

ETICS 2026

When neural networks grow infinite:
insights into training dynamics via
Gaussian process and Neural
Tangent Kernel

DTIPD - Data Science and Applied
Mathematics Competency Center (DSAM CC)

Laurent Lefebvre, Mathieu Segond, Gaspard
Mestrot, Joachim Carvalho, Eline Pot

30/09/26



Confidentiality



This document contains *Framatome's* know-how
When the box is checked, add the information in the 'Footer'

Export Control COEDM

EU EC	US EC	OTH EC
country code = FR	country code = N	country code = N
marking code = N	marking code = N	marking code = N

Goods marked EU EC different to "N" are subject to export authorization of the States members of the European Union (EU) in case of export outside EU, and also for certain sensitive goods (e. g. goods marked "EU EC": "0E001") in case of transfer within UE ; they cannot be reexported nor transferred back without a prior agreement from the State of the first export. Goods marked US EC different to "N" are subject to export, re-export or transfer authorization of the United States of America. Goods marked OTH EC (country code(s)) different to "N" are subject to export, re-export or transfer authorization of the mentioned county(ies). Even for goods marked "EU EC: N" and "US EC: N" and "OTH EC: N", an export authorization may be required depending on their final destination or end-use.

Export Control Other

AL = N	ECCN = N
--------	----------

Goods labeled with "AL not equal to N" are subject to European or German export authorization when being exported within or out of the EU.

Goods labeled with "ECCN not equal to N or EAR99" are subject to U.S. reexport authorization. Even without a label, or with label: "AL: N" or "ECCN: N" or "ECCN: EAR99," authorization may be required due to the final whereabouts and purpose for which the goods are to be used.

This document and any and all information contained therein and/or disclosed in discussions supported by this document, are confidential, protected by applicable intellectual property regulations and contain data subject to trade secrets regulations. Any reproduction, alteration, disclosure to any third party and/or publication in whole or in part of this document and/or its content is strictly prohibited without prior written express approval of Framatome. This document and any information it contains shall not be used for any other purpose than the one for which they were provided. Legal and disciplinary actions may be taken against any infringer and/or any person breaching the aforementioned obligations.

X



Check the boxes corresponding to the document's confidentiality level and indicate the applicable export control markings.

Framatome's information protection rules



C1 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are **restricted**.



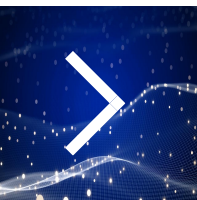
C2 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are sensitive and **Framatome confidential**, such as its disclosure, alteration or loss are detrimental with a significant-to-high impact for Framatome.

The document, if disclosed, and any information it contains are intended for the sole attendees. The disclosure or reference to such information or document shall be made only on a proper judgment basis and by mentioning expressly "this information shall not be disclosed / transferred without prior consent".



C3 : This document and any and all information contained therein and/or disclosed in discussions supported by this document are classified **Framatome Secret**. Each one must commit to keep secret any written or oral information disclosed during the meeting. It is forbidden to disclose it to any legal entity and any individual (including within Framatome) without prior consent of the meeting chairman.

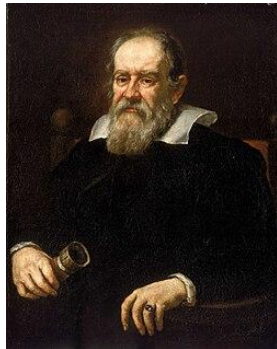
Dissemination scope: specify a perimeter as narrow as possible.



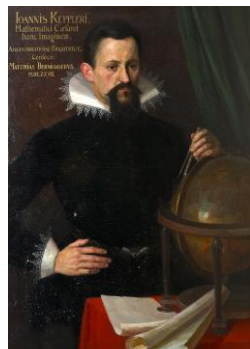
From empirical astronomy to universal laws



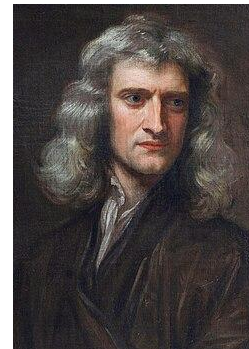
Tycho Brahe
1546-1601



Galileo Galilei
1564-1642

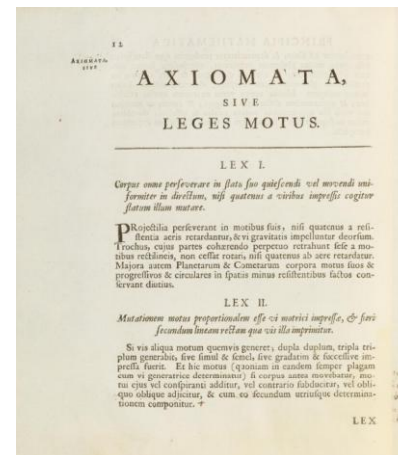
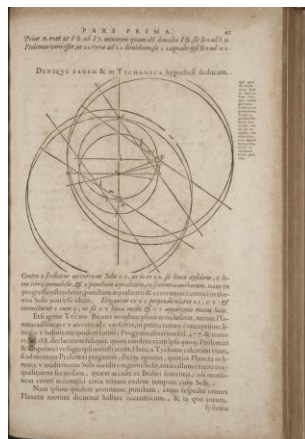
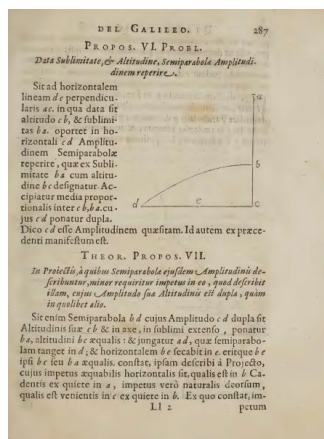
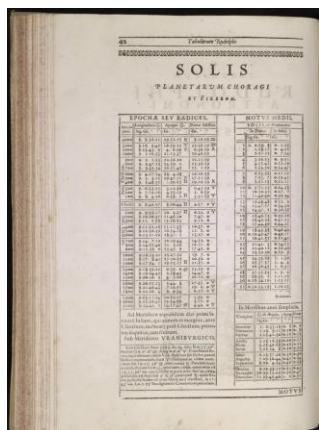


Johannes Kepler
1571-1630



Isaac Newton
1642-1727

...





Empirical evidences in deep learning...

Squeeze-and-Excitation Networks

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
krl1@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network on high-resolution images in the ImageNet ILSVRC 2010 ferent classes. On the test data, we achieved top-1 and and 17.0% which is considerably better than the previous neural network, which has 60 million parameters and of five convolutional layers, some of which are followed by three fully-connected layers with a final 1000-way linear layer. We used non-saturating neurons and a variation of the convolution operation. To reduce overfitting, we employed a recently-developed regularization that proved to be very effective. We also entered a new ILSVRC 2012 competition and achieved a winning top-5 compared to 26.2% achieved by the second-best entry.

Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* & Andrew Zisserman*
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

Jie Hu^{0000-0002-5150-1003} Li Shen^{0000-0002-2283-4976} Samuel Albanie^{0000-0001-9786-5134}
Gang Sun^{0000-0001-6913-6799} Enhua Wu^{0000-0002-2174-1428}

Abstract—The central building block of convolutional neural networks (CNNs) is the convolution operator, which enables networks to construct informative features by fusing both spatial and channel-wise information within local receptive fields at each layer. A broad range of prior research has investigated the spatial component of this relationship, seeking to strengthen the representational power of a CNN by enhancing the quality of spatial encodings throughout its feature hierarchy. In this work, we focus instead on the channel relationship and propose a novel architectural unit, which we term the “Squeeze-and-Excitation” (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. We show that these blocks can be stacked together to form SENet architectures that generalise extremely effectively across different datasets. We further demonstrate that SE blocks bring significant improvements in performance for existing state-of-the-art CNNs at slight additional computational cost. Squeeze-and-Excitation Networks formed the foundation of our ILSVRC 2017 classification submission which won first place and reduced the top-5 error to 2.351%, surpassing the winning entry of 2016 by a relative improvement of ~25%. Models and code are available at <https://github.com/hujie-frank/SENet>.

Index Terms—Squeeze-and-Excitation, Image representations, Attention, Convolutional Neural Networks.

ABSTRACT

Investigate the effect of the convolutional network depth on its scale image recognition setting. Our main contribution is of networks of increasing depth using an architecture with evolution filters, which shows that a significant improvement durations can be achieved by pushing the depth to 16–19 findings were the basis of our ImageNet Challenge 2014 r team secured the first and the second places in the localisation tracks respectively. We also show that our representations er datasets, where they achieve state-of-the-art results. We st-performing ConvNet models publicly available to facilitate the use of deep visual representations in computer vision.

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

I. INTRODUCTION

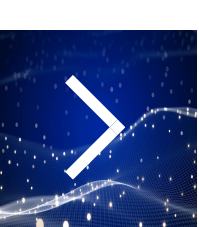
Abstract—Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modelling. A new learning paradigm, called Graph Transformer Networks (GTN), allows each multi-module system to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks.

A Graph Transformer Network for reading bank checks is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provide record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

Keywords—Neural Networks, OCR, Document Recognition, Machine Learning, Gradient-Based Learning, Convolutional Neural Networks, Graph Transformer Networks, Finite State Transducers.



Empirical evidences in deep learning...

Squeeze-and-Excitation Networks

Jie Hu^[0000-0002-5150-1003] Li Shen^[0000-0002-2283-6976] Samuel Albanie^[0000-0001-9786-5134]
Gang Sun^[0000-0001-6913-6799] Enhua Wu^[0000-0002-2174-1428]

Abstract—The central building block of convolutional neural networks (CNNs) is the convolution operator, which enables networks to construct informative features by fusing both spatial and channel-wise information within local receptive fields at each layer. A broad range of prior research has investigated the spatial component of this relationship, seeking to strengthen the representational power of a CNN by enhancing the quality of spatial encodings throughout its feature hierarchy. In this work, we focus instead on the channel relationship and propose a novel architectural unit, which we term the “Squeeze-and-Excitation” (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. We show that these blocks can be stacked together to form SENet architectures that generalise extremely effectively across different datasets. We further demonstrate that SE blocks bring significant improvements in performance for existing state-of-the-art CNNs at slight additional computational cost. Squeeze-and-Excitation Networks formed the foundation of our ILSVRC 2017 classification submission which won first place and reduced the top-5 error to 2.351%, surpassing the winning entry of 2016 by a relative improvement of ~25%. Models and code are available at <https://github.com/hujie-frank/SENet>.

Index Terms—Squeeze-and-Excitation, Image representations, Attention, Convolutional Neural Networks.

Densely Connected Convolutional Networks

Gao Huang^{*} Zhuang Liu^{*} Laurens van der Maaten^{*}
Cornell University Tsinghua University Facebook AI Research
gh149@cornell.edu liuzhuang13@mails.tsinghua.edu.cn lvdmaaten@fb.com
Kilian Q. Weinberger^{*}
Cornell University
kqw6@cornell.edu

Abstract

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections—one between each layer and its subsequent layer—our network has $\frac{L(L+1)}{2}$ direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. We evaluate our proposed architecture on four highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet). DenseNets obtain significant improvements over the state-of-the-art on most of them, whilst requiring less computation to achieve high performance. Code and pre-trained models are available at <https://github.com/11xushuang13/DenseNet>.

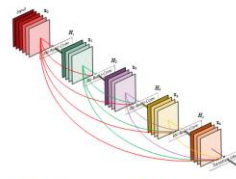


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

Networks [14] and Residual Networks (ResNets) [11] have surpassed the 100-layer barrier.

As CNNs become increasingly deep, a new research problem emerges: as information about the input or gradient passes through many layers, it can vanish and “wash out” by the time it reaches the end (or beginning) of the network. Many recent publications address this or related

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky^{*} Ilya Sutskever^{*} Geoffrey E. Hinton^{*}
University of Toronto University of Toronto University of Toronto
kriz@cs.utoronto.ca ilsya@cs.utoronto.ca hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network on high-resolution images in the ImageNet ILSVRC 2010 ferent classes. On the test data, we achieved top-1 and 17.0% which is considerably better than the previous neural network, which has 60 million parameters and of five convolutional layers, some of which are followed by three fully-connected layers with a final 1000-way linear layer. We used non-saturating neurons and a very-tation of the convolution operation. To reduce overfitting layers we employed a recently-developed regularization that proved to be very effective. We also entered a e ILSVRC 2012 competition and achieved a winning top compared to 26.2% achieved by the second-best entry.

Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan^{*} & Andrew Zisserman^{*}
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

ABSTRACT

Investigate the effect of the convolutional network depth on its scale image recognition settings. Our main contribution is of networks of increasing depth using an architecture with rotation filters, which shows that a significant improvement gurations can be achieved by pushing the depth to 16–19 findings were the basis of our ImageNet Challenge 2014 r team secured the first and the second places in the localiza-tracks respectively. We also show that our representations er datasets, where they achieve state-of-the-art results. We st-performing ConvNet models publicly available to facili-n the use of deep visual representations in computer vision.

Deep Residual Learning for Image Recognition

Kaiming He^{*} Xiangyu Zhang^{*} Shaoqing Ren^{*} Jian Sun^{*}
Microsoft Research
{khe, y-xiangz, v-shren, jiansun}@microsoft.com

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Mingqing Tan¹ Qian Xie¹

Abstract

Convolutional Neural Networks (CNNs) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. We demonstrate the effectiveness of this method on scaling up MobileNet and ResNet.

