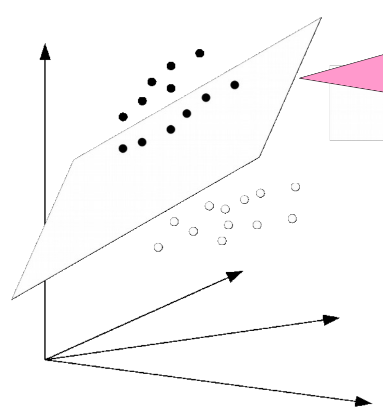
6. 識別 - ニューラルネットワーク -

- 識別関数法
  - 確率の枠組みにはとらわれず、

    $$f_{Positive}(\boldsymbol{x}) > f_{Negative}(\boldsymbol{x})$$

    ならば ***x*** を Positive と判定する関数 *f* を推定する
  - 単層パーセプトロン
    - 識別関数として 1 次式（＝直線・平面）を仮定

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w} \cdot \boldsymbol{x}$$



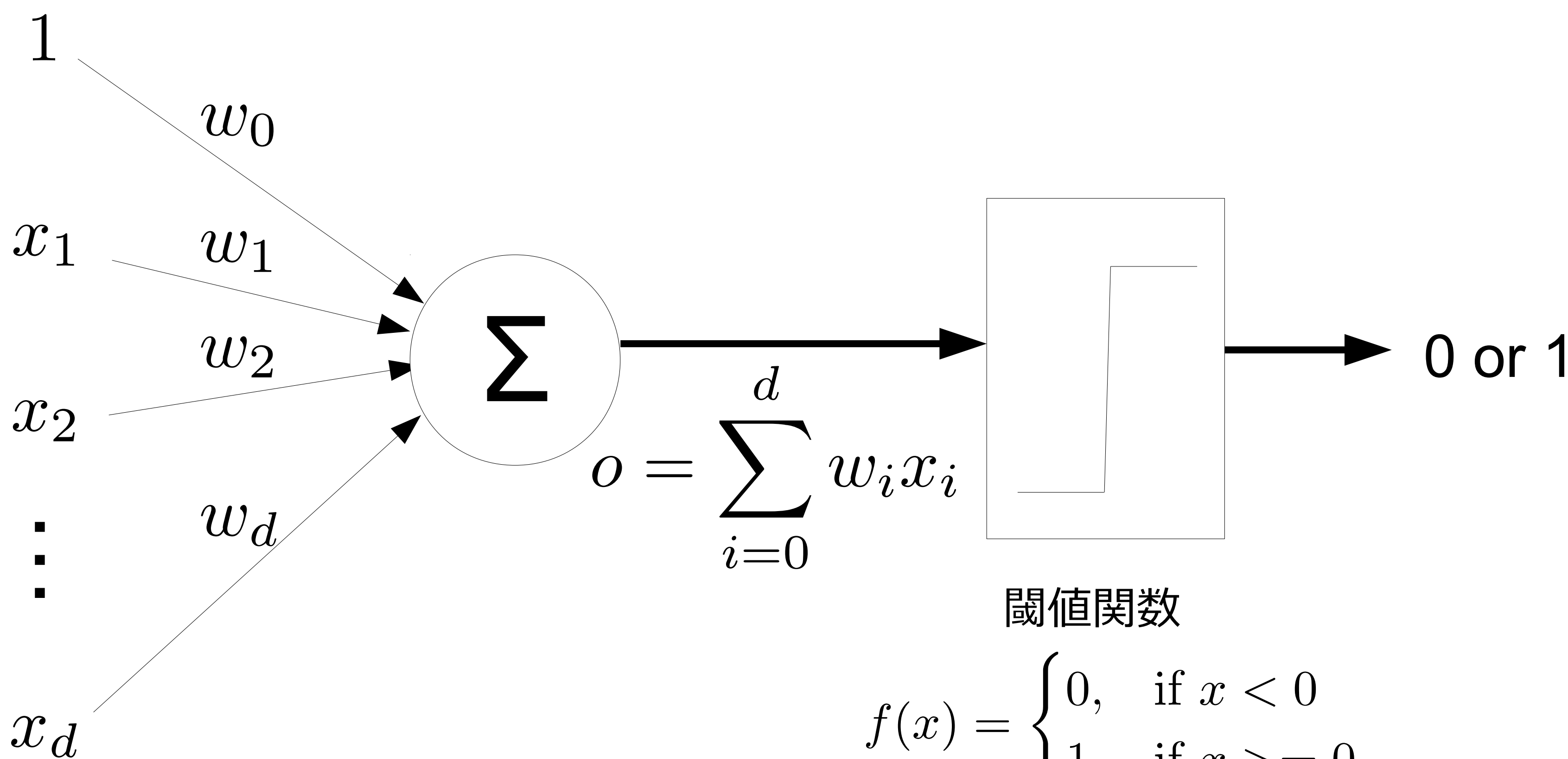
この平面を
求めている
ことになる

パーセプトロンの学習

- 単層パーセプトロンの定義

以後、$\boldsymbol{w}$ は w_0 を含む

  - $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ という特徴空間上の超平面を表現



$$o = \sum_{i=0}^{d} w_i x_i$$

闘値関数

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x >= 0 \end{cases}$$

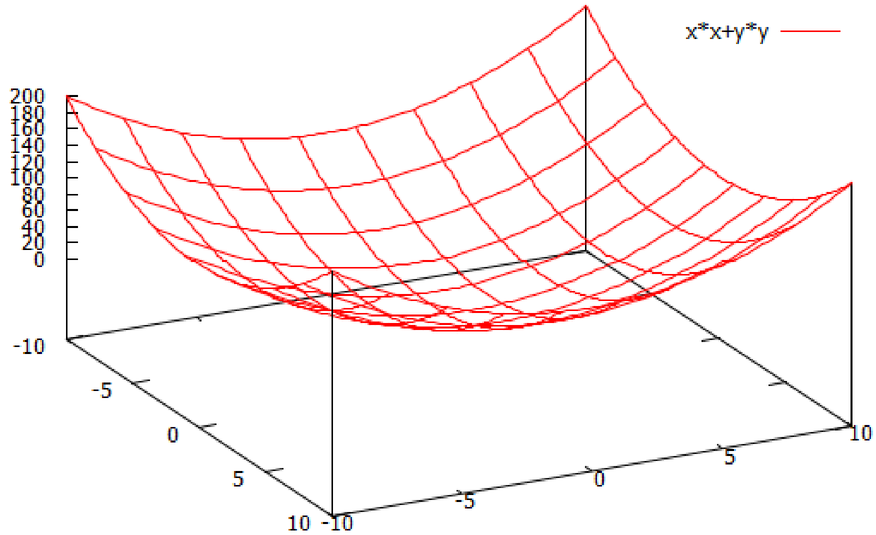
