Soutenance TER

Crochemar Théo

Derius Billy

Optimization and Generalization in Neural networks

Contents

I. Introduction to Deep Learning

Since 2012, deep learning has experienced rapid expansion, driven by the success of deep neural networks in diverse areas such as computer vision, natural language processing, and speech recognition. This breakthrough stems from neural networks' ability to automatically learn hierarchical representations from large datasets. However, this learning is only achievable through optimization algorithms, which are the driving force behind adjusting the internal parameters of neural networks. Without optimization, no learning occurs. Optimization represents a central component in deep learning architectures.

Problematic

How do the mathematical properties of optimization algorithms influence the performance of deep neural networks on a given task ?

Objectives and Methodology

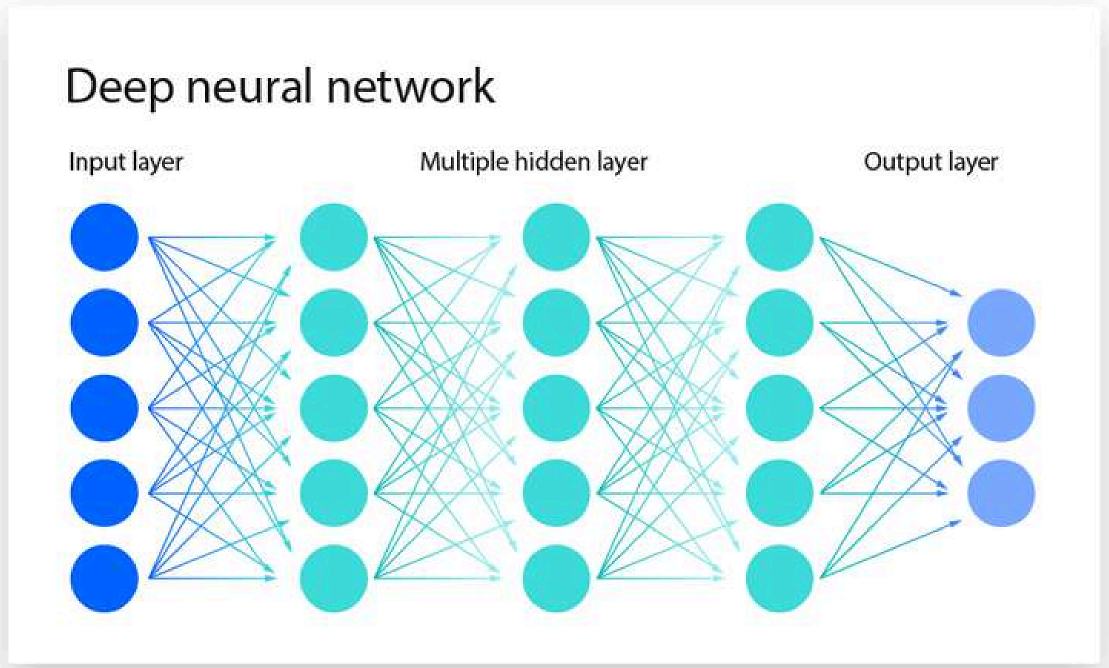
Objectives

The primary goal of this study was to enhance our understanding of the central role played by optimization in deep learning by combining rigorous theoretical analysis with comprehensive empirical evaluation.

Approach

The idea is to integrate mathematical modeling with concrete empirical observations to achieve a rigorous yet operational understanding of the performance of various optimization methods.

II. Neural Network



- ✓ Output of the neuron :
  $y = \sigma(w^\top x + b)$
- ✓ Functional representation :
  $f_\theta(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x)$
- ✓ Cybenko's Theorem (1989).

The different activation function

- **Sigmoid** : σ(z) = 1 / (1 + e^(–z)), saturating, less used today.
- **Tanh** : centered in 0,but also saturating.
- **ReLU** : σ(z) = max(0, z), very effiecient and fast to compute.
- **Leaky ReLU** : keep a steep for z < 0.
- **Softplus** : approximationof ReLU, log(1 + e^z).

The choice of σ impacts the convergence and the stability of the gradient.

