Generalisation in Deep Learning: Towards a Prescriptive Theory

Charles Li

Abstract

One interesting phenomenon in deep learning is the non-overfitting puzzle: despite having much more parameters than training samples, deep neural networks tend not to overfit even in the absence of explicit regularisation[62], demonstrating qualitatively different properties compared with old-school learning models. However, elucidating why deep neural networks generalise so well on real-world data remains an ongoing area of research.

In this review paper we endeavour to present results and theories that may contribute to the excellent generalisation capability of deep neural networks. Furthermore, we illustrate that studies of generalisation is not merely of academic interest, using adversarial attacks as an example.

Contents

1 Introduction

The last decade has witnessed rapid development of deep learning techniques and deployment of machine learning algorithms powered by deep learning. Successful applications of deep neural networks include areas such as computer vision[33][25][52], natural language processing[22][12] and machine translation[61]. Despite the huge achievements, a puzzle still remains to be resolved: why does deep neural networks generalise so well on unseen data?

It is believed[37] that deep learning works via means of "hierarchical feature extraction": as primitive features are identified in layers close to the input nodes, more abstract features are recognised in deeper layers after processing features forwarded from shallower layers. However, this is merely a *descriptive* explanation, as it does not tell us exactly how features are extracted. Our goal should be to obtain a *prescriptive* theory that can ideally predict the generalisation power of a given model.

Such a unified framework is still elusive, but there have been incremental discoveries which I believe point towards the right direction. This review will encompass many of those findings, along with some necessary background, and examples showing the importance of understanding generalisation. Specifically, the following sections will be covered:

- Background knowledge in classical statistical learning theory including three classical complexity control measures; introduction to the manifold hypothesis;
- Discussion about an important experiment, as well as factors that affect generalisation capabilities of deep neural networks;
- A historical perspective of neural networks, and possible interpretations of deep learning via statistical mechanics and renormalisation group;
- Realistic settings such as generative adversarial networks and adversarial examples.

A key metric to assess how well machine learning models perform (or how well they *generalise*) is the generalisation error. Throughout this review it is defined as the difference between test error and training error. As is mentioned earlier, given a specific model architecture and training algorithm, a prescriptive theory should be able to produce a bound on the generalisation error *a priori*; such a theory would help facilitate more interpretable and reliable model designs.

2 Statistical Learning Theory and Manifold Hypothesis

2.1 Complexity Measures in Statistical Learning Theory

Statistical learning theory has proposed a few complexity measures that are supposed to control generalisation errors and indeed work well with convex optimisation models such as linear regression and support vector machine (SVM). Here we introduce three of them: Vapnik-Chervonenkis dimension[57]; Rademacher complexity[5]; uniform stability[8].

2.1.1 Vapnik-Chervonenkis Dimension

Vapnik proposed the empirical risk minimisation framework in his 1998 work[57]. Using the ERM framework he obtained an upper-bound on the *real risk* (or *real* generalisation error) $R(f)$ with respect to the *empirical risk* $R_{\text{emp}}(f)$ and *VC dimension* of a function

class $\mathcal{F}$. Here, the real risk is defined as the expectation of a loss function $\mathcal{L}$ over the underlying distribution, whereas the empirical risk is the average loss over all data points, and f $\in \mathcal{F}$. Two concepts are introduced to obtain this result: shattering and VC dimension.

Definition 2.1 A set of n instances $X_1, ..., X_n$ from the input space $\mathcal{X}$ is said to be *shattered* by a function class $\mathcal{F}$ if all the (possible) 2^n labellings of them can be generated using functions from $\mathcal{F}$.

Definition 2.2 The *VC dimension* of a function class $\mathcal{F}$, denoted VC($\mathcal{F}$), is the largest integer h such that there exists a sample of size h which is shattered by $\mathcal{F}$. VC($\mathcal{F}$) = ∞ if arbitrarily large samples can be shattered.

For example, the set of linear classifiers (straight lines) in $\mathbb{R}^2$ can shatter any set of 3 non-collinear points, but not any set of 4 points. Thus, the VC dimension is 3.