To go even further, we use neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous CNNs. In particular, our EfficientNet-B7 achieves state-of-the-art 84.3% top-1 accuracy on ImageNet, while being 8.4% smaller and 4.4% faster on inference than the best existing ConvNet. Our EfficientNet-B0 also matches well and achieves state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 1 other transfer learning datasets, with an order of magnitude fewer parameters. Source code is at <https://github.com/tan-mingqing/efficientnet>.

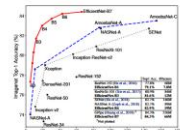


Figure 1: Model Size vs. ImageNet Accuracy. All numbers are for single-core, single-model. Our EfficientNets significantly outperform other CNNs. In particular, EfficientNet-B7 achieves new state-of-the-art 84.3% top-1 accuracy for being 8.4% smaller and 4.4% faster than GoNet. EfficientNet-B7 is 1.4x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

time larger. However, the process of scaling up ConvNets has never been well understood and there are currently many ways to do it. The most common way is to scale up ConvNets by their depth (He et al., 2016) or width (Zagoruyko & Komodakis, 2016). Another less common, but increasingly popular, method is to scale up models by image resolution (Huang et al., 2016). In previous work, it is common to scale only one of the three dimensions—depth, width, and image size. Though it is possible to scale two or three dimensions

independently, uniform scaling requires tedious manual tuning

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

1. INTRODUCTION

Abstract—Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called Graph Transformer Networks (GTN), allows each module to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks. A Graph Transformer Network for reading bank checks is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provide record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

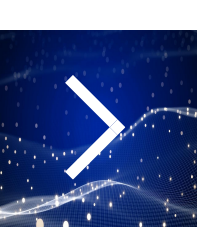
Keywords—Neural Networks; OCR; Document Recognition; Machine Learning; Gradient-Based Learning; Convolutional Neural Networks; Graph Transformer Networks; Finite State Transducers.

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed knowledge. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on grid images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually integrating individually designed modules can be replaced by a unified and well-structured design paradigm, called Graph Transformer Networks, that allows training all the modules to optimize a global performance criterion.

Since the early days of pattern recognition it has been known that the variability and richness of natural data, be it speech, glyphs, or other types of patterns, make it

ing residual functions with a instead of learning unrefined, preheuristic empirical evidence networks are easier to optimize considerably increased depth evaluate residual nets with deeper than VGG nets [41]. An ensemble of these res- The ILSVRC 2015 classification on CIFAR-10 with 100 and the depth of representa- for many visual recognition tasks deeply deep representation improvement on the COCO residual nets are foundation A COCO 2015 competition places on the tasks of Image-caption, COCO detection, as



Empirical evidences in deep learning...

Squeeze-and-Excitation Networks

Jie Hu^[0000-0002-5150-1003] Li Shen^[0000-0002-2283-4976] Samuel Albanie^[0000-0001-9786-5134]
Gang Sun^[0000-0001-6913-6799] Enhua Wu^[0000-0002-2174-1428]

Abstract—The central building block of convolutional neural networks (CNNs) is the convolution operator, which enables networks to construct informative features by fusing both spatial and channel-wise information within local receptive fields at each layer. A broad range of prior research has investigated the spatial component of this relationship, seeking to strengthen the representational power of a CNN by enhancing the quality of spatial encodings throughout its feature hierarchy. In this work, we focus instead on the channel relationship and propose a novel architectural unit, which we term the ‘‘Squeeze-and-Excitation’’ (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. We show that these blocks can be stacked together to form SE-Net architectures that generalise extremely effectively across different datasets. We further demonstrate that SE blocks bring significant improvements in performance for existing state-of-the-art CNNs at slight additional computational cost. Squeeze-and-Excitation Networks formed the foundation of our ILSVRC 2017 classification submission which won first place and reduced the top-5 error to 2.351%, surpassing the winning entry of 2016 by a relative improvement of ~25%. Models and code are available at <https://github.com/hujie-frank/SENet>.

Index Terms—Squeeze-and-Excitation, Image representations, Attention, Convolutional Neural Networks.

Densely Connected Convolutional Networks

Gao Huang^{*} Tsinghua University
gshi98@cornell.edu
Zhuang Liu^{*} Tsinghua University
liuzhuang13@mails.tsinghua.edu.cn
Kilian Q. Weinberger^{*} Cornell University
kq2@cornell.edu
Laurens van der Maaten^{*} Facebook AI Research
lvdm@fb.com

Attention Is All You Need

Ashish Vaswani^{*} Google Brain
avaswani@google.com
Noam Shazeer^{*} Google Brain
noam@google.com
Niki Parmar^{*} Google Research
nikip@google.com
Jakob Uszkoreit^{*} Google Research
usz@google.com
Llion Jones^{*} Google Research
llion@google.com
Aidan N. Gomez^{*} University of Toronto
aidan@cs.toronto.edu
Lukas Kaiser^{*} Google Brain
lukasz.kaiser@google.com
Illia Polosukhin[†]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe Google Inc., sioffe@google.com
Christian Szegedy Google Inc., szegedy@google.com

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer’s inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making each layer a part of the model architecture and performing normalization for each training mini-batch. Batch normalization allows us to use much higher learning rates, less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification Batch Normalization achieves the same accuracy with fewer training steps, and beats the original by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best result on ImageNet classification: reaching 4.1% validation error (and 4.8% test error), exceeding the accuracy of human raters.

Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a batch can be much more efficient than *in* computations for individual examples, due to the parallelism afforded by the modern computing platforms.

MobileNetV2: Inverted Residuals and Linear Bottlenecks

Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen
{sandler, howard, menglong, azhmogin, lchen}@google.com

Abstract

In this paper we describe a new mobile architecture, MobileNetV2, that improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through a reduced form of DeepLabv3 which we call Mobile DeepLabv3. This is based on an inverted residual structure where the shortcut connections are between the skip bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. Additionally, we find that it is important to remove non-linearities in the narrow layers maintain representational power. We demonstrate improved performance and provide an insight into this design. Our approach allows decoupling of the insights from the expressiveness of the transform provides a convenient framework for us. We measure our performance on classification, COCO object detection [1], segmentation [1]. We evaluate the trade-off accuracy, and number of operations measured by (Madd), as well as actual latency, and parameters.

This paper introduces a new neural network architecture that is specifically tailored for mobile and resource constrained environments. Our network pushes the state of the art for mobile tailored computer vision models, by significantly decreasing the number of operations and memory needed while retaining the same accuracy. Our main contribution is a novel layer module: the inverted residual with linear bottleneck. This module takes as an input a low-dimensional compressed representation which is first expanded to high dimension and filtered with a lightweight depthwise convolution. Features are subsequently projected back to a low-dimensional representation with a linear convolution. The official implementation is available as part of TensorFlow-Slim model library in [1].

This module can be efficiently implemented using standard operations in any modern framework and allows our models to beat state of the art along multiple performance points using standard benchmarks. Furthermore, this convolutional module is particularly suitable for mobile designs, because it allows to significantly reduce the memory footprint needed during inference by never fully materializing large intermediate tensors. This reduces the need for main memory access in many embedded hardware designs, that provide small amounts of very fast software controlled cache memory.

2. Related Work

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

1. INTRODUCTION

Abstract—Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

Back-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modelling. A new learning paradigm, called Graph Transformer Networks (GTN), allows each multi-module system to be trained globally using Gradient-Based methods so as to minimize an overall performance measure. Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks. A Graph Transformer Network for reading task check is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provide record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

Keywords—Neural Networks; OCR; Document Recognition; Machine Learning; Gradient-Based Learning; Convolutional Neural Networks; Graph Transformer Networks; Finite State Transducers.

Abstract
We investigate the effect of the convolutional network depth on its scale image recognition setting. Our main contribution is of networks of increasing depth using an architecture with rotation filters, which shows that a significant improvement in accuracy can be achieved by pushing the depth to 16–19 layers. We are the basis of our ImageNet Challenge 2014 team secured the first and the second places in the localisation tracks respectively. We also show that our representations on datasets, where they achieve state-of-the-art results. We first-performing ConvNet models publicly available to facilitate the use of deep visual representations in computer vision.

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{khe, v-xiangz, v-shren, jiansun}@microsoft.com

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

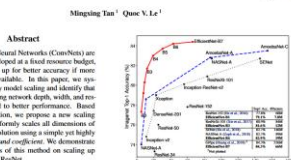


Figure 1: Model Size vs. ImageNet Accuracy. All numbers are for single-core, single-model. Our EfficientNet significantly outperforms other CNNs. In particular, EfficientNet-B0 achieves state-of-the-art 84.3% top-1 accuracy for being 8.4x smaller and 6.1x faster than VGG16. EfficientNet-B1 is 1.4x smaller and 5.7x faster than ResNet-101. Details are in Table 2 and 4.

time larger. However, the process of scaling up CNNs has never been well understood and there are currently many ways to do it. The most common way is to scale up CNNs by their depth (He et al., 2016) or width (Zagoruyko & Komodakis, 2016). Another less common, but increasingly popular, method is to scale up models by image resolution (Huang et al., 2016). In previous work, it is common to scale only one of the three dimensions—depth, width, and image size. Though it is possible to scale two or three dimensions simultaneously, uniform scaling reduces network training

Export Control COEDM – EU EC: country FR – marking N / US EC: marking N / OTH EC: country N – marking N | Export Control Other – AL: N / ECCN: N



... with (on-going) theoretical insights

Published as a conference paper at ICLR 2018

DEEP NEURAL NETWORKS AS GAUSSIAN PROCESSES

Jaehoon Lee^{*1}, Yasaman Bahri^{*1}, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, Jascha Sohl-Dickstein

Google Brain
{jaehlee, yasamanb, romann, schsam, jpen}



ABSTRACT

It has long been known that a single-layer full i.i.d. prior over its parameters is equivalent to of infinite network width. This corresponds for infinite width neural networks on regress corresponding GP. Recently, kernel function neural networks have been developed, but only As such, previous work has not identified th variance functions for GPs and allow fully Ba network.

In this work, we derive the *exact equivalence* works and GPs. We further develop a compu pute the covariance function for these GPs. We form Bayesian inference for wide deep neu 10. We observe that trained neural network ac sponding GP with increasing layer width, and correlated with trained network prediction er mance increases as finite-width trained netwo to a GP, and thus that GP predictions typical networks. Finally we connect the performan of signal propagation in random neural netwo

Neural Networks, Vol. 2, pp. 359–366, 1989
Printed in the USA. All rights reserved.

ORIGINAL CONTRIBUTION

Multilayer Feedforward Networks are Universal Approximators

KURT HORNIK

Technische Universität Wien

MAXWELL STINCHCOMBE AND HALBERT WHITE

University of California, San Diego

(Received 16 September 1988; revised and accepted 9 March 1989)

Abstract—This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.



G. Cybenko



NEUROCOMPUTING

Neurocomputing 25 (1999) 81–91

Lower bounds for approximation by MLP neural networks

Vitaly Maiorov*, Allan Pinkus

Department of Mathematics, Technion, I.I.T., Haifa, 32000 Israel

Received 30 May 1998; accepted 6 December 1998

Abstract

The degree of approximation by a single hidden layer A layer is bounded below by the degree of approximation functions. We prove that there exists an analytic, stric function for which this lower bound is essentially attain activation function, that one can approximate arbitrarily compact domain by a two hidden layer MLP using a layer. © 1999 Elsevier Science B.V. All rights reserved.

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

Google Brain

{jaehlee, xlc, schsam, yasamanb, romann, jaschaad, jpenning}@google.com

Abstract

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 for wide neural networks the learning dynamics simplify considerably and that, in the infinite width limit, they are governed by a linear model obtained from the first-order Taylor expansion of the network around its initial parameters. Furthermore, mirroring the correspondence between wide Bayesian neural networks and Gaussian processes, gradient-based training of wide neural networks with a squared loss produces test set predictions drawn from a Gaussian process with a particular compositional kernel. While these theoretical results are only exact in the infinite width limit, we nevertheless find excellent empirical agreement between the predictions of the original network and those of the linearized version even for finite practically-sized networks. This agreement is robust across different architectures, optimization methods, and loss functions.

Neural Tangent Kernel: Convergence and Generalization in Neural Networks

Arthur Jacot
École Polytechnique F
arthur.jacot@

Franck Coeurwille
Imperial College London and École F
francrgabrie

Clément Hong
École Polytechnique F
clement.hong@

Abstract

At initialization, artificial neural networ processes in the infinite-width limit (16) kernel methods. We prove that the evolue described by a kernel: during gradient the network function f_0 (which maps inp kernel gradient of the functional cost (whi cost) w.r.t. a new kernel: the Neural Tangent Kernel (NTK).

res of ning, stays is in function space instead of parameter space. n be related to the positive-definiteness of the e-definiteness of the limiting NTK when the the non-linearity is non-polynomial.

source compression and show that in the infinite

On the Spectral Bias of Neural Networks

Nasim Rahaman^{*1,2} Aristide Baratin^{*1} Devansh Arpit¹ Felix Draxler² Min Lin¹ Fred A. Hamprecht² Yoshua Bengio¹ Aaron Courville¹

Abstract

Neural networks are known to be a class of highly expressive functions able to fit even random input-output mappings with 100% accuracy. In this work we present properties of neural networks that complement this aspect of expressivity. By using tools from Fourier analysis, we highlight a learning bias of deep networks towards low frequency functions – i.e. functions that vary globally without local fluctuations – which manifests itself as a frequency-dependent learning speed. Intuitively, this property is in line with the observation that over-parameterized networks prioritize learning simple patterns that generalize across data samples. We also investigate the role of the shape of the data manifold by presenting empirical and theoretical evidence that, somewhat counter-intuitively, learning higher frequencies gets easier with increasing manifold complexity.

expose this bias by taking a closer look at neural networks through the lens of Fourier analysis. While they can approximate arbitrary functions, we find that these networks prioritize learning the low frequency modes, a phenomenon we call the *spectral bias*. This bias manifests itself not just in the process of learning, but also in the parameterization of the model itself: in fact, we show that the lower frequency components of trained networks are more robust to random parameter perturbations. Finally, we also expose and analyze the rather intricate interplay between the spectral bias and the geometry of the data manifold by showing that high frequencies get easier to learn when the data lies on a lower-dimensional manifold of complex shape embedded in the input space of the model. We focus the discussion on networks with rectified linear unit (ReLU) activations, whose continuous piece-wise linear structure enables an analytic treatment.

