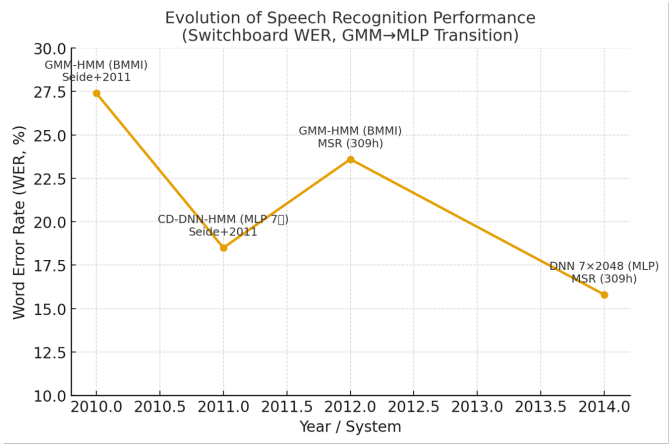
Universal Approximation Theorems for DNNs

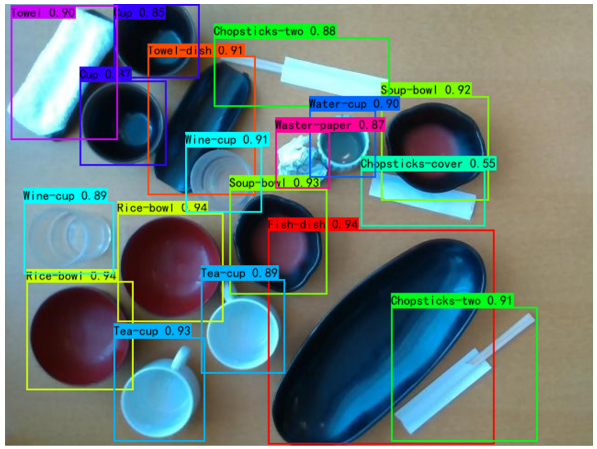
Changhoon Song

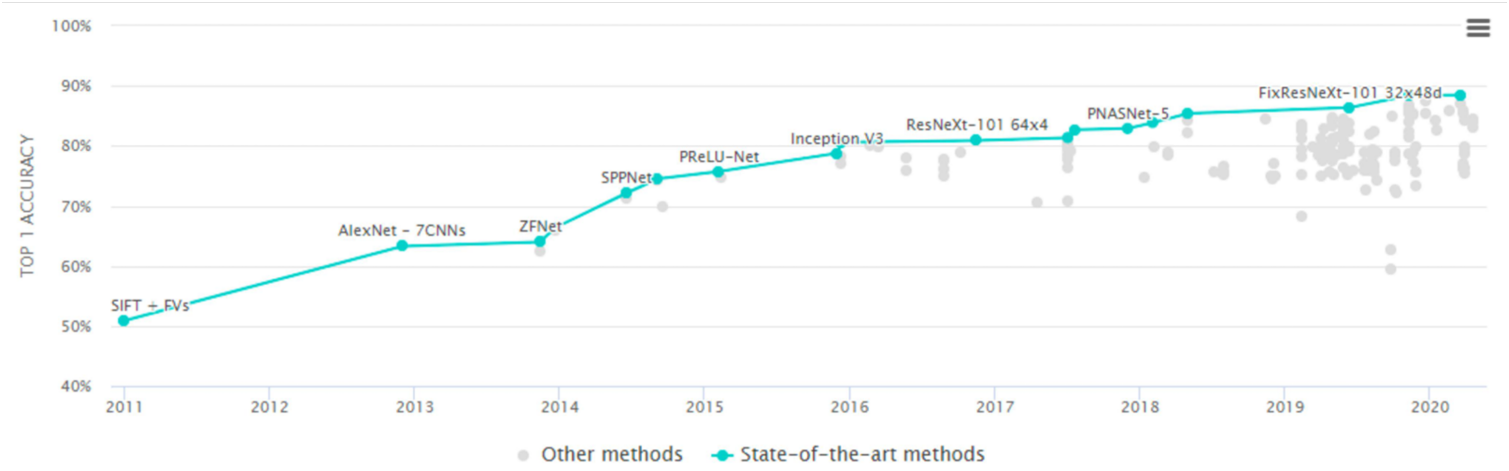
Research Institute of Mathematics

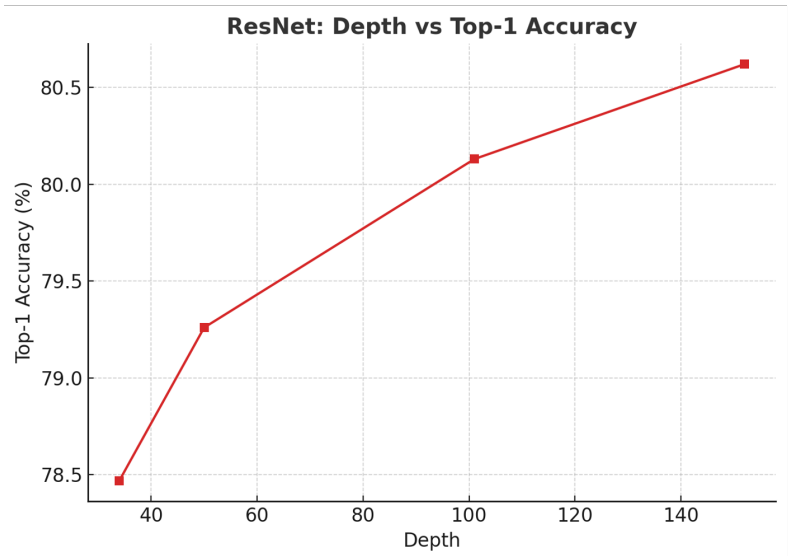
2025. 12. 22.

Index

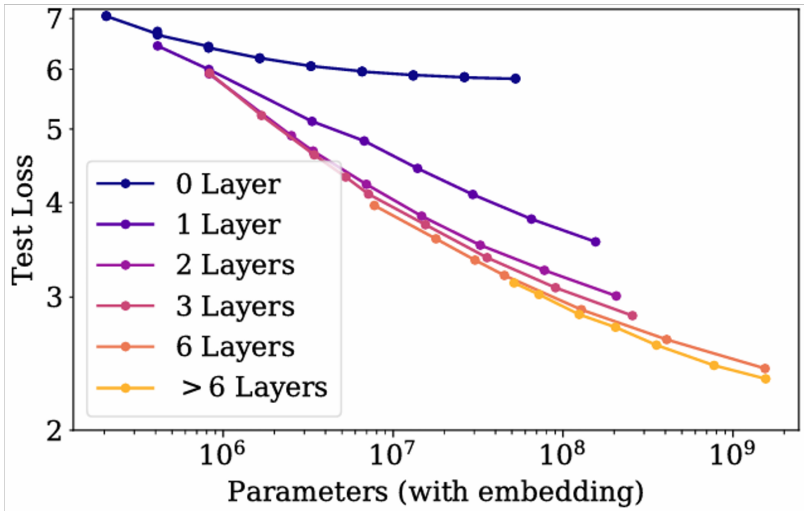

Evolution of Speech Recognition Performance
(Switchboard WER, GMM→MLP Transition)
Word Error Rate (WER, %)
Year / System
GMM-HMM (BMMI)
Seide+2011
GMM-HMM (BMMI)
MSR (309h)
CD-DNN-HMM (MLP 7□)
Seide+2011
DNN 7×2048 (MLP)
MSR (309h)
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
2010.0
2010.5
2011.0
2011.5
2012.0
2012.5
2013.0
2013.5
2014.0


Towel 0.90
Chopsticks-two 0.88
Towel-dish 0.91
Soup-bowl 0.92
Water-cup 0.90
Waster-paper 0.87
Wine-cup 0.91
Chopsticks-cover 0.55
Soup-bowl 0.93
Wine-cup 0.89
Rice-bowl 0.94
Fish-dish 0.94
Rice-bowl 0.94
Tea-cup 0.89
Chopsticks-two 0.91
Tea-cup 0.93


TOP 1 ACCURACY
100%
90%
80%
70%
60%
50%
40%
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
SIFT + FVs
AlexNet - 7CNNs
ZFNet
SPPNet
PReLU-Net
Inception-V3
ResNeXt-101 64x4
PNASNet-5
FixResNeXt-101 32x48d
Other methods
State-of-the-art methods

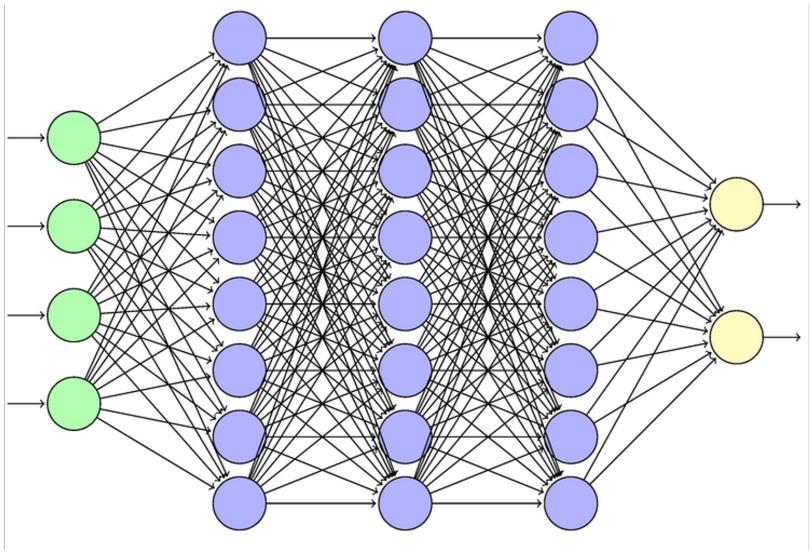


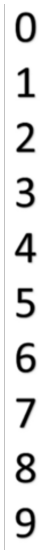
Scaling law in Transformer



Larger architecture $\Longrightarrow$ Better performance.

Q: Can large network do anything?





- Input x: digitized vector in $\mathbb{R}^{d_x}$.

  Image, Tokenized words, Coordinates, etc.