Fundamental Result For all $f \in \mathcal{F}$, with probability at least 1 - δ, we have:

$$R(f) \leq R_{\text{emp}}(f) + \sqrt{\frac{h(\log(2n/h) + 1) - \log(\delta/4)}{n}} \tag{1}$$

where h = VC($\mathcal{F}$) and n is the sample size.

Hence, given more and more data, the empirical risk eventually approaches the minimum value of expected error of the functions in $\mathcal{F}$ if and only if $\mathcal{F}$ has finite VC dimension.

2.1.2 Rademacher Complexity

Definition 2.3 The Rademacher complexity of a hypothesis class $\mathcal{H}$ on a dataset $\{x_i\}_{i=1}^n$ is defined as[5]

$$\text{Rad}(\mathcal{H}) := \mathbb{E}_\sigma[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i)] \tag{2}$$

where $\sigma_i \in \{\pm 1\}$ are independent and identically distributed uniform random variables. It can be understood as a measure of how well $\mathcal{H}$ can fit randomly assigned binary labels, or how easy to learn the function class $\mathcal{H}$ is. Note that Rad($\mathcal{H}$) $\approx$ 1 if the model class $\mathcal{H}$ fits **any** random labels perfectly. Normally models with low Rademacher complexity are easier to train.

2.1.3 Uniform Stability

Uniform stability[8] is different from VC dimension and Rademacher complexity in that it is based on *sensitivity analysis*, that is, how the training algorithm reacts to replacement of a single example in the training set. It can be applied to regularisation based algorithms and obtains interesting bounds in the case of SVMs. This approach is solely training-algorithm-dependent.

2.1.4 Remark

VC dimension and Rademacher complexity are decisively descriptive. As we will see in the next section, these three criteria are not particularly informative in the context of deep neural networks.

2.2 Manifold Hypothesis

The manifold hypothesis states that real-world high-dimensional data tend to lie on the vicinity of a low-dimensional manifold.

If the manifold hypothesis is true (it appears to be), we would naturally conclude that deep neural networks could "magically" discover patterns in data by confining to a low-dimensional manifold, thanks to their "inductive biases" such as architecture, regularisation etc.. This descriptive intuition is consistent with the hierarchical feature extraction hypothesis introduced earlier.

3 How Do Deep Neural Nets Generalise?

3.1 Memorisation or Pattern Recognition?

3.1.1 Zhang et al. (2017)

In ICLR 2017, a paper titled "Understanding Deep Learning Requires Rethinking Generalization" received the best paper award and sparked much discussion. The authors question the conventional view of generalisation that it is due to either properties of the model family or regularisations used during training, and suggest that a new theory of network capacity is required[62]. They support their view by running randomisation tests[15], in which different neural networks (Inception[52], AlexNet[33] and MLP) are trained on images from CIFAR10[32] and ImageNet[33] with randomly assigned labels.

If the proposition of "hierarchical feature extraction" is correct, we would intuitively expect the training not to converge, or at least converge very slowly, since there is no structure in randomised input whatsoever and thus learning is impossible. However, it appears that training not only converges (fairly quickly), but achieves perfect error (zero) and accuracy (100% or close) on the training set for all neural network models.

The authors conduct two more sets of experiments. First, they replace image pixels with completely random pixels such as Gaussian noises, and obtain zero training error again. They then try to control the degree of label randomisation in the original test so that only a certain proportion of labels are randomly assigned, leading to a series of learning problems where there are varying degrees of "real signals" and noises. In this scenario, the generalisation error increases as the noise level is increased gradually. They summarise their finding as: *Deep neural networks easily fit random labels.*

Moreover, since the *effective capacity* (or *representation power*, *expressivity*) of deep neural networks are more than enough to memorise the entire dataset (they prove that even a two-layer neural network can express any labelling of any finitely-sized sample), they further suggest that neural networks fit noisy data by "brute-force memorisation" (while still able to capture valid signals in the data if there are any).

A direct implication is that, the three aforementioned criteria in statistical learning theory are not good explanations for the generalisation power of deep neural networks. For neural networks as a whole, the corresponding VC dimension is ∞ and the bound in section 2.1.1 becomes meaningless since it involves square root of $-\infty$; the Rademacher complexity is approximately one (because the neural networks fit random data perfectly well) which is a trivial upper bound with little use in a realistic setting. As for uniform stability, it is solely determined by the training algorithm and thus is not data-dependent.

