DeepMind

Smoothness constraints in Deep Learning

Mihaela Rosca
Research Engineer at DeepMind, PhD student at UCL

02/03/2022

A case for new neural network smoothness constraints

Mihaela Rosca[1,2] **Theophane Weber**[1] **Arthur Gretton**[2] **Shakir Mohamed**[1]
[1]DeepMind [2]University College London
{mihaelacr,theophane,shakir}@google.com, arthur.gretton@gmail.com

Abstract

How sensitive should machine learning models be to *input* changes? We tackle the question of model smoothness and show that it is a useful inductive bias which aids generalization, adversarial robustness, generative modeling and reinforcement learning. We explore current methods of imposing smoothness constraints and observe they lack the flexibility to adapt to new tasks, they don't account for data modalities, they interact with losses, architectures and optimization in ways not yet fully understood. We conclude that new advances in the field are hinging on finding ways to incorporate *data*, *tasks* and *learning* into our definitions of smoothness.

References

References are provided at the end of the talk, additional references can be found in the paper.

If I missed out any reference, please let me know.

DeepMind

1 What is smoothness?

02/03/2022

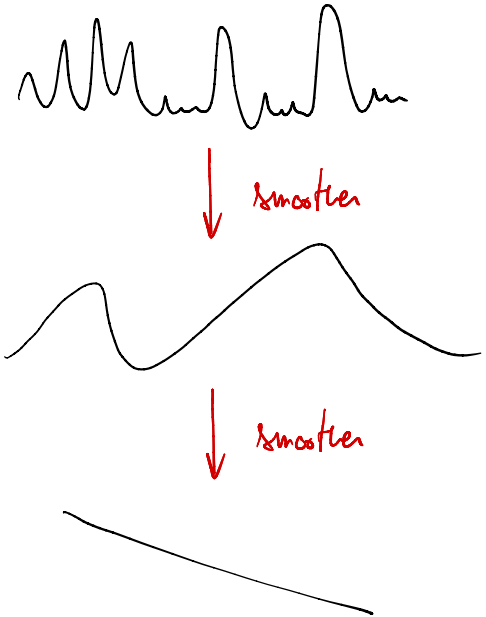
Smoothness with respect to inputs

This talk is about smoothness with respect to ĺóăĚĐĊ, not parameters.

$$\dot{U}(\hat{\mathbf{h}};\quad)$$

We are talking about how the model's øĚÐăĚÐ, not the loss changes with changes in input.

Intuition



How do we measure smoothness?

A few intuitive ways:

Lipschitz smoothness. A function is P–Lipschitz if

$$k\grave{U}(\hat{\mathbf{h}}_1; \quad) - \grave{U}(\hat{\mathbf{h}}_2; \quad)k_{\mathcal{Y}} \ 6 \ P\, k\hat{\mathbf{h}}_1 - \hat{\mathbf{h}}_2 k_{\mathcal{X}} \qquad 8\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2 \ 2 \ \mathrm{X} \tag{1}$$

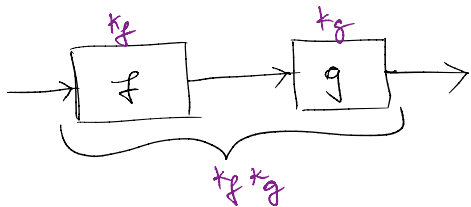
Rademacher's theorem: if $\mathrm{X} \quad \mathrm{R}^{\grave{o}}$ is an open set and $\mathrm{Y} = \mathrm{R}^{\check{a}}$ and $\grave{U}$ is P–Lipschitz then $k0\grave{U}(\hat{\mathbf{h}})k \ 6 \ P$ wherever the total derivative $0\grave{U}(\hat{\mathbf{h}})$ exists.

The norm of the model Jacobian $0(\hat{\mathbf{h}}) = \frac{\acute{E}\grave{U}(\hat{\mathbf{h}})}{\acute{E}\hat{\mathbf{h}}}$. Jacobian metrics account for how $\acute{I} \cdot \ddot{A}\dot{P}$ $\acute{E}\r{s}\grave{o}\ \acute{I}\acute{o}\dot{C}\r{k}\o\acute{o}$ of the function output is allowed to vary as individual input dimensions change.

