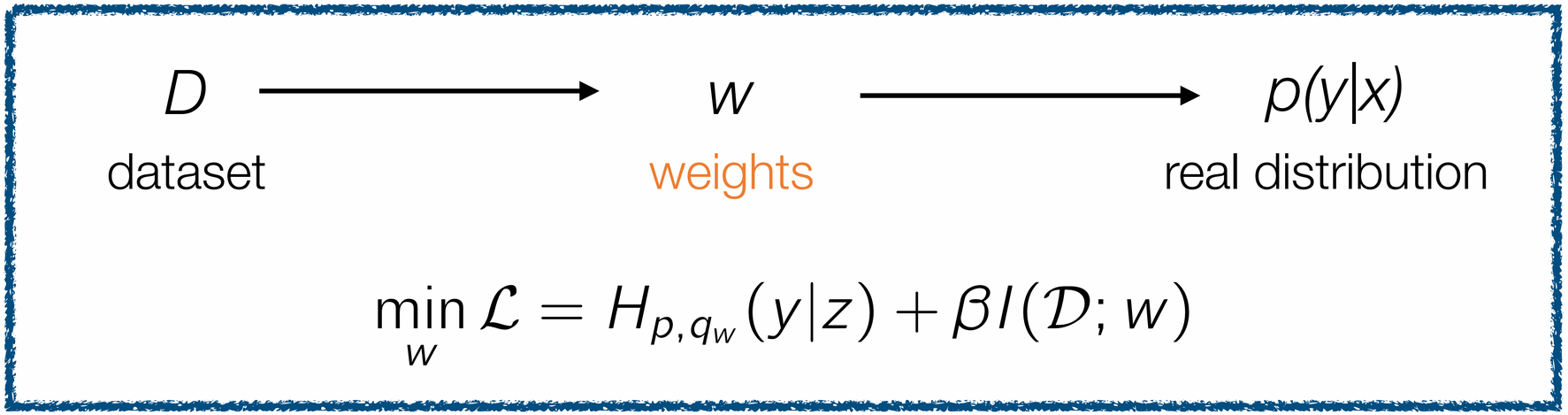
Achille and Soatto, Emergence of Invariance and Disentanglement in Deep Representations, JMLR 2018

Two Bottlenecks

Activations IB
Invariance



Weights IB
Generalization



The Emergence Bound: connecting the two bottlenecks

Let $z = f_w(x)$ be a layer of a network, and let z_n be the representation obtained by adding noise to the weigths. We define the effective information as $I_{eff}(x; z) = I(x; z_n)$

Proposition. To the first order the information in the activations is given by:

$$I_{\mathrm{eff}}(x;z) \approx H(x) - \log\left(\frac{(2\pi e)^k}{|\nabla_x f_w(x)^t \; J_f^t \; F(w) \; J_f \; \nabla_x f_w(x)|}\right)$$

