Familiar With Back-Propagation Step by Step

Dr. Tiansi Dong, Prof. Dr. Christian Bauckhage
dongt@bit.uni-bonn.de, christian.bauckhage@iais.fraunhofer.de

March 2019, IPEC Winter School 2019
B-IT, University of Bonn
The back-propagation algorithm is the fundamental algorithm in Deep Learning.
The back-propagation algorithm is the fundamental algorithm in Deep Learning.

Let us learn this algorithm by coding.
The back-propagation algorithm is the fundamental algorithm in Deep Learning.
Let us learn this algorithm by coding.
Experiment
You are in this maze, equipped with a computer, and given a route instruction.
Now, the instruction reaches the end, you are still in the maze
Now, the instruction reaches the end, you are still in the maze

**What can you do?**
The Problem

- Now, the instruction reaches the end, you are still in the maze
- What can you do?
- Build a neural network to tell you the next action!
Let the given route instruction be $ROUTE = [R_1, R_2, \ldots, R_n]$. 
Let the given route instruction be $ROUTE = [R_1, R_2, \ldots, R_n]$. As there are only four unit instructions, we encode $R_i$ by a two-element vector as follows:
Let the given route instruction be $ROUTE = [R_1, R_2, \ldots, R_n]$. As there are only four unit instructions, we encode $R_i$ by a two-element vector as follows:

- $R_1 =$ turn-left: $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $R_2 =$ turn-right: $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$, go-ahead: $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, turn-around: $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$.
Neural Network

(a)

(b)

(c)
The inputs of the hidden layer are computed by the following matrix

\[
\begin{bmatrix}
h_{1in} \\
h_{2in}
\end{bmatrix} = 
\begin{bmatrix}
w_{11} & w_{12} & w_{13} \\
w_{21} & w_{22} & w_{23}
\end{bmatrix} \cdot 
\begin{bmatrix}
i_1 \\
i_2 \\
b_1
\end{bmatrix}
\]  

(1)
Hidden Layer

- Logistic Function

\[
    h_{1\text{out}} = \frac{1}{1 + e^{-h_{1\text{in}}}} \quad (2)
\]

\[
    h_{2\text{out}} = \frac{1}{1 + e^{-h_{2\text{in}}}} \quad (3)
\]
Hidden Layer

- Logistic Function

\[
h_{1\text{out}} = \frac{1}{1 + e^{-h_{1\text{in}}}} \tag{2}
\]

\[
h_{2\text{out}} = \frac{1}{1 + e^{-h_{2\text{in}}}} \tag{3}
\]

- The inputs to the output layer are computed by the following matrix

\[
\begin{bmatrix}
  o_{1\text{in}} \\
  o_{2\text{in}}
\end{bmatrix} =
\begin{bmatrix}
  V_{11} & V_{12} & V_{13} \\
  V_{21} & V_{22} & V_{23}
\end{bmatrix}
\cdot
\begin{bmatrix}
  h_{1\text{out}} \\
  h_{2\text{out}} \\
  b_2
\end{bmatrix} \tag{4}
\]
Output Layer

The logistic function to use in the output layer.

\[ o_{1\text{out}} = \frac{2}{1 + e^{-o_{1\text{in}}}} - 1 \]  
\[ o_{2\text{out}} = \frac{2}{1 + e^{-o_{2\text{in}}}} - 1 \]
Initial Values

We initialise values of \( w_{ij} = v_{ij} = 0.5 \), and \( b_1 = b_2 = 1 \).
The first instruction in the route is $R_1$.

$$\begin{bmatrix} h_{1\text{in}} \\ h_{2\text{in}} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$h_{1\text{out}} = \frac{1}{1 + e^{-h_{1\text{in}}}} = \frac{1}{1 + e^{-1}} = 0.731$$  \hspace{1cm} (7)

$$h_{2\text{out}} = \frac{1}{1 + e^{-h_{2\text{in}}}} = \frac{1}{1 + e^{-1}} = 0.731$$  \hspace{1cm} (8)
\[
\begin{bmatrix}
o_{1in} \\
o_{2in}
\end{bmatrix} =
\begin{bmatrix}
v_{11} & v_{12} & v_{13} \\
v_{21} & v_{22} & v_{23}
\end{bmatrix} \cdot
\begin{bmatrix}
h_{1out} \\
h_{2out} \\
b_2
\end{bmatrix}
= \begin{bmatrix}
0.5 & 0.5 & 0.5 \\
0.5 & 0.5 & 0.5
\end{bmatrix} \cdot
\begin{bmatrix}
0.731 \\
0.731 \\
1
\end{bmatrix}
= \begin{bmatrix}
1.231 \\
1.231
\end{bmatrix}
\]