3.1.2 Discussion

Some people disagree on the notion of "brute-force memorisation" being part of the learning strategy of neural networks. Krueger et al.[34] argue that "brute-force memorisation" cannot fit a dataset without *first* trying to capture patterns in the dataset. They argue that in order to rapidly reduce training loss, neural networks attempt to learn and refine patterns common to training examples first, and only employ case-by-case memorisation as a last resort. They assert their view by demonstrating empirically some qualitative difference between training on noisy and real data.

They run randomisation tests similar to Zhang et al.'s, train two-layer ReLU-MLPs on MNIST, and report five findings:

1) Training noisy data requires more capacity, i.e. a larger network with more hidden units;

2) Time to convergence is longer for random labels (consistent with Zhang et al.'s finding), but interestingly *shorter* if the image pixels are randomised (by a Gaussian) instead;

3) Reducing network capacity or increasing size of training data slows down training for real data and noise alike, but the impact on real data is lighter. This is possibly because patterns are consistent among real data, and adding more real training examples require less change to the learned parameters, whereas noisy examples are uncorrelated and require much more adjustment;

4) The loss function of real data around learned minima is more "flat" compared with that of noisy data (see further discussion in section 3.3), indicating a "simpler" form of loss function[27];

5) For certain fine-tuned networks, explicit regularisation like dropout hinders the model's ability to memorise noise while maintaining their generalisation performance. This is an important observation; see Section 3.4.

They thus contend deep neural networks prefer simple hypotheses over brute-force memorisation whenever possible. Nevertheless, they acknowledge the mystery around the generalisability of deep neural networks, and hope this question can be addressed in a formal manner.

Remark 1) does not constitute a convincing argument. Bartlett[3] points out that weight norms are far more important a contributing factor to network capacity than the network size, and Neyshabur et al.[46] demonstrate empirically that **size does not behave as a capacity control parameter for multilayer perceptrons** - yet another counterintuitive finding. See Section 3.2 for recent work.

Both arguments have merits. Zhang's paper raises up an important question in deep learning by designing a smart experiment, whereas mounting empirical evidence suggests deep neural networks are indeed capable of extracting features from real world data. Luckily we have made *some* progress since their paper's publication, and some of those works are presented in the few following sections.

3.2 Recent Work on Generalisation Bound

Building upon Vapnik, Bartlett and Neyshabur et al.'s works, substantial progress have been made on obtaining a more precise bound on generalisation errors. Returning to the

"old" idea of PAC-Bayes bound[36], Dziugaite & Roy[14] are able to obtain a meaningful bound on one family of stochastic neural networks whose weights are randomly deviated away from a trained MNIST classifier. Surprisingly, the PAC-Bayes bound does not grow much even as the size of the stochastic neural networks become several times larger. Bartlett et al.[4] present a generalisation bound based on "spectral complexity": the product of the Lipschitz constant (product of spectral norms of the weight matrices in a network) and a correction factor. Importantly, their result does not explicitly depend on the size of the network. However, it is still implicitly bounded by $\mathcal{O}(d^{1.5})$ where d denotes the depth of the network. Combining the two approaches together, Neyshabur et al.[45] identify a generalisation bound for feedforward neural networks in terms of the Lipschitz constant and the Froebenius norm of the weights, improving the dependency on network depth d to $\mathcal{O}(d\sqrt{\ln d})$. In a very recent work of Golowich et al.[18], the dependency is pushed further to $\mathcal{O}(\sqrt{d})$, and under some technical assumptions, $\mathcal{O}(1)$, i.e. **the generalisation bound is fully independent of the network size**. This is a truly monumental result.

3.3 Sharp versus Flat Minima

Some researchers[27] have suggested that the "sharpness" of minima that training algorithms converge to may play a role in determining how well these solutions generalise. There are different interpretations of "flatness" but here we will stick to curvature. Curvature can be linked to the second derivative for functions in 1D, and eigenvalues of the Hessian matrix in higher dimensions.