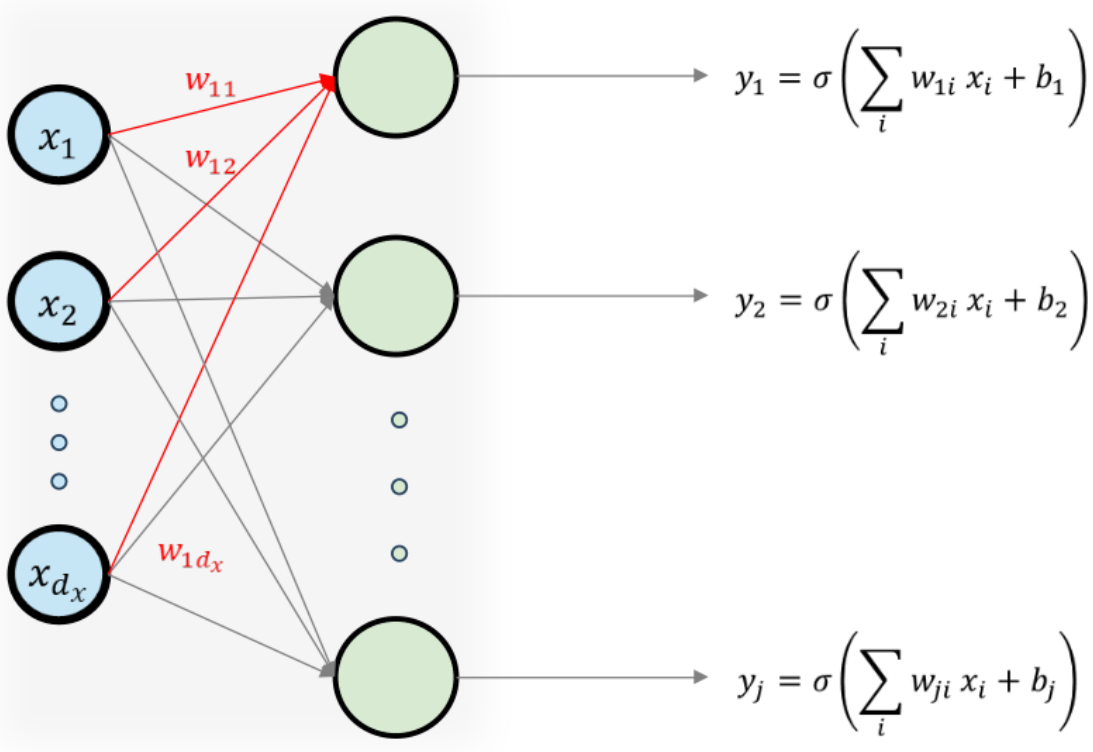
- Output y: digitized vector in $\mathbb{R}^{d_y}$.

  Class label, Tokenized words, Function value, etc.

- Model f: Function $y = f(x)$.

$$x \to z_0, \quad z_\ell \to z_{\ell+1}, \quad z_{L+1} \to y = f(x).$$

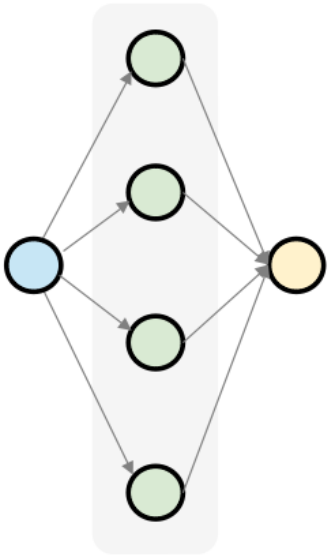
Input

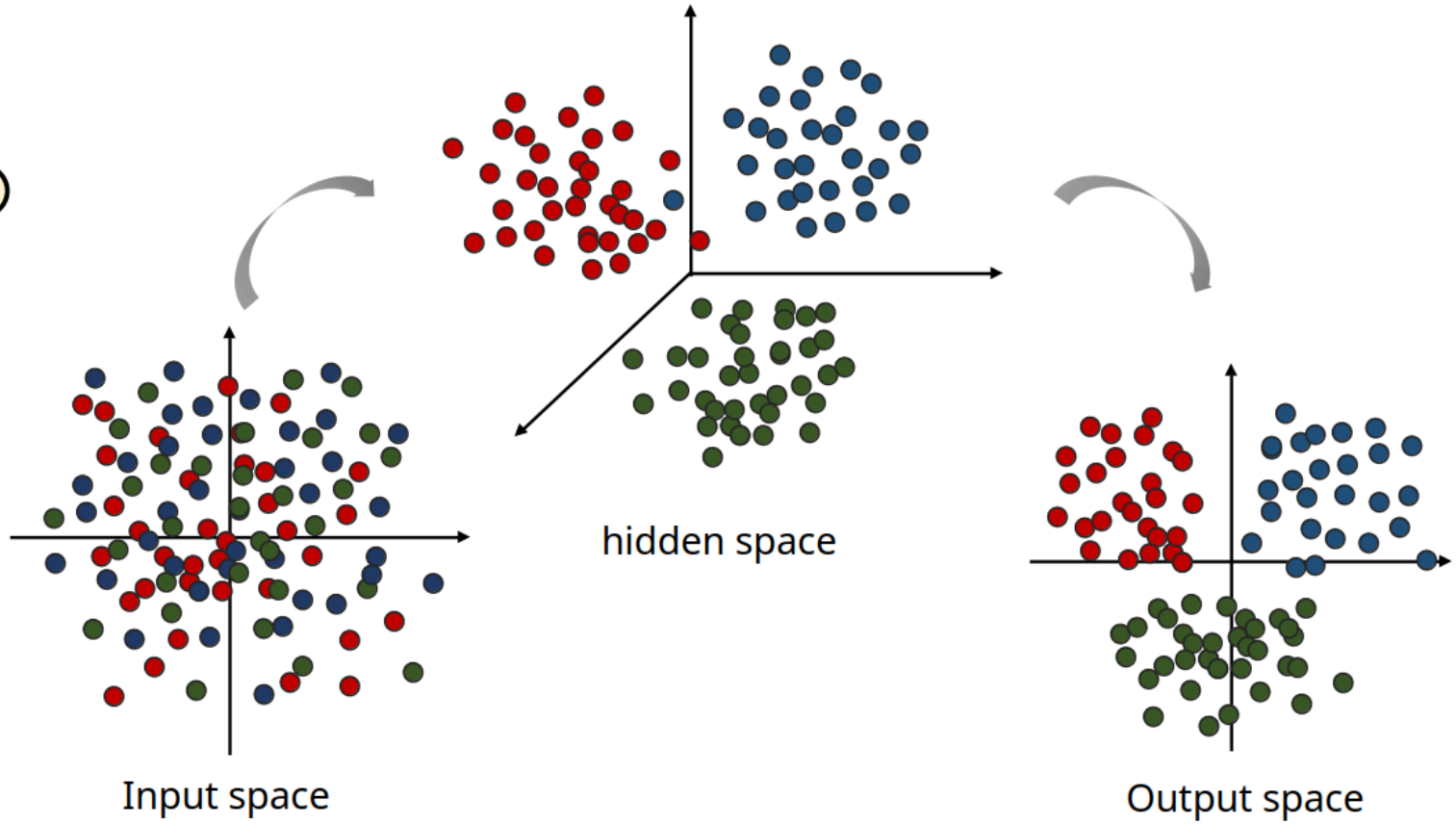


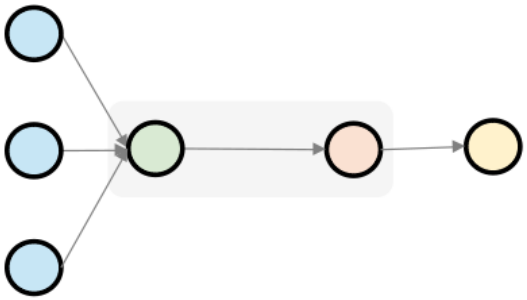
Output

$$y = \sigma(Wx + B)$$

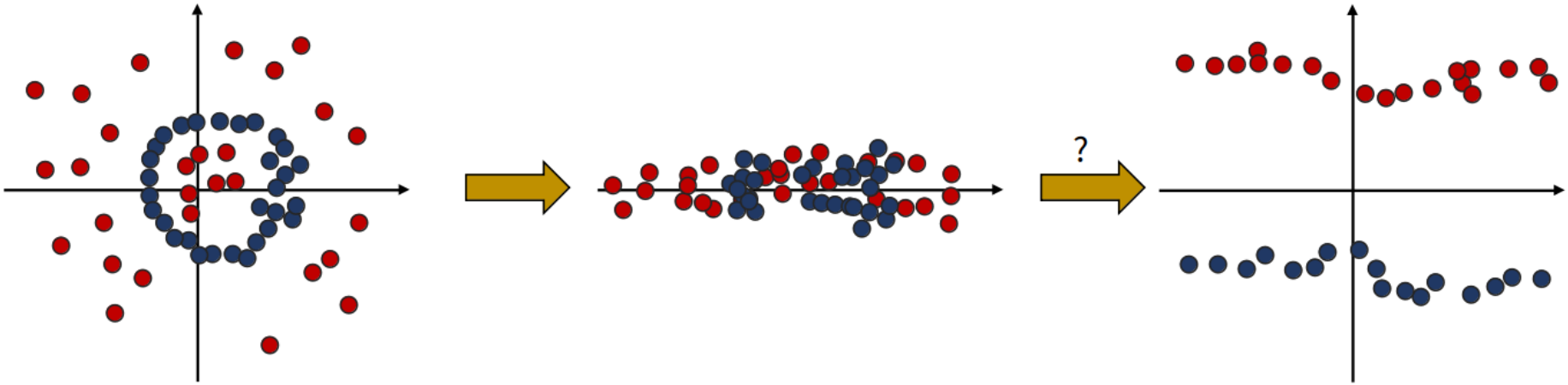
Wide network : embedding to **higher dimension**

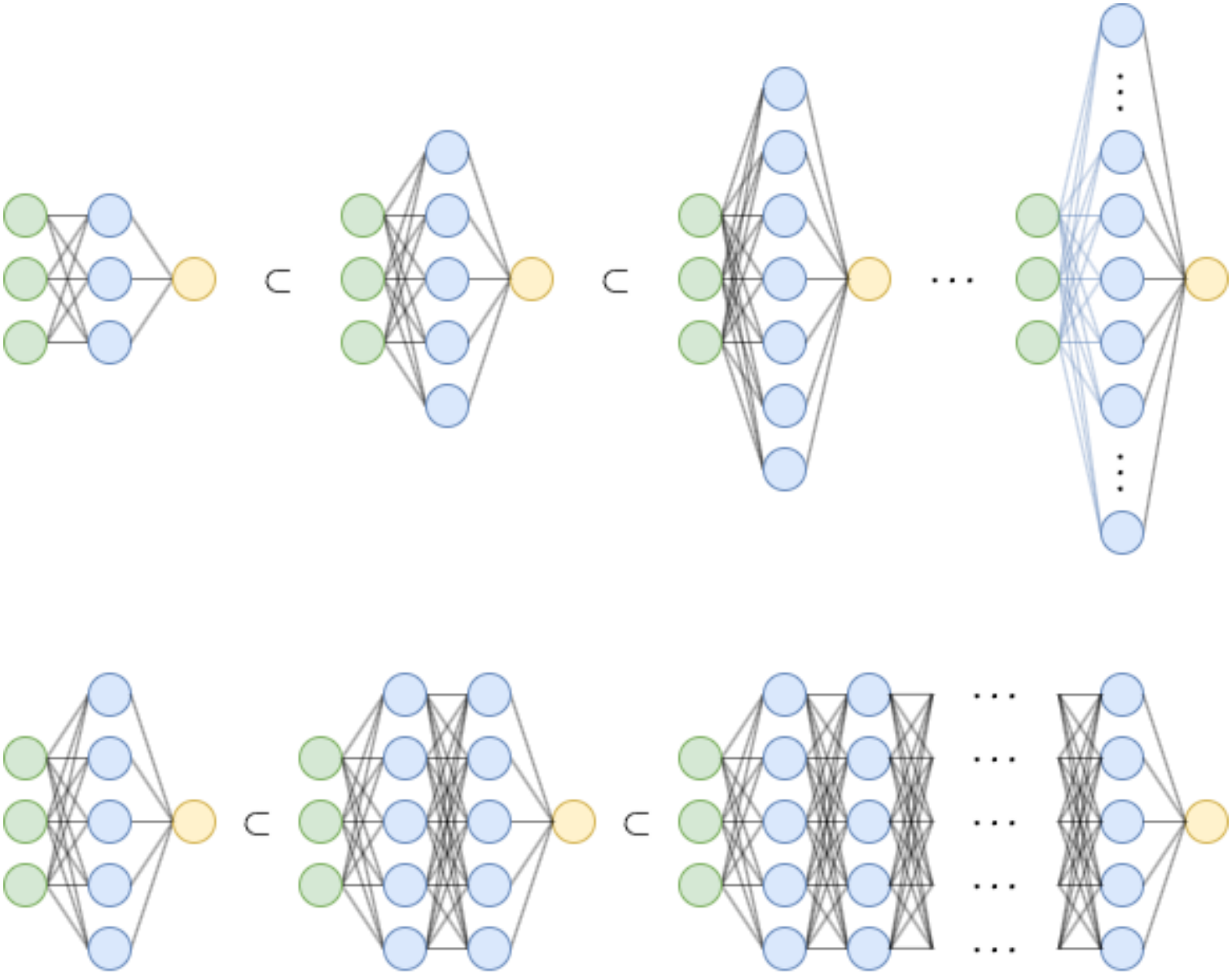






Deep network : applying **multiple transformation**





Topics in Universal Approximation Theorem

Possiblility: For any function, neural network can approximate it.

