Double Descent Demystified: Identifying, Interpreting & Ablating the Sources of a Deep Learning Puzzle

Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo

A review by Jack Hanke

October 28, 2024

Setting the scene...

It's 2001, and Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton want to win the ImageNet LSVRC-2010 contest, an image classification competition with over 1000 different classes of images.

Setting the scene...

They use a subset of the ImageNet dataset consisting of 1.2 Million 256×256 images to train a 60 million parameter convolutional neural network they called *AlexNet*.

Setting the scene...

They use a subset of the ImageNet dataset consisting of 1.2 Million 256×256 images to train a 60 million parameter convolutional neural network they called *AlexNet*.

Alexnet achieves state-of-the-art performance and propels the study of deep learning into the mainstream.

Setting the scene...

They use a subset of the ImageNet dataset consisting of 1.2 Million 256×256 images to train a 60 million parameter convolutional neural network they called *AlexNet*.

Alexnet achieves state-of-the-art performance and propels the study of deep learning into the mainstream.

Likely indirectly due to their work, we have all been in a similar situation. You solved a problem with a neural network and now have a large collection of inscrutable weights θ.

Congrats! You just trained a model!

Congrats! You just trained a model!

Question: How does your model work?

Congrats! You just trained a model!

Question: How does your model work?

- What does θ_{343} do in service of the final output? This is the *blackbox problem*.

Congrats! You just trained a model!

Question: How does your model work?

- What does θ_{343} do in service of the final output? This is the *blackbox problem*.
- The answer to this is the world of interpretability research, and is dependent on the specific problem your model is trying to solve.

Congrats! You just trained a model!

Question: How does your model work?

- What does θ_{343} do in service of the final output? This is the *blackbox problem*.
- The answer to this is the world of interpretability research, and is dependent on the specific problem your model is trying to solve.

Question: Why does your model work?

Congrats! You just trained a model!

Question: How does your model work?

- What does θ_{343} do in service of the final output? This is the *blackbox problem*.
- The answer to this is the world of interpretability research, and is dependent on the specific problem your model is trying to solve.

Question: Why does your model work?

- Why does a model with so many parameters not just memorize the data? This is the *double descent problem*.

Congrats! You just trained a model!

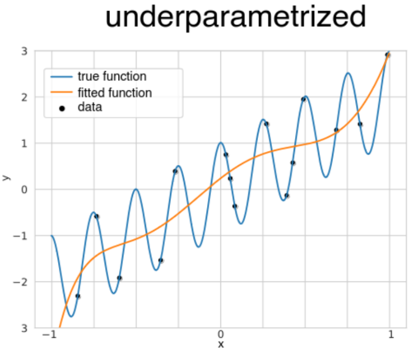
Question: How does your model work?

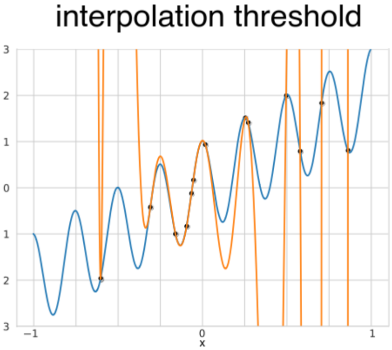
- What does θ_{343} do in service of the final output? This is the *blackbox problem*.
- The answer to this is the world of interpretability research, and is dependent on the specific problem your model is trying to solve.

Question: Why does your model work?

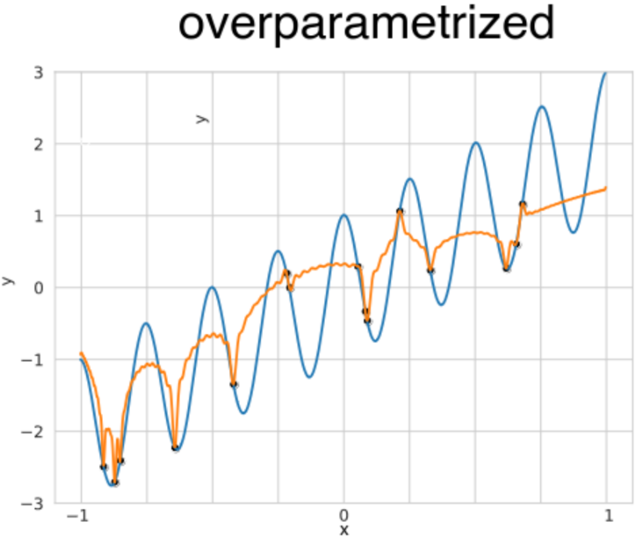
- Why does a model with so many parameters not just memorize the data? This is the *double descent problem*.
- The answer to this is (in part) this paper.

The Traditional View





What (often) actually happens



What is double descent?