Hochreiter & Schmidhuber[27] and Chaudhari et al.[10] propose that flat minima with small curvature generalise better than sharp minima with large curvature. The aforementioned Krueger et al.[34] also observe a significant increase in the absolute value of the highest eigenvalue of the Hessian when models are trained on random inputs. Thus it should come as a surprise that "flat minima" may be a necessary but not sufficient condition on generalisation performance: Dinh et al.[13] prove that **flat minima can be made arbitrarily sharp** by utilising a reparametrisation trick.

Hence it seems that flatness of minima of loss functions alone does not account for generalisation. However, it is possible to reconcile these two opposite arguments in a Bayesian framework; see Section 3.5. Although the reconciliation is possible, I personally do not think flatness is a fundamental characteristic of well-behaved minima. As pointed out by Liang et al.[38], *geometric invariance* could play a central role in capacity control, and if some features ("flatness" for example) are not preserved under certain geometric transformation then they should not be the defining factor.

3.4 The Role of Regularisation

Zhang et al.'s work also concerns roles explicit (data augmentation, weight decay or ℓ_2 regulariser, dropout[51]) and implicit regularisation (early stopping, batch normalisation[29], stochastic gradient descent) play in deep neural networks. They find that although properly-tuned regularisation can *potentially* improve generalisation error, it is neither necessary nor by itself sufficient: impact of a simple change of model easily outweighs implementation of the most sophisticated regularisation scheme, and networks with regularisers removed still generalise well. On the other hand, the addition of regularisation terms (in some cases) fails to prevent the networks from overfitting random labels. **These observations are in sharp contrast to convex optimisation problems, where absence of regularisation in models with more free parameters than training samples necessitates overfitting and failure to generalise.** We will discuss related ideas in this subsection.

3.4.1 Origin of Regularisation: Tikhonov Regulariser

Tikhonov regularisation[56] was first proposed to resolve *ill-posed* linear problems where the matrix to be inverted is (almost) singular, resulting in stability issues. Suppose we want to solve the linear system

$$X\beta = Y \tag{3}$$

the Moore-Penrose pseudoinverse is given by $\beta = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$. However, if the matrix X is nearly singular, the solution will become highly sensitive to noise in X. To fix this problem, we can introduce noise terms in the linear model. If we assume the noise has constant variance (a Bayesian *prior*), we arrive at the ℓ_2 regularisation: $\beta = (X^{\mathrm{T}}X + \sigma^2 I)^{-1}X^{\mathrm{T}}Y$, and the impact of singular vectors with small singular values are suppressed.

Another old-school regulariser, ℓ_1 (sparse regularisation), sets small coefficients straight to zero. It is used in methods such as LASSO regression[55]. These traditional regularisers reduce effective degrees of freedom in a model in the context of convex optimisation, thus avoiding overfitting.

3.4.2 Regularisation in Deep Neural Networks

Apart from ℓ_2 and ℓ_1, some new forms of regularisation are introduced to deep neural network training. Those include early stopping, drop-out, batch normalisation etc., and they differ from the old-school methods in many ways: they have nothing to do with weight stability; they tend not to reduce network degrees of freedom **significantly** (although one may argue early stopping is approximately equivalent to ℓ_2 regularisation[19], and batch normalisation works by reducing degrees of freedom in each layer slightly); and they cannot be associated with any Bayesian prior in any obvious way.

Researchers have also taken notice of the commonly used training algorithm, stochastic gradient descent, and question why SGD training almost always leads to local minima that generalise well[6]. One approach[7][24] utilises uniform stability, an algorithm-dependent analysis introduced in Section 2.1.3. Another attempt is to connect SGD training to "flat minima" discussed in Section 3.3. Specifically, Keskar et al.[30] argue with the support of empirical evidence that small-batch SGD training consistently converges to flat minimisers. Their observation is affirmed by Zhang et al.[63], Welling & Teh[59], and Smith & Le[50], who make analogy with the stochastic Langevin equation describing Brownian motions and prove that noise in the stochastic differential equation drives SGD away from sharp minima. Moreover, Smith & Le[50] prove that there is an "optimal batch size" for SGD that is proportional to both the learning rate and the size of the training set.

3.5 A Bayesian Perspective

The phenomenon observed by Zhang et al.[62] is in fact not unique to deep neural networks. Smith & Le[50] show that the same result happens in a simple logistic regression model as well. We will interpret the result using Bayesian evidence[39], and argue how Dinh et al.[13]'s finding can be reconciled with previous works on "flat minima".

