Approximation Theory

Yao Zhang

The guy is a populace

Mostly based on Thomas Hofmann's lecture in ETH

https://zhims.github.io/

Dec 7, 2019

Question

How can we characterize the elementary functions implemented by comptational unit in neural networks?

Level Sets

Definition 1 (Level set)

The level set of a function $f : \mathbb{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ is one-parametric family of sets defined as

$$L_f(c) = \{x : f(x) = c\} \subseteq \mathbb{D}, \quad c \in \mathbb{R} \tag{1}$$

Level sets generalize the concept of an inverse function.

Proposition 1

The Level sets of a function f over $D \subseteq \mathbb{R}^n$ form a partition of $\mathbb{D}$.

Affine Functions and Subspace

Definition 2 (Affine Function)

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine, if it can be written as

$$f(x) = Ax + b, \quad \textit{for some} \quad A \in \mathbb{R}^{m \times n}, \; b \in \mathbb{R}^m. \tag{2}$$

Proposition 2

f being affine is equivalent to the condition

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y), \quad \forall \alpha, \beta : \quad \alpha + \beta = 1 \tag{3}$$

Definition 3 (Affine Subspace)

U is an affine subspace of $\mathbb{R}^n$, if $U = v + V$, where $v \in \mathbb{R}^n$ and V is a linear subspace.

Proof of Proposition 2

Proof.

1. $\Rightarrow$

$$\begin{aligned} f(\alpha x+\beta y) &= A(\alpha x+\beta y)+b = \alpha Ax+\beta Ay+b \\ &= (\alpha Ax+\alpha b)+(\beta Ay+\beta b) = \alpha f(x)+\beta f(y) \end{aligned} \tag{4}$$

2. $\Leftarrow$ Show that $L(x) \triangleq f(x)-f(0)$ is linear.

$$\begin{aligned} L(\alpha x) &= f(\alpha x+(1-\alpha)0)-f(0) \\ &= \alpha f(x)-\alpha f(0) = \alpha L(x) \end{aligned} \tag{5}$$

$$\begin{aligned} L(x+y) &= 2L\left(\frac{1}{2}x+\frac{1}{2}y\right) = 2\left(\frac{1}{2}f(x)+\frac{1}{2}f(y)-f(0)\right) \\ &= f(x)-f(0)+f(y)-f(0) = L(x)+L(y) \end{aligned} \tag{6}$$

□

Ridge functions

Definition 4 (Ridge function)

$f : \mathbb{R}^n \to \mathbb{R}$ is a ridge function, if it can be written as

$$f = \sigma \circ \overline{f}, \quad \textit{where} \quad \overline{f} : \mathbb{R}^n \to \mathbb{R} \quad \textit{is affine} \tag{7}$$

and $\sigma : \mathbb{R} \to \mathbb{R}$ an arbitrary scalar function.

Ridge Function = Affine Function + Scalar Non-Linearity.

Explicit form of a ridge function

$$f(x) = \sigma\left(w^T x + b\right), \quad \textit{for some} \quad w \in \mathbb{R}^n,\ f(x), b \in \mathbb{R} \tag{8}$$

Level Sets of Affine Functions

Proposition 3

The level sets of an affine function $f(x) = w^T x + b$ *are affine subspaces* $\alpha w + V,\ \alpha \in \mathbb{R},\ V = \left\{x : w^T x = 0\right\}$

Proof.

Let us write $x = \alpha w + x^0$, where $x^0 \perp w$.

