Advanced Topic A01: *The Double Descent Phenomenon*

Data Science with Python – BSc Advanced Lectures

Joerg Osterrieder

10 Minutes

Motivation: Bigger Models Should Overfit

The classical story says: more complexity, more overfitting

- Textbooks teach the bias-variance tradeoff as a U-curve
- Pick the sweet spot, regularize, done
- But modern neural nets have billions of parameters and generalize well
- In 2019, Belkin et al. showed the classical picture is incomplete

The U-curve is right – but it only shows half the picture

The Question

What really happens to test error as model complexity grows?

- Classical answer: test error falls, reaches a minimum, then rises (overfitting)
- Modern observation: keep increasing complexity past the point where training error hits zero, and test error falls *again*
- This second descent is the double descent phenomenon

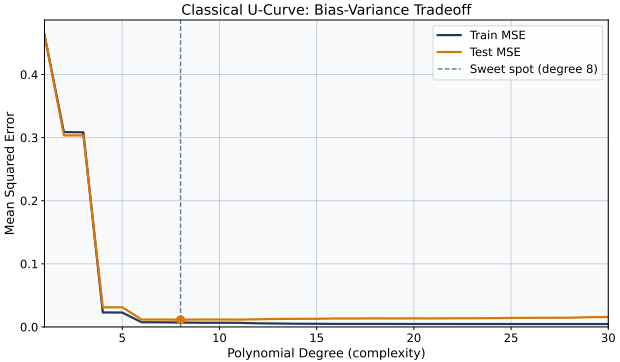
"Test risk as a function of model complexity has two descents, not one"

Belkin, Hsu, Ma, Mandal (2019): Reconciling modern machine learning and the bias-variance trade-off

The Classical U-Curve

Bias-variance tradeoff: the foundation

- Low-complexity models: high bias (underfitting), low variance
- High-complexity models: low bias, high variance (overfitting)
- The sweet spot minimises total test error



Polynomial regression on synthetic data: test MSE peaks near degree 10-15, then stabilises

What Changes with Overparameterization

Cross the interpolation threshold and something unexpected happens

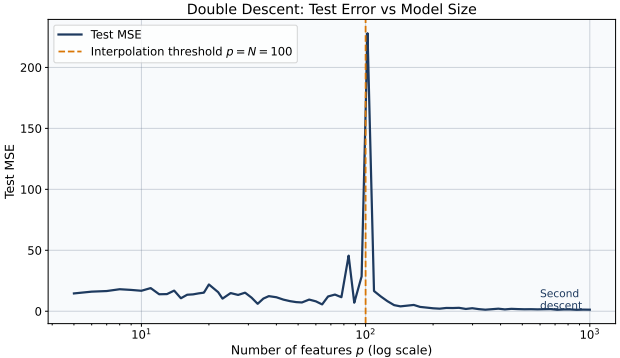
- *Interpolation threshold*: the model has enough parameters to fit every training point exactly ($p \approx N$, parameters equals data)
- Right at $p = N$: test error spikes sharply (the "peak")
- Beyond $p = N$: test error falls again as the model becomes increasingly overparameterized
- The second descent can reach lower error than the classical sweet spot

The spike at $p = N$ is the key signature of double descent

The Double Descent Curve

Two regimes separated by an interpolation threshold

- Left of the threshold ($p < N$): classical U-curve territory
- At the threshold ($p \approx N$): error explodes (model is simultaneously fitting all training points and highly sensitive to noise)
- Right of the threshold ($p > N$): minimum-norm interpolation, error falls as implicit regularization takes over

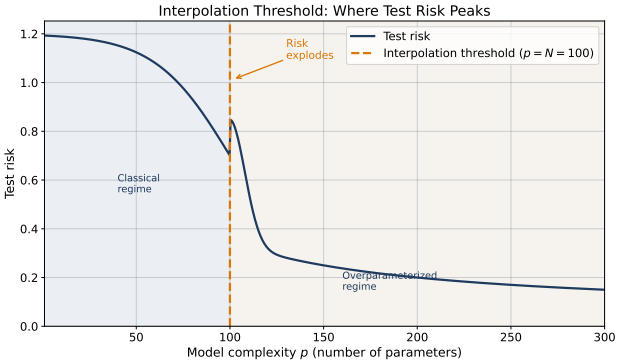


AMBER line marks $p = N = 100$; NAVY curve shows test MSE vs log(p)

The Interpolation Threshold: Why a Spike?

