Natural Language Engineering (2022), **28**, pp. 673–682
doi:10.1017/S1351324922000365

CAMBRIDGE UNIVERSITY PRESS

EMERGING TRENDS

Emerging trends: Deep nets thrive on scale

Kenneth Ward Church

Northeastern University, San Jose, CA, USA
E-mail: Kenneth.Ward.Church@gmail.com, k.church@northeastern.edu

(Received 22 July 2022; revised 24 July 2022)

Abstract

Deep nets are becoming larger and larger in practice, with no respect for (non)-factors that ought to limit growth including the so-called curse of dimensionality (CoD). Donoho suggested that dimensionality can be a blessing as well as a curse. Current practice in industry is well ahead of theory, but there are some recent theoretical results from Weinan E's group suggesting that errors may be independent of dimensions *d*. Current practice suggests an even stronger conjecture: deep nets are not merely immune to CoD, but actually, deep nets thrive on scale.

Keywords: Deep nets; Curse of dimensionality; Blessing; Model size; Scale

1. Introduction

1.1 Deep nets: Method of choice

Following Devlin *et al.* (2019), deep nets have become the method of choice for a number of tasks in:

1. Natural language: fill-mask, question answering, sentence similarity, summarization, text classification, text generation, token classification, translation
2. Audio: audio classification, audio-to-audio, automatic speech recognition, text-to-speech (speech synthesis)
3. Computer vision: image classification, image segmentation, object detection.

There are dozens/thousands of models on HuggingFace for each of these tasks.[a] There are also many secondary sources on deep nets and machine learning including:

1. text books (Bishop 2016; Goodfellow *et al.*, 2016),
2. more practical books (Géron 2019; Chollet 2021),
3. surveys[b] (LeCun *et al.*, 2015; Pouyanfar *et al.*, 2018; Kumar *et al.*, 2018; Liu *et al.*, 2020; Qiu *et al.*, 2020; Dong *et al.*, 2021), and
4. tutorials: ACL-2022 (Church *et al.*, 2022a), plus three articles in the Emerging Trends column in this journal (Church *et al.*, 2021b, 2021a, 2022b).

[a] https://huggingface.co/tasks
[b] https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

Table 1. Deep nets are becoming larger and larger over time.

Year	Deep nets	Billions of parameters
2016	ResNet-50 (He *et al.*, 2016)	0.023
2019	BERT (Devlin *et al.*, 2019)	0.34
2019	GPT-2 (Radford *et al.*, 2019)	1.5
2020	GPT-3 (Brown *et al.*, 2020; Dale 2021)	175
2022	PaLM (Chowdhery *et al.*, 2022)	540

1.2 Growth is out of control

As illustrated in Table 1, deep nets are becoming larger and larger (for better or for worse). There have been dramatic increases in size over time, where size can be measured in a variety of different ways:

1. model size, m (number of parameters),
2. number of dimensions, d (problem size),
3. size of (annotated and unannotated) training data,
4. staff (authors per paper),
5. hardware (number of CPUs, GPUs, TPUs, and data centers), and
6. costs (including externalities such as global warming).

The consensus in industry, at least for practical applications, is that bigger nets are better, especially if one cares about performance on test sets (and little else). It is not clear why bigger is better, or even that it is, or that it is a good thing,[c] though there are quite a few blogs on this topic. Social media may not be a good way to judge consensus; social media can easily become an echo chamber with multiple blogs promoting the same paper (Bubeck and Sellke 2021).[d,e]

Most of these larger models are coming from industry; training large models has become too expensive for academia (Bommasani *et al.*, 2021). A recent model from Google, PaLM (Chowdhery *et al.*, 2022), for example, produces impressive results, but the size of the investment is, perhaps, even more impressive. The model was trained on thousands of TPUs, too many for a single data center. The PaLM paper has dozens of authors. Suffice it to say: PaLM is big in every imaginable way.

PaLM's contribution is more in Systems Research than Computational Linguistics. It is an amazing engineering and logistical feat to make productive use of so much hardware. Before PaLM, efficiency had been declining. Previous attempts to scale training tended to increase waste (idle time). PaLM not only produced a larger net (at greater expense), but perhaps more importantly, they found a more effective approach to scaling, opening a path toward even larger (and even more expensive) nets, coming soon to an application on a phone near you.