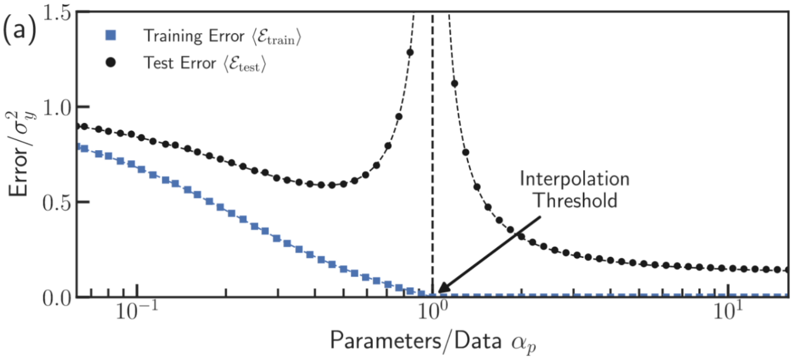
This paper defines double descent as:

A phenomenon in machine learning that many classes of models can, under relatively broad conditions, exhibit where as the number of parameters increases, the test loss falls, rises, then falls again.

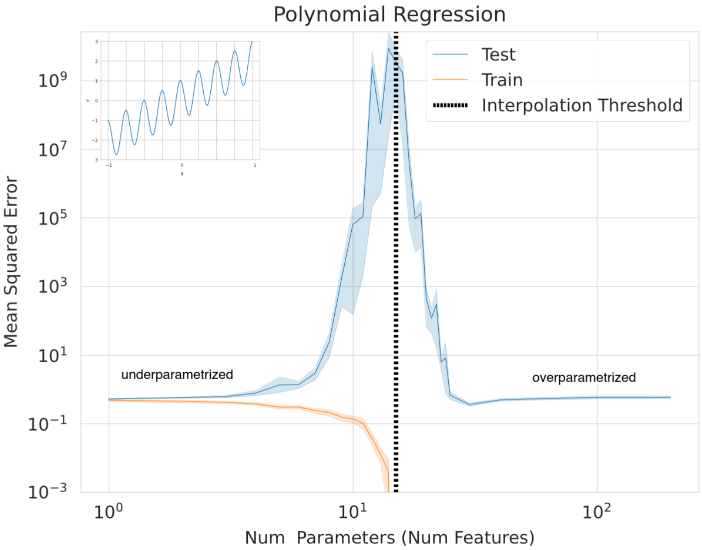
What is double descent?

This paper defines double descent as:

A phenomenon in machine learning that many classes of models can, under relatively broad conditions, exhibit where as the number of parameters increases, the test loss falls, rises, then falls again.



Double descent in polynomial regression



Terminology

- Let P be the number of models parameters
- Let N be the number of training data
- Let D be the dimensionality of the data

Terminology

- Let P be the number of models parameters
- Let N be the number of training data
- Let D be the dimensionality of the data
- A model is *underparameterized* if $\frac{N}{P} > 1$
- A model is *overparameterized* if $\frac{N}{P} < 1$
- A model is at the *interpolation threshold* if $\frac{N}{P} = 1$

Terminology

- Let P be the number of models parameters
- Let N be the number of training data
- Let D be the dimensionality of the data
- A model is *underparameterized* if $\frac{N}{P} > 1$
- A model is *overparameterized* if $\frac{N}{P} < 1$
- A model is at the *interpolation threshold* if $\frac{N}{P} = 1$

We will next study linear models, which have a fixed value of $P = D + 1$. Therefore, double descent occurs in the direction of increasing N.

Double descent in linear regression

The underparametrized regime is the classic least-squares minimization problem:

$$\hat{\vec{\beta}}_{under} = \text{argmin}_{\vec{\beta}} ||X\vec{\beta} - Y||_2^2,$$

which is solved by

$$\vec{\beta}_{under} = (X^T X)^{-1} X^T Y.$$

Double descent in linear regression

The underparametrized regime is the classic least-squares minimization problem:

$$\hat{\vec{\beta}}_{under} = \text{argmin}_{\vec{\beta}} ||X\vec{\beta} - Y||_2^2,$$

which is solved by

$$\vec{\beta}_{under} = (X^T X)^{-1} X^T Y.$$

For the overparameterized regime, the above optimization problem has infinite solutions. Therefore, we need to choose a different optimization problem:

$$\hat{\vec{\beta}}_{over} = \text{argmin}_{\vec{\beta}} ||\vec{\beta}||_2^2 \text{ s.t. } \forall n \in (1, \ldots, N)\ \vec{x}_n \vec{\beta} = y_n$$

which is solved by

$$\vec{\beta}_{over} = X^T (X X^T)^{-1} Y.$$

Why this choice?

$$\hat{\vec{\beta}}_{over} = \text{argmin}_{\vec{\beta}} ||\vec{\beta}||_2^2 \text{ s.t. } \forall n \in (1, \ldots, N)\ \vec{x}_n \vec{\beta} = y_n$$

We choose this optimization problem because *it is the optimization problem that gradient decent implicity minimizes*!