At $p = N$ the system is on a knife-edge

- Exactly one model (the least-squares solution) fits the training data
- Any noise in the labels is absorbed at full magnitude
- The solution is highly sensitive: remove one training point and the interpolating model changes drastically
- Just beyond $p = N$, many interpolating solutions exist; the optimizer picks the minimum-norm one, which is smoother



Interpolation threshold at $p = N$: test risk peaks sharply, then declines as p increases further

Why It Happens: Minimum-Norm Interpolation

The intuition behind the second descent

- When $p > N$, there are infinitely many models that fit the training data exactly
- Gradient descent (SGD) converges to the *minimum-norm* solution among them
- Minimum-norm solutions tend to be smooth and well-behaved on test data
- This is implicit regularization: the optimizer itself acts as a regulariser without any explicit penalty term

References: Belkin & Mei 2019; Bartlett, Montanari, Rakhlin 2020

SGD with zero learning-rate decay converges to the min-norm interpolant in linear models

Empirical Evidence

Double descent is not just theory

- ResNets trained on CIFAR-10: test error shows a spike at the interpolation threshold, then decreases as model width grows (Nakkiran et al. 2019)
- Transformer language models: scaling laws show monotone improvement past interpolation (Kaplan et al. 2020)
- Random feature models, decision trees, boosting: all show the phenomenon under the right conditions
- It also appears *sample-wise*: fix model size, vary training set size

Nakkiran et al. (2019): Deep double descent: where bigger models and more data hurt

Sample-wise Double Descent

Two axes, both show the phenomenon

Model-wise (what we have seen):

- Fix N, vary p
- Spike when $p \approx N$
- Second descent as $p \gg N$

Sample-wise:

- Fix p, vary N
- Spike when $N \approx p$
- Error improves as $N \gg p$
- Adding data can *temporarily* hurt test error

Both axes of the bias-variance story need updating

Regularization Smooths the Spike

Explicit L2 regularization can eliminate the peak

- Ridge regression adds $\lambda\|\mathbf{w}\|^2$ to the loss
- At the interpolation threshold, this shrinks the explosive direction
- Result: the spike disappears; test error transitions smoothly between the two regimes
- Practical implication: a small amount of weight decay is always worth trying, not just for preventing overfitting in the classical sense

Ridge with $\lambda > 0$ converts the spike into a smooth plateau

Cross-Links to Other Advanced Topics

Double descent touches several topics in this series

- **A11 (Lottery Ticket):** a dense overparameterized net contains a sparse subnetwork that trains as well as the full net; both rely on overparameterization being beneficial, not harmful
- **A14 (Information Bottleneck):** a parametric lens on what models compress vs. predict; compression analysis changes when $p > N$
- Both phenomena arise from the same root: overparameterization creates a large solution space that SGD navigates implicitly

Understanding one of A01/A11/A14 deepens understanding of the other two

Why Practitioners Should Care

Model selection heuristics need updating

- The classical advice "stop when validation loss stops improving" assumes a single minimum; with double descent there may be two
- A model slightly past the interpolation threshold may look worse than one before it, but keep training (add width) and error may recover
- Cross-validation near $p = N$ can be unreliable: small changes in fold assignment shift the spike location
- Ensemble methods and early stopping interact with double descent in non-obvious ways

Never trust a single validation curve point; sweep the whole p/N axis

Limitations of the Double Descent Picture

The phenomenon is real but the details depend on many factors

- Optimizer choice: Adam vs. SGD can shift or smooth the spike
- Initialization: random seed affects sharpness of the interpolation peak
- Label noise level: high noise amplifies the spike; near-zero noise eliminates it
- Architecture: convolutional nets, residual connections, and attention all interact differently with the threshold
- Most clean theoretical results are for linear or kernel models; deep networks are harder to analyse formally