Contributions¹

1. We show that the low frequency components of trained networks are more robust to random parameter perturbations.

Minimum Width for Universal Approximation

Sejun Park*, Chulhee Yun†, Jaeho Lee*, Jinw

June 17, 2020

Abstract

The universal approximation property of width-bounded networks has been of classical universal approximation results on depth-bounded networks. How width enabling the universal approximation has not been exactly characterized input dimension d_x and the output dimension d_y . In this work, we provide a result in this direction for networks using the ReLU activation functions: The required for the universal approximation of the L^p functions is exactly $\max\{d_x, d_y\}$ prove that the same conclusion does not hold for the uniform approximation does hold with an additional threshold activation function. Our proof technique to derive a tighter upper bound on the minimum width required for the universal approximation using networks with general activation functions.

Reconciling modern machine learning practice and the bias-variance trade-off

Mikhail Belkin^a, Daniel Hsu^b, Siyuan Ma^a, and Soumik Mandal^a

^aThe Ohio State University, Columbus, OH

^bColumbia University, New York, NY

September 12, 2019

Abstract

breakthroughs in machine learning are rapidly changing science and society, yet our fundamental understanding of this technology has lagged far behind. Indeed, one of the central issues of the field, the bias-variance trade-off, appears to be at odds with the observed behavior of machine learning models used in the modern machine learning practice. The bias-variance trade-off implies a model should balance under-fitting and over-fitting: rich enough to express underlying structure in data, simple enough to avoid fitting spurious patterns. However, in the modern practice, very rich models such as neural networks are trained to exactly fit (i.e., interpolate) training data. Classically, such models would be considered over-fit, and yet they often obtain high accuracy on test data. This apparent contradiction has raised questions about the mathematical foundations of machine learning and their relevance to practitioners.

In this paper, we reconcile the classical understanding and the modern practice within a new performance curve. This “double descent” curve subsumes the textbook U-shaped bias-variance trade-off curve by showing how increasing model capacity beyond the point of interpolation results in improved performance. We provide evidence for the existence and ubiquity of double descent for a wide spectrum of models and datasets, and we posit a mechanism for emergence. This connection between the performance and the structure of machine learning models delineates the limits of classical analyses, and has implications for both the theory and practice of machine learning.

Universal Approximation with Deep Narrow Networks

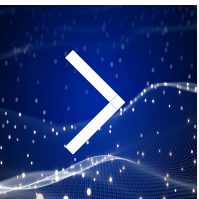
Patrick Kidger
Terry Lyons

Mathematical Institute, University of Oxford

KIDGER@MATHS.OX.AC.UK
TLYONS@MATHS.OX.AC.UK

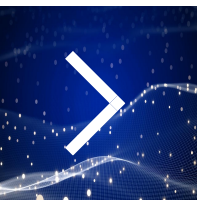
Abstract

The classical Universal Approximation Theorem holds for neural networks of arbitrary width and bounded depth. Here we consider the natural “dual” scenario for networks of bounded width and arbitrary depth. Precisely, let n be the number of input neurons, m be the number of output neurons, and let ρ be any nonaffine continuous function, with a continuous nonzero derivative at some point. Then we show that the class of neural networks of arbitrary depth, width $n + m + 2$, and activation function ρ , is dense in $C(K; \mathbb{R}^m)$ for $K \subseteq \mathbb{R}^n$ with K compact. This covers every activation function possible to use in practice, and also includes polynomial activation functions, which is unlike the classical version of the theorem, and provides a qualitative difference between deep narrow networks and shallow wide networks. We then consider several extensions of this result. In particular we consider nowhere differentiable activation functions, density in noncompact domains with respect to the L^p -norm, and how the width may be reduced to just $n + m + 1$ for “most” activation functions.



Questions

- In the last five years or so, a parallel effort to empirical studies has grown to ask why any of these empirical researches works:
 - How does gradient descent, run on a wildly non-convex loss, reliably converge?
 - What does neural network learn first, and in what order?
 - **Why do over-parametrized networks generalize instead of just memorizing?**



Motivations

Reconciling modern machine learning practice and the bias-variance trade-off

Mikhail Belkin^a, Daniel Hsu^b, Siyuan Ma^a, and Soumik Mandal^a

^aThe Ohio State University, Columbus, OH

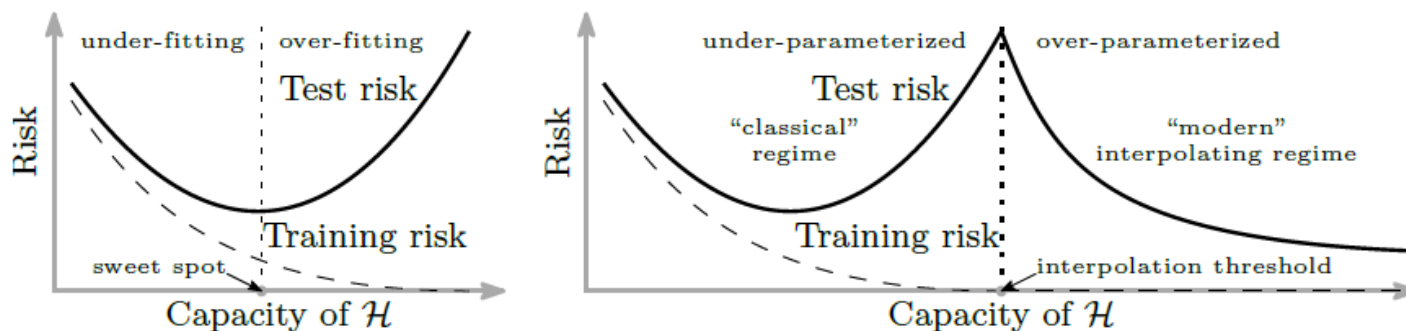
^bColumbia University, New York, NY

September 12, 2019

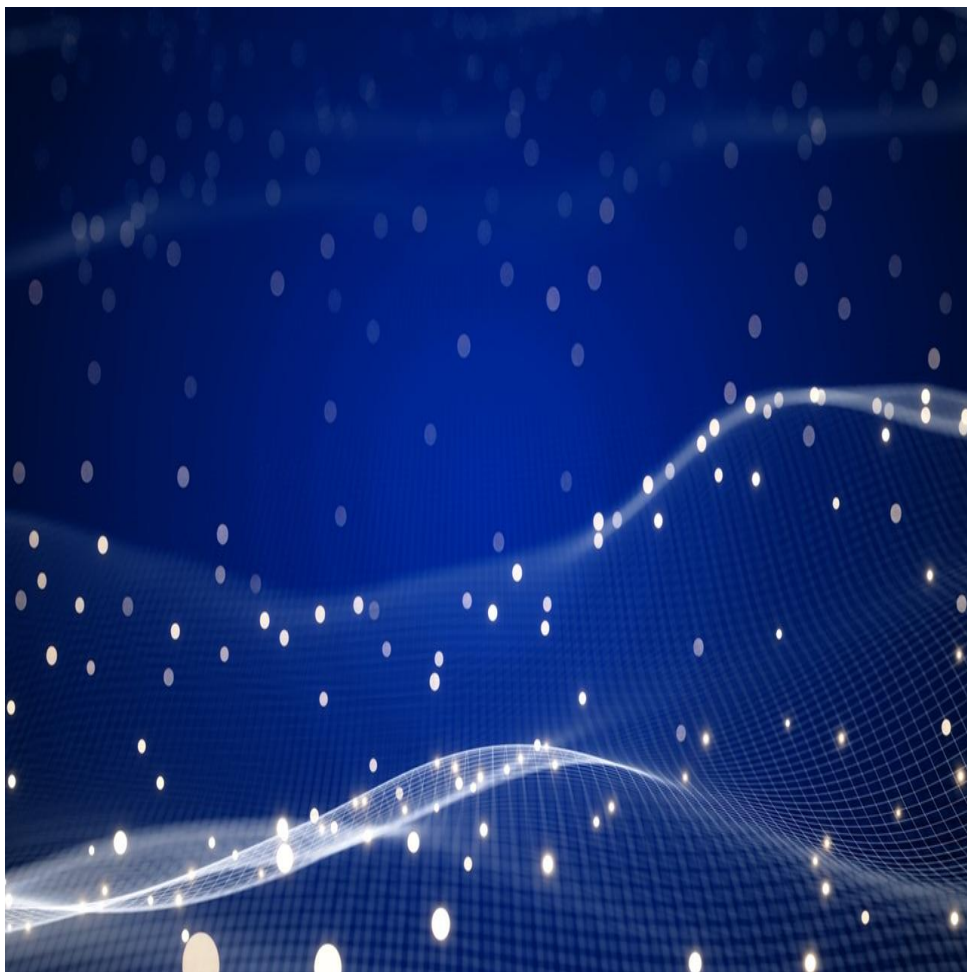
Abstract

Breakthroughs in machine learning are rapidly changing science and society, yet our fundamental understanding of this technology has lagged far behind. Indeed, one of the central tenets of the field, the bias-variance trade-off, appears to be at odds with the observed behavior of methods used in the modern machine learning practice. The bias-variance trade-off implies that a model should balance under-fitting and over-fitting: rich enough to express underlying structure in data, simple enough to avoid fitting spurious patterns. However, in the modern practice, very rich models such as neural networks are trained to exactly fit (i.e., interpolate) the data. Classically, such models would be considered over-fit, and yet they often obtain high accuracy on test data. This apparent contradiction has raised questions about the mathematical foundations of machine learning and their relevance to practitioners.

In this paper, we reconcile the classical understanding and the modern practice within a unified performance curve. This “double descent” curve subsumes the textbook U-shaped bias-variance trade-off curve by showing how increasing model capacity beyond the point of interpolation results in improved performance. We provide evidence for the existence and ubiquity of double descent for a wide spectrum of models and datasets, and we posit a mechanism for its emergence. This connection between the performance and the structure of machine learning models delineates the limits of classical analyses, and has implications for both the theory and practice of machine learning.

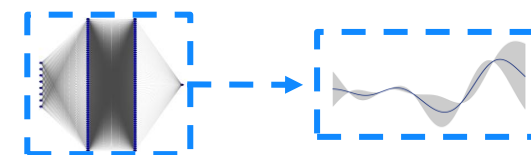


Reference: [1] M. Belkin, D. Hsu, S. Ma, & S. Mandal,
Reconciling modern machine-learning practice and the classical bias-variance trade-off,
Proc. Natl. Acad. Sci. U.S.A. 116 (32), 2019.

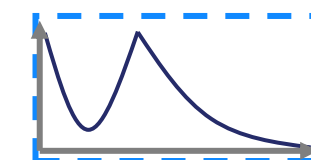
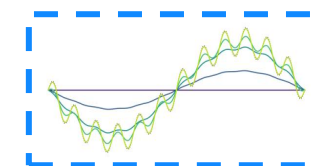


Contents

- 1 From Artificial Neural Network to Gaussian process
- 2 Gradient flow dynamics and Neural Tangent Kernel
- 3 Spectral bias
- 4 Over-parametrization and double descent
- 5 Conclusions and perspectives



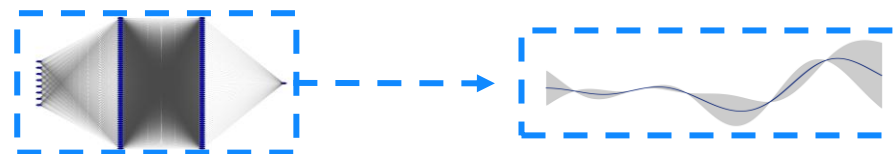
$$\frac{d\theta(t)}{dt} = -\nabla_{\theta} \mathcal{L}(\theta)$$

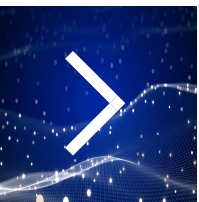




1.