- Wide Network. (non-constructive proof)
- Deep Network. (constructive proof)
- Approximation Error. (Curse of dimensionality)

Impossiblity: There exists a function neural network cannot approximate under some constraints.

- Wide Network. (with Bounded Weights)
- Deep Network. (with Narrow Widths)

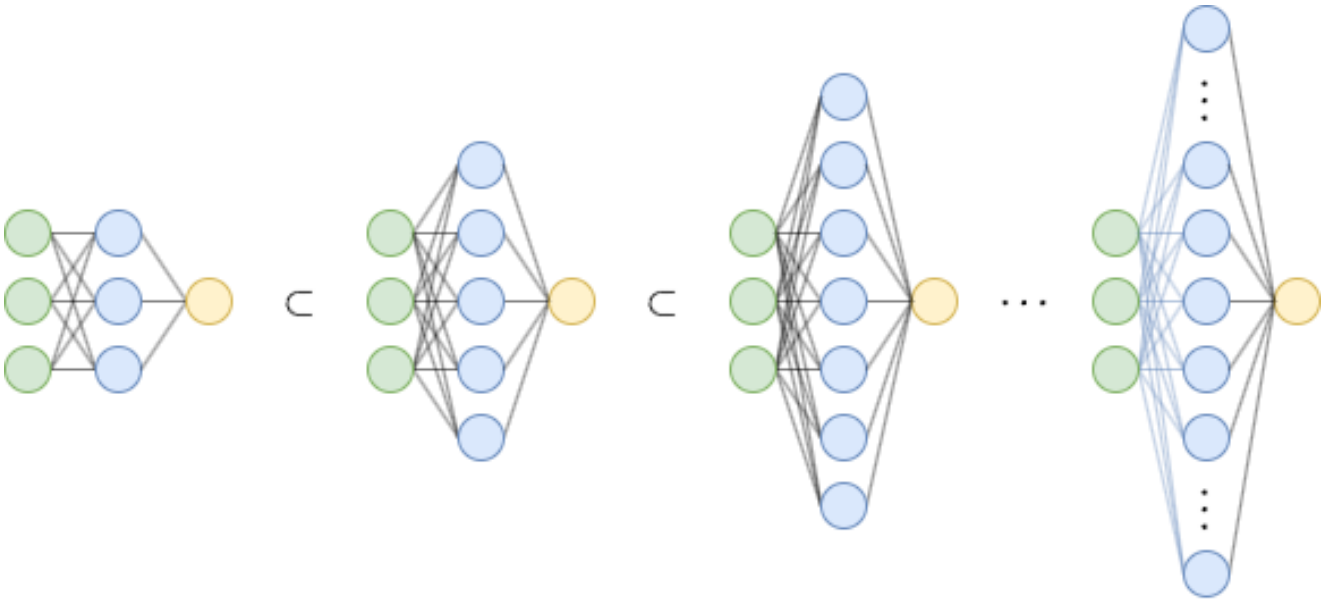
Index

UAT for Wide Network

Theorem ([Cybenko, 1989, Hornik, 1991])

Suppose $f \in C\left(\mathbb{R}^d\right)$ *or* $L^p\left(\mathbb{R}^d\right)$.
For any $\varepsilon > 0$ *and a compact set* $\Omega \subset \mathbb{R}^d$, *there exists a two-layer neural network* F *such that*

$$\|f - F\| < \varepsilon.$$



Sketch of Proof

- If $\sigma(t) \to 1$ and $\sigma(-t) \to 0$ as $t \to \infty$, we call σ is *sigmoidal function*.
- If $\int_\Omega \sigma(w \cdot x + b) f(x) \, dx = 0$ for all $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ implies $f = 0$, we call σ is *discriminatory*.

We will show that if σ is a discriminatory, then the set

$$S = \left\{ g(x) \mid g(x) = \sum_{i=1}^{m} v_i \sigma(w_i \cdot x + b_i), m \in \mathbb{N} \right\}$$

is dense subset of $C(\Omega)$, i.e. $\bar{S} = C(\Omega)$.

Sketch of Proof

If $\bar{S} \neq C(\Omega)$, then there is a linear map that separates $\bar{S}$ and $C(\Omega)$.

Theorem (Hahn-Banach Theorem)

Let A, B be closed vector spaces and $A \lneq B$. Then, there exists a linear map L such that

$$L(A) = 0, \quad L(B) \neq 0.$$

Example 1: $A = \{(x, 0) \mid x \in \mathbb{R}\}$, $B = \mathbb{R}^d$. $L(x, y) = y$.

Example 2: P_n is the set of polynomial of degree at most n. $P_n \lneq C(\Omega)$, and $L(f) = D^{n+1} f$ is zero on P_n but not on $C(\Omega)$.

Sketch of Proof

If $\bar{S} \lneq C(\Omega)$, there exists L such that $L(\bar{S}) = 0$ but $L(C(\Omega)) \neq 0$.

Theorem (Riesz Representation Theorem)

Let L be a bounded linear functional on $C(\Omega)$. Then, there is a measure μ such that for any $h \in C(\Omega)$,

$$L(h) = \int_{\Omega} h(x) \, \mathrm{d}\mu(x).$$

Roughly, we may say that

$$L(h) = \int_{\Omega} h(x) \, g(x) \, \mathrm{d}x,$$

for some nonzero $g : \mathbb{R}^d \to \mathbb{R}$.

Sketch of Proof

Consider $h = \sigma(w \cdot x + b)$ for all w and b.

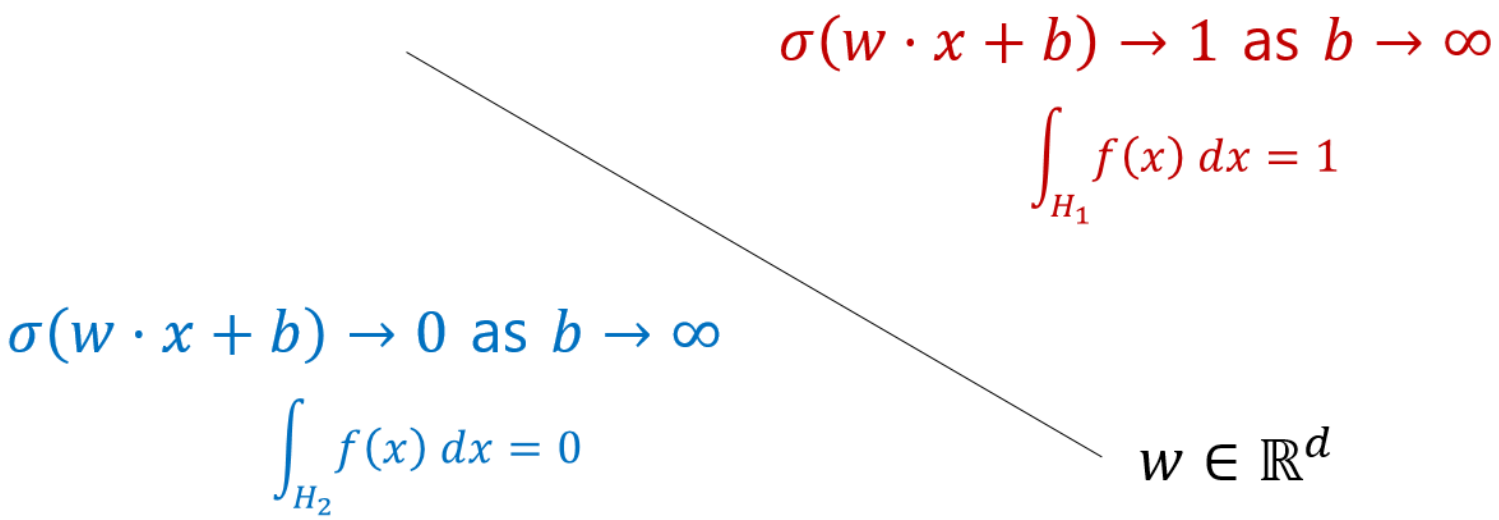
$$L(h) = L(\sigma(w \cdot x + b)) = \int_{\Omega} \sigma(w \cdot x + b) g(x) \, \mathrm{d}x = 0.$$

Since σ is discriminatory, we have contradiction $g = 0$.

Sketch of Proof

Lemma

Any bounded, continuous sigmoidal function, σ is discriminatory.



Corollary

Suppose $\sigma = \max\{0, x\}$. *Then, the set*

$$S = \left\{ g(x) \mid g(x) = \sum_{i=1}^{m} v_i \sigma(w_i \cdot x + b_i)\, m \in \mathbb{N} \right\}$$

is dense subset of $C(\Omega)$.

Since $\rho(x) := \sigma(x+1) - \sigma(x)$ is bounded continuous sigmoidal function, the set

$$\left\{ g(x) \mid g(x) = \sum_{i=1}^{m} v_i \rho(w_i \cdot x + b_i)\, m \in \mathbb{N} \right\}$$

is dense.

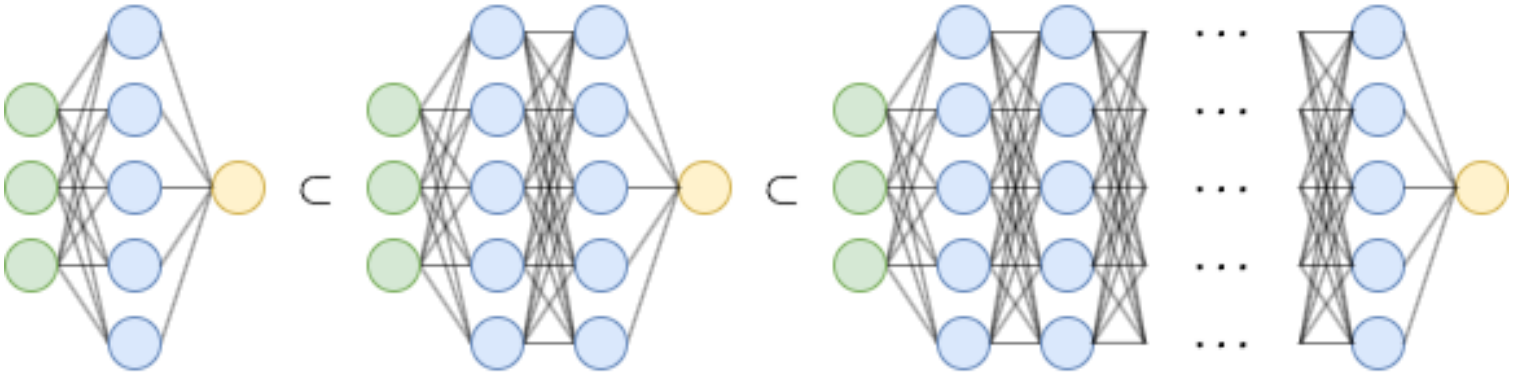
There are many kinds of UAT for deep learning, but proofs of those theorems differ each other.