最急降下法

- エラーの定義
  - 二乗誤差

$$E(\mathbf{w}) \equiv \frac{1}{2} \sum_{\boldsymbol{x}_i \in D} (y_i - o_i)^2$$

全データに対する正解と関数の出力との差の２乗和

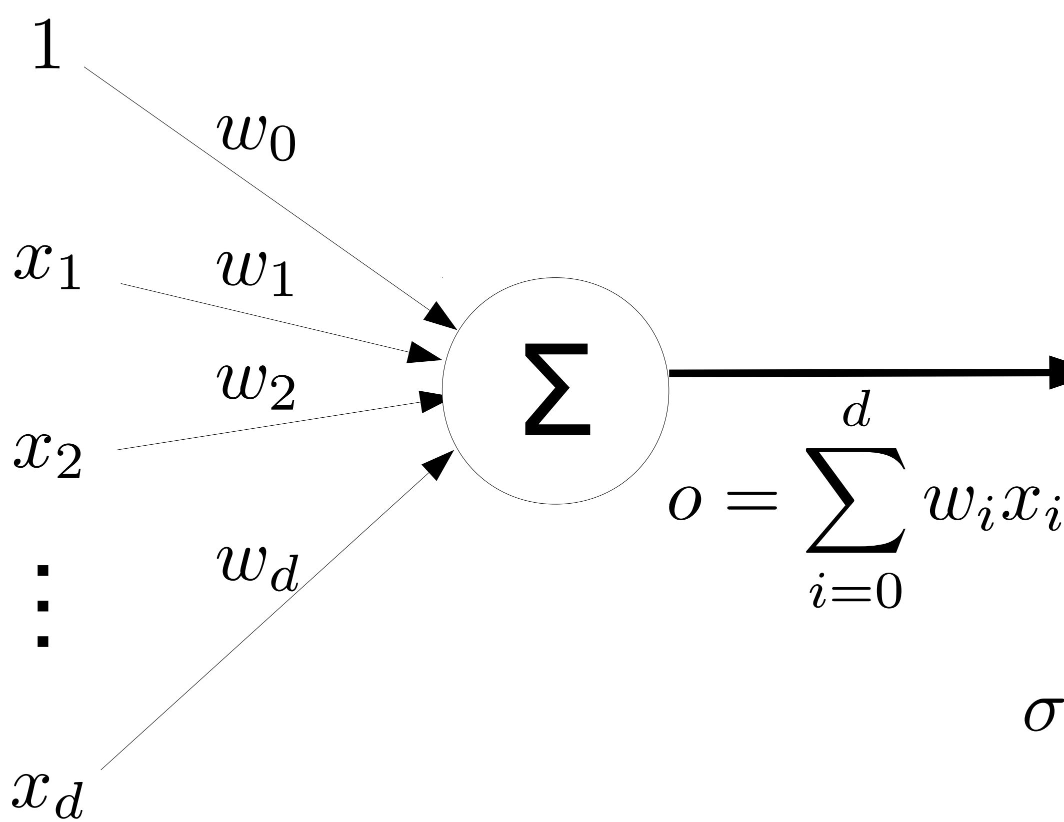
  - E は w の関数
    - w を E の勾配方向へ一定量だけ動かすことを繰り返して、最適解へ収束させる（→最急降下法）



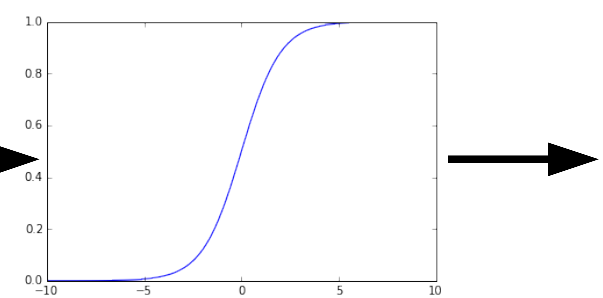
$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