$$\begin{aligned} x \in L_f(c) &\Leftrightarrow w^T x + b = c \Leftrightarrow \alpha \|w\|^2 = c - b \\ &\Leftrightarrow \alpha = \frac{c-b}{\|w\|^2} \end{aligned} \tag{9}$$

which means that α is constant for all $x \in L_f(c)$. □

Corollary 1

$$L_f(c) = \frac{(c-b)\,w}{w^T w} + \left\{x : w^T x = 0\right\} \tag{10}$$

Level Sets of Ridge Functions

Proposition 4

The level sets of ridge functions $f = \sigma \circ \overline{f}$ are unions of affine subspaces, specifically

$$L_f(c) = \bigcup_{d:\, \sigma(d)=c} L_{\overline{f}}(d) \tag{11}$$

Corollary 2

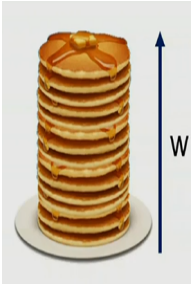
If σ is one-to-one with inverse σ^{-1} then

$$L_f(c) = L_{\overline{f}}\left(\sigma^{-1}(c)\right) \tag{12}$$

and the level sets of f and $\overline{f}$ are in one-to-one correspondence.

Ridge Functions

Pancake Metaphor



each pancake slice = same function value = level sets, also ridge functions are rich.

Proposition 5

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a ridge function, differentiable at x. Then either $\nabla f(x) = 0$ or $\nabla f(x) \perp L_f(f(x))$.

Proof.

$$\nabla f(x) = \nabla\left(\sigma \circ \overline{f}\right)(x) = \sigma'\left(\overline{f}(x)\right)\nabla\overline{f}(x) \propto w \perp L_f(c) \qquad \square$$

Question

Is the class of ridge function rich enough to approximate a sufficiently large class of function, e.g. $C(\mathbb{R}^n)$?

Dense Approximations

Definition 5 (Dense Approximation)

A function class $\mathcal{H} \in C(\mathbb{R}^n)$ is a dense approximation of $C(\mathbb{R}^n)$ or is dense in $C(\mathbb{R}^n)$, if and only if $\forall f \in C(\mathbb{R}^n), \forall \epsilon > 0, \forall K$ compact, $K \in \mathbb{R}^n$:

$$\exists h \in \mathcal{H} \quad s.t. \quad \max_{x \in K} |f(x) - h(x)| = \|f - h\|_{\infty, K} < \varepsilon$$

Remark 1

1. *uniform approximation on compact (i.e. use of ∞-norm)*
2. *sup $\rightarrow$ max (Bolzano-Weierstrass)*
3. *informally speaking: we can approximate any continuous f to arbitrary accuracy on K with suitable member of $\mathcal{H}$*

Universal Approximation with Ridge Functions

Definition 6

$$\mathcal{G}_\sigma^n \triangleq \left\{ g : g(x) = \sigma\left(w^T x + b\right) \quad \text{for some} \quad x, w \in \mathbb{R}^n, b \in \mathbb{R} \right\}$$
$$\mathcal{G}^n \triangleq \bigcup_{\sigma \in C(\mathbb{R})} \mathcal{G}_\sigma^n \qquad \text{universe of continuous ridge functions} \tag{13}$$

Theorem 1 (Vostrecov and Kereines, 1961)

$\mathcal{H}^n \triangleq span\{\mathcal{G}^n\}$ *is dense in* $C(\mathbb{R}^n)$

Remark 2

Note that $\mathcal{H}^n = \left\{ h : \ h = \sum_{j=1}^m g_j, g_j \in \mathcal{G}^n \right\}$, *i.e. one can absorb linear combination weights in functions* g_j.

Ridge Function Network

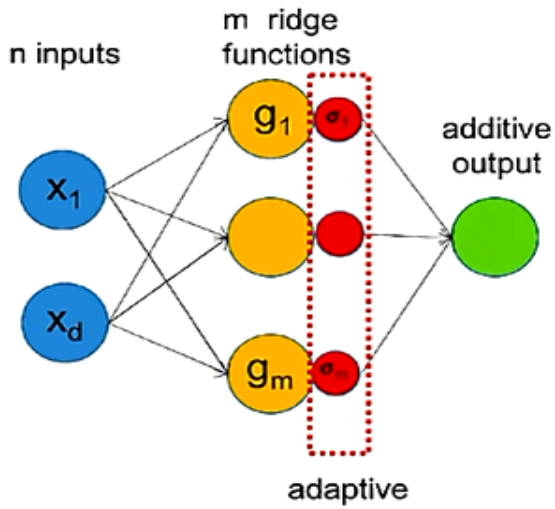


Figure 1: Framework of Ridge Function Network

Thm.1 uses additive combinations of arbitrary (unspecified) ridge function.

