Back Propagation

论文标题: Learning representations by back-propagating errors[1].

1 研究问题

对于简单的神经网络——输出单元直接与输入单元相连，我们很容易去学习连接过程的参数. 但当并没有直接定义的隐藏单元被引入到神经网络之中, 学习这些连接参数变得较为困难.

2 解决方法

提出反向传播的算法, 通过学习连接过程的参数使得实际值与目标值的偏差度量最小. *总偏差*定义为

$$E = \frac{1}{2}\sum_{c}\sum_{j}(y_{j,c} - d_{j,c})^2 \tag{1}$$

其中c是案例(输入-输出对)的索引, j是输出单元的索引, y是输出单元的实际值而d是目标值. 我们的*目标*变为寻找$\operatorname{argmin}_w E$.
不失一般性, 我们考虑单元j的输入x_j与输出y_j, 总输入x_j定义为下层单元输出值的线性函数:

$$x_j = \sum_i y_i w_{ji} \tag{2}$$

其中y_i为第i个单元的输出, w_{ji}表示连接第i个单元到第j个单元的参数.
每个单元都有其输出值, 通过非线性函数Sigmoid函数定义

$$y_j = \frac{1}{1+\mathrm{e}^{-x_j}} \tag{3}$$

按照反向传播的定义, 我们对E关于y求偏导. 固定(1)中的c, 对单元j的输出y_j求偏导

$$\frac{\partial E}{\partial y_j} = y_j - d_j \tag{4}$$

再对x_j求偏导

$$\frac{\partial y_j}{x_j} = y_j(1-y_j) \tag{5}$$

对(4)与(5)应用链式法则

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_j} = (y_j - d_j) \cdot y_j(1 - y_j) \tag{6}$$

再次利用链式法则与(2)对w_{ji}求偏导

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{w_{ji}} = \frac{\partial E}{\partial x_j} \cdot y_i \tag{7}$$

上式便可用于我们后续对于参数w_{ji}的学习.
注意到(2)与(3)的定义并不固定, 任何有界可微函数都可以替代, 但用一个线性函数将所有输入结合再用非线性函数可以大大简化学习的过程.

接下来我们便可以利用(7)来学习w来达到最小化E的目的. 利用最简单的梯度下降的方法, 每一次学习后我们都更新参数w.

$$\Delta w = -\varepsilon \frac{\partial E}{\partial w} \tag{8}$$

其中ε为我们设定的学习率. 这种方法收敛的并不是很快, 但相对简单也易实现. 在不失简单性和局部性的情况下，将当前梯度用于优化参数的速度而不是其位置，可以对上述方法进行改进

$$\Delta w(t) = -\varepsilon \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \tag{9}$$

其中t表示每次利用输入-输出对对整个网络进行优化的次数, α是介于0到1之间的指数衰减因子，它决定了当前梯度和早期梯度对权重变化的相对贡献。

---

1. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986). https://doi.org/10.1038/323533a0 ↩