MLP (Multi-layer perceptron)は、いわゆる ANN (Artificial neural network)だ。

以下のサイトが参考になる。

http://deeplearning.net/tutorial/mlp.html

http://www.codeproject.com/Articles/821348/Multilayer-Perceptron-in-Python

1 シグモイド？それとも tanh？

まず、sigmoid 関数よりも tanh 関数が良く使われるみたい。理由は、tanh 関数は「anti-symmetric」つまり、$f(-x) = -f(x)$なので、収束しやすいみたい。さらに、sigmoid 関数と同様に、微分が簡単。

$$f'(x) = 1 - f(x)^2$$

証明：

まず、tanh 関数の定義は、

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

そして、微分は、

$$\frac{d}{dx}\tanh(x) = \frac{4}{(e^x + e^{-x})^2}$$

一方、

$$1 - \tanh^2(x) = \frac{4}{(e^x + e^{-x})^2}$$

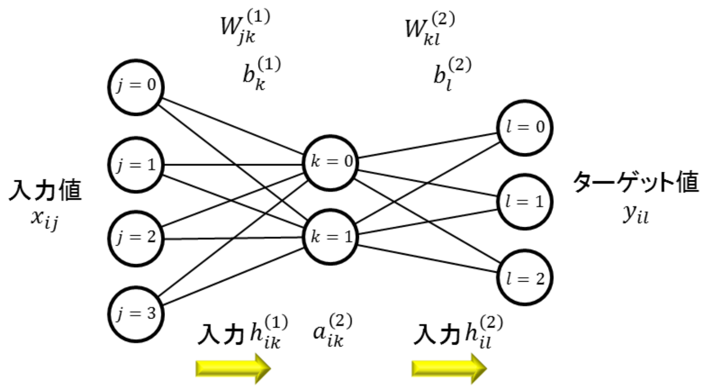
よって、

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$$

証明終わり。■

2 コスト関数と微分

まず、notation。



$a_{ik}^{(2)}$は、入力レイヤからの入力$h_{ik}^{(1)}$に対して activation 関数f（sigmoid や tanh）を適用した値。つまり、

$$\begin{aligned} a_{ik}^{(2)} &= f\left(h_{ik}^{(1)}\right) \\ &= f\left(\sum_j x_{ij} W_{jk}^{(1)} + b_k^{(1)}\right) \end{aligned} \tag{1}$$

また、出力レイヤの activation 関数をgとすると、

$$\begin{aligned} \hat{y}_{il} &= g\left(\sum_k a_{ik}^{(2)} W_{kl}^{(2)} + b_l^{(2)}\right) \\ &= g\left(\left[\sum_k f\left(\sum_j x_{ij} W_{jk}^{(1)} + b_k^{(1)}\right) W_{kl}^{(2)}\right] + b_l^{(2)}\right) \end{aligned} \tag{2}$$

とりあえず、今回は簡単のために、$g(x) = x$、つまり、線形変換を使用する。つまり、

$$\hat{y}_{il} = h_{il}^{(2)} = \sum_k a_{ik}^{(2)} W_{kl}^{(2)} + b_l^{(2)} \tag{3}$$

この時、コスト関数を squared error と L2 正規化で定義すると、

$$\begin{aligned} L &= \frac{1}{2N} \sum_{i=1}^N \sum_l \left(y_{il} - h_{il}^{(2)}\right)^2 + \frac{\lambda}{2}\left[\sum_j \sum_k {W_{jk}^{(1)}}^2 + \sum_k {b_k^{(1)}}^2 + \sum_k \sum_l {W_{kl}^{(2)}}^2 + \sum_l {b_l^{(2)}}^2\right] \\ &= \frac{1}{2N} \sum_{i=1}^N \sum_l \left[y_{il} - \sum_k a_{ik}^{(2)} W_{kl}^{(2)} - b_l^{(2)}\right]^2 + \frac{\lambda}{2}\left[\sum_j \sum_k {W_{jk}^{(1)}}^2 + \sum_k {b_k^{(1)}}^2 + \sum_k \sum_l {W_{kl}^{(2)}}^2 + \sum_l {b_l^{(2)}}^2\right] \end{aligned}$$

微分は、

$$\frac{\partial L}{\partial W_{kl}^{(2)}} = \frac{1}{N} \sum_{i=1}^N \left[y_{il} - \sum_k a_{ik}^{(2)} W_{kl}^{(2)} - b_l^{(2)}\right]\left(-a_{ik}^{(2)}\right) + \lambda W_{kl}^{(2)} = -\frac{1}{N} \sum_{i=1}^N \left(y_{il} - h_{il}^{(2)}\right) a_{ik}^{(2)} + \lambda W_{kl}^{(2)}$$

$$\frac{\partial L}{\partial b_l^{(2)}} = \frac{1}{N}\sum_{i=1}^{N}\left[y_{il} - \sum_k a_{ik}^{(2)} W_{kl}^{(2)} - b_l^{(2)}\right](-1) + \lambda b_l^{(2)} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_{il} - h_{il}^{(2)}\right) + \lambda b_l^{(2)}$$

