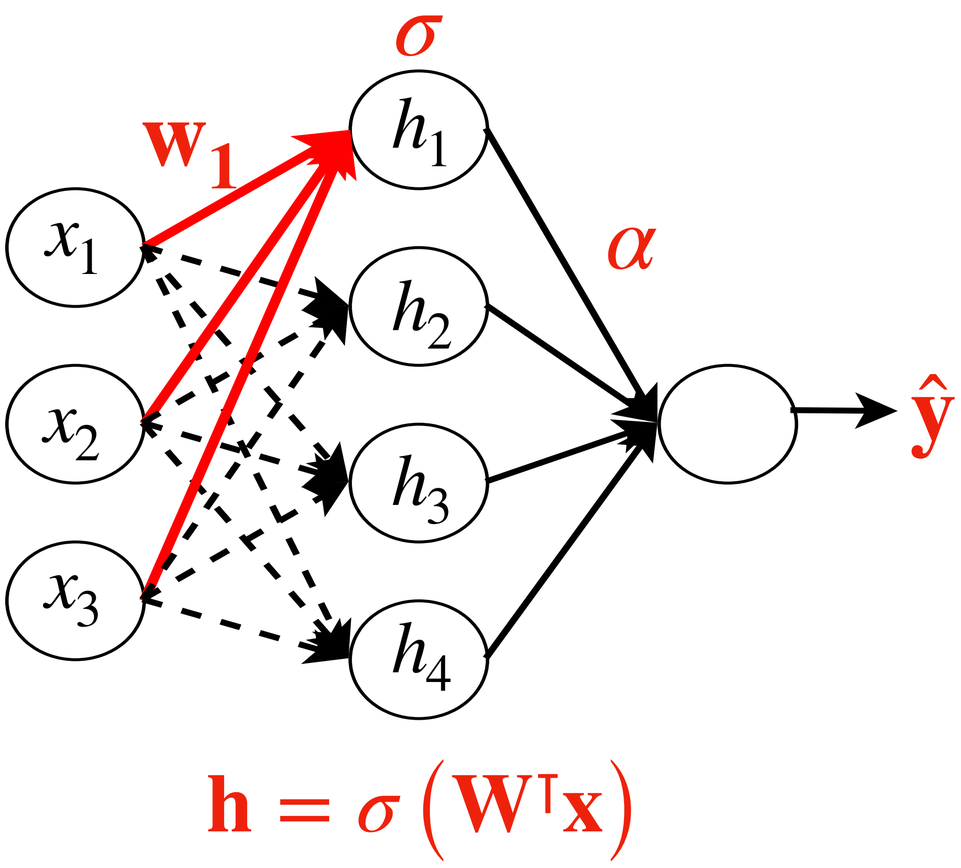
Analysis of bio-inspired initialization in the NTK regime

Biraj Pandey & Aleksei Sholokhov

Neural networks are usually initialized iid normal.



$$\mathbf{h} = \sigma\left(\mathbf{W}^\intercal \mathbf{x}\right)$$

- $\mathbf{W}, \alpha$ are sampled from $\mathcal{N}(0,\sigma^2)$
- σ prevents gradients from exploding/ vanishing[1].
- Doesn't assume anything on input structure.
- This works well in practice.

[1] *Kaiming He. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*. NeurIPS.

Assuming you knew the input structure, can you find better distributions to init from?

Image classification

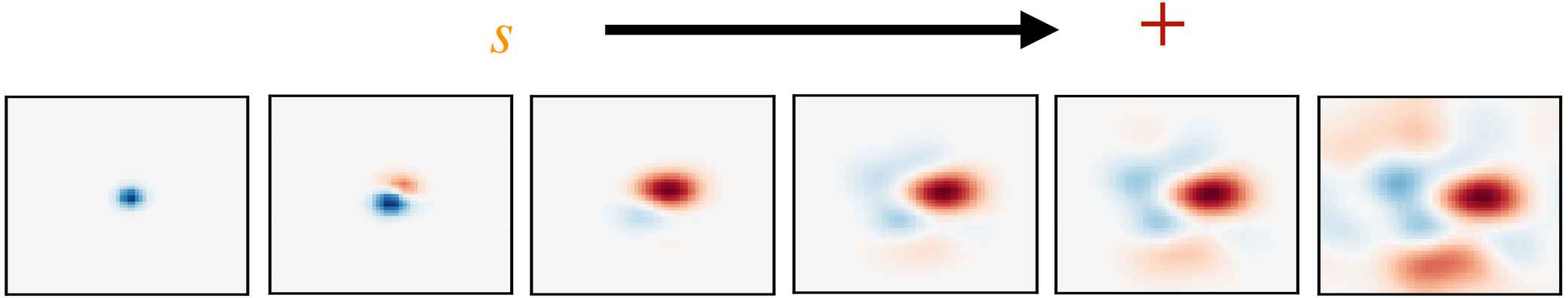
$$\mathbf{W}, \alpha \sim ?$$

Initialize from multivariate normal distributions inspired by biology[2].