Capacity of the network

Two major way

VC Dimension

Mesure how much the configureation of a network can exactly separate. Also the bigger it is, the more the network can express itself, but possibly under the constraint of overfitting.

Complexité de Rademacher

Mesure the capacity of the model to adapt to the noise randomly :

$\mathscr{R}_n(\mathscr{H}) = \mathbb{E}_\sigma [\sup_{h \in \mathscr{H}} (1/n) \Sigma\ \sigma_i h(x_i)]$, with σ_i, the variables of Rademacher. A low value of complexity favors generalization.

Cybenko's Theorem

Every network with only one hidden layer and one non polynomial activation can approximate any continous function on a compact set:

$\forall f \in C([0,1]^d), \forall \varepsilon > 0, \exists\ f_\theta$ such that $||f - f_\theta||_\infty < \varepsilon$

III. Optimization Methods

The neural networks are functions that are strongly parametized, their training comes down to solve an optimization problem non convex in space of high dimension In this part we explore :

- The geometry behind the landscape of the loss function
- The algorithm of descent,
- The garanties of convergence that we can obtain.

The landscape for optimization

The **loss function** $\mathscr{L}(\theta)$, where $\theta \in \mathbb{R}^n$ is the vector of the network's weights, is generally

- non-convex,
- sometimes non-smooth (e.g., because of ReLU activations),
- defined in a high-dimensional space.

This leads to:

- many saddle points ($\nabla \mathscr{L} = 0$ but the Hessian is indefinite),
- plateaus where $||\nabla \mathscr{L}|| \approx 0$ without being minima,
- flat vs. sharp minima, which are important for generalization.

Example: In deep linear networks, all local minima are global minima (Kawaguchi, 2016)

3 Groups of algorithm

First order :

- Gradient Descent: $\theta_{t+1} = \theta_t - \eta \nabla \mathscr{L}(\theta_t)$
- Stochastic GD: $\theta_{t+1} = \theta_t - \eta \nabla \mathscr{L}\_B(\theta_t)$ (mini-batch B)
- Momentum: $v_t = \beta v_{t-1} + \nabla \mathscr{L}(\theta_t)$, $\theta_{t+1} = \theta_t - \eta v_t$

Adaptative methods :

- Adam: $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$; $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
- $\theta_{t+1} = \theta_t - \eta \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$

Second-order methods :

- Newton: $\theta_{t+1} = \theta_t - H^{-1}(\theta_t) \nabla \mathscr{L}(\theta_t)$
- Natural Gradient: $\theta_{t+1} = \theta_t - \eta F^{-1}(\theta_t) \nabla \mathscr{L}(\theta_t)$, where F is the Fisher information matrix

The theory of convergence

Even without convexity, certain assumptions still guarantee convergence.

- Polyak–Łojasiewicz (PL) inequality:

There exists $\mu > 0$ such that

$\frac{1}{2} \|\nabla \mathscr{L}(\theta)\|^2 \geqslant \mu\,(\mathscr{L}(\theta) - \mathscr{L}^*)$ $\Rightarrow$ linear convergence.

- Lipschitz gradient:

If $\nabla \mathscr{L}$ is L-Lipschitz, then any step size $\eta < 1/L$ ensures monotonic decrease of the loss.

- Hilbert spaces — Radon-Riesz theorem:

If a sequence f_n converges weakly to f ($f_n \rightharpoonup f$) and $\|f_n\| \to \|f\|$, then f_n actually converges strongly to f.

IV. Generalization

Mechanism that allows a neural network to learn effectively and in a stable w

- The role of the bias, of the variance and of the noise in the phenomenon of generalization
- The statistical tools used to estimate model performance
- The recent theoretical frameworks for controlling generalization, including PAC-Bayesian bounds and algorithmic stability
- The double descent phenomenon, a modern behavior observed in highly over-parameterized neural network

Bias, variance and overfitting

1. the **bias:** the systematic gap between the average prediction and the true function
2. the **variance**: the instability of the predictor when the training set changes
3. the **irreducible noise**: the intrinsic variance of the response variable.