Composition of Lipschitz functions

If $\grave{U}$ and $\acute{U}$ are Lipschitz with constants $P_{\grave{U}}$ and $P_{\acute{U}}$, $\grave{U} \quad \acute{U}$ is Lipschitz with constant $P_{\grave{U}}P_{\acute{U}}$.

Since commonly used activation functions are 1-Lipschitz, the task of ensuring a neural network is Lipschitz reduces to constraining the learneable layers to be Lipschitz.



The Lipschitz constant of linear operators

The Lipschitz constant of a linear operator A under common norms ($\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$) is $\sup_{h \neq 0} \frac{\|Ah\|}{\|h\|}$.

Many neural networks layers are linear operators:

- linear layers
- convolutional layers
- BatchNom

DeepMind

2 Smoothness constraints for neural networks

02/03/2022

Indirect smoothness constraints

A lot of common regularization methods indirectly target smoothness:

- early stopping
- dropout
- weight decay
- data augmentation

While interesting in their own right, in this talk we will primarily focus on methods which explicitly target smoothness regularisation.

Soft constraints: gradient penalties

Soft constraints add a regularisation term to the loss function to encourage Lipschitz smoothness, by adding a gradient penalty to the loss function $\mathrm{L}(\quad)$:

$$\mathrm{L}(\quad) + \quad \mathrm{E}_{\check{a}_{\acute{c}\acute{t}\acute{u}(\hat{\mathbf{h}})}} \left(k \, r_{\hat{\mathbf{h}}} \grave{U}_{\theta}(\hat{\mathbf{h}}) k_2^2 \; - \; P^2 \right)^2 \tag{2}$$

where

is a regularization coefficient

$\check{a}_{\acute{c}\acute{t}\acute{u}(\hat{\mathbf{h}})}$ is the distribution at which the regularization is applied, which can either be the data distribution or around it.

Soft constraints - Spectral Regularisation

Spectral regularization uses the sum of the spectral norms - the largest singular value - of each layer as a regularization loss to encourage Lipschitz smoothness:

$$L(\ ) + \sum_{\S} \text{jj } \S \text{ jj}_2 \tag{3}$$

where

is a regularization coefficient

jj § jj$_2$ is the spectral norm of §, computed using power iteration.

For convolutional layers, the weights get reshaped to a 2D matrix. This technically does not compute the Lipschitz constant of the operator, but seems good enough in practice.

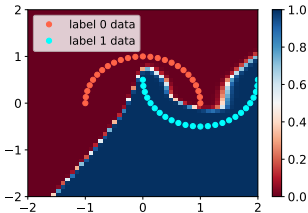
Hard constraints - Spectral Normalisation

Spectral Normalization ensures the learned models are 1-Lipschitz by adding a node in the computational graph of the model layers by replacing the weights with their normalized version:

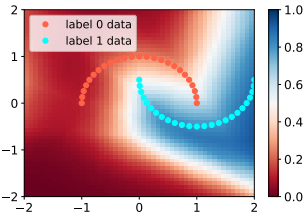
$$L(\S) \; ! \; L(\;(\S)) \tag{4}$$

where $(\S) = \S = jj \S jj_2$ and $jj \S jj_2$ is the spectral norm of $\S$.

Smoothness constraints on two moons



(a) MLP.
No regularisation.



(b) Gradient penalty at data;
$P = 1$.

(c) Spectral norm;
$P = 1$.

DeepMind

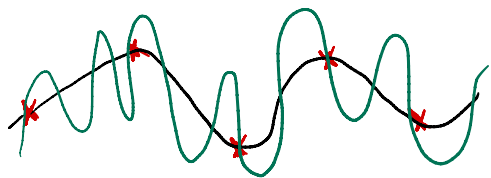
3 The benefits of smoothness constraints

02/03/2022

Generalisation

Methods that encourage smoothness such as weight decay, dropout, data augmentation and early stopping have been long shown to aid generalization.

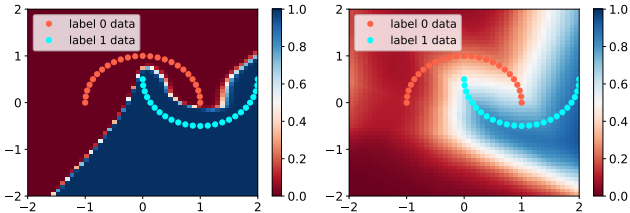
Recent works directly connect smoothness to classification margins, and use that to obtain empirical gains on standard image classification tasks.