2. Factors that ought to limit growth

There are a number of factors that one might expect to limit growth on deep nets. Of course, none of these factors matter, as evidenced by the fact that growth is out of control. This paper will focus on the last (non)-factor: the so-called curse of dimensionality (CoD).

[c] https://bdtechtalks.com/2019/11/25/ai-research-neural-networks-compute-costs/
[d] https://www.quantamagazine.org/computer-scientists-prove-why-bigger-neural-networks-do-better-20220210/
[e] https://dataconomy.com/2022/06/bigger-neural-networks-do-better/

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

1. budget constraints (including externalities such as global warming),
2. constraints imposed by deployment platforms (such as phones),
3. availability of (annotated and unannotated) training data (including externalities such as concerns for invisible workers),
4. overfitting, and
5. the CoD.

2.1 Budget constraints and global warming

Budgets are not unlimited, of course, even in industry. Industry would not train such large (and expensive) nets without compelling motivations to do so.

Budget constraints include capital and expense, as well as less obvious factors such as carbon emissions. Following Strubell *et al.* (2019),[f] there have been concerns about accounting and externalities. One might hope that more appropriate taxes on carbon emissions would discourage industry from training larger and larger nets, though we have our doubts. The cost of training is a one-time upfront cost. If a net is used by millions of users every day for years, then recurring costs (inference) dominate one-time costs (training). As a result, it has become standard practice in industry to train larger and larger nets but reduce costs just in time with compression methods such as distillation (DistilBERT)[g,h] (Sanh *et al.*, 2019). Compression makes it possible to train larger nets but reduce costs before deployment (where costs matter).

2.2 Deployment platforms (phones)

The same just-in-time compression technology will become important as nets migrate to phones. In the past, most of the computation tended to be in the cloud (at the center of the network), but more and more computation will likely migrate to edge devices (phones). Given resource limitations on phones (power, memory, CPU/GPU cycles), one might expect this migration to limit growth.[i] However, the same compression methods mentioned above are also being used to address these migration issues. Thus, constraints on deployment (cost, power, etc.) have important consequences for compression technology, but less so for training, where out-of-control growth is likely to continue for the foreseeable future.

2.3 Training data are not unlimited

Another non-factor mentioned above is training data. There are many use cases with limitations on training data. Annotated data are particularly expensive, both in monetary terms and other terms (impacts on invisible workers).[j] Because of concerns such as these, there is more and more interest in prompting (Liu *et al.*, 2021), zero-shot learning and few-shot learning. These methods reduce demand for annotated (and unannotated) training data. However, many of the most successful such methods make use of extremely large pretrained models such as GPT-3 and PaLM. Thus, it is unlikely that limitations on annotated (and unannotated) training data will stem the out-of-control growth (for pretrained models).

[f] https://www.technologyreview.com/2022/07/06/1055458/ai-research-emissions-energy-efficient/
[g] https://huggingface.co/models?sort=downloads&search=distil
[h] https://paperswithcode.com/method/distillbert
[i] https://huggingface.co/models?sort=downloads&search=mobile
[j] https://www.technologyreview.com/2020/12/11/1014081/ai-machine-learning-crowd-gig-worker-problem-amazon-mechanical-turk/

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

2.4 Overfitting

With many traditional methods, such as regression, if we have too many parameters, we are likely to overfit the training set. A number of traditional methods for addressing overfitting will be mentioned in Section 3 such as feature selection and regularization. It is widely believed that deep nets do not suffer from overfitting, even when heavily over-parameterized.[k] Deep nets have developed methods such as stochastic gradient descent with random restarts. There are a few theoretical suggestions that such methods are effective (Li and Liang 2018) and over-parameterization does not lead to overfitting (Brutzkus *et al.*, 2018; Allen-Zhu *et al.*, 2019; Oymak and Soltanolkotabi 2020).

Overfitting can be viewed as a special case of the CoD.

