UNIVERSITY OF CALIFORNIA SANTA CRUZ

NEURAL INFORMATION PROCESSING SYSTEMS

Understanding the Role of Optimization in Double Descent

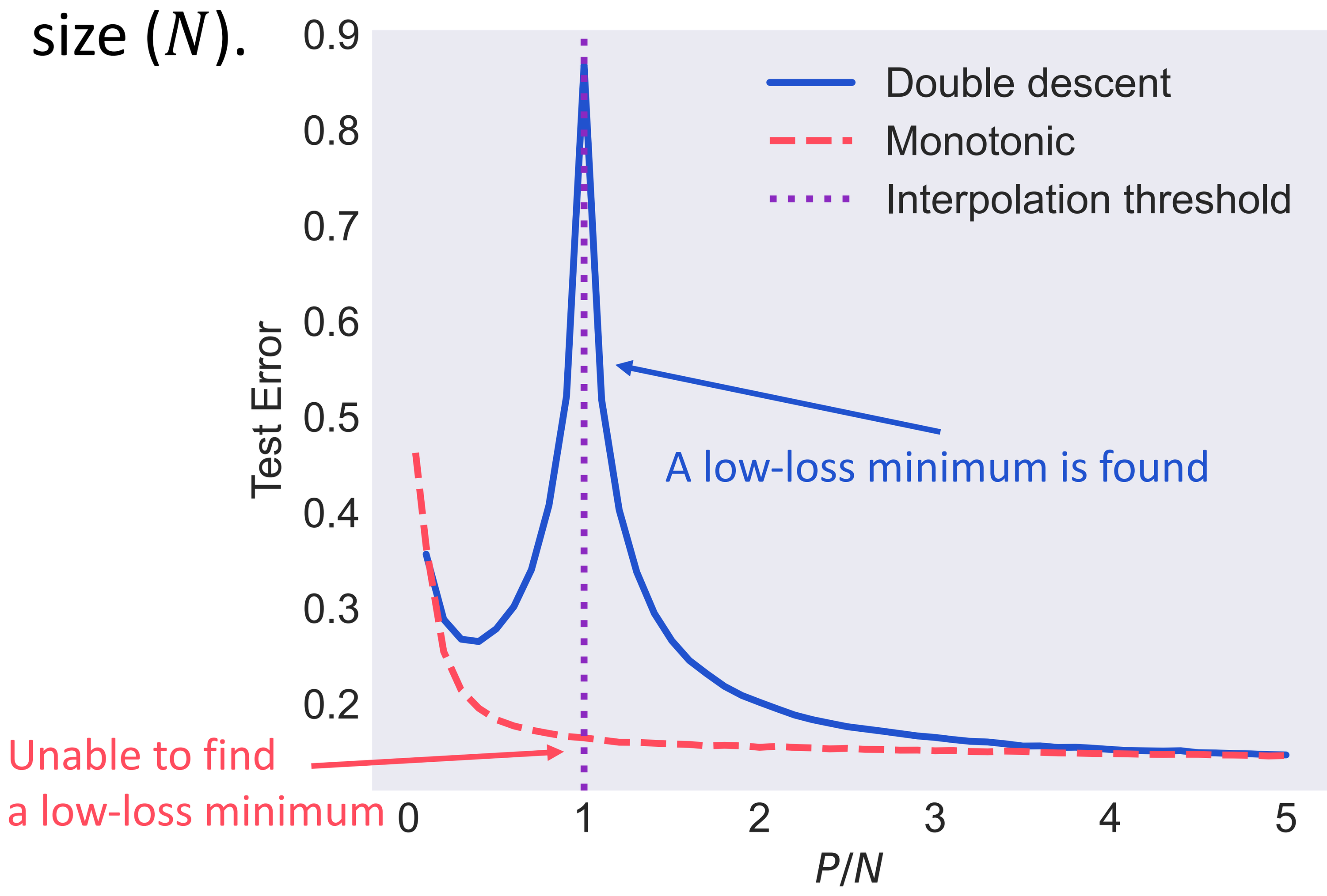
Chris Yuhao Liu & Jeffrey Flanigan

University of California, Santa Cruz

Model-Wise Double Descent

Double descent generalization curve: The generalization error peaks when the number of parameters (P) equals the number of data points (N).

