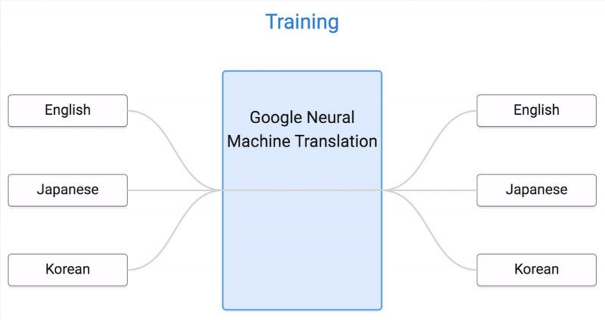
Towards an understanding of representational structure in deep neural networks

Ari Morcos
University of Bristol
June 14, 2018

DeepMind

Why should we care about understanding neural networks?

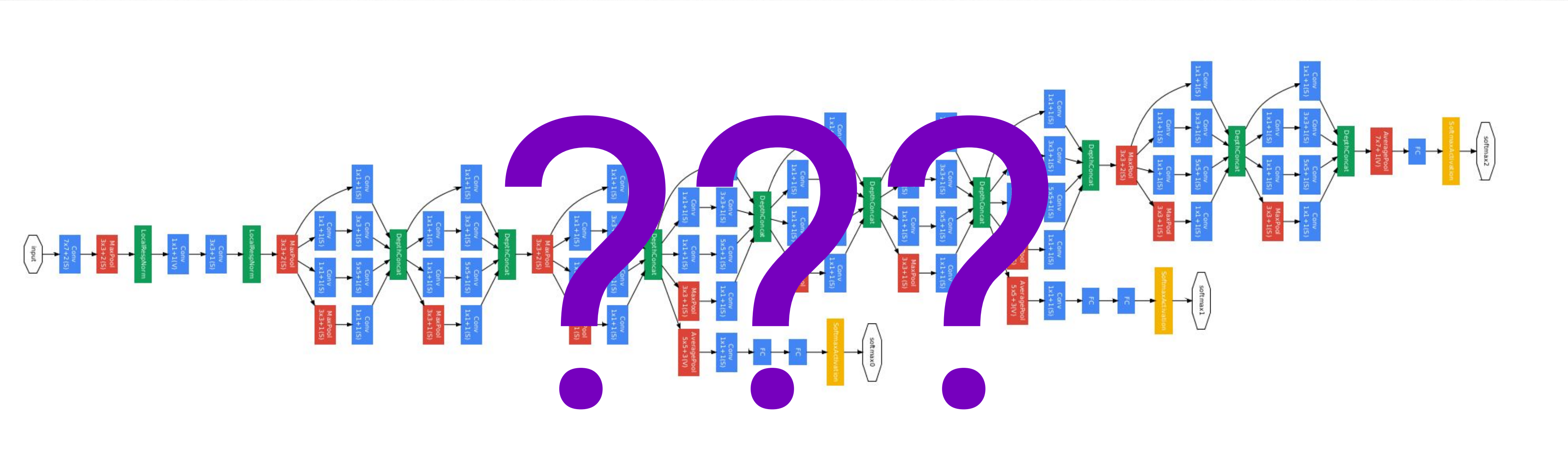
Silver et al., 2016, Silver et al., 2017



Wu et al., 2016

Karras et al., 2017

Why should we care about understanding neural networks?



Szegedy et al., 2015

Why should we care about understanding neural networks?

- Allows us to understand and *predict* failure modes of our models
- Understanding bottlenecks allows us to intelligently design bigger and better machine learning systems
- Many properties, such as abstraction, are intricately linked to representational structure
- May provide insights into neuroscience as well, at least methodologically

Outline

Single direction reliance as a predictor of generalization

Relationship between class selectivity and importance

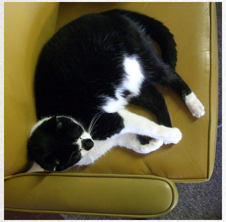
Using representational similarity to understand neural networks

What differentiates networks which generalize from those which memorize?

Networks can memorize random functions

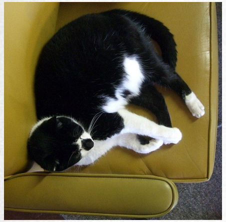
True labels

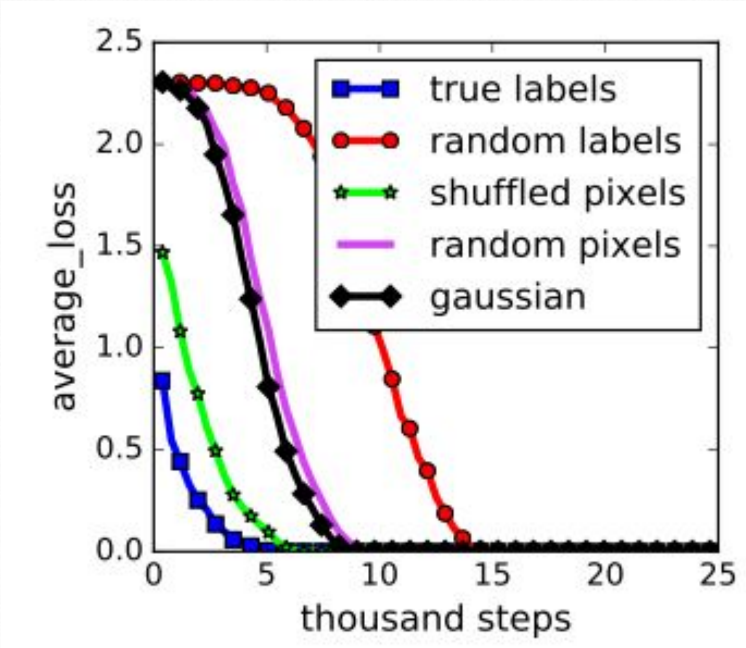
Cat | Airplane | House | Airplane




Random labels

Airplane | Cat | Airplane | House



Zhang et al., 2017

What differentiates networks which memorize from those which generalize?

- Sharpness of minima (Hochreiter and Schmidhuber, 1997, Keskar et al., 2017, Neyshabur et al., 2017)
  - But see Dinh et al., 2017

- Information complexity (Achille and Soatto, 2017)

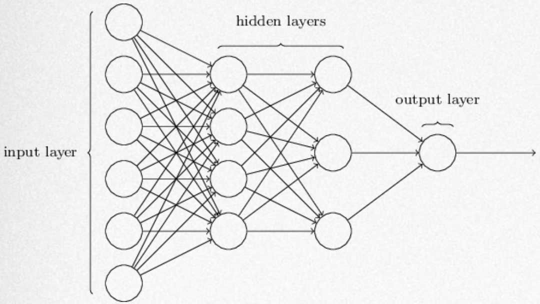
Is the importance of single directions in activation space correlated with generalization?

A possible relationship between overfitting and single direction importance

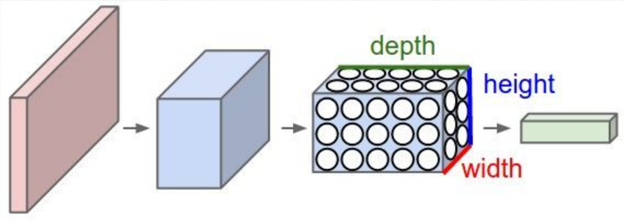
- If the training dataset has structure and is large enough, the minimal description length of memorizing the training set should be greater than or equal to that of the true data-generating function
- A network which memorized the training set will likely use much more of the network's capacity than one which learned the true data-generating function
- A memorizing network should use more single directions than one which learns the true data-generating function
- Therefore, if a random direction is deleted, the probability that this deletion disrupts the network should be higher for a memorizing network

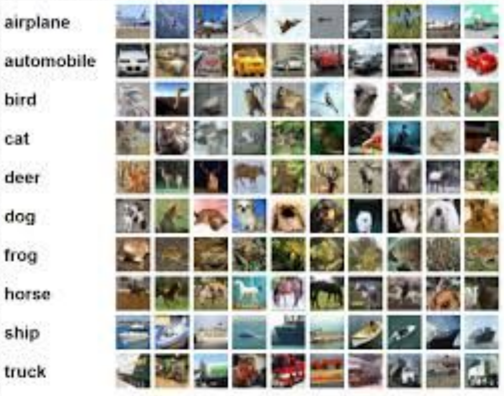
Models analyzed

MNIST MLP
(2 hidden layers)

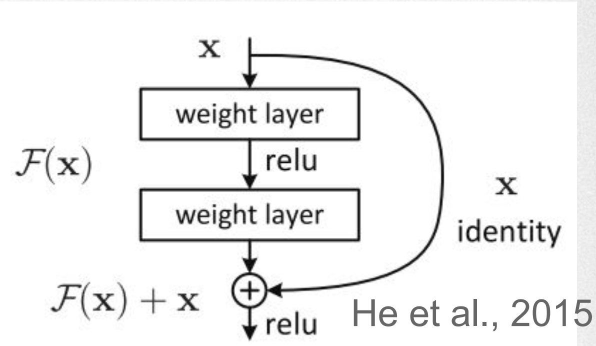


CIFAR-10 ConvNet
(11 layers)



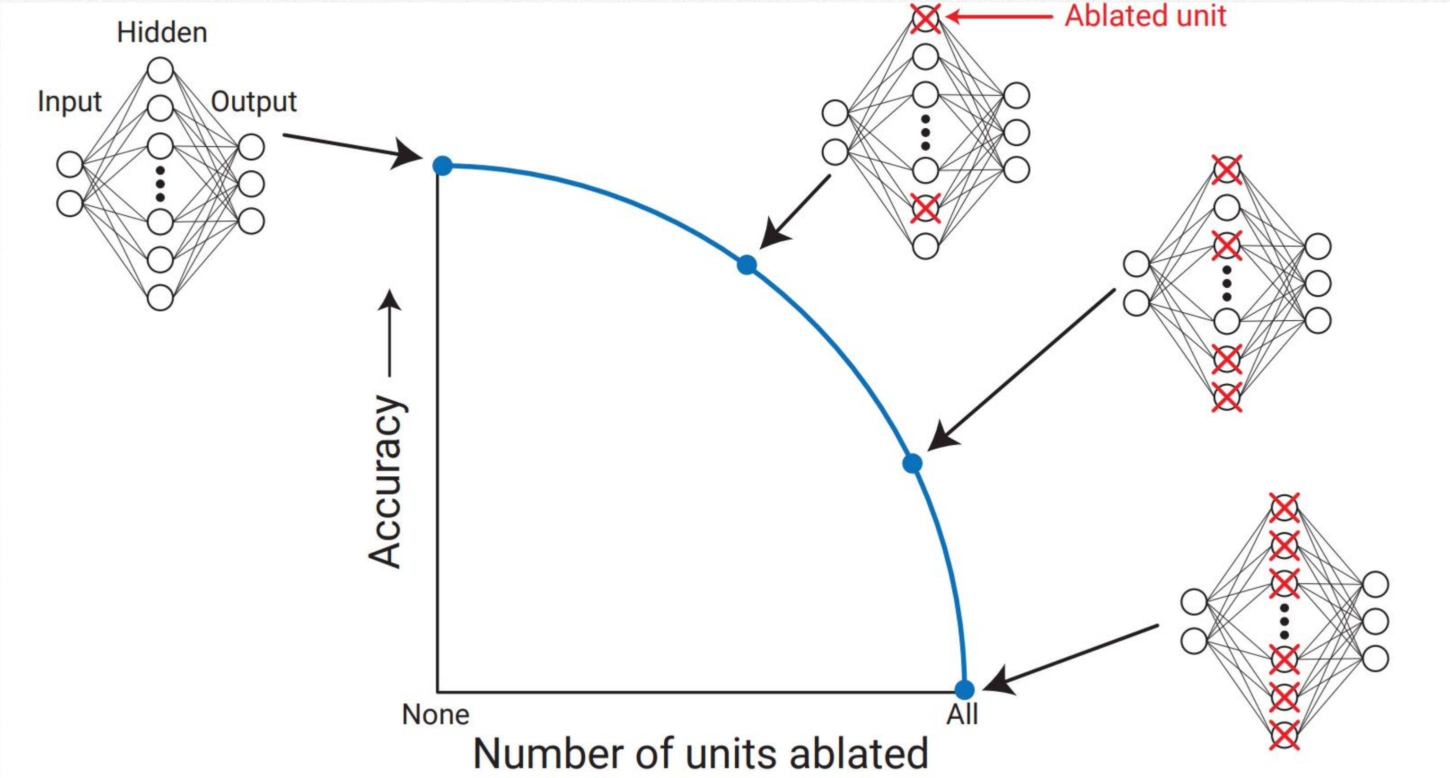


ImageNet ResNet
(50 layers)

