NỀN TẢNG AI TẠO SINH

(IT5410 – Foundation of Generative AI)

MỘT SỐ VẤN ĐỀ CỦA LÝ THUYẾT HỌC SÂU

Thân Quang Khoát

Trường CNTT&TT, ĐHBKHN

2024

Nội dung

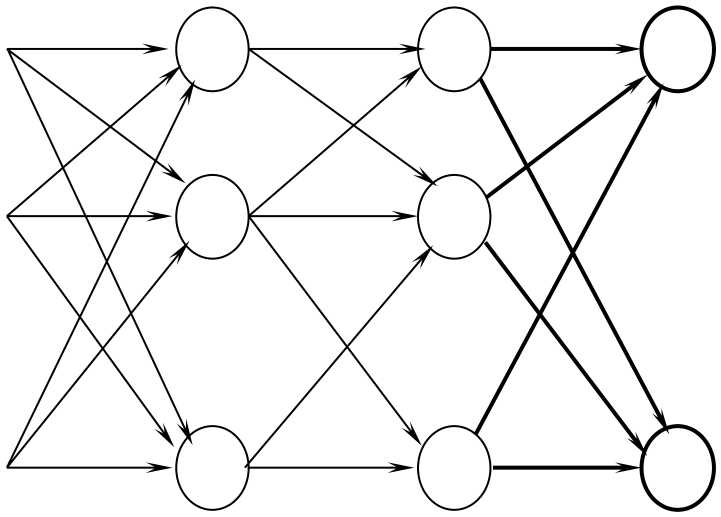
- Mở đầu
- **Một số vấn đề của Học sâu**
- Một số kiến trúc mạng nơron
- Mô hình sinh sâu
- Đánh giá chất lượng
- Học tăng cường

Theoretical results for deep neural networks

A short summary

Neural network

- Artificial neural networks (ANN):
  - Biologically inspired by human brain
  - A rich family to represent complex functions
- An ANN:
  - Consists of many neurons, organized in a layer-wise manner
  - Each *neuron* computes a simple function
  - A neuron can have few *connections* to other neurons
- Each configuration about #neurons, #layers, #connections, … ➔ an architecture
- Shallow vs. Deep NNs:
  - One hidden layer >< many hidden layers



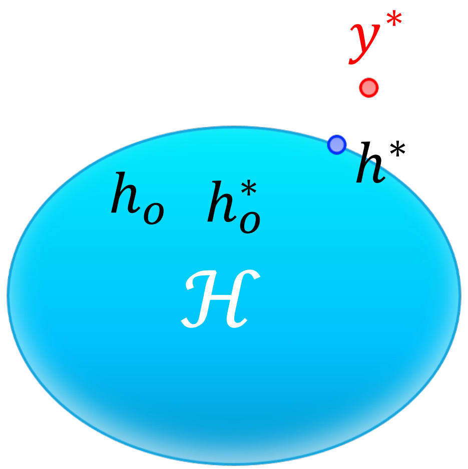
Mathematical description

$$h(\boldsymbol{x}, \boldsymbol{W}) = g_K(\boldsymbol{W}_K h_{K-1}), \qquad \text{where} \quad h_i = g_i(\boldsymbol{W}_i h_{i-1}), \qquad h_0 = \boldsymbol{x}$$

- An NN with K layers
- $\boldsymbol{W}_i$ is the weight matrix at layer i
- h_i is the output of layer i
- g_i is the activation function at layer i

(feedforward network)

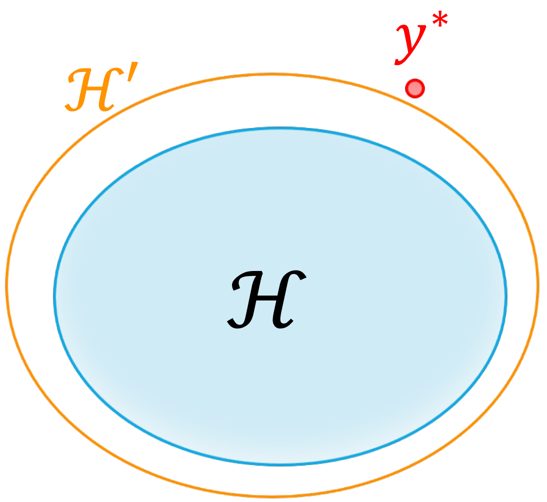
- A NN maps an input $\boldsymbol{x}$ to an output $\mathrm{y} = h(\boldsymbol{x}, \boldsymbol{W})$
- *Training:* often find weights **W**, by minimizing a loss $F(\boldsymbol{D}, h)$