Information in activations

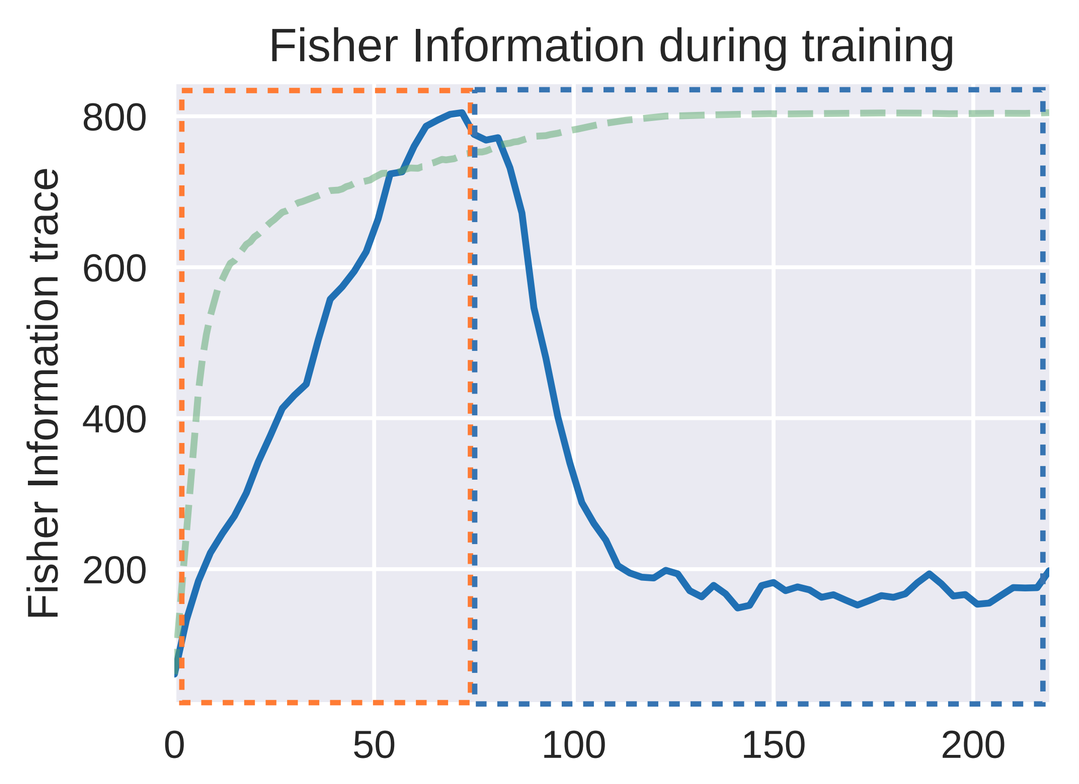
Fisher Information in weights

Take-away: Reducing information in the weights reduces information in the activations, hence it promotes invariant classifiers.

A. and Soatto, *Emergence of Invariance and Disentanglement in Deep Representations*, JMLR 2018
A. and Soatto, *Where is the Information in a Deep Network?*, 2019

Compression of weights leads to compression of activations

We empirically observe that the Fisher Information decreases later during training.

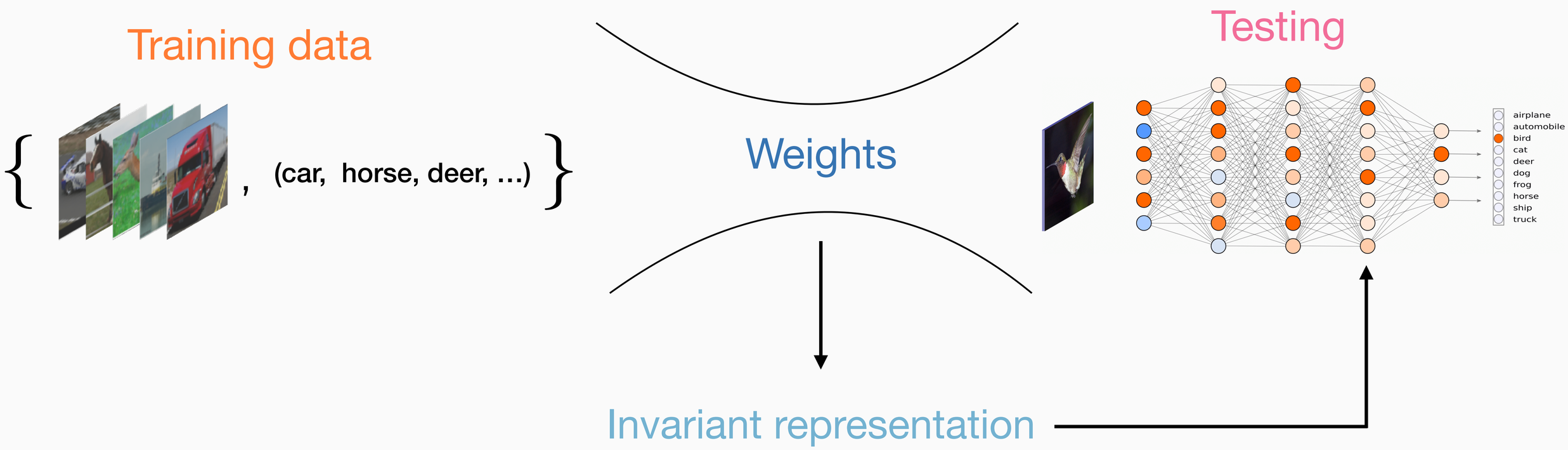


The (Fisher) information in the weights decreases later in training

The effective information in the activations decreases

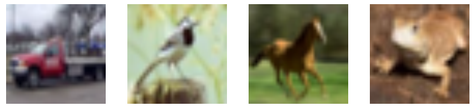
A., Rovere, Soatto, Critical Learning Periods in Deep Networks, 2018

Compression of the **weights** biases toward invariant and disentangled **representations**.



PAST	FUTURE
Weights	Activations
Generalization (PAC-Bayes)	Invariance (Emergence)
Minimality (Shannon)	Minimality (Fisher)

What is the distance between two tasks?