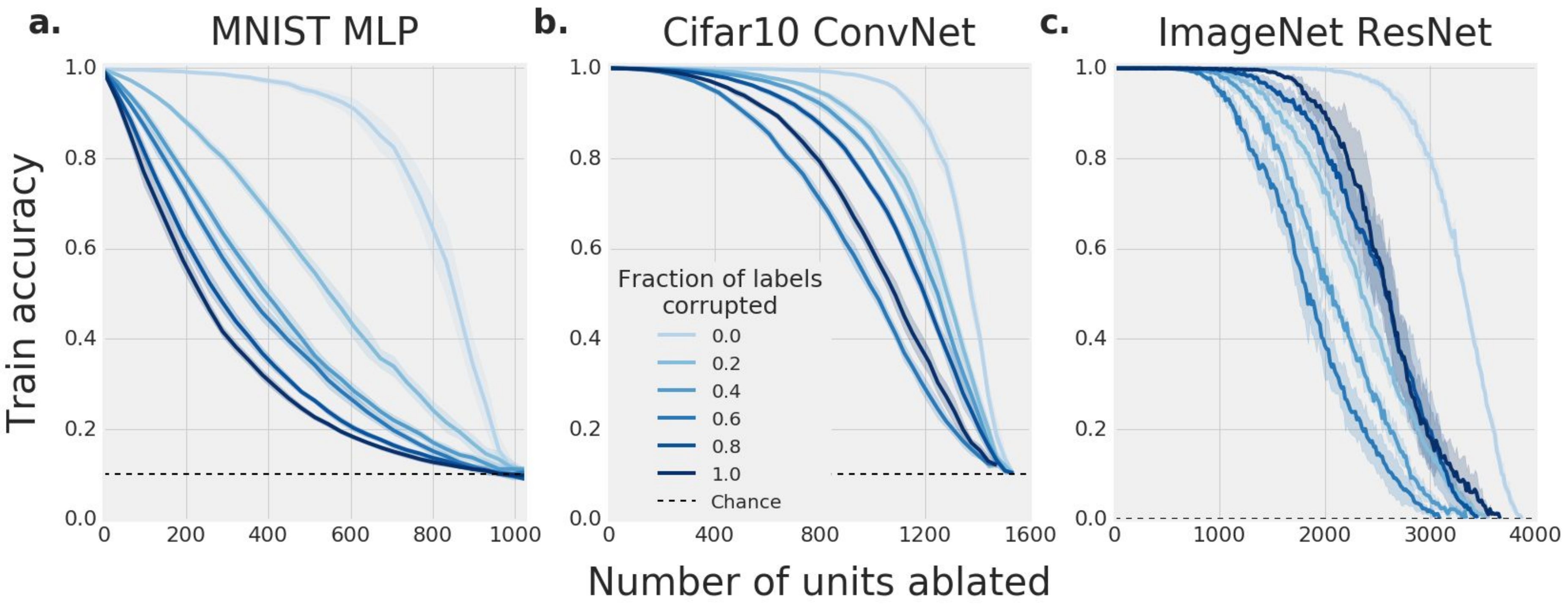


He et al., 2015

Experimental design

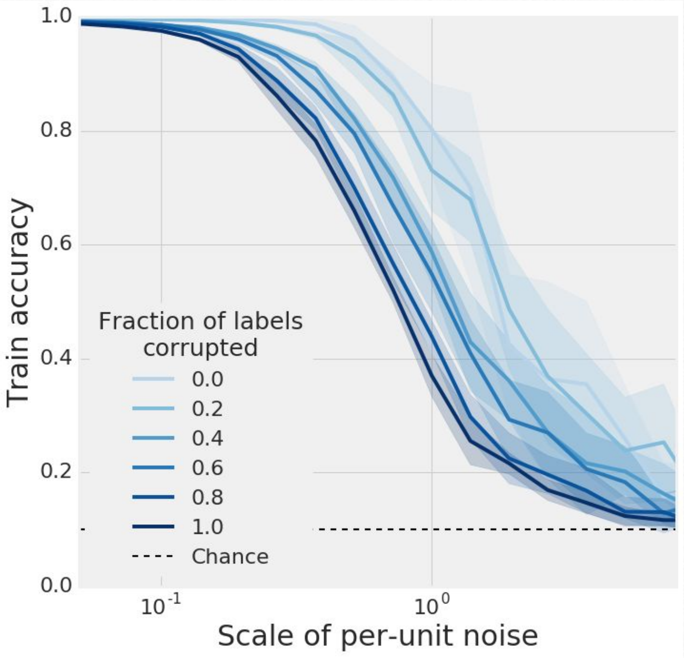


Memorizing networks are more susceptible to random ablations than networks which generalize

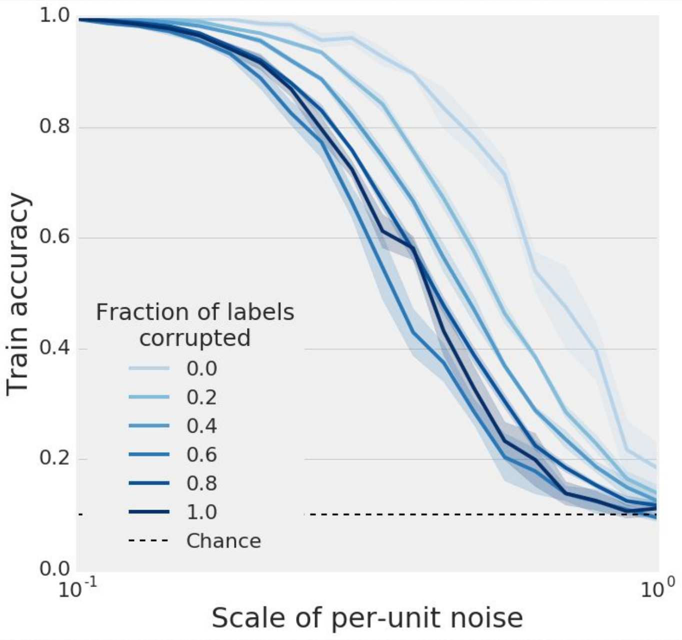


Memorizing networks are more susceptible to random ablations than networks which generalize

MNIST MLP

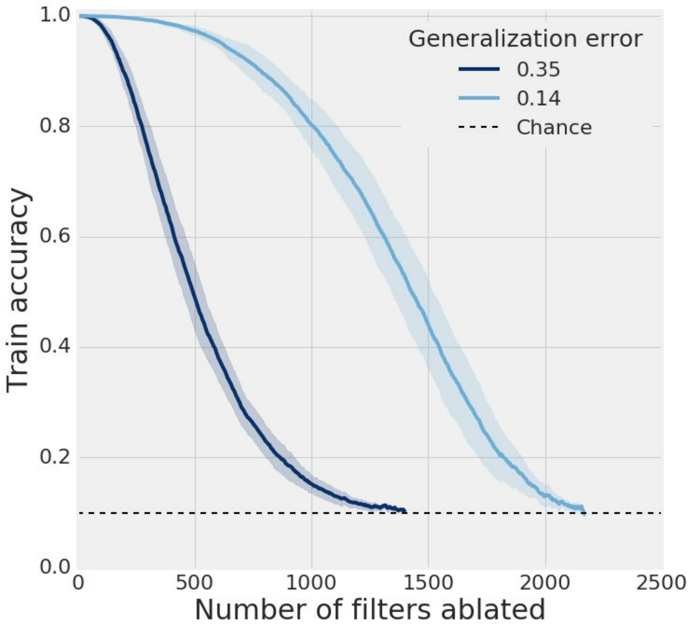


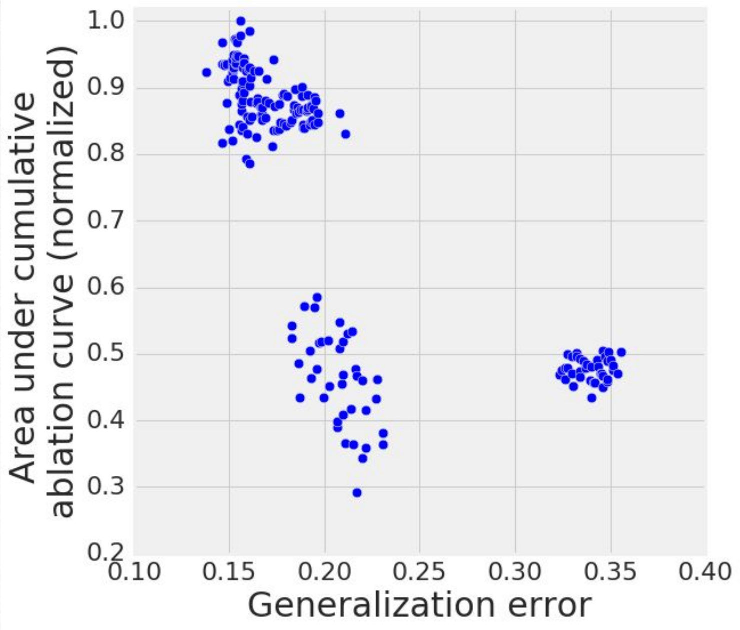
Cifar10 ConvNet



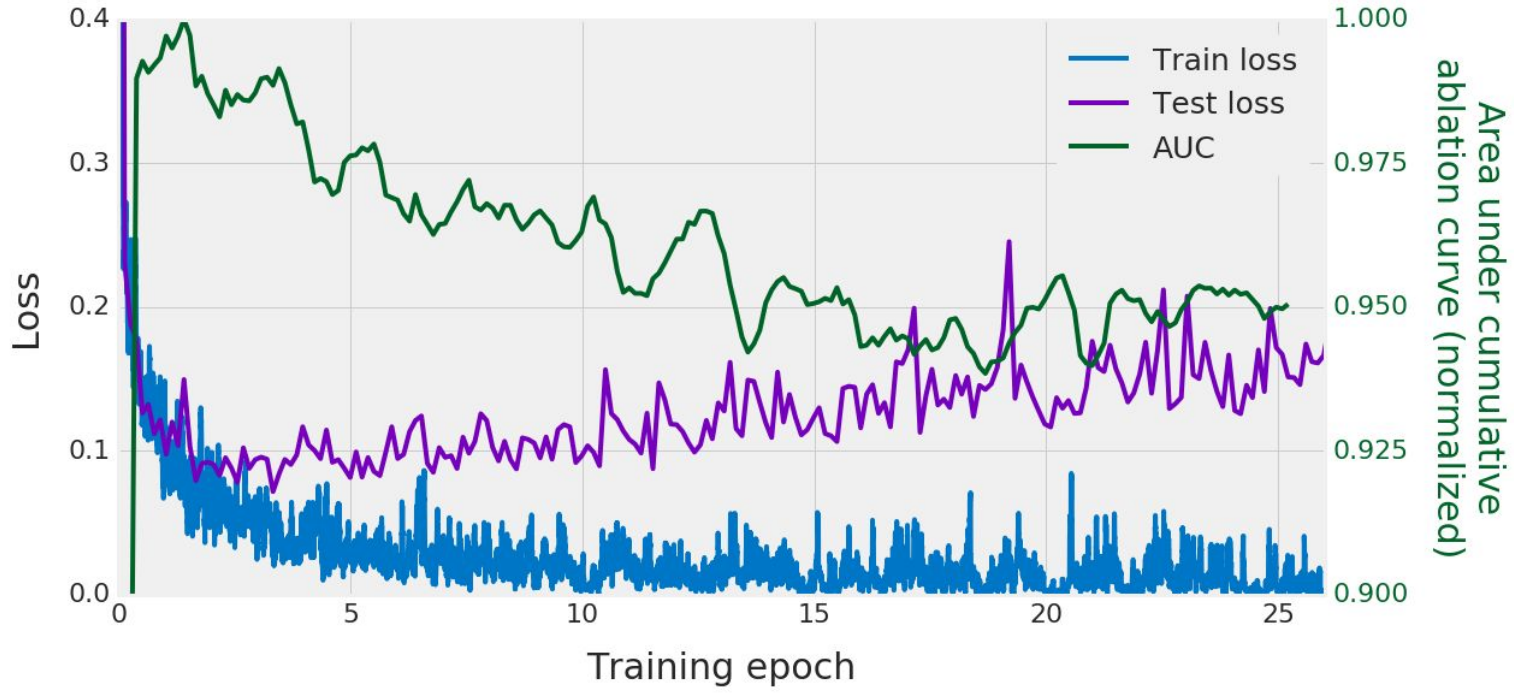
Networks which generalize well are more robust than those which generalize poorly

