Scaling ResNets in the large-depth regime

ROUEN, AUGUST 2022

Gérard Biau

SORBONNE UNIVERSITÉ

Team

Adeline Fermanian
MINES PARIS - PSL

Pierre Marion
SORBONNE UNIVERSITY

Jean-Philippe Vert
GOOGLE RESEARCH

Agenda

Learning with ResNets

Scaling deep ResNets

Scaling in the continuous-time setting

Beyond initialization

Agenda

Learning with ResNets

Scaling deep ResNets

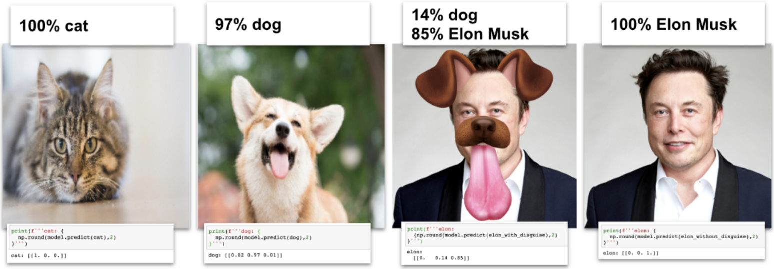
Scaling in the continuous-time setting

Beyond initialization

How most people see the supervised learning problem

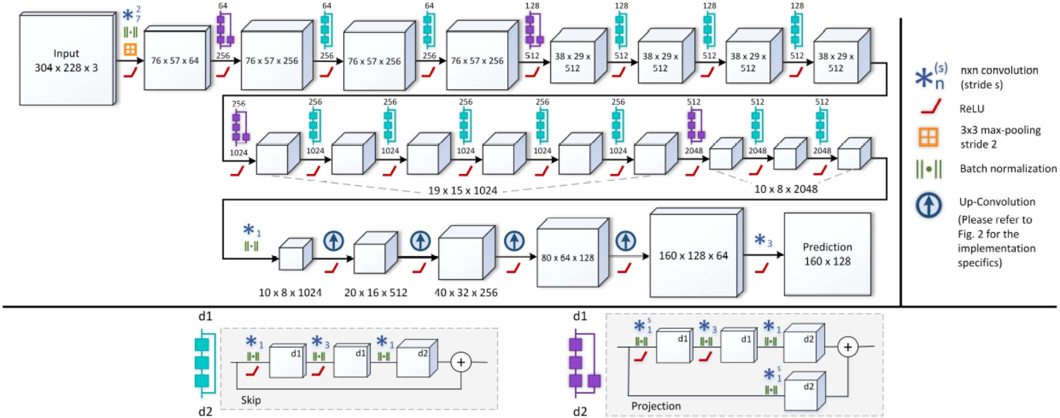
Learn how to build an image-recognizing convolutional neural network with Python and Keras in less than 15minutes!

Fabian Bosler Oct 5, 2019 · 10 min read ★



https://towardsdatascience.com/cat-dog-or-elon-musk-145658489730

How machine learners see the supervised learning problem



https://medium.datadriveninvestor.com/depth-estimation-with-deep-neural-networks-part-2-81ee374888eb

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.
- Regression: $\ell(F_\pi(x), y) = (y - F_\pi(x))^2$

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.
- Regression: $\ell(F_\pi(x), y) = (y - F_\pi(x))^2$ Binary classification: $\ell(F_\pi(x), y) = \mathbb{1}_{[yF_\pi(x) \leqslant 0]}$.

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.
- Regression: $\ell(F_\pi(x), y) = (y - F_\pi(x))^2$ Binary classification: $\ell(F_\pi(x), y) = \mathbb{1}_{[yF_\pi(x) \leqslant 0]}$.
- Theoretical risk minimization: choose

$$\pi^\star \in \operatorname*{argmin}_{\pi \in \Pi} \mathscr{L}(\pi) = \mathbb{E}(\ell(F_\pi(x), y)).$$

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.
- Regression: $\ell(F_\pi(x), y) = (y - F_\pi(x))^2$ Binary classification: $\ell(F_\pi(x), y) = \mathbb{1}_{[yF_\pi(x) \leqslant 0]}$.
- Theoretical risk minimization: choose

$$\pi^\star \in \operatorname*{argmin}_{\pi \in \Pi} \mathscr{L}(\pi) = \mathbb{E}(\ell(F_\pi(x), y)).$$

- Empirical risk minimization: choose

$$\pi_n \in \operatorname*{argmin}_{\pi \in \Pi} \mathscr{L}_n(\pi) = \frac{1}{n} \sum_{i=1}^{n} \ell(F_\pi(x_i), y_i).$$

How statisticians see the supervised learning problem

- Goal: understand the relationship between $x \in \mathbb{R}^{n_{\text{in}}}$ and $y \in \mathbb{R}^{n_{\text{out}}}$.
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^{n_{\text{in}}} \times \mathbb{R}^{n_{\text{out}}}$, i.i.d. $\sim (x, y)$.
- Model: $\{F_\pi : \mathbb{R}^{n_{\text{in}}} \mapsto \mathbb{R}^{n_{\text{out}}}, \pi \in \Pi\}$.
- Loss function $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$.
- Regression: $\ell(F_\pi(x), y) = (y - F_\pi(x))^2$ Binary classification: $\ell(F_\pi(x), y) = \mathbb{1}_{[yF_\pi(x) \leqslant 0]}$.
- Theoretical risk minimization: choose

$$\pi^\star \in \underset{\pi \in \Pi}{\text{argmin}}\ \mathscr{L}(\pi) = \mathbb{E}(\ell(F_\pi(x), y)).$$

- Empirical risk minimization: choose

$$\pi_n \in \underset{\pi \in \Pi}{\text{argmin}}\ \mathscr{L}_n(\pi) = \frac{1}{n} \sum_{i=1}^{n} \ell(F_\pi(x_i), y_i).$$

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

$$h_0 = Ax, \quad h_{k+1} = h_k + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

- Different forms for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ = different architectures.

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \dots, h_L \in \mathbb{R}^d$ defined by recurrence:

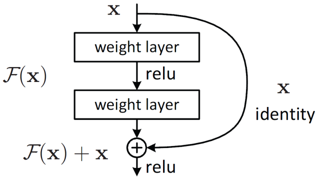
$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

- Different forms for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ = different architectures.

Original Parametric Simple General ResNet

$$f(h_k, \theta_{k+1}) = V_{k+1} \operatorname{ReLU}(W_{k+1} h_k + b_{k+1})$$

- ▷ $\operatorname{ReLU}(x) = \max(x, 0) =$ activation function
- ▷ $\theta_k = (W_k, b_k) =$ weight matrice $+$ bias
- ▷ $\pi = (A, B, (V_k)_{1 \leqslant k \leqslant L}, (\theta_k)_{1 \leqslant k \leqslant L})$



He et al. (2016)

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

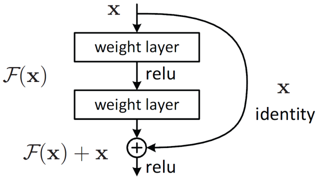
$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

- Different forms for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ = different architectures.

Original Parametric Simple General ResNet

$$f(h_k, \theta_{k+1}) = V_{k+1}\sigma(W_{k+1}h_k + b_{k+1})$$

- σ = activation function
- $\theta_k = (W_k, b_k)$ = weight matrice + bias
- $\pi = (A, B, (V_k)_{1\leqslant k\leqslant L}, (\theta_k)_{1\leqslant k\leqslant L})$



He et al. (2016)

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \ldots, h_L \in \mathbb{R}^d$ defined by recurrence:

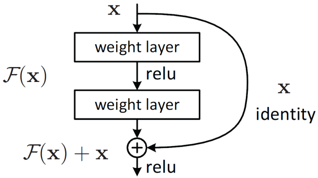
$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

- Different forms for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ = different architectures.

Original Parametric Simple General ResNet

$$f(h_k, \theta_{k+1}) = V_{k+1}\sigma(h_k)$$

- $\sigma =$ activation function
- $\theta_k = \emptyset$
- $\pi = (A, B, (V_k)_{1 \leqslant k \leqslant L})$



He et al. (2016)

Residual neural networks (ResNets)

- Sequence of hidden states $h_0, \dots, h_L \in \mathbb{R}^d$ defined by recurrence:

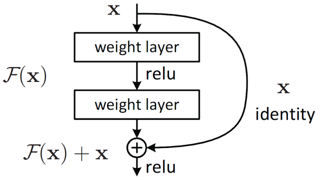
$$h_0 = Ax, \quad h_{k+1} = \boldsymbol{h_k} + f(h_k, \theta_{k+1}), \quad F_\pi(x) = Bh_L.$$

- Different forms for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ = different architectures.