CIFAR-10

ImageNet

KITTI

SVHN

MNIST

Fashion MNIST

A Topology on the Space of Tasks

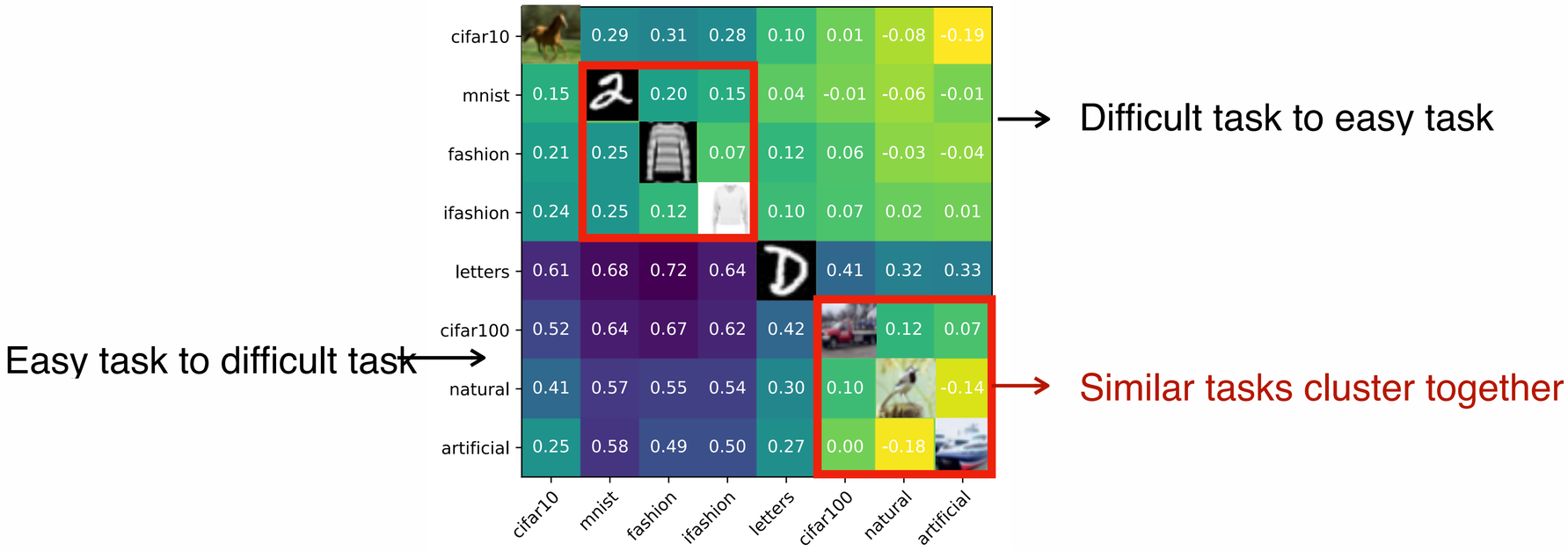
Distance between tasks:

$$d(\mathscr{D}_1 \to \mathscr{D}_2) = I(\mathscr{D}_1\mathscr{D}_2; w) - I(\mathscr{D}_1; w)$$