CIFAR-10 ConvNet

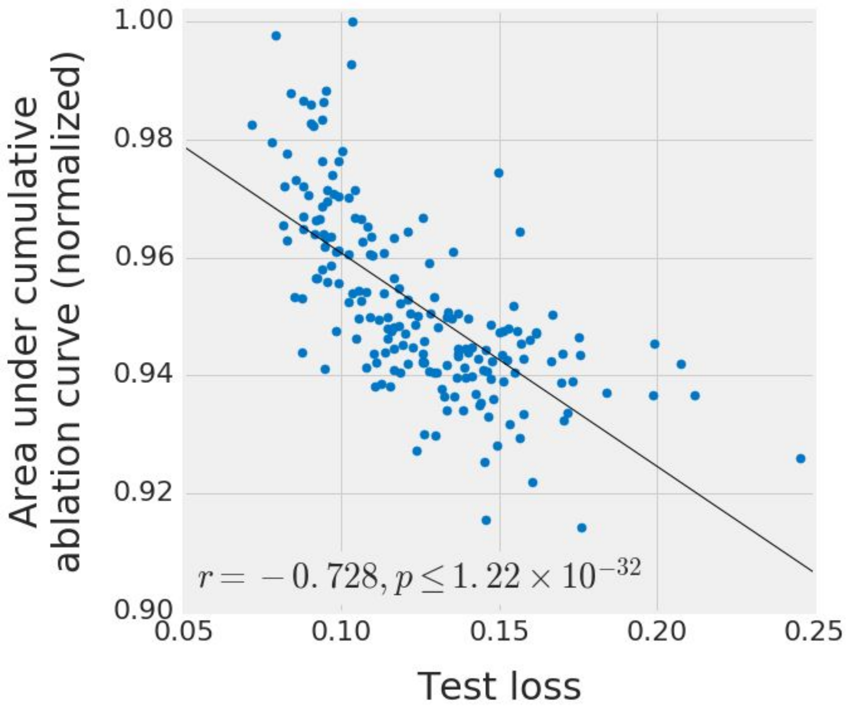




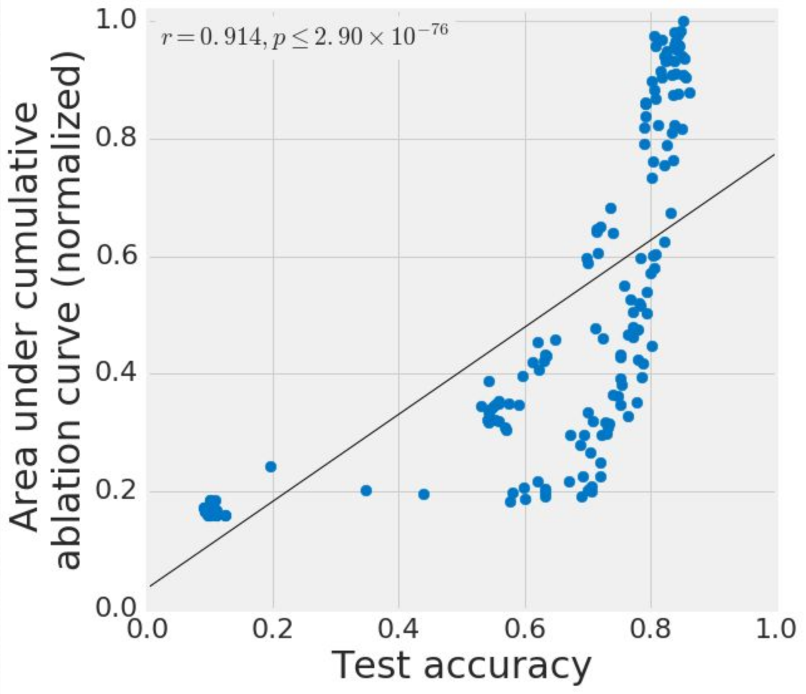
Single direction reliance as a signal for early stopping



Single direction reliance as a signal for early stopping



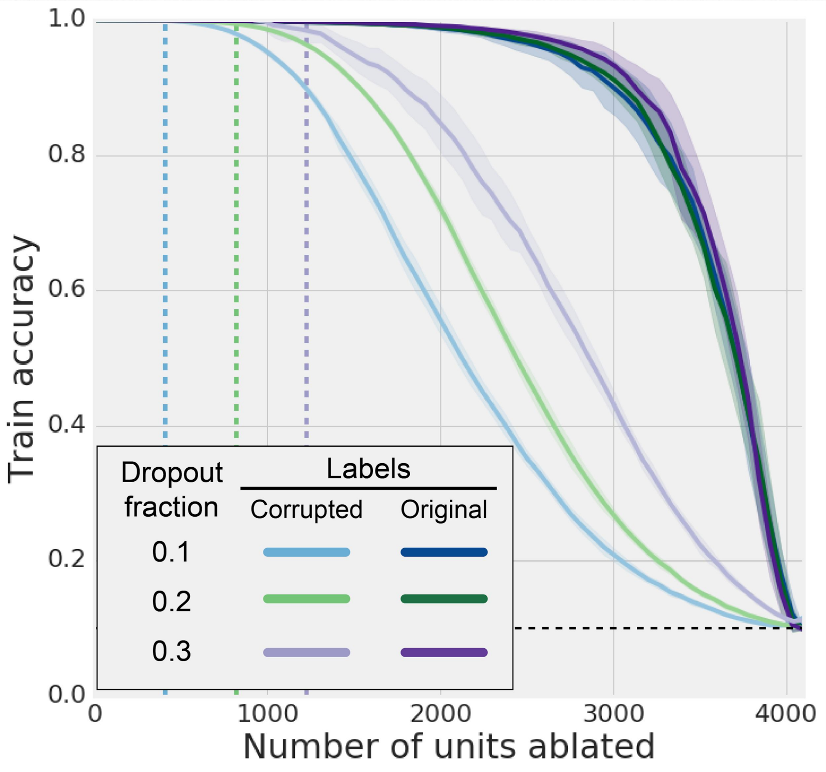
Single direction reliance as a signal for hyperparameter selection



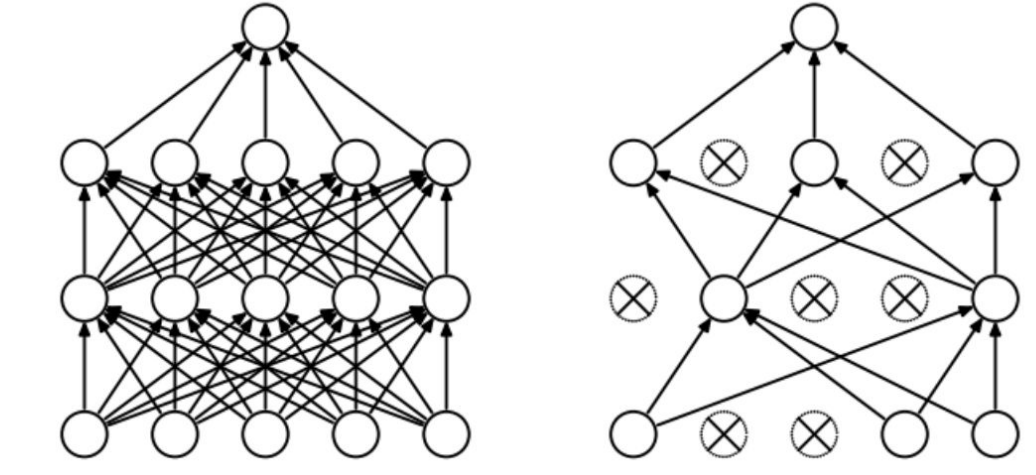
	Probability of selection
Top 1 of 48	0.13
Top 5 of 48	0.83
Top 10 of 48	0.98

Average error: 1 ± 1.1%

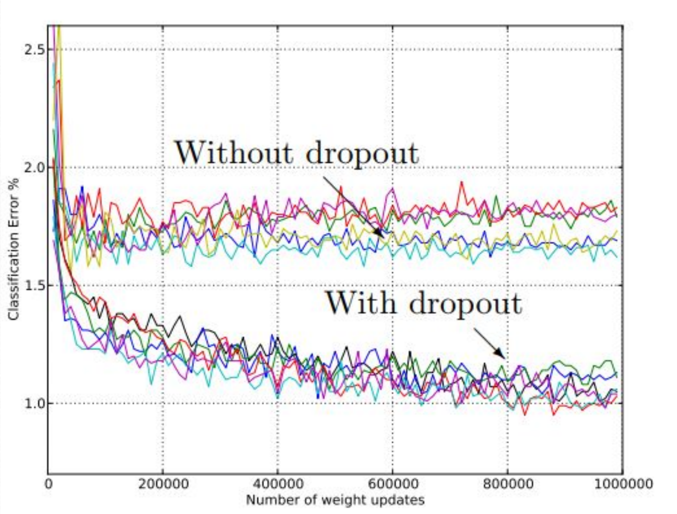
Dropout discourages memorization, but does not increase robustness to ablation past the dropout threshold



Dropout



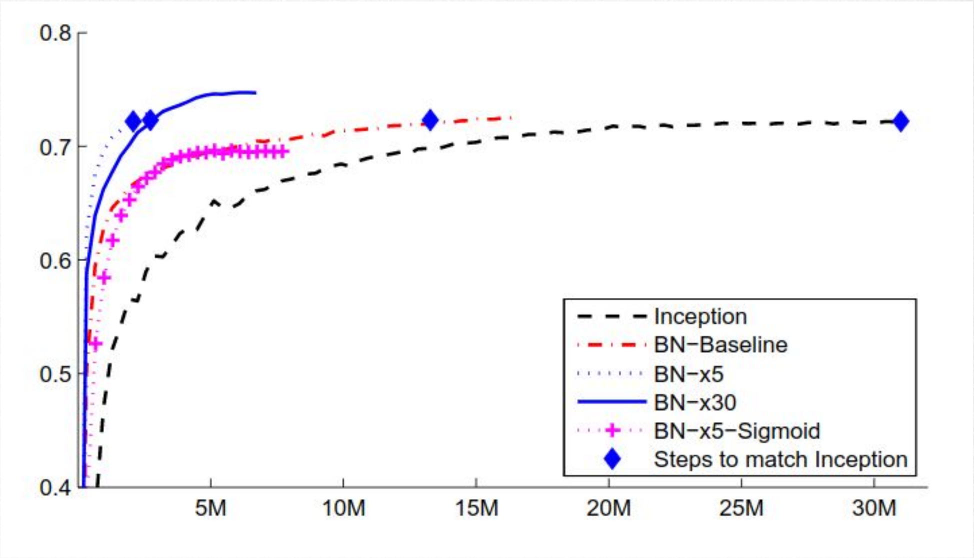
(a) Standard Neural Net (b) After applying dropout.