Original Parametric Simple General ResNet

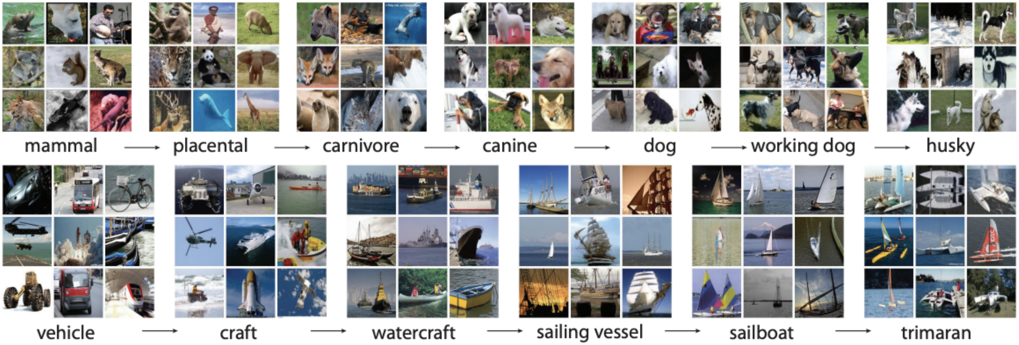
$$f(h_k, \theta_{k+1}) = V_{k+1} g(h_k, \theta_{k+1})$$

- ▷ $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$
- ▷ θ_k = parameters
- ▷ $\pi = (A, B, (V_k)_{1 \leqslant k \leqslant L}, (\theta_k)_{1 \leqslant k \leqslant L})$



He et al. (2016)

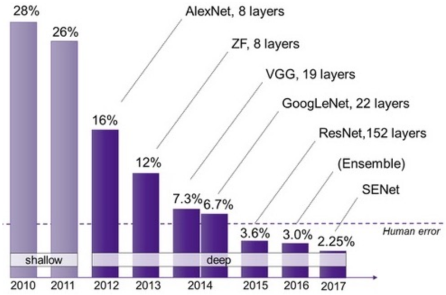
The revolution of ResNets



Examples from the ImageNet dataset

https://blog.roboflow.com/introduction-to-imagenet

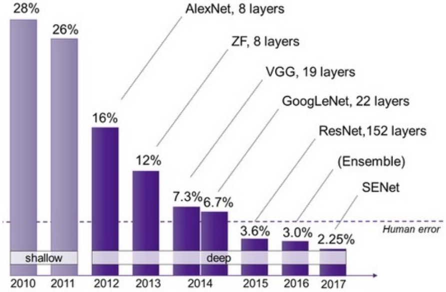
The revolution of ResNets



ImageNet performance over time

https://semiengineering.com/new-vision-technologies-for-real-world-applications

The revolution of ResNets



ImageNet performance over time

https://semiengineering.com/new-vision-technologies-for-real-world-applications

Deep learning $\rightarrow$ neural ODE $\leftarrow$ ODE

- Traditional neural networks

$$h_{k+1} = f(h_k, \theta_{k+1})$$

Deep learning $\rightarrow$ neural ODE $\leftarrow$ ODE

- Traditional neural networks

$$h_{k+1} = f(h_k, \theta_{k+1})$$

- Residual neural networks (He et al., 2016)

$$h_{k+1} = \boldsymbol{h_k} + \quad f(h_k, \theta_{k+1})$$

Deep learning $\rightarrow$ neural ODE $\leftarrow$ ODE

- Traditional neural networks

$$h_{k+1} = f(h_k, \theta_{k+1})$$

- Residual neural networks (He et al., 2016)

$$h_{k+1} = \boldsymbol{h_k} + \frac{1}{L} f(h_k, \theta_{k+1})$$

Deep learning $\rightarrow$ neural ODE $\leftarrow$ ODE

- Traditional neural networks

$$h_{k+1} = f(h_k, \theta_{k+1})$$

- Residual neural networks (He et al., 2016)

$$h_{k+1} = \boldsymbol{h_k} + \frac{1}{L} f(h_k, \theta_{k+1})$$

- Neural ODE (Chen et al., 2018)

$$dH_t = f(H_t, \Theta_t)dt$$

Deep learning $\rightarrow$ neural ODE $\leftarrow$ ODE

- Traditional neural networks

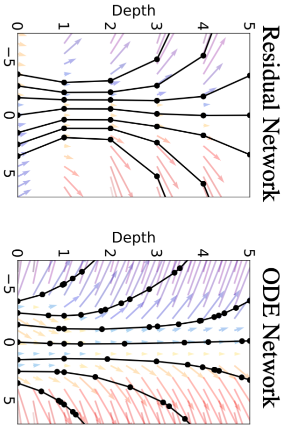
$$h_{k+1} = f(h_k, \theta_{k+1})$$

- Residual neural networks (He et al., 2016)

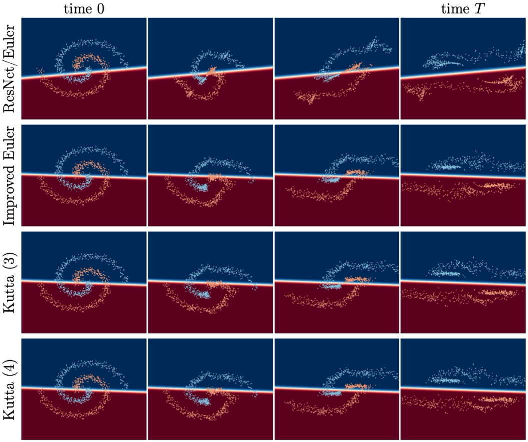
$$h_{k+1} = \boldsymbol{h_k} + \frac{1}{L} f(h_k, \theta_{k+1})$$

- Neural ODE (Chen et al., 2018)

$$dH_t = f(H_t, \Theta_t)dt$$

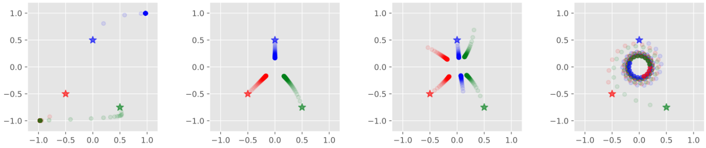


New network architectures: Runge-Kutta networks



Benning et al. (2019)

New network architectures: antisymmetric networks

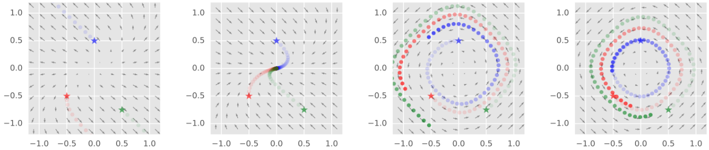


(a) Vanilla RNN with a random weight matrix.

(b) Vanilla RNN with an identity weight matrix.

(c) Vanilla RNN with a random orthogonal weight matrix (seed = 0).

(d) Vanilla RNN with a random orthogonal weight matrix (seed = 1).



(e) RNN with feedback with positive eigenvalues.

(f) RNN with feedback with negative eigenvalues.

(g) RNN with feedback with imaginary eigenvalues.

(h) RNN with feedback with imaginary eigenvalues and diffusion.

Chang et al. (2019)

In summary

ResNet	Neural ODE
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{L} f(h_k, \theta_{k+1})$	$dH_t = f(H_t, \Theta_t)dt$
$F_\pi(x) = Bh_T$	$F_\Pi(x) = BH_1$

$$f(h, \theta) = V\sigma(Wh + b)$$

In summary

ResNet	**Neural ODE**
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{L} f(h_k, \theta_{k+1})$	$dH_t = f(H_t, \Theta_t)\,dt$
$F_\pi(x) = Bh_T$	$F_\Pi(x) = BH_1$

$$f(h, \theta) = V\sigma(Wh + b)$$

⚠ ResNet $\neq$ RNN

**Framing RNN as a kernel method:
A neural ODE approach**

Adeline Fermanian[1*] Pierre Marion[1*] Jean-Philippe Vert[2] Gérard Biau[1]
[1] Sorbonne Université, CNRS,
Laboratoire de Probabilités, Statistique et Modélisation, LPSM,
F-75005 Paris, France
{adeline.fermanian, pierre.marion, gerard.biau}@sorbonne-universite.fr
[2] Google Research, Brain team,
Paris, France
jpvert@google.com

Abstract

Building on the interpretation of a recurrent neural network (RNN) as a continuous-time neural differential equation, we show, under appropriate conditions, that the solution of a RNN can be viewed as a linear function of a specific feature set of the input sequence, known as the signature. This connection allows us to frame a RNN as a kernel method in a suitable reproducing kernel Hilbert space. As a consequence, we obtain theoretical guarantees on generalization and stability for a large class of recurrent networks. Our results are illustrated on simulated datasets.