However, most of such theorems have used the classical result from [Cybenko, 1989, Hornik, 1991].

Theorem (UAT for deep network [Lu et al., 2017, Kidger and Lyons, 2020])

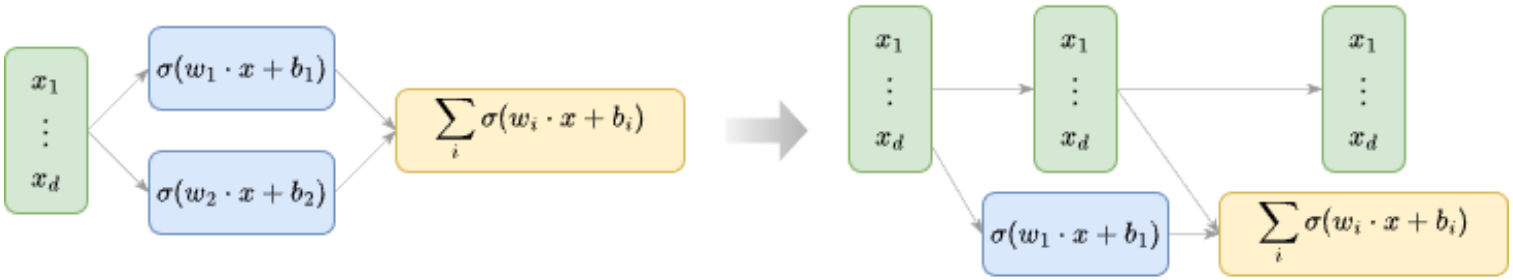
Suppose $f \in C\left(\mathbb{R}^d\right)$ *or* $L^p\left(\mathbb{R}^d\right)$.
For any $\varepsilon > 0$ *and a compact set* $\Omega \subset \mathbb{R}^d$, *there exists a neural network* F *of width* $d + 4$ *such that*

$$\|f - F\| < \varepsilon.$$



Sketch of Proof

Construct deep network to copy wide two-layered network.



Since the two-layer network can approximate arbitrary function, and deep network can copy any two-layer network, it is also able to approximate a target function.

Basic results on UAT is proved by contradiction with Hahn-Banach Theorem, which guarantees only the existence.

Q: How many parameters are required to achieve ε error?

Index

Theorem ([Yarotsky, 2017])

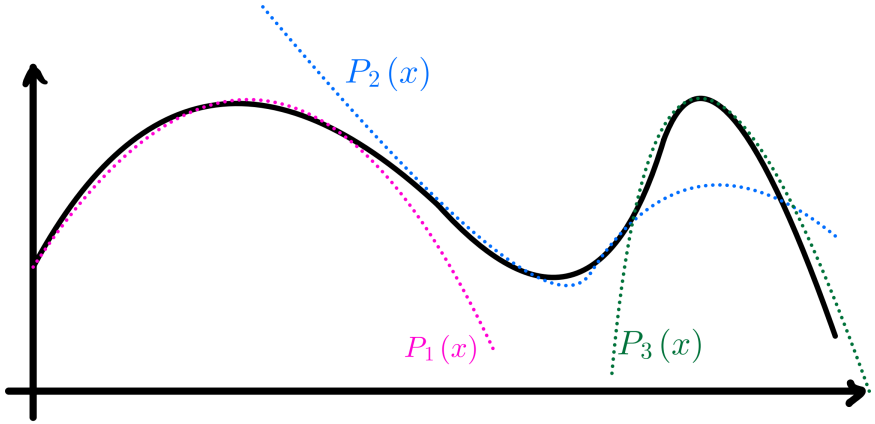
Let f be a Lipschitz function with $\|f\|_\infty, \|Df\|_\infty \leq 1$. For any $\varepsilon > 0$, there is a ReLU network F that has at most $O\left(\varepsilon^{-d} \log(1/\varepsilon) + 1\right)$ weights, such that $\|f - F\|_\infty \leq \varepsilon$.

The theorem has constructive proof, so we can count the number of computation unit in every node and layers.

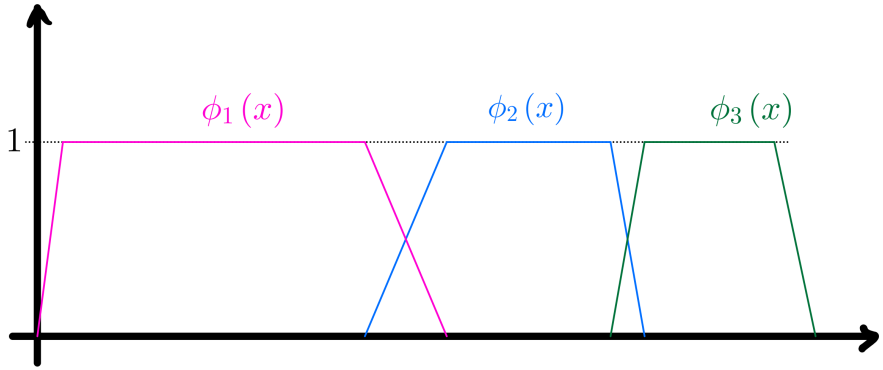
The exponential term ε^{-d} is called *the Curse of Dimensionality*.

Sketch of Proof

1. The target function f is uniformly approximated by the Taylor expansion, locally.



2. Merge the 'local Taylor expansions' via a partition of unity.



Sketch of Proof

Approximating multiplication $\times(x, y) = xy$

Proposition

Given $M > 0$ and $\varepsilon \in (0, 1)$, there is a ReLU network η with two input units that implements a function $\widetilde{\times} : \mathbb{R}^2 \to \mathbb{R}$ so that

1. *for any inputs x, y, if $|x| \leq M$ and $|y| \leq M$, then $\left|\widetilde{\times}(x, y) - xy\right| \leq \varepsilon$;*
2. *if $x = 0$ or $y = 0$, then $\widetilde{\times}(x, y) = 0$;*
3. *the depth and the number of weights and computation units in η are not greater than $c_1 \log\left(\frac{1}{\varepsilon}\right) + c_2$ with an absolute constant c_1 and a constant $c_2 = c_2(M)$.*

Sketch of Proof

1. A multiplication $\times(x, y) := xy$ is derived from square on $[0, 1]^2$:

$$xy = \frac{M^2}{8}\left(\left(\frac{|x+y|}{2M}\right)^2 - \left(\frac{|x|}{2M}\right)^2 - \left(\frac{|y|}{2M}\right)^2\right).$$

2. A sum of compositions $g_s(x) = \overbrace{g \circ g \circ \cdots \circ g}^{s}$ of the 'tooth' function

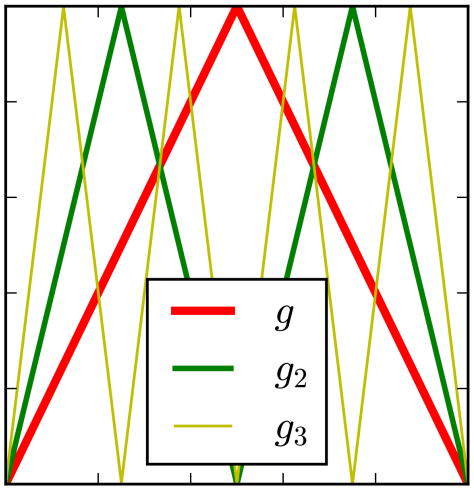
$$g(x) = \sigma(x) - 4\sigma\left(x - \frac{1}{2}\right) + 2\sigma(x-1)$$

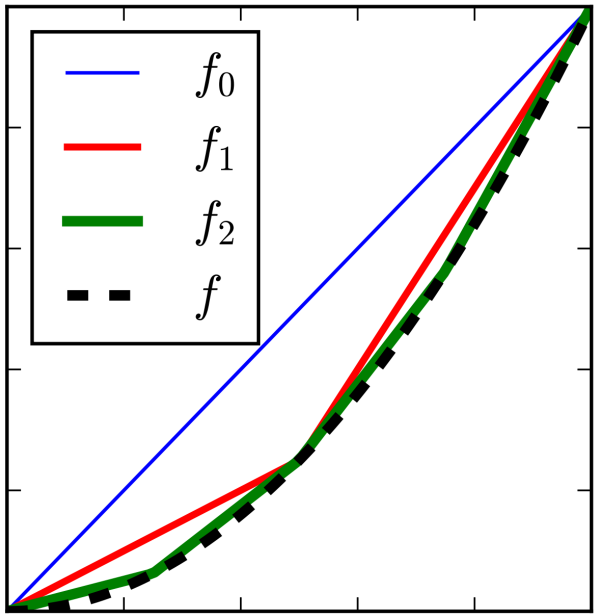
efficiently approximates the square function.

Sketch of Proof

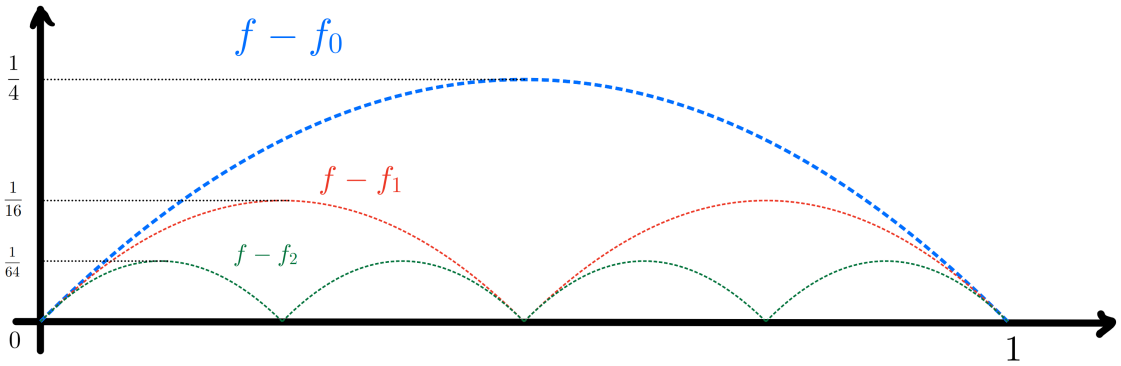
$$g(x) = \begin{cases} 2x & x \in \left[0, \frac{1}{2}\right] \\ 2(1-x) & x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

$$g_s(x) = \begin{cases} 2^s \left(x - \frac{2k}{2^s}\right) & x \in \left[\frac{2k}{2^s}, \frac{2k+1}{2^s}\right] \\ 2^s \left(\frac{2k}{2^s} - x\right) & x \in \left[\frac{2k-1}{2^s}, \frac{2k}{2^s}\right] \end{cases}$$