Srivastava et al., 2014

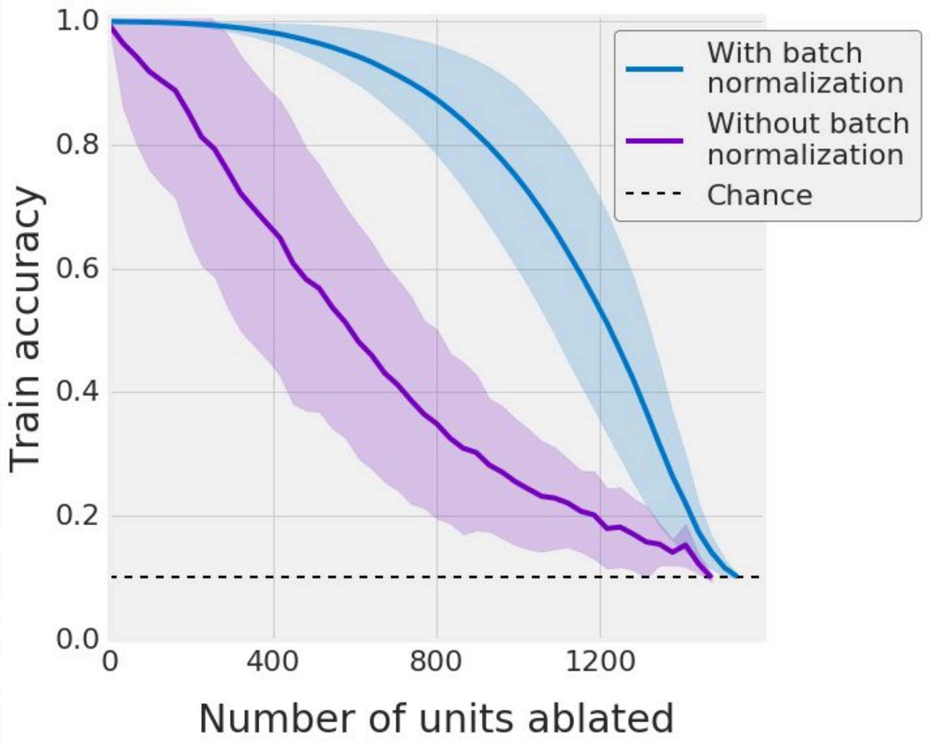
Batch normalization

- Normalizes the statistics across a mini-batch
- Aims to ensure that the distribution of activations across a batch is constant



Ioffe and Szegedy, 2015

Batch normalization makes networks more robust to random ablations



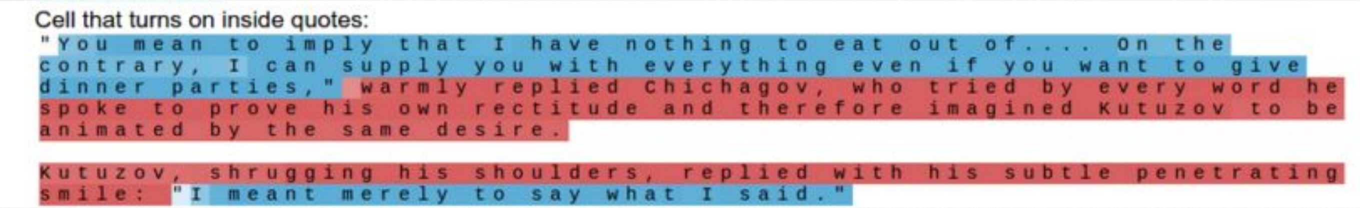
What have we learned?

- Networks which memorize the training set are substantially more sensitive to cumulative ablations and noise than networks which approximate the data-generating function
- Even among networks trained with the same topology and data, instances with better generalization performance are more robust to cumulative ablations
- Batch normalization implicitly regularizes robustness to cumulative ablations

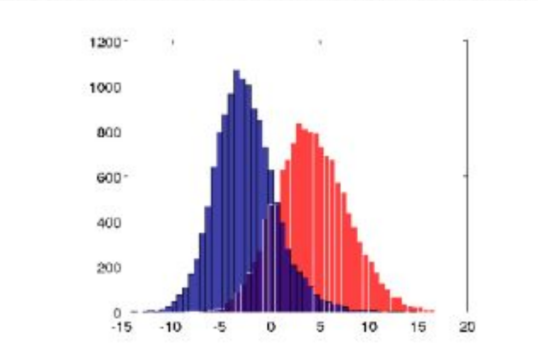
Networks which are less reliant on single directions are better at generalization

Single unit selectivity, performance, and importance

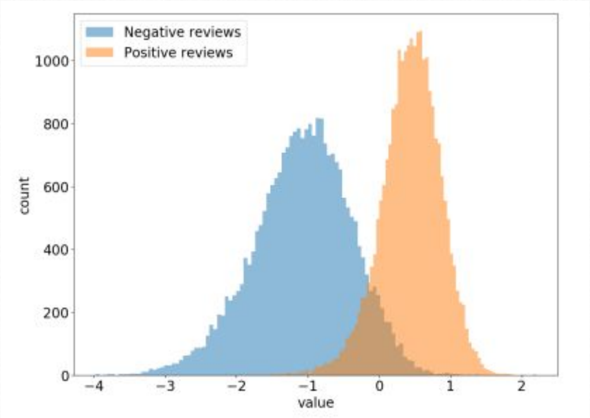
Selective single neurons



Karpathy et al., 2016

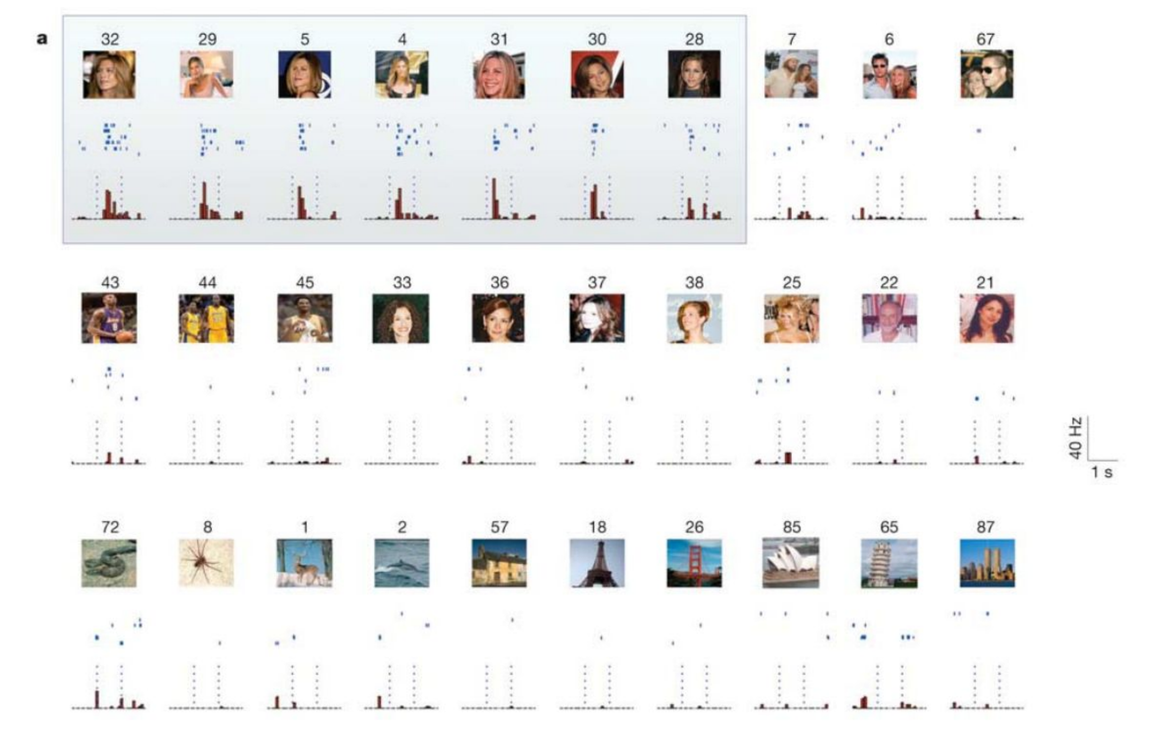


Le et al., 2011



Radford et al., 2017

Selective single neurons in the brain



Quian Quiroga et al., 2005

Quantifying selectivity

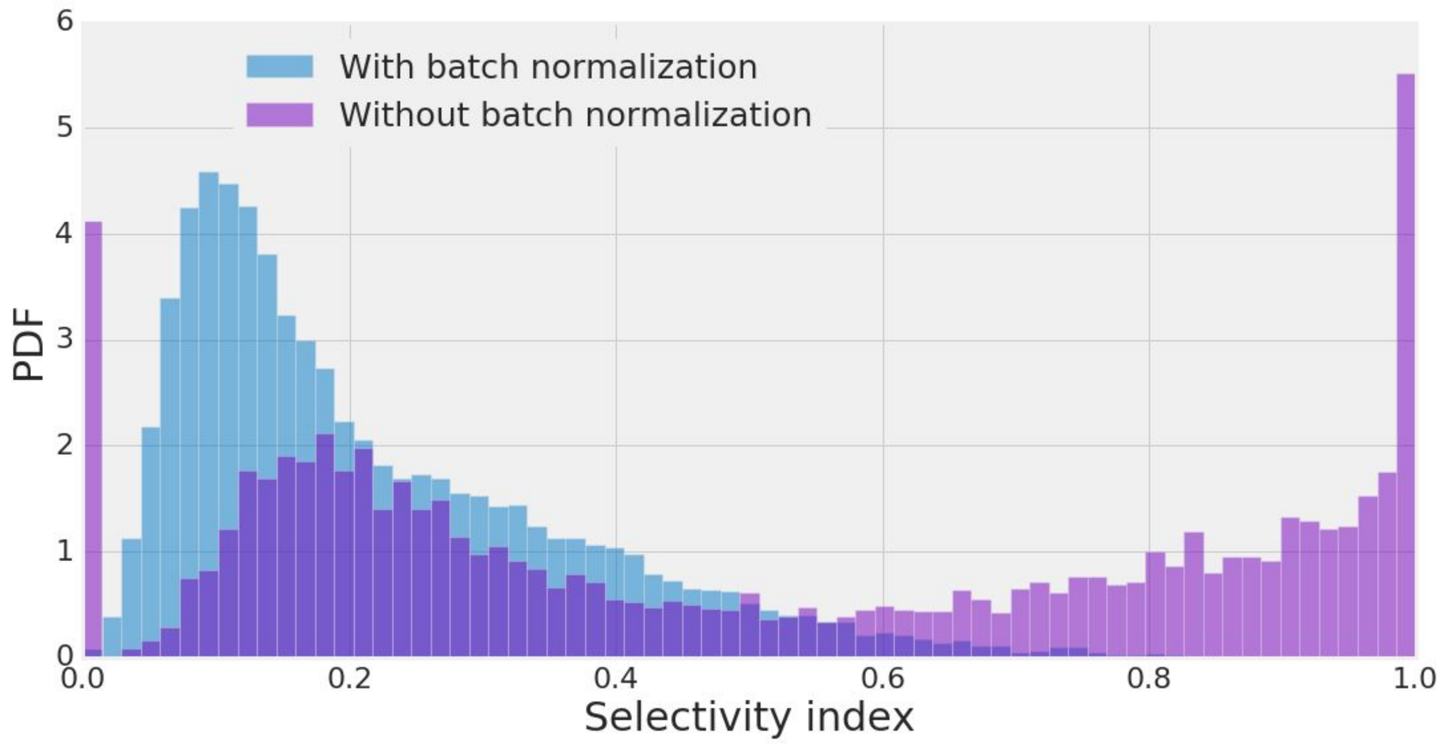
$$best = \arg\max_i(\mu_i)$$

$$selectivity = \frac{\mu_{best} - mean(\mu_{-best})}{\mu_{best} + mean(\mu_{-best})}$$

