Theory I: Why Deep Structure?
Given structure, each set of parameter is a function.

The network structure defines a function set.
$f(x) = 2(2\cos^2(x) - 1)^2 - 1$

Source of image:
Outline

• Q1: Can shallow network fit any function?
• Potential of deep
• Q2: How to use deep to fit functions?
• Q3: Is deep better than shallow?
• Review some related theories

Scalar $x$ [0, 1] → NN → Scalar $y$

ReLU as activation function
Notice: We do not discuss **optimization** and **generation** today.

1. **Shallow**
   
   Eventually cover $\hat{\theta}$?

2. **Deep**

3. What is the difference?

   A target function to fit e.g. $y = x^2$
Can shallow network fit any function?
Universality

- Given a shallow network structure with one hidden layer with ReLU activation and linear output

A piece-wise linear functions

- Given a L-Lipschitz function $f^*$
  - How many neurons are needed to approximate $f^*$?
Universality

• Given a $L$-Lipschitz function $f^*$
  • How many neurons are needed to approximate $f^*$?

$L$-Lipschitz Function (smooth)

\[ \|f(x_1) - f(x_2)\| \leq L \|x_1 - x_2\| \]

Output change
Input change

$L=1$ for "1 − Lipschitz"
Universality

- Given a L-Lipschitz function $f^*$
  - How many neurons are needed to approximate $f^*$?

$$f \in N(K) \quad \text{The function space defined by the network with } K \text{ neurons.}$$

Given a small number $\varepsilon > 0$

What is the number of $K$ such that

$$\exists f \in N(K), \quad \max_{0 \leq x \leq 1} |f(x) - f^*(x)| \leq \varepsilon$$

The difference between $f(x)$ and $f^*(x)$ is smaller than $\varepsilon$. 
Universality

- L-Lipschitz function $f^*$

All the functions in $N(K)$ are piecewise linear.

Approximate $f^*$ by a piecewise linear function $f$

How to make the errors $\leq \varepsilon$

$$l \times L \leq \varepsilon \quad l \leq \varepsilon / L$$

$$\|f(x_1) - f(x_2)\| \leq L \|x_1 - x_2\|$$
Universality

• L-Lipschitz function $f^*$

How to make a 1 hidden layer relu network have the output like green curve?
The summation of the blue functions is the green one.

Each blue function can be obtained by two relu neurons.
I do not say this is the most efficient way to use the neurons.
Potential of deep
Why we need deep?

Yes, shallow network can represent any function.

However, using deep structure is more effective.
Analogy – Programming

• Solve any problem by two lines (shallow)
  • Input = K
  • Line 1: row no. = MATCH_KEY(K)
  • Line 2: Output the value at row no.

<table>
<thead>
<tr>
<th>Input (key)</th>
<th>Output (value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A’</td>
</tr>
<tr>
<td>B</td>
<td>B’</td>
</tr>
<tr>
<td>C</td>
<td>C’</td>
</tr>
<tr>
<td>D</td>
<td>D’</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

• Considering SVM with kernel

\[ y = \sum_{n} \alpha_n K(x^n, x) \]

• Using multiple steps to solve problems is more efficient (deep)
Analogy

**Logic circuits**

- Logic circuits consists of gates
- A two layers of logic gates can represent any Boolean function.
- Using multiple layers of logic gates to build some functions are much simpler

**Neural network**

- Neural network consists of neurons
- A hidden layer network can represent any continuous function.
- Using multiple layers of neurons to represent some functions are much simpler

This page is for EE background.
Analogy

- E.g. **parity check**
  
  For input sequence with $d$ bits,
  
  Two-layer circuit need $O(2^d)$ gates.
  
  With multiple layers, we need only $O(d)$ gates.
Why we need deep?

- ReLU networks can represent piecewise linear functions.

Shallow & wide \( \approx \) the same number of parameters Deep & Narrow

Less pieces

More pieces
Upper Bound of Linear Pieces

Each “activation pattern” defines a linear function.

