Neural Networks

Lecture 5: Back-propagation
<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Probability and Information Theory</td>
</tr>
<tr>
<td>3</td>
<td>Basics of Machine Learning</td>
</tr>
<tr>
<td>4</td>
<td>Deep Feed-forward networks</td>
</tr>
<tr>
<td>5</td>
<td>Back-propagation</td>
</tr>
<tr>
<td>6</td>
<td>Optimization and regularization</td>
</tr>
<tr>
<td>7</td>
<td>Convolutional Networks</td>
</tr>
<tr>
<td>8</td>
<td>Sequence Modeling: recurrent networks</td>
</tr>
<tr>
<td>9</td>
<td>Applications</td>
</tr>
<tr>
<td>10</td>
<td>Software and Practical methods</td>
</tr>
<tr>
<td>11</td>
<td>Representational Learning</td>
</tr>
<tr>
<td>12</td>
<td>Deep Generative Models</td>
</tr>
<tr>
<td>13</td>
<td>Deep Reinforcement Learning</td>
</tr>
<tr>
<td>14</td>
<td>Deep Learning and the Brain</td>
</tr>
<tr>
<td>15</td>
<td>Conclusions and Future Perspective</td>
</tr>
<tr>
<td>16</td>
<td>Project presentations</td>
</tr>
</tbody>
</table>

- **Foundations**
- **Basics**
- **Advanced**
- **Selected topics**
inputs weights

\[ \sum \]

weighted sum

step function
**Network diagrams**

\[
\begin{align*}
\text{Output} & \quad y = f^{(2)}(h; w, b) \\
\text{Hidden} & \quad h = f^{(1)}(x, W, c) \\
\text{Input} & \quad f(x; W, c, w, b) = f^{(2)}(f^{(1)}(x))
\end{align*}
\]
Universal approximation

A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions in compact subsets of $\mathbb{R}^n$, under mild assumptions of the activation functions.

Continuous functions \[ y = f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x))) \]

Approx to any desired degree # hidden units
Universal approximation
Universal approximation
Universal approximation

\[ f(x) \]

\[ s_1 = 0.40 \quad w_1 = 0.6 \]

\[ s_2 = 0.60 \quad w_2 = 1.2 \]

Weighted output from hidden layer
Universal approximation
Universal approximation

\[ f(x) \]

\[ h = 0.6 \]

Weighted output from hidden layer
Universal approximation
Universal approximation
Better Generalization with Greater Depth

Figure 6.6: Empirical results showing that deeper networks generalize better when used to transcribe multi-digit numbers from photographs of addresses. Data from Goodfellow et al. (2014d). The test set accuracy consistently increases with increasing depth. See figure 6.7 for a control experiment demonstrating that other increases to the model size do not yield the same effect.

Another key consideration of architecture design is exactly how to connect a pair of layers to each other. In the default neural network layer described by a linear transformation via a matrix $W$, every input unit is connected to every output unit. Many specialized networks in the chapters ahead have fewer connections, so that each unit in the input layer is connected to only a small subset of units in the output layer. These strategies for reducing the number of connections reduce the number of parameters and the amount of computation required to evaluate the network, but are often highly problem-dependent. For example, convolutional networks, described in chapter 9, use specialized patterns of sparse connections that are very effective for computer vision problems. In this chapter, it is difficult to give much more specific advice concerning the architecture of a generic neural network. Subsequent chapters develop the particular architectural strategies that have been found to work well for different application domains.
Learning objectives

• Understand the basics of gradient descent

• Apply back-propagation to estimate the gradients with respect to weights
Data    Model    Cost function    Optimization
\[ p_{\text{model}}(y \mid x) = \mathcal{N}(y; f(x; \theta), I) \]

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]

\[ f(x; W, c, w, b) = f^{(2)}(f^{(1)}(x)) \]
Data  Model  Cost function  Optimization
Neural networks

Input:

Output:
Neural networks
Neural networks

Output:

Input:
Neural networks

Input:

Output:
Neural networks
Learning algorithm

• while not done
  • pick a random training case \((x, y)\)
  • run neuronal network on input \(x\)
  • modify connection weights to make prediction closer to \(y\)
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]

Analytical
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]

Analytical

Exhaustive search
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]

Analytical

Exhaustive search

Random search
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y \mid x) \]

Analytical
Exhaustive search
Random search
Random local search
Minimize

\[ J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(y | x) \]

Analytical
Exhaustive search
Random search
Random local search
Gradient descent
Cost function \[ J(\theta) \]

Gradient
\[ g = \nabla_{\theta} J(\theta) \]
\[ g_i = \frac{\partial}{\partial \theta_i} J(\theta) \]

\[ \theta \leftarrow \theta - \epsilon g \]
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]

\[ \frac{dz}{dx} \]
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]
\[ \frac{dz}{dx} ? \]
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]

Find \( \frac{dz}{dx} \)?
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]
\[ \frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \]

\( y \in \mathbb{R}^n \quad z = f(y) \)

\( x \in \mathbb{R}^m \quad y = g(x) \)
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]

\[ \frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \]

\[ y \in \mathbb{R}^n \quad z = f(y) \]
\[ x \in \mathbb{R}^m \quad y = g(x) \]
Chain rule (derivatives)

\[ z = f(y) \]
\[ y = g(x) \]

\[ \frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx} \]

\[ y \in \mathbb{R}^n \quad z = f(y) \]
\[ x \in \mathbb{R}^m \quad y = g(x) \]

\[ \frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i} \]
Example 1

Input Layer (first)

\( x_1 \)

Pre-sigmoidal output layer (second)

\( W_{11} \)

\( W_{12} \)

Pre sigmoidal outputs

\( y_1 \)

Sigmoidal layer

sigmoid

\( y_{o1} \)

\( y_2 \)

Post sigmoidal outputs

sigmoid

\( y_{o2} \)

Cost Function
\[ W_{ij} = W_{ij} - h \cdot \frac{\partial C}{\partial W_{ij}} \]

Input Layer (first)

\( x_1 \)

\( x_2 \)

\( x_3 \)

Pre-sigmoidal output layer (second)

\( W_{11} \)

\( W_{12} \)

\( W_{21} \)

\( W_{22} \)

\( W_{31} \)

\( W_{32} \)

Sigmoidal layer

\( y_1 \)

\( y_2 \)

Pre sigmoidal outputs

Post sigmoidal outputs

Cost Function

\( y_{o1} \)

\( y_{o2} \)
\[ W_{ij} = W_{ij} - h \cdot \frac{\partial C}{\partial W_{ij}} \]
Feed-forward pass
Feed-forward pass

\[ \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \]
Feed-forward pass

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3
\end{bmatrix} \begin{bmatrix}
  W_{11} & W_{12} \\
  W_{21} & W_{22} \\
  W_{31} & W_{32}
\end{bmatrix} =
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix}
\]
Feed-forward pass

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
\end{bmatrix}
\begin{bmatrix}
  W_{11} & W_{12} \\
  W_{21} & W_{22} \\
  W_{31} & W_{32} \\
\end{bmatrix}
\]

\[
y_1 = x_1 W_{11} + x_2 W_{21} + x_3 W_{31}
\]

\[
y_2 = x_1 W_{12} + x_2 W_{22} + x_3 W_{32}
\]
Feed-forward pass

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 
\end{bmatrix}
\begin{bmatrix}
  W_{11} & W_{12} \\
  W_{21} & W_{22} \\
  W_{31} & W_{32} 
\end{bmatrix}
\begin{align*}
y_1 &= x_1 W_{11} + x_2 W_{21} + x_3 W_{31} \\
y_2 &= x_1 W_{12} + x_2 W_{22} + x_3 W_{32}
\end{align*}
\]