\[
o_{1out} = \frac{2}{1 + e^{-o_{1in}}} - 1 = \frac{2}{1 + e^{-1.231}} - 1 = 0.547
\]

\[
o_{2out} = \frac{2}{1 + e^{-o_{2in}}} - 1 = \frac{2}{1 + e^{-1.231}} - 1 = 0.547
\]
We would interpret the result as a route instruction, which is an orientation information.
Energy Cost

- We would interpret the result as a route instruction, which is an orientation information.
- The second instruction should be $R_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, which is quite different to the computed value.
Energy Cost

- We would interpret the result as a route instruction, which is an orientation information.

- The second instruction should be $R_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, which is quite different to the computed value.

- We use $\cos(R_{true}, R_{out})$ to measure the quality of the computed orientation: the best case would be 1, the worse case would be -1.
Energy Cost

- We would interpret the result as a route instruction, which is an orientation information.
- The second instruction should be $R_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, which is quite different to the computed value.
- We use $\cos(R_{true}, R_{out})$ to measure the quality of the computed orientation: the best case would be 1, the worse case would be -1.
- We compute the error $E = 1 - \cos(R_{true}, R_{out})$, so that in the best case $E = 0$, and in the worse case $E = 2$. 
Energy Cost

- We would interpret the result as a route instruction, which is an orientation information.
- The second instruction should be \( R_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \), which is quite different to the computed value.
- We use \( \cos(R_{true}, R_{out}) \) to measure the quality of the computed orientation: the best case would be 1, the worse case would be -1.
- We compute the error \( E = 1 - \cos(R_{true}, R_{out}) \), so that in the best case \( E = 0 \), and in the worse case \( E = 2 \).
- Our first error value \( E = 1 - \cos(\begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.547 \\ 0.547 \end{bmatrix}) = 1.7071 \).
Backward Updating 1

- We need to update parameters to reduce errors.
Backward Updating 1

- We need to update parameters to reduce errors.
- The question is simple: Shall we increase or decrease a parameter?
We need to update parameters to reduce errors.
The question is simple: Shall we increase or decrease a parameter?
We only need to compute the partial derivative of this parameter to the total error $E$.

$$\frac{\partial E}{\partial v_{11}} = \frac{\partial E}{\partial o_{1out}} \times \frac{\partial o_{1out}}{\partial o_{1in}} \times \frac{\partial o_{1in}}{\partial v_{11}}$$ \hspace{1cm} (11)$$

$$E = 1 - \cos(\begin{bmatrix} o_{1true} \\ o_{2true} \end{bmatrix}, \begin{bmatrix} o_{1out} \\ o_{2out} \end{bmatrix})$$ \hspace{1cm} (12)$$

$$E = 1 - \frac{o_{1true} o_{1out} + o_{2true} o_{2out}}{\sqrt{o_{1true}^2 + o_{2true}^2} \sqrt{o_{1out}^2 + o_{2out}^2}}$$ \hspace{1cm} (13)$$
Backward Updating 2

\[
\frac{\partial E}{\partial o_{1\text{out}}} = -\frac{\partial}{\partial o_{1\text{true}} o_{1\text{out}} + o_{2\text{true}} o_{2\text{out}}} \frac{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}
\]

\[
= -\frac{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}} \frac{\partial o_{1\text{true}} o_{1\text{out}} + o_{2\text{true}} o_{2\text{out}}}{\partial o_{1\text{true}}}
\]

\[
= -\frac{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}} \frac{(o_{1\text{true}} o_{1\text{out}} + o_{2\text{true}} o_{2\text{out}})}{(o_{1\text{true}}^2 + o_{2\text{true}}^2)(o_{1\text{out}}^2 + o_{2\text{out}}^2)}
\]

\[
= -\frac{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}} \frac{\partial o_{1\text{true}} o_{1\text{out}} + o_{2\text{true}} o_{2\text{out}}}{\partial o_{1\text{true}}}
\]