N neurons $\rightarrow 2^N$ “activation patterns” $\rightarrow 2^N$ “linear pieces”
Upper Bound of Linear Pieces

• Not all the “activation patterns” available

In shallow network, each neuron only provides one linear piece.
Abs Activation Function

Use two relu to implement an abs activation function

$$\text{Abs Activation Function}$$

$$x \rightarrow w \rightarrow + \rightarrow b \rightarrow |wx + b|$$

$$x \rightarrow 1 \rightarrow + \rightarrow w \rightarrow + \rightarrow b \rightarrow 1 \rightarrow + \rightarrow -w \rightarrow + \rightarrow -b \rightarrow 1 \rightarrow + \rightarrow -wx - b$$
\[
\begin{align*}
&x \rightarrow + \rightarrow a_1 \rightarrow + \rightarrow a_2 \\
&2^1 \text{ lines} \\
&2^2 \text{ lines}
\end{align*}
\]
Each node added  ➔  The regions are twice.
Shallow

Deep
Lower Bound of Linear Pieces

If $K$ is width, $H$ is depth

We can have at least $K^H$ pieces

Depth has much larger influence than depth.

Raman Arora, Amitabh Basu, Poorya Mianjy, Anirbit Mukherjee, “Understanding Deep Neural Networks with Rectified Linear Units”, ICLR 2018
Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl-Dickstein, On the Expressive Power of Deep Neural Networks, ICML, 2017
Experimental Results (MNIST)
How much is deep better than shallow?
Fit the function by equally spaced linear pieces

\[ f_m(x): \text{a function with } 2^m \text{ pieces} \]

\[ \max_{0 \leq x \leq 1} |f(x) - f_m(x)| \leq \varepsilon \]

What is the minimum \( m \)?

\[ m \geq -\frac{1}{2} \log_2 \varepsilon - 1 \]

\[ 2^m \geq \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \] pieces

Shallow: \( O \left( \frac{1}{\sqrt{\varepsilon}} \right) \) neurons
\[ f(x) = x^2 \]
$f_m(x) = \begin{cases} 
1 & \text{for } x \geq -\frac{1}{2}\log_2\varepsilon - 1 \\
\text{otherwise} & 
\end{cases}$

$m \geq -\frac{1}{2}\log_2\varepsilon - 1$

$O(m)$ neurons $O(m)$ layers

$O\left(\log_2\frac{1}{\sqrt{\varepsilon}}\right)$ neurons $O\left(\log_2\frac{1}{\sqrt{\varepsilon}}\right)$ layers

$2^m$ pieces
Why care about $y = x^2$?

Square Net

$x \rightarrow x^2 \leq \varepsilon$

$y = x_1 x_2 = \frac{1}{2} ((x_1 + x_2)^2 - x_1^2 - x_2^2)$

Multiply Net

$O \left( \log_2 \frac{1}{\sqrt{\varepsilon}} \right)$ neurons

$O \left( \log_2 \frac{1}{\sqrt{\varepsilon}} \right)$ neurons
Polynomial

\[ y = x^n \]

**Power(n) Net**

\[ y = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0 \]

Use polynomial function to fit other functions.
Deep v.s. Shallow

This is not sufficient to show the power of deep.
Is Deep better than Shallow?
Best of Shallow

- A relu network is a piecewise linear function.
- Using the least pieces to fit the target function.

\[
\begin{align*}
\max_{0 \leq x \leq 1} |f(x) - f^*(x)| & \leq \varepsilon \\
\sqrt{\int_0^1 |f(x) - f^*(x)|^2 \, dx} & \leq \varepsilon
\end{align*}
\]

*Use Euclidean*

The lines do not have to connect the end points.
Best of Shallow

• Given a piece, what is the smallest error

\[
\sqrt{\int_0^1 |f(x) - f^*(x)|^2 \, dx} \leq \varepsilon
\]

Use Euclidean

\[
e^2 = \int_{x_0}^{x_0+l} (x^2 - (ax + b))^2 \, dx
\]

Find \(a\) and \(b\) to minimize \(e^2\)

The minimum value of \(e^2\) is \(\frac{l^5}{180}\)
Warning of Math
Intuition

\[ e^2 = \int_{x_0}^{x_0+l} \left( x^2 - (ax + b) \right)^2 \, dx \]

\[ f_v = x^2 \quad f_w = x \quad f_u = 1 \]

Minimize
\[ ||\vec{v} - (a\vec{w} + b\vec{u})||^2 \]

Minimize
\[ ||f_v - (af_w + bf_u)||^2 \]
End of Warning
If you have \( n \) pieces, what is the best way to arrange the \( n \) pieces.

\[
\sum_{i=1}^{n} l_i = 1
\]

The best way is “equal segment” with 

\[
l_i = 1/n
\]