Consider a classification problem with training input $X = \{x_i\}_{i=1}^N$, training output $Y = \{y_i\}_{i=1}^N$ and a classifier M parametrised by ω. The likelihood $P(Y|\omega, X; M) = \prod_i P(y_i|\omega, x_i; M) = e^{-H(\omega;M)}$ where $H(\omega; M) = -\sum_i \ln\left(P(y_i|\omega, x_i; M)\right)$ is the cross-entropy. If we assume a Gaussian prior $P(\omega; M) = \sqrt{\lambda/2\pi}e^{-\lambda\omega^2/2}$, the posterior distribution is given by Bayes' theorem

$$P(\omega|Y, X; M) \propto P(Y|\omega, X; M)P(\omega; M) \propto \sqrt{\lambda/2\pi}e^{-C(\omega;M)} \tag{4}$$

where $C(\omega; M) = H(\omega; M) + \lambda\omega^2/2$ denotes the ℓ_2 regularised cross-entropy. We can find ω_0 that minimises the cross-entropy which in turn gives the maximum of $P(\omega|Y, X; M)$.

Bayesian Model Comparison We want to compare two models M_1 and M_2 based on our training data. We compare them by evaluating the probability ratio

$$\frac{P(M_1|Y,X)}{P(M_2|Y,X)} = \frac{P(Y|X;M_1)}{P(Y|X;M_2)}\frac{P(M_1)}{P(M_2)} \tag{5}$$

We normally set the second term on the right hand side to 1 since we should not have any prior preference for models. The first ratio on the right hand side is the evidence ratio that determines how the training data changes our prior belief. We can estimate the ratio via Taylor expansion and arrive at

$$P(Y|X;M) \approx \exp\left\{-(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P}\ln\left(\lambda_i/\lambda\right))\right\} \tag{6}$$

where P is the dimension of the parameter ω, and λ_i are the eigenvalues of the Hessian $|\nabla^2 C(\omega)|$ at $\omega = \omega_0$. Sharp minima have larger eigenvalues λ_i (thus smaller probability), hence those models are less plausible than models with flat minima.

Dinh et al.'s work applies reparametrisation to the model parameter and regularisation parameter alike, thus the changes in $\ln(\lambda_i/\lambda)$ cancel out exactly. We can indeed make flat minima arbitrarily sharp, but the process does not alter the Bayesian evidence, therefore having no impact on generalisation.

4 Statistical Physics and Neural Networks

(This section is mainly inspired by Martin & Mahoney[42]'s work. Please note this section mainly relies on drawing analogy and sketching ideas, and includes unproved claims aplenty.)

4.1 Hopfield Networks and Spin Glass Transition

The study of neural networks for *associative memory*[26] started in the 1960s, and over the course of two decades many models for associative memory have been designed, e.g. Amari[2], Grossberg[23], Kohonen[31]. In the 1970s physicists discovered that there is in fact a connection between neural networks and magnetic systems; specifically, the *Hopfield network*[28] (a special case of the *Ising model*) for associative memory is linked to the *spin glass*[44] model that explains ferromagnetic frustration in condensed matter physics.

The Hopfield network consists of a collection of connected two-state (excitatory and inhibitory) neurons. The network is characterised by its Hamiltonian (energy functional), and training of the network is equivalent to minimising the (free) energy. An important parameter of the network is its *temperature*. When the temperature is high, the system is in an *ergodic* state: a state where given sufficient time, it is possible to reach any point in the energy landscape. However, as the temperature approaches zero, ergodicity is broken: the energy of the system is confined to the nearest saddle point and cannot escape due to the energy barrier. A *phase transition* happens at zero temperature, and we call the resultant phase the spin glass phase.