\[
y_{o1} = \\
y_{o2} = 
\]
Feed-forward pass

$$[x_1 \ x_2 \ x_3] \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{bmatrix} y_1 = x_1 W_{11} + x_2 W_{21} + x_3 W_{31}$$

$$y_2 = x_1 W_{12} + x_2 W_{22} + x_3 W_{32}$$

$$y_{o1} = \text{sigmoid}(y_1)$$

$$y_{o2} = \text{sigmoid}(y_2)$$
Cost function

\[ C = -\frac{1}{N} \sum_{i} p_i \times \log(q_i) \]
Cost function

\[ C = -\frac{1}{N} \sum_{i} p_{i} \cdot \log(q_{i}) \]
Cost function

\[ C = -\frac{1}{N} \sum_i p_i \cdot \log(q_i) \]
Cost function

\[ C = - \frac{1}{N} \sum_{i} p_i \cdot \log(q_i) \]

\[ C = - \frac{1}{N} \cdot (p_1 \log(q_1) + p_2 \log(q_2)) \]
Cost function

\[ C = -\frac{1}{N} \sum_{i} p_i \log(q_i) \]

\[ C = -\frac{1}{N} \cdot (p_1 \log(q_1) + p_2 \log(q_2)) \]

\[ C = -(p_1 \log(y_{o1}) + p_2 \log(y_{o2})) \]
Back-propagation

\[ \frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \times \frac{\partial y_{o1}}{\partial y_1} \times \frac{\partial y_1}{\partial W_{11}} \]
Back-propagation

\[
\frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \times \frac{\partial y_{o1}}{\partial y_{1}} \times \frac{\partial y_{1}}{\partial W_{11}}
\]

\[
\frac{\partial C}{\partial y_{o1}} = -\left( \frac{p_{1}}{y_{o1}} \right)
\]

\[
\frac{\partial C}{\partial y_{o2}} = -\left( \frac{p_{2}}{y_{o2}} \right)
\]
Back-propagation

\[
\frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \times \frac{\partial y_{o1}}{\partial y_1} \times \frac{\partial y_1}{\partial W_{11}}
\]

\[
\frac{\partial C}{\partial y_{o1}} = -\left(\frac{p_1}{y_{o1}}\right)
\]

\[
\frac{\partial C}{\partial y_{o2}} = -\left(\frac{p_2}{y_{o2}}\right)
\]
Back-propagation

\[ \frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{01}} \cdot \frac{\partial y_{01}}{\partial y_1} \cdot \frac{\partial y_1}{\partial W_{11}} \]

\[ \frac{\partial C}{\partial y_{01}} = -\left( \frac{p_1}{y_{01}} \right) \]

\[ \frac{\partial y_{01}}{\partial y_1} = \sigma(y_1) \cdot (1 - \sigma(y_1)) \]

\[ \frac{\partial C}{\partial y_{02}} = -\left( \frac{p_2}{y_{02}} \right) \]

\[ \frac{\partial y_{02}}{\partial y_2} = \sigma(y_2) \cdot (1 - \sigma(y_2)) \]
Back-propagation

\[
\frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \cdot \frac{\partial y_{o1}}{\partial y_1} \cdot \frac{\partial y_1}{\partial W_{11}}
\]

\[
\frac{\partial C}{\partial y_{o1}} = - \left( \frac{p_1}{y_{o1}} \right)
\]

\[
\frac{\partial y_{o1}}{\partial y_1} = \sigma(y_1) \cdot (1 - \sigma(y_1))
\]

\[
\frac{\partial C}{\partial y_{o2}} = - \left( \frac{p_2}{y_{o2}} \right)
\]

\[
\frac{\partial y_{o2}}{\partial y_2} = \sigma(y_2) \cdot (1 - \sigma(y_2))
\]
Back-propagation

\[
\frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \cdot \frac{\partial y_{o1}}{\partial y_1} \cdot \frac{\partial y_1}{\partial W_{11}}
\]

