Theory for Deep Learning

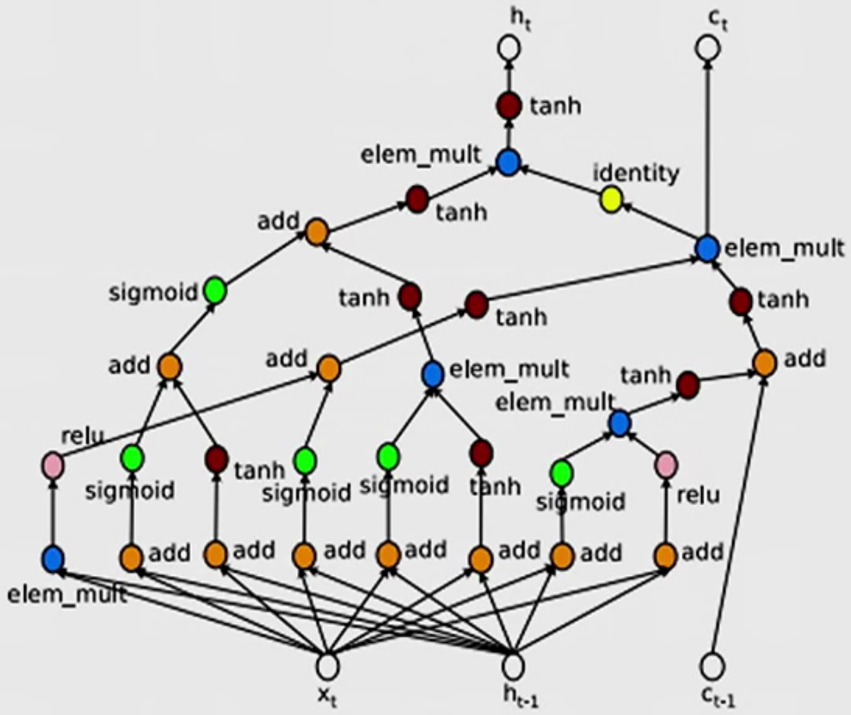
Arnu Pretorius

University of the Witwatersrand, 1 October 2018

Motivation

Why theory if we have AutoML?

Neural Architecture Search



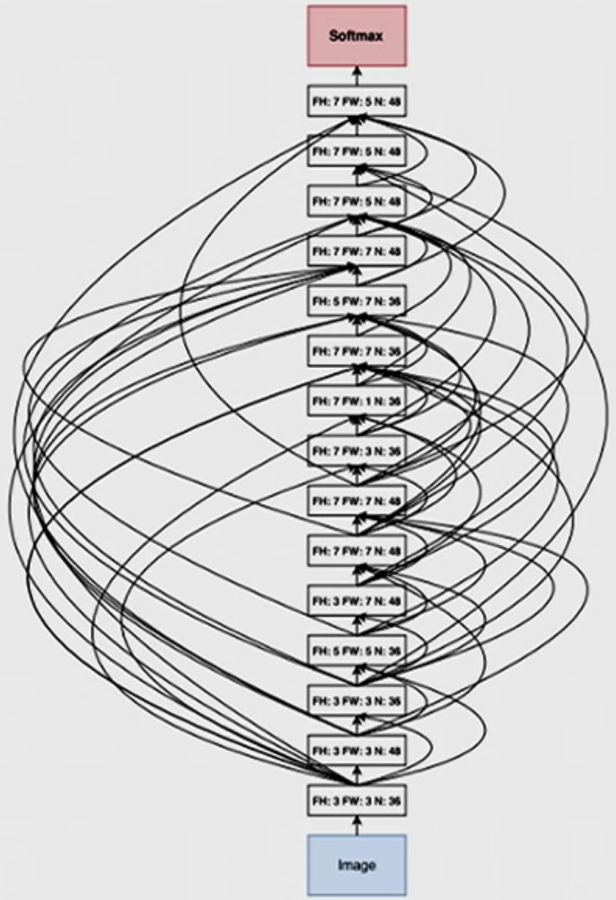


Figure 7: Convolutional architecture discovered by our method, when the search space does not have strides or pooling layers. FH is filter height, FW is filter width and N is number of filters.

- Improve understanding

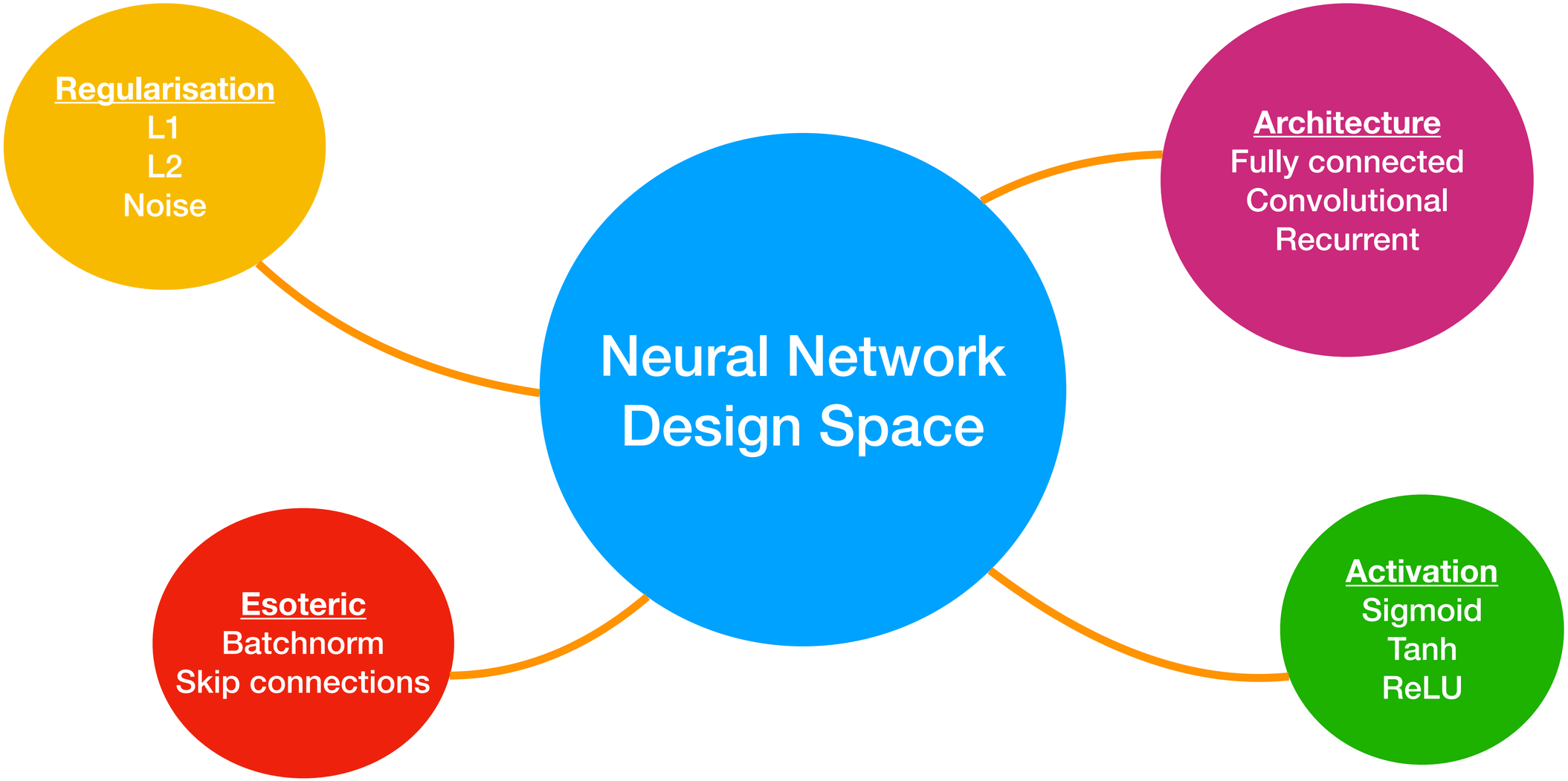
- Improve understanding
- Principled design