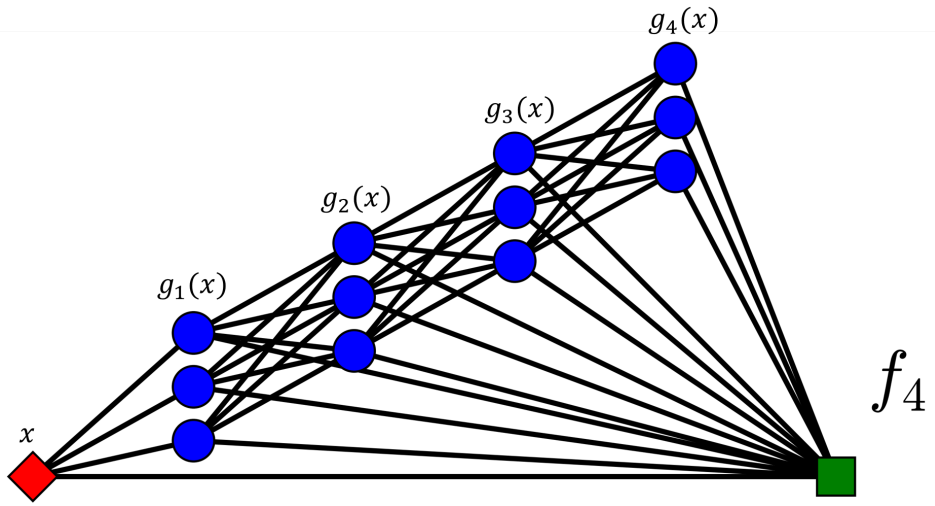




Sketch of Proof



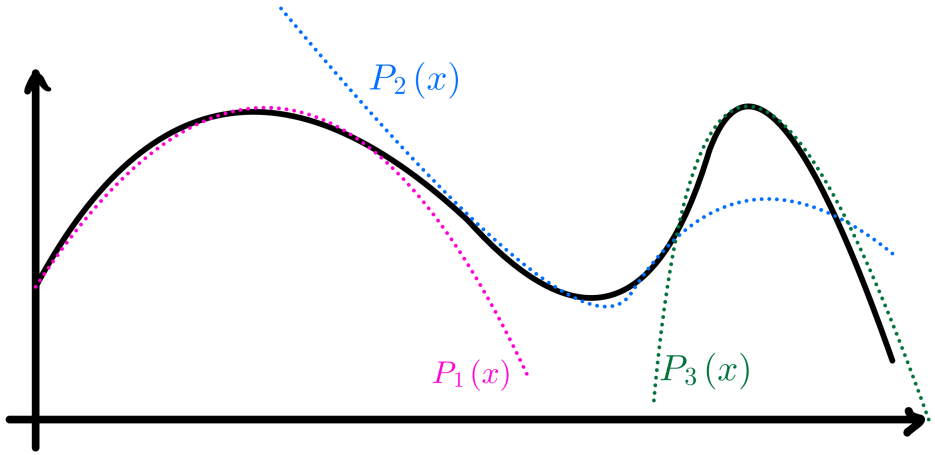
$$\sup_{x\in[0,1]} |f(x) - f_s|(x) \leq \frac{1}{4}2^{-2s}$$



Sketch of Proof

The remainder term of the n-th order Taylor expansion is uniformly bounded on a small range $\left[-\frac{1}{N}, \frac{1}{N}\right]$:

$$\left| f(x_0 + x) - \sum_{k=0}^{n-1} \frac{1}{k!} f^{(k)}(x_0) x^k \right| = \frac{1}{n!} \left| f^{(n)}(x_0 + x^*) x^n \right| \leq \frac{1}{n!} \left(\frac{1}{N} \right)^n .$$



Sketch of Proof

For given $N \in \mathbb{N}$ and $m = 0, 1, \ldots, N$, $\phi_m(x) = \psi\left(3N\left(x - \frac{m}{N}\right)\right)$ forms a partition of unity, where

$$\psi(x) = \begin{cases} 1, & |x| < 1, \\ 0, & 2 < |x|, \\ 2 - |x|, & 1 \leq |x| \leq 2. \end{cases}$$

Sketch of Proof

Expand f as the Taylor series with the partition:

$$
\begin{aligned}
f(x) &= \sum_{m=0}^{N} \phi_m(x) f(x) && \text{(partition of unity)} \\
&\approx \sum_{m=1}^{N} \phi_m(x) \sum_{k=1}^{n-1} \frac{1}{k!} f^{(k)}\left(\frac{m}{N}\right)\left(x-\frac{m}{N}\right)^k && \left(\text{error} \le C_1(n)\left(\frac{1}{N}\right)^n\right) \\
&= \sum_{m=1}^{N} \sum_{k=1}^{n-1} \frac{1}{k!} f^{(k)}\left(\frac{m}{N}\right) \phi_m(x)\left(x-\frac{m}{N}\right)^k
\end{aligned}
$$

Sketch of Proof

Each $\phi_m(x)\left(x-\frac{m}{N}\right)^k$ includes at most n multiplications:

$$\phi_m(x)\left(x-\frac{m}{N}\right)^k \approx \underbrace{\widetilde{\times}\left(\phi_m(x), \widetilde{\times}\left(x-\frac{m}{N}, \cdots\right)\right)}_{k \text{ times of } \widetilde{\times}}.$$

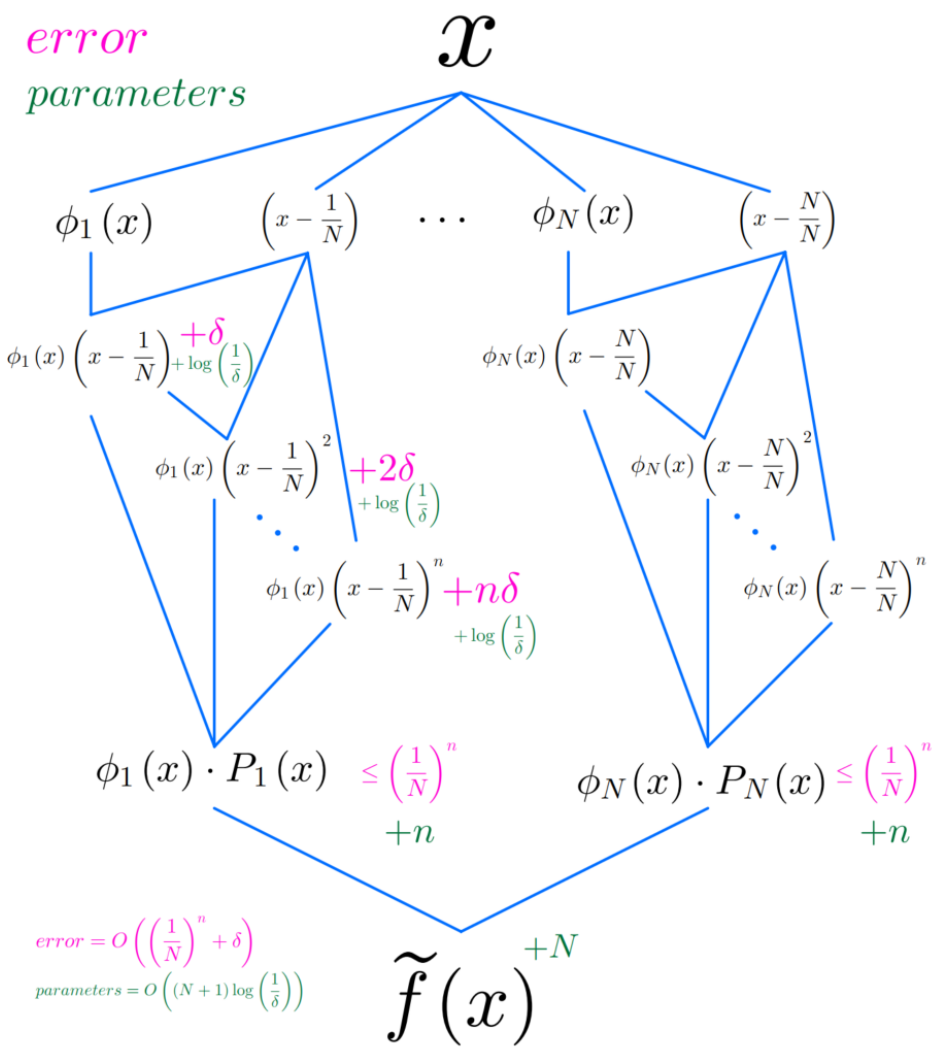
Since each $\widetilde{\times}$ yields at most δ, at most $n\delta$ error occurs.

$$\sum_{m=1}^{N}\sum_{k=1}^{n-1}\frac{1}{k!}f^{(k)}\left(\frac{m}{N}\right)\phi_m(x)\left(x-\frac{m}{N}\right)^k$$

$$\approx \sum_{m=1}^{N}\sum_{k=1}^{n-1}\frac{1}{k!}f^{(k)}\left(\frac{m}{N}\right)\widetilde{\times}\left(\phi_m(x), \widetilde{\times}\left(x-\frac{m}{N}, \cdots\right)\right)$$

$$(\text{error} \leq C_2(n)\delta)$$

Sketch of Proof



This constructive proof takes *divide and conquer* strategy: construct network approximating *local bases* such as polynomial and merge them using the partition of unity.

It is known that the curse of dimensionality occurs in any parameterized function class[DeVore et al., 1989].

A study on *which functions can be efficiently approximated by neural network* results in Barron space.

Index

Theorem (UAT for Deep Network [Johnson, 2019])

- $d \geq 2$: *input dimension*
- σ : *uniformly continuous activation function that can be approximated by one-to-one functions*

Then neural networks with width d cannot approximate any function with a level set containing a bounded path component.

Activation function classes include *sigmoid*, *ReLU*, or other frequently used functions.

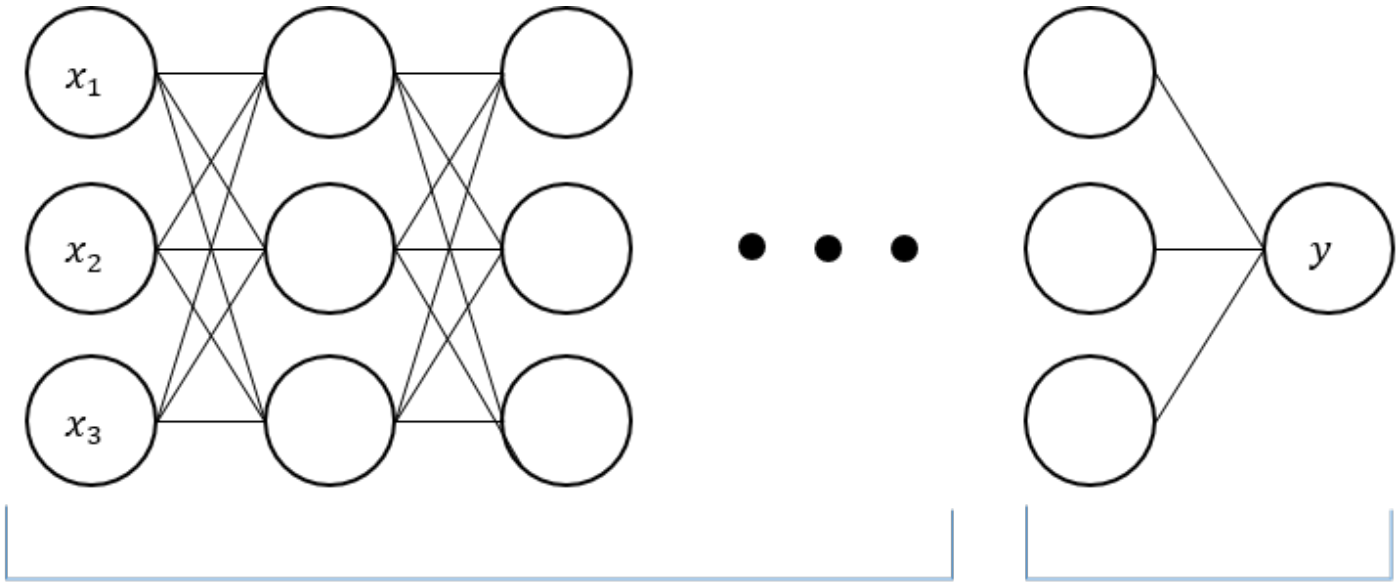
Sketch of Proof

Lemma

If all weight matrices of width d networks are invertible and σ is one-to-one activation function, then $f^{-1}(y)$ is homeomorphic to an open (possibly empty) subset of $\mathbb{R}^{d-1}$.