Remark 3

1. *it would require some adaptivity of the non-linearity (= learning the activation function)*
2. *it is not inconceivable, but not commonly done*

Question

Can we further specialize the class of ridge function e.g. by choosing one $\sigma \in C(\mathbb{R})$ such that $\mathcal{G}_\sigma^n$ is dense in $C(\mathbb{R}^n)$?

Lemma: Dimension Lifting

Theorem 2 ([Pinkus, 1999])

The density of $\mathcal{H}_\sigma^1$

$$\mathcal{H}_\sigma^1 \triangleq span\left\{\mathcal{G}_\sigma^1\right\} = span\left\{\sigma\left(\lambda t + \theta\right) : \lambda, \theta \in \mathbb{R}\right\} \tag{14}$$

in $C(\mathbb{R})$ *implies the density of*

$$\mathcal{H}_\sigma^n \triangleq span\left\{\mathcal{G}_\sigma^n\right\} = span\left\{\sigma\left(w^T x + b\right) : x, w \in \mathbb{R}^n, b \in \mathbb{R}\right\} \tag{15}$$

in $C\left(\mathbb{R}^n\right)$ *for any* $n \geq 1$.

1. informally: we can lift the density property of ridge function families from $C(\mathbb{R})$ to$C(\mathbb{R}^n)$

Key Advantage of Ridge Function

Key advantage of ridge functions:

1. Picking out a direction of change: done in linear part $\Rightarrow$ essentially equivalent to linear case
2. Non-linear activation: models rate of change in the chosen direction. Just a $C(\mathbb{R})$ function, dimension independent
3. Continuous activation function can be approximated by expansions with fixed activation function. Simplification at the cost of increased representation size.

Question

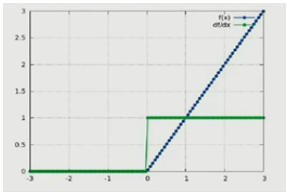
What activation functions are commonly used in modern DNNs and what are the representational powers of resulting networks?

Rectified Linear Units

Definition 7 (ReLU = rectified linear units)

The activation function of a ReLU is defined as

$$(x)_+ = \max\{0, x\}, \qquad \underbrace{\partial(x)_+}_{\textit{subdifferential}} = \begin{cases} \{1\} & \textit{if } x > 0 \\ \{0\} & \textit{if } x < 0 \\ [0,1] & \textit{if } x = 0 \end{cases} \tag{16}$$



1. liner function over half-space $\mathcal{H}$
2. zero on complement $\mathcal{H}^c = \mathbb{R}^n - \mathcal{H}$
3. non-smooth, but simple subgradient

Rectified Linear Units

Closely related alternative activation function

Definition 8 (Absolute value rectification AVU)

The activation function of an absolute value unit (AVU) is given by

$$|x| = \begin{cases} x & \text{if } x \geqslant 0 \\ -x & \text{otherwise} \end{cases}, \quad \partial |x| = \begin{cases} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{17}$$

1. relation to ReLU activation

$$(x)_+ = \frac{x + |x|}{2} \quad \text{and} \quad |x| = 2(x)_+ - x \tag{18}$$

Best Practice: Rectification in Computer Vision

What is the Best Multi-Stage Architecture for Object Recognition?