$Error(h_o) \approx$ Optimization error +Generalization error +Approximation error

Approximation error: classical

$$\|y^* - h\| \leq \epsilon_a$$



- Increase capacity ➔ approximate better
  - Larger family $\mathcal{H}'$
  - More complex NNs ➔ stronger representational power
  - E.g., wider or deeper NNs
- Any binary function can be learnt (approximately well) by a feedforward network using one hidden layer, when the **width goes to infinity**
  (bất kỳ hàm nhị phân nào đều có thể học được bởi một mạng lan truyền tiến với một tầng ẩn, khi số lượng nơron ở tầng nào đó tiến ra vô hạn)
- Any bounded continuous function can be learnt (approximately) by a *feedforward network* using one hidden layer [Cybenko, 1989; Hornik, 1991]

Cybenko, G. (1989). Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems.*
Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251-257.

Approximation error: modern

- Any continuous function can be approximated arbitrarily well by Convolutional neural network, when the depth is large [Zhou, 2020]
- Any Lebesgue-integrable function can be approximated arbitrarily well by a ResNet with c
- **Deep** NNs av Lipschitz functions [Pog
  - Shallow NNs cannot
  - To approximate a Lipschitz function (mapping $[0,1]^n$ to $\mathbb{R}$) with error $O(N^{-\sqrt{L}})$, width $\max\{n, 5N+13\}$ and depth $64nL+3$ are sufficient

Universal approximators

Lin, H., & Jegelka, S. (2018). ResNet with one-neuron hidden layers is a universal approximator. *NeurIPS*.
Lu, J., Shen, Z., Yang, H., & Zhang, S. (2021). Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*.
Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., & Liao, Q. (2017). Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*.
Zhou, D. X. (2020). Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*.

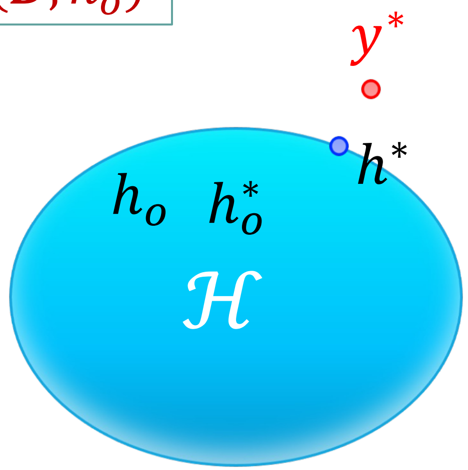
Approximation: existence $\not\to$ method

Unclear
how to find such DNNs,
based on a training set

Optimization error

- Training is often by minimizing a loss $F(\boldsymbol{D}, h)$
- The training loss is *highly non-convex*
- Theory:
  - Exponentially large number of iterations may be needed
  - Intractable in the worst case [Nesterov, 2018]
- Practice:
  - Often have zero training error ➔ global solution h_o^*?
  - Easily perfectly fit random labelling of data [Zhang et al. 2021] (training seems to be easy!)
- Contradiction? What's missing?

$\boxed{F(\boldsymbol{D}, h_o) - F(\boldsymbol{D}, h_o^*)}$



Nesterov, Y. (2018). *Lectures on convex optimization*. Springer.
Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*.

Optimization: theoretically easy

- Gradient descent (GD) achieves zero training loss in polynomial time for a deep over-parameterized ResNet [Du et al. 2019]
  - Over-parameterization: #parameters $\gg$ training size
- GD can find a global optimum when the width of the last hidden layer of an MLP exceeds the number of training samples [Nguyen, 2021]
- Stochastic gradient descent (SGD) can find global minima on the training objective of DNNs in polynomial time [Allen-Zhu et al. 2019]
  - Architecture: MLP, CNN, ResNet

Du, S., Lee, J., Li, H., Wang, L., & Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*.
Nguyen, Q. (2021). On the proof of global convergence of gradient descent for deep relu networks with linear widths. In *International Conference on Machine Learning*.
Allen-Zhu, Z., Li, Y., & Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*.