In particular, $f^{-1}(y)$ has unbounded component.

Sketch of Proof



$\mathbb{R}^n$ → Open Set (I)

Level Set (C) ← $H \approx \mathbb{R}^{n-1}$ ← Point

C → $H \cap I$ (Open in H)

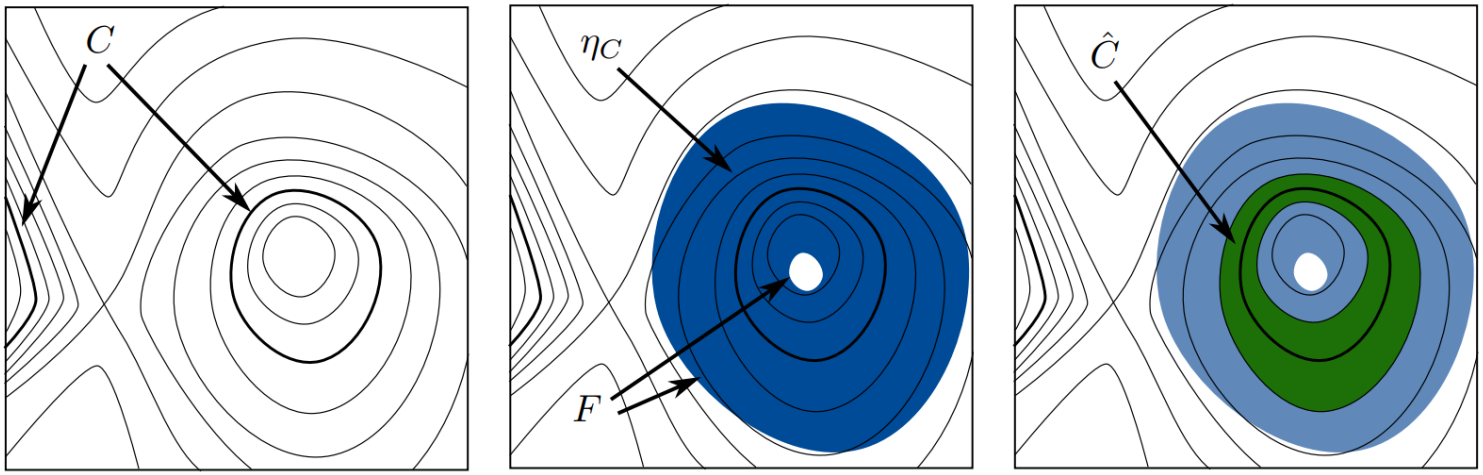
If Bounded → Compact (in H)

Sketch of Proof

Lemma

If $\mathcal{M}$ is a model family in which every function has unbounded level components then any function approximated by $\mathcal{M}$ has unbounded level components.

If two outputs are close enough then level sets of them are close.



If width is less than input dimension, neural networks cannot approximate a function with bounded level set.

Width is required for breaking topological property.

Index

Convolutional Neural Networks(CNNs)

Image

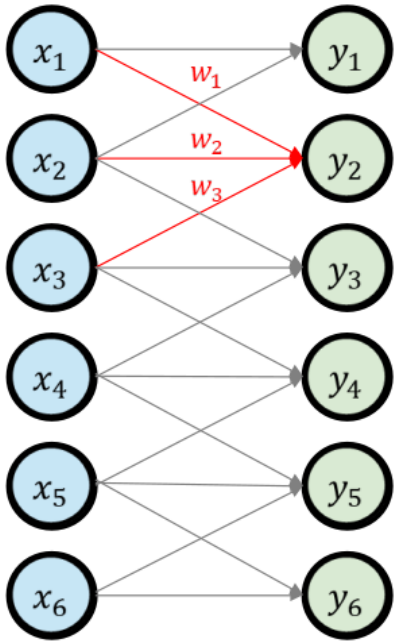
Segmentation: Tiger

Translated Image

Segmentation: Tiger

Some task or data are essentially *translation equivariant/invariant*.

Input



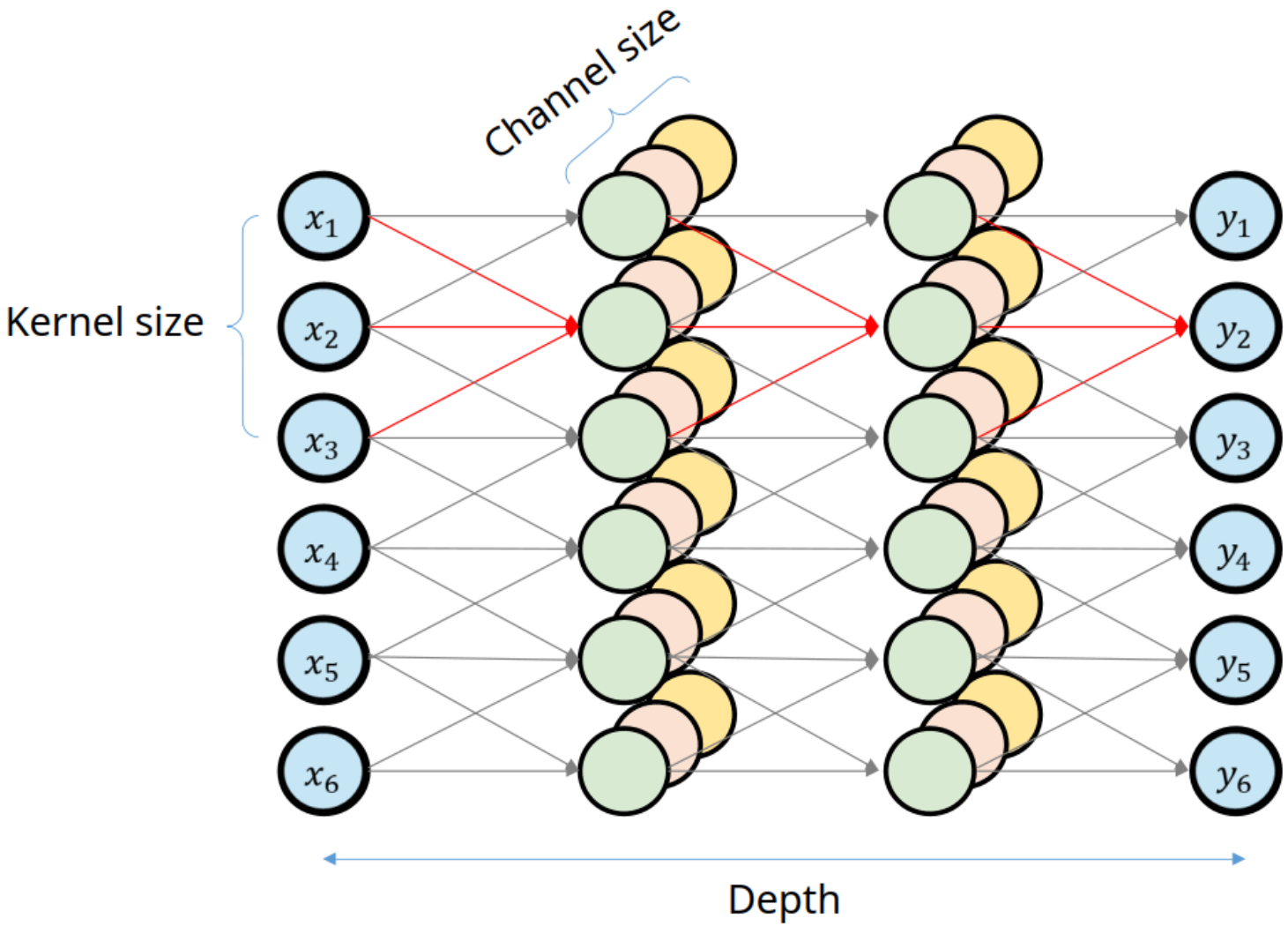
$$y_2 = \sigma\left(\sum_i w_i\, x_{2-i} + b\right)$$

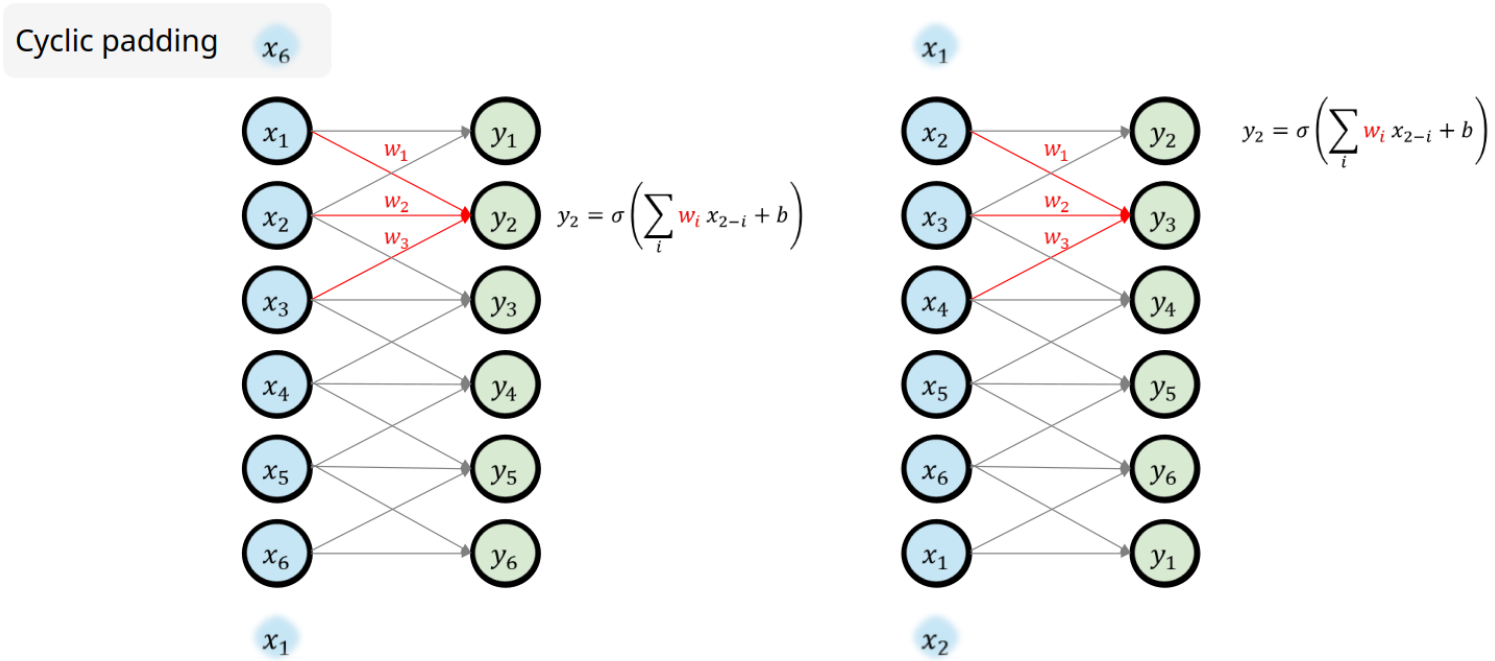
$$y_4 = \sigma\left(\sum_i w_i\, x_{4-i} + b\right)$$

Output

$$Y = \sigma(w * X + b)$$

Use convolution instead of dense linear map.