1. The surprising answer is that using a rectifying non-linearity is the single most important factor in improving the performance of a recognition system.
2. experimental results

Two Stage System: $[64.F^{9\times9}_{CSG} - R/N/P^{5\times5}] - [256.F^{9\times9}_{CSG} - R/N/P^{4\times4}]$ - log_reg					
	$R_{abs} - N - P_A$	$R_{abs} - P_A$	$N - P_M$	$N - P_A$	P_A
U^+U^+	65.5%	60.5%	61.0%	34.0%	32.0%
R^+R^+	64.7%	59.5%	60.0%	31.0%	29.7%
UU	63.7%	46.7%	56.0%	23.1%	9.1%
RR	62.9%	33.7%(±1.5)	37.6%(±1.9)	19.6%	8.8%
GT	55.8%				

3. uses $|x|$, but similar results for $(x)_+$

Rectification in the 19th Century

Theorem 3 ([Shektman, 1982])

Piecewise linear functions are dense in $C[0,1]$.

Theorem 4 ([Lebesgue, 1898])

A piecewise linear function with m *pieces can be written*

$$g(x) = \underbrace{ax + b}_{linear} + \sum_{i=1}^{m-1} c_i (x - x_i)_+ \tag{19}$$

1. knots: $0 = x_0 < x_1 < \cdots < x_{m-1} < x_m = 1$
2. $m + 1$ parameters, $a, b, c_i \in \mathbb{R}$
3. ReLU function approximation in 1D.

Approximation by Rectified Networks

Theorem 5

Networks with one hidden layer of ReLU or absolute value units are universal function approximators.

Proof.

Sketch:

1. Universally approximate $C(K)$ functions(K, compact) by polygonal lines
2. Represent polygonal lines by (linear function $+$) linear combinations of $(\cdot)_+$ or $(\cdot)_-$ functions
3. Apply dimension lifting lemma **??** to show density of the linear span of resulting ridge function families $\mathcal{G}^n_{(\cdot)}$ and $\mathcal{G}^n_{|\cdot|}$

ReLU Network

Theorem allows for the use of restricted set of ridge functions (e.g. ReLU).

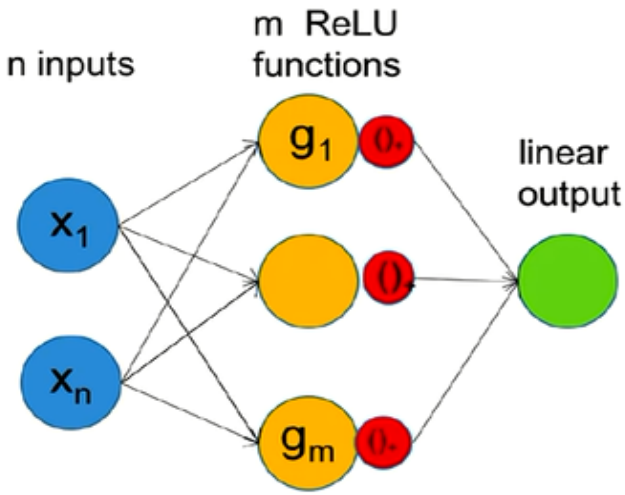


Figure 2: ReLU Network

1. no adaptivity of the non-linearity required (fixed)
2. possibly at the price of increasing hidden layer width (m)

Question

What is the minimal non-linearity required to obtain universal function approximators?

Hinging Hyperplanes

Another piecewise linear set of functions [Breiman, 1993]

Definition 9 (Hinge function)

If $g : \mathbb{R}^n \to R$ can be written with parameters $w_1, w_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$ as below it is called a hinge function

$$g(x) = \max\left(w_1^T x + b_1, w_2^T x + b_2\right) \tag{20}$$

1. two hyperplanes, "glued" together at the face $(w_1 - w_2)^T x + (b_1 - b_2) = 0$
2. easy to fit single hinging hyperplane (iterative algorithm)
3. representational power: $2\max(f, g) = f + g + |f - g|$

Hinging Hyperplanes

Definition 10 (k-Hinge Functions)

$$g(x) = \max\left(w_1^T x + b_1, \ldots, w_k^T x + b_k\right) \tag{21}$$

Theorem 6 ([Wang et.al 2005])