Optimization: reminder

However

global optimality

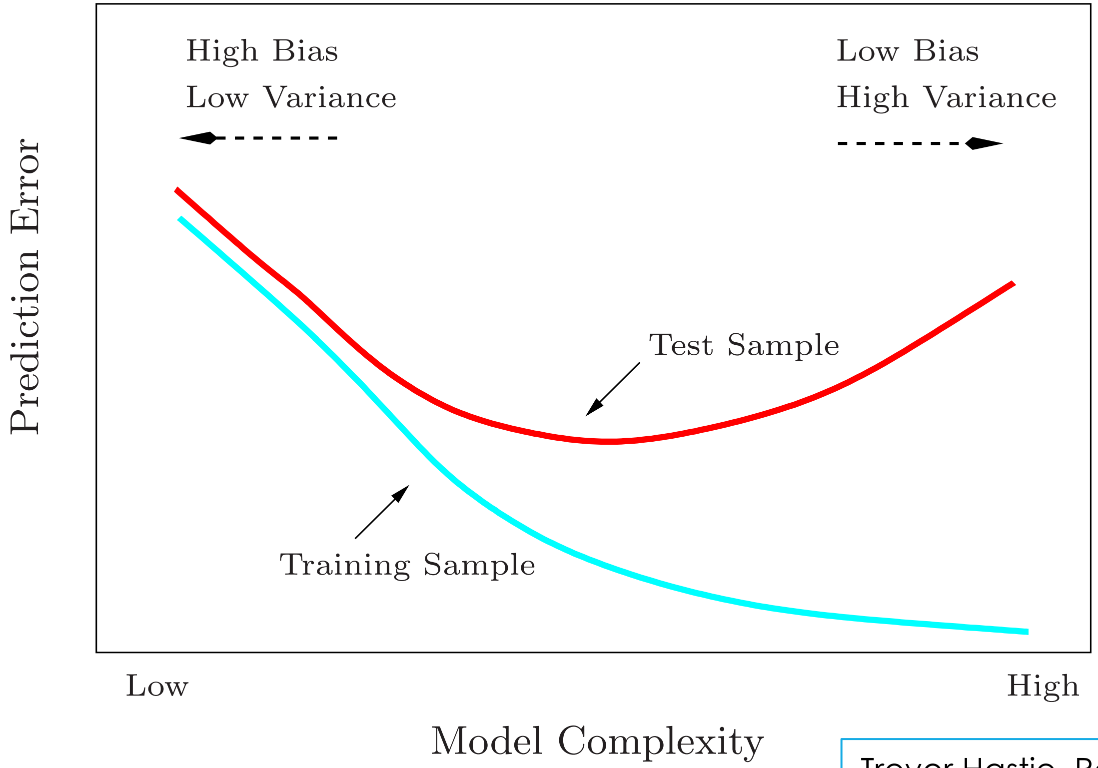
of the training problem

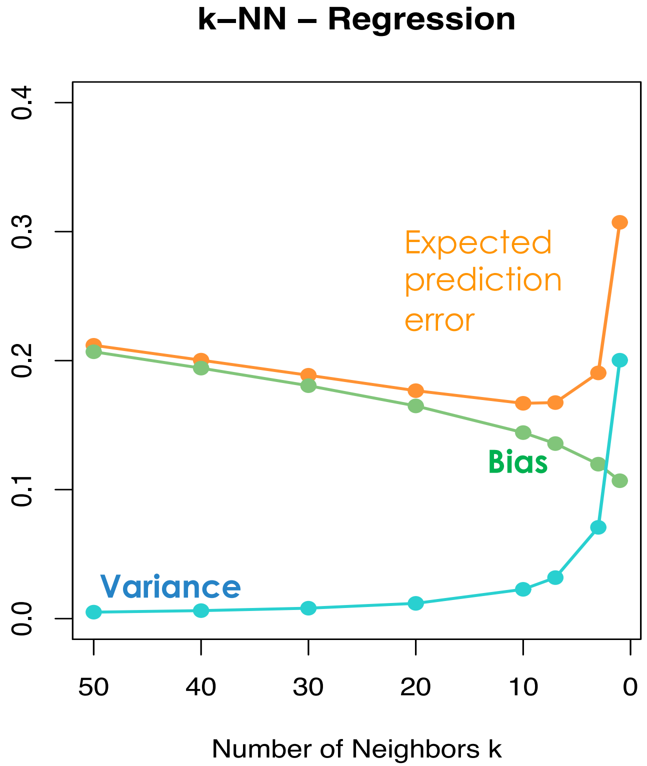
does not imply

good predictive ability

Bias-Variance tradeoff: classical view

- The more complex the model is, the more data points it can capture, and the lower the bias can be
  - However, higher complexity will make the model "move" more to capture the data points, and hence its variance will be larger.





Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

Bias-Variance: modern behavior

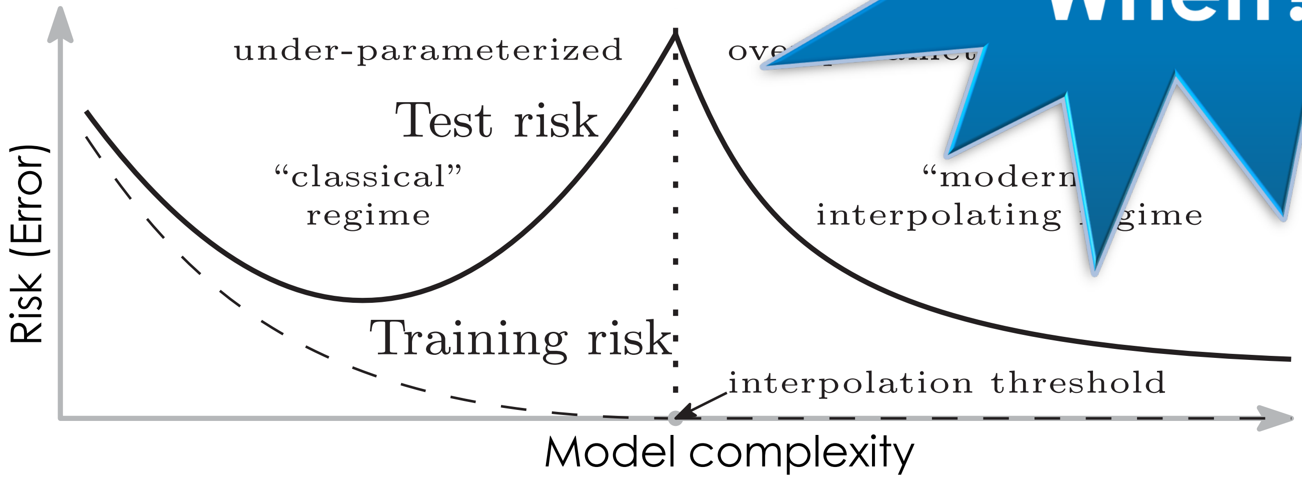
- Modern phenomenon:

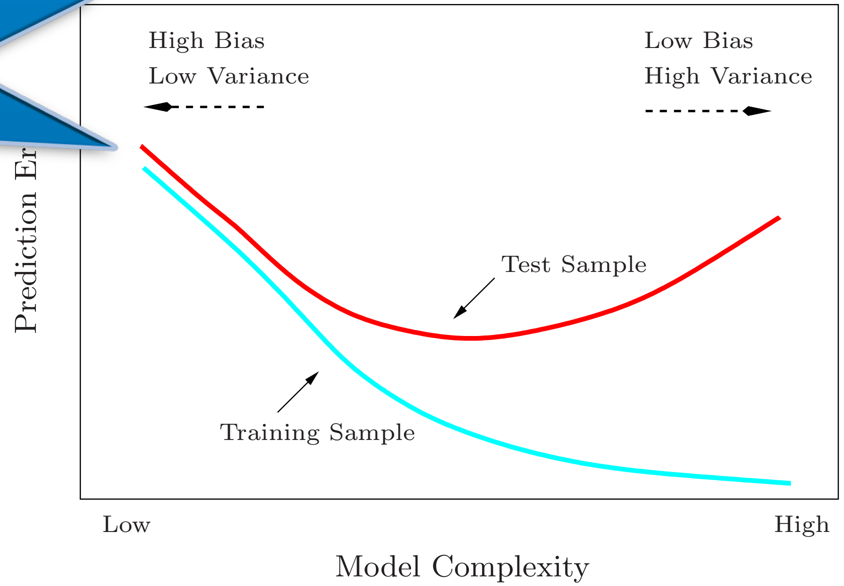
  *Very rich models such as DNNs are trained to **exactly fit** the data, but often obtain **high accuracy** on test data* [Belkin et al., 2019]

  - $Bias \cong 0$
  - GPT-4, ResNets, StyleGAN,

- Classical view:
  more complex model
  - Lower bias, higher variance

Why?
When?





Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849-15854.