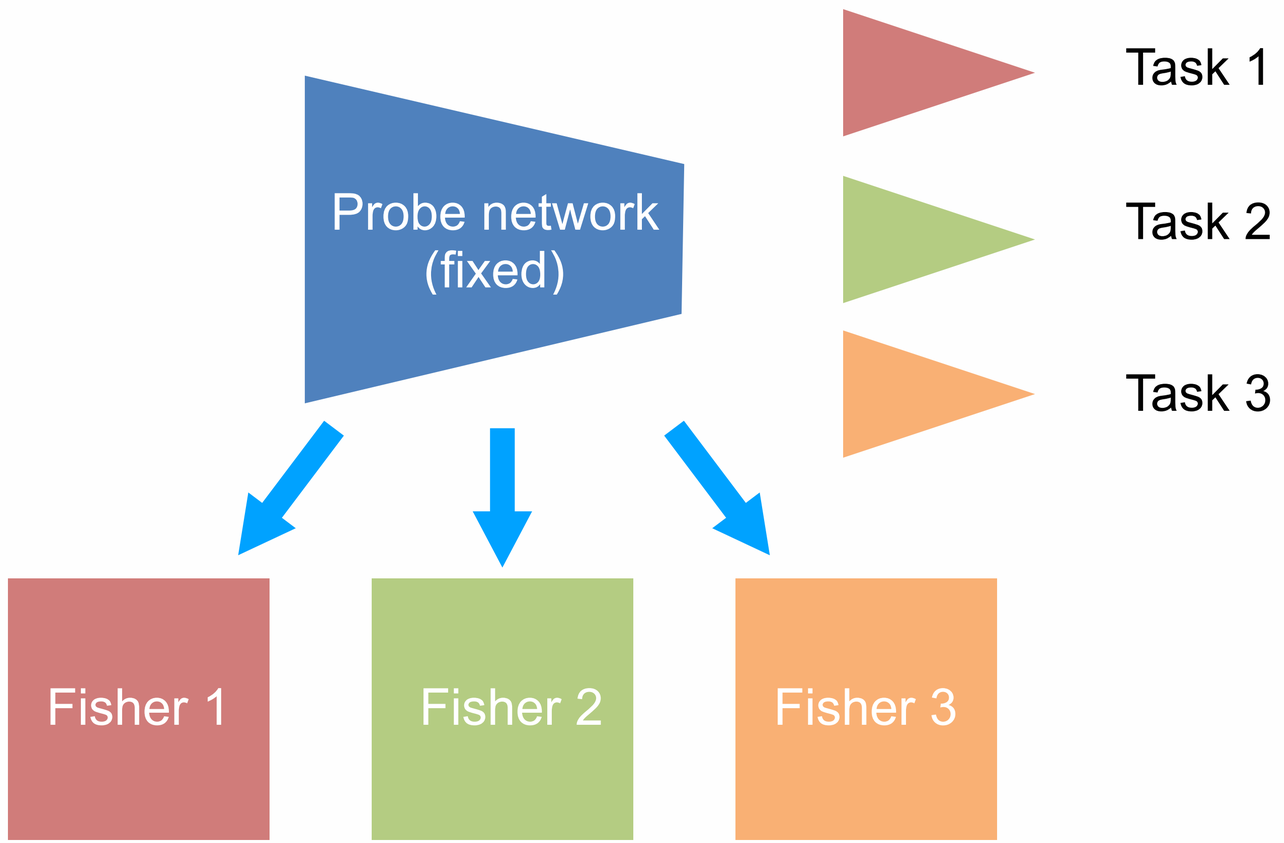
Complexity of learning together

Complexity of learning one

That is, how much more information do we need to learn?



Scaling this up: TASK2VEC



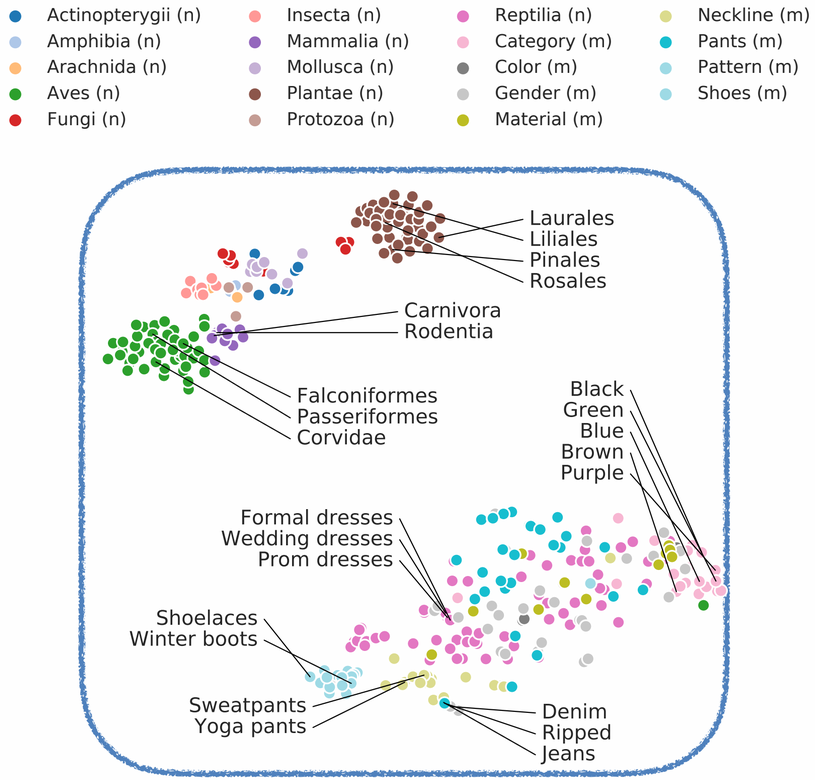
We compute the Fisher Information for the task using a pretrained set of weights and measure the distance between the compute the cosine distance between the Fisher:

$$d(\mathscr{D}_i, \mathscr{D}_j) := \cos(F_i, F_j)$$

A. et al., TASK2VEC: Task embedding for meta-learning, 2019

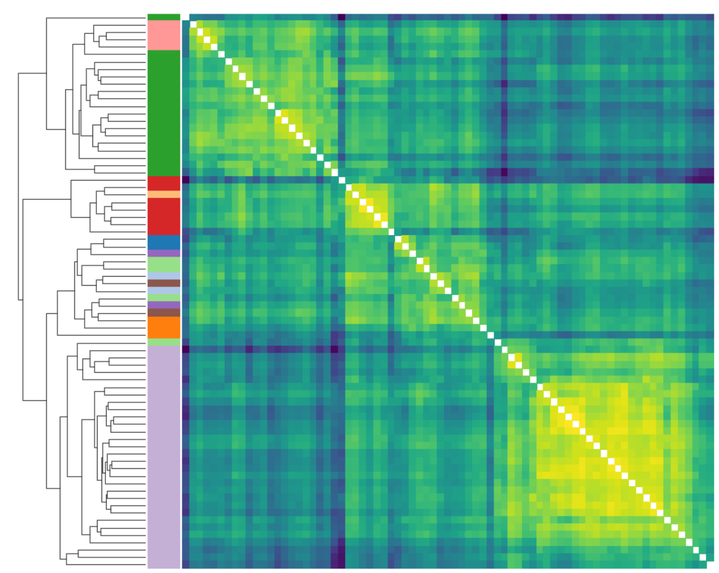
TASK2VEC: Embedding tasks in a metric space

Idea: Represent each tasks in a metric space using its Fisher Information Matrix diagonal.



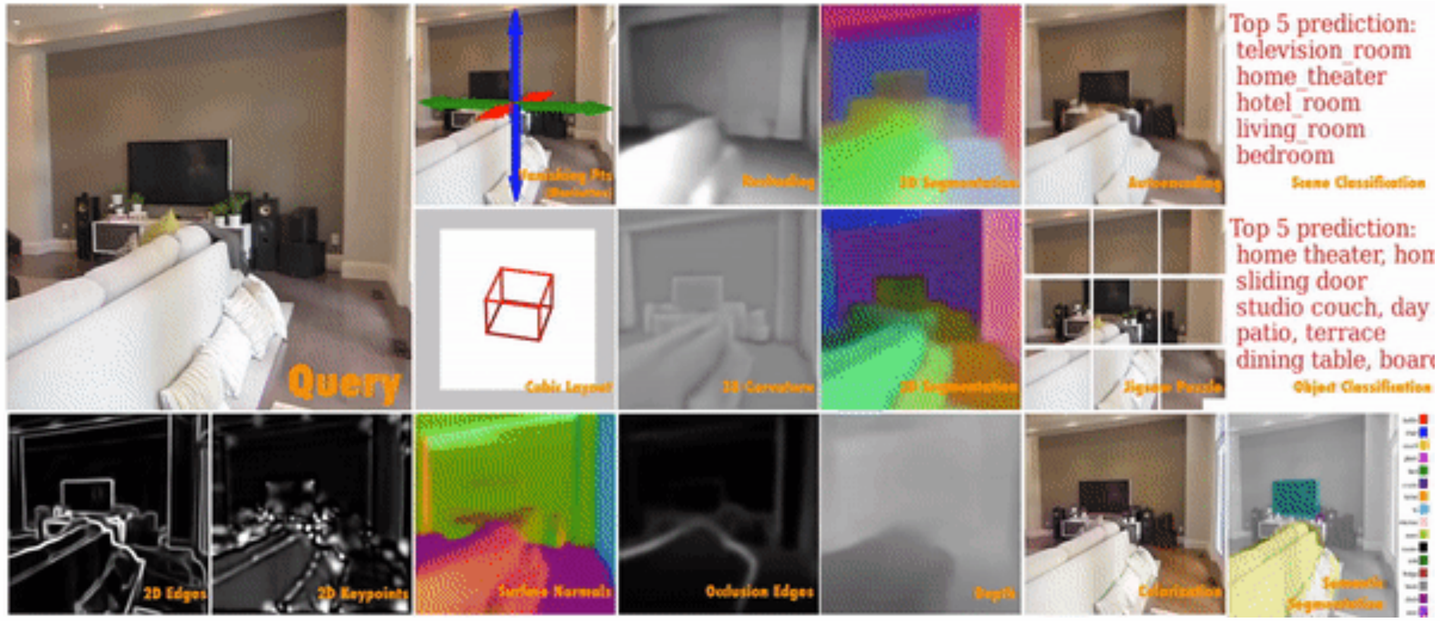
Recovers a meaningful topology on hundred of tasks