多層パーセプトロンへの拡張

- シグモイド関数の適用
  - 勾配計算の際に微分可能なものを用いる



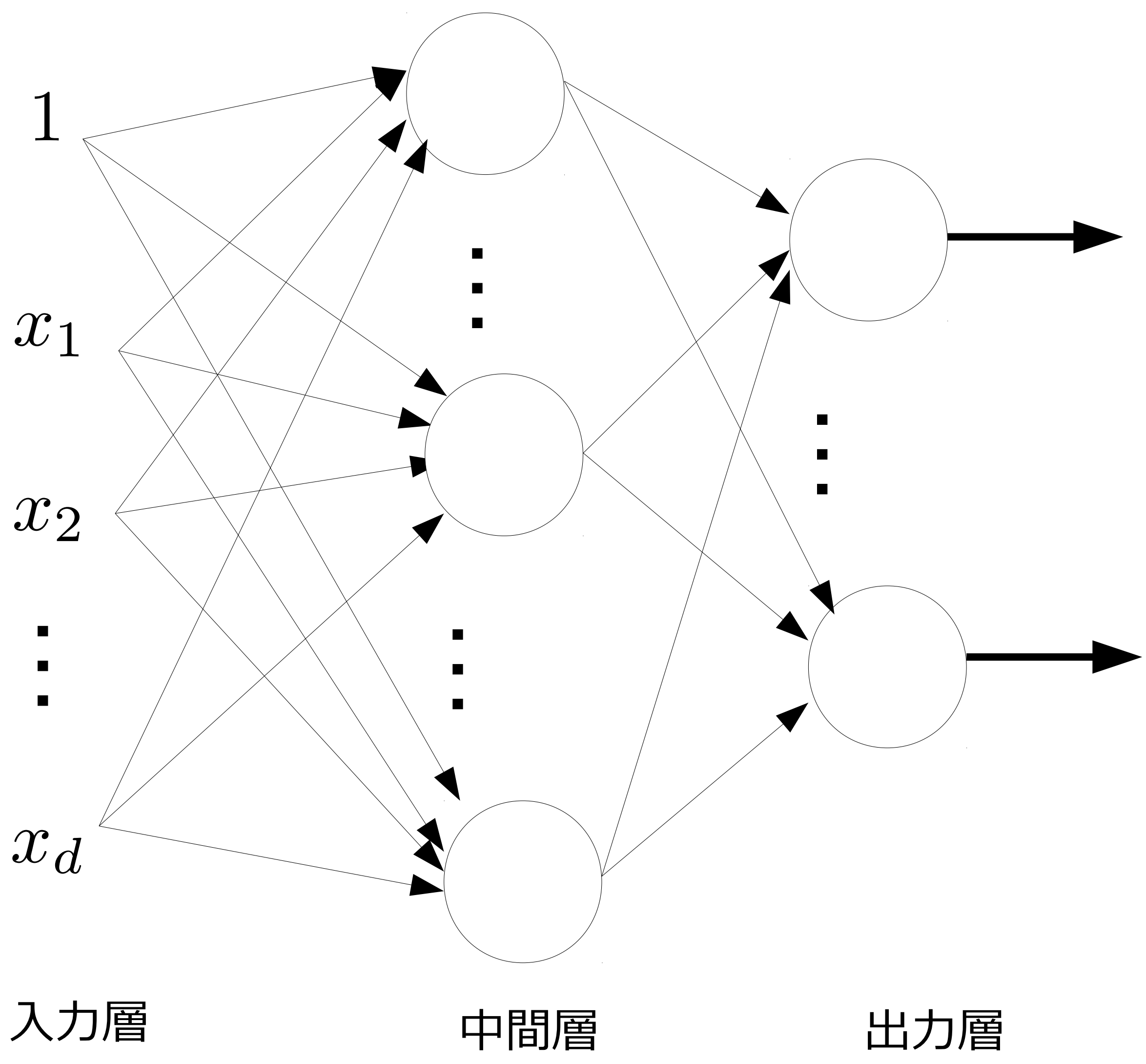
$$o = \sum_{i=0}^{d} w_i x_i$$



$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

$$\sigma'(y) = \sigma(y) \cdot (1 - \sigma(y))$$

多層パーセプトロンの構成



誤差逆伝播法による学習

1. リンクの重みを小さな初期値に設定