\[
= -\frac{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}}{\sqrt{o_{1\text{true}}^2 + o_{2\text{true}}^2} \sqrt{o_{1\text{out}}^2 + o_{2\text{out}}^2}} \frac{\partial o_{1\text{true}} o_{1\text{out}} + o_{2\text{true}} o_{2\text{out}}}{\partial o_{1\text{true}}}
\]

\[
= 0.6463
\]
Backward Updating 3

\[
\frac{\partial o_{1\text{out}}}{\partial o_{1\text{in}}} = o_{1\text{out}}(1 - o_{1\text{out}}) = 0.547 \times (1 - 0.547) = 0.2478
\]

\[
\frac{\partial o_{1\text{in}}}{\partial v_{11}} = \frac{\partial(v_{11} h_{1\text{out}} + v_{12} h_{2\text{out}} + v_{13} b_2)}{\partial v_{11}} = h_{1\text{out}} = 0.731
\]

Therefore,

\[
\frac{\partial E}{\partial v_{11}} = \frac{\partial E}{\partial o_{1\text{out}}} \times \frac{\partial o_{1\text{out}}}{\partial o_{1\text{in}}} \times \frac{\partial o_{1\text{in}}}{\partial v_{11}} \quad (14)
\]

\[
= 0.6463 \times 0.2478 \times 0.731 
\]

\[
= 0.1171 \quad (16)
\]
To decrease the value of $E$, we need to decrease $v_{11}$ as follows.

\[
v^{(next)}_{11} = v_{11} - \eta \frac{\partial E}{\partial v_{11}} = 0.5 - 10 \times 0.1171 = -0.671
\]

\[
v^{(next)}_{12} = v_{12} - \eta \frac{\partial E}{\partial v_{12}} = 0.5 - 10 \times 0.1171 = -0.671
\]

\[
v^{(next)}_{13} = v_{13} - \eta \frac{\partial E}{\partial v_{13}} = 0.5 - 10 \times 0.160 = -1.1
\]

\[
v^{(next)}_{21} = v_{21} - \eta \frac{\partial E}{\partial v_{21}} = 0.5 - 10 \times (-0.1171) = 1.6171
\]

\[
v^{(next)}_{22} = v_{22} - \eta \frac{\partial E}{\partial v_{22}} = 0.5 - 10 \times (-0.1171) = 1.6171
\]

\[
v^{(next)}_{23} = v_{23} - \eta \frac{\partial E}{\partial v_{23}} = 0.5 - 10 \times (-0.160) = 2.1
\]
Next, we need to update $w_{ij}$

\[
\frac{\partial E}{\partial w_{11}} = \frac{\partial E}{\partial h_{1\text{out}}} \times \frac{\partial h_{1\text{out}}}{\partial h_{1\text{in}}} \times \frac{\partial h_{1\text{in}}}{\partial w_{11}} \\
= \frac{\partial E}{\partial o_{1\text{out}}} \times \frac{\partial o_{1\text{out}}}{\partial o_{1\text{in}}} \times \frac{\partial o_{1\text{in}}}{\partial h_{1\text{out}}} \times \frac{\partial h_{1\text{out}}}{\partial h_{1\text{in}}} \times \frac{\partial h_{1\text{in}}}{\partial w_{11}} \\
+ \frac{\partial E}{\partial o_{2\text{out}}} \times \frac{\partial o_{2\text{out}}}{\partial o_{2\text{in}}} \times \frac{\partial o_{2\text{in}}}{\partial h_{1\text{out}}} \times \frac{\partial h_{1\text{out}}}{\partial h_{1\text{in}}} \times \frac{\partial h_{1\text{in}}}{\partial w_{11}} \\
= \frac{\partial E}{\partial o_{1\text{out}}} \times \frac{\partial o_{1\text{out}}}{\partial o_{1\text{in}}} v_{11}^{(\text{next})} \times \frac{\partial h_{1\text{out}}}{\partial h_{1\text{in}}} \times \frac{\partial h_{1\text{in}}}{\partial w_{11}} \\
+ \frac{\partial E}{\partial o_{2\text{out}}} \times \frac{\partial o_{2\text{out}}}{\partial o_{2\text{in}}} v_{21}^{(\text{next})} \times \frac{\partial h_{1\text{out}}}{\partial h_{1\text{in}}} \times \frac{\partial h_{1\text{in}}}{\partial w_{11}} \\
= 0.6463 \times 0.2478 \times (−0.671) \times 0.731 \times 0.269 \times 1 \\
+ (−0.6463) \times 0.2478 \times 1.6171 \times 0.731 \times 0.269 \times 1 \\
= −0.0721
\]
Backward Updating 6