Reliable uncertainty estimates

Neural networks provide notoriously unreliable uncertainty estimates.

To leverage the power of neural networks to obtain reliable uncertainty estimates, by combining smooth neural feature learners with non-softmax decision surfaces.



(a) 4 layer MLP.

(b) Gradient penalty at data; $P = 1$.

Robustness to adversarial attacks

Robustness for classifiers can be defined by ensuring that inputs in the same -ball result in the same function output:

$$k\hat{h} - \hat{h}^{0}k \; 6 \quad =) \quad \arg\max \dot{U}(\hat{\mathbf{h}}) = \arg\max \dot{U}(\hat{\mathbf{h}}^{0}) \tag{5}$$

This definition is directly connected with Lipschitz smoothness.

Initial approaches to combating adversarial attacks focused on data augmentation methods and only more recently smoothness constraints have come into focus.

Improved generative modelling performance

Smoothness constraints have become part of many state of the art generative models:

- GANs: Spectral Normalisation or gradient penalties are present in many GANs.
- Variational Autoencoders: Spectral regularization boosts performance and stability.
- Normalising Flows: benefit from smoothness constraints through powerful invertible layers built using residual connections $\acute{U}(\hat{\mathbf{h}}) = \hat{\mathbf{h}} + \grave{U}(\hat{\mathbf{h}})$ where $\grave{U}$ is Lipschitz.

Figure: GAN performance is improved when the discriminator uses Spectral Normalisation.

More informative critics

Critics (ìÍ·ĆóÍÉ ·ăăĆøĥßò·ÐøĆĊ Ðø ßóÐĆ·ÄÐ·ÃìÍ ÉÍÄßĊßøó ÙĚóÄÐßøóĊ), have become more and more important in machine learning:

- Generative models: The GAN critic is used to approximate distributional divergences and distances.
- Representation learning: parametric critics are trained to approximate another intractable quantity, the mutual information, using the Donsker–Varadhan or similar bounds.
- Reinforcement learning: parametric critics are used to approximate value and state-value functions.

More informative critics

Smooth critics provide more informative models the are training:

- Generative models: Smooth approximations to decision surfaces of $\check{U}$-divergences provide useful gradients when the underlying divergence does not.
- Representation learning: tighter bounds do not lead to better representations; the success of these methods is attributed to the inductive biases of the critics.
- Reinforcement learning: see next talk.

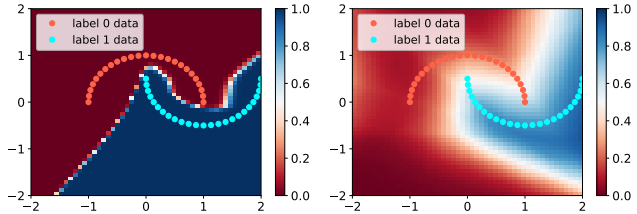
DeepMind

4 The downsides of smoothness con-strains

02/03/2022

Weak models

Needlessly limiting the capacity of our models by enforcing smoothness constraints is a significant danger: a constant function is very smooth, but not very useful.

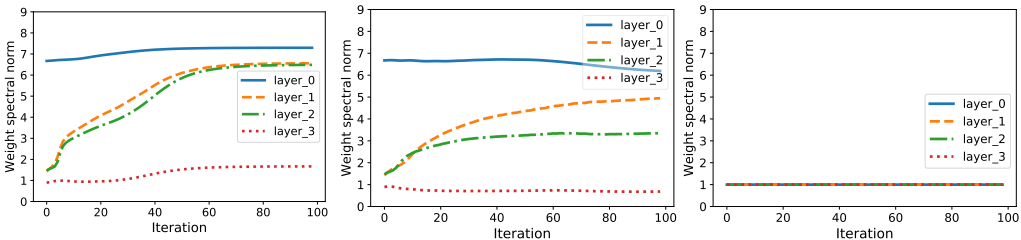


(a) 4 layer MLP. (b) Gradient penalty at data; $P = 1$. (c) Spectral normalisation $P = 1$.

Figure: Smoothness constraints can limit model capacity and decrease performance.