1 Introduction

Recurrent neural networks (RNN) are among the most successful methods for modeling sequential data. They have achieved state-of-the-art results in difficult problems such as natural language processing (e.g., Mikolov et al., 2010; Collobert et al., 2011) or speech recognition (e.g., Hinton et al., 2012; Graves et al., 2013). This class of neural networks has a natural interpretation in terms of (discretization of) ordinary differential equations (ODE), which casts them in the field of neural ODE (Chen et al., 2018). This observation has led to the development of continuous-depth models for handling irregularly-sampled time-series data, including the ODE-RNN model (Rubanova et al., 2019), GRU-ODE-Bayes (De Brouwer et al., 2019), or neural CDE models (Kidger et al., 2020; Morrill et al., 2020a). In addition, the time-continuous interpretation of RNN allows to leverage the rich theory of differential equations to develop new recurrent architectures (Chang et al., 2019; Herrera et al., 2020; Erichson et al., 2021), which are better at learning long-term dependencies.

On the other hand, the development of kernel methods for deep learning offers theoretical insights on the functions learned by the networks (Cho and Saul, 2009; Belkin et al., 2018; Jacot et al., 2018). Here, the general principle consists in defining a reproducing kernel Hilbert space (RKHS)—that is, a function class $\mathscr{H}$—, which is rich enough to describe the architectures of networks. A good example is the construction of Bietti and Mairal (2017, 2019), who exhibit an RKHS for convolutional neural networks. This kernel perspective has several advantages. First, by separating the representation of the data from the learning process, it allows to study invariances of the representations learned by the network. Next, by reducing the learning problem to a linear one in $\mathscr{H}$, generalization bounds can be more easily obtained. Finally, the Hilbert structure of $\mathscr{H}$ provides a natural metric on neural networks, which can be used for example for regularization (Bietti et al., 2019).

*Equal contribution

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Agenda

Learning with ResNets

Scaling deep ResNets

Scaling in the continuous-time setting

Beyond initialization

Stability at initialization

- Original ResNet:

$$
\begin{aligned}
h_0 &= Ax \\
h_{k+1} &= h_k + V_{k+1}\,\mathrm{ReLU}(W_{k+1} h_k) \\
F_\pi(x) &= B h_L.
\end{aligned}
$$

Stability at initialization

- Original ResNet:

$$\begin{aligned} h_0 &= Ax \\ h_{k+1} &= h_k + V_{k+1} \operatorname{ReLU}(W_{k+1} h_k) \\ F_\pi(x) &= B h_L. \end{aligned}$$

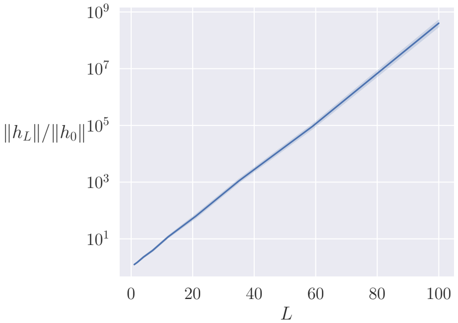
- At initialization: A, B, $(V_k)_{1 \leqslant k \leqslant L}$, and $(W_k)_{1 \leqslant k \leqslant L}$ are i.i.d. Gaussian matrices.

Stability at initialization

- Original ResNet:

$$\begin{aligned} h_0 &= Ax \\ h_{k+1} &= h_k + V_{k+1}\,\mathrm{ReLU}(W_{k+1}h_k) \\ F_\pi(x) &= Bh_L. \end{aligned}$$

- At initialization: A, B, $(V_k)_{1 \leqslant k \leqslant L}$, and $(W_k)_{1 \leqslant k \leqslant L}$ are i.i.d. Gaussian matrices.

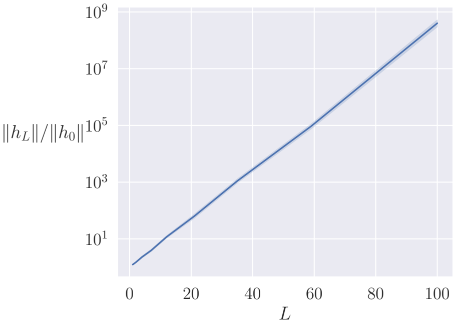


Stability at initialization

- Original ResNet:

$$h_0 = Ax$$
$$h_{k+1} = h_k + V_{k+1}\,\mathrm{ReLU}(W_{k+1} h_k)$$
$$F_\pi(x) = B h_L.$$

- At initialization: A, B, $(V_k)_{1 \leqslant k \leqslant L}$, and $(W_k)_{1 \leqslant k \leqslant L}$ are i.i.d. Gaussian matrices.



☞ Solution: batch normalization or scaling.

Scaling ResNets

- A scaling factor ${}^{1}\!/\!{}_{L^\beta}$:

$$h_{k+1} = h_k + \frac{1}{L^\beta} V_{k+1} \,\mathrm{ReLU}(W_{k+1} h_k).$$

Scaling ResNets

- A scaling factor $^1/_{L^\beta}$:

$$h_{k+1} = h_k + \frac{1}{L^\beta} V_{k+1} \, \mathrm{ReLU}(W_{k+1} h_k).$$

- Question: choice of β.

Scaling ResNets

- A scaling factor ${}^{1}/_{L^{\beta}}$:

$$h_{k+1} = h_k + \frac{1}{L^{\beta}} V_{k+1} \,\mathrm{ReLU}(W_{k+1} h_k).$$

- Question: choice of β.
- $\beta = 0$ (original ResNets)?

Scaling ResNets

- A scaling factor ${}^{1}\!/\!{}_{L^\beta}$:

$$h_{k+1} = h_k + \frac{1}{L^\beta} V_{k+1} \,\mathrm{ReLU}(W_{k+1} h_k).$$

- Question: choice of β.
- $\beta = 0$ (original ResNets)? $\beta = 1$ (neural ODE)?

Scaling ResNets

- A scaling factor $1/L^\beta$:

$$h_{k+1} = h_k + \frac{1}{L^\beta} V_{k+1} \operatorname{ReLU}(W_{k+1} h_k).$$

- Question: choice of β.
- $\beta = 0$ (original ResNets)? $\beta = 1$ (neural ODE)?
- Many empirical studies, no consensus.

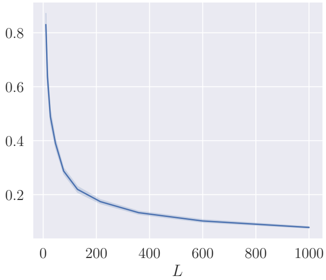
Scaling ResNets

- A scaling factor $1/L^\beta$:

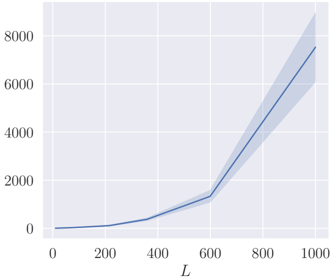
$$h_{k+1} = h_k + \frac{1}{L^\beta} V_{k+1} \,\mathrm{ReLU}(W_{k+1} h_k).$$

- Question: choice of β.
- $\beta = 0$ (original ResNets)? $\beta = 1$ (neural ODE)?
- Many empirical studies, no consensus.
- Our approach: mathematical analysis at initialization.

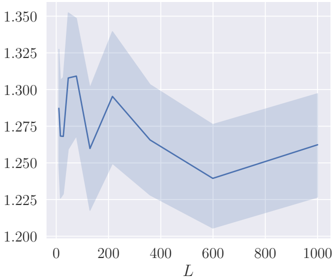
Scaling with standard initialization



(a) $\|h_L - h_0\|/\|h_0\|$, $\beta = 1$

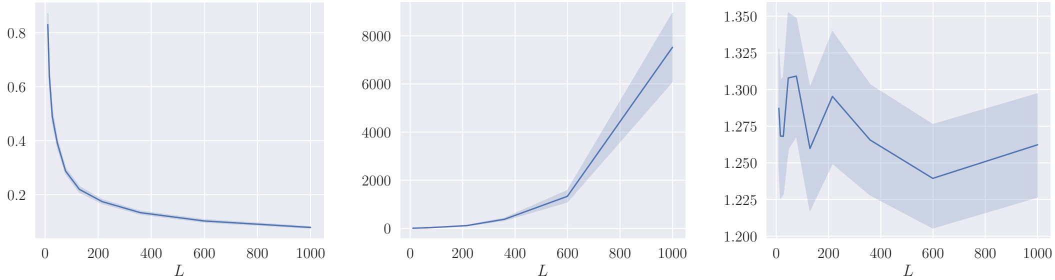


(b) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.25$



(c) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.5$

Scaling with standard initialization



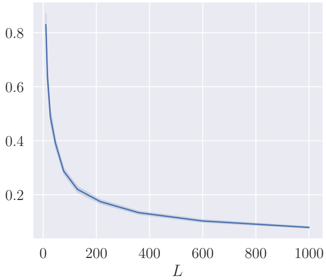
(a) $\|h_L - h_0\| / \|h_0\|$, $\beta = 1$

(b) $\|h_L - h_0\| / \|h_0\|$, $\beta = 0.25$

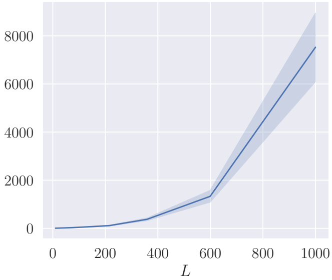
(c) $\|h_L - h_0\| / \|h_0\|$, $\beta = 0.5$

- With an i.i.d. initialization, the critical value for scaling is $\beta = {}^1\!/_2$.