The minimum value of \( e^2 \) is \( \frac{l^5}{180} \)

\[
E^2 = \sum_{i=1}^{n} \frac{(1/n)^5}{180} = \frac{1}{180} \frac{1}{n^4}
\]
Warning of Math
Hölder's inequality

\[ \sum^n_{i=1} l_i = 1 \]

Minimize \( \sum^n_{i=1} (l_i)^5 \)

- Given \( \{a_1, a_2, \ldots, a_n\} \) and \( \{b_1, b_2, \ldots, b_n\} \)

\[ \left( \sum^n_{i=1} |a_i|^p \right)^{1/p} \left( \sum^n_{i=1} |b_i|^q \right)^{1/q} \leq \left( \sum^n_{i=1} |a_i b_i| \right) \leq \left( \sum^n_{i=1} |a_i|^p \right)^{1/p} \left( \sum^n_{i=1} |b_i|^q \right)^{1/q} \]

- Given \( \{l_1, l_2, \ldots, l_n\} \) and \( \{1,1, \ldots, 1\} \)

\[ n^{-1/q} \leq \left( \sum^n_{i=1} l_i^p \right)^{1/p} \leq \left( \sum^n_{i=1} l_i \right)^{1/p} \]

\[ n^{-p/q} \leq \left( \sum^n_{i=1} l_i^p \right)^{1/p} \leq \left( \sum^n_{i=1} l_i \right)^{1/p} \]

\[ 1 - p \]

\[ n^{-p/q} \leq \sum^n_{i=1} l_i^p \leq n^{-4} \leq \sum^n_{i=1} l_i^5 \]
End of Warning
Best of Shallow

• If you have $n$ pieces, what is the best way to arrange the $n$ pieces.

$$E^2 = \frac{1}{180} \frac{1}{n^4} \quad \Rightarrow \quad E = \sqrt[4]{\frac{1}{180} \frac{1}{n^2}}$$

To make $E \leq \varepsilon$, what is the $n$ we need?

$$E = \sqrt{\frac{1}{180} \frac{1}{n^2}} \leq \varepsilon \quad n^2 \geq \sqrt{\frac{1}{180} \frac{1}{\varepsilon}} \quad n \geq 4 \sqrt{\frac{1}{180} \sqrt{\frac{1}{\varepsilon}}}$$

The minimum value of $e^2$ is $\frac{l^5}{180}$

At least $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$ neurons
Deep v.s. Shallow

Deep is exponentially better than shallow.

Shallow neurons \( O \left( \frac{1}{\sqrt{\varepsilon}} \right) \) neurons

Deep neurons \( O \left( \log_2 \frac{1}{\varepsilon} \right) \) neurons

獵人第二十卷

(獵人第二十卷)
More related theories
More Theories

• A function expressible by a 3-layer feedforward network cannot be approximated by 2-layer network.
  • Unless the width of 2-layer network is VERY large
  • Applied on activation functions beyond relu

The width of 3-layer network is $K$.

The width of 2-layer network should be $Ae^{BK^{4/19}}$.

Ronen Eldan, Ohad Shamir, “The Power of Depth for Feedforward Neural Networks”, COLT, 2016
More Theories

• A function expressible by a deep feedforward network cannot be approximated by a shallow network.
  • Unless the width of the shallow network is VERY large
  • Applied on activation functions beyond relu

Deep Network:
\[ \Theta(k^3) \text{ layers, } \Theta(1) \text{ nodes per layer, } \Theta(1) \text{ distinct parameters} \]

Shallow Network: \[ \Theta(k) \text{ layers } \rightarrow \Omega(2^k) \text{ nodes} \]

Itay Safran, Ohad Shamir, “Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks”, ICML, 2017
More Theories

Dmitry Yarotsky, “Optimal approximation of continuous functions by very deep ReLU networks”, arXiv 2018
Itay Safran, Ohad Shamir, “Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks”, ICML, 2017

If a function $f$ has “certain degree of complexity”

Approximating $f$ to accuracy $\varepsilon$ in the L2 norm using a fixed depth ReLU network requires at least $\text{poly}(1/\varepsilon)$

There exist a ReLU network of depth and width at most $\text{poly}(\log(1/\varepsilon))$ that can achieve the approximation.
The Nature of Functions

Hrushikesh Mhaskar, Qianli Liao, Tomaso Poggio, When and Why Are Deep Networks Better Than Shallow Ones?, AAAI, 2017
Concluding Remarks