From Artificial Neural Network to Gaussian process





Neural Network Gaussian Process (NNGP)

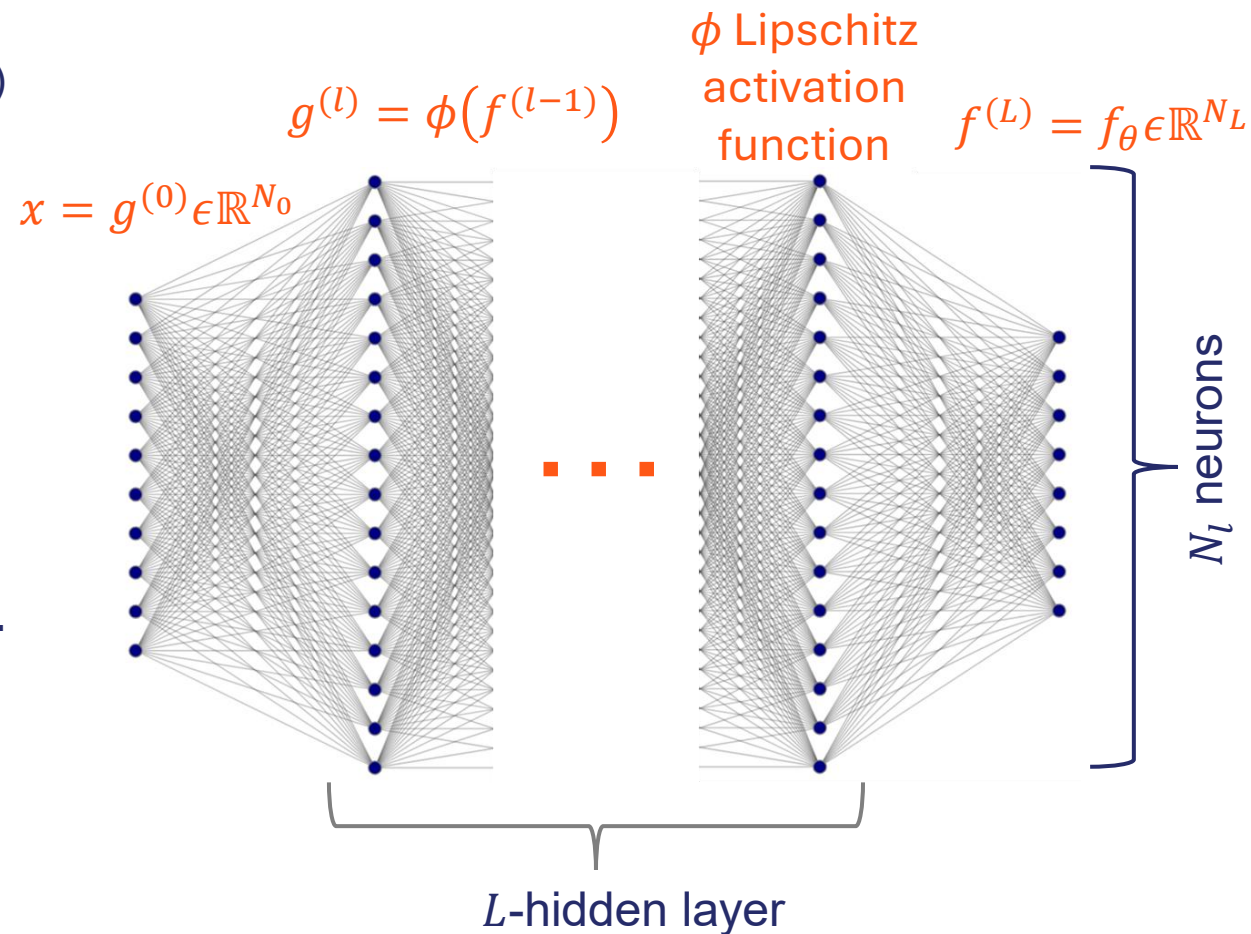
- We consider an **L -hidden Multi-Layer Perceptron (MLP)**
 $f_{\theta}: \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_L}$, $f_{\theta}(x) = f^L(x)$:

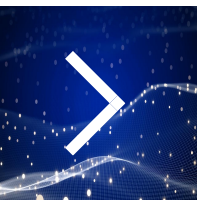
Initialization: $g^{(0)} = x$

Pre – activation: $f_i^{(l)}(x) = b_i^{(l)} + \frac{1}{\sqrt{N_l}} \sum_{j=1}^{N_l} W_{ij}^{(l)} g_j^{(l)}(x)$

Post – activation: $g_j^{(l)}(x) = \phi \left(f_j^{(l-1)}(x) \right)$.

- Weight and bias parameters for the l -th layer are **i.i.d.**
 $W^{(l)} \sim \mathcal{N}(0, 1)$ and $b^{(l)} \sim \mathcal{N}(0, 1)$.
- $\theta = \{W^{(0)}, b^{(0)}, \dots, W^{(L)}, b^{(L)}\} \in \mathbb{R}^P$.





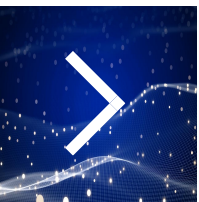
Neural Network Gaussian Process (NNGP)

- In 1996, Neal [2] (for 1 hidden layer), and later in 2018, Lee et al. [3] (for deep network) proved that when $N_l \rightarrow +\infty$, a **MLP tends to a GP**.
- Indeed, following Neal, for 1 hidden layer and the i -th component:

$$\text{Pre-activation: } f_{\theta}(x) = f_i^{(1)}(x) = b_i^{(1)} + \frac{1}{\sqrt{N_1}} \sum_{j=1}^{N_1} W_{ij}^{(1)} g_j^{(1)}(x).$$

$$\text{Post-activation: } g_j^{(1)}(x) = \phi\left(f_j^{(0)}(x)\right) = \phi\left(b_j^{(0)} + \frac{1}{\sqrt{N_0}} \sum_{k=1}^{N_0} W_{jk}^{(0)} x_k\right).$$

Input layer

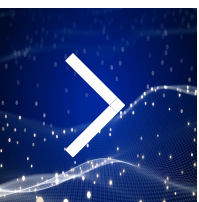


Neural Network Gaussian Process (NNGP)

- > In 1996, Neal [2] (for 1 hidden layer), and later in 2018, Lee et al. [3] (for deep network) proved that when $N_l \rightarrow +\infty$, a **MLP tends to a GP**.
- > Indeed, following Neal, for 1 hidden layer and the i -th component:

$$\text{Pre-activation: } f_\theta(x) = f_i^{(1)}(x) = \underbrace{b_i^{(1)}}_{\text{i.i.d.}} + \frac{1}{\sqrt{N_1}} \sum_{j=1}^{N_1} \underbrace{W_{ij}^{(1)} g_j^{(1)}(x)}_{\text{i.i.d.}}. \quad \text{By Central Limit Theorem: } f_i^{(1)} \rightarrow \mathcal{N} \text{ when } N_1 \rightarrow +\infty.$$

$$\text{Post-activation: } g_j^{(1)}(x) = \phi\left(f_j^{(0)}(x)\right) = \phi\left(\underbrace{b_j^{(0)}}_{\text{i.i.d.}} + \frac{1}{\sqrt{N_0}} \sum_{k=1}^{N_0} \underbrace{W_{jk}^{(0)} x_k}_{\text{i.i.d.}}\right). \quad \Rightarrow g_j^{(1)} \text{ and } g_{j'}^{(1)} \text{ are independent for } j \neq j'.$$



Neural Network Gaussian Process (NNGP)

- In 1996, Neal [2] (for 1 hidden layer), and later in 2018, Lee et al. [3] (for deep network) proved that when $N_l \rightarrow +\infty$, a **MLP tends to a GP**.
- Indeed, following Neal, for 1 hidden layer and the i -th component:

Pre-activation: $f_\theta(x) = f_i^{(1)}(x) = \underbrace{b_i^{(1)}}_{\text{i.i.d.}} + \frac{1}{\sqrt{N_1}} \sum_{j=1}^{N_1} \underbrace{W_{ij}^{(1)} g_j^{(1)}(x)}_{\text{i.i.d.}}.$ **By Central Limit Theorem:**
 $f_i^{(1)} \rightarrow \mathcal{N}$ when $N_1 \rightarrow +\infty$.

Post-activation: $g_j^{(1)}(x) = \phi\left(f_j^{(0)}(x)\right) = \phi\left(\underbrace{b_j^{(0)}}_{\text{i.i.d.}} + \frac{1}{\sqrt{N_0}} \sum_{k=1}^{N_0} \underbrace{W_{jk}^{(0)} x_k}_{\text{i.i.d.}}\right).$ $\Rightarrow g_j^{(1)}$ and $g_{j'}^{(1)}$ are independent for $j \neq j'$.

- From the multidimensional Central Limit Theorem, any finite collection of $\{f_i^{(1)}(x_1), \dots, f_i^{(1)}(x_k)\}$ will have a joint multivariate Gaussian distribution.
- Thus, it is a Gaussian process:

$$f_i^{(1)}(x) \sim \mathcal{GP}(\mu^{(1)}, \Sigma^{(1)}),$$

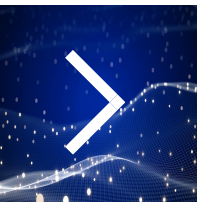
$$\text{with } \mu^{(1)}(x) = \mathbb{E}[f_i^{(1)}(x)] = 0 \text{ and } \Sigma^{(1)}(x, x') = \mathbb{E}[f_i^{(1)}(x)f_i^{(1)}(x')] = 1 + \frac{1}{N_1} \mathbb{E}[g_i^{(1)}(x)g_i^{(1)}(x')].$$



Neural Network Gaussian Process (NNGP)

- › Lee et al. [3] proved that the work of Neal can be extended to deeper layers by **induction**.
- › The output functions of a L -hidden network f_θ , are i.i.d. centered Gaussian process of covariance, defined recursively as:

$$\Sigma^{(0)}(x, x') = 1 + \frac{1}{N_0} x^T x',$$
$$\Sigma^{(l)}(x, x') = \mathbb{E} \left[f_i^{(l)}(x) f_i^{(l)}(x') \right] = 1 + \frac{1}{N_l} \mathbb{E}_{f_i^{(l-1)} \sim \mathcal{GP}(0, \Sigma^{(l-1)})} \left[\phi \left(f_i^{(l-1)}(x) \right) \phi \left(f_i^{(l-1)}(x') \right) \right].$$



Neural Network Gaussian Process (NNGP)

- Lee et al. [3] proved that the work of Neal can be extended to deeper layers by **induction**.
- The output functions of a L -hidden network f_θ , are i.i.d. centered Gaussian process of covariance, defined recursively as:

$$\Sigma^{(0)}(x, x') = 1 + \frac{1}{N_0} x^T x',$$
$$\Sigma^{(l)}(x, x') = \mathbb{E} \left[f_i^{(l)}(x) f_i^{(l)}(x') \right] = 1 + \frac{1}{N_l} \mathbb{E}_{f_i^{(l-1)} \sim \mathcal{GP}(0, \Sigma^{(l-1)})} \left[\phi \left(f_i^{(l-1)}(x) \right) \phi \left(f_i^{(l-1)}(x') \right) \right].$$

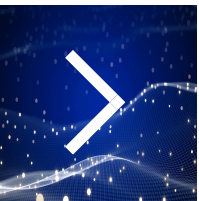
- Thus, in NNGP, each layer replaces a set of **random hidden features** with the population of **all possible random hidden features**.
- At initialization, an infinitely-wide network is equivalent to sampling a function from a $\mathcal{GP}(0, \Sigma^{(L)})$, where $\Sigma^{(L)}$ gives the similarity between x and x' .
- Indeed, when ϕ is the ReLU, Cho et al. [4] proved that the kernel is **the arc-cosinus**. Thus, the kernel computes the angle between input vectors in feature space. For highly correlated $\phi(u)$ and $\phi(v)$, the angle will be very small, and vice versa.
- NNGP only describes the network at **initialization**. It doesn't explain what happens during gradient descent on all layers.



2.

Gradient flow dynamics and Neural Tangent Kernel

$$\frac{d\theta(t)}{dt} = -\nabla_{\theta} \mathcal{L}(\theta)$$



Gradient flow dynamics

- › Empirical loss function $\mathcal{L}: \mathbb{R}^P \rightarrow \mathbb{R}_+$ over N samples using a per-sample cost function $\ell: \mathbb{R}^{N_0} \times \mathbb{R}^{N_L} \rightarrow \mathbb{R}_+$:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(f_{\theta}(x^{(i)}), y^{(i)}).$$

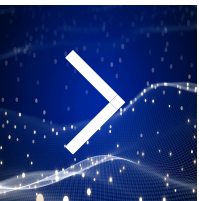
- › Gradient descent update rule:

$$\theta(t+1) = \theta(t) - \eta \nabla_{\theta} \mathcal{L}(\theta) \Leftrightarrow \frac{\theta(t+1) - \theta(t)}{\eta} = -\nabla_{\theta} \mathcal{L}(\theta)$$

with η the learning rate.