2.5 Curse of dimensionality (COD)

One might expect the CoD to limit out-of-control growth. Donoho (2000) provides some hints why this might not be the case. He introduces a novel perspective, suggesting that large d (dimensions) can be both a blessing and a curse. Donoho starts with Bellman's original argument (Bellman 1966). Bellman's argument introduced the term, CoD, to motivate his work on dynamic programming:

> *Bellman reminded us that, if we consider a cartesian grid of spacing 1/10 on the unit cube in 10 dimensions, we have* 10^{10} *points; if the cube in 20 dimensions was considered, we would have of course* 10^{20} *points. His interpretation: if our goal is to optimize a function over a continuous product domain of a few dozen variables by exhaustively searching a discrete search space defined by a crude discretization, we could easily be faced with the problem of making tens of trillions of evaluations of the function. Bellman argued that this curse precluded, under almost any computational scheme then foreseeable, the use of exhaustive enumeration strategies, and argued in favor of his method of dynamic programming.* (Donoho 2000)

How can CoD be a blessing? While it may be easy to find examples where large d causes trouble, there are also examples where large d is a blessing. Donoho calls out concentration of measure (Ledoux 2001) as one of the better examples of a blessing.

> *The concentration of measure phenomenon in product spaces roughly states that, if a set A in a product* Ω^N *of probability spaces has measure at least one half, "most" of the points of* Ω^N *are "close" to A.* (Talagrand 1995)

This observation has many applications. Donoho, for example, uses the concentration of measure in a widely cited paper on compressed sensing (Donoho 2006).

Blessings show up in many practical applications. Consider web search, for example. Web search is more effective in larger networks. Enterprise search can be frustrating. Why is it easier to find good stuff on the web than on a small website for a company or a university? Larger networks can be a blessing because there are more links to what you are looking for in larger communities. Page rank (Page *et al.*, 1999), for example, has more dynamic range on larger graphs.

Under Metcalfe's Law,[l] larger graphs have advantages because edges scale faster than vertices. A telephone network is a popular example of Metcalfe's Law. The cost of adding another cell phone to the network is a constant, but the benefits scale with the number of other phones already in the network. It is said that Metcalfe's Law makes it hard for second movers to challenge an establish incumbent with a dominant position in the market. When benefits scale with edges and costs scale with vertices, then the rich get richer.

[k] https://medium.com/mlearning-ai/intuitive-explanations-of-why-over-parameterised-deep-nets-dont-overfit-8a323b223ba6

[l] https://en.wikipedia.org/wiki/Metcalfe%27s_law

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

There are many examples of methods that thrive on scale such as approximate nearest neighbors[m] (Indyk and Motwani 1998), random projections (Li *et al.*, 2006), and sketches (a method originally designed to remove near duplicate web pages from large crawls (Broder 2000), but has many generalizations (Li and Church 2007)).

Consider eigenvector and node2vec-like embeddings of graphs (Grover and Leskovec 2016; Zhou *et al.*, 2020). Again, larger graphs have advantages. If we use vectors to represent vertices in a graph (such as a telephone network or web pages), and we estimate similarity of two vertices as a cosine of two vectors, then estimates of similarity improve with larger graphs and longer vectors with more hidden dimensions.

This paper is more concerned with deep nets. Are there reasons to believe that scale could be a blessing for deep nets? Before addressing that question, we will discuss some historical background. Why did we used to believe that scale was a problem?

3. What is our problem with scale?

Researchers today have become comfortable with scale, but we used to feel differently. It is common practice these days to use models with more parameters than observations, especially when discussing topics such as zero-shot and few-shot learning.

It is hard for people from our generation to get used to this new world order. We used to assume, as a matter of faith, that we need more observations than parameters. By Occam's razor, we preferred models with fewer degrees of freedom. At least in the case of regression, if there are too many degrees of freedom, and not enough training data, then regression coefficients will not reach significance. Even when the coefficients reach significance, if there are too many parameters, then overfitting is likely, producing large errors on the test set.

We used to assume that concepts such as degrees of freedom, significance, and feature selection were important for most models under consideration, not just regression. Much has been written on methods to avoid over-parameterization such as feature selection (LeCun *et al.*, 1989; Dash and Liu 1997; Guyon and Elisseeff 2003; Fan and Lv 2010; Kumar and Minz 2014; Li *et al.*, 2017), ANOVA (Scheffe 1999), regularization (Tibshirani 1996; Bickel *et al.*, 2006, 2009), invariant features (Fant 1973; Stevens and Blumstein 1981; Acero and Stern 1991; Lowe 1999; Brown and Lowe 2002), feature engineering (Scott and Matwin 1999), and term weighting (Salton and Buckley 1988). Over-parameterization is problematic for many/most traditional methods (Fan and Lv 2008), though there may be a few exceptions (Bartlett *et al.*, 2020).