Generalization ability: long-standing open

- **Main goal:** small expected loss $F(P, h_o)$
  - Practice: training loss $F(\boldsymbol{D}, h_o) \cong 0$ for overparameterized NNs
- Why can a trained DNN generalize well?
  (Generalization: ability to well perform on unseen data)
- We want to assure, for $\delta > 0$,

$$\Pr(|F(P, h_o) - F(\boldsymbol{D}, h_o)| \leq \epsilon) \geq 1 - \delta$$

  - Generalization gap should be small with a high probability over the random choice of **D**
  - How fast does $F(\boldsymbol{D}, h_o)$ converge to $F(P, h_o)$?
    (as the training size m increases)

$Error(h_o) :=$
Approximation error
+Optimization error
+Generalization error

A long-standing challenge in DL theory

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT press.
Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*.

Generalization: VC dimension

- Vapnik–Chervonenkis (VC) dimension:
  - Measure of the capacity (complexity, expressive power, richness) of a set of functions
  - The cardinality of the largest set of points that the learning algorithm can shatter
  - A higher VC dim ➔ richer model family $\mathcal{H}$
- Example: in n-dimensional space
  - Linear models: $VC(\mathcal{H}) = n + 1$
  - ReLU networks with W weights: $VC(\mathcal{H}) = \Omega(W \log W)$

Bartlett, P. L., Harvey, N., Liaw, C., & Mehrabian, A. (2019). Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*.

- Classical bound: for any $\delta > 0$, with probability at least $1 - \delta$

$$F(P, h) - F(\boldsymbol{D}, h) \leq \sqrt{\frac{2}{m} VC(\mathcal{H}) \log \frac{2e.m}{VC(\mathcal{H})} + \frac{1}{m} \log \frac{2}{\delta}}$$

- Vacuous/meaningless for modern DNNs, due to $W \gg m$ (training size)

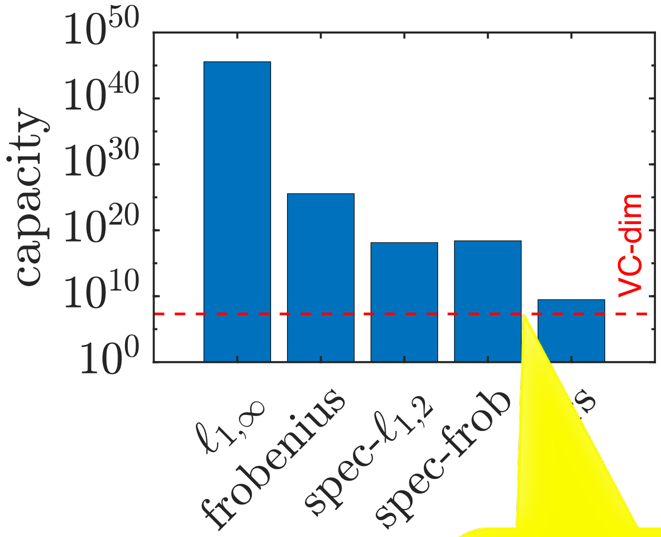
Generalization: Weight norm

- DNN: $h(\boldsymbol{x}, \boldsymbol{W}) = g_K(\boldsymbol{W}_K h_{K-1})$
- Bartlett: **#params** is not important
  - Size of weights may be more important
- Neyshabur et al.; Golowich et al.:

$$F(P, h) - F(\boldsymbol{D}, h) \leq O(\|\boldsymbol{W}_1\|_F \cdots \|\boldsymbol{W}_K\|_F)/\sqrt{m}$$

- Bartlett et al.:

$$F(P, h) - F(\boldsymbol{D}, h) \leq O(\|\boldsymbol{W}_1\|_2 \cdots \|\boldsymbol{W}_K\|_2)/\sqrt{m}$$



Uninformative for modern DNNs

Arora, S., Ge, R., Neyshabur, B., & Zhang, Y. (2018). Stronger generalization bounds for deep nets via a compression approach. In *ICML*.
Bartlett, P. (1998). 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*.
Bartlett, P. L., Foster, D. J., & Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. *Neural Information Processing Systems*.
Golowich, N., Rakhlin, A., & Shamir, O. (2020). Size-independent sample complexity of neural networks. *Information and Inference: A Journal of the IMA*.
Neyshabur, B., Bhojanapalli, S., & Srebro, N. (2018). A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *ICLR*.