2. 個々の学習データ $(\mathbf{x}_i, t_i)$ に対して以下繰り返し

 a)入力 $\mathbf{x}_i$ に対するネットワークの出力 $\mathbf{o}_i$ を計算

 b)出力層の k 番目のユニットに対してエラー量 δ 計算

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

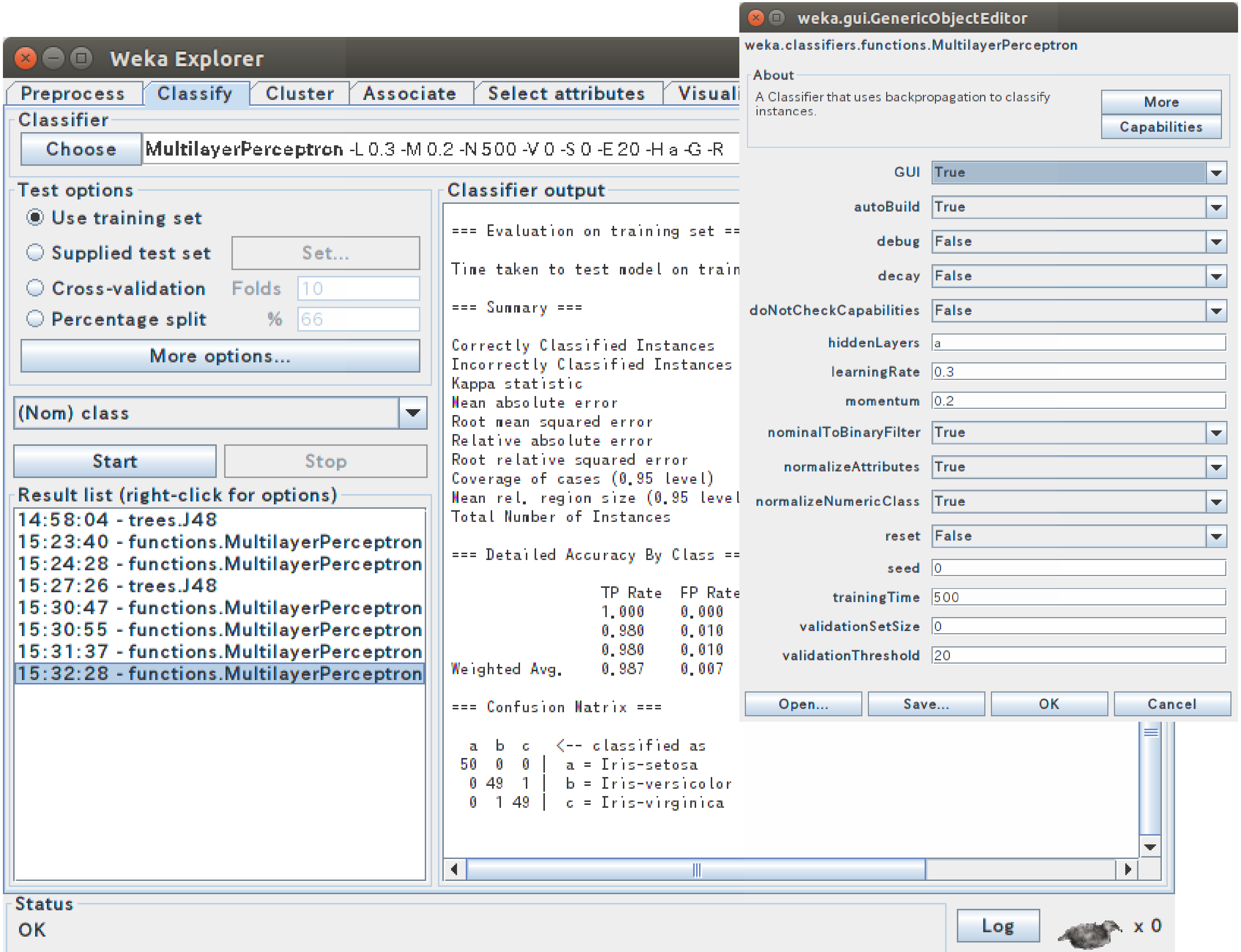
 c)中間層の h 番目のユニットに対してエラー量 δ 計算

$$\delta_k \leftarrow o_k(1 - o_k) \sum_{k \in outputs} w_{kh}\delta_k$$