- Improve understanding
- Principled design
- Limit design search space

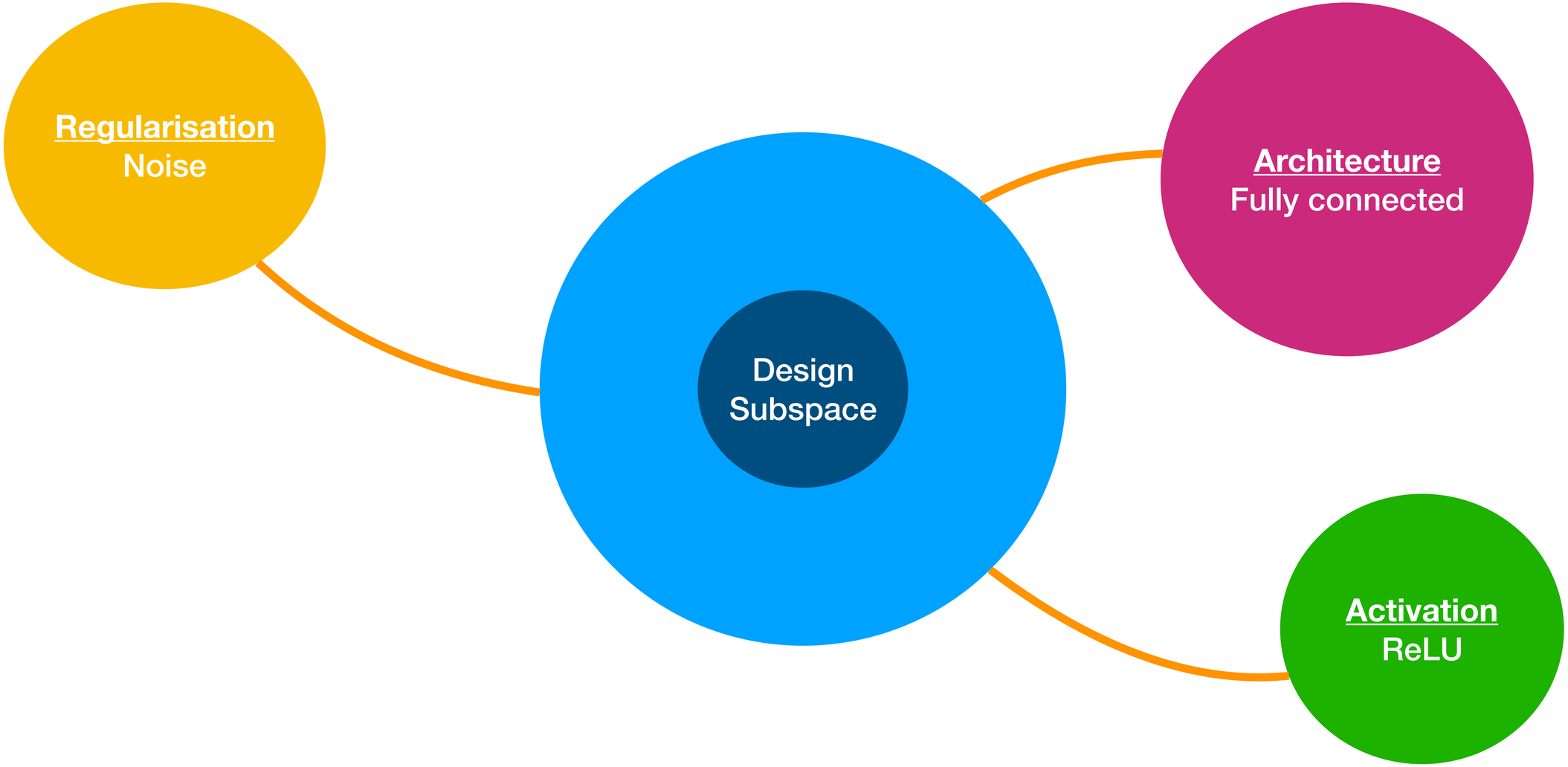
Our work

Joint with Elan Van Biljon, Steve Kroon & Herman Kamper

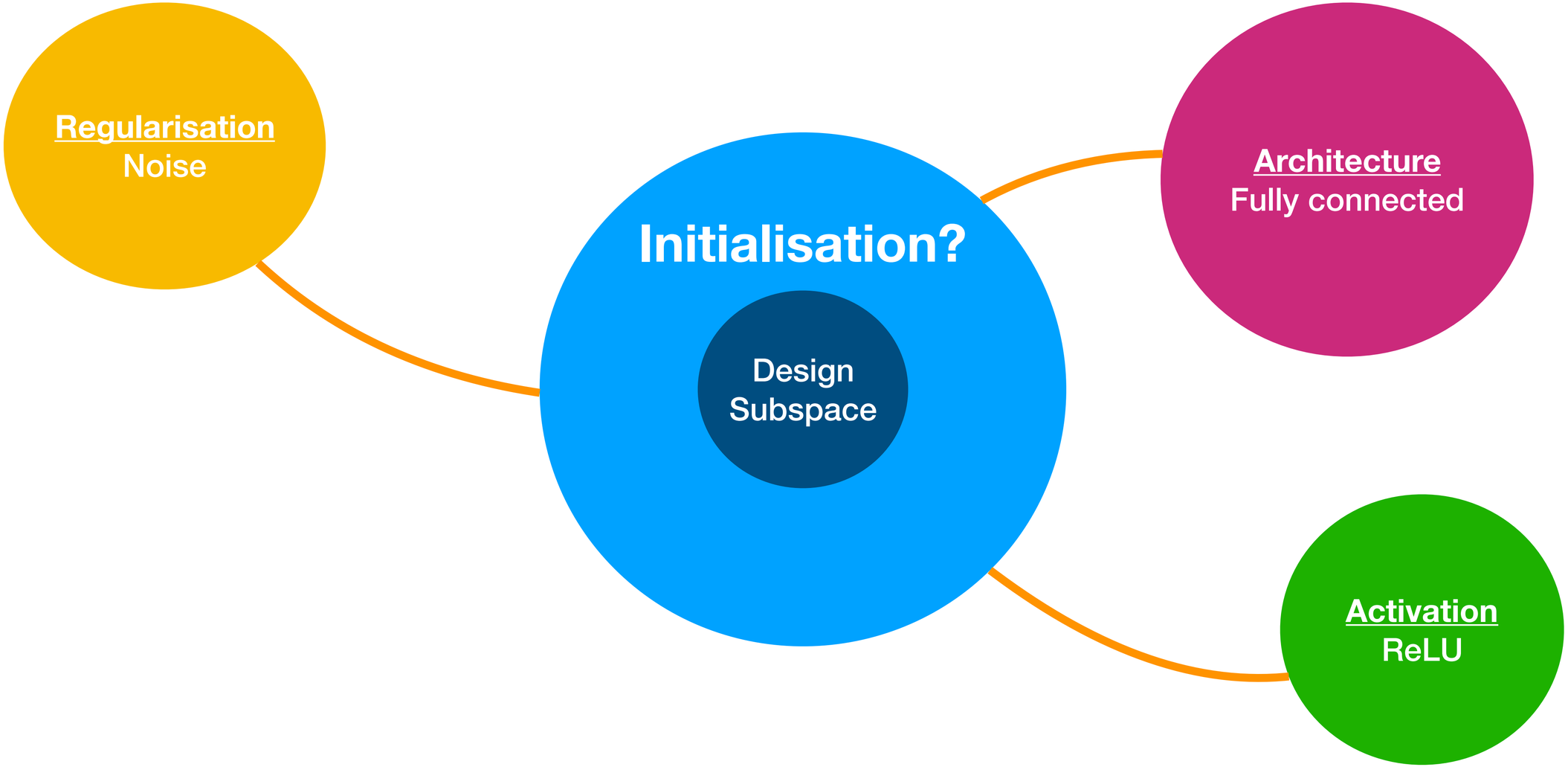
Design space



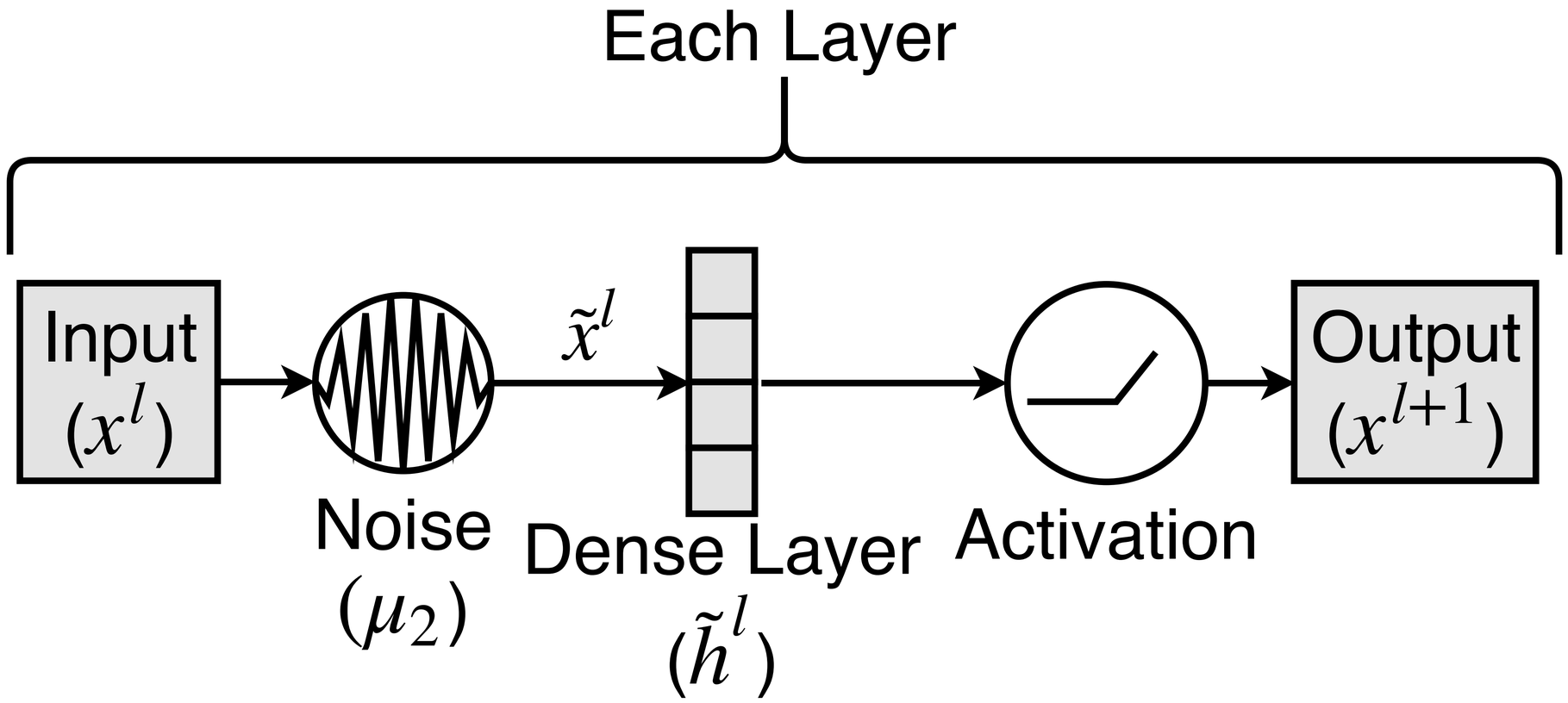
Chosen subspace



