Deep Learning in Action
Table 3. Test error rates for stock price prediction

<table>
<thead>
<tr>
<th>Records</th>
<th>Ranker</th>
<th>SVM</th>
<th>DBN</th>
<th>BNN-DD</th>
<th>M + DBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sony</td>
<td>43.97</td>
<td>48.73</td>
<td>45.58</td>
<td>45.82</td>
<td></td>
</tr>
<tr>
<td>Hitachi</td>
<td>35.71</td>
<td>57.29</td>
<td>32.60</td>
<td>32.88</td>
<td></td>
</tr>
<tr>
<td>Toshiba</td>
<td>39.52</td>
<td>41.95</td>
<td>35.58</td>
<td>36.50</td>
<td></td>
</tr>
<tr>
<td>Fujitsu</td>
<td>40.80</td>
<td>40.25</td>
<td>32.09</td>
<td>34.80</td>
<td></td>
</tr>
<tr>
<td>Sharp</td>
<td>42.10</td>
<td>47.38</td>
<td>40.08</td>
<td>40.90</td>
<td></td>
</tr>
<tr>
<td>Sony</td>
<td>43.80</td>
<td>47.40</td>
<td>41.43</td>
<td>40.95</td>
<td></td>
</tr>
<tr>
<td>Nissan</td>
<td>40.90</td>
<td>45.54</td>
<td>37.58</td>
<td>37.90</td>
<td></td>
</tr>
<tr>
<td>Toyota</td>
<td>44.29</td>
<td>53.30</td>
<td>43.01</td>
<td>42.90</td>
<td></td>
</tr>
<tr>
<td>Canon</td>
<td>43.81</td>
<td>52.39</td>
<td>43.08</td>
<td>40.21</td>
<td></td>
</tr>
<tr>
<td>Mitsu</td>
<td>40.95</td>
<td>47.85</td>
<td>41.43</td>
<td>41.43</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>43.81</td>
<td>49.15</td>
<td>45.33</td>
<td>48.43</td>
<td></td>
</tr>
</tbody>
</table>

Sources: [1],[3],[4],[5],[6]
So what is a neural network?

Biological neuron and artificial neuron

Source: [10]
Prototype of a neuron: the perceptron (Rosenblatt, 1958)

Source: [10]
Deep neural networks: introducing hidden layers

Source: [10]
A deep representation is a composition of many functions

\[ x \xrightarrow{w_1} h_1 \xrightarrow{w_2} h_2 \xrightarrow{w_3} \ldots \xrightarrow{w_n} h_n \xrightarrow{w_{n+1}} y \]
Why go deep? A bit of background

Easy? Difficult?

- walk
- talk
- play chess
- solve matrix computations
Easy for us - difficult for computers

- controlled movement
- speech recognition
- speech generation
- object recognition and object localization
Representation matters
Just feed the network the right features?

What are the correct pixel values for a “bike” feature?

- race bike, mountain bike, e-bike?
- pixels in the shadow may be much darker
- what if bike is mostly obscured by rider standing in front?
Let the network pick the features
... a layer at a time

Source: [12]
So how does a network learn?

Just a sec - let's meet a real neural network first!

Play around in the browser:

- ConvNetJS
- TensorFlow playground
So how DOES a neural network learn?

We need:

- a way to quantify our current (e.g., classification) error
- a way to reduce error on subsequent iterations
- a way to propagate our improvement logic from the output layer all the way back through the network!
Quantifying error: Loss functions

The loss (or cost) function indicates the cost incurred from false prediction / misclassification.

Probably the best-known loss function in machine learning is **mean squared error**:

\[
\frac{1}{n} \sum_n (\hat{y} - y)^2
\]

Most of the time, in deep learning we use **cross entropy**:

\[
- \sum_j t_j \log(y_j)
\]

This is the negative log probability of the right answer.
Learning from errors: Gradient Descent

Global minimum at $x = 0$. Since $f'(x) = 0$, gradient descent halts here.

For $x < 0$, we have $f'(x) < 0$, so we can decrease $f$ by moving rightward.

For $x > 0$, we have $f'(x) > 0$, so we can decrease $f$ by moving leftward.

Source: [12]
Propagate back errors ... well: Backpropagation!

- basically, just the chain rule: \( \frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \)
- chained over several layers:

Source: [14]
Forward pass and backward pass: Intuition

- imagine output of $f = (x + y) \ast z = -12$ “wants” to get bigger
- this could happen by $q$ getting smaller $\rightarrow q$ receives negative gradient $\frac{df}{dq} = -4$
- $q$ just passes on this gradient to $x$ and $y$, as $\frac{dq}{dx} = 1$ and $\frac{dq}{dy} = 1$
- alternatively, it could happen by $z$ getting bigger $\rightarrow z$ receives positive gradient $\frac{df}{dz} = 3$

Source: [13]
Applications by example

- CNNs (Convolutional Neural Networks) for computer vision
- RNNs (Recurrent Neural Networks) for Natural Language Processing
- Deep Reinforcement Learning for real-life learning
Easy vs. hard, revisited

VISION
Why computer vision is hard

Figure 1. The deformable and truncated cat. Cats exhibit (almost) unconstrained variations in shape and layout. The cat examples shown here are detected by our Distinctive Part Model, but missed by the template based method of [11].