Theorem : Let ˆf(X) be an estimator of f(X) obtained from a training set, and assume that the data follows
Y = f(X) + ε, E[ε | X] = 0, Var(ε | X) = σ 2 .
Then the expected squared prediction error at a point x ∈ X is:
ED,ε h (Y − ˆf(x))2 i = ED[ˆf(x)] − f(x) 2 + ED h (ˆf(x) − ED[ˆf(x)])2 i + σ 2

Definition: Overfitting and the bias-variance balance :
A model is said to overfit when it achieves excellent performance on the training set but poor generalization on new data. In this case, the variance of the estimator dominates the bias. This occurs when the model is excessively flexible and captures not only the true signal but also the noise in the training data.

Two comparisons

Why over-fitting ⇒ high var

(i) **Training error falls** : At large C, $\hat{f}_C$ can interpolate the sample: $\hat{f}_C(x_i) = y_i \; \forall i$, so the empirical risk is close to 0

(ii) **Bias shrinks** : Greater flexibility lets $E[\hat{f}_C(x)]$ track $f(x)$ ever more closely, so Bias^2 decreases (often $\to 0$).

(iii) **Variance explodes** : To honour every sample point, the fitted curve must move drastically when any data point changes. Formally $\text{Var}\,\hat{f}_C(x) = E_D\,\hat{f}_C(x) - E_D[\hat{f}_C(x)]^2$ grows roughly like the effective degrees of freedom of the model, so for large C we have :

$\text{Var}\,\hat{f}_C(x) \gg \text{Bias}^2\,\hat{f}_C(x)$

Fit vs stability

Increasing model capacity improves the ability to fit training data but reduces stability: small changes in the training set lead to large variations in predictions. This instability manifests itself as high variance. Thus, there exists a fundamental tension between the model's fitting power and its generalization stability

Statistical parameter estimation

Statistical estimation theory tells us how to quantify the uncertainty that surrounds the unknown parameters of a learning model.

Starting from a statistical specification $y_i = f(x_i ; \theta) + \varepsilon_i$, we examine three classical estimators of θ: the maximum-likelihood estimator (MLE), the maximum-a-posteriori estimator (MAP) and the fully Bayesian posterior distribution.

The key tool for measuring their precision is the Fisher information.

Definition:
Statistical model : A statistical model is a family of probability distributions $M = \{ P_\theta : \theta \in \Theta \subseteq R^p \}$ on a common sample space, indexed by an unknown parameter θ.
In supervised learning we observe i.i.d. pairs $D = \{(x_i , y_i)\}_{i=1}^n$ drawn from $P_{\theta^\star}$, and the modelling assumption is $y_i = f(x_i ; \theta) + \varepsilon_i$, ε_i i.i.d. $N(0, \sigma^2)$, where $f(\cdot; \theta)$ is a deterministic regression function and the ε_i represent additive noise.
The task of parameter estimation is to construct from D an estimator $\hat{\theta}$ that is close to the ground-truth $\theta^\star$ in some sense.
Classical strategies include maximum-likelihood (MLE), maximum-a-posteriori (MAP), and full Bayesian inference.

Estimators of θ and Fisher

Estimators of θ

Let ℓ(θ) = Pn i=1 log p yi xi , θ denote the log-likelihood of the sample D = {(xi , yi)} n i=1.

1. Maximum-likelihood estimator (MLE) .

2 2. Maximum-a-posteriori estimator (MAP)