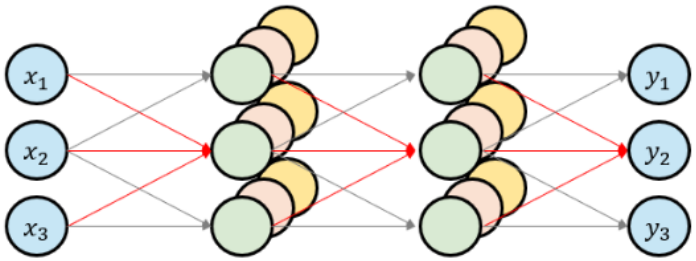

Channel size
Kernel size
x_1
x_2
x_3
x_4
x_5
x_6
y_1
y_2
y_3
y_4
y_5
y_6
Depth



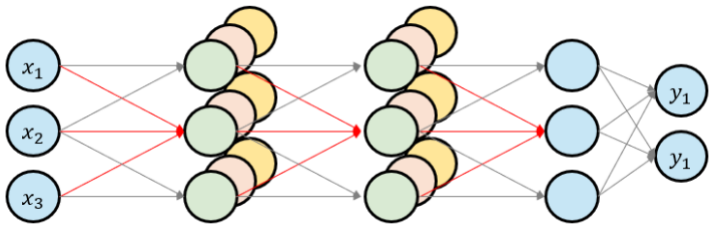
- Cyclic padding: exact translation equivariance.
- Zero padding: more common, break translation equivariance.

Theorem ([Yarotsky, 2022, Maron et al., 2019])

- *Provided cyclic padding, CNNs are dense in the space of translation equivariant function.*
- *If the CNN is followed by linear layer, then it is dense in the space of translation invariant function.*



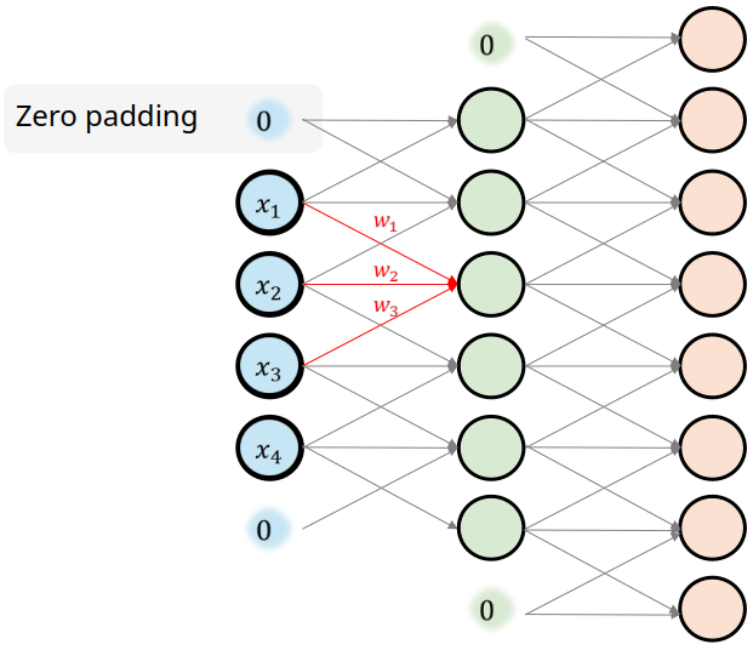
Can approximate translation equivariant function.



Can approximate translation invariant function.

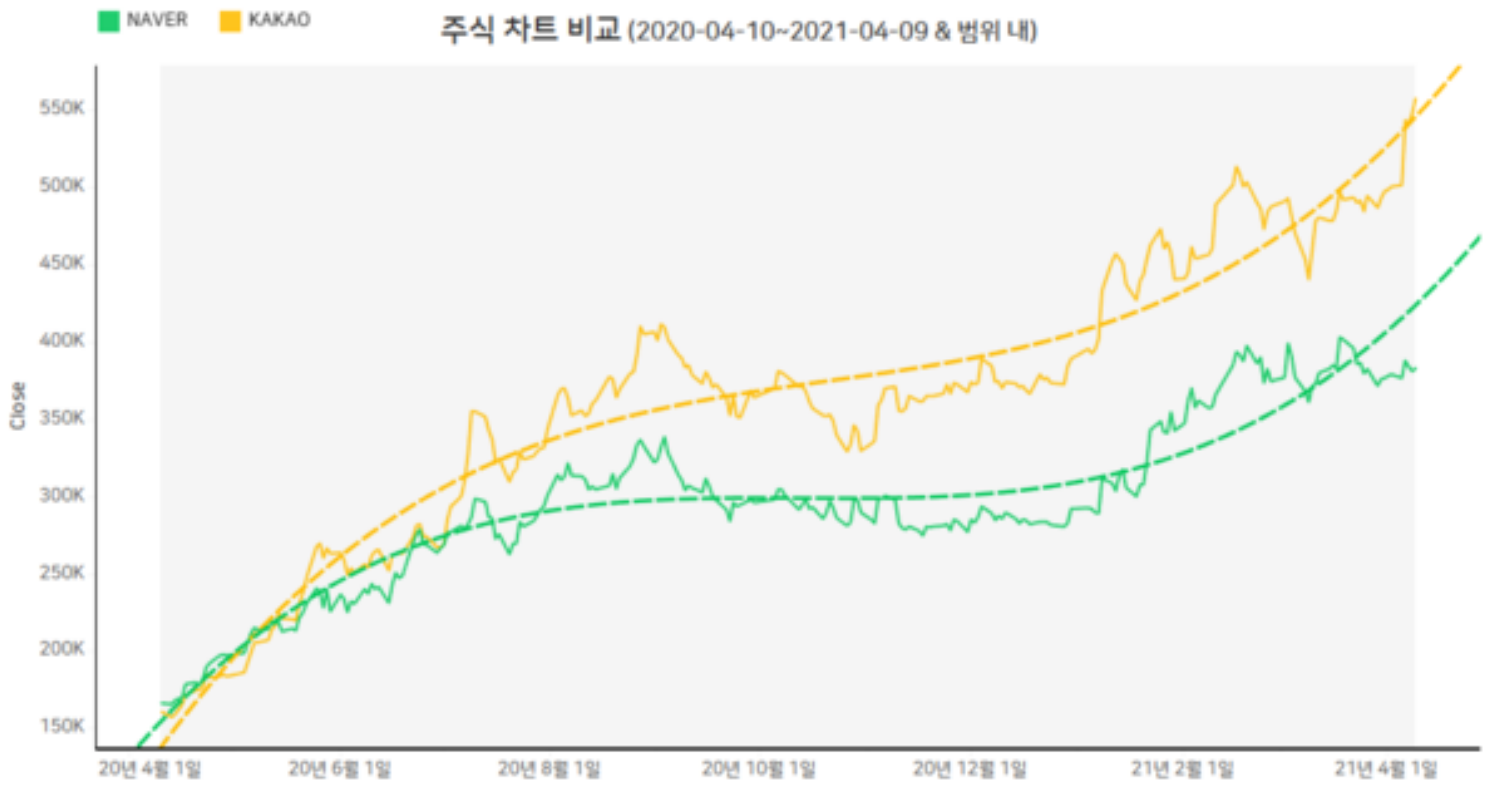
Theorem ([Zhou, 2020])

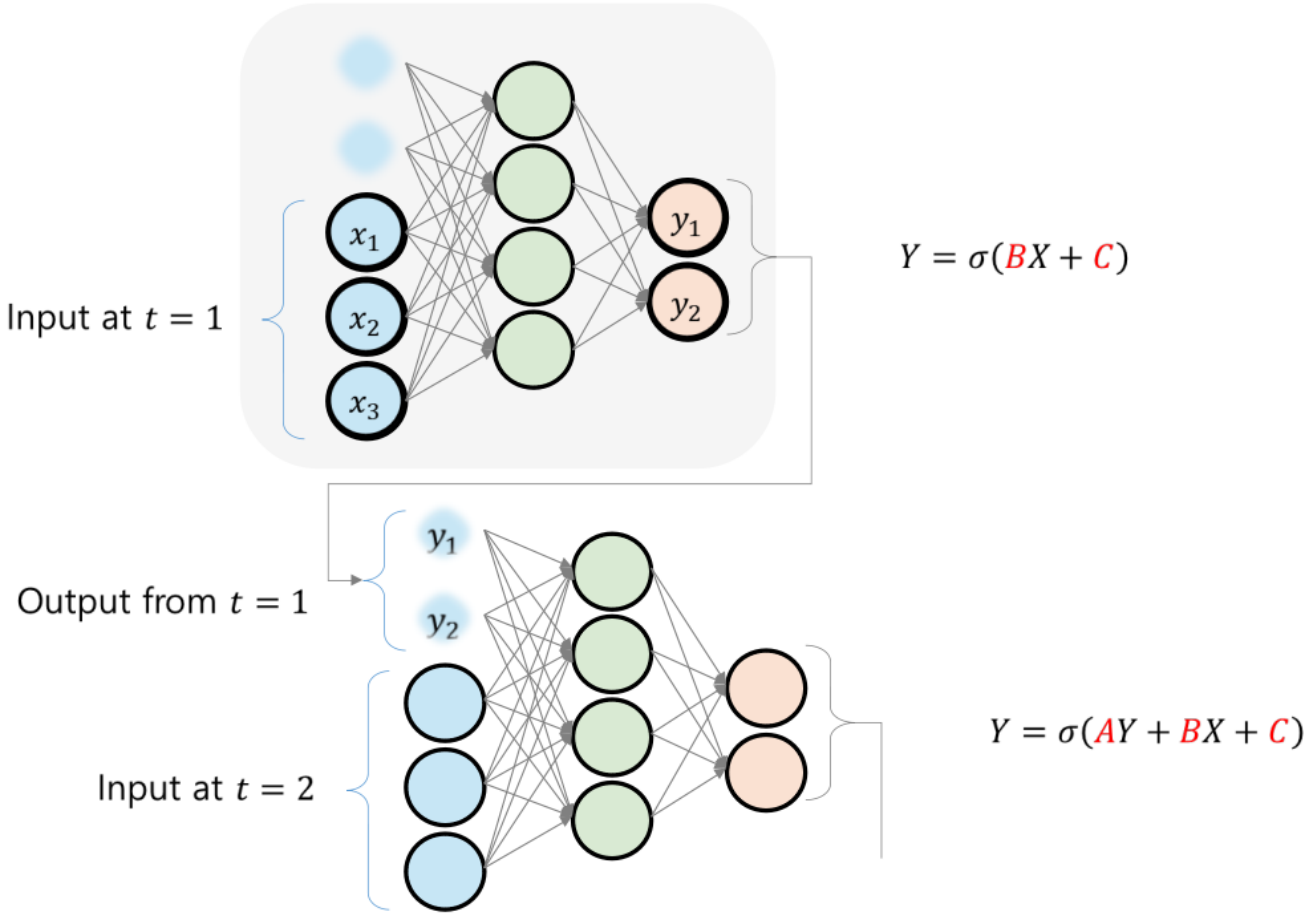
Provided zero padding, CNNs are dense in the space of continuous function.