- $\mathbf{W}$ are sampled from $\mathcal{N}(0, C(t, t'))$

$$C(t,t') = \exp\left(-\frac{\|t-t'\|^2}{2f^2}\right) \cdot \exp\overbrace{\left(-\frac{\|t-c\|^2 + \|t'-c\|^2}{2s^2}\right)}^{\textit{localized to center } \mathbf{c}}$$

size parameter



[2] K D Harris. (2019). Additive function approximation in the brain. **https://arxiv.org/abs/1909.02603**

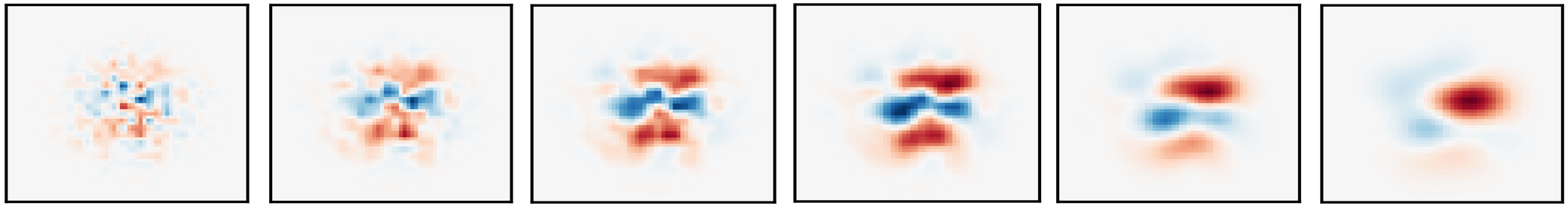
Initialize from multivariate normal distributions inspired by biology.

smooth weights

$$C(t,t') = \overbrace{\exp\left(-\frac{\|t-t'\|^2}{2f^2}\right)} \cdot \exp\left(-\frac{\|t-c\|^2 + \|t'-c\|^2}{2s^2}\right)$$