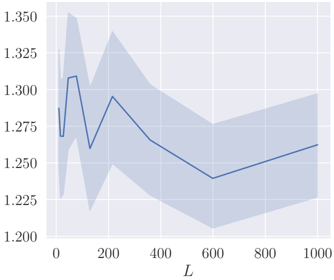
Scaling with standard initialization



(a) $\|h_L - h_0\|/\|h_0\|$, $\beta = 1$



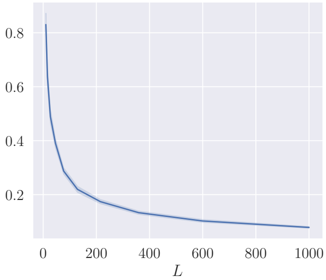
(b) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.25$



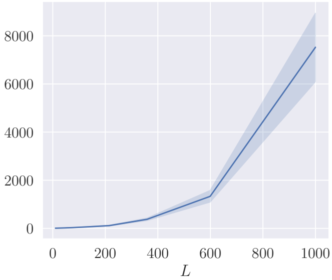
(c) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.5$

- With an i.i.d. initialization, the critical value for scaling is $\beta = {}^1\!/_2$.
- Similar results (identity/explosion/stability) for the gradients.

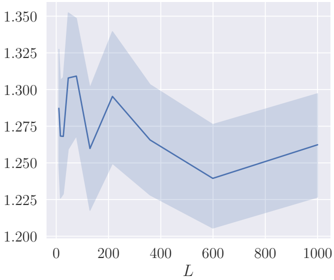
Scaling with standard initialization



(a) $\|h_L - h_0\|/\|h_0\|$, $\beta = 1$



(b) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.25$



(c) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.5$

- With an i.i.d. initialization, the critical value for scaling is $\beta = 1/2$.
- Similar results (identity/explosion/stability) for the gradients.
- Not the ODE scaling! 🤔

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$
2. If $\beta < 1/2$
3. If $\beta = 1/2$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$
2. If $\beta < 1/2$
3. If $\beta = 1/2$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\| / \|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0$. $\rightarrow$ identity
2. If $\beta < 1/2$
3. If $\beta = 1/2$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0$. $\rightarrow$ identity
2. If $\beta < 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} \infty$.
3. If $\beta = 1/2$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\| / \|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < 1/2$ then $\|h_L - h_0\| / \|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\rightarrow$ explosion
3. If $\beta = 1/2$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\rightarrow$ explosion
3. If $\beta = 1/2$ then, with probability at least $1-\delta$,

$$\exp\left(\frac{3}{8} - \sqrt{\frac{22}{d\delta}}\right) - 1 < \frac{\|h_L - h_0\|^2}{\|h_0\|^2} < \exp\left(1 + \sqrt{\frac{10}{d\delta}}\right) + 1.$$

Scaling with standard initialization

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < 1/2$ then $\|h_L - h_0\|/\|h_0\| \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\rightarrow$ explosion
3. If $\beta = 1/2$ then, with probability at least $1 - \delta$,

$$\exp\left(\frac{3}{8} - \sqrt{\frac{22}{d\delta}}\right) - 1 < \frac{\|h_L - h_0\|^2}{\|h_0\|^2} < \exp\left(1 + \sqrt{\frac{10}{d\delta}}\right) + 1.$$

$\rightarrow$ stability

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\| / \|p_L\|$ when L is large.

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\|/\|p_L\|$ when L is large.
- Backpropagation formula:

$$p_k = p_{k+1} + \frac{1}{L^\beta} \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1}$$

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\| / \|p_L\|$ when L is large.
- Backpropagation formula:

$$p_k = p_{k+1} + \frac{1}{L^\beta} \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \qquad \to \text{wrong way.}$$

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\| / \|p_L\|$ when L is large.
- Backpropagation formula:

$$p_k = p_{k+1} + \frac{1}{L^\beta} \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \quad \to \text{wrong way.}$$

- Our approach: with $q_k(z) = \frac{\partial h_k}{\partial h_0} z$,

$$q_{k+1}(z) = q_k(z) + \frac{1}{L^\beta} V_{k+1} \frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k(z)$$

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\| / \|p_L\|$ when L is large.
- Backpropagation formula:

$$p_k = p_{k+1} + \frac{1}{L^\beta} \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \qquad \to \text{wrong way.}$$

- Our approach: with $q_k(z) = \frac{\partial h_k}{\partial h_0} z$,

$$q_{k+1}(z) = q_k(z) + \frac{1}{L^\beta} V_{k+1} \frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k(z) \qquad \to \text{flow of information} = \checkmark.$$

Gradients

- Objective: assess the backwards dynamics of the gradients $p_k = \frac{\partial \mathscr{L}_n}{\partial h_k} \in \mathbb{R}^d$.
- Target: $\|p_0 - p_L\| / \|p_L\|$ when L is large.
- Backpropagation formula:

$$p_k = p_{k+1} + \frac{1}{L^\beta} \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \qquad \to \text{wrong way.}$$

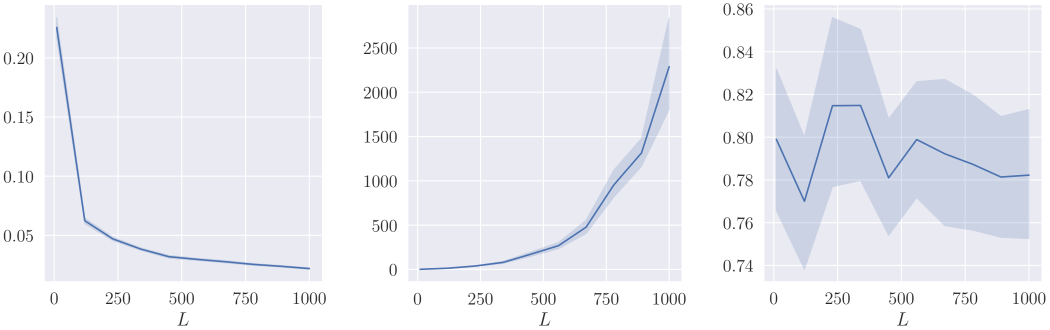
- Our approach: with $q_k(z) = \frac{\partial h_k}{\partial h_0} z$,

$$q_{k+1}(z) = q_k(z) + \frac{1}{L^\beta} V_{k+1} \frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k(z) \qquad \to \text{flow of information} = \checkmark.$$

- Conclusion with

$$\frac{\|p_0\|^2}{\|p_L\|^2} = \mathbb{E}_{z \sim \mathcal{N}(0, I_d)} \left(\left| \left(\frac{p_L}{\|p_L\|} \right)^\top q_L(z) \right|^2 \right).$$

Scaling with standard initialization – Gradients



(a) $\|p_0 - p_L\| / \|p_L\|$, $\beta = 1$ (b) $\|p_0 - p_L\| / \|p_L\|$, $\beta = 0.25$ (c) $\|p_0 - p_L\| / \|p_L\|$, $\beta = 0.5$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > {}^1\!/_2$
2. If $\beta < {}^1\!/_2$
3. If $\beta = {}^1\!/_2$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|p_0 - p_L\|/\|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$
2. If $\beta < 1/2$
3. If $\beta = 1/2$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > {}^1\!/\!{}_2$ then $\|p_0 - p_L\| / \|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < {}^1\!/\!{}_2$
3. If $\beta = {}^1\!/\!{}_2$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\, V_k$ and $\sqrt{d}\, W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|p_0 - p_L\| / \|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < 1/2$ then $\mathbb{E}(\|p_0 - p_L\| / \|p_L\|) \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$
3. If $\beta = 1/2$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|p_0 - p_L\|/\|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\rightarrow$ identity
2. If $\beta < 1/2$ then $\mathbb{E}(\|p_0 - p_L\|/\|p_L\|) \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\rightarrow$ explosion
3. If $\beta = 1/2$

Scaling with standard initialization – Gradients

Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

1. If $\beta > 1/2$ then $\|p_0 - p_L\|/\|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\quad\rightarrow$ identity
2. If $\beta < 1/2$ then $\mathbb{E}(\|p_0 - p_L\|/\|p_L\|) \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\quad\rightarrow$ explosion
3. If $\beta = 1/2$ then

$$\exp\left(\frac{1}{2}\right) - 1 \leqslant \mathbb{E}\left(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\right) \leqslant \exp(4) - 1.$$

Scaling with standard initialization – Gradients

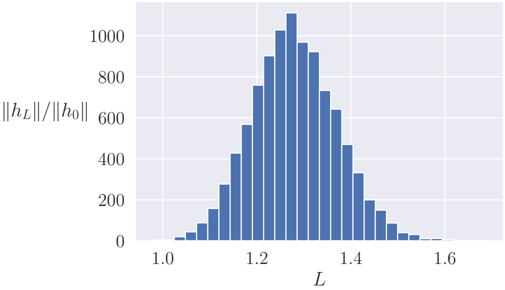
Theorem

Assumption: the entries of $\sqrt{d}\,V_k$ and $\sqrt{d}\,W_k$ are symmetric i.i.d. sub-Gaussian.

