Deep Learning
Tutorial and Applications

Artit “Art” Wangperawong
artitw@gmail.com
Some things you can ask me:

“Did the Giants win?”
“Game schedules”
“My mom is Susan Conway”
“Read it again”
“Delete my 7:30 alarm”

http://www.imore.com/siri
A

B

C

D

E

F

https://www.boredpanda.com/computer-deep-learning-algorithm-painting-masters/
Sensor representation in the brain

[Roe et al., 1992; BrainPort; Welsh & Blasch, 1997]
Deep Learning – Simulating the Brain

Neuron in the brain

Computer learns to read handwriting
2D pixel intensity input data

28x28 = 784
Input and output from a neuron

Learning algorithm #1:

\[
\text{output} = \begin{cases} 
0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\
1 & \text{if } \sum_j w_j x_j > \text{threshold}
\end{cases} \quad (1)
\]

\[
\text{output} = \begin{cases} 
0 & \text{if } w \cdot x + b \leq 0 \\
1 & \text{if } w \cdot x + b > 0
\end{cases} \quad (2)
\]
Network of neurons = neural network
Train a neural network – make it learn

small change in any weight (or bias) causes a small change in the output
Learning algorithm #2:

But the output is not 0 or 1. Instead, it's $\sigma(w \cdot x + b)$, where $\sigma$ is called the sigmoid function*, and is defined by:

$$
\sigma(z) \equiv \frac{1}{1 + e^{-z}}.
$$

(3)

To put it all a little more explicitly, the output of a sigmoid neuron with inputs $x_1, x_2, \ldots$, weights $w_1, w_2, \ldots$, and bias $b$ is

$$
\frac{1}{1 + \exp(-\sum_j w_j x_j - b)}.
$$

(4)
Sigmoid Neurons

\[ \sigma(z) \equiv \frac{1}{1 + e^{-z}} \]
Simple neural network for OCR

504192
How does it work?
SO FAR NOT DEEP, STILL SHALLOW
Hierarchical breakdown of features

Input pixels

- An eye at the top-left?
- Eye at the top-right?
- Nose in the middle?
- Mouth below?
- Hair on top?

Is human?
Hierarchical breakdown of features

- Eye at top-left?
- Has eyebrow?
- Has eyelashes?
- Has iris?
Deep Learning learns layers of features

http://www.andrewng.org/
Deep neural network (DNN)
Convolution – feature mapping


Artit Wangprapanich
Pooling – aggregate and reduce

Single depth slice

max pool with 2x2 filters and stride 2

DNN + conv + pooling = Deep Learning

https://www.clarifai.com
How well does DL work?

<table>
<thead>
<tr>
<th>MNIST</th>
<th>Simple NN</th>
<th>DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>96.59</td>
<td>99.67</td>
</tr>
</tbody>
</table>

![MNIST Images]
DL ON TELCO
Target Data Selection

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Implied date range</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 day churn window start</td>
<td>24 Sep 2015</td>
<td>24 Sep - 23 Oct</td>
</tr>
<tr>
<td>Last call window</td>
<td>14 days</td>
<td>10 Sep – 23 Sep</td>
</tr>
<tr>
<td>Time before last call</td>
<td>14 days</td>
<td>27 Aug – 23 Sep</td>
</tr>
<tr>
<td>Predictor window</td>
<td>30 days</td>
<td>28 Jul – 9 Sep</td>
</tr>
<tr>
<td>Minimum age on network</td>
<td>90 days</td>
<td>&lt; 28 Jul</td>
</tr>
</tbody>
</table>
Telco churn DL – customer as image

Label:

- Row 1 = Day 1
- Row 2 = Day 2
- ... 
- Row n = Day n

Data usage, SMS in, voice out, etc.

Artit Wangperawong
One architecture: DL-1

- 30x10
- Convolutional layer 1: 4x7x1
- Convolutional layer 2: 2x1x10
- Fully connected layer: 128
- Pooling: 2x1
- Softmax output
One architecture: DL-1

Convolution - 4 filters
7x1

Convolution - 2 filters
1x10

Fully Connected

Fully Connected

ReLU
ReLU
ReLU

SOFTMAX

ReLu

Artit Wangperawong
DL versus SPSS

<table>
<thead>
<tr>
<th>AUC</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSS CHAID</td>
<td>0.699</td>
<td>0.665</td>
</tr>
<tr>
<td>DL-1</td>
<td>0.751</td>
<td>0.706</td>
</tr>
<tr>
<td>DL-2</td>
<td>0.748</td>
<td>0.743</td>
</tr>
</tbody>
</table>
Autoencoder – unsupervised learning

\[
a_i^{(2)} = f \left( \sum_{j=1}^{100} W_{ij}^{(1)} x_j + b_i^{(1)} \right).
\]

\[
x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{100} (W_{ij}^{(1)})^2}}.
\]

http://deeplearning.stanford.edu/wiki/index.php/Visualizing_a_Trained_Autoencoder

Artit Wangperawong
Images that maximally activate hidden units of autoencoder for all customers

Calls+SMS OT  |  Beg&End  |  SMS MT  |  Data  

Data conn on second half
Images that maximally activate the weights of the autoencoder for churn

- General usage dropoff
- Strong data usage
- Pure data user
Customer who churned

1) Very high usage of data and only data
2) Usage of the phone only from 14-16th days
3) Almost no usage during the 30 days period
Churn analysis using deep convolutional neural networks and autoencoders

Artit Wangperawong\textsuperscript{a,b,*}  
Cyrille Brun\textsuperscript{c}  
Dr. Olav Laudy\textsuperscript{c}  
Rujikorn Pavasuthipaisit\textsuperscript{a}

\textsuperscript{a} True Corporation Public Company Limited, Bangkok 10310, Thailand  
\textsuperscript{b} Department of Electrical Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand  
\textsuperscript{c} International Business Machines Corporation, Singapore 486048

* To whom correspondence should be addressed; Email: artitw@gmail.com

Customer temporal behavioral data was represented as images in order to perform churn prediction by leveraging deep learning architectures prominent in image classification. Supervised learning was performed on labeled data of over 6 million customers using deep convolutional neural networks, which achieved an AUC of 0.743 on the test dataset using no more than 12 temporal features for each customer. Unsupervised learning was conducted using autoencoders to better understand the reasons for customer churn. Images that maximally activate the hidden units of an autoencoder trained with churned customers reveal ample opportunities for action to be taken to prevent churn among strong data, no voice users.

Keywords: machine learning, deep learning, big data, churn prediction, telecommunications

Deep learning by convolutional neural networks (CNNs) has demonstrated superior performance in many image processing tasks [1,2,3]. In order to leverage such advances to predict churn and take pro-active measures to prevent it, we represent customers as images. Specifically, we construct a 2-dimensional array of normalized pixels where each row is for each day and each column is for each type of behavior tracked (Fig. 1). The type of behavior can include data usage, top up amount, top up frequency, voice calls, voice minutes, SMS messages, etc. In the training and testing data, each image is also accompanied by its label – 1 for churned and 0 for not churned. For this analysis, we examine prepaid customers in particular.