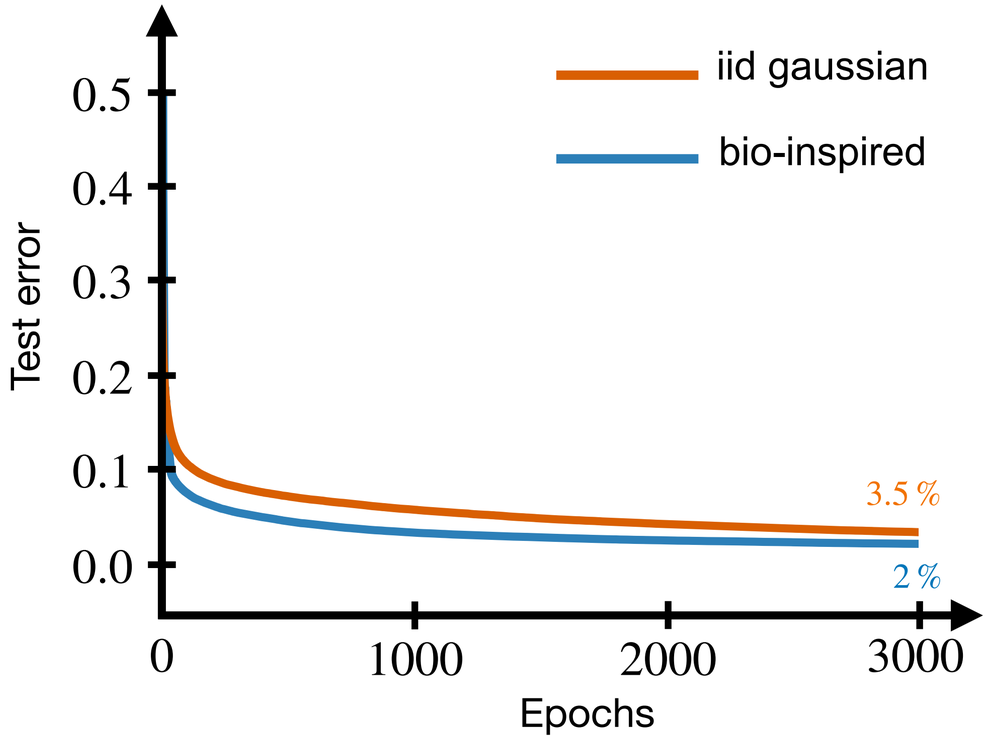
smoothness parameter

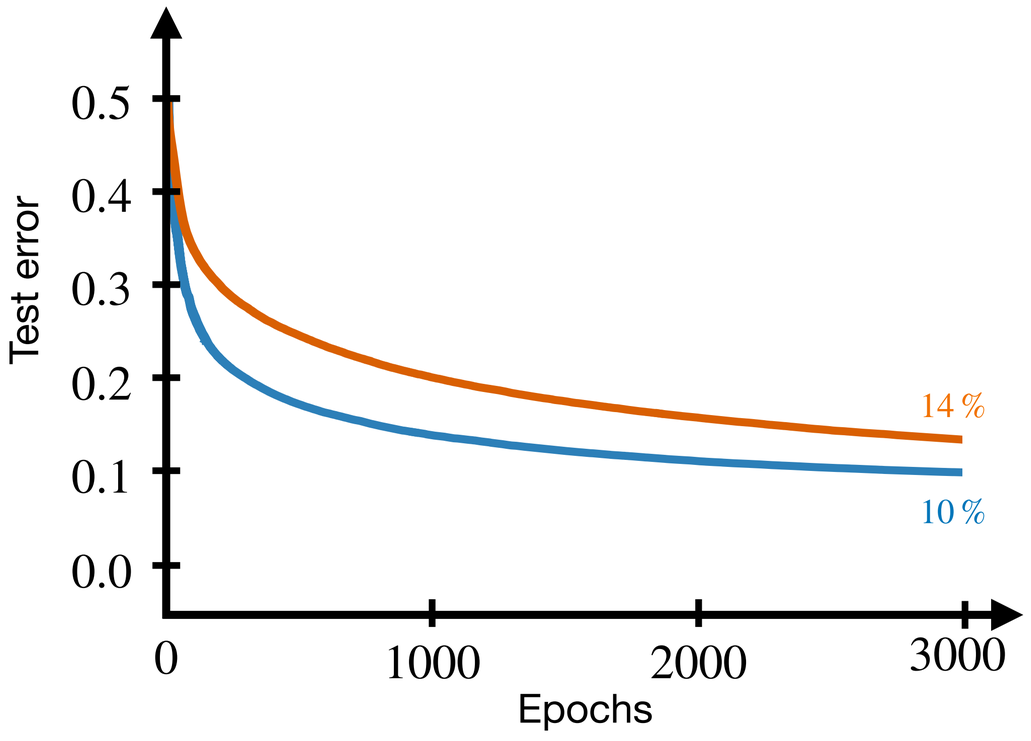
f ⟶ –



Bio-inspired initialization leads to faster loss convergence.

- $h = 1000, \eta = 0.1$, trained with gradient descent





Calculating matrix H* with non-diagonal covariance C

The new matrix H* is different in arccos-term

$$[H^*_{\text{bio}}]_{ij} = \mathbb{E}_{w\sim\mathcal{N}(0,C)}\left[x_i^T x_j \mathbb{I}\{w^T x_i \geq 0,\ w^T x_j \geq 0\}\right]$$

$$H^*_{\text{bio}} = \frac{XX^T(\pi - \arccos XCX^T)}{2\pi}$$

Its eigenvalues and projections of labels on its eigenvectors define the convergence speed of GD

$$H^* = \sum_{i=1}^{n} \lambda_i v_i v_i^\top, \lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0, v_i \in \mathbb{R}^n$$