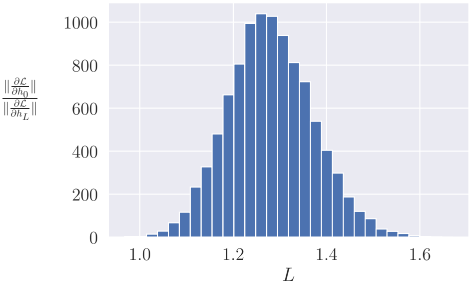
1. If $\beta > 1/2$ then $\|p_0 - p_L\| / \|p_L\| \xrightarrow[L\to\infty]{\mathbb{P}} 0.$ $\to$ identity
2. If $\beta < 1/2$ then $\mathbb{E}(\|p_0 - p_L\| / \|p_L\|) \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$ $\to$ explosion
3. If $\beta = 1/2$ then

$$\exp\left(\frac{1}{2}\right) - 1 \leqslant \mathbb{E}\left(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\right) \leqslant \exp(4) - 1.$$ $\to$ stability

Stability – output/gradients



(a) Distribution of $\|h_L\|/\|h_0\|$



(b) Distribution of $\|\frac{\partial \mathscr{L}_n}{\partial h_0}\|/\|\frac{\partial \mathscr{L}_n}{\partial h_L}\|$

How to interpret the critical value $\beta = 1/2$?

- Simple ResNet: $h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$.

How to interpret the critical value $\beta = 1/2$?

- Simple ResNet: $h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$.
- The entries of V_k are i.i.d. $\mathcal{N}(0, 1/d)$.

How to interpret the critical value $\beta = 1/2$?

- Simple ResNet: $h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$.
- The entries of V_k are i.i.d. $\mathcal{N}(0, 1/d)$.
- For $\mathbf{B} : [0,1] \to \mathbb{R}^{d \times d}$ a $(d \times d)$-dimensional Brownian motion

How to interpret the critical value $\beta = 1/2$?

- Simple ResNet: $h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$.
- The entries of V_k are i.i.d. $\mathcal{N}(0, 1/d)$.
- For $\mathbf{B} : [0,1] \to \mathbb{R}^{d \times d}$ a $(d \times d)$-dimensional Brownian motion

$$\mathbf{B}_{(k+1)/L,i,j} - \mathbf{B}_{k/L,i,j} \sim \mathcal{N}\Big(0, \frac{1}{L}\Big).$$

How to interpret the critical value $\beta = {}^1\!/_2$?

- Simple ResNet: $h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$.
- The entries of V_k are i.i.d. $\mathcal{N}(0, {}^1\!/_d)$.
- For $\mathbf{B} : [0,1] \to \mathbb{R}^{d \times d}$ a $(d \times d)$-dimensional Brownian motion

$$\mathbf{B}_{(k+1)/L,i,j} - \mathbf{B}_{k/L,i,j} \sim \mathcal{N}\Big(0, \frac{1}{L}\Big).$$

- Consequence:

$$h_0 = Ax, \quad h_{k+1}^\top = h_k^\top + \frac{1}{\sqrt{d}} \sigma(h_k^\top)(\mathbf{B}_{(k+1)/L} - \mathbf{B}_{k/L}), \quad 0 \leqslant k \leqslant L-1.$$

SDE regime

ResNet	**Neural SDE**
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k)$	$dH_t^\top = \frac{1}{\sqrt{d}} \sigma(H_t^\top) dB_t$
$F_\pi(x) = Bh_L$	$F_\Pi(x) = BH_1$

SDE regime

ResNet	**Neural SDE**
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1}\sigma(h_k)$	$dH_t^\top = \frac{1}{\sqrt{d}}\sigma(H_t^\top)dB_t$
$F_\pi(x) = Bh_L$	$F_\Pi(x) = BH_1$

Proposition

Assumption: the entries of V_k are i.i.d. Gaussian $\mathcal{N}(0, {}^1\!/_d)$ and σ is Lipschitz continuous.

SDE regime

ResNet	**Neural SDE**
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1}\sigma(h_k)$	$dH_t^\top = \frac{1}{\sqrt{d}}\sigma(H_t^\top)dB_t$
$F_\pi(x) = Bh_L$	$F_\Pi(x) = BH_1$

Proposition

Assumption: the entries of V_k are i.i.d. Gaussian $\mathcal{N}(0, 1/d)$ and σ is Lipschitz continuous.

Then the SDE has a unique solution H and, for any $0 \leqslant k \leqslant L$,

$$\mathbb{E}\big(\|H_{k/L} - h_k\|\big) \leqslant \frac{C}{\sqrt{L}}.$$

Summary so far

For deep ResNets with i.i.d. initialization:

Summary so far

For deep ResNets with i.i.d. initialization:

- the critical value for scaling is $\beta = {}^{1}\!/_{2}$
- this value corresponds in the deep limit to a SDE.

Summary so far

For deep ResNets with i.i.d. initialization:

- the critical value for scaling is $\beta = {}^1\!/_2$
- this value corresponds in the deep limit to a SDE.

Remaining questions:

- Can we obtain other limits? For example ODEs?
- Do they correspond to the same critical value?

Summary so far

For deep ResNets with i.i.d. initialization:

- the critical value for scaling is $\beta = 1/2$
- this value corresponds in the deep limit to a SDE.

Remaining questions:

- Can we obtain other limits? For example ODEs?
- Do they correspond to the same critical value?

Key: link between β and the weight distributions.

Summary so far

For deep ResNets with i.i.d. initialization:

- the critical value for scaling is $\beta = {}^1\!/_2$
- this value corresponds in the deep limit to a SDE.

Remaining questions:

- Can we obtain other limits? For example ODEs?
- Do they correspond to the same critical value?

Key: link between β and the weight distributions.

Agenda

Learning with ResNets

Scaling deep ResNets

Scaling in the continuous-time setting

Beyond initialization

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1\leqslant k\leqslant L}$ and $(\theta_k)_{1\leqslant k\leqslant L}$ are discretizations of smooth functions.

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1\leqslant k\leqslant L}$ and $(\theta_k)_{1\leqslant k\leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1\leqslant k\leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d\times d}$

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1 \leqslant k \leqslant L}$ and $(\theta_k)_{1 \leqslant k \leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1 \leqslant k \leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d \times d}$ $\quad (\theta_k)_{1 \leqslant k \leqslant L} \hookrightarrow \Theta : [0,1] \to \mathbb{R}^p.$

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1 \leqslant k \leqslant L}$ and $(\theta_k)_{1 \leqslant k \leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1 \leqslant k \leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d \times d}$ $\quad (\theta_k)_{1 \leqslant k \leqslant L} \hookrightarrow \Theta : [0,1] \to \mathbb{R}^p$.
- Model:

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1,$$

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1\leqslant k\leqslant L}$ and $(\theta_k)_{1\leqslant k\leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1\leqslant k\leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d\times d}$ $\quad (\theta_k)_{1\leqslant k\leqslant L} \hookrightarrow \Theta : [0,1] \to \mathbb{R}^p$.
- Model:

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1,$$

where $V_k = \mathscr{V}_{k/L}$ and $\theta_k = \Theta_{k/L}$.

Leaving the i.i.d. world behind

- Idea: the weights $(V_k)_{1\leqslant k\leqslant L}$ and $(\theta_k)_{1\leqslant k\leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1\leqslant k\leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d\times d}$ $\quad (\theta_k)_{1\leqslant k\leqslant L} \hookrightarrow \Theta : [0,1] \to \mathbb{R}^p$.
- Model:

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1,$$

where $V_k = \mathscr{V}_{k/L}$ and $\theta_k = \Theta_{k/L}$.