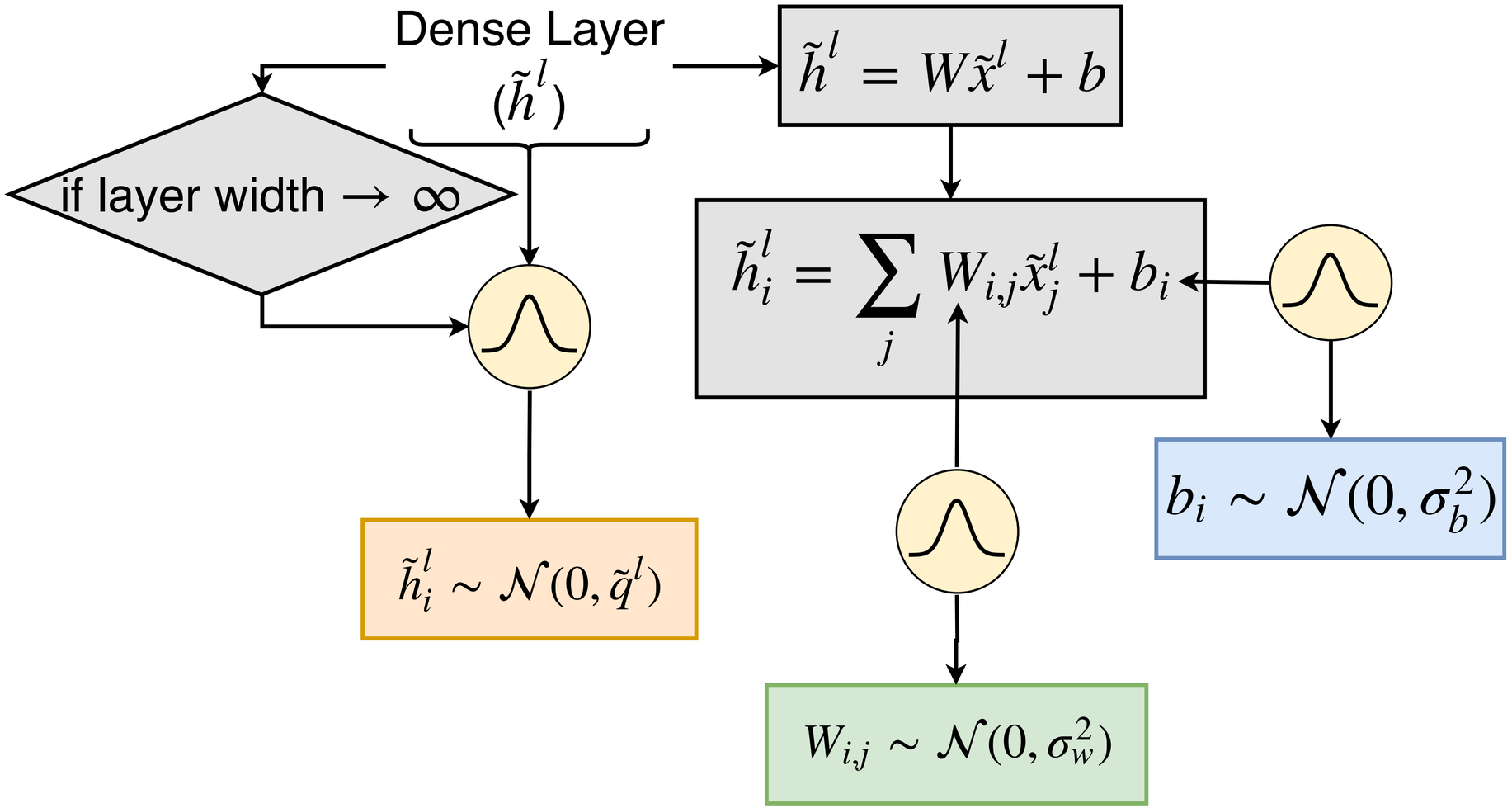
Chosen subspace



Noisy neural network model

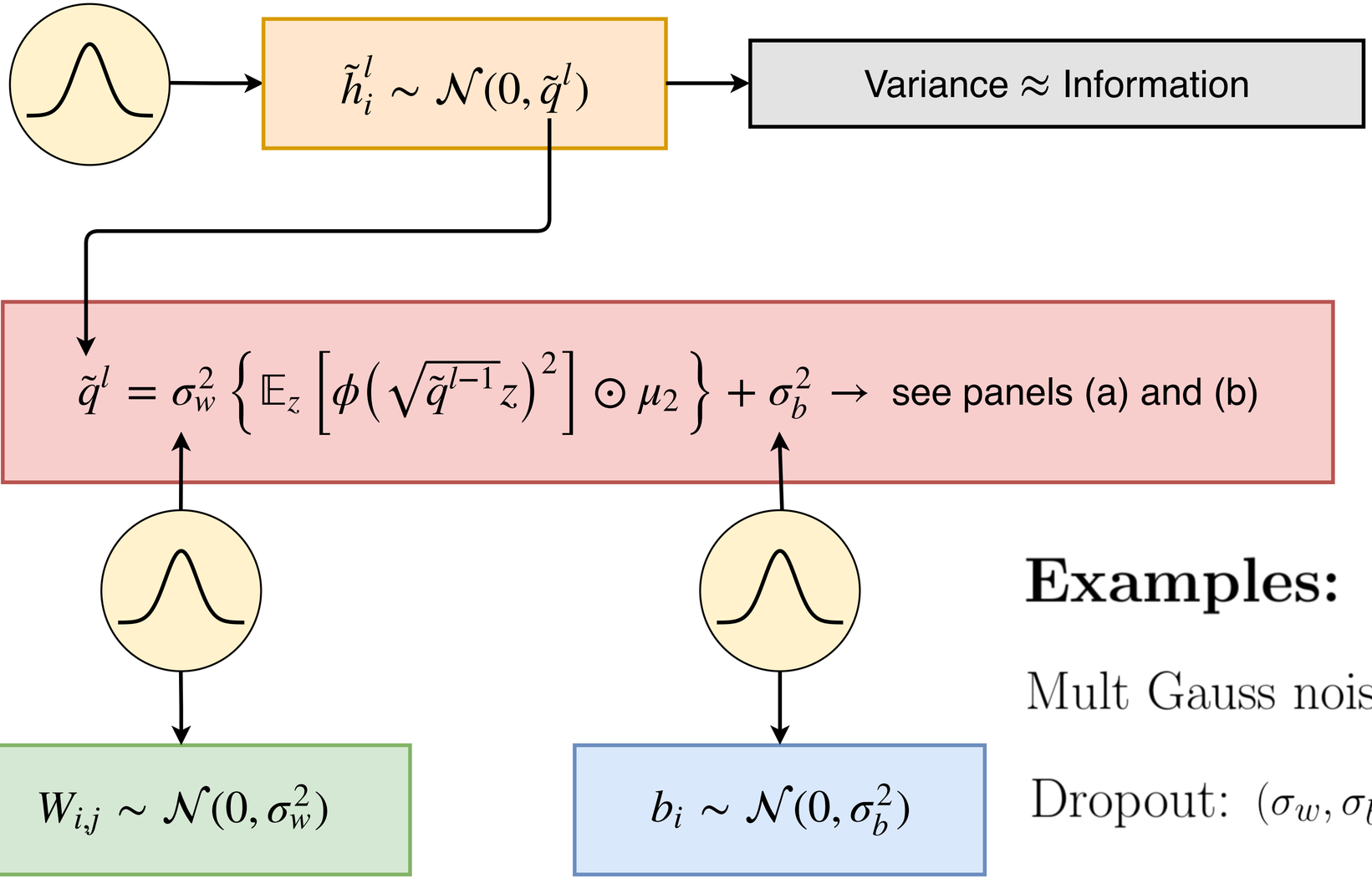


Mean field theory



Poole, Ben, et al. "Exponential expressivity in deep neural networks through transient chaos." *Advances in neural information processing systems*. 2016.