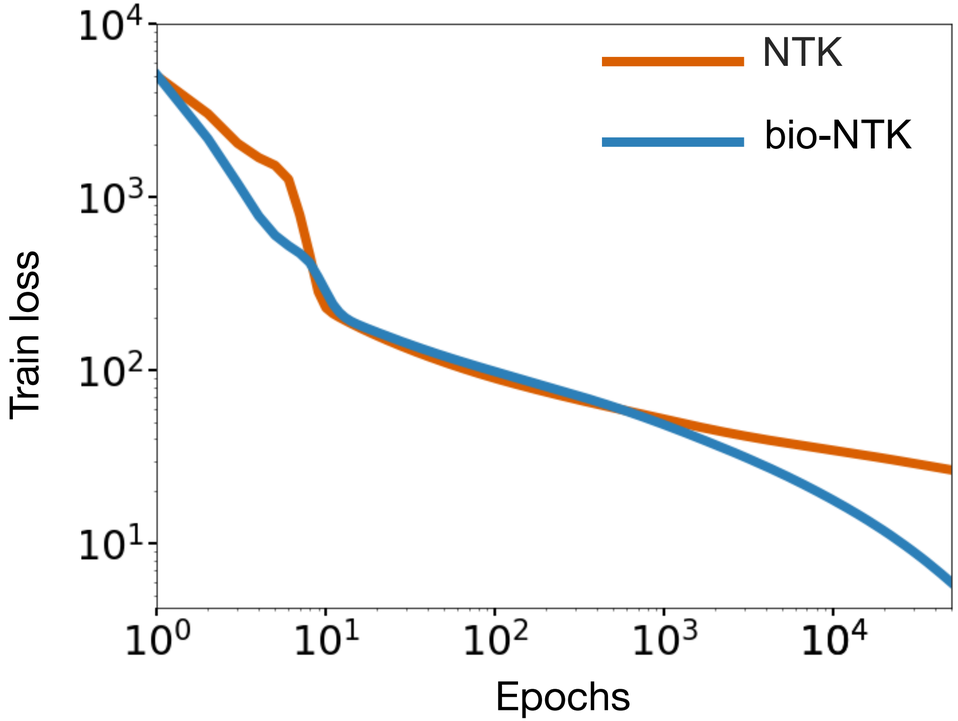
$$\|u(t) - y\|_2^2 \approx \sum_{i=1}^{n} \exp(-\lambda_i t)(v_i^\top (u(0) - y))^2$$

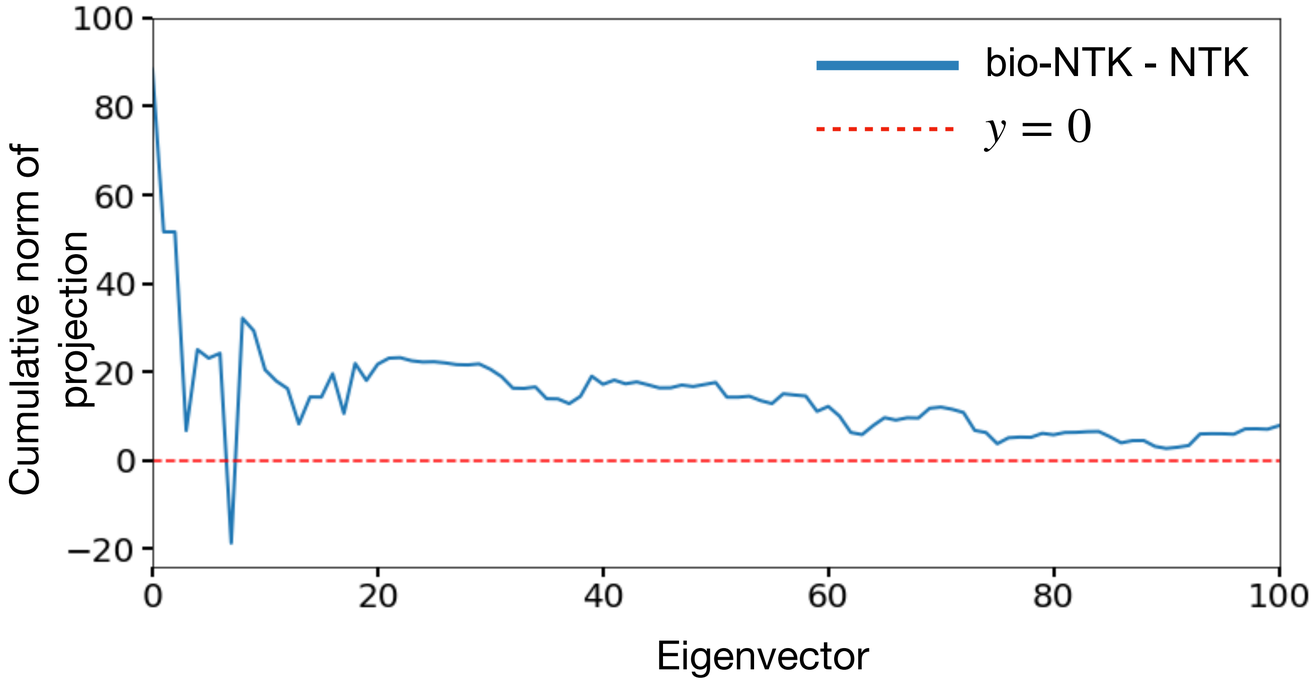
Convergence speed of GD depends on projections of y