These days, hill-climbing is the method of choice for fitting deep nets. But we were warned by our teachers that hill-climbing cannot possibly scale up to problems with large d and rich (non-convex) structure (with many local minima)[n] (Minsky 1961; Minsky and Papert 1969). Bishop rejects our teachers' concerns as "incorrect conjecture" on page 193 (Bishop 2006), but it took the field many decades to appreciate that hill-climbing is feasible in high dimensions, and we are still trying to figure out why that is the case.

It has been claimed that certain methods such as support vector machines (Cortes and Vapnik 1995; Hearst *et al.*, 1998) work relatively well in high dimensional spaces (Joachims 1998), but those technologies did not lead to out-of-control growth like we are seeing for deep nets. Apparently, deep nets are not merely robust to high dimensions, but they thrive on them. How can that be?

4. Weinan E: A mathematical perspective on machine learning

It is generally accepted, at least among practitioners, that deep nets thrive on scale. The big question is: *why*. Is there a theoretical justification for what we are all doing?

[m] https://pypi.org/project/gensim/

[n] Recent work (Kawaguchi 2016; Du *et al.*, 2019) suggests local minima may not be as much a problem as once thought.

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

Weinan E recently gave a theoretical talk on recent progress on somewhat related questions.[o,p] The discussion below will use slide numbers and page numbers to refer to the talk[q] and an overview article (E, 2020), respectively.

This work is very much a work in progress. Currently, their results are better for two-level nets; extensions to multi-layer networks are "unsatisfactory" (slide 37). Of course, much work remains to be done, as explained in a paper with a brutally honest title that ends with: *what we know and what we don't* (E *et al.*, 2020). Actually, the theory community has a long tradition of sharing lists of promising open problems with their students. Our field would have less (pointless) SOTA-chasing (Church and Kordoni 2022) if we produced more brutally honest papers like this. Such papers help students find good projects to work on.

Weinan E's talk is divided into three sections:

1. Introduction (slides 1–20): Apparently, deep nets are better than alternatives (polynomials) in high dimensions
2. Theoretical discussion of errors
   (a) Approximation error, E_a (slides 21–37; pp. 17–19): errors due to the choice of the hypothesis space
   (b) Estimation error, E_e (slides 38–45; pp. 19–21): additional errors due to finiteness of data
   (c) Optimization error, E_o (slides 46–54): additional errors caused by training
3. Applications of deep nets to solve problems in high dimensions (slides 56–70).

The discussion of errors starts with an example of the CoD (slide 13). Suppose we want to approximate a function f^* with f_m using a classical method such as piecewise linear functions over a mesh of size h. Assuming $h \sim m^{1/d}$, where d is the dimensionality of the problem and m is the size of the model (in terms of free parameters), then computational costs grow exponentially with d. That is, errors, $E = |f^* - f_m|$, scale in a nasty way with d:

$$E = |f^* - f_m| \sim h^2 |\nabla^2 f^*| \sim m^{-2/d} |\nabla^2 f^*| \tag{1}$$

Thus, to reduce the error by a factor of 10, we need to increase m by a factor of $10^{d/2}$.

Weinan E concludes with the observation:

Compared with polynomials, neural networks provide a much more effective tool for approximating functions in high dimension. (slide 68)

Similar comments probably hold for regression-like methods and other traditional methods that are being replaced by neural nets.

The crux of Weinan E's talk is to replace grid-based methods with Monte Carlo estimation. Slide 24 suggests that newer methods (based on Monte Carlo estimation) have error rates that are independent of dimensions d, in contrast to older methods (slide 23) based on uniform grids.

Much of the Weinan E's talk attempts to make this intuition more rigorous and more general. The discussion on slides 23 and 24 is specific to a number of particulars: a particular type of error (approximation error), and a particular type of network (a two-layer network), and a particular method on a particular grid.

They are making great progress, with many recent promising results, and there will be more results in the future. The discussion of approximation errors, E_a, is relatively long (16 slides on E_a versus 7 slides on E_e and 8 slides on E_o), suggesting there has been more progress on approximation errors.