Source: [15]
Tasks in computer vision

Source: [13]
In classification, the required output is a probability for each class.

Source: [13]
In localization, the network needs to identify the position of an object in an image.

Source: [17]
In object detection (a.k.a. image recognition), the network has to classify and localize multiple objects in an image.

Source: [18]
In segmentation, the network needs to predict a class value for each input pixel.
Semantic vs. Instance Segmentation

Semantic segmentation segments by *class*, instance segmentation by *class instance*.

Source: [20]
How do we identify the required features? Enter:

**Convolutional Neural Networks**
A convolutional neural network

Source: [13]
The convolution operation

(Strictly, this is \textit{cross-correlation}, but it doesn't matter)

Source: [13]
Gimp demo

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 5 & -1 \\
0 & -1 & 0 \\
\end{bmatrix}, \quad
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

see: https://docs.gimp.org/en/plug-in-convmatrix.html
Easy vs. hard, revisited

VISION
LANGUAGE
Until now, all we’ve seen are static snapshots

How do we handle sequences

- language: words, sentences, paragraphs...
- all kinds of *serial* information: sensor data, stock prices...

?  

Jane walked into the room. John walked in too. It was late in the day, and everyone was walking home after a long day at work. Jane said hi to ___

Source: [21]
How do we remember the past? Enter:

Recurrent neural networks
Hidden state

Sources: [22], [12]
The basic RNN at every step combines new input and existing state.

Source: [22]
Remembering is not enough

Sometimes we also need to forget!
Two kinds of state: the LSTM "conveyor belt"

The LSTM (Long Term Short Memory) architecture adds an additional state layer, the cell state.

Source: [22]
The LSTM cell state is protected by three gates, the forget, input, and output gates:

Source: [22]
In translation, we have two sets of sequential data, one on the source and one on the target side!

Enter: sequence-to-sequence models

Source: [23]
Real-life seq-2-seq: Google’s Neural Machine Translation System

Source: [24]
Easy vs. hard, revisited

VISION
LANGUAGE
VISION <-> LANGUAGE
Generating image captions

Source: [26]
Easy vs. hard, revisited

VISION
LANGUAGE
VISION <-> LANGUAGE
VISION <-> SOUND
Linking video and sound: adding audio to silent film

Source: [27]
Easy vs. hard, revisited

VISION
LANGUAGE
VISION <-> LANGUAGE
VISION <-> SOUND
LIFE (SORT OF)
@Neuro_Skeptic I say we build a Dopamine wall. And make neuroscientists pay for it.  
#makebrainsgreatagain
Reinforcement learning

Types of machine learning:

Supervised Learning
Unsupervised Learning
Semi-Supervised Learning
Reinforcement Learning

Source: [1]
Reinforcement learning: the task

- reward +1 at [4,3], -1 at [4,2]
- reward -0.04 for each step

actions: UP, DOWN, LEFT, RIGHT

- UP
  - 80% move UP
  - 10% move LEFT
  - 10% move RIGHT

Source: [1]
Reinforcement learning: The problem

If I get a reward many many actions later...

... how do I find out what concrete action I'm getting the reward for?
Reinforcement learning: The dilemma

EXPLOITATION vs. EXPLORATION
Learn to maximize future (discounted) reward:

\[
Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left( R_{t+1} + \gamma \max_{a} Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right)
\]

Source: [1]
The quest for real intelligence

- supervised learning: reasoning by memorization
- pure reinforcement learning: reasoning by trial-and-error
- can we get less brute-force here?

“Reinforcement learning + deep learning = AI” (David Silver, Google Deep Mind)
Deep Q-Learning

Approximate the Q function by a (deep) network

\[ Q(s, a; \theta) \approx Q^*(s, a) \]

Source: [1]
Deep Q-Learning @AlphaGo

Source: [29]
Deep Learning, where to go next?

For structured reading:

- “Awesome deep learning”: https://github.com/ChristosChristofidis/awesome-deep-learning

Just wanna have some cool fun?

- Andrey Karpathy's blog: http://karpathy.github.io/ (especially: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)
- Christopher Olah's blog: http://colah.github.io/
Thanks for your attention!
Sources (1)


[7] Neural Network Learns to Select Potential Anticancer Drugs


[10] Stergiou, C. and Siganos, D. Artificial neurons


[13] Stanford CS231n Convolutional Neural Networks Lecture Notes


[16] Krizhevsky et al. ImageNet Classification with Deep Convolutional Neural Networks
[17] Erhan et al. Scalable Object Detection using Deep Neural Networks
[18] ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)
[19] Long et al. Fully Convolutional Networks for Semantic Segmentation
[22] Chris Olah. Understanding LSTM Networks
[23] Tensorflow seq2seq tutorial


[27] Owens et al. Visually indicated sounds