Assumption: the stochastic processes $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

Leaving the i.i.d. world behind

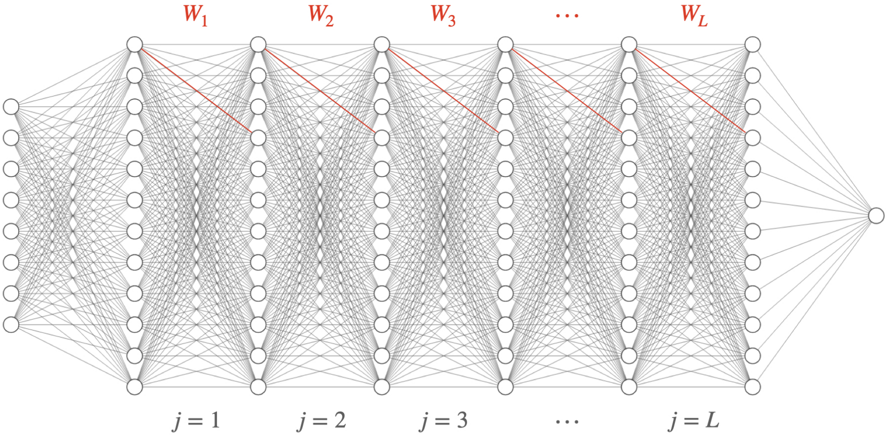
- Idea: the weights $(V_k)_{1 \leqslant k \leqslant L}$ and $(\theta_k)_{1 \leqslant k \leqslant L}$ are discretizations of smooth functions.
- $(V_k)_{1 \leqslant k \leqslant L} \hookrightarrow \mathscr{V} : [0,1] \to \mathbb{R}^{d \times d} \quad (\theta_k)_{1 \leqslant k \leqslant L} \hookrightarrow \Theta : [0,1] \to \mathbb{R}^p$.
- Model:

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1,$$

where $V_k = \mathscr{V}_{k/L}$ and $\theta_k = \Theta_{k/L}$.

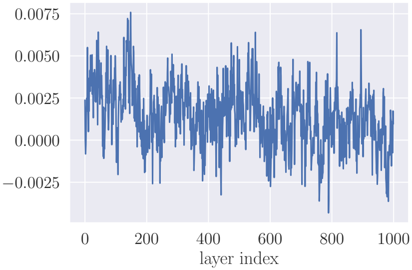
Assumption: the stochastic processes $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

- Example: the entries of $\mathscr{V}$ and Θ are independent Gaussian processes with zero expectation and covariance $K(x, x') = \exp(-\frac{(x-x')^2}{2\ell^2})$.



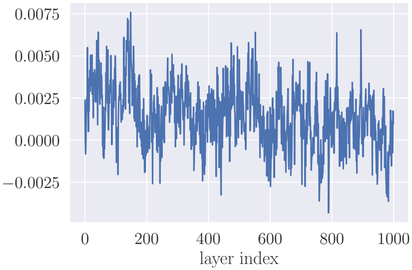
W_1
W_2
W_3
$\cdots$
W_L
$j = 1$
$j = 2$
$j = 3$
$\cdots$
$j = L$

Scaling and weight regularity

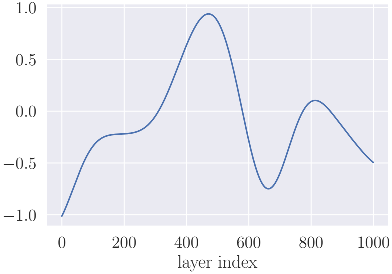


(a) I.i.d.

Scaling and weight regularity



(a) I.i.d.



(b) Smooth

ODE regime

ResNet	Neural ODE
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1})$	$dH_t = \mathscr{V}_t g(H_t, \Theta_t) dt$
$F_\pi(x) = B h_L$	$F_\Pi(x) = B H_1$

ODE regime

ResNet	**Neural ODE**
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1})$	$dH_t = \mathscr{V}_t g(H_t, \Theta_t) dt$
$F_\pi(x) = B h_L$	$F_\Pi(x) = B H_1$

Proposition

Assumption: the function g is Lipschitz continuous on compact sets.

ODE regime

ResNet	Neural ODE
$h_0 = Ax$	$H_0 = Ax$
$h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1})$	$dH_t = \mathscr{V}_t g(H_t, \Theta_t) dt$
$F_\pi(x) = Bh_L$	$F_\Pi(x) = BH_1$

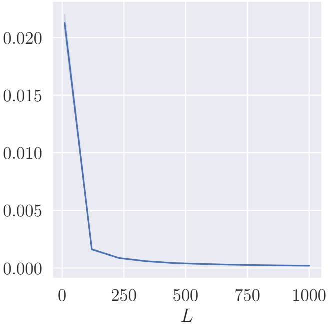
Proposition

Assumption: the function g is Lipschitz continuous on compact sets.

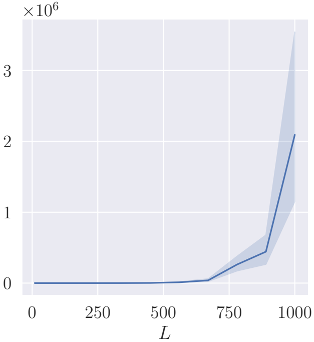
Then the ODE has a unique solution H and, a.s., for any $0 \leqslant k \leqslant L$,

$$\|H_{k/L} - h_k\| \leqslant \frac{c}{L}.$$

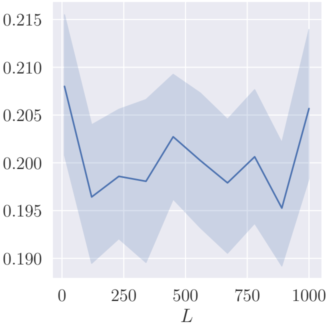
Scaling with a smooth initialization



(a) $\|h_L - h_0\|/\|h_0\|$, $\beta = 2$

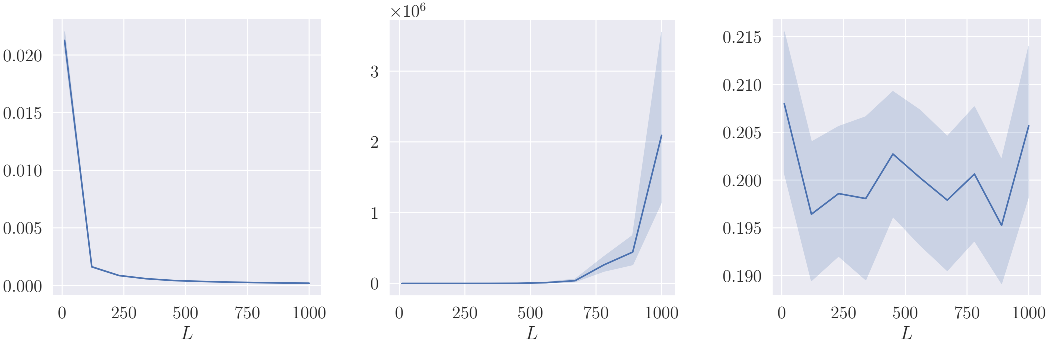


(b) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.5$



(c) $\|h_L - h_0\|/\|h_0\|$, $\beta = 1$

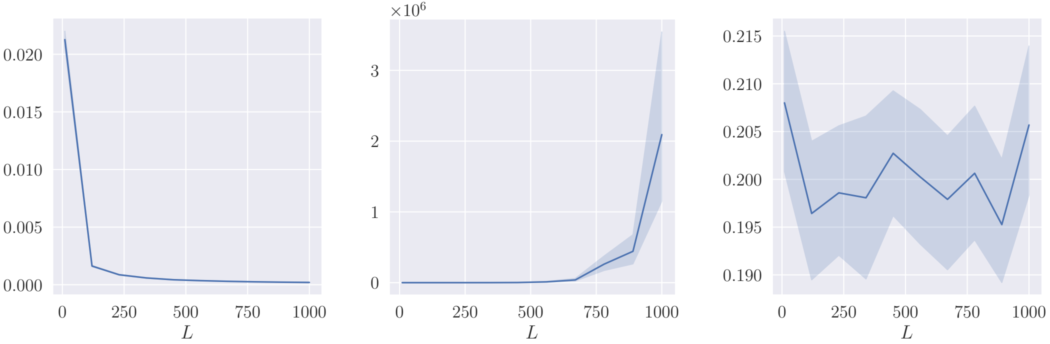
Scaling with a smooth initialization



(a) $\|h_L - h_0\| / \|h_0\|$, $\beta = 2$ (b) $\|h_L - h_0\| / \|h_0\|$, $\beta = 0.5$ (c) $\|h_L - h_0\| / \|h_0\|$, $\beta = 1$

- Again 3 cases: identity/explosion/stability.