[o] https://www.youtube.com/watch?v=xjQ8PcIMrf8
[p] https://www.youtube.com/watch?v=VwIj3d6Lp24
[q] http://web.math.princeton.edu/~weinan/ICM-update.pdf

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

The three types of errors are defined on slide 20. Weinan E splits the error, $E = |f^* - \hat{f}|$, into three sub-errors, $E = E_a + E_e + E_o$, by introducing two milestones between f^* and $\hat{f}$:

1. milestone f_m: best approximation of f^* in a hypothesis space, $\mathcal{H}$
2. milestone $\tilde{f}_{n,m}$: best approximation of f_m using only the dataset, S.

The three sub-errors defined in terms of these milestones:

- Approximation error: gap from goal, f^*, to first milestone,
- Estimation error: gap between two milestones, and
- Optimization error: gap from last milestone to approximation, $\hat{f}$.

The discussion of these errors mentions CoD, but in different ways:

1. Approximation error (slide 23; on p. 17): CoD is challenging for grid-based approximation methods, where errors scale in a nasty way with d.
2. Estimation error (slide 43; p. 18): size of training data grows exponentially quickly with d.
3. Optimization error (slide 48): convergence rate for gradient-based training algorithms must suffer from CoD.

One of the more exciting results is a bound on approximation errors that is independent of d (E 2022) (slide 34; p. 19). This result establishes that certain types of nets are immune to CoD, though there is some fine-print. This result is currently limited to two-layer nets, and it only covers one type of error (approximation errors).

5. Conclusions

As mentioned above, current practice is well ahead of theory; some even compare what we do to alchemy (Church and Liberman 2021).[r] Over the last few years, industry has been producing deep nets that are bigger in every imaginable way (for better and for worse): model size (m), dimensions (d), cost, training data, staff, hardware, carbon emissions, negative impacts on invisible workers, etc. Recent progress on the theory side is not able to keep up with practice in industry, but there are some exciting theoretical results suggesting that approximation errors for two-layer nets may be independent of d. There will likely be more progress on the theory side, relaxing much of the fine-print.

Eventually, theory will catch up to practice and explain why it makes sense for industry to do what it is doing. Theory is on a path toward explaining why deep nets might be immune to the CoD, but the out-of-control growth suggests a more bullish conjecture: deep nets are succeeding because of scale (West 2018), not in spite of scale.

References

Acero A. and **Stern R. M.** (1991). Robust speech recognition by normalization of the acoustic space. *In ICASSP*, vol. **91**, pp. 893–896.

Allen-Zhu Z., **Li Y.** and **Liang Y.** (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*, vol. 32.

Bartlett P. L., **Long P. M.**, **Lugosi G.** and **Tsigler A.** (2020). Benign overfitting in linear regression. *Proceedings of The National Academy of Sciences of The United States of America* **117**, 30063–30070.

Bellman R. (1966). Dynamic programming. *Science* **153**, 34–37.

[r] https://www.youtube.com/watch?v=x7psGHgatGM

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

Bickel P. J., **Li B.**, **Tsybakov A. B.**, **van de Geer S. A.**, **Yu B.**, **Valdés T.**, **Rivero C.**, **Fan J.** and **van der Vaart A.** (2006). Regularization in statistics. *Test* **15**, 271–344.

Bickel P. J., **Ritov Y.** and **Tsybakov A. B.** (2009). Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics* **37**, 1705–1732.

Bishop C. (2016). *Pattern Recognition and Machine Learning. Information Science and Statistics*. New York: Springer.