Weak models

Soft, local methods like gradient penalties which only apply regularisation in one part of the space can be less restrictive.



(a) Unregularized. (b) Gradient penalty. $P = 1$. (c) Spectral Normalization. $P = 1$.

Figure: Lipschitz constant of each layer of an MLP trained on the two moons dataset. Smaller means smoother.

Overlooked interactions with optimization

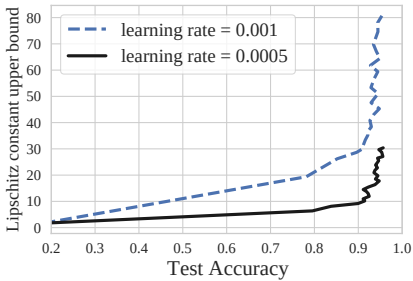
Smoothness has been traditionally seen as changing the **ò øÉÍì**. We show here that smoothness has strong interactions with optimisation (more in the next talk!).

Some smoothness regularization techniques affect optimization by changing the loss function (gradient penalties, spectral regularization) or the optimization regime directly (early stopping).

Even if they don’t explicitly change the loss function or optimization regime, **Ċ ò øøÐÞóÍĊĊ ÄøóĊÐĆ·ßóÐĊ ·ÙÙÍÄÐ ÐÞÍ ă·ÐÞ ÐÞÍ ò øÉÍì Ð·êÍĊ Ðø ĆÍ·ÄÞ ÄøóğÍĆÚÍóÄÍ**.

Overlooked interactions with optimization

Training with different learning rates leads to different smoothness properties of models; imposing the Ċ· ò Í constraint on ÐÞÍ ò øÉÍì trained with different learning rates will have vastly different outcomes.



Lower means smoother.

Overlooked interactions with optimization

Other hyperparameters, like momentum, are also affected by smoothness constraints.

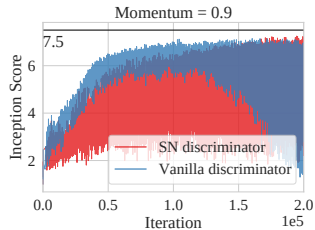
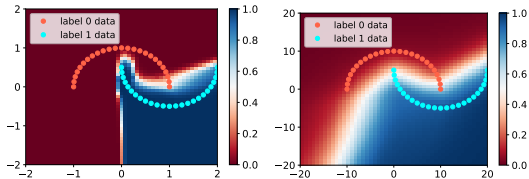


Figure: Spectral Normalization requires low momentum in GAN training. Higher is better.

Sensitivity to data scaling

Sensitivity to data scaling of smoothness constraints can make training neural network models sensitive to additional hyperparameters.



(a) Spectral norm; $P = 1$.

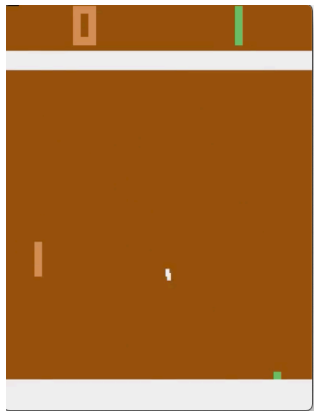
(b) Spectral norm; $P = 10$.

(c) Spectral norm; scaling the data by 10.

Figure: Changing the scale of the data or the Lipschitz constant can lead to vastly different results.

Wrong model priors

Depending on the task, smoothness might not be the right model prior. In reinforcement learning, one pixel change might require a big change in the value function.



DeepMind

5 The future of smoothness constraints

02/03/2022

New ways of defining smoothness

Improving model generalization and robustness requires specifying the right level of invariance by using task information to define smoothness constraints.

We have to ask what are the desired properties of p such that

$$k\dot{\psi}(p(\hat{\mathbf{h}})) \; - \; \dot{\psi}(p(\mathbf{H}))k \; 6 \; kp(\hat{\mathbf{h}}) \; - \; p(\mathbf{H})k \tag{6}$$

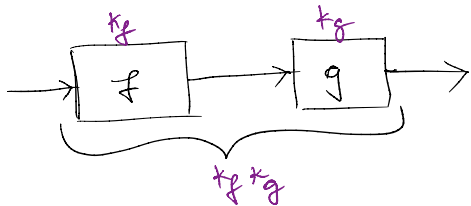
To ensure the mapping p does not discard task relevant information in the data, maintains useful diversity and accounts for input modalities, it has to be $\mathbf{\acute{E}\cdot\eth\cdot}$ and $\mathbf{\eth\cdot\dot{C}\hat{e}}$ dependent.