Classical generalization curve: The generalization error decreases monotonically as the number of parameters (P) increases w.r.t. the training dataset size (N).



We show that double descent is observed if and only if **the optimization setting is able to find a sufficiently low-loss minimum around P/N = 1**.

Role 1: Poor Conditioning Reduces Double Descent

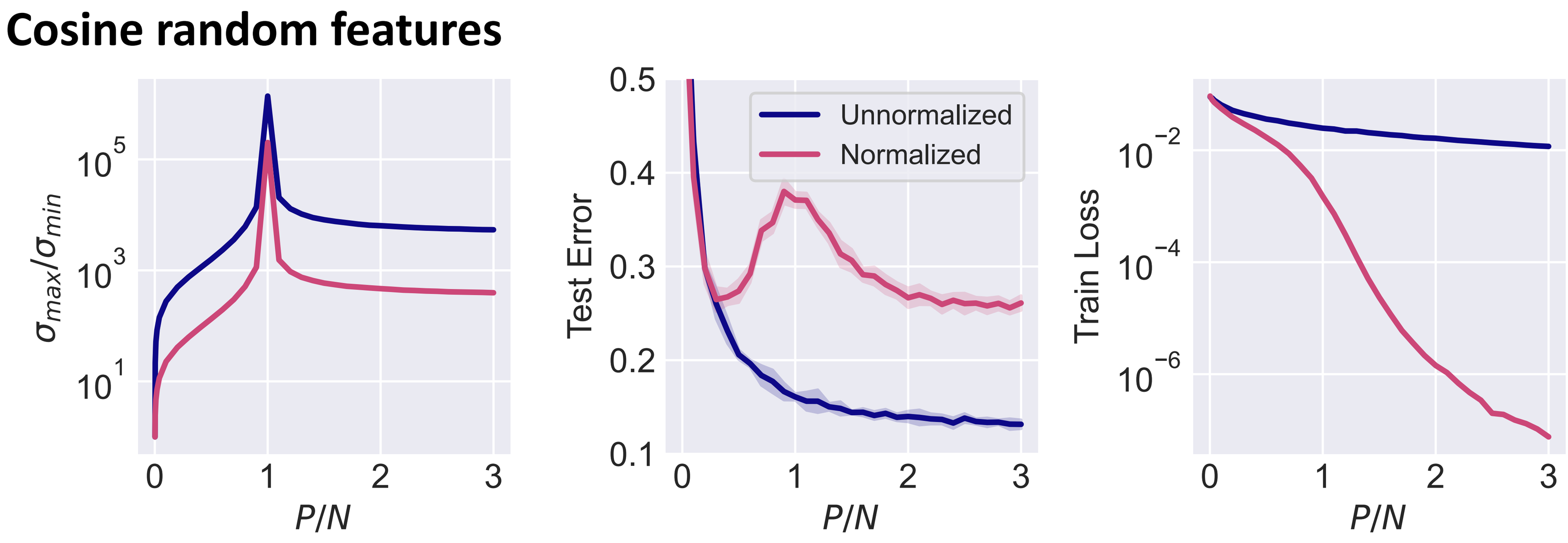
Random features: $Z = \phi(X \cdot \boldsymbol{W}^{\mathrm{T}})$ ($\boldsymbol{W}$ is fixed)

Condition number: $\kappa(Z) = \frac{\sigma_{max}(Z)}{\sigma_{min}(Z)}$ (σ_{max} and σ_{min} are the maximum and the minimum singular values of Z.)

Normalization of the features:

$$X_{normalized} = \frac{X - mean(X)}{std(X)}$$

Cosine random features

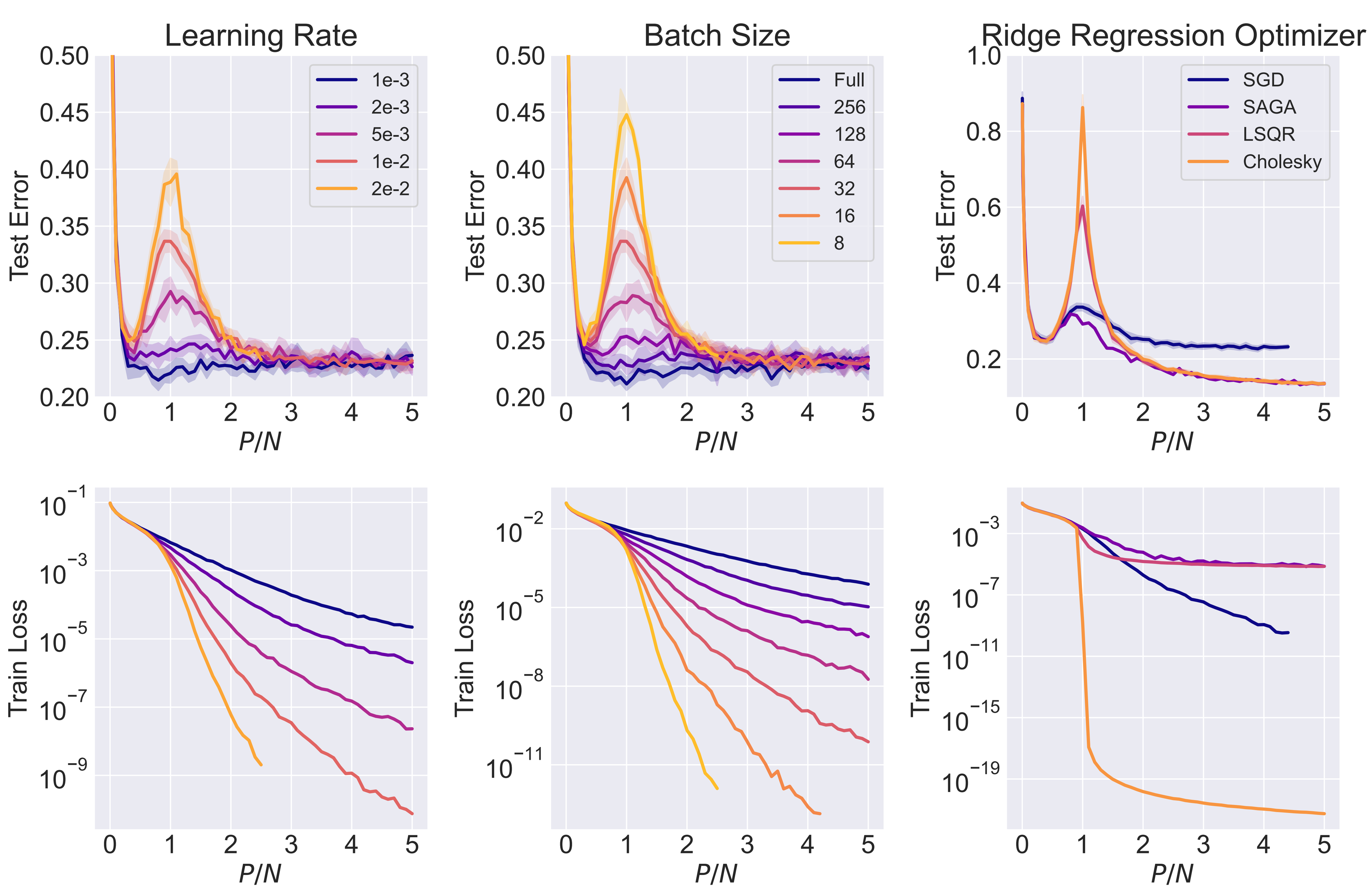


- The same result holds for ReLU and other non-linear functions.
- Similar results hold for other operations that change the condition number of Z, such as scaling the features X or initializing the random matrix $\boldsymbol{W}$ with a different variance. See paper for more plots.

Take-away: Operations leading to **poor condition (high condition number)** reduce and eliminate double descent.