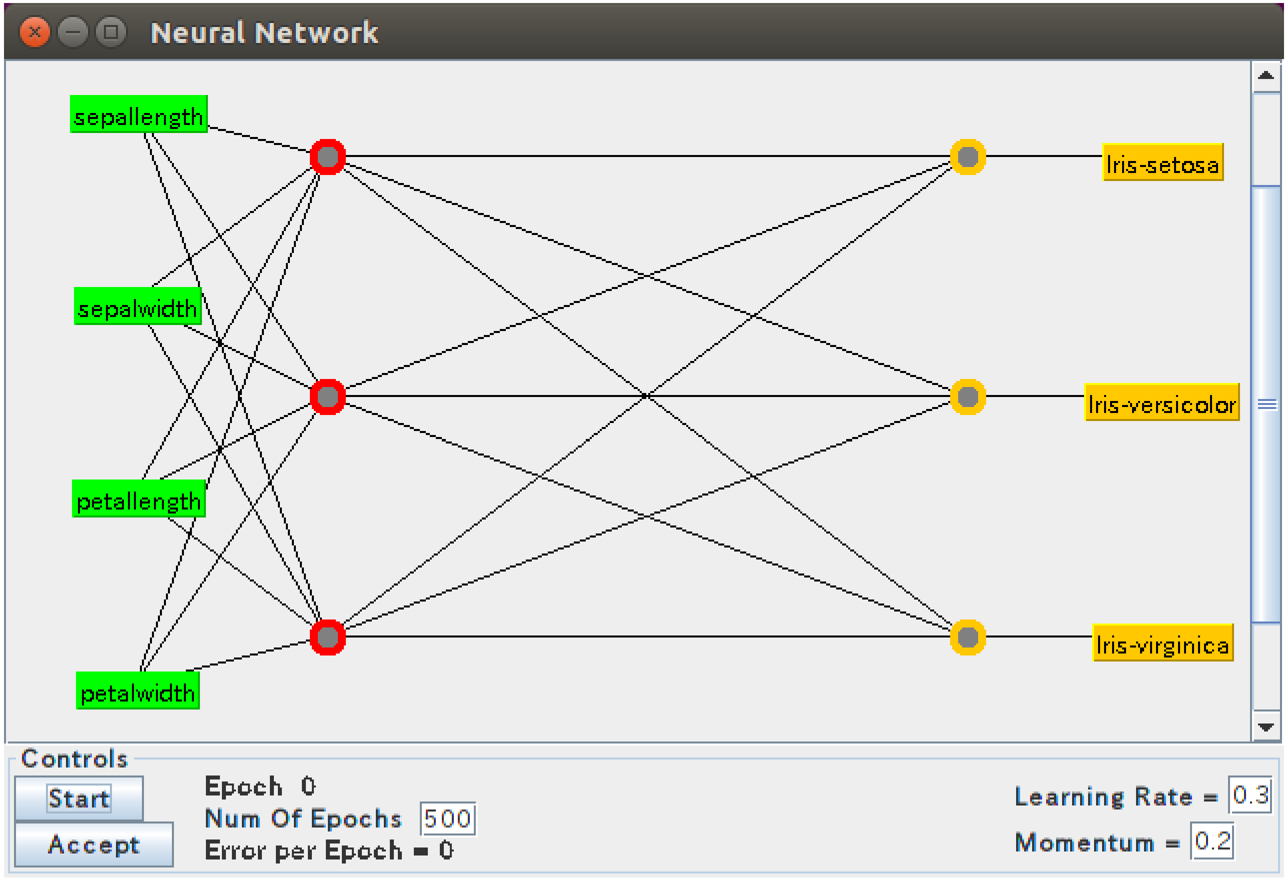
 d)重みの更新

$$w_{ji} \leftarrow w_{ji} + \eta\delta_j x_{ji}$$

Weka の MultilayerPerseptron



Weka の MultilayerPerseptron



多層パーセプトロンの特質

- 識別面の複雑さ
  - 中間層のニューロンの個数に関係する
  - シグモイド関数（非線形）を任意の重み・方向で足し合わせることで複雑な非線形識別面を構成