Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent

Jaehoon Lee*, Lechao Xiao*, Samuel S. Schoenholz, Yasaman Bahri,
Roman Novak, Jascha Sohl-Dickstein, Jeffrey Pennington

*Equal contribution, work done as part of Google AI Residency, g.co/airesidency

NeurIPS 2019
arXiv: 1902.06720

Is Training Dynamics Tractable?

A longstanding goal in deep learning research has been to precisely characterize training and generalization. However, the often complex loss landscapes of neural networks have made a theory of learning dynamics elusive. In this work, we show that wide neural networks :

- **Evolve like linear models:** The learning dynamics simplify considerably and that, in the infinite width limit, they are governed by a linearized model obtained from the first-order Taylor expansion of the network around its initial parameters
- **Have solvable dynamics:** The dynamics of gradient descent become analytically tractable. One can *solve* the time evolution dynamics of the neural networks *without* training.
- **Approximate Gaussian processes**: For squared loss, the exact learning dynamics admit a closed-form solution that allows us to characterize the evolution of the predictive distribution in terms of a Gaussian process with tractable mean and variance.

Motivation: Gradient descent learning dynamics of neural networks is intractable

- Loss function of neural networks

$$\mathcal{L} = \sum_{(x,y)\in\mathcal{D}} \ell(f_t(x,\theta), y).$$

- Gradient descent (flow) dynamics of the parameters and outputs (logits)

$$\dot{\theta}_t = -\eta \nabla_\theta f_t(\mathcal{X})^T \nabla_{f_t(\mathcal{X})}\mathcal{L}$$
$$\dot{f}_t(\mathcal{X}) = \nabla_\theta f_t(\mathcal{X})\,\dot{\theta}_t = -\eta\,\hat{\Theta}_t(\mathcal{X},\mathcal{X})\nabla_{f_t(\mathcal{X})}\mathcal{L}$$

- With a time evolving tangent kernel

$$\hat{\Theta}_t = \nabla_\theta f_t(\mathcal{X})\nabla_\theta f_t(\mathcal{X})^T$$

- The ODEs are complicated in general!

Linearized dynamics

- When the network is sufficiently wide, approximate neural network by its first order Taylor expansion at initialization (t=0)

$$\boxed{f_t^{\text{lin}}(x) \equiv f_0(x) + \nabla_\theta f_0(x)\,\omega_t}$$

- The linearized dynamics become

$$\dot{\omega}_t = -\eta \nabla_\theta f_0(\mathcal{X})^T \nabla_{f_t^{\text{lin}}(\mathcal{X})}\mathcal{L}$$
$$\dot{f}_t^{\text{lin}}(x) = -\eta\,\hat{\Theta}_0(x,\mathcal{X})\nabla_{f_t^{\text{lin}}(\mathcal{X})}\mathcal{L}\,.$$

- The kernel does not evolve with time, the ODEs are solvable. For squared loss, there are closed form solutions

$$\omega_t = -\nabla_\theta f_0(\mathcal{X})^T \hat{\Theta}_0^{-1}(I - e^{-\eta\hat{\Theta}_0 t})(f_0(\mathcal{X}) - \mathcal{Y})$$
$$f_t^{\text{lin}}(\mathcal{X}) = (I - e^{-\eta\hat{\Theta}_0 t})\mathcal{Y} + e^{-\eta\hat{\Theta}_0 t} f_0(\mathcal{X})$$