- › Gradient flow dynamics, when $\eta \rightarrow 0$:

$$\frac{d\theta(t)}{dt} = -\nabla_{\theta} \mathcal{L}(\theta): \text{Ordinary Differential Equation (ODE).}$$



Neural Tangent Kernel (NTK)

- Following Jacot et al. [5], according to the **chain rule**, the gradient of the loss is:

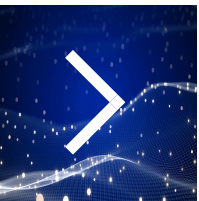
$$\frac{d\theta}{dt} = -\nabla_{\theta} \mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \nabla_{\theta} f_{\theta}(x^{(i)}) \nabla_{f_{\theta}} \ell(f_{\theta}, y^{(i)}).$$

- Again, by the **chain rule**, the network output evolves according to the derivative:

$$\frac{df_{\theta}(x)}{dt} = \frac{df_{\theta}(x)}{d\theta} \frac{d\theta}{dt} = -\frac{1}{N} \sum_{i=1}^N \underbrace{\nabla_{\theta} f_{\theta}(x)^T \nabla_{\theta} f_{\theta}(x^{(i)})}_{\text{Neural Tangent Kernel (NTK)}} \nabla_{f_{\theta}} \ell(f_{\theta}, y^{(i)}).$$

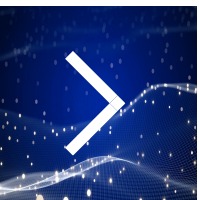
$K: \mathbb{R}^{N_0} \times \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_L \times N_L}$

- The NTK is defined as: $K(x, x'; \theta) = \nabla_{\theta} f_{\theta}(x)^T \nabla_{\theta} f_{\theta}(x')$.
- The feature map form is: $\varphi(x) = \nabla_{\theta} f_{\theta}(x) \Rightarrow K(x, x'; \theta) = \varphi(x)^T \varphi(x')$. It measures how much correcting the error at x' moves the prediction at x .
- If input x and input x' have gradients pointing in the same direction, updating parameters to fix the error at x' will simultaneously move the prediction at x in the same direction.
- If the gradients are nearly orthogonal (small values of $K(x, x'; \theta)$), training on x' barely touches $f_{\theta}(x)$.



NTK properties

- › Jacot et al. [5] proved (by induction) that when $N_1, \dots, N_L \rightarrow \infty$:
 1. **At initialization**, $K(x, x'; \theta(t=0))$ **converges** (in probability) to a **deterministic** kernel $K_\infty(x, x')$, independent of the random draw of $\theta(t=0)$.
 2. **During training** (gradient flow), $K(x, x'; \theta)$ stays constant, equal to $K_\infty(x, x')$. The kernel **remains frozen** throughout training.
- › Naively $K(x, x'; \theta)$ depends on $\theta(t)$ which evolves. But in the infinite-width limit, each individual parameter **moves by a vanishing amount** $\mathcal{O}\left(\frac{1}{\sqrt{N_l}}\right)$ even though their collective effect on f_θ is $\mathcal{O}(1)$.
- › So, the kernel **itself doesn't change**.
- › In other words, $\varphi(x) = \nabla_{\theta(t)} f_{\theta(t)}(x) \approx \nabla_{\theta(t=0)} f_{\theta(t=0)}(x)$ **remains almost constant** during the training.

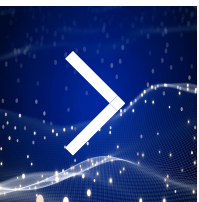


Lazy regime

- › The NTK $K_\infty(x, x')$ is frozen, thus the network behaves as its first-order Taylor expansion around initialization [6, 7]:

$$f_{\theta(t)}(x) \approx f_{\theta(t)}^{lin}(x) = f_{\theta(t=0)}(x) + \underbrace{\nabla_{\theta} f_{\theta(t=0)}}_{\varphi(x): \text{constant}} (\theta(t) - \theta(t=0)).$$

- › Thus, in infinite width, training a network (with **non-linear activation**) is equivalent to **training a linear model** on fixed random features $\nabla_{\theta} f_{\theta(t=0)}$.
- › In other words, during the training, the parameters stay around the initialization.
- › This is what we call **lazy regime or NTK regime**.



Kernel components interpretation

- › Jacot et al. [5] also proved (still by induction) that (when $N_1, \dots, N_L \rightarrow \infty$), the NTK can be defined recursively for each layer by:

$$K^{(l)}(x, x') = \Sigma^{(l)}(x, x') + K^{(l-1)}(x, x') \dot{\Sigma}^{(l)}(x, x'),$$

where $\Sigma^{(l)}$ involves the derivatives of the activation function with respect to their inputs.

- › To summarize:

$$K_{\infty}(x, x') = K(x, x'; \theta) = K^{(L)}(x, x') = \sum_{l=1}^L \Sigma^{(l)}(x, x') \prod_{l'=l+1}^L \dot{\Sigma}^{(l')}(x, x').$$

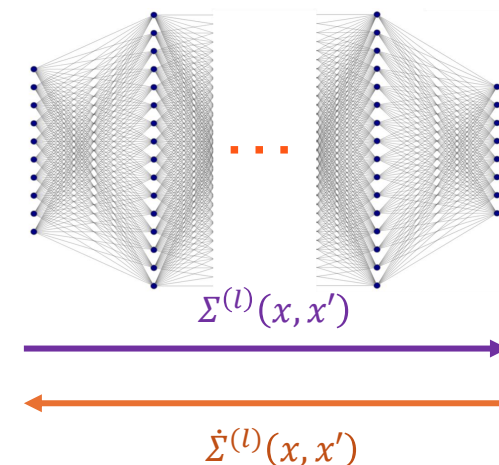
- › Example for 3 layers:

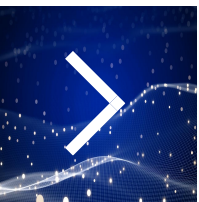
$$K_{\infty}(x, x') = K^{(3)}(x, x') = \Sigma^{(3)}(x, x') + \dot{\Sigma}^{(3)}(x, x') \Sigma^{(2)}(x, x') + \dot{\Sigma}^{(3)}(x, x') \dot{\Sigma}^{(2)}(x, x') \Sigma^{(1)}(x, x').$$

- › We can recognize 2 components:

- “NNGP kernel” $\Sigma^{(l)}(x, x')$: it gives a “**forward sensitivity**”, i.e., the similarity between x and x' .
- “Derivative kernel” $\dot{\Sigma}^{(l')}(x, x')$: it gives a “**backward local sensitivity**” based on the gradient.

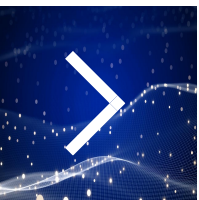
- › Thus, NTK encodes the training dynamics of the neural network.





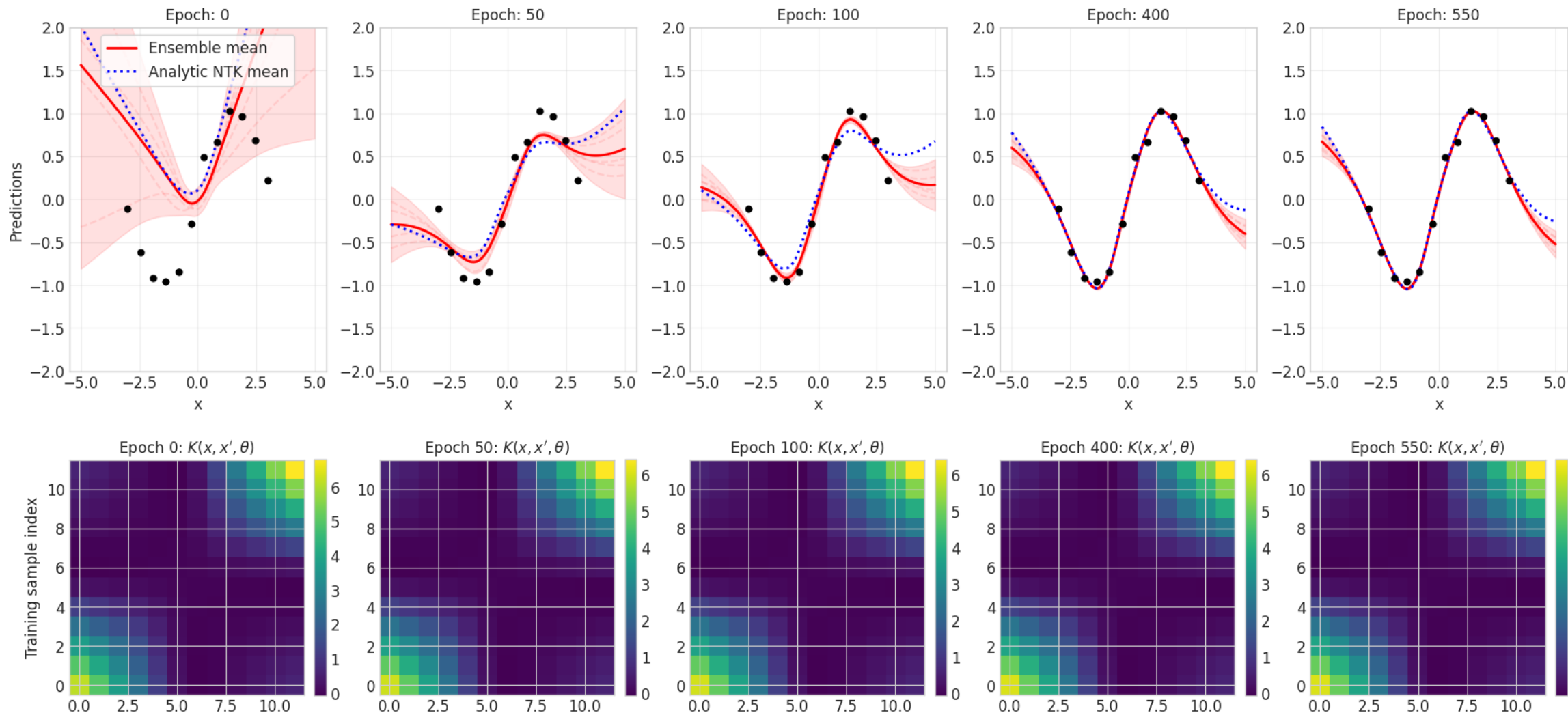
Empirical study

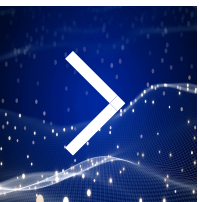
- › Toy function: $f(x) = \sin(x) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0,1)$.
- › MLP with 2048 neurons (*infinite*) and 3 layers.
- › Stochastic Gradient Descent (SGD) optimizer.
- › Number of iterations: 600.
- › Activation function: GELU (Gaussian Error Linear Units) $GELU(x) = x \cdot \Phi(x)$ with Φ the cumulative distribution function for Gaussian distribution.