Wide CNNs can approximate any function, since they are equivalent to MLPs.

Recurrent Neural Networks(RNNs)





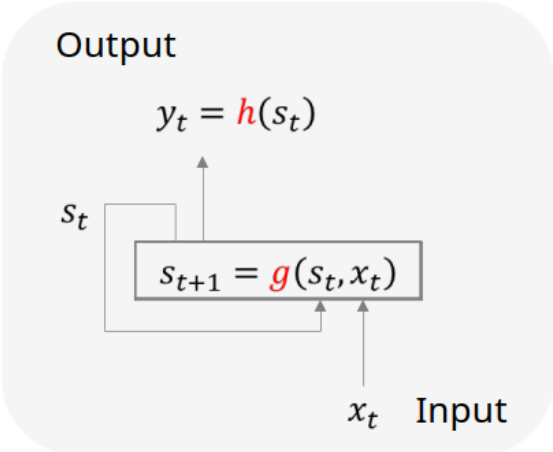
Use output at time t to attain output at time $t + 1$.

Theorem ([Schäfer and Zimmermann, 2006])

The set of recurrent neural networks

$$s_{t+1} = \sigma\left(As_t + Bx_t + C\right)$$
$$y_t = Ws_t$$

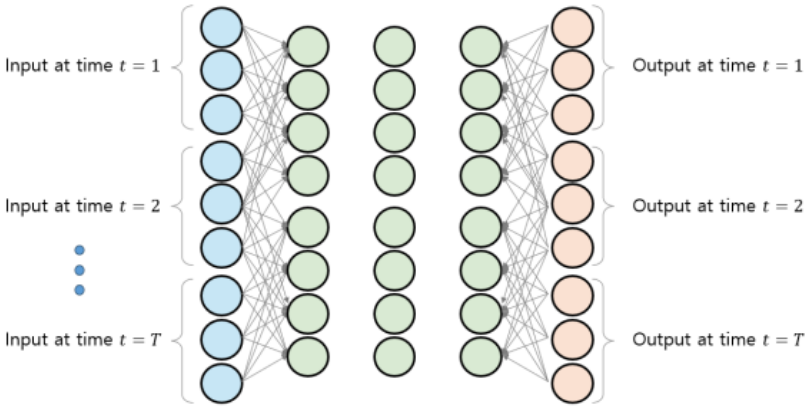
is dense in the space of open dynamical systems.

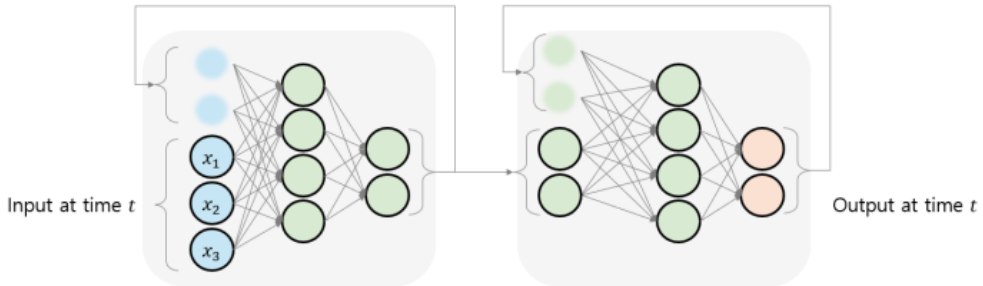


Every *Causal mapping* can be represented as open dynamical system.

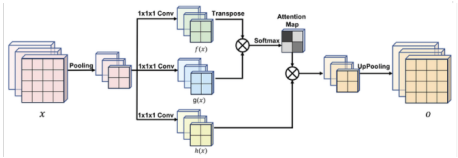
Theorem ([Song et al., 2023])

The set of deep RNNs of width $d_x + d_y + 4$ is dense in the space of causal functions.





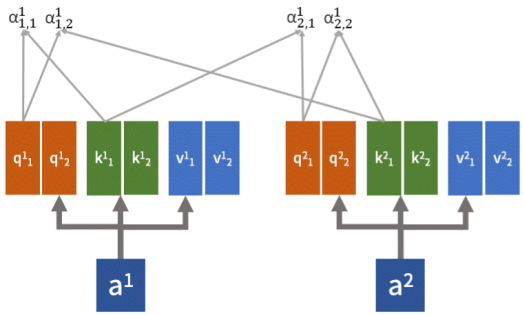
Attention Layer (Transformer)



Attention layer

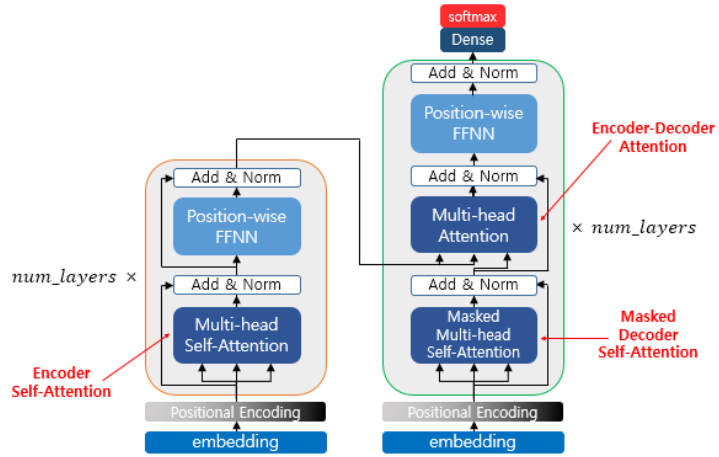
$$z_{\ell+1} = W_O W_V z_\ell \sigma\left((W_K z_\ell)^{\mathrm{T}} W_Q z_\ell\right)$$

Attention is inner product of key and quary.



$$A^1 = \begin{bmatrix} \alpha^1_{1,1} & \alpha^1_{1,2} \\ \alpha^1_{2,1} & \alpha^1_{2,2} \end{bmatrix}$$
Attention Matrix 1

$$A^2 = \begin{bmatrix} \alpha^2_{1,1} & \alpha^2_{1,2} \\ \alpha^2_{2,1} & \alpha^2_{2,2} \end{bmatrix}$$
Attention Matrix 2



Transformer

Attention is permutation-equivariant.

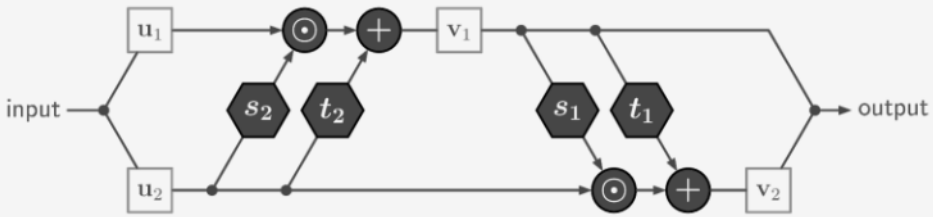
Transformer architecture is composition of linear and attention layers.

Theorem ([Yun et al., 2020])

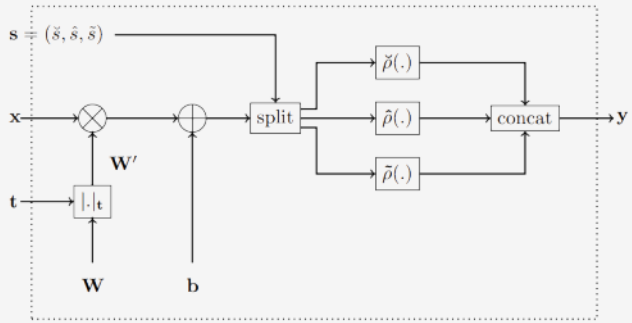
- *Without positional encoding, transformers are dense in the space of permutation equivariant functions.*
- *With positional encoding, transformers are dense in the space of continuous functions.*

Other Layers

Invertible Neural Network



Monotonic Neural Network



Lipschitz Neural Network

Algorithm 1 1-Lipschitz convolutional layer

Require: $h_{\text{in}} \in \mathbb{R}^{p \times s \times s}$, $P \in \mathbb{R}^{(p+q) \times q \times s \times s}$, $d \in \mathbb{R}^q$

$\tilde{h}_{\text{in}} \leftarrow \text{FFT}(h_{\text{in}})$

$\Psi \leftarrow \text{diag}(e^d)$, $\begin{bmatrix} \tilde{A} & \tilde{B} \end{bmatrix}^* \leftarrow \text{Cayley}(\text{FFT}(P))$

$\tilde{h}[:,i,j] \leftarrow \sqrt{2}\Psi^{-1}\tilde{B}[:,:,i,j]\tilde{h}_{\text{in}}[:,i,j]$

$\tilde{h} \leftarrow \text{FFT}\left(\sigma(\text{FFT}^{-1}(\tilde{h}) + b)\right)$

$\tilde{h}_{\text{out}}[:,i,j] \leftarrow \sqrt{2}A[:,:,i,j]^*\Psi\tilde{h}[:,i,j]$

$h_{\text{out}} \leftarrow \text{FFT}^{-1}(\tilde{h}_{\text{out}})$

For each layer designed to satisfy specific properties, the universal approximation theorem in the space of functions with the properties need to hold and have been proved.

Thank You

References I


Cybenko, G. (1989).
Approximation by superpositions of a sigmoidal function.
Mathematics of control, signals and systems, 2(4):303–314.


DeVore, R. A., Howard, R., and Micchelli, C. (1989).
Optimal nonlinear approximation.
Manuscripta mathematica, 63(4):469–478.


Hornik, K. (1991).
Approximation capabilities of multilayer feedforward networks.
Neural networks, 4(2):251–257.


Johnson, J. (2019).
Deep, skinny neural networks are not universal approximators.
In *7th International Conference on Learning Representations, ICLR 2019*.

References II


Kidger, P. and Lyons, T. (2020).
Universal approximation with deep narrow networks.
In *Conference on learning theory*, pages 2306–2327. PMLR.


Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017).
The expressive power of neural networks: A view from the width.
Advances in neural information processing systems, 30.


Maron, H., Fetaya, E., Segol, N., and Lipman, Y. (2019).
On the universality of invariant networks.
In *International conference on machine learning*, pages 4363–4371. PMLR.


Schäfer, A. M. and Zimmermann, H. G. (2006).
Recurrent neural networks are universal approximators.
In *International conference on artificial neural networks*, pages 632–640. Springer.

References III


Song, C., Hwang, G., ho Lee, J., and Kang, M. (2023).
Minimal width for universal property of deep rnn.
Journal of Machine Learning Research, 24(121):1–41.


Yarotsky, D. (2017).
Error bounds for approximations with deep relu networks.
Neural networks, 94:103–114.


Yarotsky, D. (2022).
Universal approximations of invariant maps by neural networks.
Constructive Approximation, 55(1):407–474.


Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S., and Kumar, S. (2020).
Are transformers universal approximators of sequence-to-sequence functions?
In *International Conference on Learning Representations*.

References IV


Zhou, D.-X. (2020).
Universality of deep convolutional neural networks.
Applied and computational harmonic analysis, 48(2):787–794.