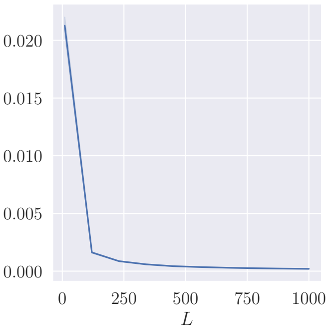
Scaling with a smooth initialization



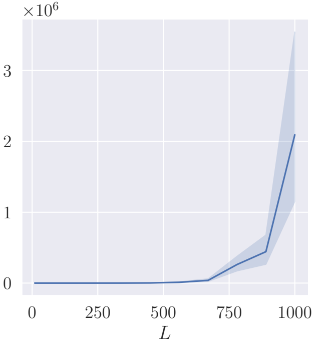
(a) $\|h_L - h_0\|/\|h_0\|$, $\beta = 2$ (b) $\|h_L - h_0\|/\|h_0\|$, $\beta = 0.5$ (c) $\|h_L - h_0\|/\|h_0\|$, $\beta = 1$

- Again 3 cases: identity/explosion/stability.
- With a smooth initialization, the critical scaling is $\beta = 1$.

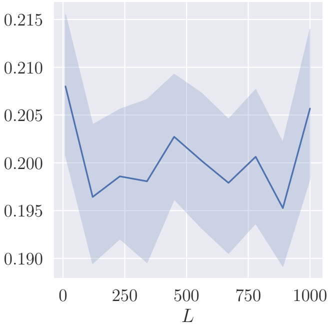
Scaling with a smooth initialization



(a) $\|h_L - h_0\| / \|h_0\|$, $\beta = 2$



(b) $\|h_L - h_0\| / \|h_0\|$, $\beta = 0.5$



(c) $\|h_L - h_0\| / \|h_0\|$, $\beta = 1$

- Again 3 cases: identity/explosion/stability.
- With a smooth initialization, the critical scaling is $\beta = 1$.
- It is the scaling that corresponds in the deep limit to an ODE.

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$
2. If $\beta = 1$
3. If $\beta < 1$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \xrightarrow{L\to\infty} 0.$
2. If $\beta = 1$
3. If $\beta < 1$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\|/\|h_0\| \xrightarrow{L\to\infty} 0.$ $\to$ identity
2. If $\beta = 1$
3. If $\beta < 1$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \xrightarrow{L\to\infty} 0.$ $\rightarrow$ identity
2. If $\beta = 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \leqslant c.$
3. If $\beta < 1$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \xrightarrow{L \to \infty} 0.$ $\rightarrow$ identity
2. If $\beta = 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \leqslant c.$ $\rightarrow$ stability
3. If $\beta < 1$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \xrightarrow{L \to \infty} 0.$ $\rightarrow$ identity
2. If $\beta = 1$ then, a.s., $\|h_L - h_0\| / \|h_0\| \leqslant c.$ $\rightarrow$ stability
3. If $\beta < 1$ + assumptions, then $\max_k \dfrac{\|h_k - h_0\|}{\|h_0\|} \xrightarrow{L \to \infty} \infty.$

Scaling with with a smooth initialization

Theorem

Assumption: $\mathscr{V}$ and Θ are a.s. Lipschitz continuous and bounded.

1. If $\beta > 1$ then, a.s., $\|h_L - h_0\|/\|h_0\| \xrightarrow{L\to\infty} 0.$ $\rightarrow$ identity
2. If $\beta = 1$ then, a.s., $\|h_L - h_0\|/\|h_0\| \leqslant c.$ $\rightarrow$ stability
3. If $\beta < 1$ + assumptions, then $\max_k \dfrac{\|h_k - h_0\|}{\|h_0\|} \xrightarrow{L\to\infty} \infty.$ $\rightarrow$ explosion

Intermediate regimes

- Challenge: describe the transition between the i.i.d. and smooth cases.

Intermediate regimes

- Challenge: describe the transition between the i.i.d. and smooth cases.
- We initialize the weights as increments of a fractional Brownian motion $(B_t^H)_{t\in[0,1]}$.

Intermediate regimes

- Challenge: describe the transition between the i.i.d. and smooth cases.
- We initialize the weights as increments of a fractional Brownian motion $(B_t^H)_{t\in[0,1]}$.
- Recall: B^H is Gaussian, starts at zero, has zero expectation, and covariance function

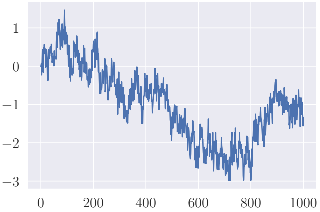
$$\mathbb{E}(B_s^H B_t^H) = \frac{1}{2}(|s|^{2H} + |t|^{2H} - |t-s|^{2H}), \quad 0 \leqslant s, t \leqslant 1.$$

Intermediate regimes

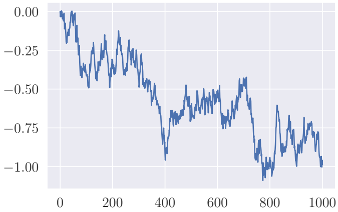
- Challenge: describe the transition between the i.i.d. and smooth cases.
- We initialize the weights as increments of a fractional Brownian motion $(B_t^H)_{t\in[0,1]}$.
- Recall: B^H is Gaussian, starts at zero, has zero expectation, and covariance function

$$\mathbb{E}(B_s^H B_t^H) = \frac{1}{2}(|s|^{2H} + |t|^{2H} - |t-s|^{2H}), \quad 0 \leqslant s, t \leqslant 1.$$

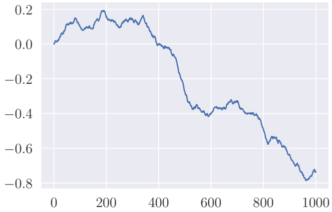
- The Hurst index $H \in (0,1)$ describes the raggedness of the process.



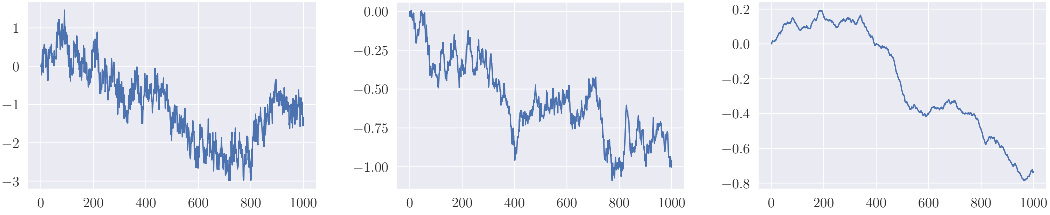
(a) $H = 0.2$



(b) $H = 0.5$

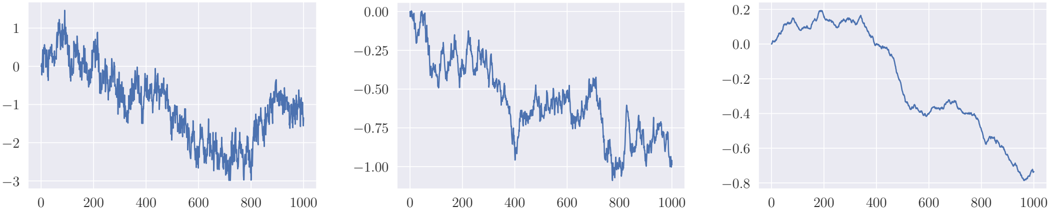


(c) $H = 0.8$



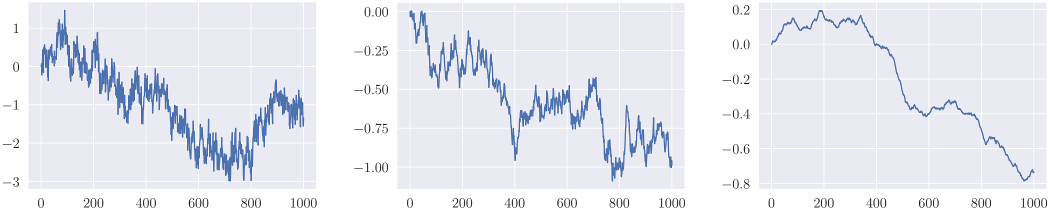
(a) $H = 0.2$ (b) $H = 0.5$ (c) $H = 0.8$

▷ $H = {}^1\!/_2$: standard Brownian motion (SDE regime).