The Belkin picture is most crisp for linear overparameterized models

Open Theory Questions

The formal picture is still incomplete

- What are the precise conditions (loss, architecture, optimizer) under which double descent occurs?
- Can we predict the spike location from first principles for deep nets?
- What is the right complexity measure? Parameter count is a proxy; the true "effective dimension" is debated
- Does double descent extend to all loss functions, or is it specific to squared loss and cross-entropy?

Active research area: see NeurIPS 2023 workshops on overparameterization

Code Pointer

Reproduce model-wise double descent in 20 lines

- `sklearn.linear_model.Ridge` with `alpha=1e-6`
- Use `sklearn.preprocessing.PolynomialFeatures` to sweep degree
- Or use random Fourier features to get $p > N$ easily
- Plot train MSE and test MSE vs. number of features on a log scale
- You will see the spike at $p \approx N$ and the second descent

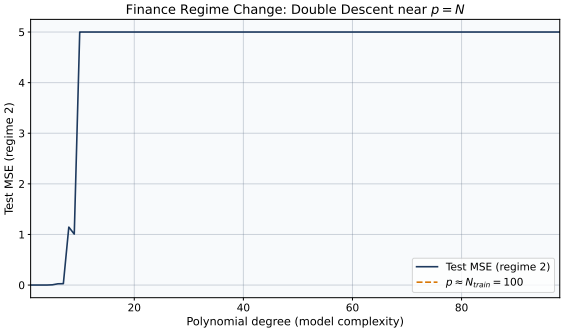
```
from sklearn.linear_model import Ridge
from sklearn.kernel_approximation import RBFSampler
```

Full experiment in under 25 lines; seed=42 for reproducibility

Finance Application: Regime-Change Models

Overparameterization near the threshold is especially dangerous in finance

- Short training windows: a risk model trained on 100 days with 80 features sits right at $p \approx N$, in the danger zone
- Feature set stays large (price ratios, macro indicators) while effective sample size shrinks during stress periods
- Putting a model near $p = N$ produces estimates that are maximally sensitive to individual observations
- Caveat: no published double-descent-in-finance study; this is an illustrative analogy based on effective sample size arguments



AR(1) regime-change example: test MSE spikes near polynomial degree $= N$

What to Watch For in Practice

Diagnostic checklist

- Compute the ratio p/N for your model before tuning
- If $p/N \in [0.5, 2]$, you are near the danger zone
- Use Ridge or dropout to smooth the spike before searching for the optimal complexity
- Sweep the full p/N axis, not just the classical underparameterized regime
- Sample-wise: be careful when adding data near $N \approx p$; test error may temporarily worsen

The ratio p/N is the single most important diagnostic for double descent risk

Practical Takeaway

Overparameterization is not the enemy

- The classical message "too many parameters overfits" is only half right
- The real danger zone is right around $p = N$, not at large p
- Massive overparameterization (large language models, wide ResNets) can generalise well through implicit regularization
- The prescription changes: either stay well below $p = N$ with explicit regularization, or go well above with a large enough model
- Sitting in the middle is the worst place to be

Rule of thumb: $p < 0.1N$ **or** $p > 10N$**; avoid** $p \in [0.5N, 2N]$

Summary and Open Questions

Three things to remember

- The bias-variance U-curve is incomplete: beyond the interpolation threshold ($p = N$), test error descends a second time
- The spike at $p = N$ is caused by minimum-norm interpolation in an under-constrained system; explicit regularization removes it
- Finance applications: keep p/N away from 1; short windows with large feature sets are the most dangerous regime

Open questions:

1. Does double descent extend to all loss functions?
2. Can we predict the spike location from architecture alone?
3. What are the implications for capacity planning in quantitative finance?

Further reading: Belkin et al. 2019; Nakkiran et al. 2019; Bartlett et al. 2020