- For unseen input, the prediction is the sum of two terms

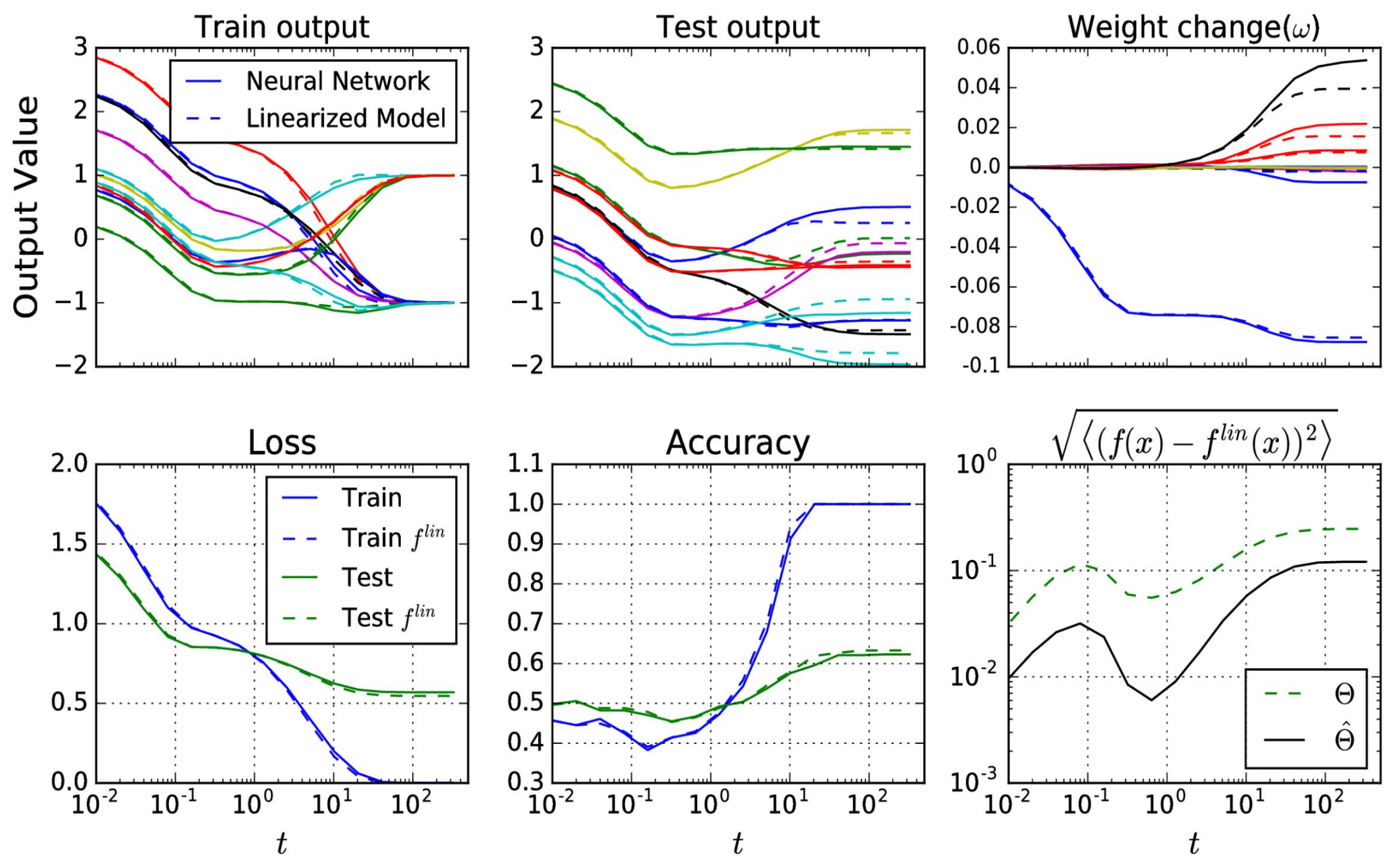
$$\mu_t(x) = \hat{\Theta}_0(x,\mathcal{X})\hat{\Theta}_0^{-1}(I - e^{-\eta\hat{\Theta}_0 t})\mathcal{Y} \qquad \text{deterministic}$$
$$\gamma_t(x) = f_0(x) - \hat{\Theta}_0(x,\mathcal{X})\hat{\Theta}_0^{-1}(I - e^{-\eta\hat{\Theta}_0 t})f_0(\mathcal{X}) \qquad \text{stochastic}$$

- Let the width go to infinity, the prediction of any input converges a Gaussian with mean and variance

$$\mu(x) = \Theta(x,\mathcal{X})\Theta^{-1}(I - e^{-\eta\Theta t})\mathcal{Y}$$
$$\Sigma(x) = \mathcal{K}(x,x) - 2\Theta(x,\mathcal{X})\Theta^{-1}(I - e^{-\eta\Theta t})\mathcal{K}(\mathcal{X},x)$$
$$+ \Theta(x,\mathcal{X})\Theta^{-1}(I - e^{-\eta\Theta t})\mathcal{K}\Theta^{-1}(I - e^{-\eta\Theta t})\Theta(\mathcal{X},x)$$

