

## 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、

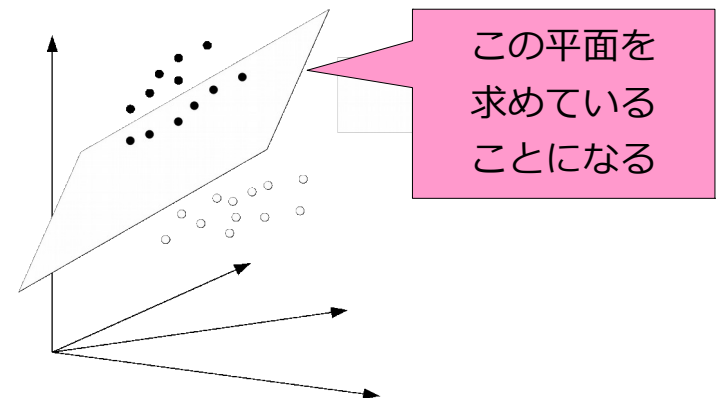
$$f_{Positive}(\boldsymbol{x}) > f_{Negative}(\boldsymbol{x})$$

ならば  $\boldsymbol{x}$  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

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

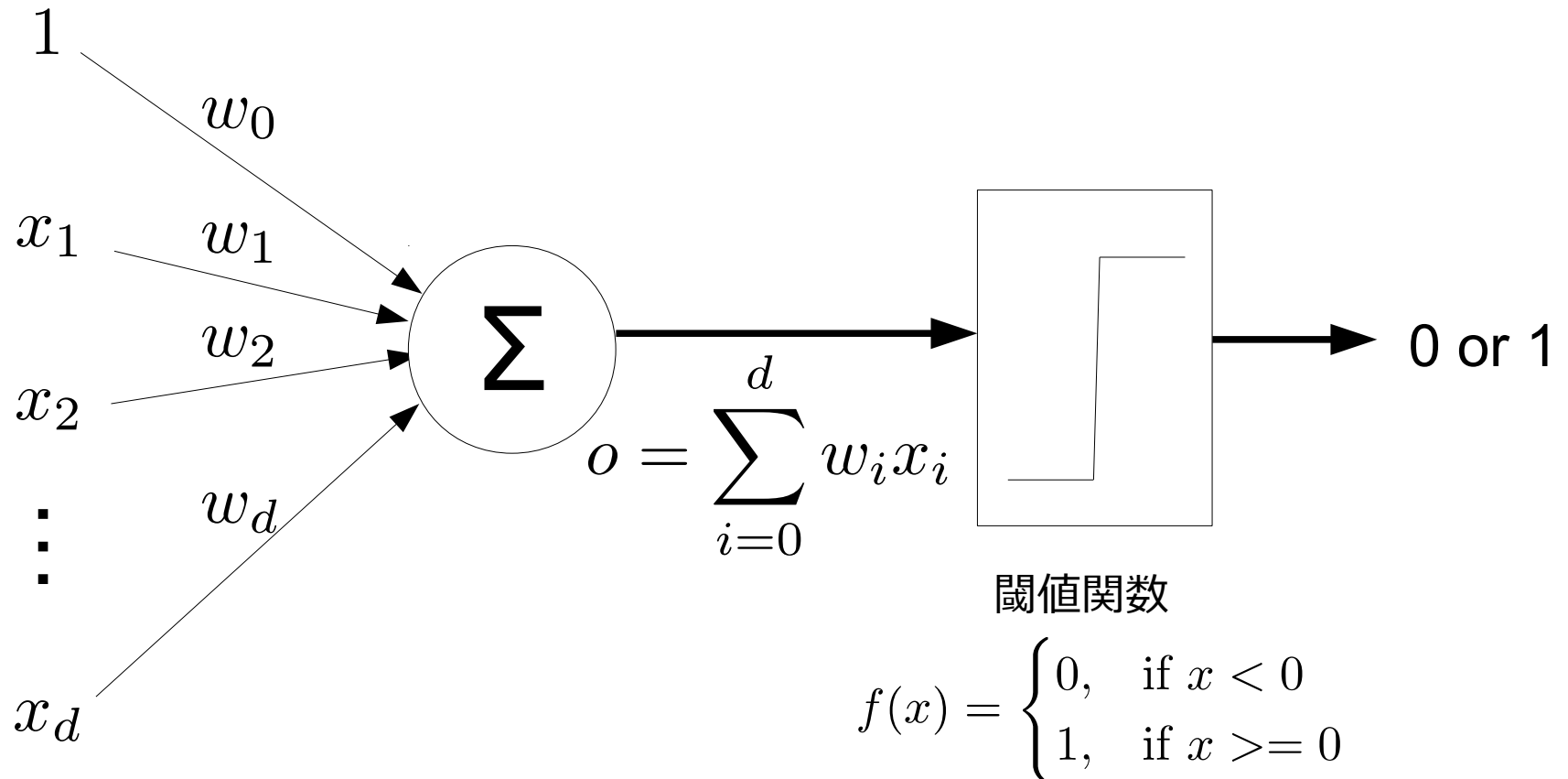


# パーセプトロンの学習

- 単層パーセプトロンの定義

以後、 $w$  は  $w_0$  を含む

- $w \cdot x = 0$  という特徴空間上の超平面を表現



# 最急降下法

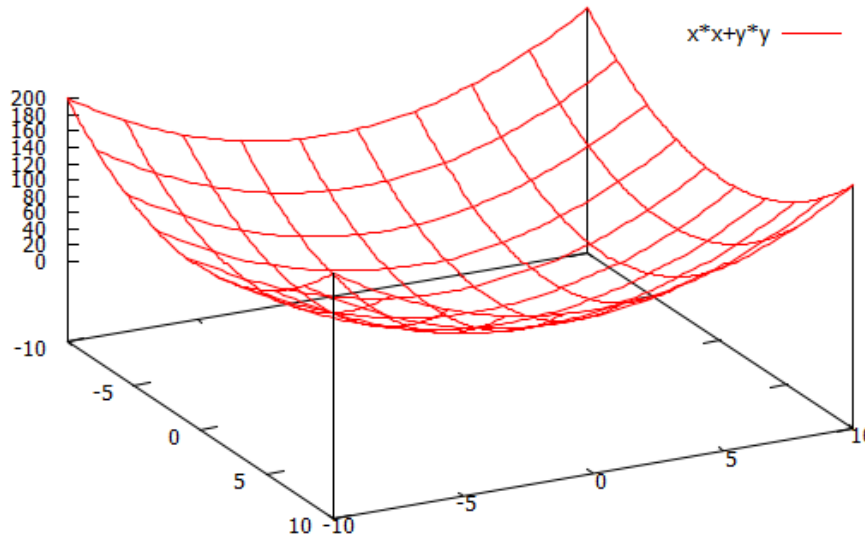