\[
\frac{\partial C}{\partial y_{o1}} = -\left( \frac{p_1}{y_{o1}} \right)
\]

\[
\frac{\partial C}{\partial y_{o2}} = -\left( \frac{p_2}{y_{o2}} \right)
\]

\[
\frac{\partial y_{o1}}{\partial y_1} = \sigma(y_1) \cdot (1 - \sigma(y_1))
\]

\[
\frac{\partial y_{o2}}{\partial y_2} = \sigma(y_2) \cdot (1 - \sigma(y_2))
\]

\[
y_1 = x_1 W_{11} + x_2 W_{21} + x_3 W_{31}
\]

\[
y_2 = x_1 W_{12} + x_2 W_{22} + x_3 W_{32}
\]
Back-propagation

\[ \frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \cdot \frac{\partial y_{o1}}{\partial y_1} \cdot \frac{\partial y_1}{\partial W_{11}} \]

\[ \frac{\partial C}{\partial y_{o1}} = -\left(\frac{p_1}{y_{o1}}\right) \]

\[ \frac{\partial y_{o1}}{\partial y_1} = \sigma(y_1) \cdot (1 - \sigma(y_1)) \]

\[ \frac{\partial C}{\partial y_{o2}} = -\left(\frac{p_2}{y_{o2}}\right) \]

\[ \frac{\partial y_{o2}}{\partial y_2} = \sigma(y_2) \cdot (1 - \sigma(y_2)) \]

\[ y_1 = x_1W_{11} + x_2W_{21} + x_3W_{31} \]

\[ y_2 = x_1W_{12} + x_2W_{22} + x_3W_{32} \]

\[ \frac{\partial y_1}{\partial W_{11}} = x_1 \]

\[ \frac{\partial y_1}{\partial W_{21}} = x_2 \]

\[ \frac{\partial y_1}{\partial W_{31}} = x_3 \]

\[ \frac{\partial y_2}{\partial W_{12}} = x_1 \]

\[ \frac{\partial y_2}{\partial W_{22}} = x_2 \]

\[ \frac{\partial y_2}{\partial W_{32}} = x_3 \]
Back-propagation

\[
\frac{\partial C}{\partial W_{11}} = \frac{\partial C}{\partial y_{o1}} \cdot \frac{\partial y_{o1}}{\partial y_1} \cdot \frac{\partial y_1}{\partial W_{11}}
\]

\[
\frac{\partial C}{\partial y_{o1}} = -(\frac{p_1}{y_{o1}})
\]

\[
\frac{\partial y_{o1}}{\partial y_1} = \sigma(y_1) \cdot (1 - \sigma(y_1))
\]

\[
\frac{\partial y_{o2}}{\partial y_2} = \sigma(y_2) \cdot (1 - \sigma(y_2))
\]

\[
y_1 = x_1 W_{11} + x_2 W_{21} + x_3 W_{31}
\]

\[
y_2 = x_1 W_{12} + x_2 W_{22} + x_3 W_{32}
\]

\[
\frac{\partial y_1}{\partial W_{11}} = x_1
\]

\[
\frac{\partial y_1}{\partial W_{21}} = x_2
\]

\[
\frac{\partial y_1}{\partial W_{31}} = x_3
\]

\[
\frac{\partial y_2}{\partial W_{12}} = x_1
\]

\[
\frac{\partial y_2}{\partial W_{22}} = x_2
\]

\[
\frac{\partial y_2}{\partial W_{32}} = x_3
\]