Generalization: PAC-Bayes

- Consider $\mathbb{E}_{h\sim\rho}[F(P,h) - F(\boldsymbol{D},h)]$
  - Generalization error on average over $\mathcal{H}$
  - ρ is the posterior distribution of h
- McAllester: with probability at least $1-\delta$

$$\mathbb{E}_{h\sim\rho}[F(P,h) - F(\boldsymbol{D},h)] \le \sqrt{\frac{KL(\rho||\mu) + \log(m/\delta)}{2m-1}}$$

  - μ is the prior distribution of h
  - KL is the Kullback-Leibler divergence

- The "distance" between posterior ρ and prior μ:
  - Plays important role
  - Depends on the *bias* of a learning algorithm
- Unclear how fast can ρ approach μ?
- Do not directly consider the complexity of family $\mathcal{H}$

Meaningful bounds appeared

McAllester, D. A. (2003). PAC-Bayesian stochastic model selection. *Machine Learning*, *51*(1), 5-21.

Generalization: non-vacuous bounds

- We can optimize the PAC-Bayes bound
  - Find the posterior ρ^* that minimizes $KL(\rho||\mu)$
- Dziugaite & Roy: non-vacuous bounds
  - MLP with 3 layers, SGD al rithm, MN ST da et
- Zhou et al.: compressibility ➔
  - Use SOTA compression al bound for ImageNet, LeN
- Lotfi et al., 2022:
  - Propose compression alg. to find onvacuous bounds for LeNet-5, ResNet-18, MobileViT

Stochastic

DNNs

Dataset	Data-independent priors	
	Err. Bound (%)	SOTA (%)
MNIST	**11.6**	21.7 [59]
+ SVHN Transfer	**9.0**	16.1†
FashionMNIST	**32.8**	46.5†
+ CIFAR-10 Transfer	**28.2**	30.1†
CIFAR-10	**58.2**	89.9†
+ ImageNet Transfer	**35.1**	54.2†
CIFAR-100	**94.6**	100†
+ ImageNet Transfer	**81.3**	98.1†
ImageNet	**93.5**	96.5 [73]

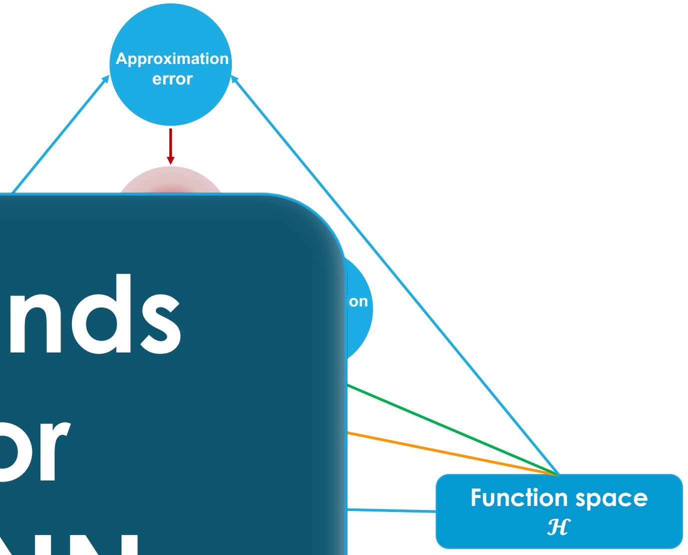
Biggs & Guedj, 2022:

- Non-vacuous bounds for a (special) **deterministic networks**
- MNIST and Fashion-MNIST datasets

Dziugaite, G., & Roy, D. (2017). Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *UAI*.
Zhou, W., Veitch, V., Austern, M., Adams, R., & Orbanz, P. (2019). Non-vacuous Generalization Bounds at the ImageNet Scale: a PAC-Bayesian Compression Approach. In *ICLR*.
Lotfi, S., Finzi, M., Kapoor, S., Potapczynski, A., Goldblum, M., & Wilson, A. G. (2022). PAC-bayes compression bounds so tight that they can explain generalization. *In NeurIPS*.
Biggs, F., & Guedj, B. (2022). Non-vacuous generalisation bounds for shallow neural networks. In *ICML*.

Generalization: long-standing open

- Some other approaches:
  - Neural tangent kernel, Mean field
  - Algorithr



Current meaningful bounds however are mostly for stochastic or shallow NNs

Unclear about Big pretrained models, Deep NNs in practice

Unclear about Why many tricks in DL improve performance