Although not rigorous, we can draw an analogy between the spin glass model and deep neural networks[42][11]. We can think of training in deep neural networks as minimising the free energy of high-temperature spin glasses. As illustrated by Figure 1[41], the

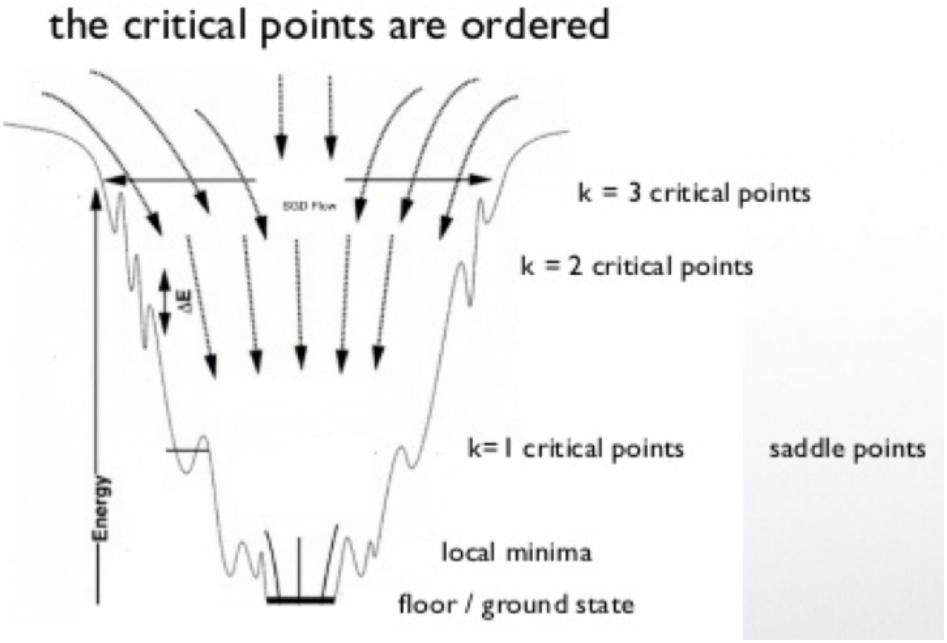

Figure 1: Energy landscape of spin-glass

optimisation is effectively convex, and convergence to local minima close to the global minimum, or the global minimum itself (which corresponds to overtraining), is guaranteed. Therefore, optimisation is easy and deep neural networks do not suffer from only converging to local minima!

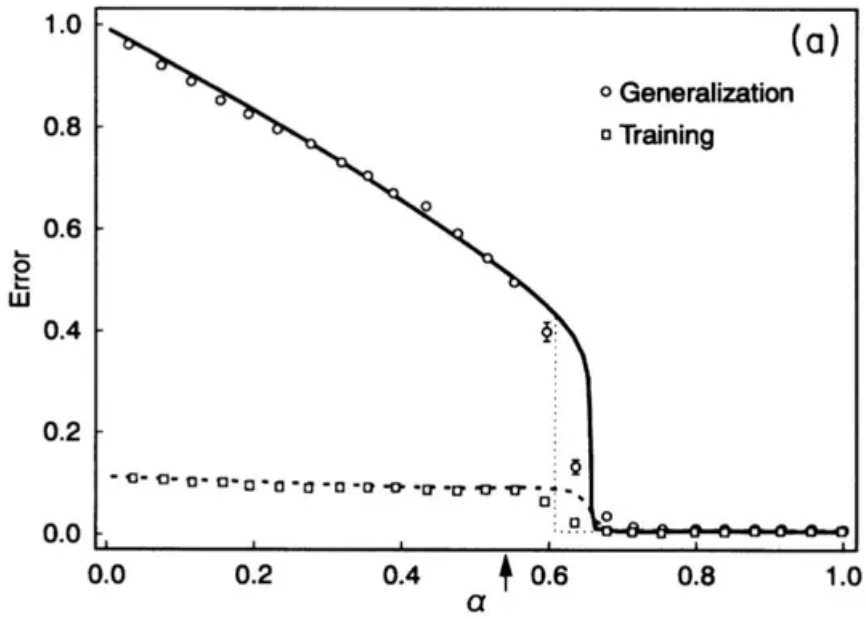

Figure 2: Learning curve of a Hopfield network and cut-off point.

Zhang et al.[62]'s discovery with respect to randomised labels can also be explained. Figure 2[49] shows the learning curve of a Hopfield network with low temperature. As we can see, a phase transition happens at a cut-off point characterised by $\alpha = N/M$, where N is the number of samples and M the number of parameters in the network. If we view

randomising labels as effectively reducing the number of training samples, a phase transition to the glassy state characterised by low training error and high generalisation error is bound to happen at some point.