- エラーの定義

- 二乗誤差  $E(\mathbf{w}) \equiv \frac{1}{2} \sum_{x_i \in D} (y_i - o_i)^2$

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

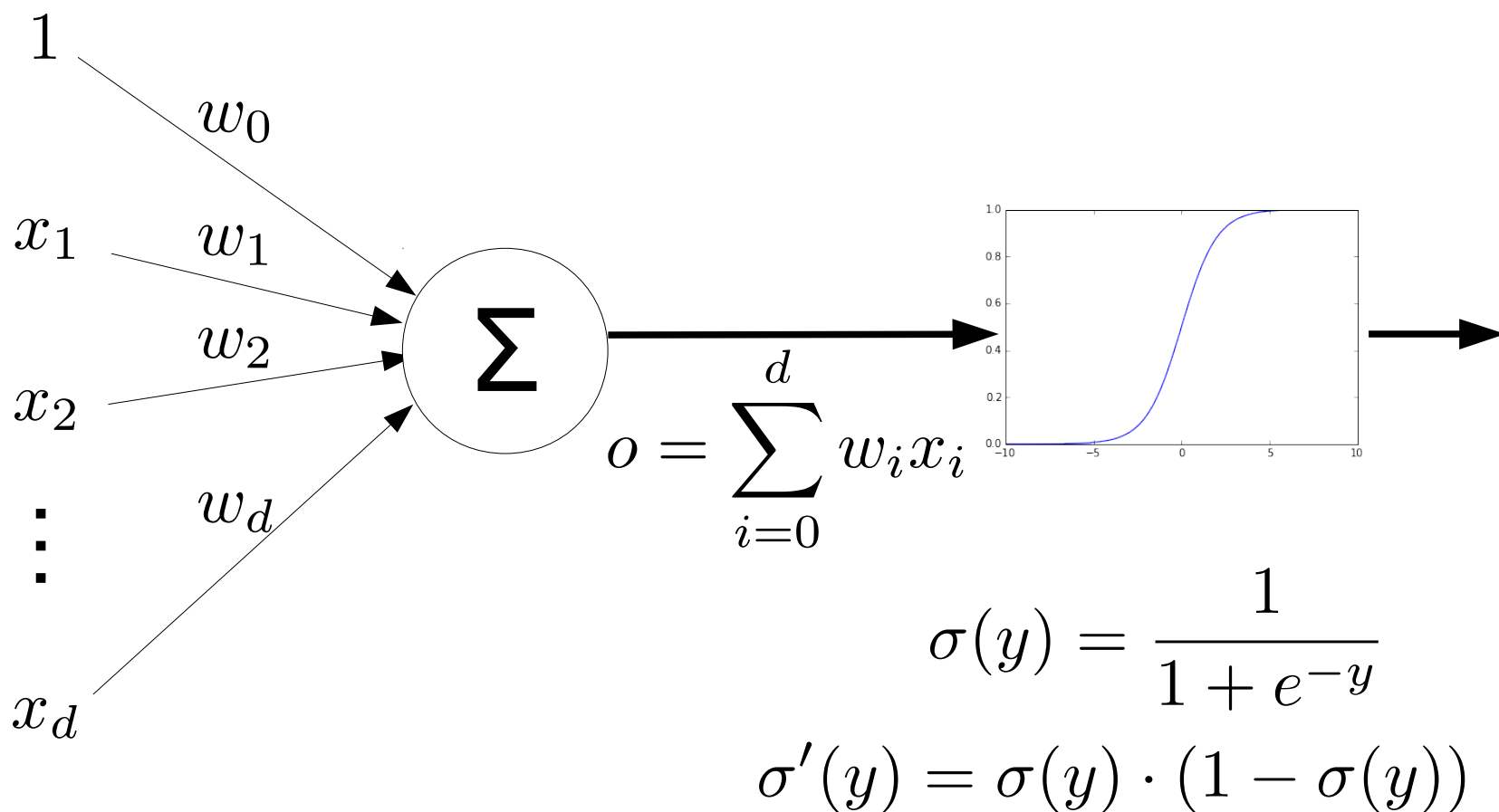
- w を E の勾配方向へ一定量だけ動かすことを繰り返して、最適解へ収束させる (→最急降下法)