(a) $H = 0.2$ (b) $H = 0.5$ (c) $H = 0.8$

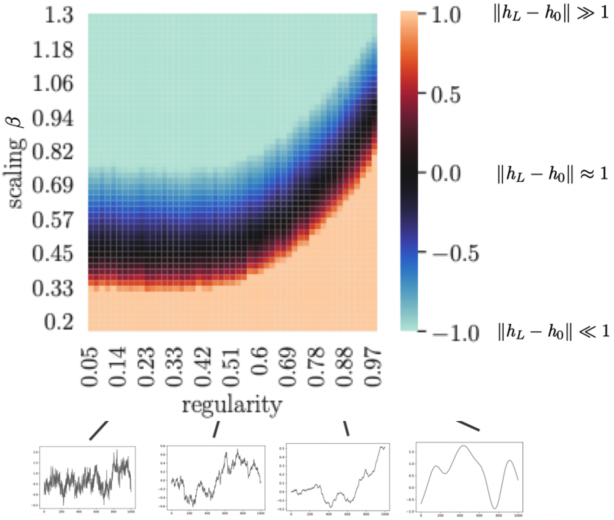
- ▷ $H = 1/2$: standard Brownian motion (SDE regime).
- ▷ $H < 1/2$: the increments are negatively correlated.
- ▷ $H > 1/2$: the increments are positively correlated.



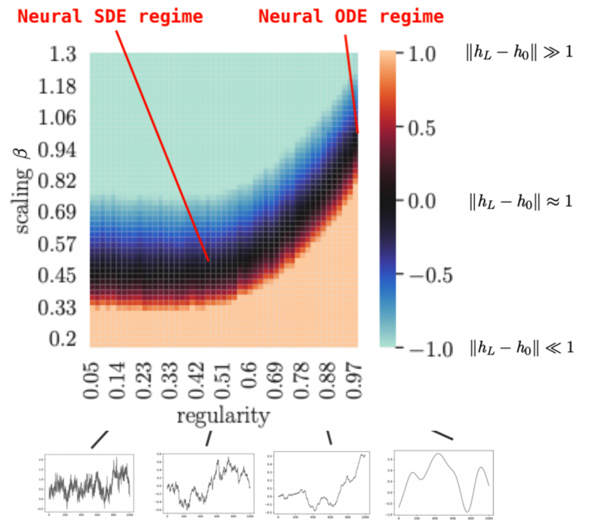
(a) $H = 0.2$ (b) $H = 0.5$ (c) $H = 0.8$

- $H = 1/2$: standard Brownian motion (SDE regime).
- $H < 1/2$: the increments are negatively correlated.
- $H > 1/2$: the increments are positively correlated.
- When $H \to 1$: the trajectories converge to linear functions (ODE regime).

A continuum of intermediate regularities



A continuum of intermediate regularities



Agenda

Learning with ResNets

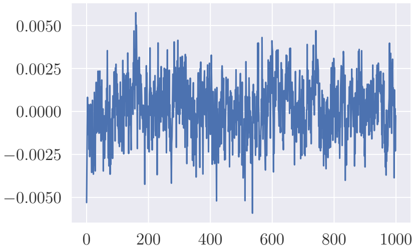
Scaling deep ResNets

Scaling in the continuous-time setting

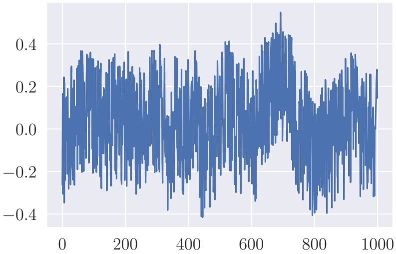
Beyond initialization

Training

Before training



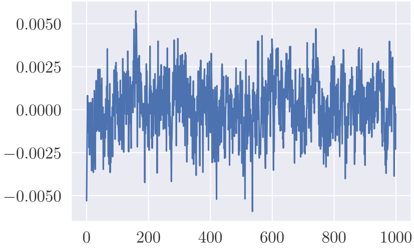
After training



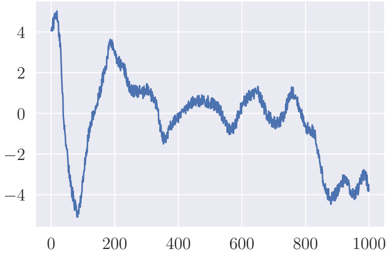
I.i.d. initialization, $\beta = {}^1\!/_2$

Training

Before training

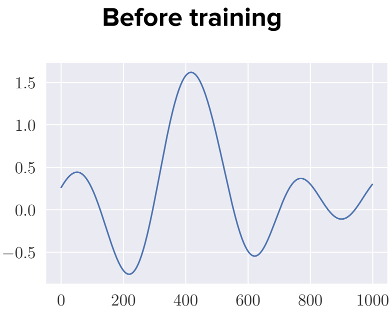


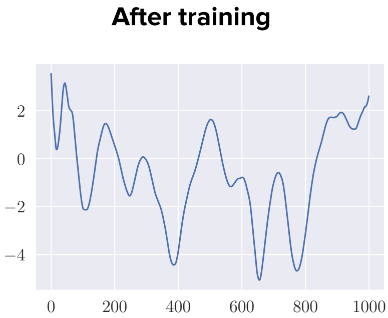
After training



I.i.d. initialization, $\beta = 1$

Training

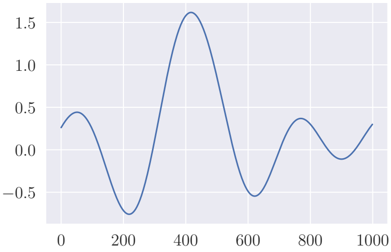




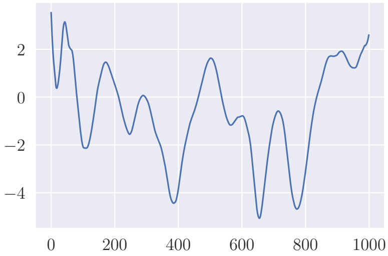
Smooth initialization, $\beta = 1$

Training

Before training



After training

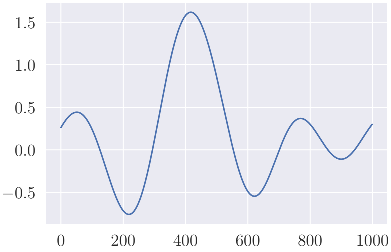


Smooth initialization, $\beta = 1$

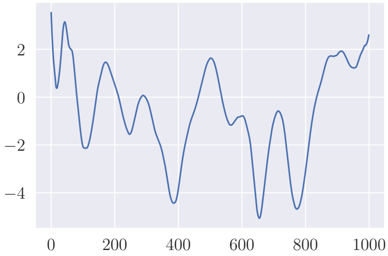
- The weights after training still exhibit a strong structure as functions of the layer.

Training

Before training



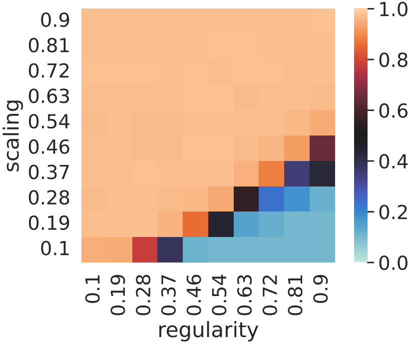
After training



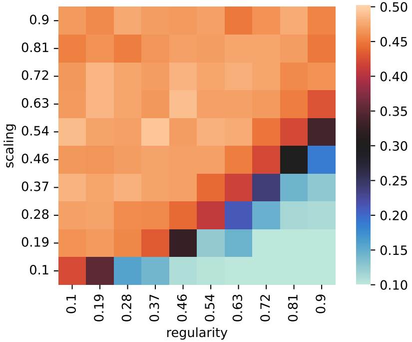
Smooth initialization, $\beta = 1$

- The weights after training still exhibit a strong structure as functions of the layer.
- Their regularity is influenced by both the initialization and the choice of β.

Performance after training



(a) On MNIST



(b) On CIFAR-10

Conclusion

- Deep limits allow to understand scaling and initialization strategies for ResNets.

Conclusion

- Deep limits allow to understand scaling and initialization strategies for ResNets.
- With standard initialization the correct scaling is $\beta = {}^1\!/_2$.

Conclusion

- Deep limits allow to understand scaling and initialization strategies for ResNets.
- With standard initialization the correct scaling is $\beta = 1/2$.
- To train very deep ResNets, it is important to adapt scaling to the weight regularity.

Conclusion

- Deep limits allow to understand scaling and initialization strategies for ResNets.
- With standard initialization the correct scaling is $\beta = 1/2$.
- To train very deep ResNets, it is important to adapt scaling to the weight regularity.
- Perspectives: what about training? how should we choose the regularity for a given problem?

Conclusion

- Deep limits allow to understand scaling and initialization strategies for ResNets.
- With standard initialization the correct scaling is $\beta = {}^1\!/_2$.
- To train very deep ResNets, it is important to adapt scaling to the weight regularity.
- Perspectives: what about training? how should we choose the regularity for a given problem?
- To know more: arXiv:2206.06929.

Thank you!

gerard.biau@sorbonne-universite.fr
perso.lpsm.paris/∼biau