3. Bayesian estimator Fully Bayesian learning keeps the whole posterior p(θ | D) ∝ p(θ) exp{ℓ(θ)}.

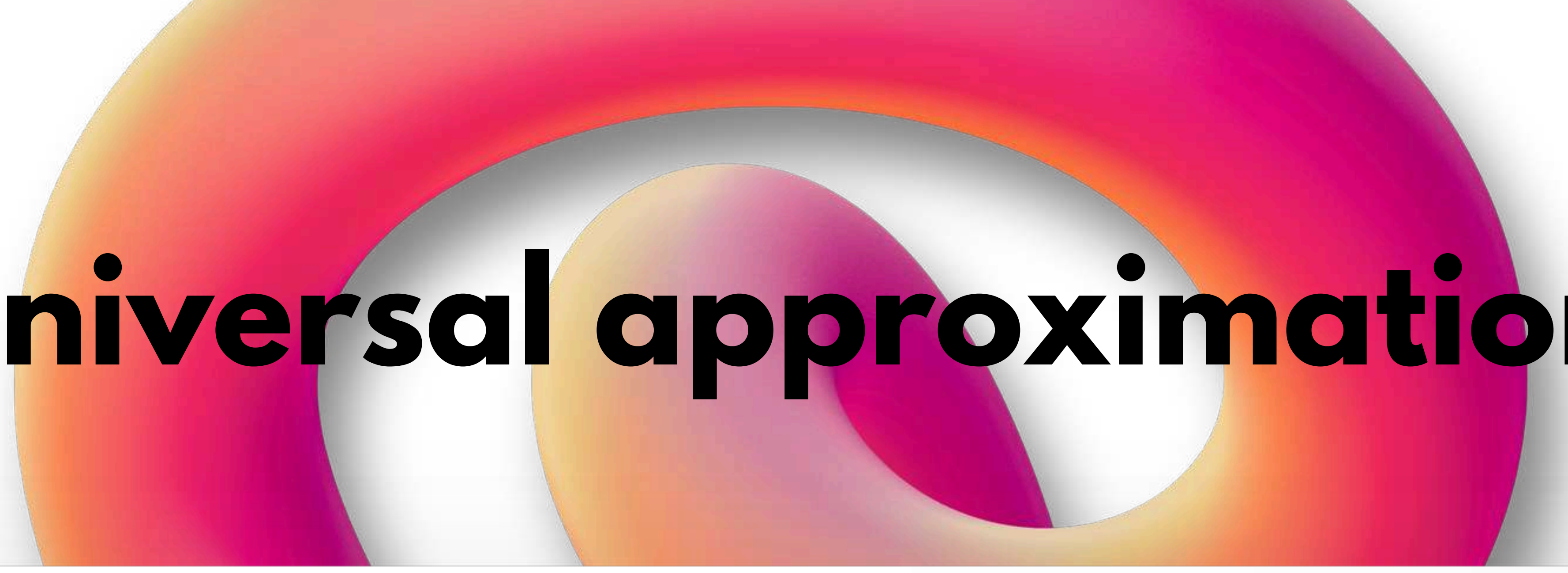
Fisher

Under the regularity conditions:

1. The partial derivative of f(X; θ) with respect to θ exists almost everywhere. (It can fail to exist on a null set, as long as this set does not depend on θ).
2. The integral of f(X; θ) can be differentiated under the integral sign with respect to θ.
3. The support of f(X; θ) does not depend on θ.

We get the alternative form.

It measures how “sharp” the log-likelihood is around θ, larger I(θ) implies tighter concentration of any regular unbiased estimator



Universal approximation

Cybenko's universal approximation theorem tells us that a feed-forward network with a single hidden layer and a non-polynomial activation can represent any continuous function on a compact domain, up to arbitrary precision. At first sight this seems to guarantee perfect learning. It does not. Representation power is necessary but not sufficient for good generalization.

Capacity to fit is different from capacity to generalize :

Making the network wider (or deeper) reduces bias but typically raises variance; after the interpolation threshold the variance term dominates, causing poor generalization even though the training error is zero (cf. Fig. 7.1 in The Elements of Statistical Learning). Hence "able to represent" is not the same as "able to predict well."

Bridging the gap: three key lever

(i) Regularization. Weight decay, early stopping, dropout and other penalties tame the variance term by restricting the effective hypothesis space.

(ii) Dataset size. More (or augmented) data decreases the variance of $\hat{f}$ by averaging out sampling noise.

(iii) Inductive structure. Architectures that embed prior knowledge (convolutions, weight sharing, equivariance, transformers with positional encodings) bias the search towards “reasonable” functions, lowering both the required sample complexity and the risk of over-fit.

These three ingredients determine whether the representational power promised by Cybenko’s theorem translates into practical predictive success

Generalization Bounds: PAC-Bayes and Algorithmic Stability

What we do :

PAC-Bayes framework.