Every continuous piecewise linear function from $\mathbb{R}^n \to \mathbb{R}$ *can be written as a signed sum of* $k-$*Hinges with* $k \leqslant \lceil n+1 \rceil$

$$\sum_i \theta_i g_i(x) \quad \theta_i \in \{\pm 1\} \tag{22}$$

- note: the representation is exact, not an approximation
- re-discovery of k-Hinges: Maxout [Goodfellow et.al 2013]
- note: depth vs. width tradeoff for $|\cdot|$-based representation: every k hinge: can be expressed via $\left\lceil \frac{ln(k+2)}{\ln(2)} \right\rceil$ levels of nesting [[Wang et.al 2005]]

Polyhedral Functions

Convex and continuous PWLs (piecewise linear function). These are also knowns as polyhedral functions.

Definition 11 (Polyhedral Set)

S is polyhedral, if it is a finite intersection of closed half-spaces

$$S = \left\{x \in \mathbb{R}^n : w_j^T x + b_j \geqslant 0,\ j = 1, ..., r\right\} \tag{23}$$

Definition 12 (Epigraph of a Function)

The epigraph of a function is defined as follows:

$$epi\,(f) \triangleq \left\{(x, t) \in \mathbb{R}^{n+1} : f\,(x) \leqslant t\right\} \tag{24}$$

Definition 13 (Polyhedral Function)

f is polyhedral, if $epi(f)$ is a polyhedral set.

Max-Representation of Polyhedral Functions

Proposition 6

For every polyhedral f, there exists $\mathcal{A} \subset \mathbb{R}^{n+1}$, $|\mathcal{A}| = m$ such that

$$f(x) = \max_{(w,b)\in\mathcal{A}} \left\{ w^T x + b \right\} \tag{25}$$

1. polyhedral function = k-Hinges
2. linear functions in $\mathcal{A}$ describe supporting hyperplanes of $epi(f)$
3. cf. more general, (proper) convex case: Fechel duality

$$f(x) = \sup_{w\in\mathbb{R}^n} \left\{ w^T x - f^*(w) \right\} \tag{26}$$

Continuous Piecewise Linear Functions

Theorem 7 ([Wang et.al 2005])

Every continuous piecewise linear function f can be written as the difference of two polyhedral functions.

1. explicitly: there exist finite $\mathcal{A}^+$, $\mathcal{A}^-$ such that

$$f(x) = \max_{(w,b)\in\mathcal{A}^+} \left\{ w^T x + b \right\} - \max_{(w,b)\in\mathcal{A}^-} \left\{ w^T x + b \right\} \tag{27}$$

2. Maxout: max (non-linearity) applied to groups of (linear) functions [Goodfellow et.al 2013]

2× Maxout = Allout

Theorem 8 ([Goodfellow et.al 2013])

Maxout networks with two maxout units are universal function approximators.

Proof.

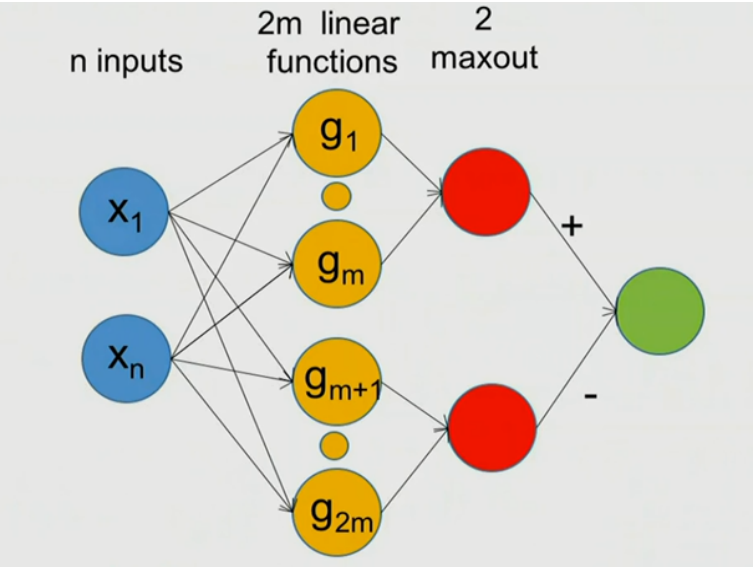
Sketch:

1. Thm. 7: linear network with two maxout units and a linear output unit (subtraction) can represent any continuous PWL function.
2. Continuous PWL functions are dense in $C(\mathbb{R}^n)$

□

1. Most minimalistic use of non-linearity

Minimalistic 2 $\times$ Maxout Network



In practice: more than 2 Maxout units.