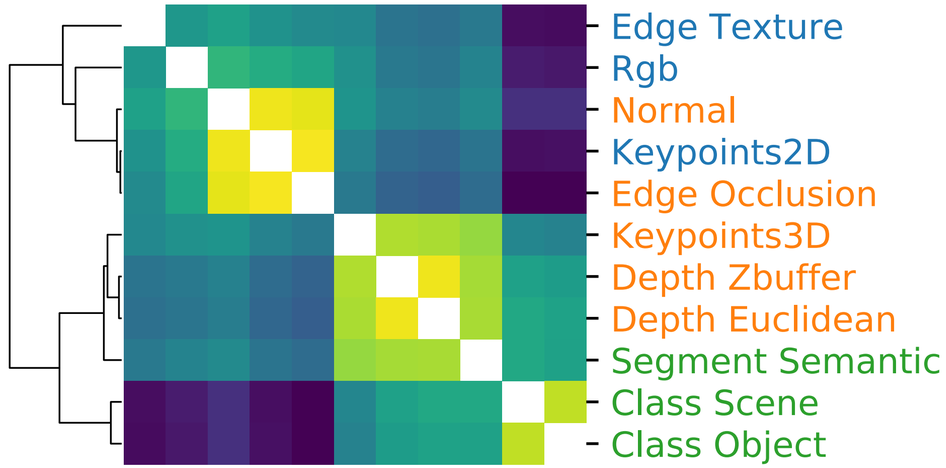
Recovers species taxonomy on iNaturalist



A. et al., TASK2VEC: Task embedding for meta-learning, 2019

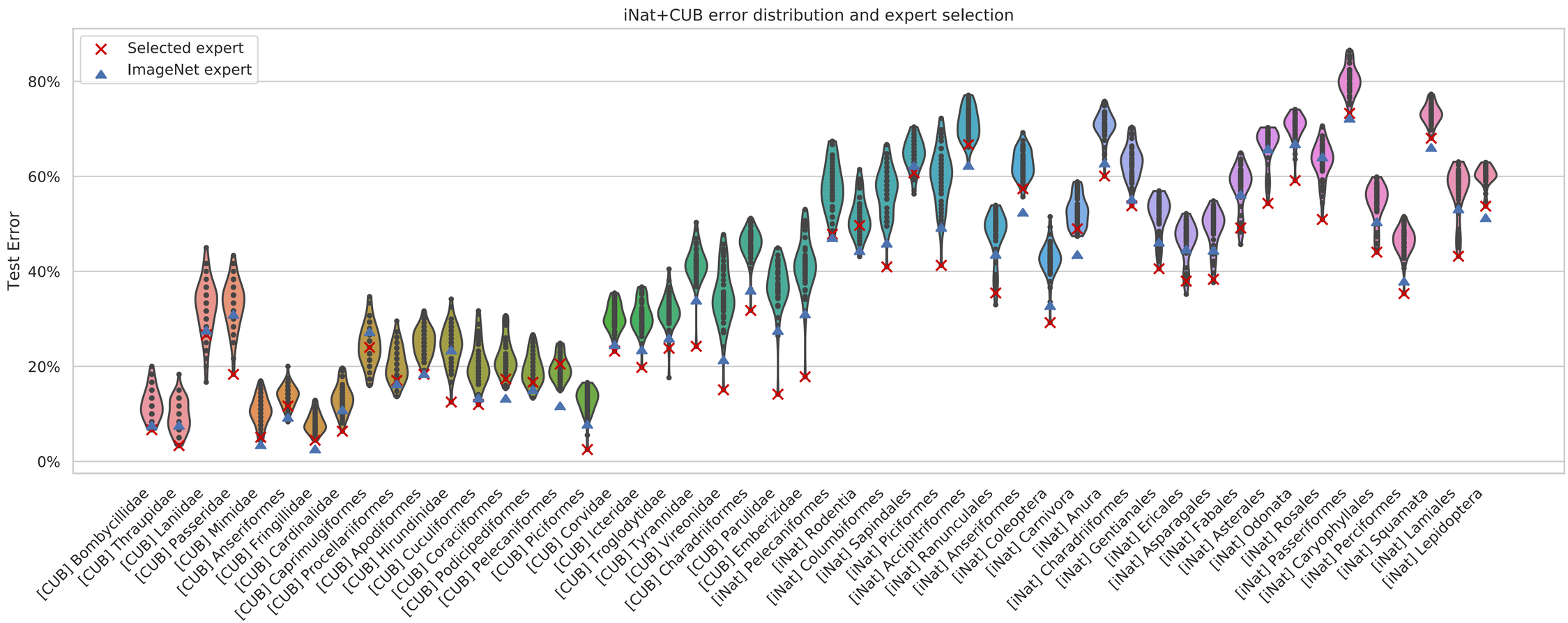
Distance between different tasks on the same inputs