Bishop C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Bommasani R., **Hudson D. A.**, **Adeli E.**, **Altman R.**, **Arora S.**, **von Arx S.**, **Bernstein M. S.**, **Bohg J.**, **Bosselut A.**, **Brunskill E.**, **Brynjolfsson E.**, **Buch S.**, **Card D.**, **Castellon R.**, **Chatterji N.**, **Chen A.**, **Creel K.**, **Davis J. Q.**, **Demszky D.**, **Donahue C.**, **Doumbouya M.**, **Durmus E.**, **Ermon S.**, **Etchemendy J.**, **Ethayarajh K.**, **Fei-Fei L.**, **Finn C.**, **Gale T.**, **Gillespie L.**, **Goel K.**, **Goodman N.**, **Grossman S.**, **Guha N.**, **Hashimoto T.**, **Henderson P.**, **Hewitt J.**, **Ho D. E.**, **Hong J.**, **Hsu K.**, **Huang J.**, **Icard T.**, **Jain S.**, **Jurafsky D.**, **Kalluri P.**, **Karamcheti S.**, **Keeling G.**, **Khani F.**, **Khattab O.**, **Kohd P. W.**, **Krass M.**, **Krishna R.**, **Kuditipudi R.**, **Kumar A.**, **Ladhak F.**, **Lee M.**, **Lee T.**, **Leskovec J.**, **Levent I.**, **Li X. L.**, **Li X.**, **Ma T.**, **Malik A.**, **Manning C. D.**, **Mirchandani S.**, **Mitchell E.**, **Munyikwa Z.**, **Nair S.**, **Narayan A.**, **Narayanan D.**, **Newman B.**, **Nie A.**, **Niebles J. C.**, **Nilforoshan H.**, **Nyarko J.**, **Ogut G.**, **Orr L.**, **Papadimitriou I.**, **Park J. S.**, **Piech C.**, **Portelance E.**, **Potts C.**, **Raghunathan A.**, **Reich R.**, **Ren H.**, **Rong F.**, **Roohani Y.**, **Ruiz C.**, **Ryan J.**, **Ré C.**, **Sadigh D.**, **Sagawa S.**, **Santhanam K.**, **Shih A.**, **Srinivasan K.**, **Tamkin A.**, **Taori R.**, **Thomas A. W.**, **Tramèr F.**, **Wang R. E.**, **Wang W.**, **Wu B.**, **Wu J.**, **Wu Y.**, **Xie S. M.**, **Yasunaga M.**, **You J.**, **Zaharia M.**, **Zhang M.**, **Zhang T.**, **Zhang X.**, **Zhang Y.**, **Zheng L. Zhou K.** and **Liang P.** (2021) On the opportunities and risks of foundation models. *Center for Research on Foundation Model*. arXiv:2108.07258v3.

Broder A. Z. (2000). Identifying and filtering near-duplicate documents. In *Annual Symposium on Combinatorial Pattern Matching*. Springer, pp. 1–10.

Brown M. and **Lowe D. G.** (2002). Invariant features from interest point groups. In *BMVC*, vol. **4**, pp. 398–410.

Brown T. B., **Mann B.**, **Ryder N.**, **Subbiah M.**, **Kaplan J.**, **Dhariwal P.**, **Neelakantan A.**, **Shyam P.**, **Sastry G.**, **Askell A.**, **Agarwal S.**, **Herbert-Voss A.**, **Krueger G.**, **Henighan T.**, **Child R.**, **Ramesh A.**, **Ziegler D. M.**, **Wu J.**, **Winter C.**, **Hesse C.**, **Chen M.**, **Sigler E.**, **Litwin M.**, **Gray S.**, **Chess B.**, **Clark J.**, **Berner C.**, **McCandlish S.**, **Radford A.**, **Sutskever I.** and **Amodei D.** (2020). *Language Models are Few-Shot Learners*. NeurIPS.

Brutzkus A., **Globerson A.**, **Malach E.** and **Shalev-Shwartz S.** (2018). SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*.

Bubeck S. and **Sellke M.** (2021). A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems* **34**, 28811–28822.

Chollet F. (2021). *Deep Learning with Python*. Manning Publications Co.: Shelter Island, NY.

Chowdhery A., **Narang S.**, **Devlin J.**, **Bosma M.**, **Mishra G.**, **Roberts A.**, **Barham P.**, **Chung H. W.**, **Sutton C.**, **Gehrmann S.**, **Schuh P.**, **Shi K.**, **Tsvyashchenko S.**, **Maynez J.**, **Rao A.**, **Barnes P.**, **Tay Y.**, **Shazeer N.**, **Prabhakaran V.**, **Reif E.**, **Du N.**, **Hutchinson B.**, **Pope R.**, **Bradbury J.**, **Austin J.**, **Isard M.**, **Gur-Ari G.**, **Yin P.**, **Duke T.**, **Levskaya A.**, **Ghemawat S.**, **Dev S.**, **Michalewski H.**, **Garcia X.**, **Misra V.**, **Robinson K.**, **Fedus L.**, **Zhou D.**, **Ippolito D.**, **Luan D.**, **Lim H.**, **Zoph B.**, **Spiridonov A.**, **Sepassi R.**, **Dohan D.**, **Agrawal S.**, **Omernick M.**, **Dai A. M.**, **Pillai T. S.**, **Pellat M.**, **Lewkowycz A.**, **Moreira E.**, **Child R.**, **Polozov O.**, **Lee K.**, **Zhou Z.**, **Wang X.**, **Saeta B.**, **Diaz M.**, **Firat O.**, **Catasta M.**, **Wei J.**, **Meier-Hellstern K.**, **Eck D.**, **Dean J.**, **Petrov S.** and **Fiedel N.** (2022). Palm: Scaling language modeling with pathways.