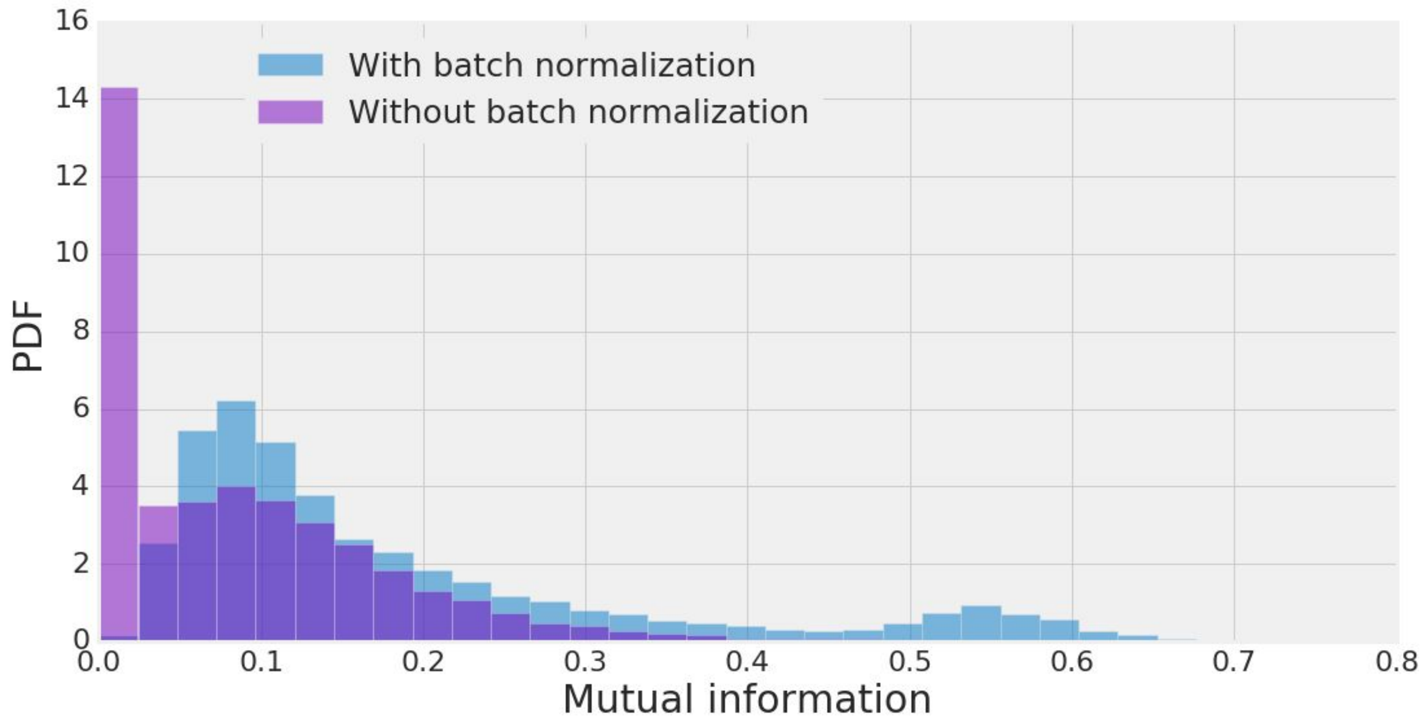
0 means a unit's average activity is the same for all classes

1 means a unit is only active for a single class, and silent for all others

Batch norm substantially decreases the selectivity of individual feature maps

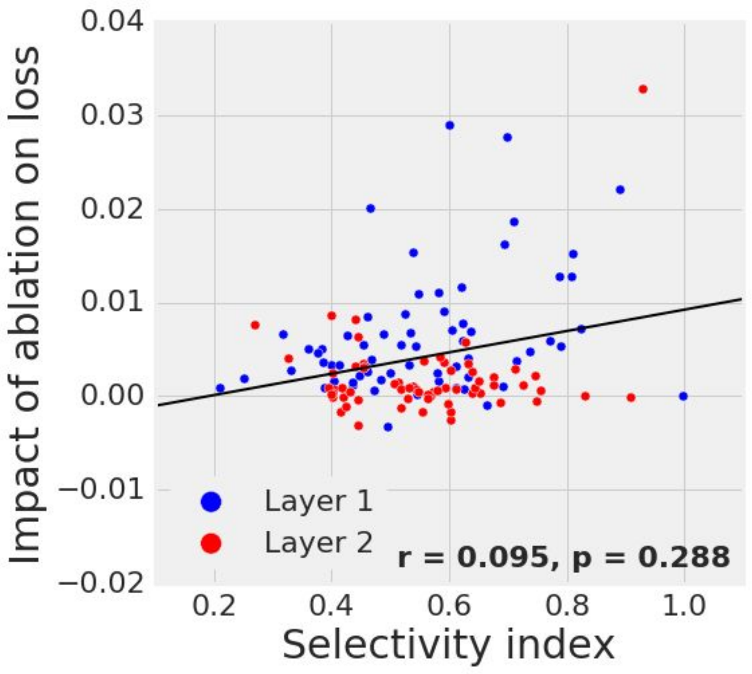


Batch norm substantially increases the mutual information of individual feature maps



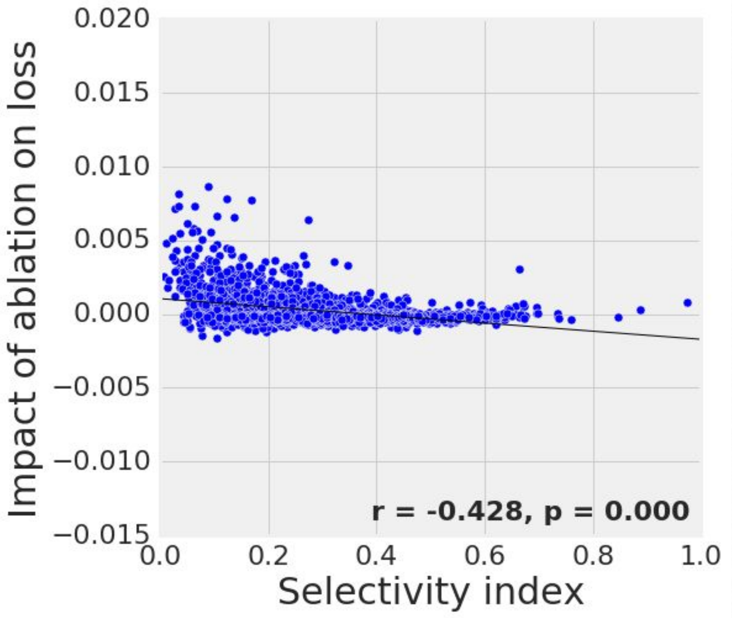
Are selective single neurons more important than non-selective single neurons?

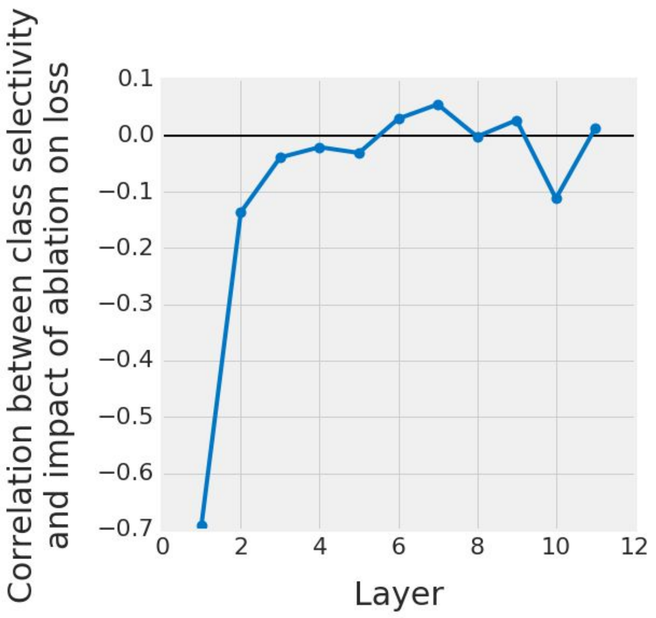
Mnist MLP



Are selective single neurons more important than non-selective single neurons?

CIFAR-10 ConvNet

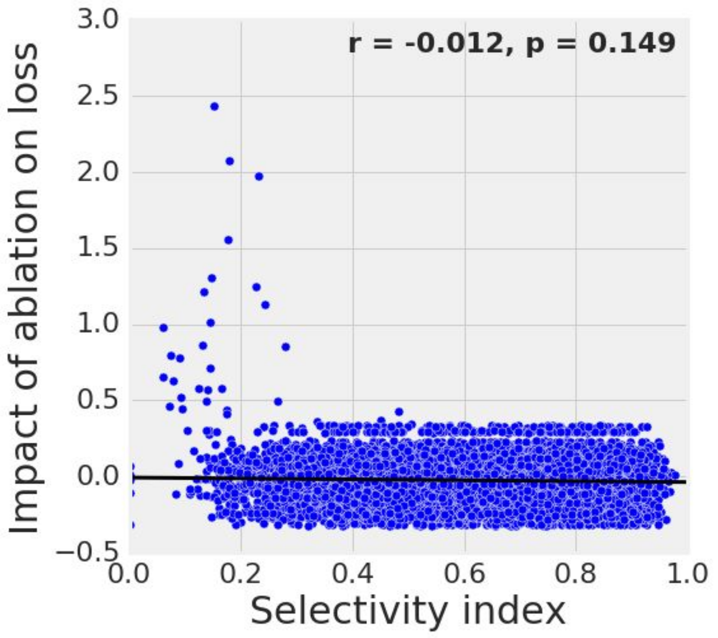


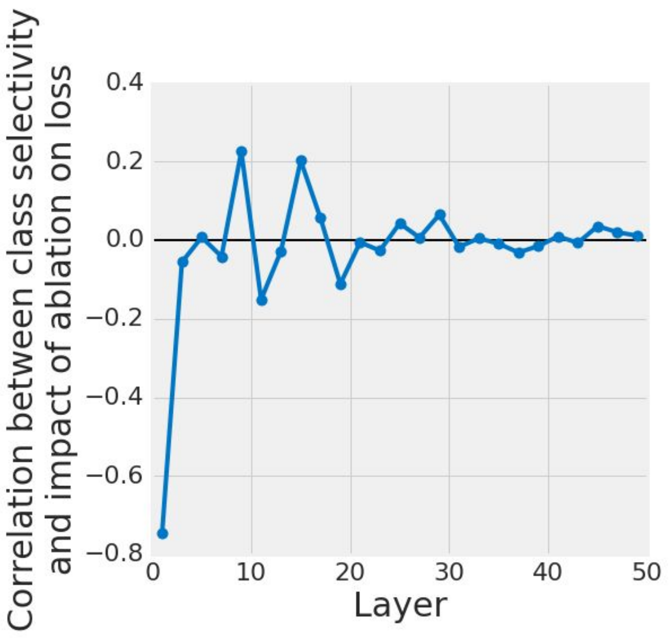


DeepMind

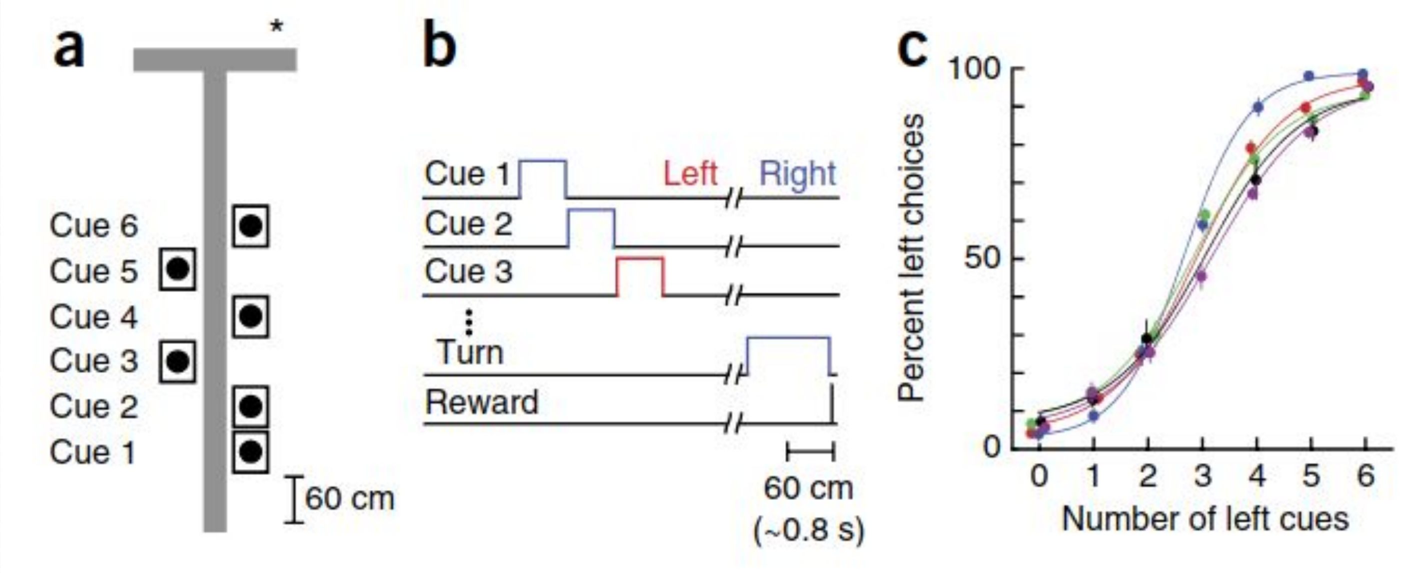
Are selective single neurons more important than non-selective single neurons?