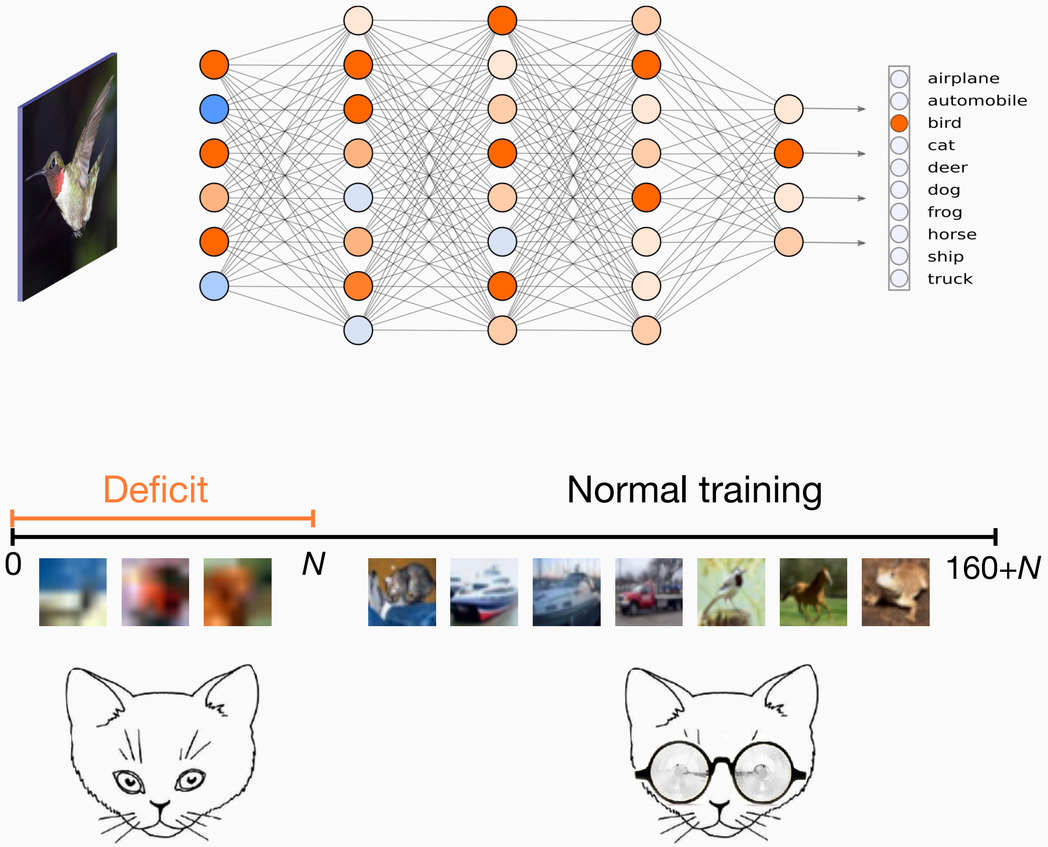
Animations from https://github.com/StanfordVL/taskonomy
Distance matrix from Achille et al,, Task2Vec

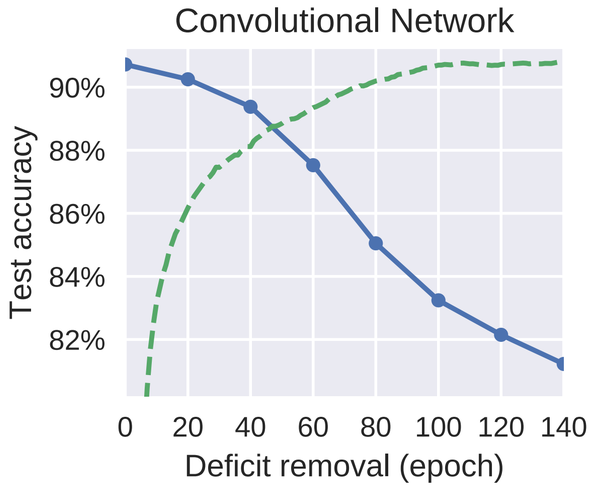
Proposing an optimal expert for the task

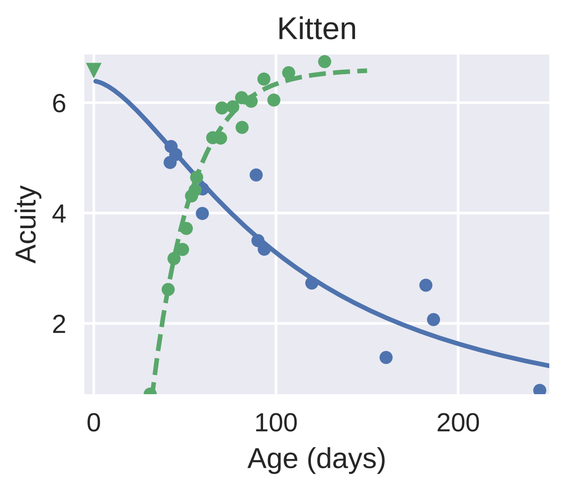


Allows to select the best expert to solve a task and substantially reduce error and training time.