**Diagram:**

- Input Layer (first)
- Pre-sigmoidal output layer (second)
- Sigmoidal layer
- Post sigmoidal outputs
- Cost Function
\[ W_{ij} = W_{ij} - h \cdot \frac{\partial C}{\partial W_{ij}} \]
\[ W_{ij} = W_{ij} - h \times \frac{\partial C}{\partial W_{ij}} \]

\[ \frac{\partial C}{\partial W_{11}} = \frac{-p_1}{y_{o1}} \times \sigma(y_1) \times (1 - \sigma(y_1)) \times x_1 \]
Example 2

\[ \frac{\partial E_{total}}{\partial w_5} = \frac{\partial \text{net}_{o1}}{\partial \text{out}_{o1}} \times \frac{\partial \text{out}_{o1}}{\partial \text{net}_{o1}} = \frac{\partial E_{total}}{\partial \text{net}_{o1}} \]

\[ E_{o1} = \frac{1}{2} (\text{target}_{o1} - \text{out}_{o1})^2 \]

\[ E_{total} = E_{o1} + E_{o2} \]
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1 = 0.05</td>
<td>o1 = 0.01</td>
</tr>
<tr>
<td>i2 = 0.10</td>
<td>o2 = 0.99</td>
</tr>
</tbody>
</table>

I. The Forward pass - Compute total error

\[ net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1 \]
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i_1 = 0.05)</td>
<td>(o_1 = 0.01)</td>
</tr>
<tr>
<td>(i_2 = 0.10)</td>
<td>(o_2 = 0.99)</td>
</tr>
</tbody>
</table>

1. The Forward pass - Compute total error

\[
net_{h1} = w_1 \times i_1 + w_2 \times i_2 + b_1 \times 1 \\
net_{h1} = 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35 \times 1 = 0.3775
\]
Let's backpropagate

\[ \text{Input} \quad \text{Target} \]

\[ i_1 = 0.05 \quad o_1 = 0.01 \]

\[ i_2 = 0.10 \quad o_2 = 0.99 \]

1. The Forward pass - Compute total error

\[ \text{net}_{h_1} = w_1 \cdot i_1 + w_2 \cdot i_2 + b_1 \cdot 1 \]

\[ \text{net}_{h_1} = 0.15 \cdot 0.05 + 0.2 \cdot 0.1 + 0.35 \cdot 1 = 0.3775 \]

\[ \text{out}_{h_1} = \frac{1}{1 + e^{-\text{net}_{h_1}}} = \frac{1}{1 + e^{-0.3775}} = 0.5933 \]

\[ f(x) = \frac{1}{1 + e^{-x}} \]
Let's backpropagate

**INPUT**  |  **TARGET**
---|---
i1 = 0.05  |  o1 = 0.01
i2 = 0.10  |  o2 = 0.99

1. The Forward pass - Compute total error

\[ net_{h1} = w_1 \times i_1 + w_2 \times i_2 + b_1 \times 1 \]
\[ net_{h1} = 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35 \times 1 = 0.3775 \]

\[ out_{h1} = \frac{1}{1 + e^{-net_{h1}}} = \frac{1}{1 + e^{-0.3775}} = 0.5933 \]

Repeat for h2 = 0.596; o1 = 0.751; o2 = 0.773
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i_1 = 0.05)</td>
<td>(o_1 = 0.01)</td>
</tr>
<tr>
<td>(i_2 = 0.10)</td>
<td>(o_2 = 0.99)</td>
</tr>
</tbody>
</table>

1. The Forward pass - Compute total error

We have \(o_1, o_2\)
Let's backpropagate

INPUT            TARGET

\( i_1 = 0.05 \)  \( o_1 = 0.01 \)
\( i_2 = 0.10 \)  \( o_2 = 0.99 \)

1. The Forward pass - Compute total error

We have \( o_1, o_2 \)

\[ E_{total} = \sum \frac{1}{2} \left( \text{target} - \text{output} \right)^2 \]
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1 = 0.05$</td>
<td>$o_1 = 0.01$</td>
</tr>
<tr>
<td>$i_2 = 0.10$</td>
<td>$o_2 = 0.99$</td>
</tr>
</tbody>
</table>

1. The Forward pass - Compute total error

We have $o_1, o_2$

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

$$E_{o1} = \frac{1}{2} (target_{o1} - out_{o1})^2 = \frac{1}{2} (0.01 - 0.7514)^2 = 0.2748$$
Let's backpropagate