Church K., **Chen Z.** and **Ma Y.** (2021a). Emerging trends: A gentle introduction to fine-tuning. *Natural Language Engineering* **27**, 763–778.

Church K., **Kordoni V.**, **Marcus G.**, **Davis E.**, **Ma Y.** and **Chen Z.** (2022a). A gentle introduction to deep nets and opportunities for the future. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 1–6.

Church K. and **Liberman M.** (2021). The future of computational linguistics: On beyond alchemy. *Frontiers in Artificial Intelligence* **4**, 625341.

Church K. W., **Cai X.**, **Ying Y.**, **Chen Z.**, **Xun G.** and **Bian Y.** (2022b). Emerging trends: General fine-tuning (gft). *Natural Language Engineering*, 1–17.

Church K. W. and **Kordoni V.** (2022). Emerging trends: SOTA-chasing. *Natural Language Engineering* **28**(2), 249–269.

Church K. W., **Yuan X.**, **Guo S.**, **Wu Z.**, **Yang Y.** and **Chen Z.** (2021b). Emerging trends: Deep nets for poets. *Natural Language Engineering* **27**, 631–645.

Cortes C. and **Vapnik V.** (1995). Support-vector networks. *Machine Learning* **20**, 273–297.

Dale R. (2021). GPT-3: what's it good for? *Natural Language Engineering* **27**, 113–118.

Dash M. and **Liu H.** (1997). Feature selection for classification. *Intelligent Data Analysis* **1**, 131–156.

Devlin J., **Chang M.-W.**, **Lee K.** and **Toutanova K.** (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota: Association for Computational Linguistics, vol. **1**, pp. 4171–4186.

Dong S., **Wang P.** and **Abbas K.** (2021). A survey on deep learning and its applications. *Computer Science Review* **40**, 100379.

Donoho D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *Aide-Memoire of a Lecture at AMS Conference on Math Challenges of the 21st Century*.

Donoho D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory* **52**, 1289–1306.

Du S., **Lee J.**, **Li H.**, **Wang L.** and **Zhai X.** (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*. PMLR, pp. 1675–1685.

E W. (2020). Machine learning and computational mathematics. arXiv preprint arXiv: 2009.14596.

E W., **Ma C.**, **Wojtowytsch S.** and **Wu L.** (2020). Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *CoRR*, abs/2009.10713.

E W., **Ma C.** and **Wu L.** (2022). The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation* **55**(1), 369–406.

Fan J. and **Lv J.** (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(5), 849–911.

Fan J. and **Lv J.** (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica* **20**, 101–148.

Fant G. (1973). Speech sounds and features.

Géron A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc.

Goodfellow I., **Bengio Y.** and **Courville A.** (2016). *Deep Learning*. MIT Press.

Grover A. and **Leskovec J.** (2016). node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864.

Guyon I. and **Elisseeff A.** (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**, 1157–1182.

He K., **Zhang X.**, **Ren S.** and **Sun J.** (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Hearst M. A., **Dumais S. T.**, **Osuna E.**, **Platt J.** and **Scholkopf B.** (1998). Support vector machines. *IEEE Intelligent Systems and Their Applications* **13**, 18–28.

Indyk P. and **Motwani R.** (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pp. 604–613.

Joachims T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*. Springer, pp. 137–142.

Kawaguchi K. (2016). Deep learning without poor local minima.*Advances in Neural Information Processing Systems*29.

Kumar A., **Verma S.** and **Mangla H.** (2018). A survey of deep learning techniques in speech recognition. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, pp. 179–185.