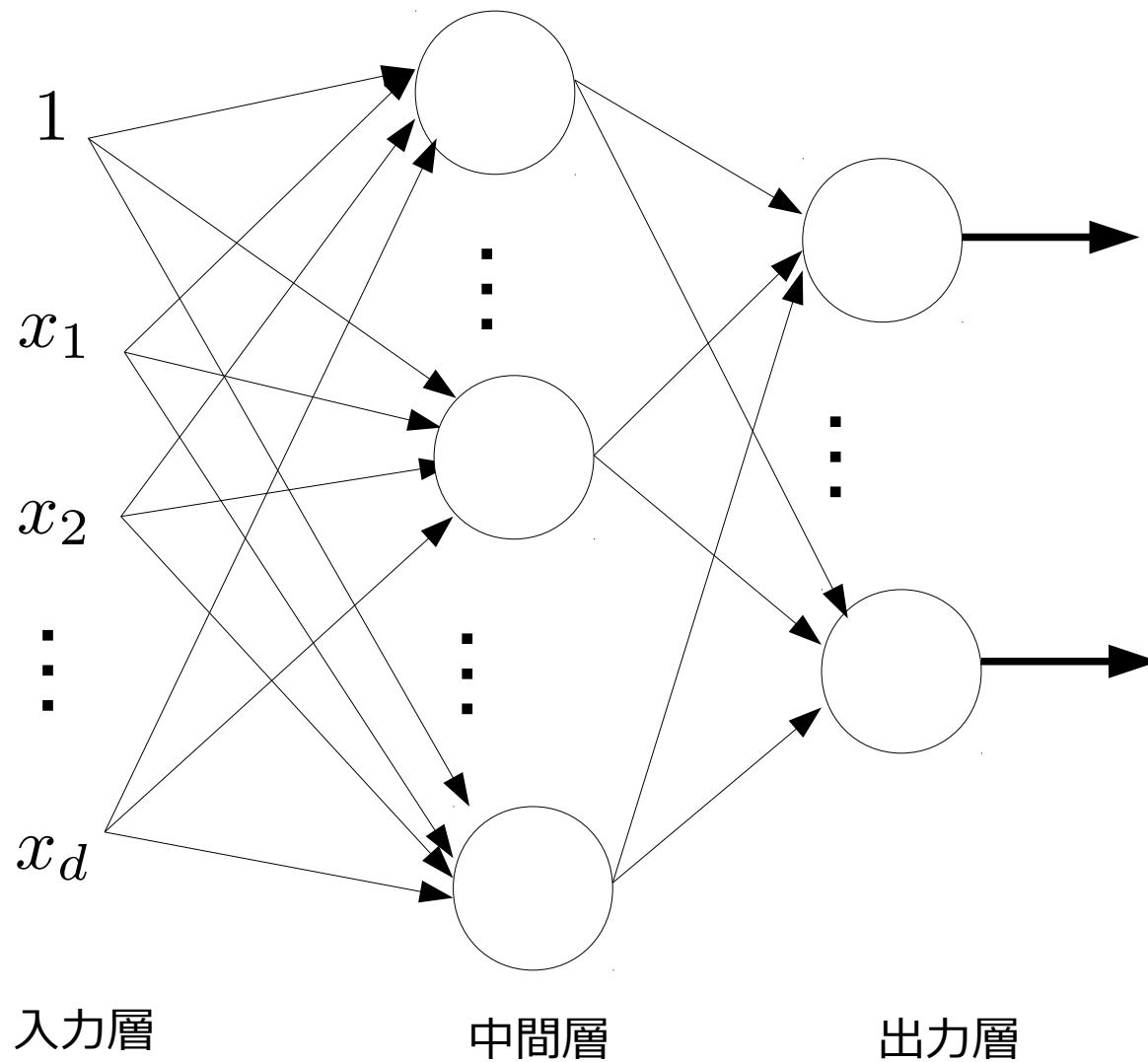
$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