INPUT | TARGET
--- | ---
i1 = 0.05 | o1 = 0.01
i2 = 0.10 | o2 = 0.99

1. The Forward pass - Compute total error

We have o1, o2

\[ E_{total} = \sum \frac{1}{2} (target - output)^2 \]

\[ E_{o1} = \frac{1}{2} (target_{o1} - output_{o1})^2 = \frac{1}{2} (0.01 - 0.7514)^2 = 0.2748 \]

\[ E_{o2} = 0.02356 \]
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1 = 0.05$</td>
<td>$o_1 = 0.01$</td>
</tr>
<tr>
<td>$i_2 = 0.10$</td>
<td>$o_2 = 0.99$</td>
</tr>
</tbody>
</table>

1. The Forward pass - Compute total error

We have $o_1, o_2$

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

$$E_{o1} = \frac{1}{2} (target_{o1} - out_{o1})^2 = \frac{1}{2} (0.01 - 0.7514)^2 = 0.2748$$

$$E_{o2} = 0.02356$$

$$E_{total} = E_{o1} + E_{o2} = 0.2748 + 0.02356 = 0.29836$$
Let's backpropagate

INPUT | TARGET
-----|-----
i1 = 0.05 | o1 = 0.01
i2 = 0.10 | o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error
Let's backpropagate

**INPUT**

\[ i_1 = 0.05 \]
\[ i_2 = 0.10 \]

**TARGET**

\[ o_1 = 0.01 \]
\[ o_2 = 0.99 \]

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}
\]

\[ E_{o1} = \frac{1}{2}(\text{target}_{o1} - \text{out}_{o1})^2 \]
\[ E_{total} = E_{o1} + E_{o2} \]
Let's backpropagate

INPUT  TARGET

i1 = 0.05  o1 = 0.01
i2 = 0.10  o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{total}}{\partial w_5} = \left( \frac{\partial E_{total}}{\partial out_{o1}} \right) \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}
\]
Let's backpropagate

**INPUT**

\[ i_1 = 0.05 \]

\[ i_2 = 0.10 \]

**TARGET**

\[ o_1 = 0.01 \]

\[ o_2 = 0.99 \]

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{total}}{\partial w_5} = \left( \frac{\partial E_{total}}{\partial out_{o1}} \right) \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}
\]

\[
E_{total} = \sum \frac{1}{2} (target - output)^2
\]
Let's backpropagate

INPUT | TARGET
--- | ---
i1 = 0.05 | o1 = 0.01
i2 = 0.10 | o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \left( \frac{\partial E_{\text{total}}}{\partial \text{out}_{o1}} \right) \cdot \frac{\partial \text{out}_{o1}}{\partial \text{net}_{o1}} \cdot \frac{\partial \text{net}_{o1}}{\partial w_5}
\]

\[
E_{\text{total}} = \sum \frac{1}{2} (\text{target} - \text{output})^2
\]

\[
\frac{\partial E_{\text{total}}}{\partial \text{out}_{o1}} = 2 \cdot \frac{1}{2} (\text{target}_{o1} - \text{out}_{o1}) \cdot -1 + 0 = -(0.01 - 0.751) = 0.741
\]
2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$
Lets backpropagate

INPUT  TARGET

i1 = 0.05  o1 = 0.01
i2 = 0.10  o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}$$

$$out_{o1} = \frac{1}{1 + e^{-net_{o1}}}$$
Lets backpropagate

INPUT               TARGET

i1 = 0.05          o1 = 0.01
i2 = 0.10          o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}$$

$$out_{o1} = \frac{1}{1 + e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1} (1 - out_{o1}) = 0.1868$$
2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial o_{out1}} \times \frac{\partial o_{out1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$
2. The Backward pass - Updating weights

