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>

# 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、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。

**証明：**  
まず、tanh関数の定義は、

そして、微分は、

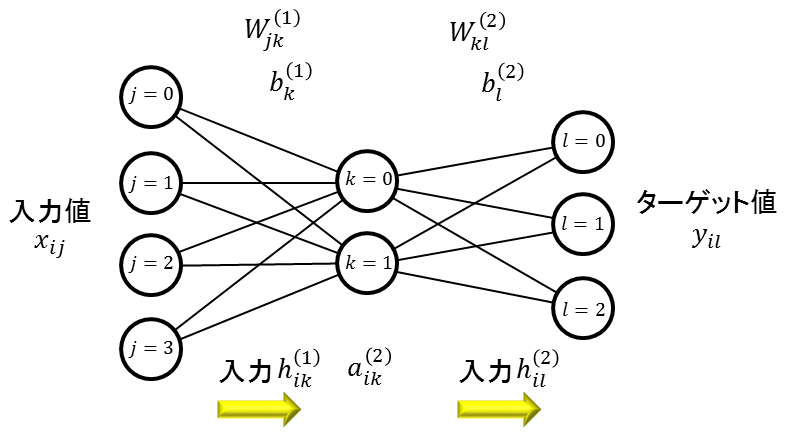
一方、

よって、

証明終わり。■

# コスト関数と微分

まず、notation。



は、入力レイヤからの入力に対してactivation関数（sigmoidやtanh）を適用した値。つまり、

(1)

また、出力レイヤのactivation関数をとすると、

(2)

とりあえず、今回は簡単のために、、つまり、線形変換を使用する。つまり、

(3)

この時、コスト関数をsquared errorとL2正規化で定義すると、

微分は、

また、

たとえば、sigmoid関数の場合、

また、tanh関数なら、

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：**

# 凸なの？

一番右のレイヤについて、二階微分を計算する。

ここまでは凸だ。でも、hiddenレイヤが駄目なんだよね。

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。