Empirical study

Evolution under NTK parameter scaling (SGD) - 3 layers, width=2048 neurons

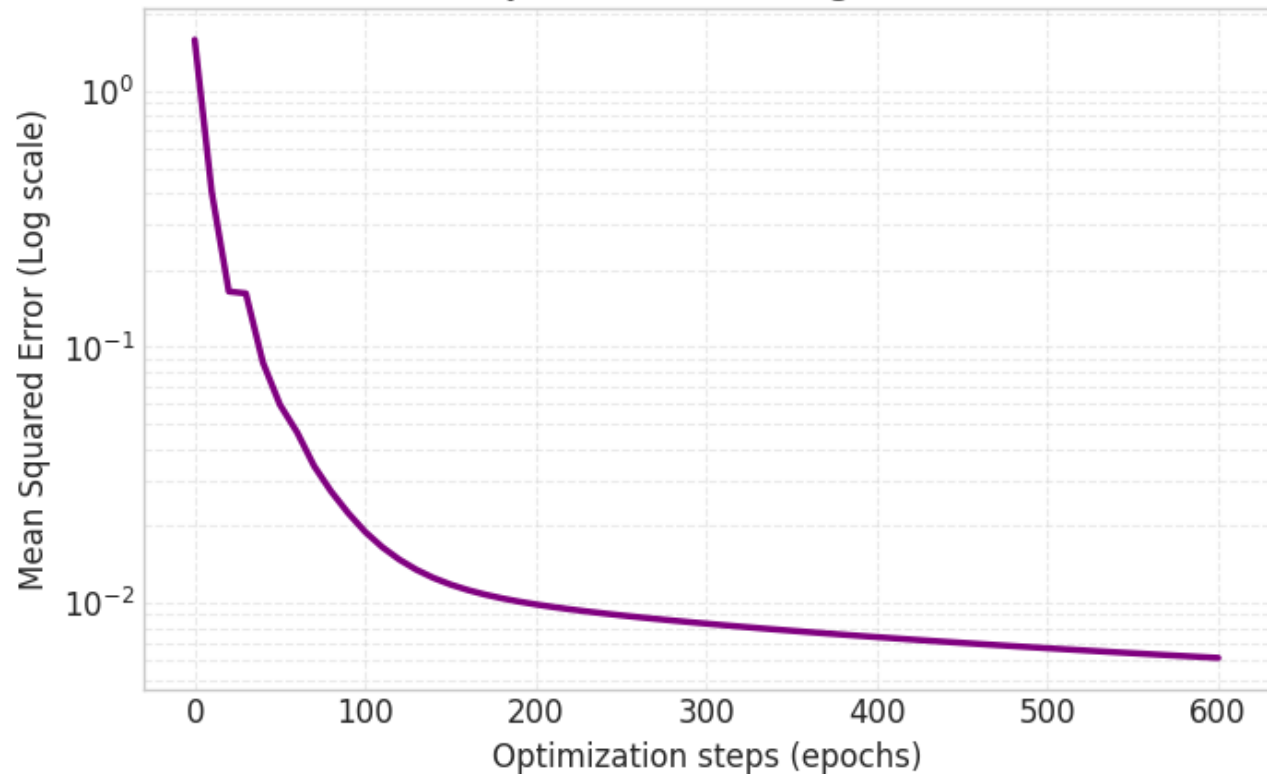




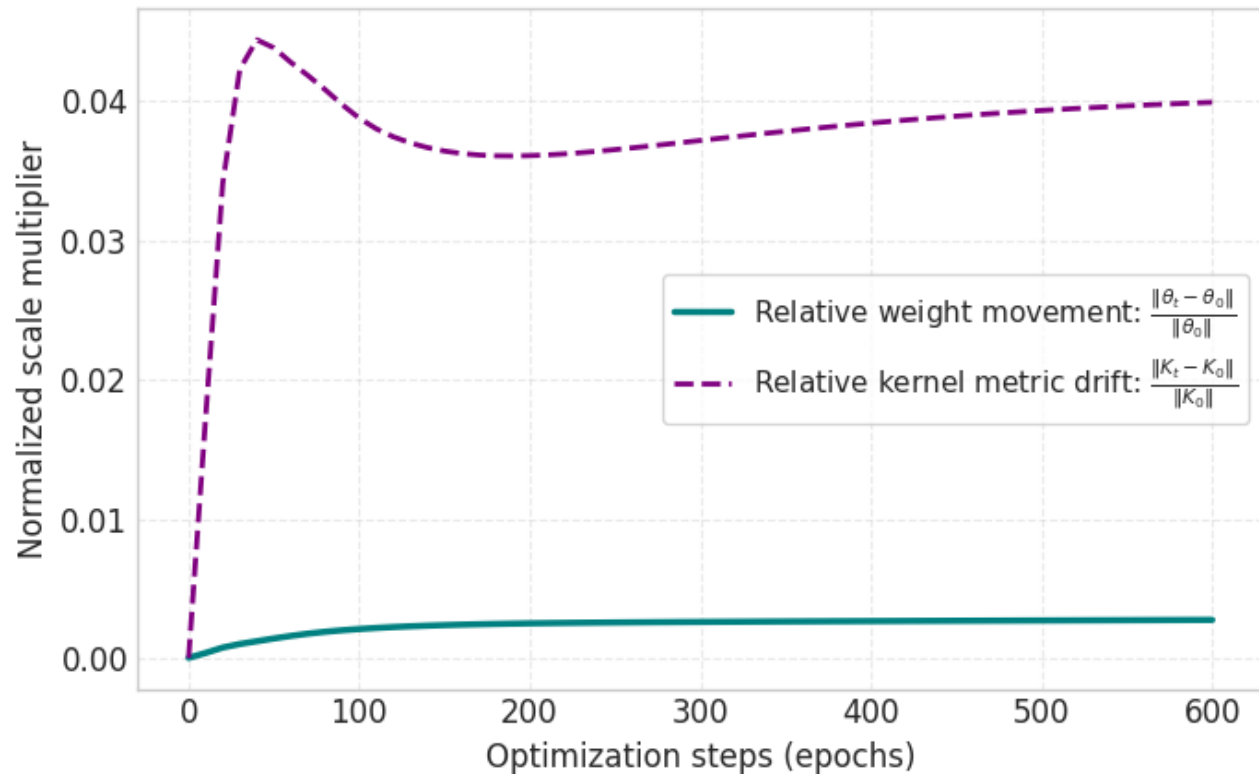
Empirical study

The Lazy Training: global convergence with near-zero feature change

Optimization convergence



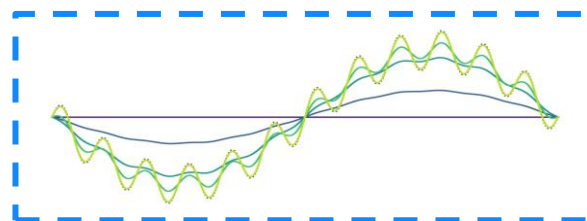
Parameter evolution

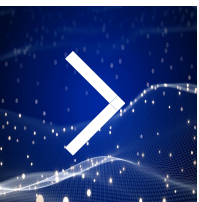




3.

Spectral bias





Kernel regression

- Back to NTK, we have:

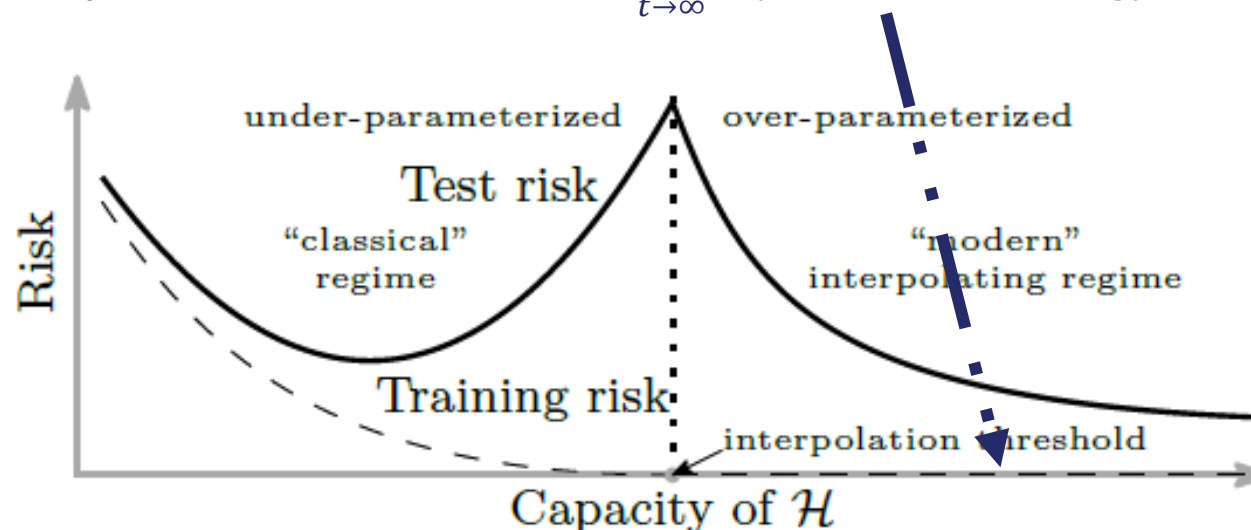
$$\frac{df_{\theta}(x)}{dt} = -K_{\infty}(x, x)\nabla_{\theta}\mathcal{L}(\theta).$$

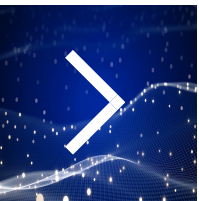
- For example, in regression task (Mean Squared Error or MSE loss): $\nabla_{\theta}\mathcal{L}(\theta) = (f_{\theta}(x) - y)$.
- So, we can solve:

$$\frac{df_{\theta}(x)}{dt} = -K_{\infty}(x, x)(f_{\theta}(x) - y).$$

- At the end, we have (for simplicity, we suppose $f_{\theta(t=0)}(x) = 0$):

$$f_{\theta}(x) = y - \exp(-K_{\infty}(x, x)t)y \xrightarrow{t \rightarrow \infty} y \text{ (zero error training).}$$





Kernel regression

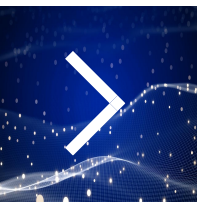
- › For test point x^* , we still have an ODE:

$$\frac{df_{\theta}(x^*)}{dt} = -K_{\infty}(x^*, x)(f_{\theta}(x) - y) = K_{\infty}(x^*, x) \exp(-K_{\infty}(x, x)t)y.$$

- › The solution is (for simplicity, we suppose $f_{\theta(t=0)}(x^*) = 0$):

$$f_{\theta}(x^*) = K_{\infty}(x^*, x)K_{\infty}(x, x)^{-1}y.$$

- › Training an infinitely wide network by gradient descent is equivalent to a **kernel (Ridge-less) regression using the NTK**.



Spectral bias

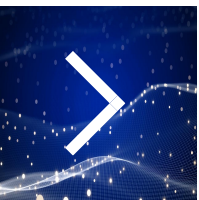
- › Since $K_\infty(x, x)$ is a real $N \times N$ symmetric matrix, we can diagonalize it:

$$K_\infty = Q\Lambda Q^T \text{ with eigenvalues } \lambda_1 \geq \dots \geq \lambda_N > 0.$$

- › Thus,

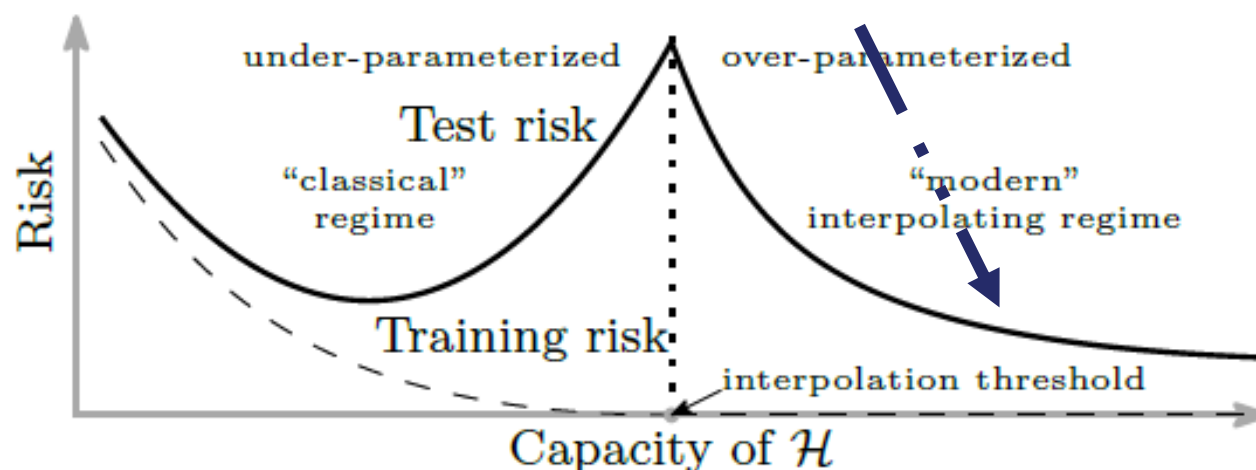
$$\exp(-K_\infty(x, x)t) = Q\exp(-\Lambda t)Q^T \text{ and } \exp(-\Lambda t) = \text{diag}(e^{-\lambda_1 t}, e^{-\lambda_2 t}, \dots, e^{-\lambda_N t}).$$

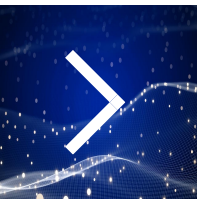
- › Each spectral component of the target is learned exponentially fast, with rate proportional to its eigenvalue λ_k .
- › Since NTK eigenvalues typically decay with the "frequency" of the corresponding eigenfunction, the gradient descent on (wide) neural nets fits **low-frequency / smooth components of the target function first**, and **high-frequency components only much later**.



Spectral bias

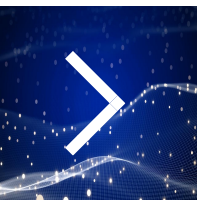
- Xu et al. [9] called that “frequency principle” or “**spectral bias**”: it is the phenomenon where neural networks trained with gradient descent learn low-frequency (smooth) functions first, fitting high-frequency (oscillatory or complex) details much later in training.
- This is a form of **implicit regularization** independent of any explicit penalty term.





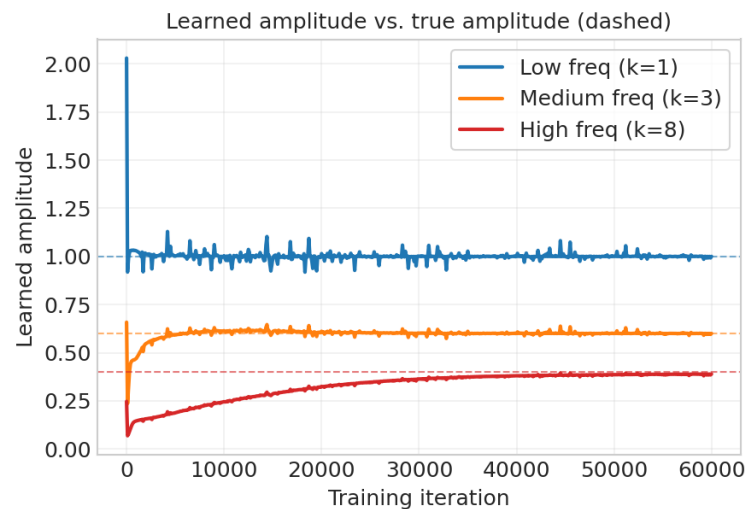
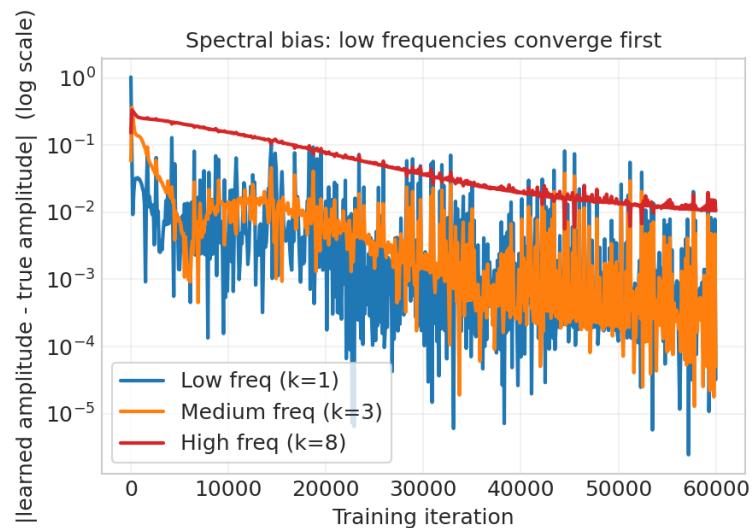
Empirical study

- › Toy function: $f(x) = \sin(2\pi \cdot 1x) + 0.6\sin(2\pi \cdot 3x) + 0.4\sin(2\pi \cdot 8x)$.
- › MLP with 2048 neurons (*infinite*) and 1 layer.
- › Adam optimizer (for faster training than SGD).
- › Number of iterations: 60000.
- › Activation function: ReLU.