Variance dynamics



$$(\sigma_w, \sigma_b, \mu_2)$$

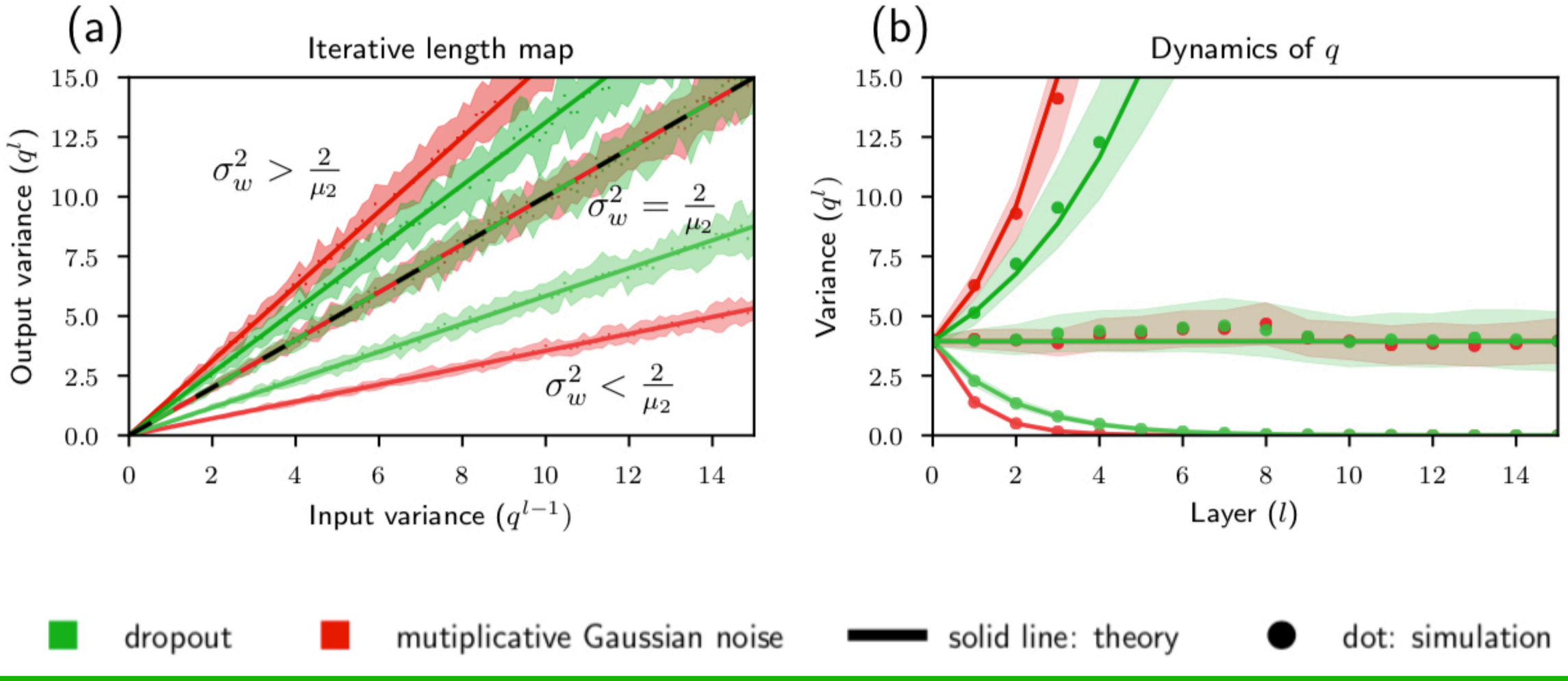
$$= \left(\sqrt{2/\mu_2}, 0, \mu_2 \right)$$

Examples:

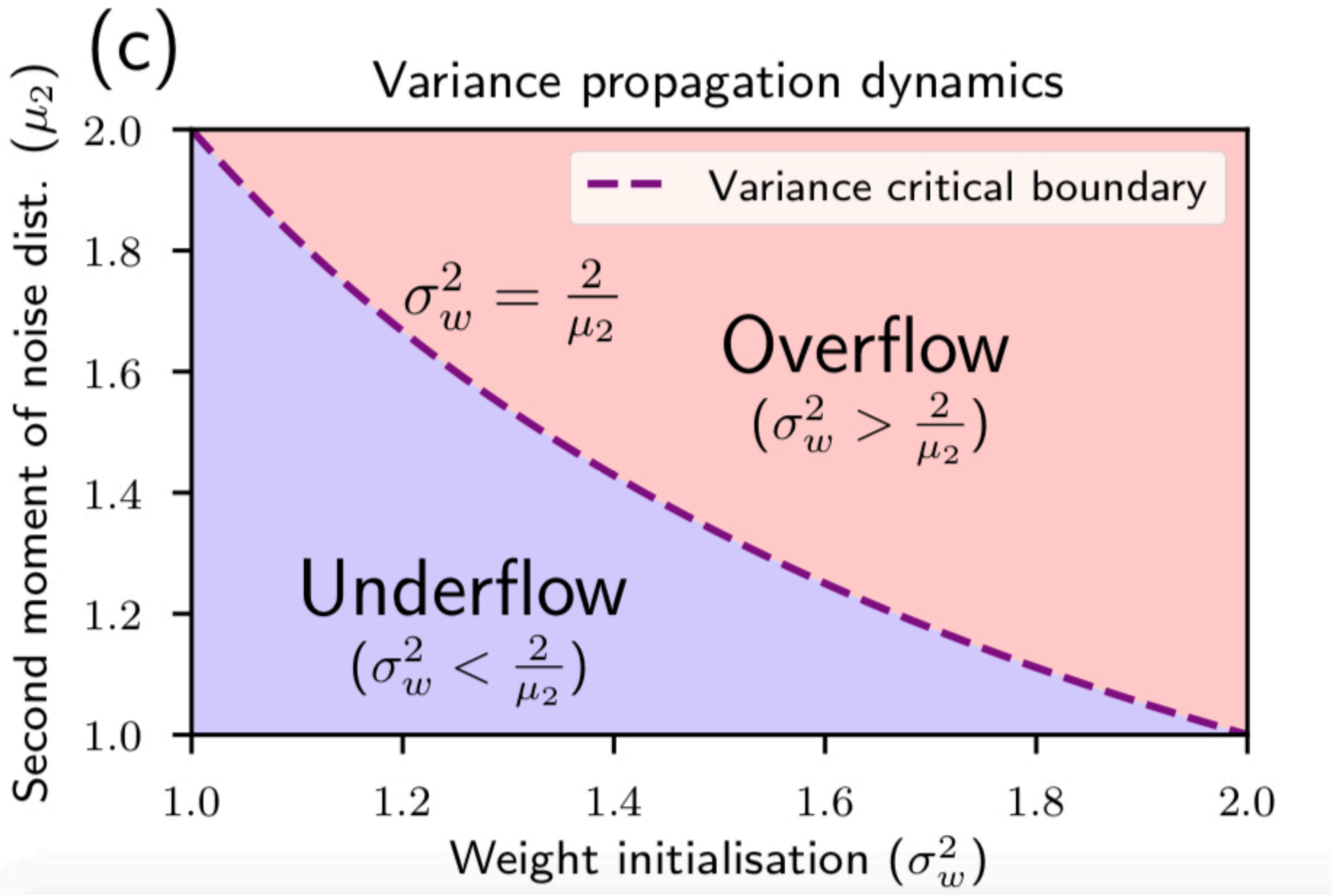
Mult Gauss noise: $(\sigma_w, \sigma_b) = \left(\sqrt{2/(\sigma^2 + 1)}, 0 \right)$