Question

What do we know about deep ReLU networks? (Little)

Linear Combinations of Rectified Units

Question:

By linearly combining m rectified units, into how many $(R(m))$ cells is $\mathbb{R}^n$ maximally partitioned?

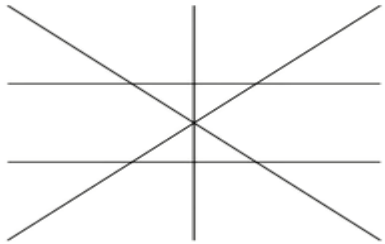


Figure 3: An example: 5 lines 14 cells

Linear Combinations of Rectified Units

Question: By linearly combining m rectified units, into how many ($R(m)$) cells is $\mathbb{R}^n$ maximally partitioned?

Theorem 9 ([Zaslavsky, 1975])

$$R(m) \leqslant \sum_{i=0}^{\min\{m,n\}} \binom{m}{i} \tag{28}$$

1. note that for $m \leq n, R(m) = 2^m$ (exponential growth)
2. for given n, asymptotically, $R(m) \in \Omega(m^n)$,
   - i.e. there is a polynomial slow-down, which is induced by the limitation of the input space dimension

Deep Combinations of Rectified Units

Question: Process n inputs through L ReLU layers with widths $m_1, \ldots, m_L \in O(m)$ Into how many ($R(m, L)$) cells can $\mathbb{R}^n$ be maximally partitioned?

Theorem 10 ([Guido el.al 2014])

$$R(m, L) \in \Omega\left(\left(\frac{m}{n}\right)^{n(L-1)} m^n\right) \tag{29}$$

1. Essentially: for any fixed n, exponential growth can be ensured by making layers sufficiently wide ($m > n$) and increasing the level of functional nesting (i.e. depth L)

Question

What about classical (smooth) activation functions?

Sigmoid Functions

What are "good" activation functions?
Sigmoid activation function: logistic function (or tanh)

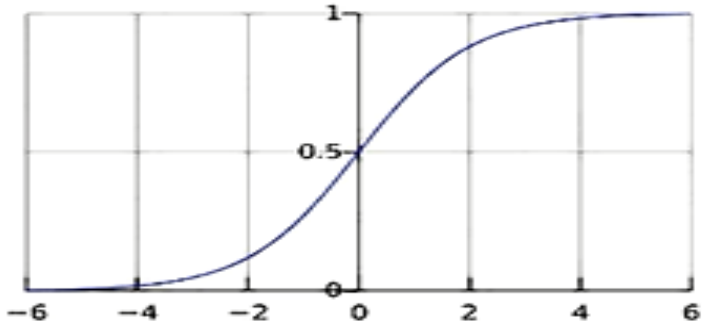


Figure 4: sigmoid function

$$\begin{aligned}
\sigma(t) &= \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t} \in (0,1) \\
\sigma^{-1}(\mu) &= \ln\left(\frac{\mu}{1-\mu}\right) \\
\tanh(t) &= 2\sigma(2t) - 1 \in (-1,1)
\end{aligned} \tag{30}$$

Approximation Theorem

Theorem 11 ([Leshno et.al 1991])

Let $\sigma \in C^{\infty}(\mathbb{R})$, not a ploynomial, then $\mathcal{H}_{\sigma}^{1}$ is dense in $C(\mathbb{R})$

Corollary 3 ([Leshno et.al 1991])

Multi layer Perceptions(MLPs) with one hidden layer and any non-polynomial, smooth activation function are universal function approximators.