ImageNet ResNet



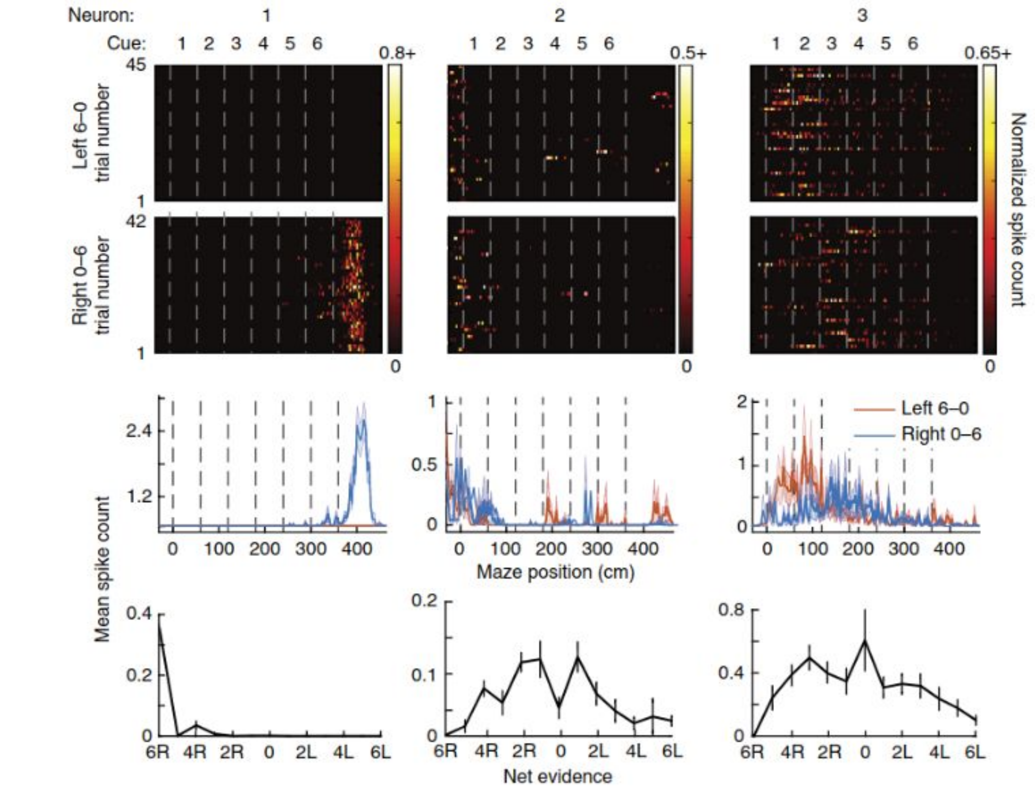


Distributed representations in the brain



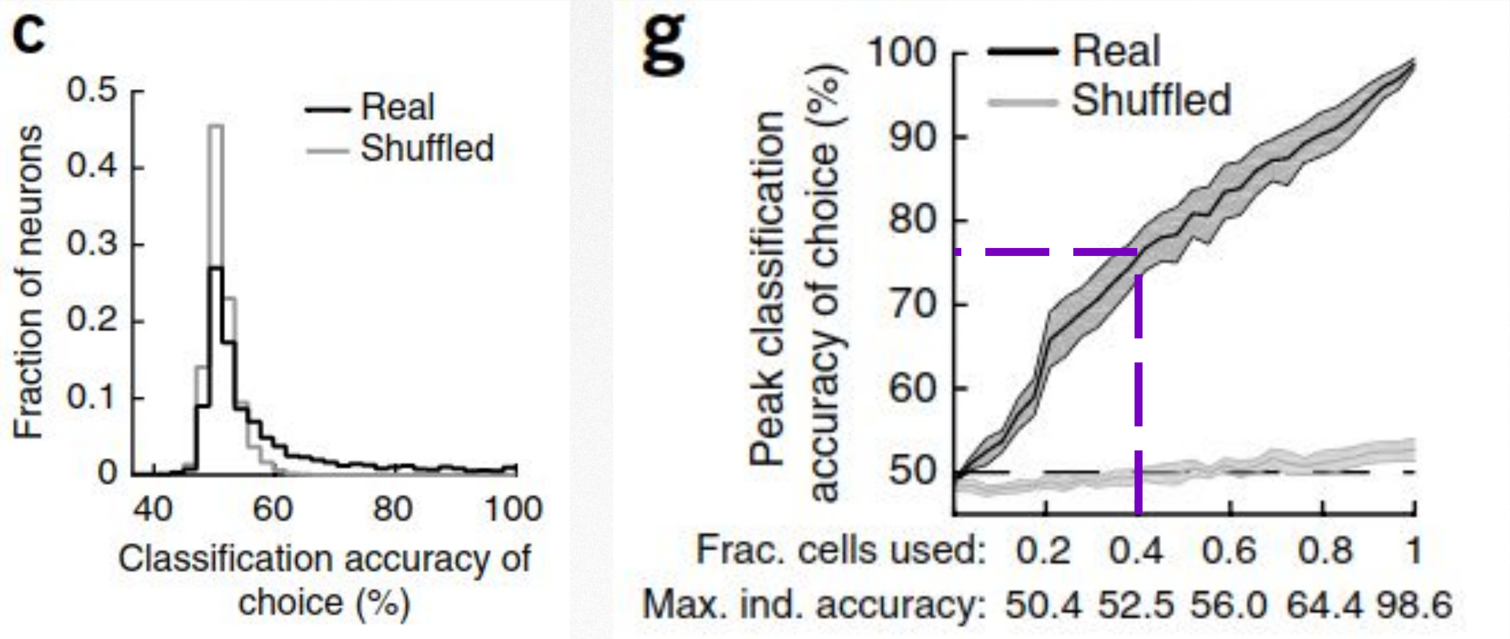
Morcos and Harvey, 2016

Distributed representations in the brain



Morcos and Harvey, 2016

Distributed representations in the brain



Morcos and Harvey, 2016

What have we learned?

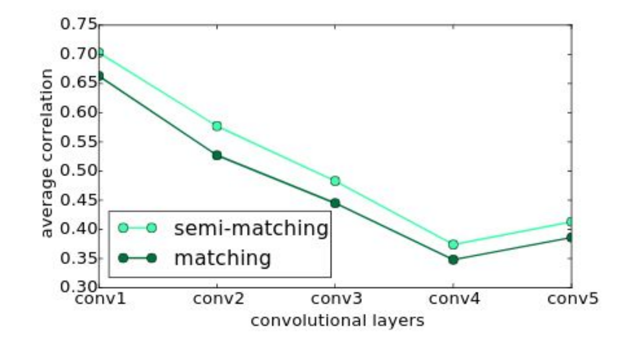
- Batch normalization, which markedly improves network performance, substantially decreases the class selectivity of feature maps, but increases the mutual information
  - This result suggests that batch normalization discourages sparse representations in which each unit encodes a lot of information about one class in favor of more distributed representations in which each unit encodes a little information about multiple classes
- The class selectivity of single units is a poor predictor of that unit's importance to the network output
- This result mirrors recent work demonstrating distributed representations in the brain (though we explicitly *do not claim* that our models are representative of the brain)

Using representational similarity to understand generalization and convergence dynamics

How can we compare representations across networks?

- Networks often have different topologies, both across networks and across layers
  - E.g., how do you compare layer 1 with 64 filters to layer 7 with 256 filters?

- Networks are highly unlikely to learn solutions with one-to-one mappings between units (Li et al, 2016)



Using CCA to compare representations

Given

$$X \in \mathbb{R}^{a \times n}$$

$$Y \in \mathbb{R}^{b \times n}$$

a, b - number of variables (neurons)

n - number of observations

Optimized

$$u \in \mathbb{R}^{a}$$

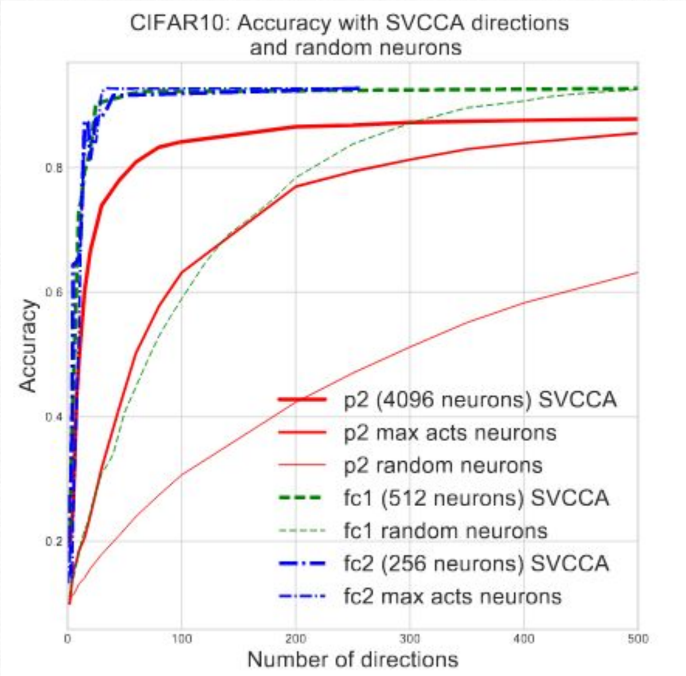
$$v \in \mathbb{R}^{b}$$

$$\arg\max_{u,v} \frac{\langle u^T X, v^T Y \rangle}{\|u^T X\| \cdot \|v^T Y\|}$$

How similar are these matrices subject to a linear transformation?