Dropout: $(\sigma_w, \sigma_b) = (\sqrt{2p}, 0)$

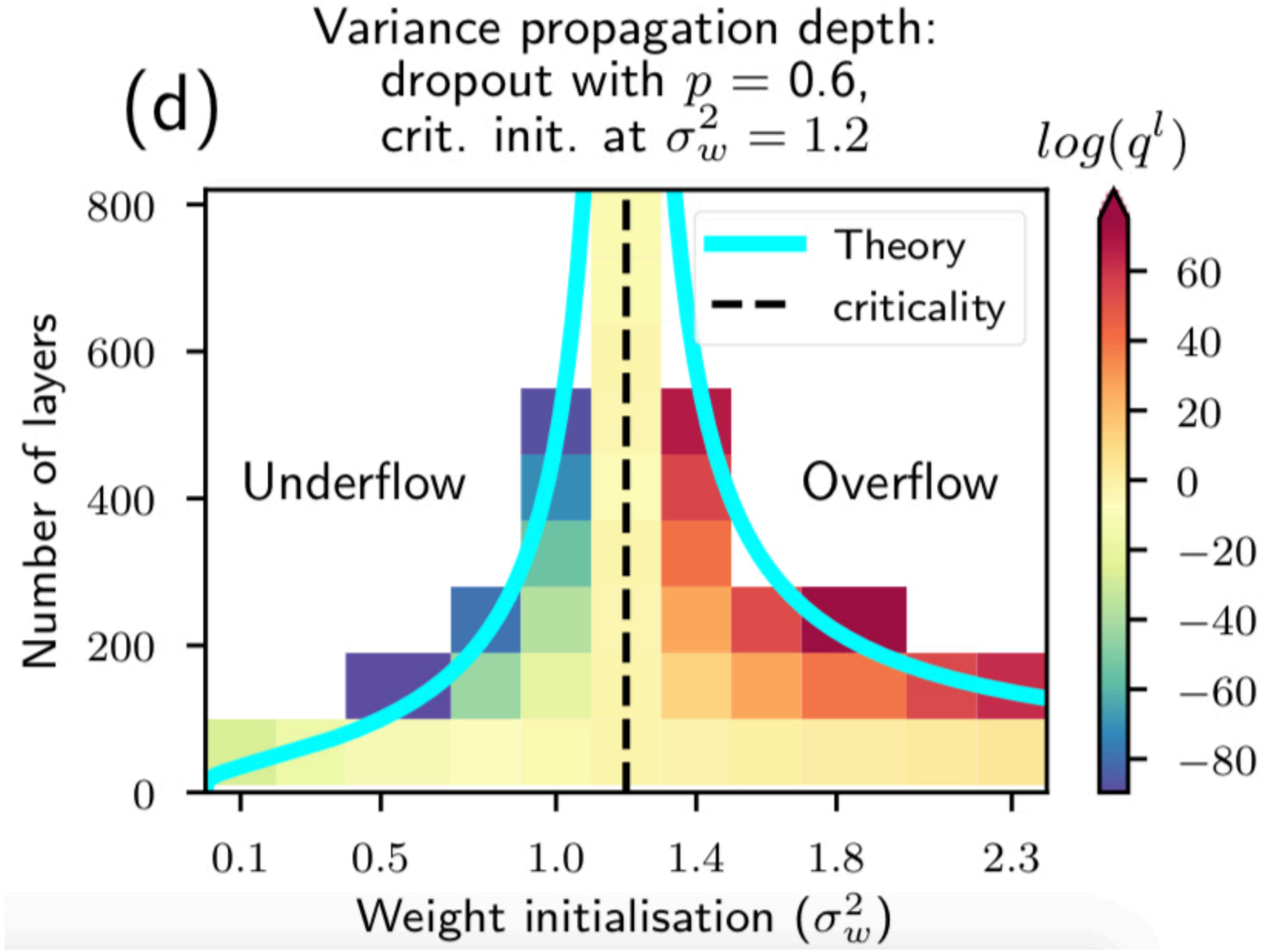
Variance dynamics



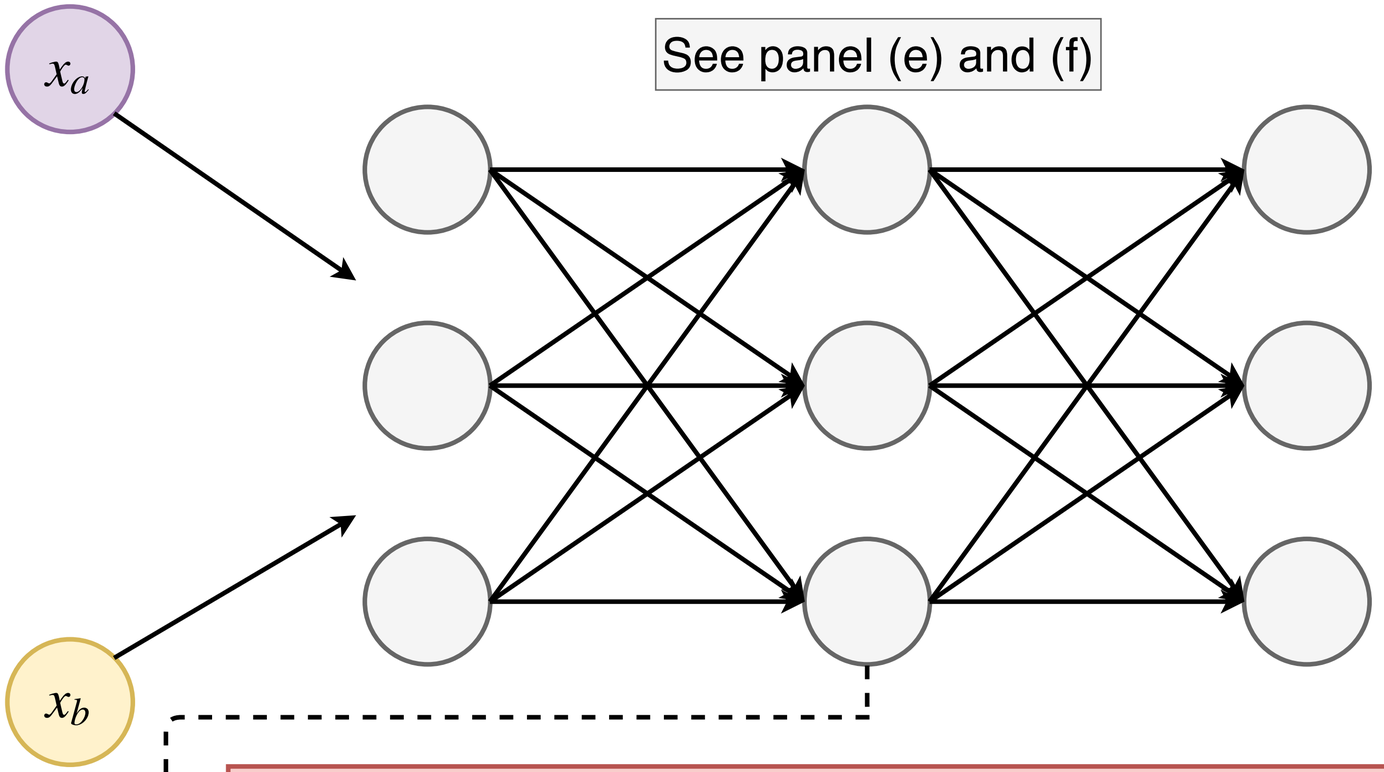
Variance dynamics



Variance dynamics

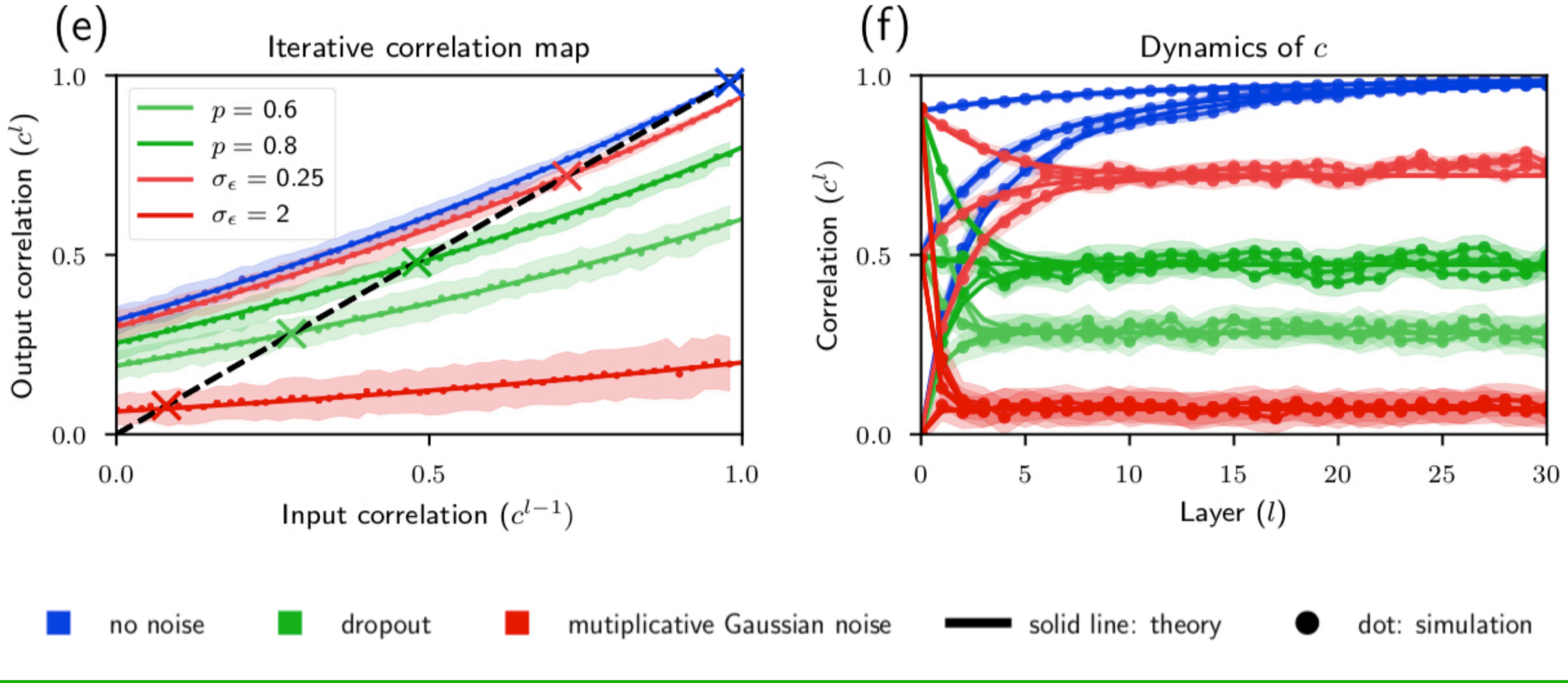


Correlation dynamics

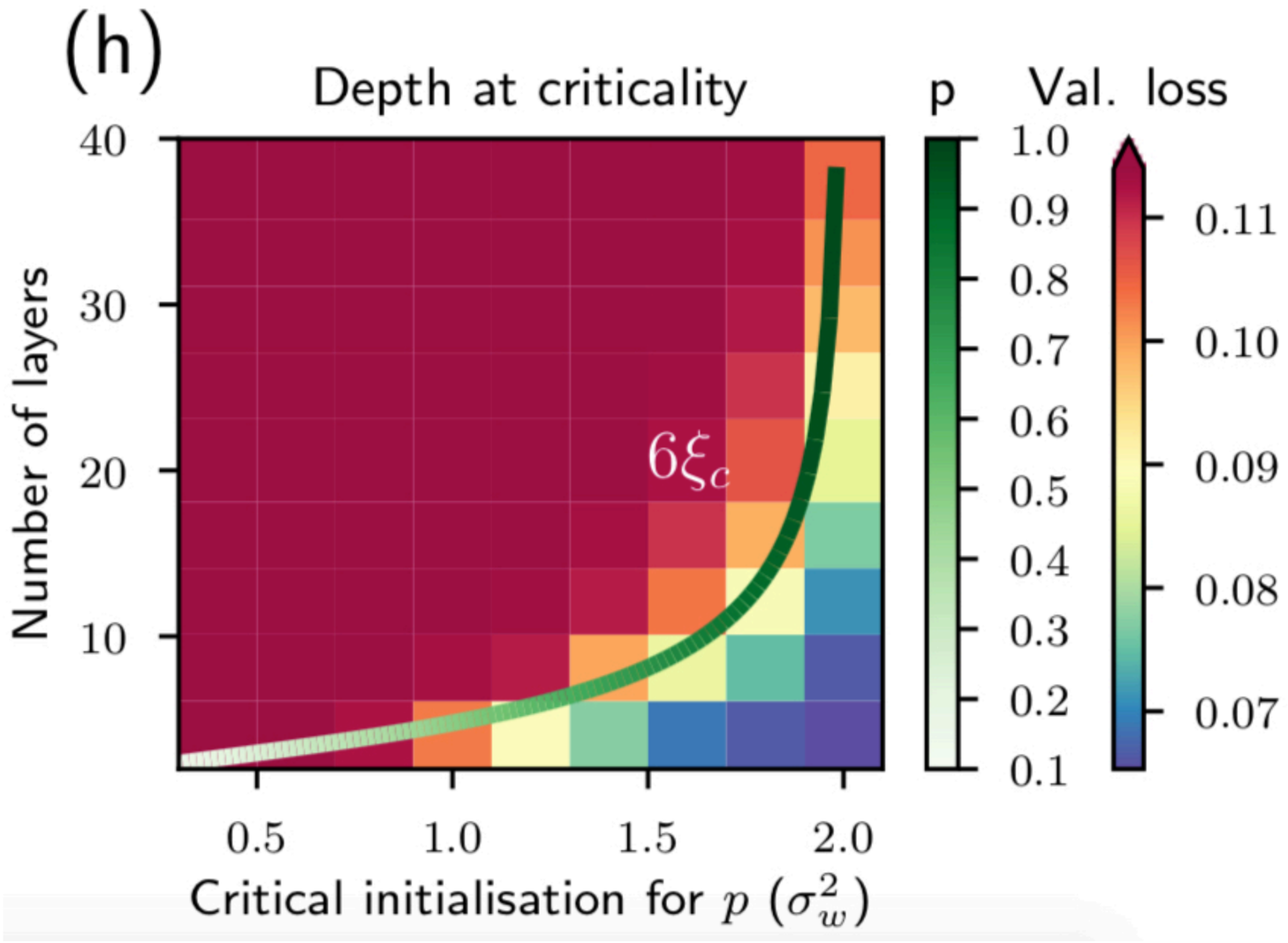


$$\tilde{c}^{l}_{a,b} = \frac{1}{\mu_2}\left\{\frac{\tilde{c}^{l-1}_{a,b}\arcsin\left(\tilde{c}^{l-1}_{a,b}\right) + \sqrt{1-\left(\tilde{c}^{l-1}_{a,b}\right)^2}}{\pi} + \frac{\tilde{c}^{l-1}_{a,b}}{2}\right\}$$