Proposition 7 ([Leshno et.al 1991])

Multi layer Perceptions(MLPs) with one hidden layer and any polynomial, activation function are not universal function approximators.

Remark 4

Smoothness requirement can be substantially weakened.
See previous results on rectified activation functions.

Approximation theorem

To prove Thms in page 38, also need:

Theorem 12 ([Corominas et.al 1954, Donoghue,1969])

If σ is C^∞ on (a, b) and it is not a polynomial, then there exists a point $\theta_0 \in (a, b)$ such that $\sigma^k(\theta_0) \neq 0 \quad \forall k$

Question

What can we say about the size of the representation (networks)?

Sigmoid MLP: Approximation Guarantees

Theorem 13 ([Barron, 1993])

$\forall f : \mathbb{R}^n \to \mathbb{R}$ with absolutely continuous Fourier transform and for every m there is a function of the form $\widetilde{f}_m$ such that:

$$\int_{B_r} \left(f(x) - \widetilde{f}(x) \right)^2 \mu(dx) \leqslant O\left(\frac{1}{m}\right) \tag{31}$$

where $B_r = \{x \in \mathbb{R}^n : \|x\| \leqslant r\}$ and μ is any probability measure on B_r.

Remark 5

1. *most remarkably, the residual bound does not depend on the input dimensional n*
2. *proof uses iterative process of adding units*

Sigmoid MLP: Approximation Guarantees

We will now state and discuss without proof (a simplified) version of the famous result of [Barron, 1993], which relates the residual to the number of sigmoidal neurons in the (single) hidden layer. Consider a multi-layer perceptron, where:

$$\widetilde{f}_m(x) = \sum_{j=1}^{m} \alpha_j \sigma\left(w_j^T x + b_j\right) + \beta \tag{32}$$

where σ is a bounded (measurable) and monotonic function such that $\sigma(t) \overset{t\to\infty}{\to} 1$ and $\sigma(t) \overset{t\to-\infty}{\to} 0$. ([Barron, 1993] version of "sigmoid")

References

Corominas and F. Sunyer Balaguer (1954)

Conditions for a purely derivable function to be a polynomial

Spanish-American Mathematical Magazine Vol.14, 26 – 43.

W. Donoghue (1969)

Distributions and Fourier Transforms

Academic Press 1969

Allan Pinkus (1999)

Approximation theory of the MLP model in neural networks

Acta numerica Vol.8, 143 – 195.

A. Barron (1993)

Universal approximation bounds for superpositions of a sigmoidal function

IEEE Transactions on Information Theory Vol.39(3), 930 – 945.

Allan Pinkus (2000)

Universal approximation bounds for superpositions of a sigmoidal function

Journal of Approximation Theory Vol.107(1), 1 – 66.

References

Boris Shekhtman (1982)

Why piecewise linear functions are dense in C[0, 1]

Journal of Approximation Theory Vol.36(3), 265 – 267.

Henri Lebesgue (1898)

Sur l'approximation des fonctions

Bulletin des Sciences Mathématiques Vol.22, 278 – 287.

Thomas Zaslavsky (1975)

Counting the faces of cut-up spaces

Bulletin of the American Mathematical Society Vol.81

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho and Yoshua Bengio (2014)

On the number of linear regions of deep neural networks

NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems Vol.2, 2924 – 2932.

References

Leo Breiman (1993)

Hinging hyperplanes for regression, classification, and function approximation

IEEE Transactions on Information Theory Vol.39(3), 999 – 1013.

Shuning Wang and Xusheng Sun (2005)

Generalization of hinging hyperplanes

IEEE Transactions on Information Theory Vol.51(12), 4425 – 4431.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville and Yoshua Bengio (2013)

Maxout Networks

Proceedings of Machine Learning Research Vol.28(3), 1319 – 1327

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, Shimon Schocken (1991)

Multilayer feedforward networks with a nonpolynomial activation function can approximate any function

Neural Networks Vol.6, 861 – 867

Last But Not Least

Thank you all of you! –Yao