We present the classical bound EQ[L] ⩽ EQ[Lˆ] + r KL(Q // P) + log(1/δ) 2n , which upper-bounds the test risk of a posterior Q in terms of its training risk, a complexity term (the KL-divergence to a prior P) and the sample size n.
This gives a rigorous, distribution-free guarantee on generalization.

Algorithmic stability.
We then show how small changes in the data translate into small changes in the predictor for specific learning algorithms.

Why it matters :

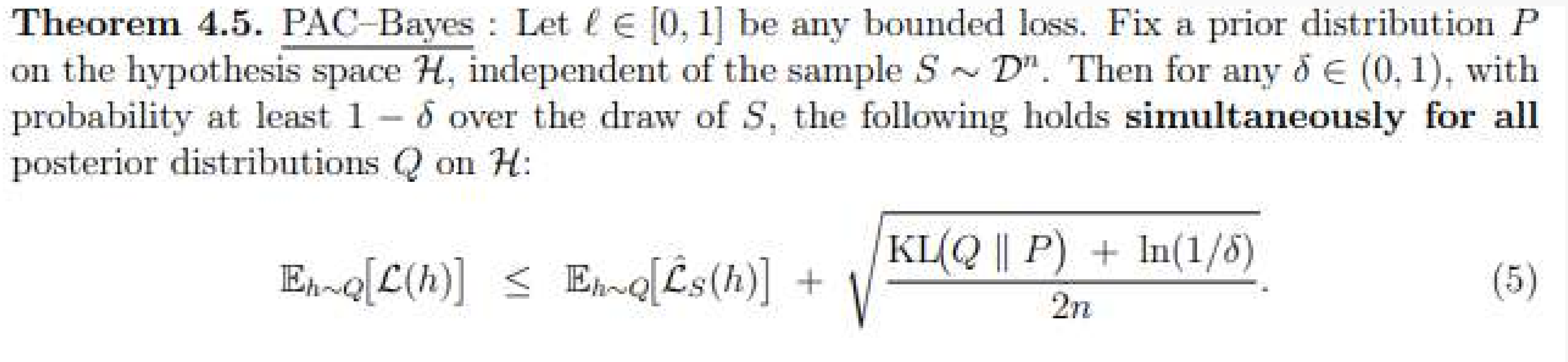
1. The PAC-Bayes bound turns the intuitive flat-minima and implicit regularization stories from Part 3 into quantitative statements: flatter solutions correspond to posteriors Q with smaller KL-divergence and hence tighter bounds.

2. Stability links optimizer dynamics to generalization directly. For example, the noise scale of SGD controls its stability constant and therefore the bound on excess test risk.

PAC-Bayes

Learning algorithms are usually evaluated by their empirical risk L^ S(h) on a sample S = {(xi , yi)} n i=1, while the quantity of real interest is the true (test) risk L(h) = E(x,y)   D[ℓ(h(x), y)], where ℓ is a bounded loss and D the unknown data distribution.

The PAC-Bayes framework provides a probabilistic bridge between the two : it upper-bounds the expected test risk of a randomised predictor by the corresponding expected training risk plus a complexity term involving the Kullback-Leibler divergence between the posterior Q and a data-independent prior P



Theorem 4.5. PAC–Bayes : Let $\ell \in [0,1]$ be any bounded loss. Fix a prior distribution P on the hypothesis space $\mathcal{H}$, independent of the sample $S \sim \mathcal{D}^n$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the draw of S, the following holds **simultaneously for all** posterior distributions Q on $\mathcal{H}$:

$$\mathbb{E}_{h\sim Q}[\mathcal{L}(h)] \;\le\; \mathbb{E}_{h\sim Q}[\hat{\mathcal{L}}_S(h)] + \sqrt{\frac{\mathrm{KL}(Q \,\|\, P) + \ln(1/\delta)}{2n}}. \tag{5}$$

Interpretation and Stability

Interpretation

Equationtells us that a posterior Q generalises well if both:
(i) its average training loss is small
(ii) it does not drift too far from the prior P, as quantified by the KL term.