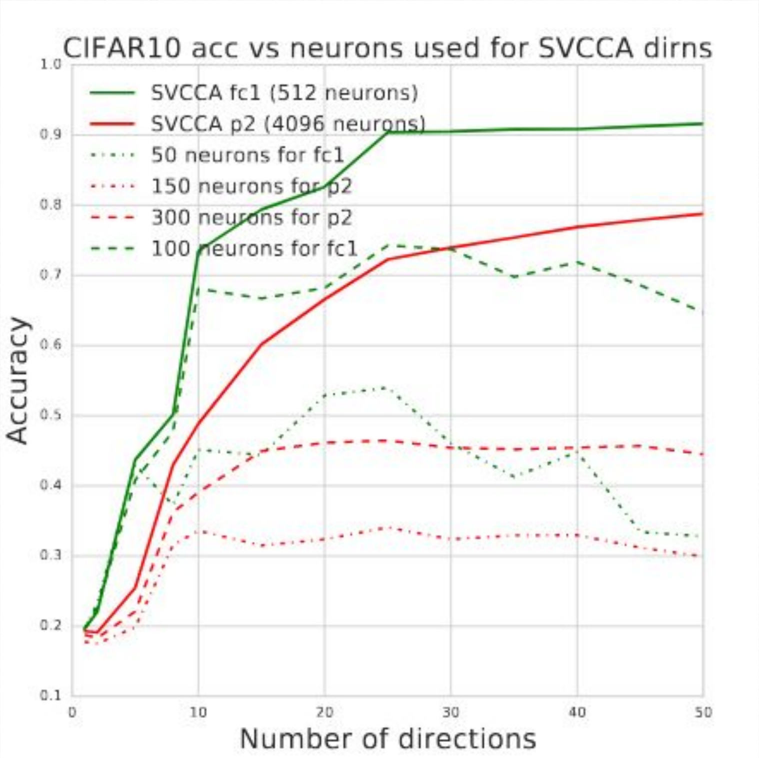
Hotelling, 1936, Raghu et al., 2017

CCA finds a small set of directions which are sufficient for computation



Raghu et al., 2017

CCA directions are distributed across neurons

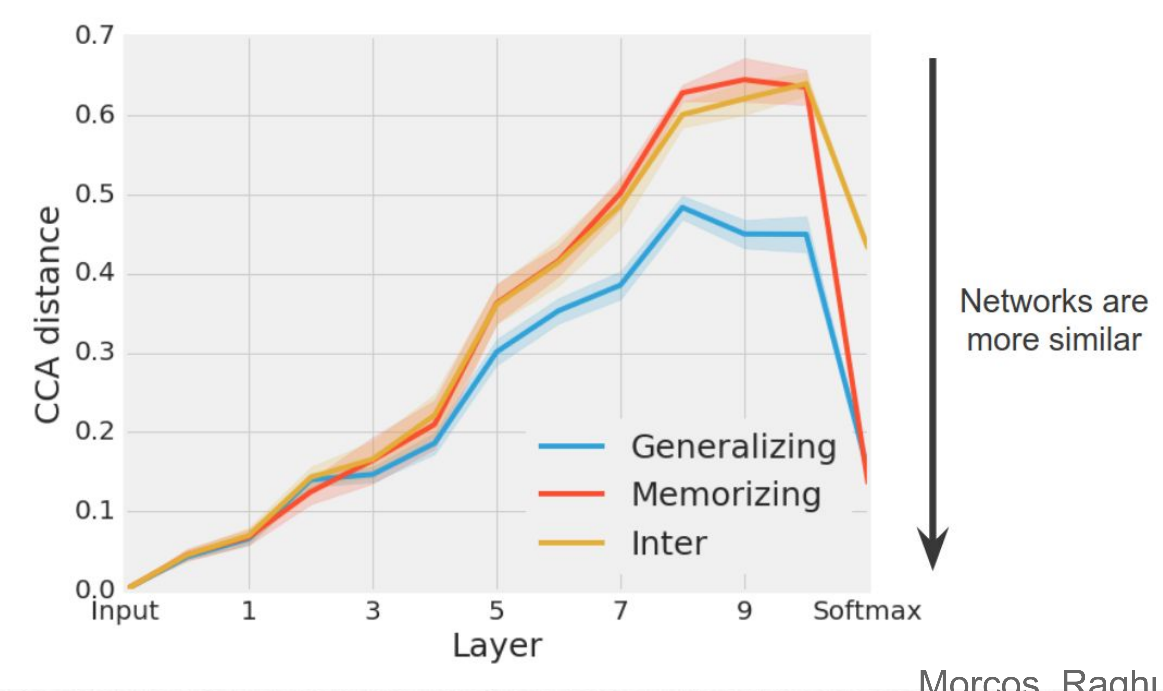


Raghu et al., 2017

Can CCA distinguish between generalizing and memorizing networks?

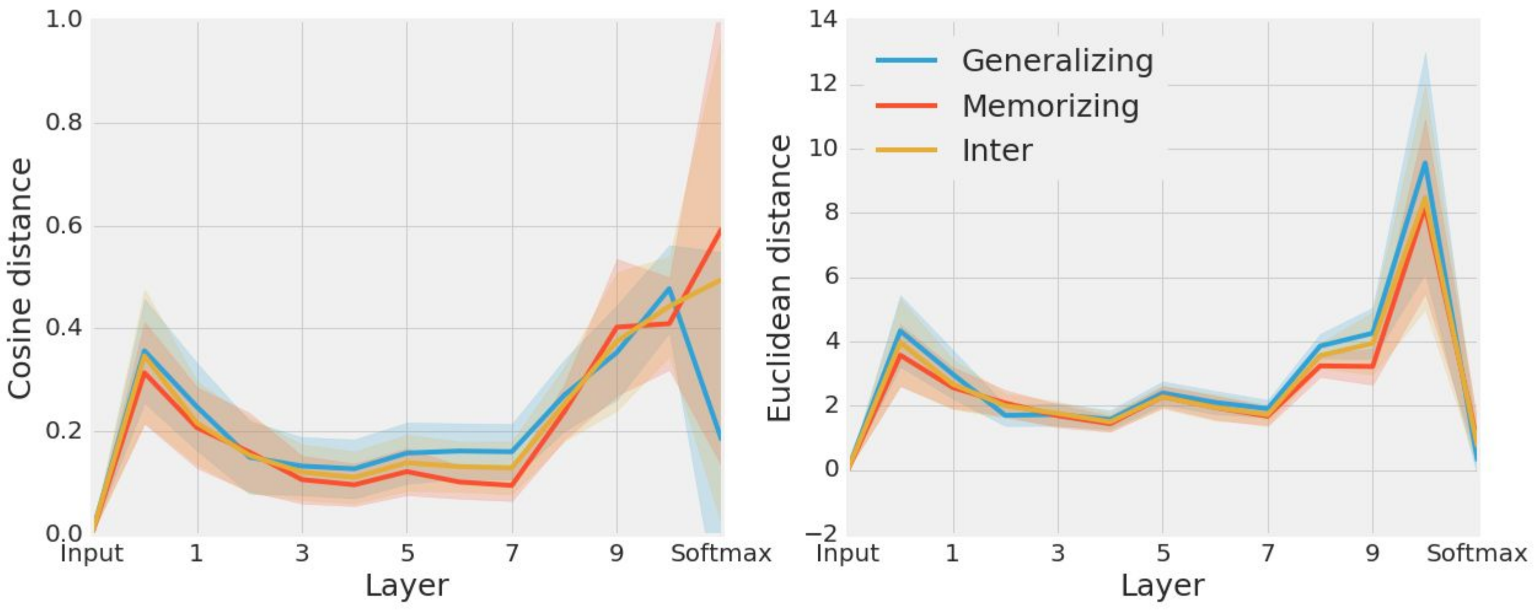
- There are likely many ways to memorize training data, but comparatively few generalizable solutions
- We would therefore expect the representations across networks which generalize to be more similar than those of memorizing networks
- We trained groups of networks on true labels ("Generalizing") and randomized labels ("Memorizing")
- Used CCA to compare representations within each group of networks and between generalizing and memorizing networks ("Inter")

Networks which generalize converge to more similar solutions than those which memorize



Morcos, Raghu, and Bengio, 2018

Cosine and euclidean distance do not recover these differences

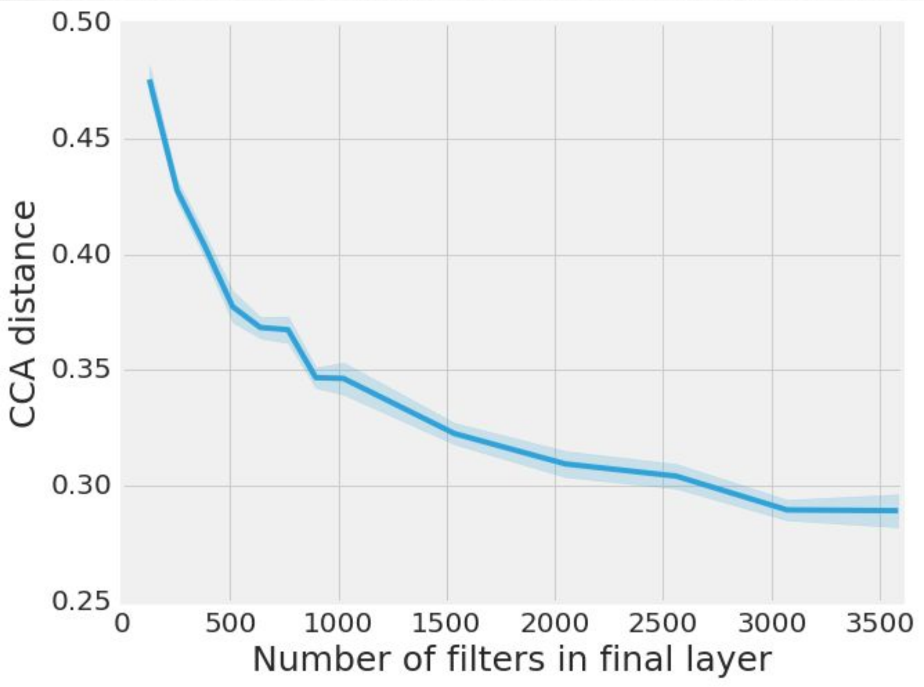


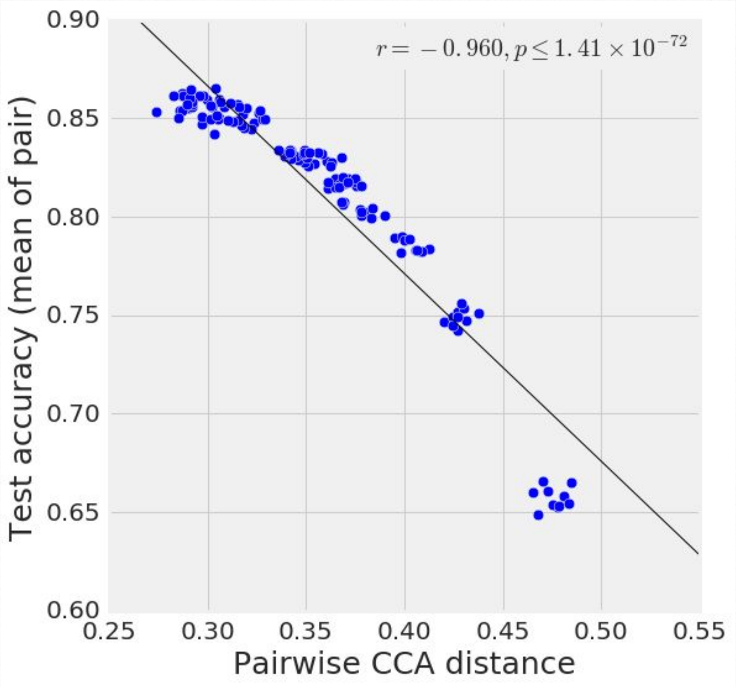
Morcos, Raghu, and Bengio, 2018

Why can we prune networks to high performance but not learn small networks in the first place?