Role 2: Slow-Convergence Optimization Settings Reduce the Peak

Random (ReLU) feature regression models varying learning rate, batch size, and optimization algorithms



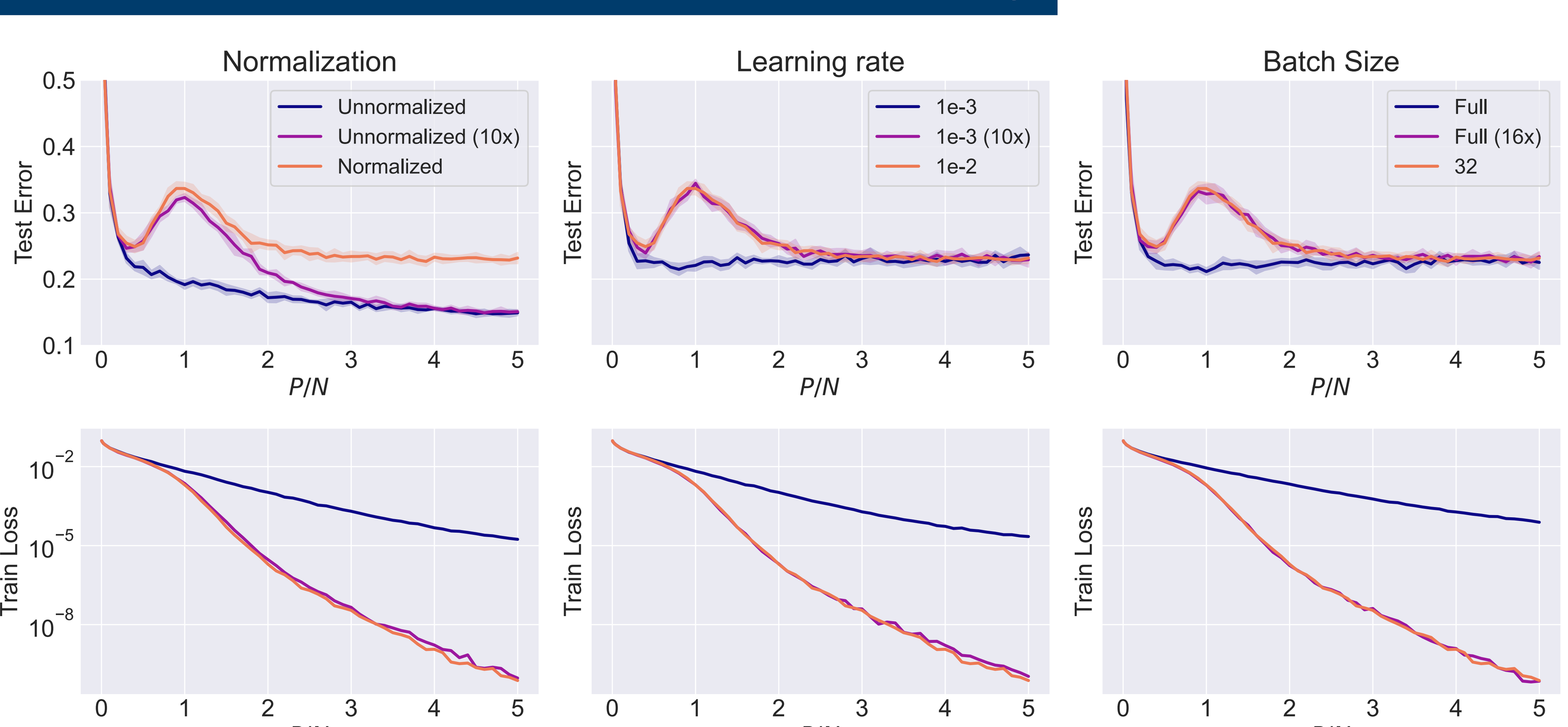
Take-away: For SGD with a fixed number of iterations, the following hyperparameter conditions **slow down the convergence** of the training loss, thereby **gradually making the generalization curve monotonic** while achieving 0 training error:

- Small weight initialization
- Small learning rate
- Aggressive learning rate decay
- Large batch size

Optimization algorithms that struggle to find a low minimum also have a similar effect (see plot to left).

We find consistent behavior on both **random feature models** (linear to input) and **two-layer neural networks**.

Double Descent Still Occurs, But Slowly



Given 1) poorly conditioned problems or 2) slow-convergence settings, double descent ultimately emerges after **a large number of iterations** (200-400 times longer after convergence).

We conclude that double descent:

1) Requires particular proper settings to occur;

2) Emerges slowly after the training error converges to zero;

3) Can be easily mitigated by stopping earlier without harming model convergence in practice.

Time evolution of training a random ReLU feature model