New ways of measuring smoothness

Measuring smoothness of a function parametrized by a neural network is challenging even for the most common measure of smoothness used in machine learning, Lipschitzness.

Currently, we only have loose upper bounds available, or more accurate methods which are very costly.

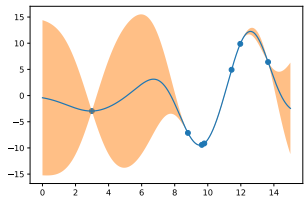
To further improve the effect of smoothness regularisation methods, we have to understand them better and measure smoothness more accurately.



New learning paradigms

Combining non parametric methods with feature learning is a promising approach to learning smooth decision surfaces. Requires:

- learning the right features (which themselves might have to be smooth)
- scaling non parametric methods such as Gaussian Processes, Support Vector Machines and Nearest Neighbours methods to large datasets.



DeepMind

6 Conclusion

02/03/2022

DeepMind

7 References

Please see the paper for a comprehensive list of references.
If I missed anything, please let me know.

02/03/2022

References - Methods

Gradient penalties and their use in GANs [SGSR17, GAA+17, FRL+18, ASBG18, KAHK17]
Spectral Regularisation [YM17]
Spectral Normalisation [MKKY18]

References - Benefits

Generalisation [SGSR17, Bar97, GRS18, HRS16, NKB+19, SHK+14]

Reliable uncertainty estimates [vASTG20, LLP+20]

Robustness to adversarial attacks [CBG+17, NBA+18, SGSR17, LGO18]

Improved generative modeling performance [MKKY18, BDS18, BGC+19, VK20]

More informative critics [FRL+18, AZG20, SZA19, ASBG18, ACB17, GAA+17, FRL+18, BDS18, ASBG18, ZLS+19, YM17, TDR+20, DJ20]

References - Future

Weak models [JBZB18, FRH+19]

Interactions with optimisation [GAA+17]

New Ways Of Measuring smoothness [VS18, CP19, FRH+19, SGSR17]

References I

Martin Arjovsky, Soumith Chintala, and Léon Bottou, *§·ĊĊÍĆĊĐÍßó ÚÍóÍĆ·ĐßğÍ ·ÉğÍĆĊ·Ćß·ì óÍĐĠøĆêĊ*, Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 214–223.

Michael Arbel, Dougal Sutherland, Mikołaj Bińkowski, and Arthur Gretton, *^ó ÚĆ·ÉßÍóĐ ĆÍÚËì·ĆßĬÍĆĊ ÙøĆ ò òÉ Ú·óĊ*, Advances in neural information processing systems, 2018, pp. 6700–6710.

Michael Arbel, Liang Zhou, and Arthur Gretton, *P·ìÍŝ §ÞÍó ÍóÍĆÚĦŰÃ·ĊÍÉ ìÍ·ĆóßóÚ ò ÍÍĐĊ ·ÉğÍĆĊ·Ćß·ì ĐĆ·ßóßóÚ*, arXiv preprint arXiv:2003.05033 (2020).

Peter L Bartlett, *?øĆ ğ·ìßÉ ÚÍóÍĆ·ìßĬ·Đßøó ĐÞÍ ĊßĬÍ øÙ ĐÞÍ ĠÍßÚÞĐĊ ßĊ ò øĆÍ ßò ăøĆĐ·óĐ ĐÞ·ó ĐÞÍ ĊßĬÍ øÙ ĐÞÍ óÍĐĠøĆê*, Advances in neural information processing systems, 1997, pp. 134–140.

Andrew Brock, Jeff Donahue, and Karen Simonyan, *R·ĆÚÍ ĊÄ·ìÍ Ú·ó ĐĆ·ßóßóÚ ÙøĆ ÞßÚÞ l·ÉÍìßĐĦ ó·ĐĚĆ·ì ßò ·ÚÍ ĊĦóĐÞÍĊßĊ*, International Conference on Learning Representations, 2018.