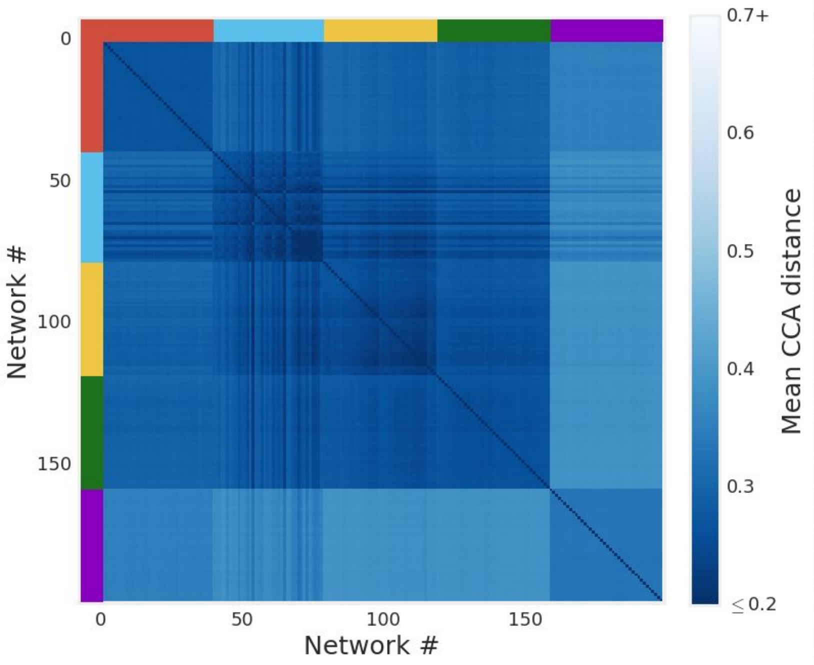
- Network pruning, in which neurons and/or weights are removed, is widespread (Li et al., 2017, Anwar et al., 2015, Molchanov et al., 2017, and more)
  - Often, >85% of parameters can be removed with minimal performance drops
- However, simply initializing and training a small network doesn't lead to good performance
  - Why?
- **Lottery ticket hypothesis**: successful training depends on a "lucky" random initialization of a smaller subcomponent of the network (Frankle and Carbin, 2018)
  - Larger networks have more subnetworks, and therefore higher probability of a "lucky" initialization

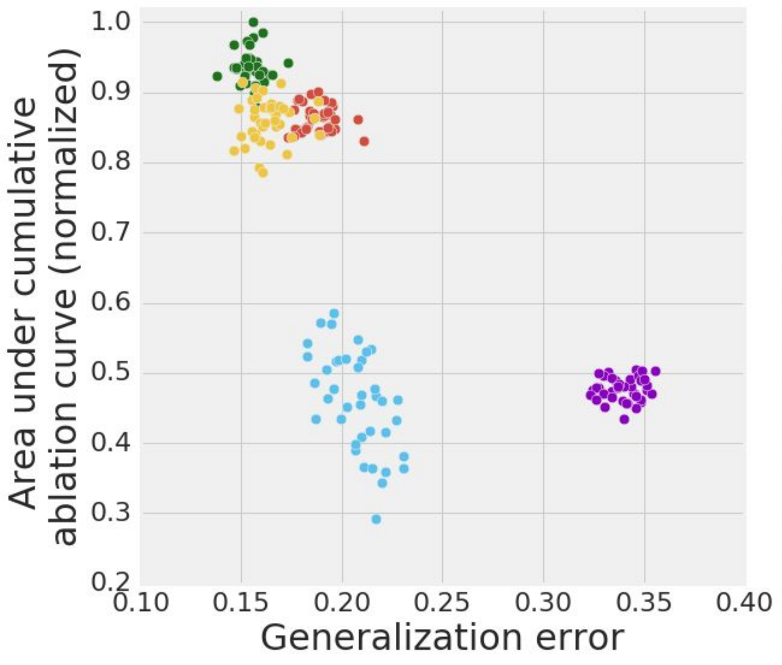
Wider networks converge to more similar solutions than narrow networks





Networks with similar performance learn diverse solutions





What have we learned?

- Networks which generalize converge to more similar solutions than those which memorize
    - There are many ways to memorize data, but few generalizable solutions
- Wider networks converge to more similar solutions than narrow networks
    - Consistent with the lottery ticket hypothesis
- Networks with identical topology and similar performance converge to highly diverse solutions, which can be recovered through two independent methods

What’s next?

- CCA enables us to find common directions across neural networks in a variety of settings
  - But what makes these directions special? Why are they consistently learned?
- In recurrent neural networks, how do representations change over the course of a sequence?
  - Are there stable and unstable components? What do these relate to?
- We found that networks which converge to similar solutions exhibit higher generalization performance
  - Can we use this insight to engineer a regularizer to improve network performance?

Acknowledgements

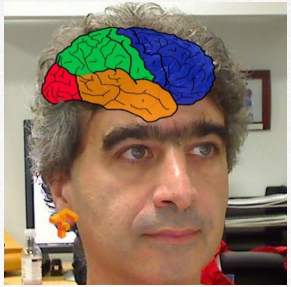
DeepMind

David Barrett

Neil Rabinowitz

Matt Botvinick

Google Brain



Samy Bengio

Maithra Raghu

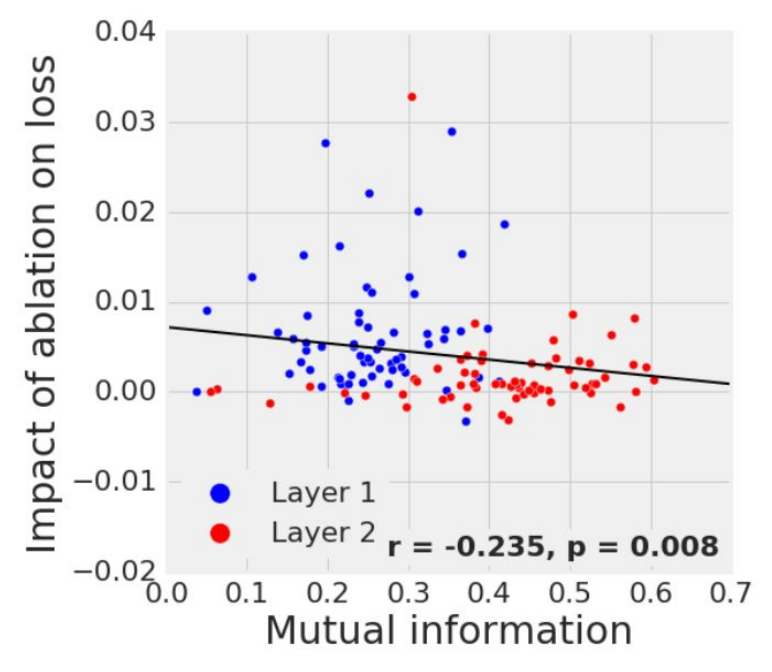
Harvard

Chris Harvey

Extra slides

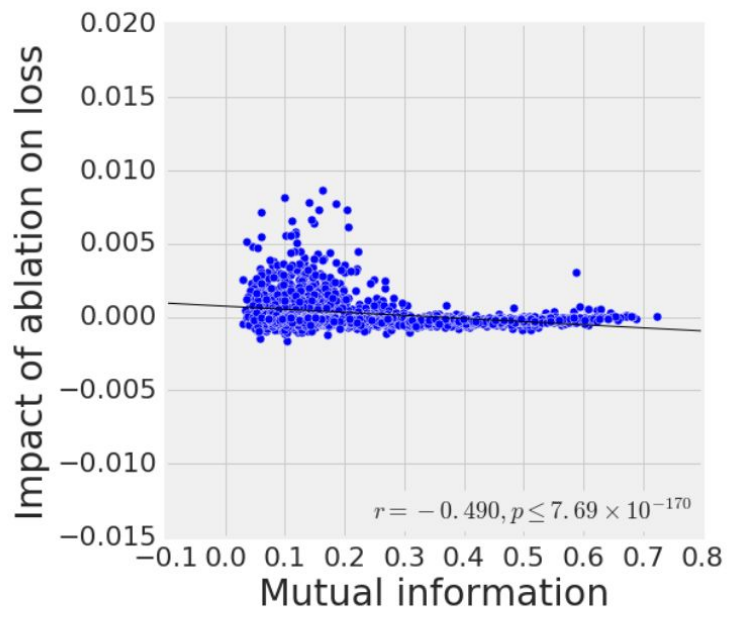
Is mutual information predictive of importance?

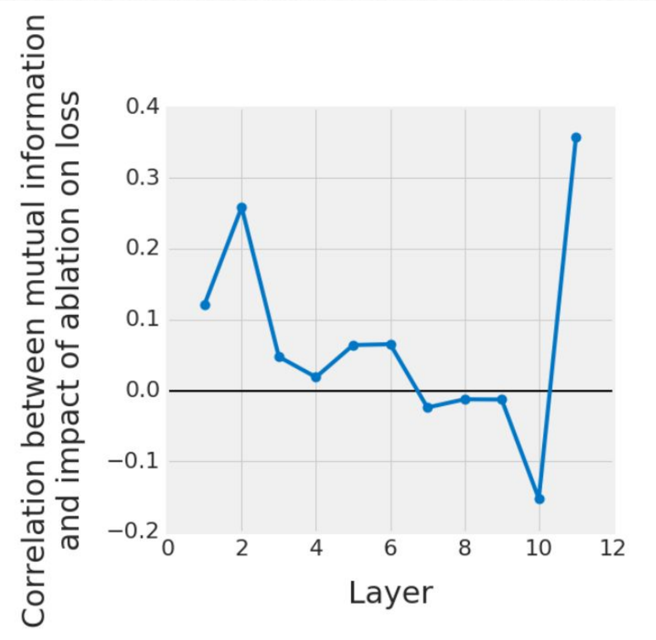
Mnist MLP



Is mutual information predictive of importance?

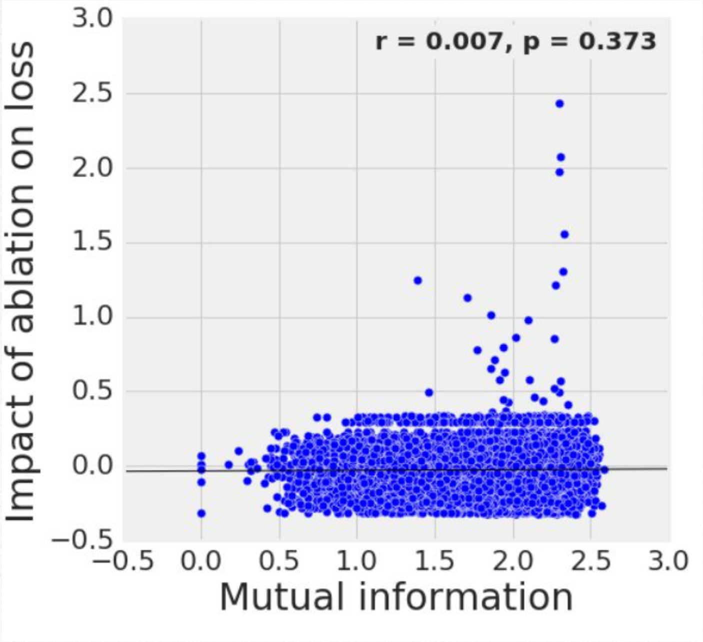
Cifar10 ConvNet

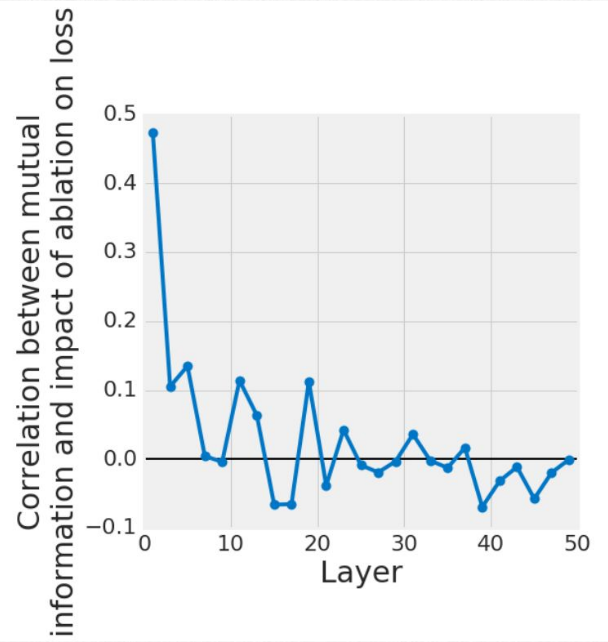




Is mutual information predictive of importance?

ImageNet resnet





RNNs exhibit bottom-up convergence dynamics