Kumar V. and **Minz S.** (2014). Feature selection: A literature review. *SmartCR* **4**, 211–229.

LeCun Y., **Bengio Y.** and **Hinton G.** (2015). Deep learning. *Nature* **521**, 436–444.

LeCun Y., **Denker J.** and **Solla S.** (1989). Optimal brain damage, *Advances in Neural Information Processing Systems*2.

Ledoux M. (2001). *The Concentration of Measure Phenomenon*, vol. **89**. American Mathematical Society: Providence, USA.

Li J., **Cheng K.**, **Wang S.**, **Morstatter F.**, **Trevino R. P.**, **Tang J.** and **Liu H.** (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* **50**, 1–45.

Li P. and **Church K. W.** (2007). A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics* **33**, 305–354.

Li P., **Hastie T. J.** and **Church K. W.** (2006). Very sparse random projections. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD. New York, NY, USA: Association for Computing Machinery, vol. **06**, pp. 287–296.

Li Y. and **Liang Y.** (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in Neural Information Processing Systems* **31**.

Liu L., **Ouyang W.**, **Wang X.**, **Fieguth P.**, **Chen J.**, **Liu X.** and **Pietikäinen M.** (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision* **128**, 261–318.

Liu P., **Yuan W.**, **Fu J.**, **Jiang Z.**, **Hayashi H.** and **Neubig G.** (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv: 2107.13586.

Lowe D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, vol. **2**, pp. 1150–1157.

Minsky M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE* **49**, 8–30.

Minsky M. and **Papert S.** (1969). *Perceptron: An Introduction to Computational Geometry*, vol. **19**, expanded ed. Cambridge: The MIT Press, 19, p. 2.

Oymak S. and **Soltanolkotabi M.** (2020). Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory* **1**, 84–105.

Page L., **Brin S.**, **Motwani R.** and **Winograd T.** (1999). The pagerank citation ranking: Bringing order to the web, Technical report, Stanford InfoLab.

https://doi.org/10.1017/S1351324922000365 Published online by Cambridge University Press

Pouyanfar S., **Sadiq S.**, **Yan Y.**, **Tian H.**, **Tao Y.**, **Reyes M. P.**, **Shyu M.-L.**, **Chen S.-C.** and **Iyengar S. S.** (2018). A survey on deep learning: algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)* **51**, 1–36.

Qiu X., **Sun T.**, **Xu Y.**, **Shao Y.**, **Dai N.** and **Huang X.** (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* **63**, 1872–1897.

Radford A., **Wu J.**, **Child R.**, **Luan D.**, **Amodei D.** and **Sutskever I.** (2019). *Language Models are Unsupervised Multitask Learners*. OpenAI Blog.

Salton G. and **Buckley C.** (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management* **24**, 513–523.

Sanh V., **Debut L.**, **Chaumond J.** and **Wolf T.** (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arXiv preprint arXiv: 1910.01108.

Scheffe H. (1999). *The Analysis of Variance*, vol. **72**. John Wiley & Sons.

Scott S. and **Matwin S.** (1999). Feature Engineering for Text Classification. *In ICML*, vol. **99**, pp. 379–388.

Stevens K. N. and **Blumstein S. E.** (1981). The search for invariant acoustic correlates of phonetic features. In Eimas P. D. and Miller J. L. (eds.) *Perspectives on the Study of Speech*, Hillsdale, USA: Lawrence Erlbaum Associates, pp. 1–38.

Strubell E., **Ganesh A.** and **McCallum A.** (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650.

Talagrand M. (1995). Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques* **81**, 73–205.

Tibshirani R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B-Methodological* **58**, 267–288.

West G. (2018). *Scale: The Universal Laws of Life, Growth, and Death in Organisms, Cities, and Companies*. Penguin.

Zhou J., **Cui G.**, **Hu S.**, **Zhang Z.**, **Yang C.**, **Liu Z.**, **Wang L.**, **Li C.** and **Sun M.** (2020). Graph neural networks: A review of methods and applications. *AI Open* **1**, 57–81.

Cite this article: Church KW (2022). Emerging trends: Deep nets thrive on scale. *Natural Language Engineering* **28**, 673–682. https://doi.org/10.1017/S1351324922000365