Correlation dynamics



Correlation dynamics



Takeaways

- Signal propagates using critical initialisation

- Signal propagates using critical initialisation
- But noise limits trainable depth

Future work

Joint with hopefully some of you :)

Is ReLU really the best?

No, actually ReLU is breaking things…

Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice

Jeffrey Pennington
Google Brain

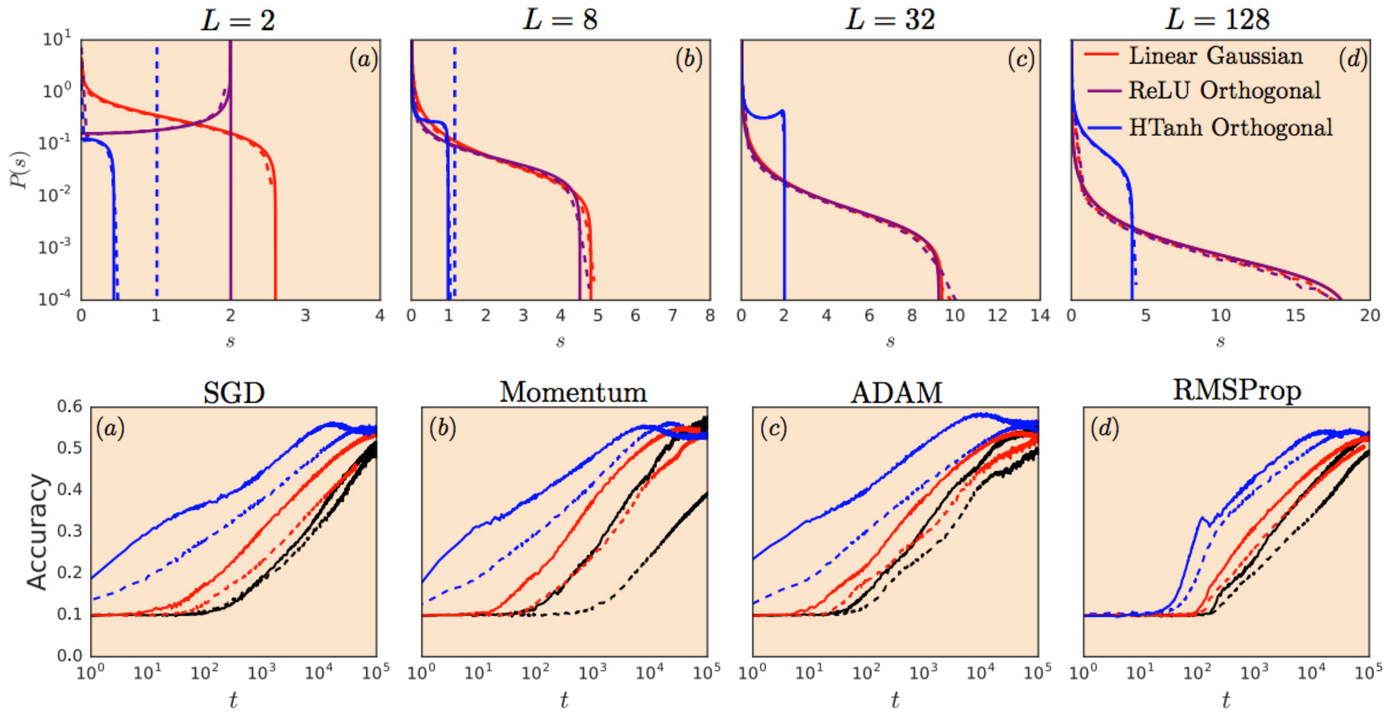
Samuel S. Schoenholz
Google Brain

Surya Ganguli
Applied Physics, Stanford University and Google Brain

Abstract

It is well known that weight initialization in deep networks can have a dramatic impact on learning speed. For example, ensuring the mean squared singular value of a network's input-output Jacobian is $O(1)$ is essential for avoiding exponentially vanishing or exploding gradients. Moreover, in deep linear networks, ensuring that *all* singular values of the Jacobian are concentrated near 1 can yield a dramatic additional speed-up in learning; this is a property known as dynamical isometry. However, it is unclear how to achieve dynamical isometry in nonlinear deep networks. We address this question by employing powerful tools from free probability theory to analytically compute the *entire* singular value distribution of a deep network's input-output Jacobian. We explore the dependence of the singular value distribution on the depth of the network, the weight initialization, and the choice of nonlinearity. Intriguingly, we find that ReLU networks are incapable of dynamical isometry. On the other hand, sigmoidal networks can achieve isometry, but only with orthogonal weight initialization. Moreover, we demonstrate empirically that deep nonlinear networks achieving dynamical isometry learn orders of magnitude faster than networks that do not. Indeed, we show that properly-initialized deep sigmoidal networks consistently outperform deep ReLU networks. Overall, our analysis reveals that controlling the *entire* distribution of Jacobian singular values is an important design consideration in deep learning.

$$\sigma^2_{JJ^T} = \frac{1-p(q^*)}{p(q^*)} L$$



Shifted ReLU to the rescue!

The Emergence of Spectral Universality in Deep Networks

Jeffrey Pennington
Google Brain

Samuel S. Schoenholz
Google Brain

Surya Ganguli
Google Brain
Applied Physics, Stanford University

Abstract

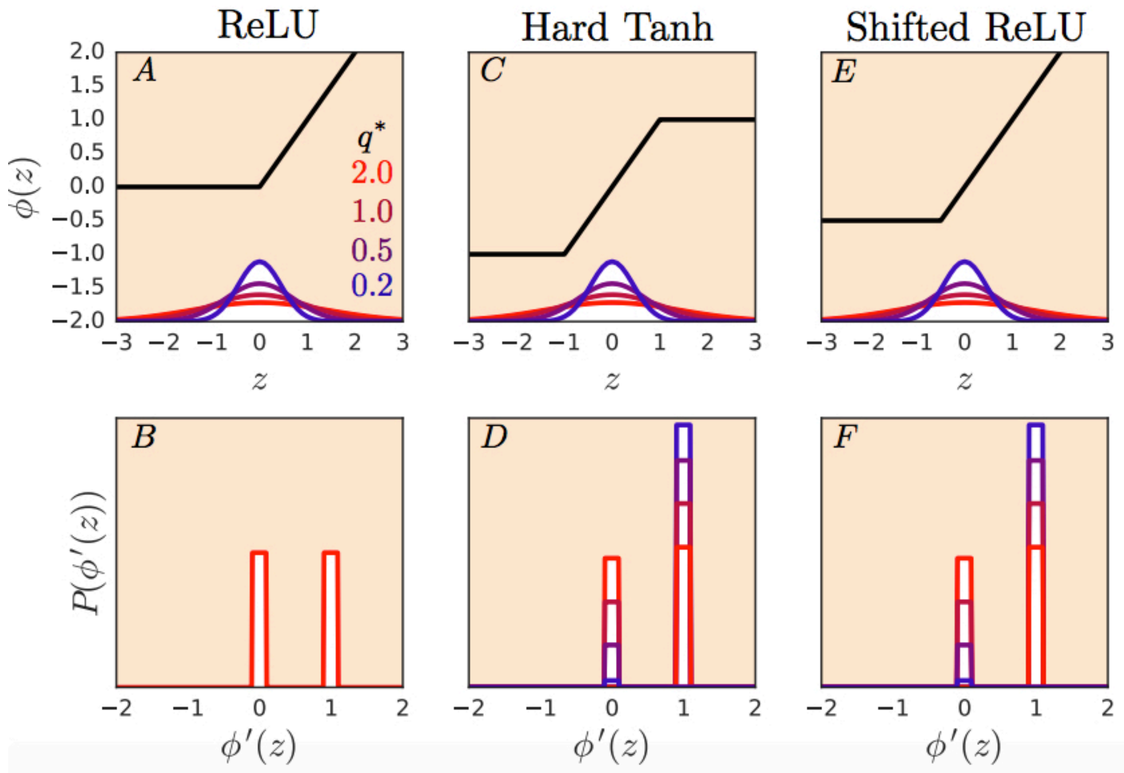
Recent work has shown that tight concentration of the *entire* spectrum of singular values of a deep network's input-output Jacobian around one at initialization can speed up learning by orders of magnitude. Therefore, to guide important design choices, it is important to build a full theoretical understanding of the spectra of Jacobians at initialization. To this end, we leverage powerful tools from free probability theory to provide a detailed analytic understanding of how a deep network's Jacobian spectrum depends on various hyperparameters including the nonlinearity, the weight and bias distributions, and the depth. For a variety of nonlinearities, our work reveals the emergence of new universal limiting spectral distributions that remain concentrated around one even as the depth goes to infinity.

can achieve depth-independent learning speeds, while the corresponding Gaussian initializations cannot [2].