4.2 The Renormalisation Group

Readers with physics background are also encouraged to study works done by Mehta & Schwab[43] and Oprisa & Toth[48], where the connection between deep learning (*Reduced Boltzmann Machine* as an example) and important concept in high-energy and condensed matter physics, namely the *Renormalisation Group*[60], is established. The renormalisation group is essentially a powerful tool for viewing and relating systems at different scales; I believe its connection to deep learning is no coincidence, but due to the "geometric invariance" mentioned in Section 3.3.

5 Adversarial Training

Adversarial training has been a popular topic of research, especially after the invention of *generative adversarial network*[20] by Goodfellow et al. in 2014. In this section, we give a brief introduction to GAN, and some *adversarial examples* in computer vision, automatic speech recognition and the physical world. The fact that the most state-of-the-art neural network architecture that achieve "surpassing-human-level performance"[25] on certain tasks are still extremely susceptible to adversarial attacks propels us to ask: what is the root cause of this phenomenon? And what is the real difference between deep neural networks' "learning" and that of humans? Unfortunately, neither question is satisfyingly answered yet.

5.1 Generative Adversarial Networks

Prior to the invention of GAN, most successful models in deep learning were *discriminative* models that map the input high-dimensional data to a class label. *Generative* models were hard to train since the training would inevitably entail intractable computations of density estimation. In the adversarial training framework, the generative model is instead trained against a discriminative model that determines if a particular sample comes from the *true* data distribution or the generative model distribution. Training continues until samples given by the generative model are indistinguishable from the true data distribution (from the discriminative model's perspective).

Both models are easy to train if they are multilayer perceptrons. Suppose we have two MLPs represented by $G(\boldsymbol{z};\theta_g)$ and $D(\boldsymbol{x};\theta_d)$ with parameters θ_g and θ_d, where $D(\boldsymbol{x})$ gives the probability that $\boldsymbol{x}$ comes from the true data rather than generative model distribution. Here $\boldsymbol{x}$ is from the data, and the prior distribution on input noise is $p_{\boldsymbol{z}}(\boldsymbol{z})$. The two models are trained simultaneously, to maximise the probability of assigning correct labels to samples from the real data and G, and to maximise chances of G deceiving D (or equivalently, minimising $\log\left(1 - D(G(\boldsymbol{z}))\right)$). It can be posed as the following optimisation problem

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log\left(1 - D(G(\boldsymbol{z}))\right)] \tag{7}$$

which can be solved with stochastic gradient descent.

A striking success of GANs lies in the area of transfer learning, where features of one dataset is extracted and applied to another dataset, e.g. changing the "style" of an image[54] and tones of human voices in an audio recording. From humans' perspective, the ability to complete these tasks reminds us of a fuzzy word: creativity. But as we will see from the next section, it is probably something else.

5.2 Adversarial Examples and Implications

Adversarial training can also cause the exact opposite of generalisation. *Adversarial examples* are samples of input data that are altered slightly in a way intended to "trick" a discriminative model. It is known that convolutional neural networks can be easily fooled by adversarial examples, misclassifying them as completely unrelated items with extreme confidence[53][47]. Worse still, computer vision is not the only affected area; a recent work proposes a method to produce adversarial waveforms that can deceive automatic speech recognisers[9]. Even in the physical world where classifiers read images through cameras instead of being fed data directly, adversarial attacks still prevail[35].

Adversarial examples pose a serious security threat, even more so as the society grows increasingly reliant on artificial intelligence: imagine what would happen if the computer vision system of autonomous vehicles is deceived. Some progress has been made on understanding the nature of adversarial examples[21][17] and constructing more robust models against attacks[40][16], but we are still far away from comprehending the role that complex geometry of real-world data play, and obtaining a truly resistant architecture. Personally I regard vulnerability to adversarial attacks as an inherent weakness of deep neural networks, and one of the fundamental differences between them and human brains.