References II

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen, Ɛóğ́ÍĆĐ℞Ãì Í ĆÍĊ℞ÉĚ·ì óÍĐĠøĆêĊ, International Conference on Machine Learning, 2019, pp. 573–582.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier, ì·ĆĊÍğ·ì óÍĐĠøĆêĈŠ ß ò ăĆøğ℞óÚ ĆøÃĚĆĐóÍĊĊ Đø ·ÉğÍĆĊ·Ć℞·ì ÍĥÌ· ò ăìÍĊ, Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 854–863.

Patrick L Combettes and Jean-Christophe Pesquet, R℞ăĊÄÞ℞ĐĬ ÄÍĆĐ℞ł·Ä·ĐÍĊ ÙøĆ óÍĚĆ·ì óÍĐĠøĆê ĊĐĆĚÄĐĚĆÍĊ ÉĆ℞ğÍó ÃĦ ·ğÍĆ·ÚÍÉ ·ÄĐ℞ğ·Đ℞øó øăÍĆ·ĐøĆĊ, arXiv preprint arXiv:1903.01014 (2019).

Pierluca D'Oro and Wojciech Jaśkowski, DøĠ Đø ìÍ·Ćó · ĚĊÍÙĚì ÄĆ℞Đ℞Äš ò øÉÍìŰÃ·ĊÍÉ ·Ä℞Đ℞øóŰÚĆ·ÉßÍóĐŰÍĊĐ℞ò ·ĐøĆ ăøì℞ÄĦ øă℞Đ℞ò ßĬ·Đ℞øó, arXiv preprint arXiv:2004.14309 (2020).

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas, 4Ùł·Ä℞ÍóĐ ·óÉ ·ÄÄĚĆ·ĐÍ ÍĊĐ℞ò ·Đ℞øó øÙ ì℞ăĊÄÞ℞ĐĬ ÄøóĆĐ·óĐĊ ÙøĆ ÉÍÍă óÍĚĆ·ì óÍĐĠøĆêĊ, Advances in Neural Information Processing Systems, 2019, pp. 11427–11438.

References III

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow, *X · óH ă · ÐÐĊ Ðø ÍqĚĞĽÃĆĽĚ òŝ @ · óĊ Éø óøÐ óÍÍÉ Ðø ÉÍÄĆÍ · ĊÍ · ÉĽğÍĆÚÍóÄÍ · Ð ÍğÍĆH ĊÐÍă*, International Conference on Learning Representations, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, *Ė ò ăĆøğÍÉ ĐĆ · ĽóĽóÚ øÙ Ġ · ĊĊÍĆĊÐÍßó Ú · óĊ*, Advances in neural information processing systems, 2017, pp. 5767–5777.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir, *pßŇÍŰßóÉÍă ÍóÉÍóÐ Ċ · ò ăìÍ Äø ò ăìÍĥßÐH øÙ óÍĚĆ · ì óÍÐĠøĆêĊ*, Conference On Learning Theory, PMLR, 2018, pp. 297–299.

Moritz Hardt, Ben Recht, and Yoram Singer, *vĆ · ßó Ù · ĊÐÍĆŝ ÚÍóÍĆ · ìßŇÍ ÃÍÐÐÍĆŝ pÐ · ÃĞĽßÐH øÙ ĊÐøÄÞ · ĊÐĽÄ ÚĆ · ÉßÍóÐ ÉÍĊÄÍóÐ*, International Conference on Machine Learning, PMLR, 2016, pp. 1225–1234.

Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge, *4ĥÄÍĊĊĽğÍ ßóğ · Ćŝ · óÄÍ Ä · ĚĊÍĊ · ÉğÍĆĊ · Ćŝ · ì ğĚìóÍĆ · ÃĞĽßÐH*, International Conference on Learning Representations, 2018.

References IV

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira, ^ó Äøóğ ÍĆÚÍóÄÍ ·óÉ ĊĐ·ÃßßĐĦ øÙ Ú·óĊ, arXiv preprint arXiv:1705.07215 (2017).

Carlos Eduardo Rosar Kos Lassance, Vincent Gripon, and Antonio Ortega, R·ăì·Äß·ó óÍÐĠøĆêĈŚ *øĚóÉßóÚ ßóÉßÄ·ÐøĆ ÙĚóÄÐßøó Ċò øøÐÞóÍĊĊ ÙøĆ óÍĚĆ·ì óÍÐĠøĆê ĊøÃĚĊĐóÍĊĊ, arXiv preprint arXiv:1805.10133 (2018).