Recently, it was shown [3] that a similarly well-conditioned Jacobian could be constructed for deep non-linear networks using a combination of orthogonal weights and tanh nonlinearities. The result of this improved conditioning was an orders-of-magnitude speedup in learning for tanh networks. However, the same study also proved that a well-conditioned Jacobian could not be achieved with Rectified Linear units (ReLUs). Together these results explained why, historically, in some cases orthogonal weight initialization had been found to improve training efficiency only slightly [4].

These empirical results connecting the conditioning of the Jacobian to a dramatic speedup in learning raise an important theoretical question. Namely, how does the entire shape of this spectrum depend on a network's nonlinearity, weight and bias distribution, and depth? Here we provide a detailed analytic answer by using powerful tools from free probability theory. Our answer provides theoretical guidance on how to choose

$$\phi(x) = [x + \tfrac{1}{2}]_+ - \tfrac{1}{2}$$



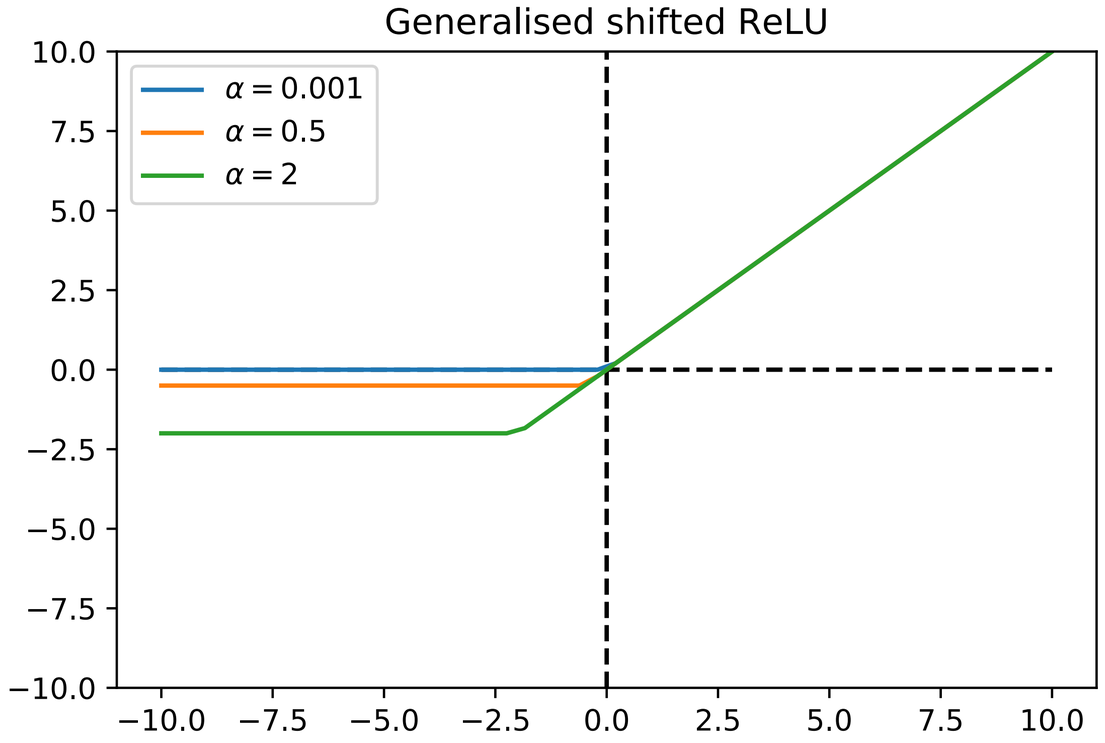
Generalised shifted ReLU

$$\phi(x) = \mathbf{max}(0, x + \alpha) - \alpha$$

$$\phi'(x) \sim \mathbf{Bern}(p(q^*))$$

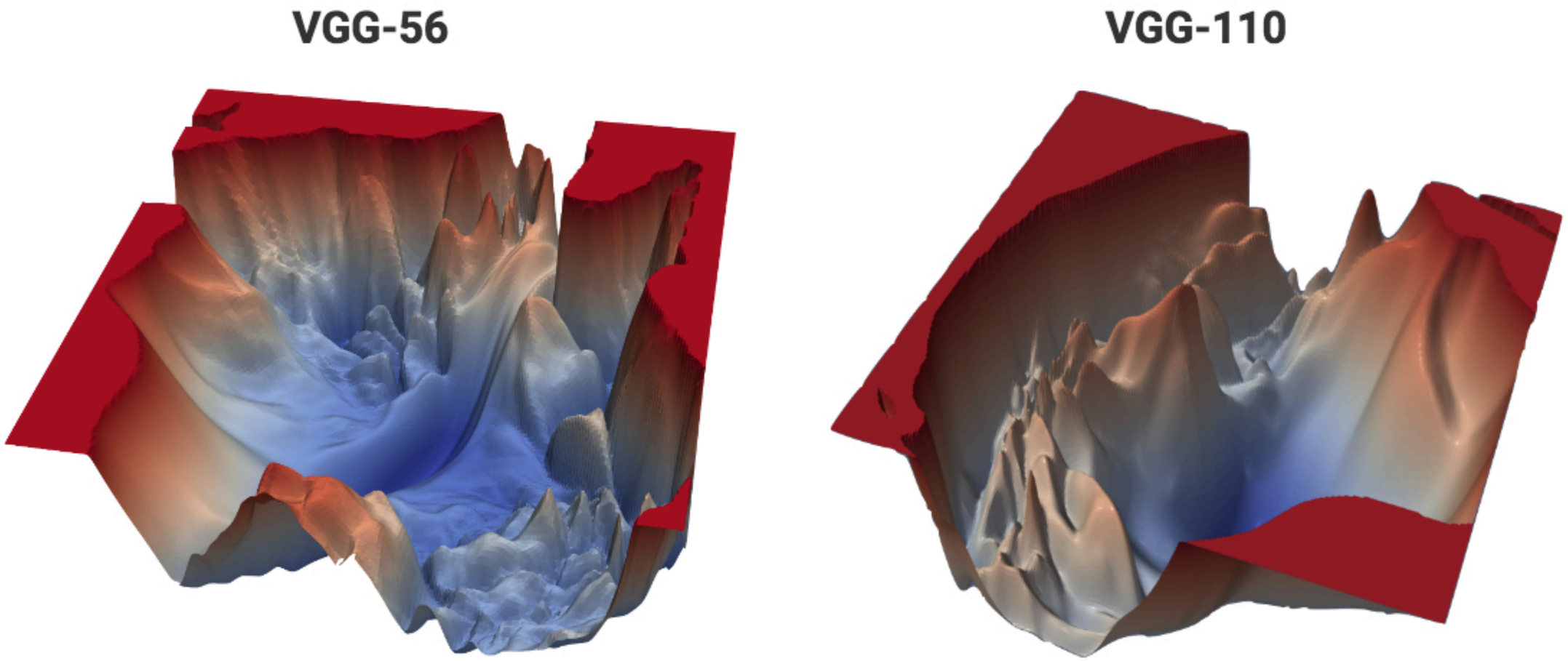
$$p(q^*) = \frac{1}{2}\mathbf{erfc}\left(-\frac{\alpha}{\sqrt{2q^*}}\right)$$

$$\mathbf{erfc}(a) = \frac{2}{\sqrt{\pi}}\int_{-a}^{\infty} e^{-t^2} dt$$



But what happens after initialisation?

Compare initialisations



Li, Hao, et al. "Visualizing the loss landscape of neural nets." *arXiv preprint arXiv:1712.09913* (2017).

Compare initialisations

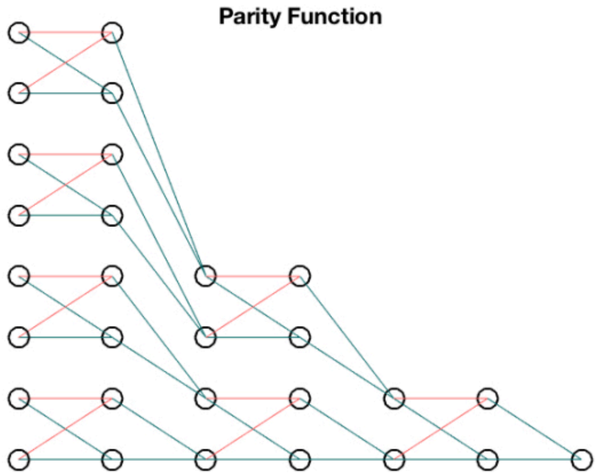
ARE EFFICIENT DEEP REPRESENTATIONS LEARNABLE?