Empirical study

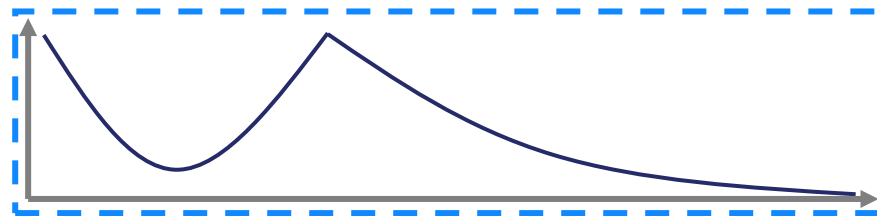
Spectral bias of gradient descent on a wide 1-hidden-layer MLP

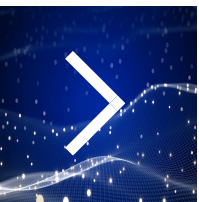




4.

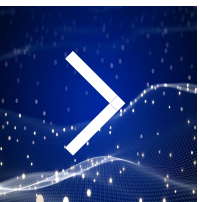
Over-parametrization and double descent



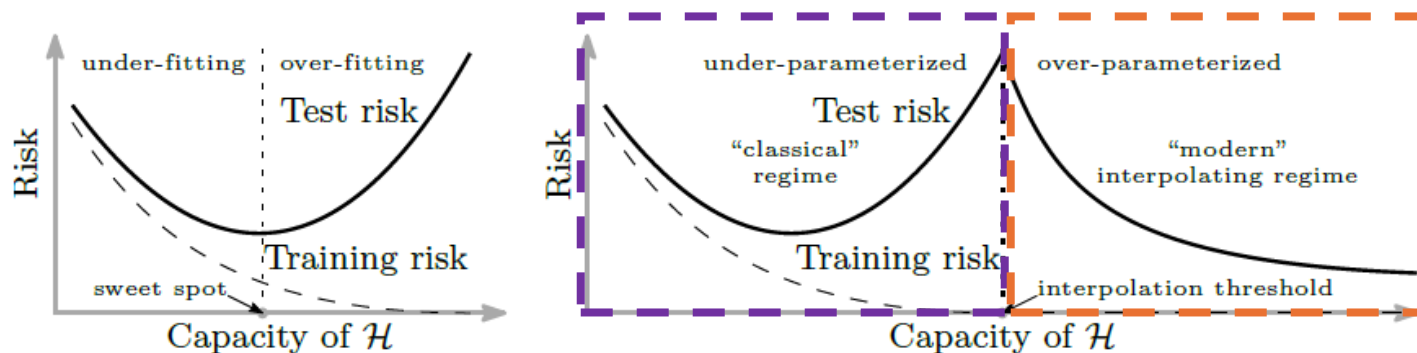


Summary

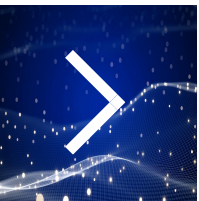
- › NTK properties: $\varphi(x) = \nabla_{\theta(t)} f_{\theta(t)}(x) \approx \nabla_{\theta(t=0)} f_{\theta(t=0)}(x)$ **remain almost constant** during the training.
- › Lazy regime: $f_{\theta(t)}(x) \approx f_{\theta(t)}^{lin}(x) = \varphi(x)^T \underbrace{(\theta(t) - \theta(t=0))}_w \approx \varphi(x)^T w$ (for simplicity, we suppose $f_{\theta(t=0)}(x) = 0$).
- › Kernel regression: for infinite width network, when $t \rightarrow \infty$: $f_{\theta}(x^*) = K_{\infty}(x^*, x) K_{\infty}(x, x)^{-1} y$ and $K_{\infty} = Q \Lambda Q^T$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_N > 0$.
- › Spectral bias: neural networks are strongly biased toward simple, **smooth functions early in training**.
- › Noise/high-frequency components correspond to **small eigenvalues and are fit last**, after the smooth signal has already been captured.
- › In NTK, SGD acts as a form of **implicit regularization** independent of any explicit penalty term.



Over-parametrization and double descent

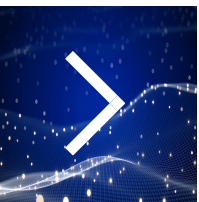


- Case of $P \lesssim N$: $K(x, x'; \theta)$ is nearly singular (at least $N - P$ zero eigenvalues), forcing $\|w\|$ (and thus test variance) to blow up. No interpolating solution, **only common bias-variance trade-off**.
- Case of $P \gg N$: the system $f_{\theta}(x^{(i)}) = y^{(i)}, \forall i$ is under-determined. Thus, gradient flow converges to $\theta(t \rightarrow \infty) - \theta(t = 0) = \operatorname{argmin}_w \|w\|_2$ s.t. $\varphi(x)^T w = y^{(i)}, \forall i$.
- Gradient descent, started at $\theta(t = 0)$ is lazy, i.e., it only ever moves in directions of data, so it stops at the interpolant that requires the **least total movement from initialization**. In feature space, this is strictly equivalent to the **minimum RKHS norm** (cf. representer theorem and O. Roustant lecture).
- In other words, $K(x, x'; \theta)$ is well-conditioned again (min eigenvalue bounded away from 0), and the min-RKHS-norm interpolant found by gradient descent is smooth by construction (spectral bias) rather than noise-fitting. So, test error descends a second time even though the model still interpolates the training data exactly.

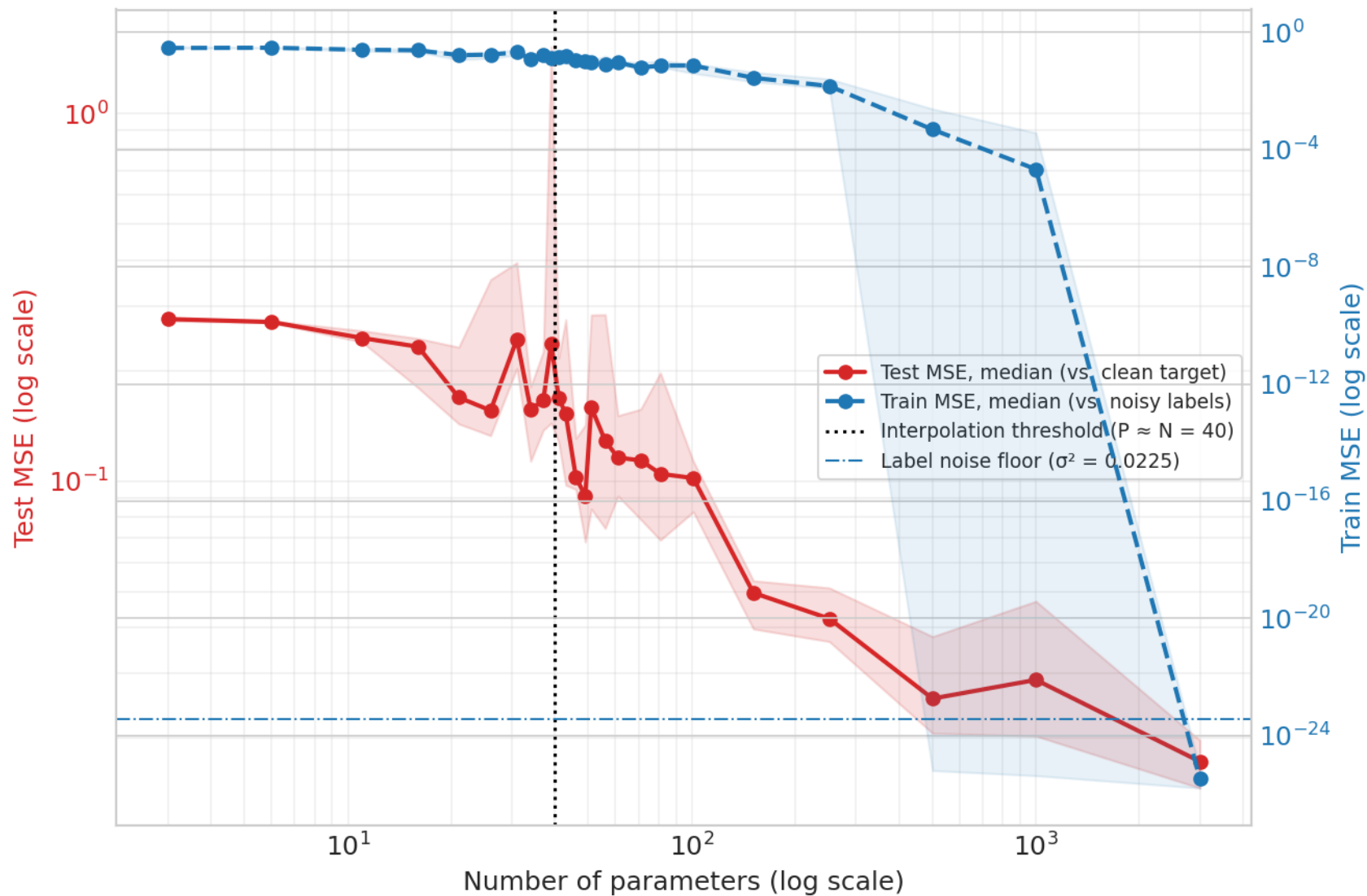


Empirical study

- › Toy function: $f(x) = \sin(2\pi \cdot 1x) + 0.6\sin(2\pi \cdot 3x) + 0.4\sin(2\pi \cdot 8x) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 0.15)$.
- › MLP with P neurons and 1 layer.
- › Activation function: ReLU.
- › Solving Ridge-less kernel regression. Indeed, gradient descent converges more and more slowly as the empirical NTK's smallest eigenvalue shrinks near the interpolation threshold (the very mechanism that causes the double-descent peak), so a fixed iteration budget badly under-trains exactly the widths that matter most. Solving in closed form avoids that confound.
- › Double precision calculation (float64).
- › 8 repetitions.
- › Use median + Inter-Quartile Range (IQR) rather than mean/std. Indeed, around the interpolation threshold, a small fraction of random-feature draws are numerically near-singular and produce astronomically large (but individually rare) test errors. The mean is dominated by those rare draws; the median is more robust while the IQR band still visibly widens near the threshold.



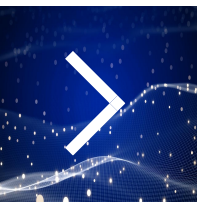
Empirical study





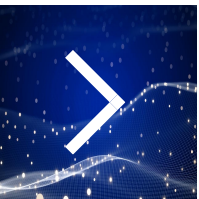
5.

Conclusion and perspectives



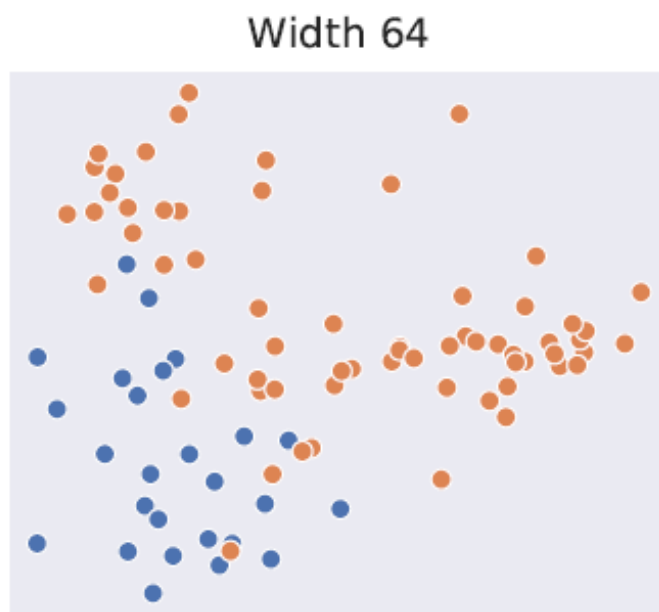
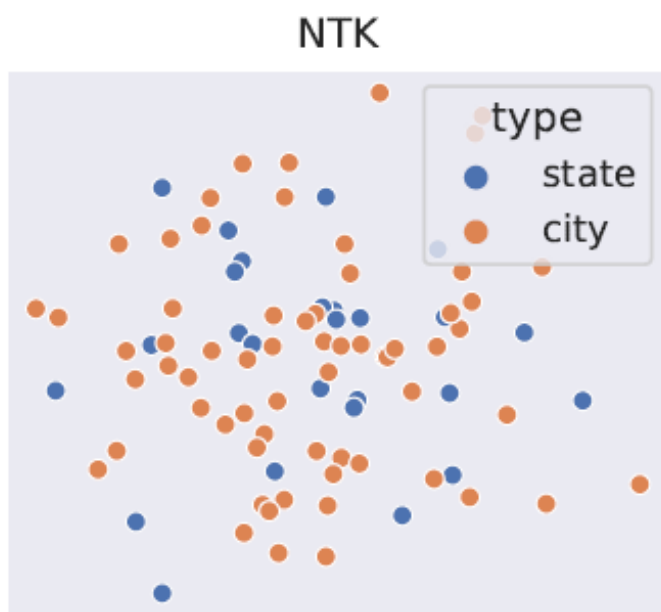
Conclusion

- › Multi-Layer Perceptron with infinite width tends to NNGP: network output is a Gaussian process at initialization.
- › Multi-Layer Perceptron with infinite width + gradient descent tends to NTK regime: the training can be summarized by a kernel regression with a fixed kernel K_∞ .
- › Spectral bias: convergence rate per NTK eigenvalue $\Rightarrow$ low frequencies learned first $\Rightarrow$ implicit smoothness bias $\Rightarrow$ implicit regularization.
- › Double descent: NTK regression's minimum-norm interpolation explains why over-parametrization beyond the interpolation threshold improves generalization again.