Experiments

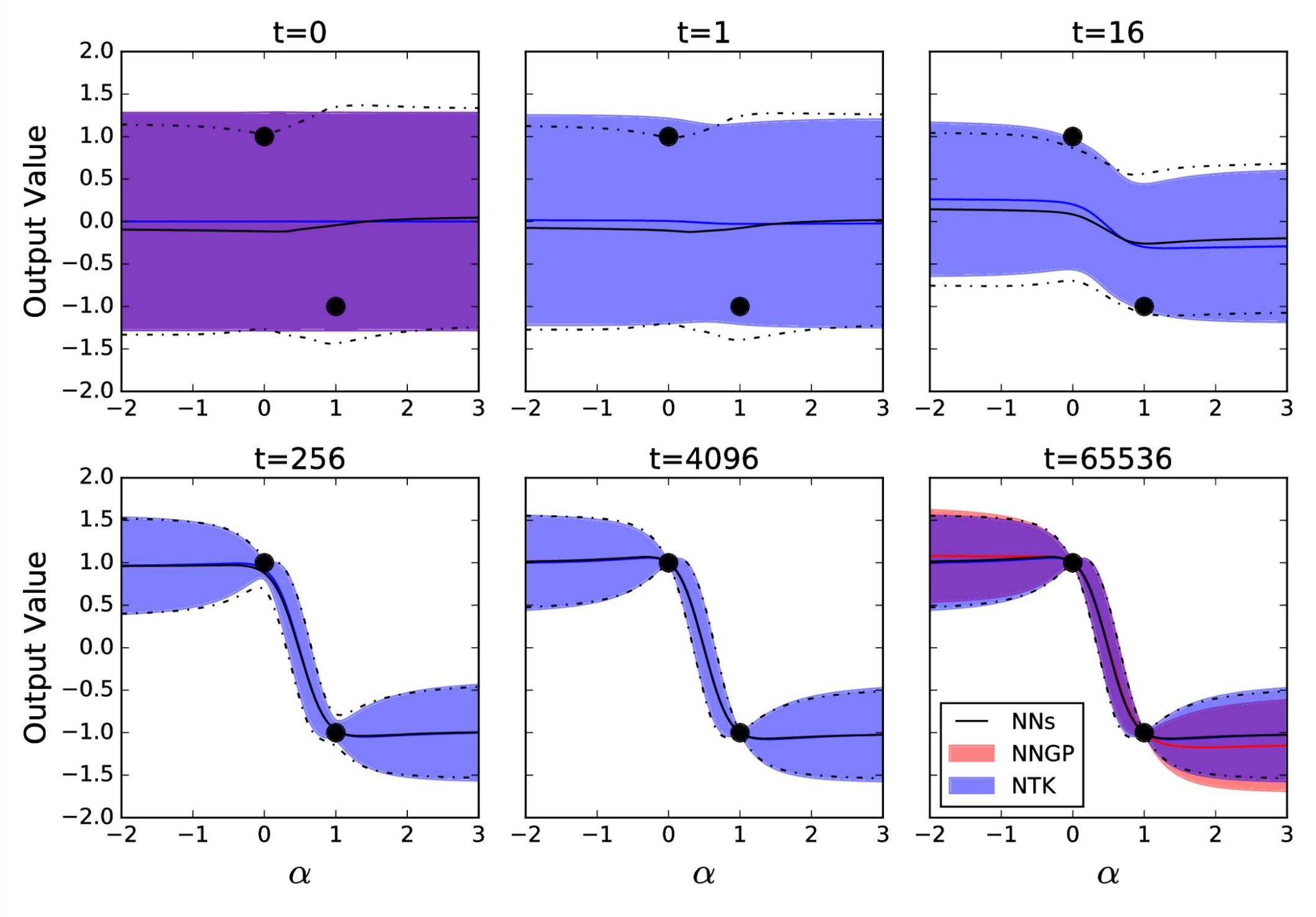
Full batch gradient descent on a model behaves similarly to analytic dynamics on its linearization