Choosing P to encode domain knowledge and optimizing Q balances data fit and complexity.

Why stability ?

Intuitively, a learning algorithm that barely changes when one training example is modified must have distilled a signal that persists across samples; such an algorithm should therefore generalize well. This intuition is formalized by the notion of uniform stabilit

Double-descent

The double-descent curve, first highlighted by Belkin and systematically explored in deep network Nakkiran, refines the classical U-shaped bias-variance picture. When model capacity (e.g. number of parameters, width, depth) is gradually increased we observe three distinct regimes:

1. **Under-parameterised (classical)** — the test risk falls as capacity helps reduce bias; variance is still moderate.

2. **Interpolation peak** — at the interpolation threshold the model fits the training data exactly, variance explodes and the test risk spikes.

3. **Over-parameterised (second descent)** — surprisingly, pushing capacity beyond interpolation drives the test risk down again. Very large networks can generalise well despite having far more parameters than data points.

Two ideas

Benign over-fitting and norm-based complexity: although the network interpolates, optimisation (typically stochastic gradient descent) tends to choose solutions with small effective norm / flat minima, yielding low norm complexity and hence good generalisation.

Implicit bias of SGD: the anisotropic noise in SGD preferentially explores wide, flat valleys of the loss surface, acting as a form of data-dependent regularisation that counteracts the variance spike at interpolation

Empirical observation

Systematic experiments reported by Belkin and later extended to deep networks by Nakkiran reveal a striking double-descent pattern for the test risk as the model capacity C (e.g. width, number of parameters or polynomial degree) increases:

(i) For C < Cinterp (under-parameterised regime) the test error decreases—classical bias reduction.

(ii) At the interpolation threshold C = Cinterp the model fits the training set exactly; variance explodes and the test error peaks.

(iii) When C > Cinterp (highly over-parameterised regime) the test error decreases again, often surpassing the best performance achievable before interpolation. These observations are consistent across linear models with random features Belkin kernel machines, and modern deep architectures Nakkiran firmly establishing the double-descent phenomenon in practice.

Interpretation

The factor **n/d-n** is purely norm-controlled.
Despite having infinitely many solutions to $X\beta = y$, choosing the smallest-norm one (or letting SGD drift towards a flat minimum) yields a predictor whose capacity, measured through // βb // 2, decreases as d increases. Hence over-parametrisation may reduce—rather than inflate—the effective complexity, explaining the second descent.

Stochastic gradient descent bias towards flat minima:
Empirical studies show that stochastic gradient descent converges preferentially to wide minima corresponding to low-norm solutions in function space.
This implicit bias bridges the gap between mere interpolation and genuine generalisation in modern deep networks

V. Experimentation

Why we need experiments. These results paint a coherent theoretical picture, yet they rely on idealised assumptions (isotropic data, Gaussian noise, infinite width, etc.) that seldom hold in real-world deep learning. To understand when and how the above bounds are tight—or hopelessly loose—we must confront them with empirical evidence.

MNIST with a Yolo-CLS

- Fast to train on CPU/GPU
- Easy to reproduce

Our results support the view that implicit regularization (stochasticity, weight decay) can yield generalizable solutions even in highly over-parameterized models.

Experimentation and results

- The bias-variance decomposition was confirmed
- The PAC-Bayes KL term was higher than expected.
- The Hessian trace revealed flat minima
- The variance across seeds was low

VI. Conclusion

Answer to the problematic : Empirical findings and theoretical analyses converge on a unified insight: optimization algorithms influence not only the speed of convergence but also the region of the loss landscape where the model settles. Stochastic first-order methods implicitly encourage flatter minima, reducing generalization risk, while adaptive methods provide initial efficiency but require careful regularization. Natural gradient methods balance curvature awareness with computational feasibility, representing a promising direction for scalable optimization

Limitations :

- Experiments were limited to medium-scale datasets; larger benchmarks could reveal additional optimizer-model complexity interactions.
- The theoretical analyses assume smooth objectives; activations like ReLU introduce nonsmoothness, warranting specialized investigation.

Thank you