We want to know how much a change in \(w_5\) affects the total error

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \frac{\partial E_{\text{total}}}{\partial \text{out}_{o1}} \cdot \frac{\partial \text{out}_{o1}}{\partial \text{net}_{o1}} \cdot \frac{\partial \text{net}_{o1}}{\partial w_5}
\]

\(\text{net}_{o1} = w_5 \cdot \text{out}_{h1} + w_6 \cdot \text{out}_{h2} + b_2 \cdot 1\)
Let's backpropagate

**INPUT**

\[ i_1 = 0.05 \]
\[ i_2 = 0.10 \]

**TARGET**

\[ o_1 = 0.01 \]
\[ o_2 = 0.99 \]

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \frac{\partial E_{\text{total}}}{\partial \text{out}_{o_1}} \times \frac{\partial \text{out}_{o_1}}{\partial \text{net}_{o_1}} \times \frac{\partial \text{net}_{o_1}}{\partial w_5}
\]

\[
\text{net}_{o_1} = w_5 \times \text{out}_{h_1} + w_6 \times \text{out}_{h_2} + b_2 \times 1
\]

\[
\frac{\partial \text{net}_{o_1}}{\partial w_5} = \text{out}_{h_1} = 0.5933
\]
Let's backpropagate

**INPUT**

\[ i_1 = 0.05 \]

\[ i_2 = 0.10 \]

**TARGET**

\[ o_1 = 0.01 \]

\[ o_2 = 0.99 \]

2. The Backward pass - Updating weights

We want to know how much a change in \( w_5 \) affects the total error

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \frac{\partial E_{\text{total}}}{\partial \text{out}_{o_1}} \times \frac{\partial \text{out}_{o_1}}{\partial \text{net}_{o_1}} \times \frac{\partial \text{net}_{o_1}}{\partial w_5}
\]
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1 = 0.05</td>
<td>o1 = 0.01</td>
</tr>
<tr>
<td>i2 = 0.10</td>
<td>o2 = 0.99</td>
</tr>
</tbody>
</table>

2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.7414 \times 0.1868 \times 0.5933 = 0.0821$$
2. The Backward pass - Updating weights

We want to know how much a change in $w_5$ affects the total error

\[
\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}
\]

\[
\frac{\partial E_{total}}{\partial w_5} = 0.7414 \times 0.1868 \times 0.5933 = 0.0821
\]

\[
w_5^{new} = w_5^{old} - \eta \times \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 \times 0.0821 = 0.3589
\]
Let's backpropagate

INPUT   TARGET

i₁ = 0.05  o₁ = 0.01
i₂ = 0.10  o₂ = 0.99

• Repeat for w₆, w₇, w₈
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1 = 0.05</td>
<td>o1 = 0.01</td>
</tr>
<tr>
<td>i2 = 0.10</td>
<td>o2 = 0.99</td>
</tr>
</tbody>
</table>

- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4
Lets backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i_1 = 0.05)</td>
<td>(o_1 = 0.01)</td>
</tr>
<tr>
<td>(i_2 = 0.10)</td>
<td>(o_2 = 0.99)</td>
</tr>
</tbody>
</table>

- Repeat for \(w_6, w_7, w_8\)
- In analogous way for \(w_1, w_2, w_3, w_4\)
- Compute the total error before: 0.298371109
Let's backpropagate

<table>
<thead>
<tr>
<th>INPUT</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1 = 0.05</td>
<td>o1 = 0.01</td>
</tr>
<tr>
<td>i2 = 0.10</td>
<td>o2 = 0.99</td>
</tr>
</tbody>
</table>

- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4
- Compute the total error before: 0.298371109
  now: 0.291027924
Let's backpropagate

INPUT          TARGET

\(i_1 = 0.05\)  \(o_1 = 0.01\)

\(i_2 = 0.10\)  \(o_2 = 0.99\)

- Repeat for \(w_6, w_7, w_8\)
- In analogous way for \(w_1, w_2, w_3, w_4\)
- Compute the total error before: \(0.298371109\)
  now: \(0.291027924\)