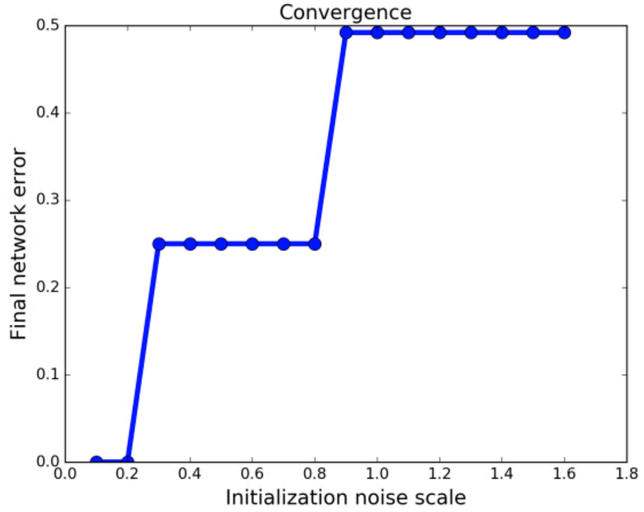
Maxwell Nye
Massachusetts Institute of Technology
mnye@mit.edu

Andrew Saxe
Harvard University
asaxe@fas.harvard.edu

ABSTRACT

Many theories of deep learning have shown that a deep network can require dramatically fewer resources to represent a given function compared to a shallow network. But a question remains: can these efficient representations be *learned* using current deep learning techniques? In this work, we test whether standard deep learning methods can in fact find the efficient representations posited by several theories of deep representation. Specifically, we train deep neural networks to learn two simple functions with known efficient solutions: the parity function and the fast Fourier transform. We find that using gradient-based optimization, a deep network does not learn the parity function, unless initialized very close to a hand-coded exact solution. We also find that a deep linear neural network does not learn the fast Fourier transform, even in the best-case scenario of infinite training data, unless the weights are initialized very close to the exact hand-coded solution. Our results suggest that not every element of the class of compositional functions can be learned efficiently by a deep network, and further restrictions are necessary to understand what functions are both efficiently representable and learnable.





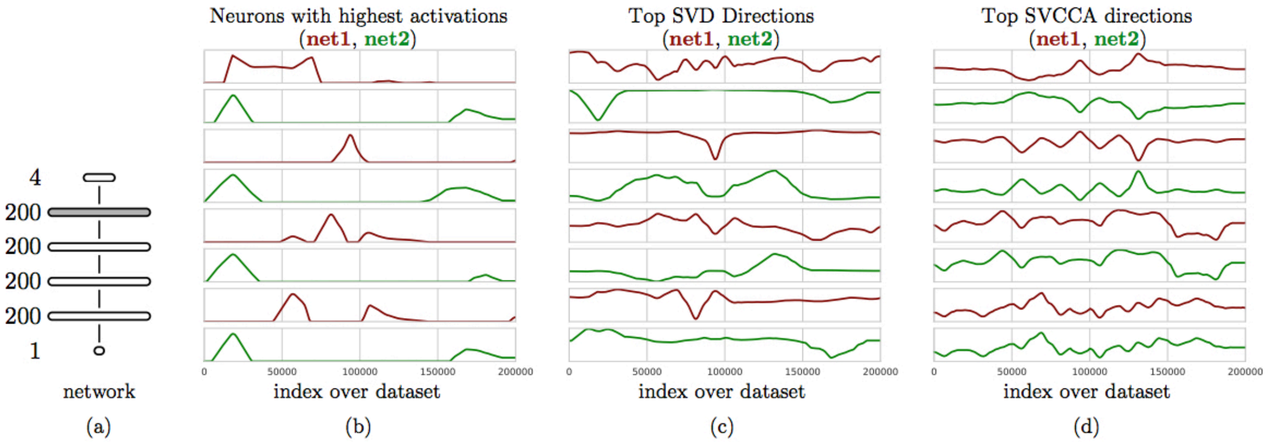
Compare initialisations

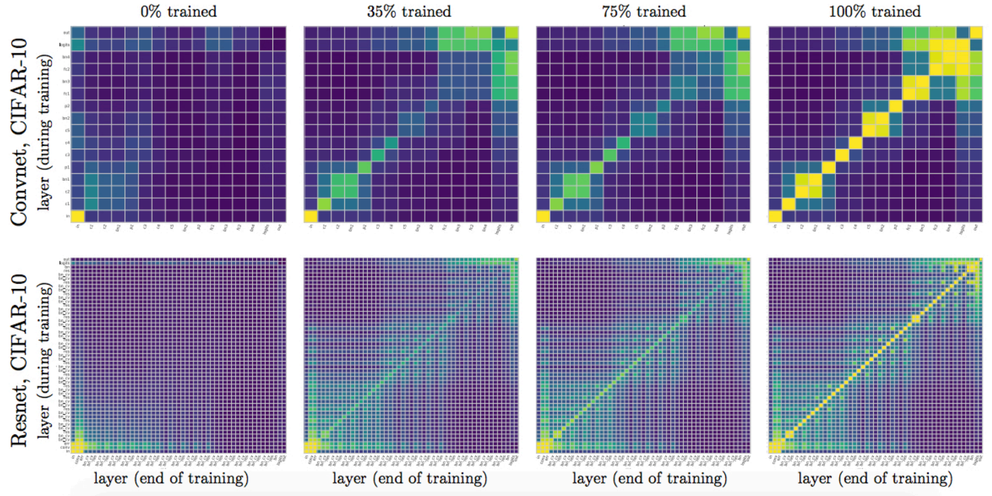
SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability

Maithra Raghu,[1,2] **Justin Gilmer,**[1] **Jason Yosinski,**[3] **& Jascha Sohl-Dickstein**[1]
[1]Google Brain [2]Cornell University [3]Uber AI Labs
maithrar@gmail.com, gilmer@google.com, yosinski@uber.com, jaschasd@google.com

Abstract

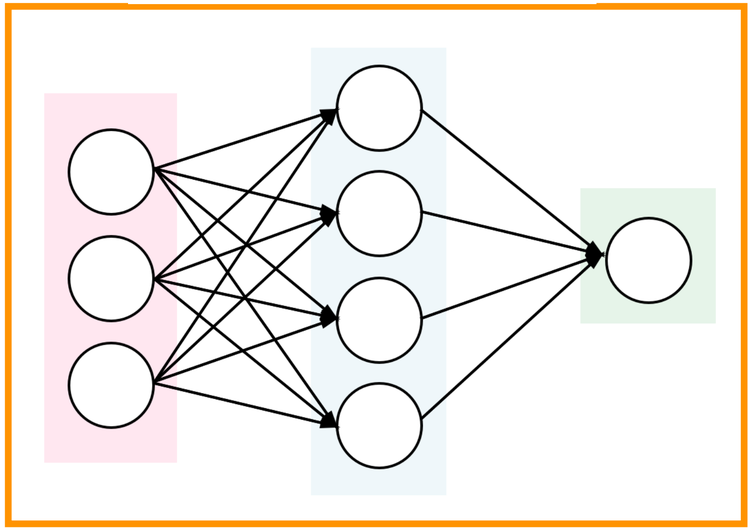
We propose a new technique, Singular Vector Canonical Correlation Analysis (SVCCA), a tool for quickly comparing two representations in a way that is both invariant to affine transform (allowing comparison between different layers and networks) and fast to compute (allowing more comparisons to be calculated than with previous methods). We deploy this tool to measure the intrinsic dimensionality of layers, showing in some cases needless over-parameterization; to probe learning dynamics throughout training, finding that networks converge to final representations from the bottom up; to show where class-specific information in networks is formed; and to suggest new training regimes that simultaneously save computation and overfit less.





Tentative plan

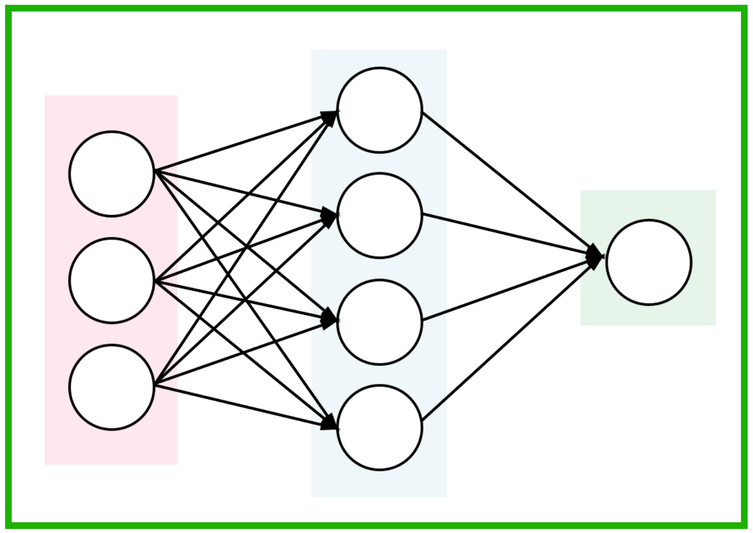
Optimal solution



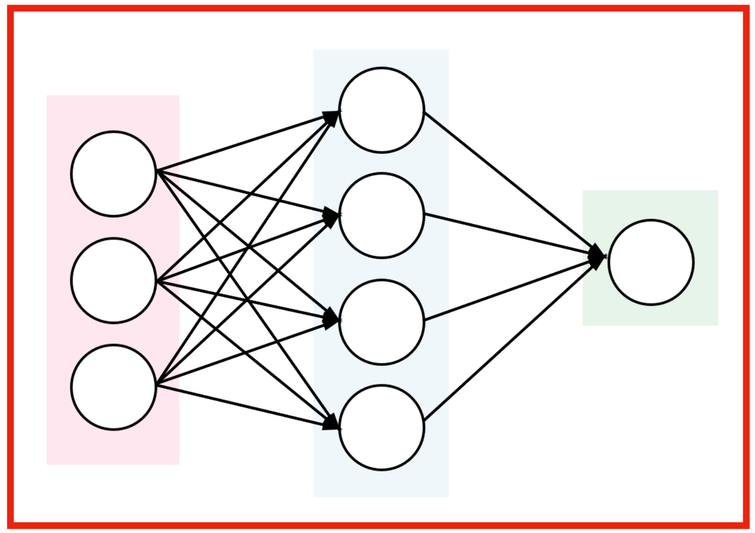
SVCCA

SVCCA

Critical initialisation



Random initialisation



ReLU
vs
Generalised shifted ReLU

Thanks!
Questions?