The main equation

Unknown to us and the model is the ideal linear parameters β^* that truly minimize the test mean squared error. We write

$$Y = X\beta^* + E$$

where E is the uncapturable error. [1]

[1] E could either be due to a inherently non-linear true relationship or a noisy but linear relationship.

The main equation

Unknown to us and the model is the ideal linear parameters β^* that truly minimize the test mean squared error. We write

$$Y = X\beta^* + E$$

where E is the uncapturable error. [1]
For the underparameterized regime we have

$$\begin{aligned}
\hat{y}_{test,under} &= \vec{x}_{test} \cdot (X^T X)^{-1} X^T Y \\
\hat{y}_{test,under} &= \vec{x}_{test} (X^T X)^{-1} X^T E + y^*_{test} \\
\hat{y}_{test,under} - y^*_{test} &= \vec{x}_{test} (X^T X)^{-1} X^T E.
\end{aligned}$$

[1] E could either be due to a inherently non-linear true relationship or a noisy but linear relationship.

The main equation

One last step before the main equation...

The main equation

One last step before the main equation...

We replace X with its singular value decomposition

$$X = U\Sigma V^T$$

with singular values $\sigma_1 > \sigma_2 \cdots > \sigma_R > 0$. The singular values are the square roots of the eigen values of X , ie $\sigma_r = \sqrt{\lambda_r}$.

The main equation

We get the following

$$\begin{aligned}\hat{y}_{test,over} - y^*_{test} = & \quad \textstyle\sum_{r=1}^{R} \frac{1}{\sigma_r}(\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \quad + \text{bias term} \\ \hat{y}_{test,under} - y^*_{test} = & \quad \textstyle\sum_{r=1}^{R} \frac{1}{\sigma_r}(\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E)\end{aligned}$$

where the bias term is $\vec{x}_{test}(X^T(XX^T)^{-1}X - I_D)\beta^*$.

The main equation

We get the following

$$\begin{aligned} \hat{y}_{test,over} - y^*_{test} &= \textstyle\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \quad + \text{bias term} \\ \hat{y}_{test,under} - y^*_{test} &= \textstyle\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \end{aligned}$$

where the bias term is $\vec{x}_{test}(X^T(XX^T)^{-1}X - I_D)\beta^*$.

The variance term causes double descent!

Intuition for components of variance term

$$\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \quad (1)$$

Double descent happens when all three of these terms grow large!

- How much the *training features* X vary in each direction

$$\frac{1}{\sigma_r}$$

Intuition for components of variance term

$$\sum_{r=1}^{R} \frac{1}{\sigma_r}(\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \tag{1}$$

Double descent happens when all three of these terms grow large!

- How much the *training features* X vary in each direction

$$\frac{1}{\sigma_r}$$

- How much, and which direction, the *test features* $\vec{x}_{test}$ vary relative to the *training features* X

$$\vec{x}_{test} \cdot \vec{v}_r$$

Intuition for components of variance term

$$\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \tag{1}$$

Double descent happens when all three of these terms grow large!

- How much the *training features* X vary in each direction

$$\frac{1}{\sigma_r}$$

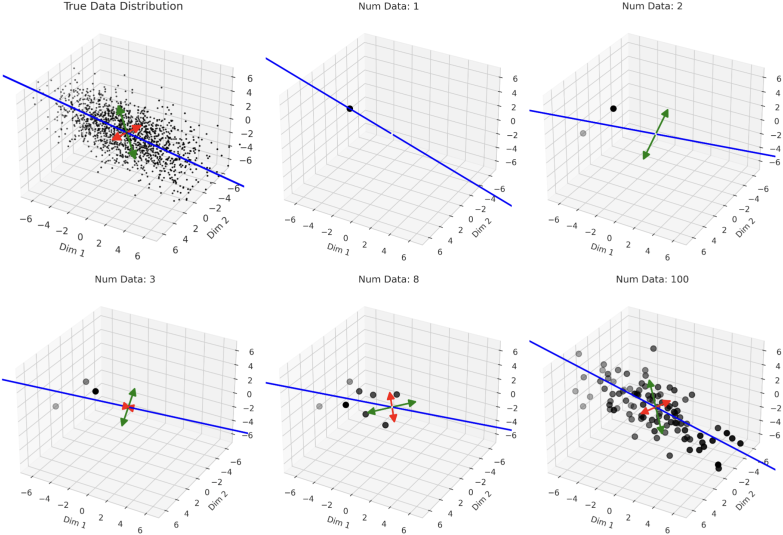
- How much, and which direction, the *test features* $\vec{x}_{test}$ vary relative to the *training features* X

$$\vec{x}_{test} \cdot \vec{v}_r$$

- How well the *best possible model* can correlate the variance in the *training features* X with the *training regression targets* Y

$$\vec{u}_r \cdot E$$

Why do small singular values σ_r happen near $P = D$?