- Repeat \(x10000\): \(0.000035085\)
Computational graphs
Computational graphs

\[ \hat{y} \]

\[ u^{(1)} \rightarrow u^{(2)} \rightarrow \text{dot} \rightarrow \text{sum} \rightarrow \hat{y} \]

\[ H \rightarrow \text{relu} \rightarrow U^{(1)} \rightarrow U^{(2)} \rightarrow \text{sum} \rightarrow H \]

\[ x \rightarrow w \rightarrow b \]
\[ a \rightarrow c = a + b \rightarrow e = c \times d \rightarrow d = b + 1 \rightarrow b \]
\[e = c \times d\]
\[e = 6\]

\[\frac{\partial e}{\partial c} = 2\]
\[\frac{\partial e}{\partial d} = 3\]

\[c = a + b\]
\[c = 3\]

\[\frac{\partial c}{\partial a} = 1\]
\[\frac{\partial c}{\partial b} = 1\]

\[\frac{\partial d}{\partial b} = 1\]

\[a = 2\]
\[a = 2\]

\[b = 1\]
\[b = 1\]

\[d = b + 1\]
\[d = 2\]
Forward-Mode Differentiation \( \frac{\partial}{\partial X} \)
Reverse-Mode Differentiation ($\frac{\partial Z}{\partial \theta}$)

\[ \frac{\partial Z}{\partial X} = (\alpha + \beta + \gamma)(\delta + \epsilon + \zeta) \]

\[ \frac{\partial Z}{\partial Y} = \delta + \epsilon + \zeta \]

\[ \frac{\partial Z}{\partial Z} = 1 \]
\[ e = c \times d \]
\[ e = 6 \]
\[ \frac{\partial e}{\partial c} = 2 \]
\[ \frac{\partial e}{\partial d} = 3 \]
\[ c = a + b \]
\[ c = 3 \]
\[ \frac{\partial c}{\partial a} = 1 \]
\[ \frac{\partial c}{\partial b} = 1 \]
\[ d = b + 1 \]
\[ d = 2 \]
\[ \frac{\partial d}{\partial b} = 1 \]
\[ a = 2 \]
\[ b = 1 \]
Forward vs Reverse derivation

$10^8$ 1 week $\rightarrow$ 200,000 years
Cost function \[ J(\theta) \]

Gradient
\[ g = \nabla_\theta J(\theta) \]
\[ g_i = \frac{\partial}{\partial \theta_i} J(\theta) \]

\[ \theta \leftarrow \theta - \epsilon g \]
<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Probability and Information Theory</td>
</tr>
<tr>
<td>3</td>
<td>Basics of Machine Learning</td>
</tr>
<tr>
<td>4</td>
<td>Deep Feed-forward networks</td>
</tr>
<tr>
<td>5</td>
<td>Back-propagation</td>
</tr>
<tr>
<td>6</td>
<td>Optimization and regularization</td>
</tr>
<tr>
<td>7</td>
<td>Convolutional Networks</td>
</tr>
<tr>
<td>8</td>
<td>Sequence Modeling: recurrent networks</td>
</tr>
<tr>
<td>9</td>
<td>Applications</td>
</tr>
<tr>
<td>10</td>
<td>Software and Practical methods</td>
</tr>
<tr>
<td>11</td>
<td>Representational Learning</td>
</tr>
<tr>
<td>12</td>
<td>Deep Generative Models</td>
</tr>
<tr>
<td>13</td>
<td>Deep Reinforcement Learning</td>
</tr>
<tr>
<td>14</td>
<td>Deep Learning and the Brain</td>
</tr>
<tr>
<td>15</td>
<td>Conclusions and Future Perspective</td>
</tr>
<tr>
<td>16</td>
<td>Project presentations</td>
</tr>
</tbody>
</table>

- **Foundations**
- **Basics**
- **Advanced**
- **Selected topics**