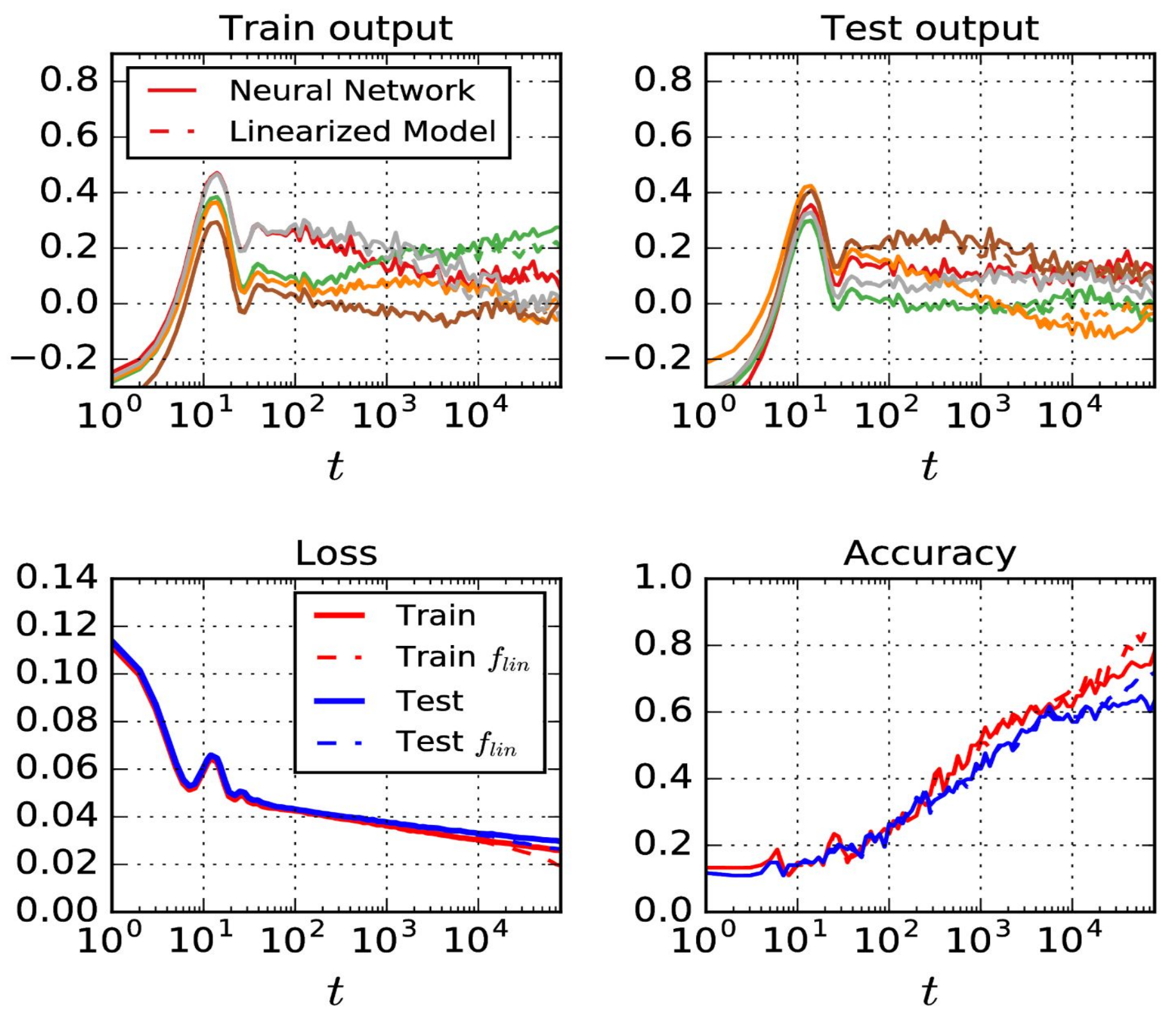
Left: A binary CIFAR classification task with MSE loss and a ReLU fully connected network with 5 hidden layers of width 2048, dataset size 256. First row: a randomly selected subset of data-points or parameters. Second row: loss and accuracy dynamics.

Right: Black lines indicate the time evolution of the predictive output distribution from an ensemble of 100 trained neural networks (NNs). The blue region indicates the analytic prediction of the output distribution throughout training. The trained network has 3 layers with width 8092.

Dynamics of mean and variance of trained neural network outputs follow analytic dynamics from linearization



A wide residual network and its linearization behave similarly



Left: A wide residual network (N=1, channel size=1024) and its linearization behave similarly when both are trained by SGD with momentum on MSE loss on full CIFAR-10. Both linearized and original model are trained directly using stochastic minibatching with batch size 8. Output dynamics for a randomly selected subset of train and test points are shown in the first row. The second row shows training and accuracy curves for original and linearized networks.

Modified WideResNet Archtecture

GROUP NAME	OUTPUT SIZE	BLOCK TYPE
CONV1	32 × 32	[3×3, CHANNEL SIZE]
CONV2	32 × 32	$\begin{bmatrix} 3\times 3, & \text{CHANNEL SIZE} \\ 3\times 3, & \text{CHANNEL SIZE}\end{bmatrix} \times \text{N}$
CONV3	16 × 16	$\begin{bmatrix} 3\times 3, & \text{CHANNEL SIZE} \\ 3\times 3, & \text{CHANNEL SIZE}\end{bmatrix} \times \text{N}$
CONV4	8 × 8	$\begin{bmatrix} 3\times 3, & \text{CHANNEL SIZE} \\ 3\times 3, & \text{CHANNEL SIZE}\end{bmatrix} \times \text{N}$
AVG-POOL	1 × 1	[8 × 8]

Code : https://github.com/google/neural-tangents

Reference

1. Neal, Radford M. “Bayesian learning for neural networks”, Ph.D. Thesis, University of Toronto, 1994.
2. Lee, J. and Bahri, Y., et al. "Deep neural networks as gaussian processes", *arXiv:1711.00165*, ICLR 2018.
3. Matthews, Alexander G. de G., et al. "Gaussian process behaviour in wide deep neural networks", *arXiv:1804.11271*, ICLR 2018.
4. Jacot, A. , et al. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks", *arXiv:1806.07572*, NeurIPS 2018.
5. Lee, J, and Xiao L., et al., “Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent”, arXiv:1902.06720.
6. Zagoruyko, S. and Komodakis, N. “Wide residual networks”, BMVC 2016.