6 Conclusion

We illustrate to readers some seemingly mysterious properties of deep neural networks, and try to offer some findings, both theoretical and empirical, descriptive and progressively prescriptive, to justify their outstanding generalisation performance. Even though substantial progress has been made in this area in recent years, a unified framework that gives rise to a truly prescriptive theory elucidating the generalisation capability of deep neural networks is still beyond our reach. Using adversarial training as an example, we attempt to convince readers that understanding generalisation truly has real-world implications.

Apart from statistical mechanics, researchers are bringing in tools from other branches of mathematics, e.g. algebraic geometry and differential geometry. I advise passionate readers to check Watanabe's[58] and Amari's book[1] for more information.

References

[1] S. Amari. *Information Geometry and Its Applications*. Springer-Verlag, 2016.

[2] S. Amari and K. Maginu. *Neural Networks*, 1:63, 1988.

[3] Peter L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2), 1998.

[4] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv:1706.08498*, 2017.

[5] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, March 2003.

[6] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *NIPS*, 2008.

[7] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv:1606.04838*, 2016.

[8] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, March 2002.

[9] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv:1801.01944*, 2018.

[10] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *ICLR*, 2017.

[11] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *AISTATS*, 2015.

[12] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv:1609.03193*, 2016.

[13] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv:1703.04933*, 2017.

[14] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

[15] E. Edgington and P. Onghena. *Randomization Tests*. Statistics: A Series of Textbooks and Monographs. Taylor and Francis, 2007.

[16] Akram Erraqabi, Aristide Baratin, Yoshua Bengio, and Simon Lacoste-Julien. A3t: Adversarially augmented adversarial training. *https://arxiv.org/abs/1801.04055*, 2018.

[17] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv:1801.02774*, 2018.

[18] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *https://arxiv.org/abs/1712.06541*, 2018.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2017.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.

[21] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[22] Alex Graves, Abdel rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *IEEE Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[23] S. Grossberg. *Neural Networks*, 1:17, 1988.

[24] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv:1509.01240*, 2016.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv:1502.01852*, 2015.

[26] D. O. Hebb. *The Organization of Behavior*. Wiley, 1949.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, pages 1–42, 1997.

[28] J. J. Hopfield and D.W. Tank. *Science*, 233:625, 1986.

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.

[30] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.

[31] T. Kohonen. *Self Organization and Associative Memory*. Springer-Verlag, 1984.

[32] Alex Krizhevsky and Geoffrey E. Hinton. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.

[34] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S. Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. *ICLR workshop*, 2017.

[35] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2017.

[36] John Langford and Rich Caruana. (not) bounding the true error. *NIPS*, 2001.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. *Nature*, 521:436, 2015.

[38] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. *arXiv:1711.01530*, 2017.

[39] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083*, 2017.

[41] Charles H. Martin. Why deep learning works: perspectives from theoretical chemistry.

[42] Charles H. Martin and Michael W. Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. *arXiv:1710.09553*, 2017.

[43] Pankaj Mehta and David J. Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv:1410.3831*, 2014.

[44] M. Mezard, G. Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, 1987.

[45] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv:1707.09564*, 2017.

[46] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *ICLR workshop*, 2015.

[47] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CVPR*, 2015.

[48] Dan Oprisa and Peter Toth. Criticality and deep learning ii: Momentum renormalisation group. *arXiv:1705.11023*, 2017.

[49] H. S. Seung and H. Sompolinsky. Statistical mechanics of learning from examples. *Physical Review A*, 45(8), April 1992.

[50] Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. *arXiv:1710.06451*, 2017.

[51] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[52] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

[53] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, 2014.

[54] Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, and Kiyoshi Tanaka. Learning a generative adversarial network for high resolution artwork synthesis. *arXiv:1708.09533*, 2017.

[55] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[56] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill Posed Problems*. Wiley, 1977.

[57] Vladimir N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications and control. Wiley, 1998.

[58] Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009.

[59] Max Welling and Yee W. Teh. Bayesian learning via stochastic gradient langevin dynamics. *ICML*, 2011.

[60] Kenneth G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Reviews of Modern Physics*, 47, 1975.

[61] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, and Maxim Krikun et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

[62] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

[63] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Brando Miranda, Noah Golowich, and Tomaso Poggio. Theory of deep learning iib: Optimization properties of sgd. *arXiv:1801.02254*, 2018.