# 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



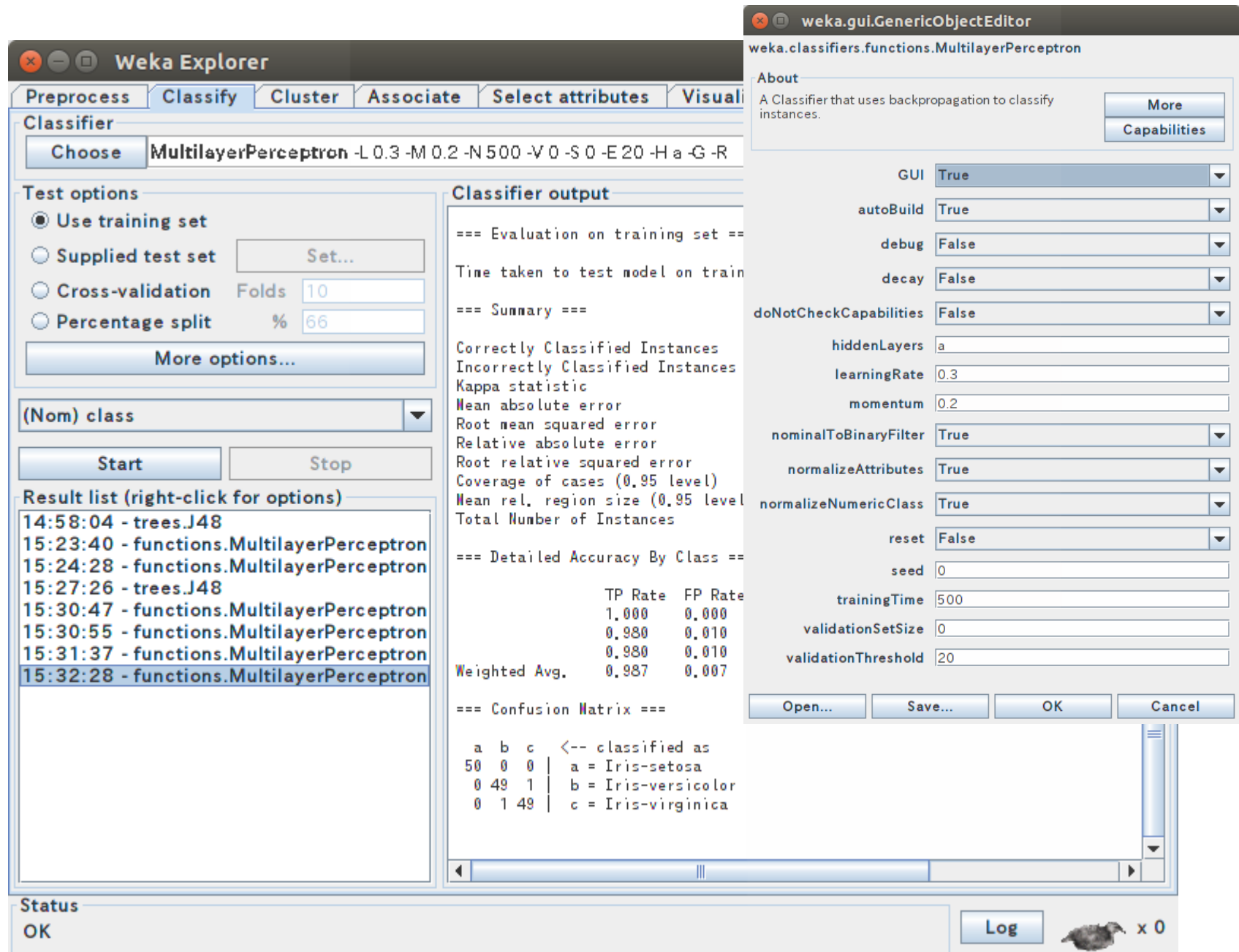
# 多層パーセプトロンの構成



# 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  $\delta$  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  $\delta$  計算
$$\delta_k \leftarrow o_k(1 - o_k) \sum_{k \in \text{outputs}} w_{kh} \delta_k$$
  - d) 重みの更新
$$w_{ji} \leftarrow w_{ji} + \eta \delta_j x_{ji}$$

# Weka の MultilayerPerceptron



The image shows the Weka Explorer interface with the MultilayerPerceptron classifier selected. The 'Classify' tab is active, and the 'Test options' section shows 'Use training set' selected. The 'Classifier output' section displays the results of the evaluation on the training set, including a summary of performance metrics and a detailed accuracy by class table. A separate window, 'weka.gui.GenericObjectEditor', is open, showing the configuration parameters for the MultilayerPerceptron classifier.

**Weka Explorer - Classifier**

Choose **MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R**

**Test options**

- ☒ Use training set
- ☐ Supplied test set
- ☐ Cross-validation Folds
- ☐ Percentage split %

**(Nom) class**

**Result list (right-click for options)**

- 14:58:04 - trees.J48
- 15:23:40 - functions.MultilayerPerceptron
- 15:24:28 - functions.MultilayerPerceptron
- 15:27:26 - trees.J48
- 15:30:47 - functions.MultilayerPerceptron
- 15:30:55 - functions.MultilayerPerceptron
- 15:31:37 - functions.MultilayerPerceptron
- 15:32:28 - functions.MultilayerPerceptron

**Classifier output**

=== Evaluation on training set ===

Time taken to test model on train

=== Summary ===

Correctly Classified Instances  
Incorrectly Classified Instances  
Kappa statistic  
Mean absolute error  
Root mean squared error  
Relative absolute error  
Root relative squared error  
Coverage of cases (0.95 level)  
Mean rel. region size (0.95 level)  
Total Number of Instances

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate
1.000	0.000	
0.980	0.010	
0.980	0.010	
Weighted Avg.	0.987	0.007

=== Confusion Matrix ===

a	b	c	<-- classified as
50	0	0	a = Iris-setosa
0	49	1	b = Iris-versicolor
0	1	49	c = Iris-virginica

**weka.gui.GenericObjectEditor**

weka.classifiers.functions.MultilayerPerceptron

**About**

A Classifier that uses backpropagation to classify instances.

