Why Deep Learning?
Deeper is Better?

<table>
<thead>
<tr>
<th>Layer X Size</th>
<th>Word Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X 2k</td>
<td>24.2</td>
</tr>
<tr>
<td>2 X 2k</td>
<td>20.4</td>
</tr>
<tr>
<td>3 X 2k</td>
<td>18.4</td>
</tr>
<tr>
<td>4 X 2k</td>
<td>17.8</td>
</tr>
<tr>
<td>5 X 2k</td>
<td>17.2</td>
</tr>
<tr>
<td>7 X 2k</td>
<td>17.1</td>
</tr>
</tbody>
</table>

Not surprised, more parameters, better performance

Fat + Short v.s. Thin + Tall

The same number of parameters

Which one is better?

Shallow

Deep
Fat + Short v.s. Thin + Tall

<table>
<thead>
<tr>
<th>Layer X Size</th>
<th>Word Error Rate (%)</th>
<th>Layer X Size</th>
<th>Word Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X 2k</td>
<td>24.2</td>
<td>1 X 3772</td>
<td>22.5</td>
</tr>
<tr>
<td>2 X 2k</td>
<td>20.4</td>
<td>1 X 4634</td>
<td>22.6</td>
</tr>
<tr>
<td>3 X 2k</td>
<td>18.4</td>
<td>1 X 16k</td>
<td>22.1</td>
</tr>
<tr>
<td>4 X 2k</td>
<td>17.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 X 2k</td>
<td>17.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 X 2k</td>
<td>17.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Modularization

- Deep → Modularization

Don’t put everything in your main function.

http://rinuboney.github.io/2015/10/18/theoretical-motivations-deep-learning.html
Modularization

• Deep → Modularization

Image

Classifier 1

Classifier 2

Classifier 3

Classifier 4

Girls with long hair

Boys with long hair

Girls with short hair

Boys with short hair

Little examples

Weak
Modularization

- Deep → Modularization

Each basic classifier can have sufficient training examples.

Classifiers for the attributes:

- Boy or Girl?
- Long or short?

Image

Basic Classifier
Modularization

• Deep → Modularization

Sharing by the following classifiers as module

Classifier 1
Girls with long hair

Classifier 2
Boys with long hair

Classifier 3
Girls with short hair

Classifier 4
Boys with short hair

Image

Basic Classifier

Boy or Girl?

Long or short?
Modularization

• Deep → Modularization → Less training data?

The modularization is automatically learned from data.

The most basic classifiers

Use 1st layer as module to build classifiers

Use 2nd layer as module ......
Modularization - Image

- Deep → Modularization

The most basic classifiers to build classifiers

Use 1st layer as module

Use 2nd layer as module

Modularization - Speech

• The hierarchical structure of human languages

what do you think?

Phoneme:

hh w aa t d uw y uw th ih ng k

Tri-phone:

...... t-d+uw d-uw+y uw-y+uw y-uw+th ......

t-d+uw1 t-d+uw2 t-d+uw3 d-uw+y1 d-uw+y2 d-uw+y3

State:
Modularization - Speech

• The first stage of speech recognition
  • Classification: input → acoustic feature, output → state

Determine the state each acoustic feature belongs to

States: a a a b b c c
Modularization - Speech

- Each state has a stationary distribution for acoustic features

Gaussian Mixture Model (GMM)

\[ P(x|"t-d+uw1") \]

\[ P(x|"d-uw+y3") \]
Modularization - Speech

• Each state has a stationary distribution for acoustic features

\[ P(x|"d-uw+y3") \]

\[ P(x|"y-uw+th3") \]

*Tied-state*

Same Address
Modularization - Speech

• In HMM-GMM, all the phonemes are modeled independently
  • Not an effective way to model human voice

The sound of vowel is only controlled by a few factors.

http://www.ipachart.com/
Modularization - Speech

- **DNN input:**
  - One acoustic feature

- **DNN output:**
  - Probability of each state

Size of output layer = No. of states

All the states use the same DNN
Modularization


Output of hidden layer reduce to two dimensions

- The lower layers detect the manner of articulation
- All the phonemes share the results from the same set of detectors.
- Use parameters effectively
Universality Theorem

Any continuous function $f$

$$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

Can be realized by a network with one hidden layer (given enough hidden neurons)

Reference for the reason:

Yes, shallow network can represent any function.

However, using deep structure is more effective.
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

less gates needed  less parameters  less data?

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.
More Analogy

手工制作 [www.xuhan.org]
More Analogy

Folding the space

Use data effectively

Folding the space

Graph and chart showing data points and relationships between variables.
More Analogy - Experiment

Different numbers of training examples

10,0000

2,0000

$\mathbf{f : R^2 \rightarrow \{0, 1\}}$

1 hidden layer

3 hidden layers
End-to-end Learning

• Production line

Model

Hypothesis Functions

Simple Function 1 → Simple Function 2 → …… → Simple Function N → “Hello”

A very complex function

End-to-end training:
What each function should do is learned automatically
End-to-end Learning - Speech Recognition

• Shallow Approach

Waveform → DFT → Waveform

“Hello” → GMM → DCT → log → Filter bank

Each box is a simple function in the production line:

- hand-crafted
- learned from data
End-to-end Learning - Speech Recognition

• Deep Learning

```
<table>
<thead>
<tr>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
</tr>
</thead>
</table>
```

"Hello"

Less engineering labor, but machine learns more
End-to-end Learning - Image Recognition

• Shallow Approach

http://www.robots.ox.ac.uk/~vgg/research/encoding_eval/

:hand-crafted :learned from data
End-to-end Learning
- Image Recognition

• Deep Learning

Complex Task ...

- Very similar input, different output
  - System \(\rightarrow\) dog \(\rightarrow\) System \(\rightarrow\) bear

- Very different input, similar output
  - System \(\rightarrow\) train \(\rightarrow\) System \(\rightarrow\) train
Complex Task ...

• Speech recognition: Speaker normalization is automatically done in DNN


Input Acoustic Feature (MFCC) 1-st Hidden Layer
Complex Task ...

• Speech recognition: Speaker normalization is automatically done in DNN


Input Acoustic Feature (MFCC)  

8-th Hidden Layer
MNIST

input

1-st hidden

2-nd hidden

3-rd hidden
To learn more ...

- Do Deep Nets Really Need To Be Deep? (by Rich Caruana)

keynote of Rich Caruana at ASRU 2015
To learn more ...

• Deep Learning: Theoretical Motivations (*Yoshua Bengio*)
  • http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

• Connections between physics and deep learning
  • https://www.youtube.com/watch?v=5MdSE-NObxs

• Why Deep Learning Works: Perspectives from Theoretical Chemistry
  • https://www.youtube.com/watch?v=kIbKHIPbxiU