また、

$$\begin{aligned}\frac{\partial L}{\partial W_{jk}^{(1)}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_l\left[y_{il} - \sum_k a_{ik}^{(2)} W_{kl}^{(2)} - b_l^{(2)}\right]\left(-W_{kl}^{(2)}\right)\frac{\partial a_{ik}^{(2)}}{\partial W_{jk}^{(1)}} + \lambda W_{jk}^{(1)} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_l\left(y_{il} - h_{il}^{(2)}\right) W_{kl}^{(2)} f'\left(h_{ik}^{(1)}\right) x_{ij} + \lambda W_{jk}^{(1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial b_k^{(1)}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_l\left[y_{il} - \sum_k a_{ik}^{(2)} W_{kl}^{(2)} - b_l^{(2)}\right]\left(-W_{kl}^{(2)}\right)\frac{\partial a_{ik}^{(2)}}{\partial b_k^{(1)}} + \lambda b_k^{(1)} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_l\left(y_{il} - h_{il}^{(2)}\right) W_{kl}^{(2)} f'\left(h_{ik}^{(1)}\right) + \lambda b_k^{(1)}\end{aligned}$$

たとえば、sigmoid 関数の場合、

$$f'\left(h_{ik}^{(1)}\right) = f(x)\big(1 - f(x)\big) = a_{ik}^{(2)}\left(1 - a_{ik}^{(2)}\right)$$

また、tanh 関数なら、

$$f'\left(h_{ik}^{(1)}\right) = 1 - [f(x)]^2 = 1 - \left[a_{ik}^{(2)}\right]^2$$

以下、tanh 関数を使用した場合について、まとめておく。

まとめ：

$$\begin{cases}\dfrac{\partial L}{\partial W_{kl}^{(2)}} = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(y_{il} - h_{il}^{(2)}\right) a_{ik}^{(2)} + \lambda W_{kl}^{(2)} \\ \dfrac{\partial L}{\partial b_l^{(2)}} = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(y_{il} - h_{il}^{(2)}\right) + \lambda b_l^{(2)} \\ \dfrac{\partial L}{\partial W_{jk}^{(1)}} = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\sum_l\left(y_{il} - h_{il}^{(2)}\right) W_{kl}^{(2)}\left(1 - \left[a_{ik}^{(2)}\right]^2\right) x_{ij} + \lambda W_{jk}^{(1)} \\ \dfrac{\partial L}{\partial b_k^{(1)}} = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\sum_l\left(y_{il} - h_{il}^{(2)}\right) W_{kl}^{(2)}\left(1 - \left[a_{ik}^{(2)}\right]^2\right) + \lambda b_k^{(1)}\end{cases}$$

3　凸なの？

一番右のレイヤについて、二階微分を計算する。

$$\frac{\partial^2 L}{\partial {W_{kl}^{(2)}}^2} = \frac{\partial}{\partial W_{kl}^{(2)}}\left[-\frac{1}{N}\sum_{i=1}^{N}\left(y_{il}-h_{il}^{(2)}\right)a_{ik}^{(2)}+\lambda W_{kl}^{(2)}\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N} a_{ik}^{(2)}\cdot a_{ik}^{(2)}+\lambda$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left[a_{ik}^{(2)}\right]^2+\lambda \geq 0$$

$$\frac{\partial^2 L}{\partial {b_{l}^{(2)}}^2} = \frac{\partial}{\partial b_{l}^{(2)}}\left[-\frac{1}{N}\sum_{i=1}^{N}\left(y_{il}-h_{il}^{(2)}\right)+\lambda b_{l}^{(2)}\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N} 1+\lambda \geq 0$$

ここまでは凸だ。でも、hidden レイヤが駄目なんだよね。

$$\frac{\partial^2 L}{\partial {W_{jk}^{(1)}}^2} = \frac{\partial}{\partial W_{jk}^{(1)}}\left[-\frac{1}{N}\sum_{i=1}^{N}\sum_{l}\left(y_{il}-h_{il}^{(2)}\right)W_{kl}^{(2)}f'\left(h_{ik}^{(1)}\right)x_{ij}+\lambda W_{jk}^{(1)}\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{l}\left(\left[W_{kl}^{(2)}f'\left(h_{ik}^{(1)}\right)x_{ij}\right]^2-\left(y_{il}-h_{il}^{(2)}\right)W_{kl}^{(2)}f''\left(h_{ik}^{(1)}\right)x_{ij}^2\right)+\lambda$$

$$\frac{\partial^2 L}{\partial {b_{k}^{(1)}}^2} = \frac{\partial}{\partial b_{k}^{(1)}}\left[-\frac{1}{N}\sum_{i=1}^{N}\sum_{l}\left(y_{il}-h_{il}^{(2)}\right)W_{kl}^{(2)}f'\left(h_{ik}^{(1)}\right)+\lambda b_{k}^{(1)}\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{l}\left(\left[W_{kl}^{(2)}f'\left(h_{ik}^{(1)}\right)\right]^2-\left(y_{il}-h_{il}^{(2)}\right)W_{kl}^{(2)}f''\left(h_{ik}^{(1)}\right)\right)+\lambda$$

というわけで、凸ではない。これが、ニューラルネットワークが万能学習器であるにもかかわらず、実際には万能でないと言われる理由だ。