**GUI**

**autoBuild**

**debug**

**decay**

**doNotCheckCapabilities**

**hiddenLayers**

**learningRate**

**momentum**

**nominalToBinaryFilter**

**normalizeAttributes**

**normalizeNumericClass**

**reset**

**seed**

**trainingTime**

**validationSetSize**

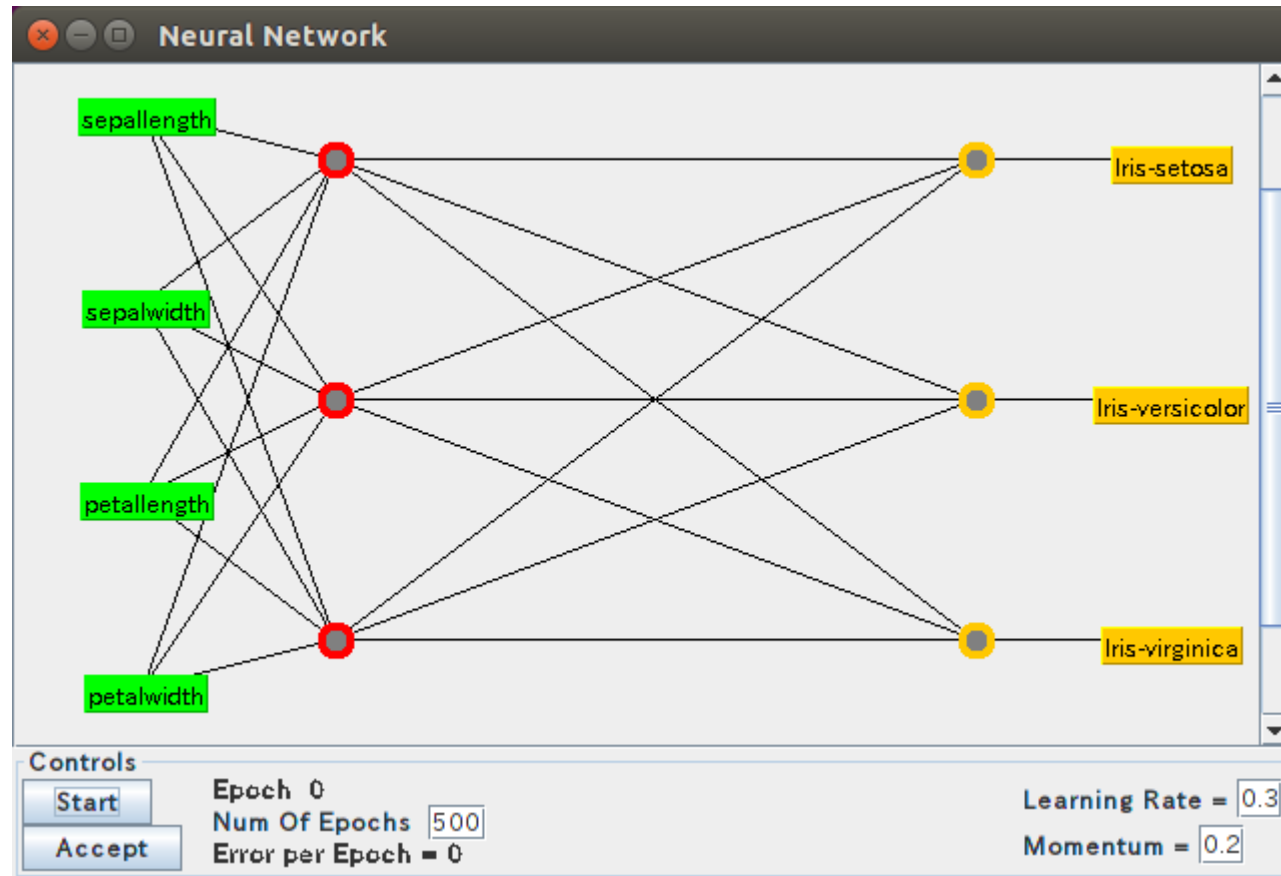
**validationThreshold**

**Status**

OK

x 0

# 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