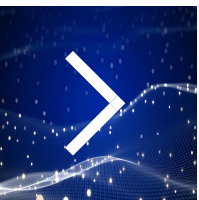


Perspectives

- NTK is a mathematical tool to better understand the training dynamics of neural network in a particular setup (infinite width).
- In practice, it is very hard to reach infinite width.
- Thus, it is much better to achieve feature learning regime instead of lazy regime.



Reference: [11] Greg Yang, Edward J. Hu, Feature Learning in Infinite-Width Neural Networks, arXiv:2011.14522, 2020.

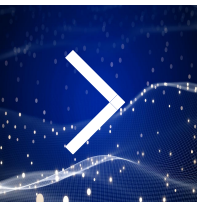


Perspectives

- › How to reach feature learning regime?
 - › With which setup, assumptions?
 - › Why is it better?
-
- › **To discuss all the answers, I invite you to follow the presentation of Eline Pot, PhD student.**

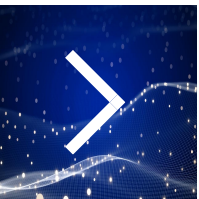


***Thanks for
your
attention***



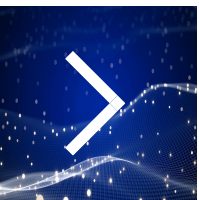
References

- [1] M. Belkin, D. Hsu, S. Ma, & S. Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off, Proc. Natl. Acad. Sci. U.S.A. 116 (32), 2019.
- [2] Radford M. Neal, Priors for Infinite Networks, Bayesian Learning for Neural Networks. Springer, New York, NY., 29-53, 1996.
- [3] Lee, Bahri et al., Deep Neural Networks as Gaussian Processes, ICLR 2018.
- [4] Youngmin Cho and Lawrence K. Saul, Kernel methods for deep learning, In Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS'09), 2009.
- [5] Arthur Jacot, Franck Gabriel, and Clément Hongler, Neural tangent kernel: convergence and generalization in neural networks, In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), 2018.
- [6] Eugene Golikov, Eduard Pokonechnyy, Vladimir Korviakov, Neural Tangent Kernel: A Survey, arXiv:2208.13614, 2022.
- [7] Lénaïc Chizat, Edouard Oyallon, and Francis Bach, On lazy training in differentiable programming. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 264, 2937-2947, 2019.
- [8] Ma, Yuheng & Xu, Zhiqin & Zhang, Jiwei, Frequency Principle in Deep Learning Beyond Gradient-descent-based Training, 10.48550/arXiv.2101.00747, 2021.
- [9] Canatar, Abdulkadir et al., Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks, Nature communications vol. 12,1 2914, 2021.



References

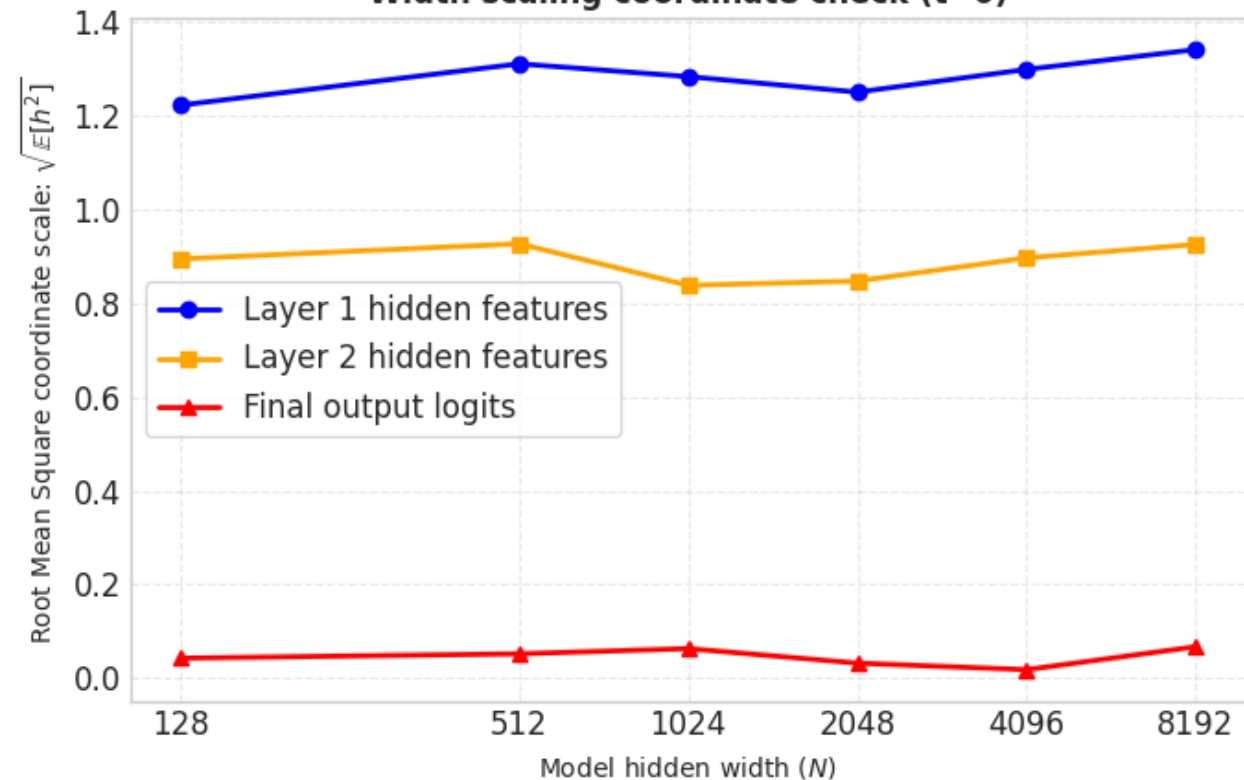
- [10] Jacot, Arthur & Simsek, Berfin & Spadaro, Francesco & Hongler, Clément & Gabriel, Franck, Kernel Alignment Risk Estimator: Risk Prediction from Training Data, 10.48550/arXiv.2006.09796, 2020.
- [11] Greg Yang, Edward J. Hu, Feature Learning in Infinite-Width Neural Networks, arXiv:2011.14522, 2020.



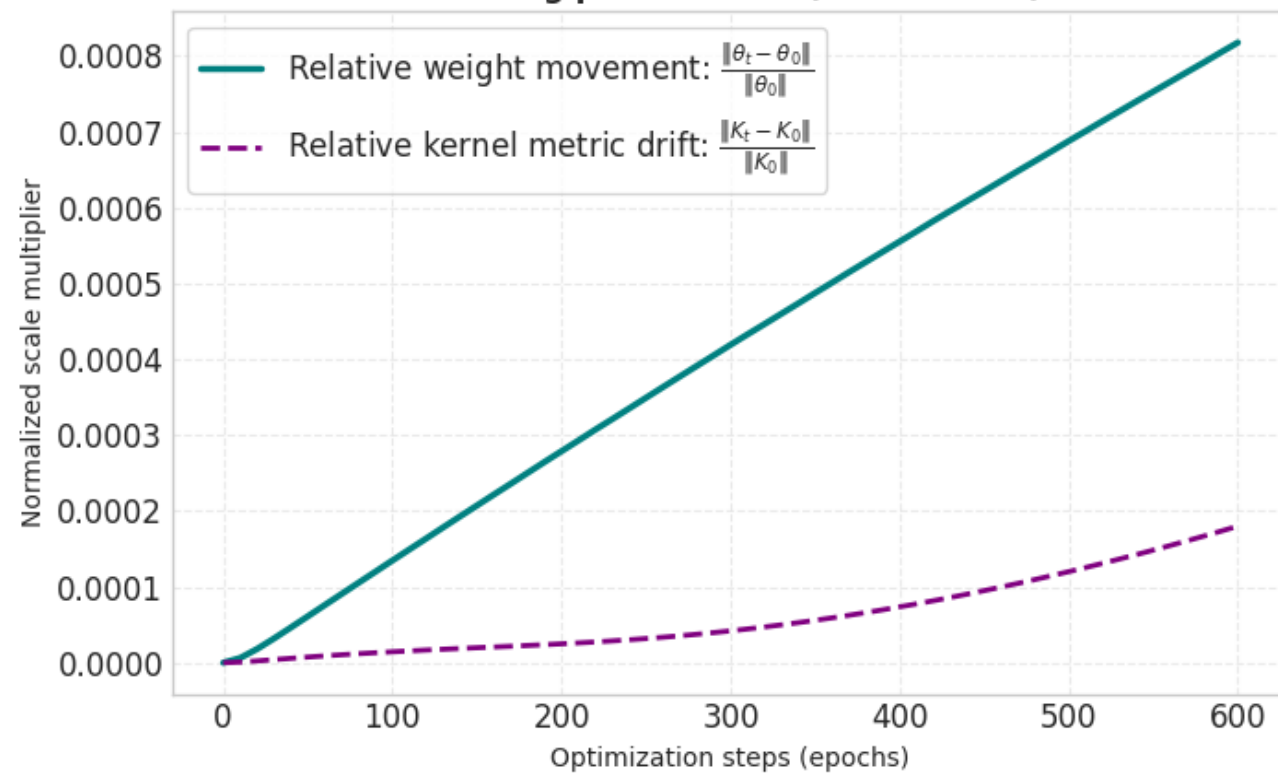
Empirical study

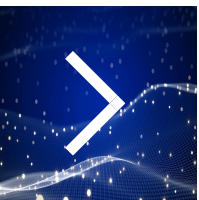
NTK verification: scaling invariant coordinates and frozen kernel - 3 layers

Width scaling coordinate check (t=0)

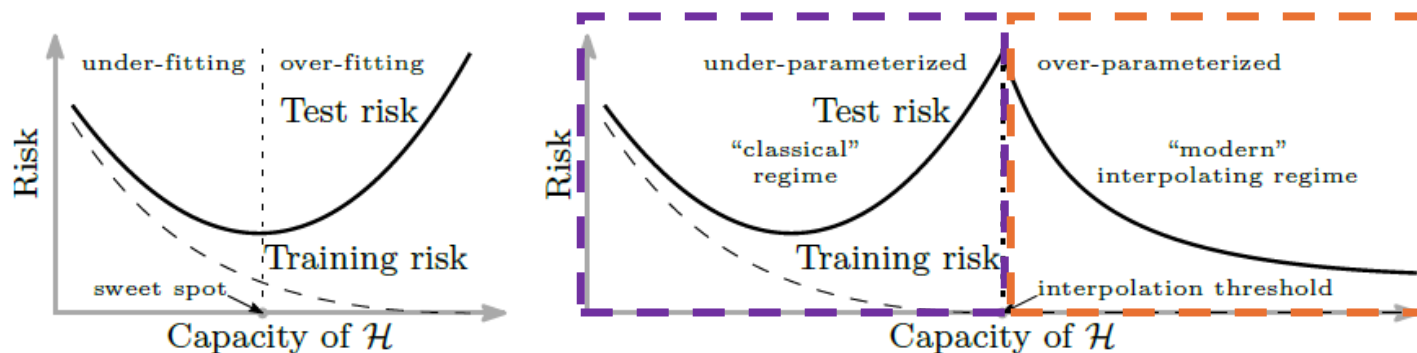


Tracking performance (width=8192)

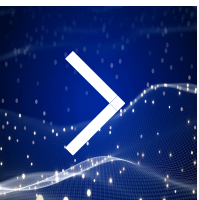




Over-parametrization and double descent



- **Representer theorem:** for any function of the form $f_w(x) = \varphi(x)^T w$, the RKHS norm induced by the kernel $K(x, x'; \theta) = \varphi(x)^T \varphi(x')$ is defined to equal $\|w\|_2$.
- **The NTK acts as a minimum RKHS norm interpolator.**



Over-parametrization and double descent

- › Expected prediction error decomposes (for squared-error loss) as:

$$\mathbb{E} \left[\left(y - \hat{f}(\mathbf{x}) \right)^2 \right] = \underbrace{\left(f(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})] \right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E} \left[\left(\hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})] \right)^2 \right]}_{\text{Variance}} + \underbrace{\sigma_\varepsilon^2}_{\text{Irreducible error}}.$$

- › This is the famous bias-variance trade-off:

- A model too simple $\Rightarrow$ high bias, low variance $\Rightarrow$ underfitting.
- A model too complex $\Rightarrow$ low bias, high variance $\Rightarrow$ overfitting.