$$H^* = \sum_{i=1}^{n} \lambda_i v_i v_i^\top, \lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0, v_i \in \mathbb{R}^n$$

$$\|u(t) - y\|_2^2 \approx \sum_{i=1}^{n} \exp(-\lambda_i t)(v_i^\top (u(0) - y))^2$$

[1]





1. Arora, S., Du, S. S., Hu, W., Li, Z., & Wang, R. (2019). Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. http://arxiv.org/abs/1901.08584

Find optimal GP parameters using NTK framework

$$C(t, t') = \exp\left(-\frac{\|t - t'\|^2}{2l^2} \right)$$

We want to accelerate the initial convergence rate:

$$\left(\|u(t) - y\|_2^2\right)'_t (0) \leq \sum_{i=1}^{n} -\lambda_i e^{-\lambda_i 0} y^T v_i v_i^T y = -y^T H(l) y$$

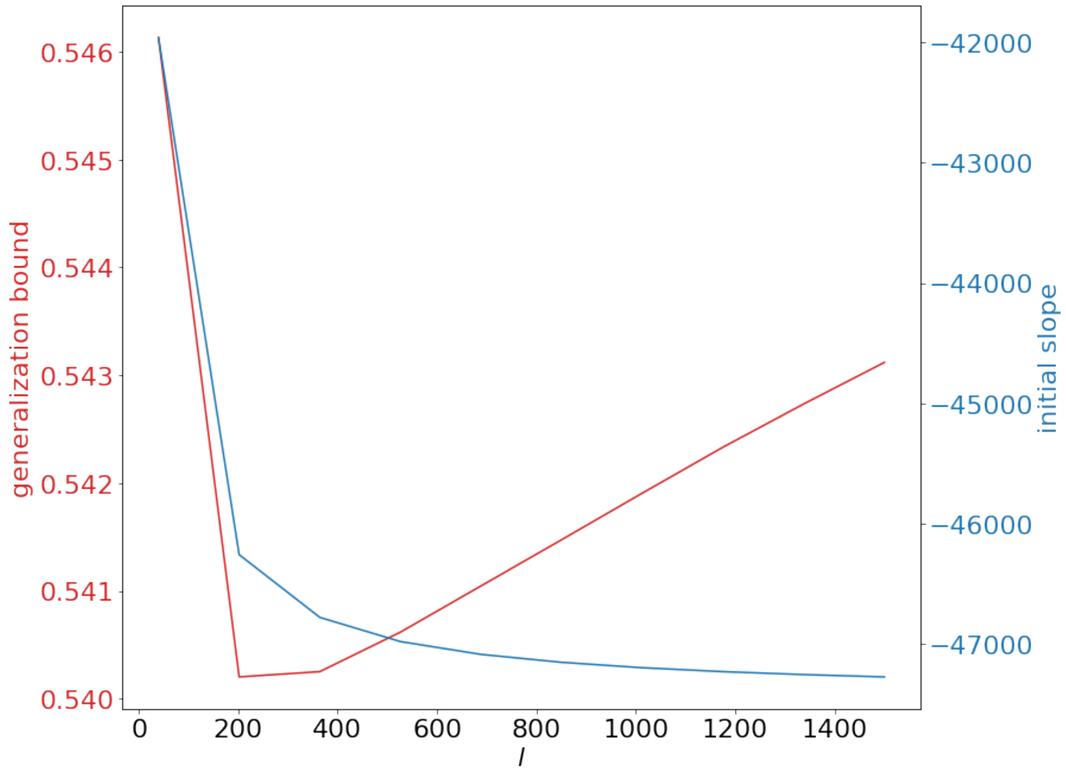
But we don't want to jeopardize generalization:

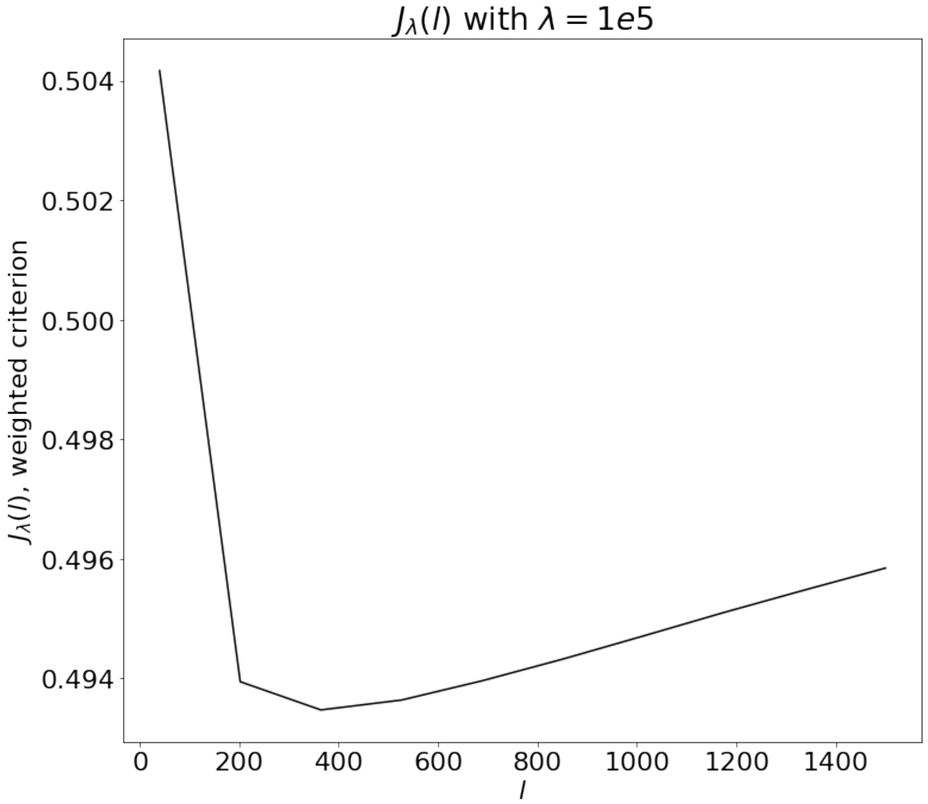
$$\frac{y^T H^{-1}(l) y}{n}$$ [1]

1. Arora, S., Du, S. S., Hu, W., Li, Z., & Wang, R. (2019). Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. http://arxiv.org/abs/1901.08584

J(l) balances generalization and convergence pace

$$J_\lambda(l) = (1-\lambda)y^T H^{-1}(l)y - \lambda y^T H(l)y$$





Can obtain optimal parameter via fast Newton iteration: $l^+ = l - J_\lambda(l)/J'_\lambda(l)$

Conclusion

- NTK framework is useful to analyze the effect of initialization on convergence and generalization of neural networks.
- The eigenvalues of the NTK matrix and the projection of top eigenvectors on data labels determine the convergence speed of the loss.
- We can find optimal init. hyperparameters that maximize convergence speed and minimize generalization error.

Analysis of representation properties

- Random filters induce a **change of basis** on inputs.

filter — Covariance matrix

$$\mathbf{w} \sim \mathrm{GP}(\mathbf{0}, \mathbf{C})$$

- Randomly sample filters

Eigenbasis

$$\mathbf{C} = \mathbf{V}\mathbf{D}^2\mathbf{V}$$

Eigenvalues

- Eigendecomposition of cov. matrix

$$\mathbf{w} = \mathbf{V}\mathbf{D}\mathbf{g}$$

White noise $\mathcal{N}(0,1)$

- Def. of sampling from GP

Analysis of representation properties

- Random filters induce a **change of basis** on inputs.

$$\mathbf{w} = \mathbf{V}\mathbf{D}\mathbf{g}$$

- Analytic form of $\mathbf{w}$

$$\mathbf{h} = \mathbf{w}^\intercal\mathbf{x} = \mathbf{g}^\intercal\mathbf{D}\mathbf{V}^\intercal\mathbf{x} = \mathbf{g}^\intercal\tilde{\mathbf{x}}$$

- Deterministic change of basis!

sensor representation

input *projected* into eigenbasis and *filtered* by eigenvalues

Organization

- Initialization with an inductive bias but also that lends itself well to already present gaussian initialization
- Use a gaussian process for initialization.
- But which GP? We will use a GP that resembles filters of biological neurons.
- Experimentally, we see gains in performance. Learning curves converge faster. The question is why?
- Let’s analyze it from the NTK regime.