\[
\begin{align*}
\hat{w}_{11} & = w_{11} - \eta \frac{\partial E}{\partial w_{11}} = 0.5 - 10 \times (-0.0721) = 1.221 \\
\hat{w}_{12} & = w_{12} - \eta \frac{\partial E}{\partial w_{12}} = 0.5 - 10 \times 0 = 0 \\
\hat{w}_{13} & = w_{13} - \eta \frac{\partial E}{\partial w_{13}} = 0.5 - 10 \times (-0.0721) = 1.221 \\
\hat{w}_{21} & = w_{21} - \eta \frac{\partial E}{\partial w_{21}} = 0.5 - 10 \times (-0.0721) = 1.221 \\
\hat{w}_{22} & = w_{22} - \eta \frac{\partial E}{\partial w_{22}} = 0.5 - 10 \times 0 = 0 \\
\hat{w}_{23} & = w_{23} - \eta \frac{\partial E}{\partial w_{23}} = 0.5 - 10 \times (-0.0721) = 1.221
\end{align*}
\]
After this parameter update, we have the

\[
\begin{bmatrix}
  h_{1in} \\
  h_{2in}
\end{bmatrix} = \begin{bmatrix}
  w_{11} & w_{12} & w_{13} \\
  w_{21} & w_{22} & w_{23}
\end{bmatrix} \cdot \begin{bmatrix}
  i_1 \\
  i_2 \\
  b_1
\end{bmatrix}
\]

(17)

\[
= \begin{bmatrix}
  1.221 & 0 & 1.221 \\
  1.221 & 0 & 1.221
\end{bmatrix} \cdot \begin{bmatrix}
  1 \\
  0 \\
  1
\end{bmatrix}
\]

(18)

\[
= \begin{bmatrix}
  2.442 \\
  2.442
\end{bmatrix}
\]

(19)

\[
h_{1out} = \frac{1}{1 + e^{-h_{1in}}} = \frac{1}{1 + e^{-2.442}} = 0.92
\]

(20)

\[
h_{2out} = \frac{1}{1 + e^{-h_{2in}}} = \frac{1}{1 + e^{-2.442}} = 0.92
\]

(21)
\[
\begin{bmatrix}
\sigma_{1in} \\
\sigma_{2in}
\end{bmatrix} =
\begin{bmatrix}
v_{11} & v_{12} & v_{13} \\
v_{21} & v_{22} & v_{23}
\end{bmatrix} \cdot \begin{bmatrix}
h_{1out} \\
h_{2out} \\
b_2
\end{bmatrix}
\]
\[
= \begin{bmatrix}
-0.671 & -0.671 & -1.1 \\
1.6171 & 1.6171 & 2.1
\end{bmatrix} \cdot \begin{bmatrix}
0.92 \\
0.92 \\
1
\end{bmatrix}
\]
\[
= \begin{bmatrix}
-2.335 \\
5.075
\end{bmatrix}
\]

\[
\sigma_{1out} = \frac{2}{1 + e^{-\sigma_{1in}}} - 1 = \frac{2}{1 + e^{2.335}} - 1 = -0.823
\]

\[
\sigma_{2out} = \frac{2}{1 + e^{-\sigma_{2in}}} - 1 = \frac{2}{1 + e^{-5.075}} - 1 = 0.988
\]
With these update parameters, the error $E$ reduces to

$$1 - \cos\left(\begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -0.823 \\ 0.988 \end{bmatrix}\right) = 0.35$$
Implement above way-finding tool in Python, and push the code into your github.
Implement above way-finding tool in Python, and push the code into your github.

reference https://julien.danjou.info/starting-your-first-python-project/
Coding

- Implement above way-finding tool in Python, and push the code into your github.
- reference https://julien.danjou.info/starting-your-first-python-project/
- Have fun!