A snag: Critical Learning Periods in Deep Networks



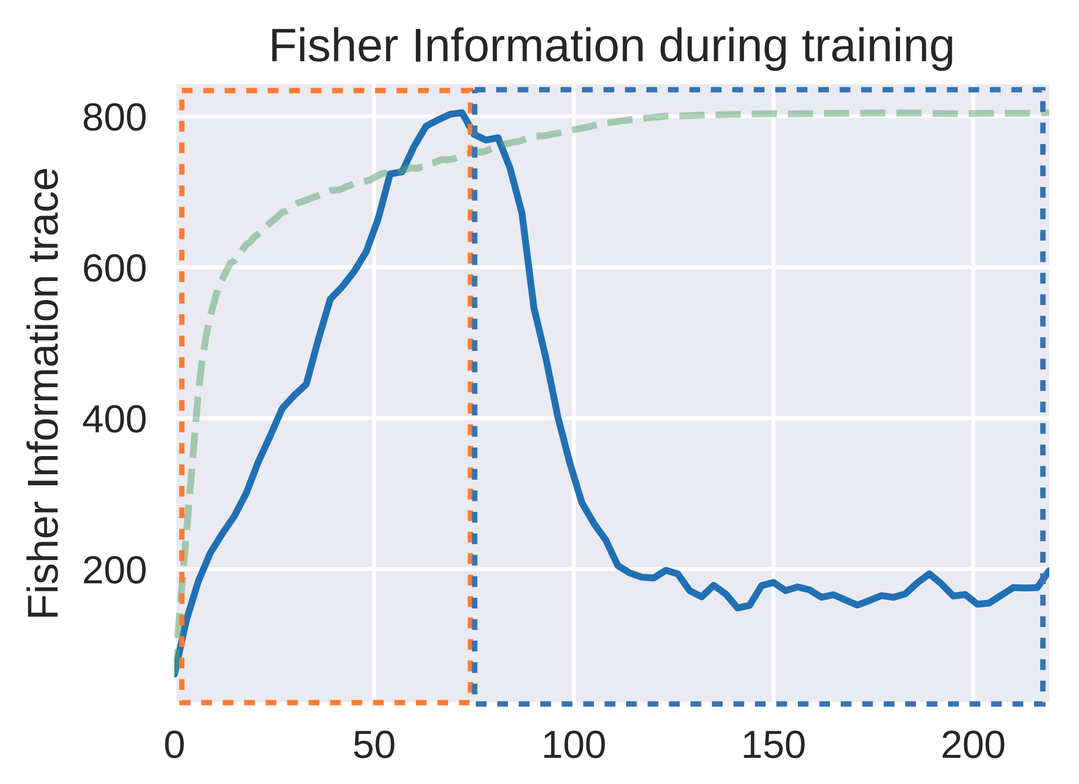




A., Rovere, Soatto, Critical Learning Periods in Deep Networks, 2018

Compression of weights leads to compression of activations

Information acquisition behaves in a non-trivial way during training.



The (Fisher) information in the weights decreases later in training

The effective information in the activations decreases

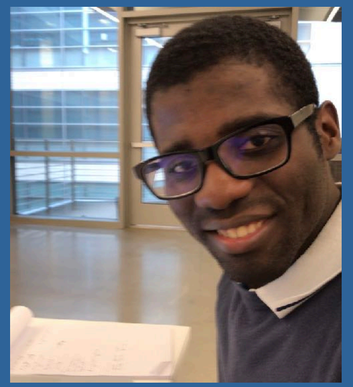
A., Rovere, Soatto, Critical Learning Periods in Deep Networks, 2018

THANKS!

- *Where is the Information in a Deep Neural Network?*, arXiv1:905.12213 (2019)
- The Information Complexity of Learning Tasks, their Structure and their Distance, arXiv:1904.03292 (2019)
- *TASK2VEC: Task embedding for meta-learning*, ICCV (2019)
- Critical Learning Periods in Deep Neural Networks, ICLR (2019)
- Emergence of Invariance and Disentangling in Deep Representations, JMLR (2018)

Matteo Rovere

Giovanni Paolini



Glen Mbeng

Stefano Soatto