The other components of the variance term

$$\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \tag{2}$$

How do the other terms contribute to double descent?

- The test datum does not vary in different directions than the training features. If the test datum lies entirely in the subspace of just a few of the leading singular directions, then double descent is unlikely to occur.
- If $E = \vec{0}$, (ie the true function is linear), the variance at and after the interpolation threshold is 0.

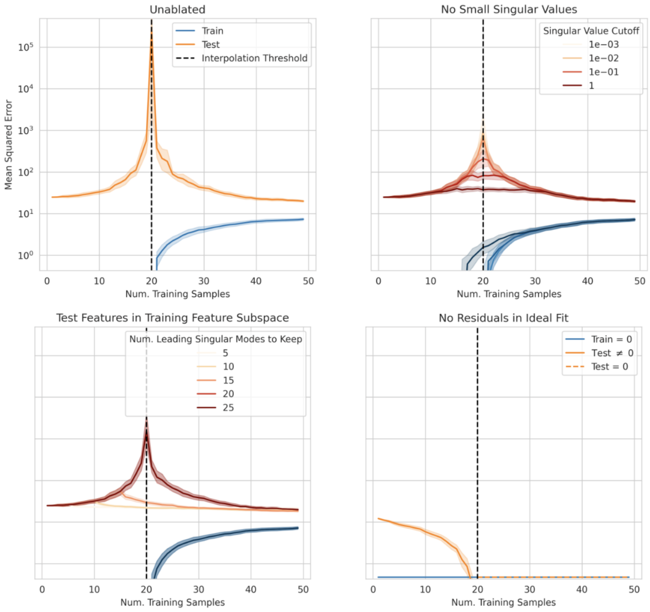
The other components of the variance term

$$\sum_{r=1}^{R} \frac{1}{\sigma_r} (\vec{x}_{test} \cdot \vec{v}_r)(\vec{u}_r \cdot E) \tag{2}$$

How do the other terms contribute to double descent?

- The test datum does not vary in different directions than the training features. If the test datum lies entirely in the subspace of just a few of the leading singular directions, then double descent is unlikely to occur.
- If $E = \vec{0}$, (ie the true function is linear), the variance at and after the interpolation threshold is 0.

We are going to "remove" each of these contributions to the variance and exhibit that double descent does not occur.


Unablated
Train
Test
Interpolation Threshold
Mean Squared Error
Num. Training Samples
No Small Singular Values
Singular Value Cutoff
1e−03
1e−02
1e−01
1
Test Features in Training Feature Subspace
Num. Leading Singular Modes to Keep
5
10
15
20
25
No Residuals in Ideal Fit
Train = 0
Test ≠ 0
Test = 0

Double descent in nonlinear models - Intuition

Neural networks are composed of multiple successive linear regression problems with non-linear activation functions. In many cases training neural networks is equivalent to linear regression on a certain set of features that are functions of the training data.

Double descent in nonlinear models - Intuition

Neural networks are composed of multiple successive linear regression problems with non-linear activation functions. In many cases training neural networks is equivalent to linear regression on a certain set of features that are functions of the training data.

"It's interesting to note that we're observing double-descent in the absence of label noise. That is to say: the inputs and targets are exactly the same. Here, the "noise" arises from the lossy compression happening in the down projection." Our tutorial clarifies that noise in the sense of a random unpredictable quantity is not necessary to produce double descent. Rather, what is necessary is residual errors from the perspective of the model class. Those residual errors could be entirely deterministic, such as a nonlinear model attempting to fit a noiseless linear relationship."

Tom Henighan, Shan Carter, Tristan Hume, Nelson Elhage, Robert Lasenby, Stanislav Fort, Nicholas Schiefer, and Christopher Olah. Double descent in the condition number. *Transformer Circuits Thread, 2023*.

Summary

In this talk we covered

- We *identified* double descent in various regression problems
- We *interpreted* the components of the variance term in the test error that contribute to double descent
- We *ablated* the components for a dataset to demonstrate that double descent occurs only when all components are large near the interpolation threshhold

We also argued that in linear and non-linear models we expect the double descent behavior in a probabalistic sense without intentional ablation.

Thank you for listening!