Back Propagation of Neural Network

Sungjoon Choi
Artificial Intelligence 2017 spring
Seoul National University
Single Layer Perceptron

In TensorFlow

\[ x = \text{tf.placeholder}(\text{tf.float32}, [\text{None}, 784]) \]

\[ W = \text{tf.Variable}(\text{tf.zeros}([784, 10])) \]

\[ b = \text{tf.Variable}(\text{tf.zeros}([10])) \]

\[ y = \text{tf.nn.relu}(\text{tf.matmul}(x, W) + b) \]
MultiLayer Perceptron
Learning is “Error Propagation”

Fig. 7.6. Extended network for the computation of the error function

\[
\text{mse} = \text{tf.reduce_min}(\text{tf.square}(y - y_{\_})) \\
\text{train\_step} = \text{tf.train.GradientDescentOptimizer}(0.3).\text{minimize}(\text{mse}) \\
\text{sess.run}((\text{train\_step, feed\_dict={x: batch\_xs, y\_: batch\_ys}}))
Learning in Multi Layer Perceptron

- Backpropagation
Learning: Back Propagation to Minimize Error

\[ v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \]

\[ y_j(n) = \varphi_j(v_j(n)) \]

\[ e_j(n) = d_j(n) - y_j(n) \]

\[ \mathcal{E}(n) = \sum_{j \in \mathcal{J}} \mathcal{E}_j(n) \]

\[ = \frac{1}{2} \sum_{j \in \mathcal{J}} e_j^2(n) \]

**FIGURE 4.3** Signal-flow graph highlighting the details of output neuron \( j \).
Activation Function

- Need to be differentiable
- Logistic

\[ \varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, \quad a > 0 \]

- tanh

\[ \varphi_j(v_j(n)) = a \tanh(bv_j(n)) \]

\[ \varphi(v) = a \tanh(bv) \]

\[ a = 1.7159 \quad b = \frac{2}{3} \]

\[ \varphi(1) = 1 \text{ and } \varphi(-1) = -1. \]
Back Propagation for $W_{ji}$ (Neuron $i$ to $j$)

\[ v_j(n) = \sum_{i=0}^{m} w_{ji}(n)y_i(n) \]

\[ \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \]

\[ y_j(n) = \varphi_j(v_j(n)) \]

\[ e_j(n) = d_j(n) - y_j(n) \]

\[ E(n) = \sum_{j \in C} e_j(n) \]

\[ = \frac{1}{2} \sum_{j \in C} e_j^2(n) \]

\[ \delta E/\delta w \]

\[ \Delta w_{ji}(n) \propto \frac{\partial E(n)}{\partial w_{ji}(n)} \]

\[ \Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \]
\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ij}}
\]

\[
\frac{\partial \text{net}_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left( \sum_{k=1}^{n} w_{kj} o_k \right) = o_i.
\]

\[
\frac{\partial o_j}{\partial \text{net}_j} = \frac{\partial}{\partial \text{net}_j} \varphi(\text{net}_j) = \varphi(\text{net}_j)(1 - \varphi(\text{net}_j))
\]

\[
\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y} = \frac{\partial}{\partial y} \frac{1}{2} (t - y)^2 = y - t
\]

\[
\frac{\partial E(o_j)}{\partial o_j} = \frac{\partial E(\text{net}_u, \text{net}_v, \ldots, \text{net}_w)}{\partial o_j}
\]

\[
\frac{\partial E}{\partial \text{net}_l} = \sum_{l \in L} \left( \frac{\partial E}{\partial o_l} \frac{\partial o_l}{\partial \text{net}_l} \right) = \sum_{l \in L} \left( \frac{\partial E}{\partial o_l} \frac{\partial o_l}{\partial \text{net}_l} w_{jl} \right)
\]

\[
\frac{\partial E}{\partial w_{ij}} = \delta_j o_i
\]

\[
\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} (o_j - t_j) o_j(1 - o_j) & \text{if } j \text{ is an output neuron,} \\ \frac{1}{(\sum_{l \in L} \delta_l w_{jl})} o_j(1 - o_j) & \text{if } j \text{ is an inner neuron.} \end{cases}
\]

\[
\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} = \begin{cases} -\alpha o_i (o_j - t_j) o_j(1 - o_j) & \text{if } j \text{ is an output neuron,} \\ -\alpha o_i (\sum_{l \in L} \delta_l w_{jl}) o_j(1 - o_j) & \text{if } j \text{ is an inner neuron.} \end{cases}
\]

From Wikipedia - Back Propagation
Accuracy Increment while learning

<table>
<thead>
<tr>
<th></th>
<th>Initially:</th>
<th>0th:</th>
<th>1th:</th>
<th>2th:</th>
<th>3th:</th>
<th>4th:</th>
<th>5th:</th>
<th>6th:</th>
<th>7th:</th>
<th>8th:</th>
<th>9th:</th>
<th>10th:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.098</td>
<td>0.3563</td>
<td>0.3622</td>
<td>0.3634</td>
<td>0.3712</td>
<td>0.3888</td>
<td>0.4237</td>
<td>0.5059</td>
<td>0.8551</td>
<td>0.8864</td>
<td>0.9123</td>
<td>0.9223</td>
</tr>
</tbody>
</table>
References


https://en.wikipedia.org/wiki/Backpropagation