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan, pßò ăìÍ ·óÉ ăĆßóÄßăìÍÉ ĚóÄÍĆĐ·ßóÐĦ ÍĊĐßò ·Ðßøó ĠßÐÞ ÉÍÐÍĆò ßóßĊĐßÄ ÉÍÍă ìÍ·Ćóßóڃß· ÉßĊĐ·óÄÍ ·Ġ·ĆÍóÍĊĊ, arXiv preprint arXiv:2006.10108 (2020).

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, pă ÍÄÐĆ·ì óøĆò ·ìßĬ·Ðßøó ÙøĆ ÚÍóÍĆ·ÐßğÍ ·ÉğÍĆĊ·Ćß·ì óÍÐĠøĆêĊ, International Conference on Learning Representations, 2018.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein, pÍóĊßÐßğßĐĦ ·óÉ ÚÍóÍĆ·ìßĬ·Ðßøó ßó óÍĚĆ·ì óÍÐĠøĆêĈŚ ·ó Í ò ăßĆßÄ·ì ĊĐĚÉĦ, International Conference on Learning Representations, 2018.

References V

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever, *Deep Double Descent: Where Bigger Models and More Data Can Hurt*, International Conference on Learning Representations, 2019.

Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues, *Generalization Error of Invariant Classifiers*, IEEE Transactions on Signal Processing **65** (2017), no. 16, 4265–4280.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, The journal of machine learning research **15** (2014), no. 1, 1929–1958.

Florian Schäfer, Hongkai Zheng, and Anima Anandkumar, *Implicit Regularization of Discrete Gradient Dynamics in Deep Linear Neural Networks*, arXiv preprint arXiv:1910.05852 (2019).

Michael Tobias Tschannen, Josip Djolonga, Paul Kishan Rubenstein, Sylvain Gelly, and Mario Lučić, *On Mutual Information Maximization for Representation Learning*, International Conference on Learning Representations, 2020.

References VI

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal, *pßò ăìÍ ·óÉ ĊÄ·ì·ÃìÍ Íă℞ĊĐÍò℞Ä ĚóÄÍĆĐ·ß óĐĦ ÍĊĐßò ·Đ℞øó ĚĊ℞óÚ · Ċ℞óÚîÍ ÉÍÍÍă ÉÍÐÍĆò℞óß℞ĊĐ℞Ä óÍĚĆ·ì óÍÐĠøĆê*, International Conference on Machine Learning (2020).

Arash Vahdat and Jan Kautz, *Yğ· Íŝ ÉÍÍÍă Þ℞ÍĆ·ĆÄÞ℞Ä·ì ğ·Ć℞·Đ℞øó·ì ·ĚĐøÍóÄøÉÍĆ*, Advances in Neural Information Processing Systems **ňň** (2020).

Aladin Virmaux and Kevin Scaman, *R℞ăĊÄÞ℞ĐĬ ĆÍÚĚì·Ć℞ĐĦ øÙ ÉÍÍÍă óÍĚĆ·ì óÍÐĠøĆêĊŝ ·ó·ìĦĊ℞Ċ ·óÉ ÍÙłÄ℞ÍóÐ ÍĊĐßò ·Đ℞øó*, Advances in Neural Information Processing Systems, 2018, pp. 3835–3844.

Yuichi Yoshida and Takeru Miyato, *pă ÍÄĐĆ·ì óøĆò ĆÍÚĚì·Ć℞Ĭ·Đ℞øó ÙøĆ ßò ăĆøğ℞óÚ ÐÞÍ ÚÍóÍĆ·ì℞Ĭ·ÃĥĦ℞ĐĦ øÙ ÉÍÍÍă ìÍ·Ćó℞óÚ*, arXiv preprint arXiv:1705.10941 (2017).

Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, and Zhihua Zhang, *R℞ăĊÄÞ℞ĐĬ ÚÍóÍĆ·Đ℞ğÍ ·ÉğÍĆĊ·Ć℞·ì óÍÐĊ*, International Conference on Machine Learning, 2019, pp. 7584–7593.