

## ニューラルネットワーク サンプルモデル

### 1. 概要

CommonMPは、演算要素モデルを組み合わせることで演算を行うためのプラットフォームです。従って、演算要素はどのようなモデルでもCommonMP上で動作させることができます。

本サンプルでは、色々な演算要素モデルを組み込む場合の例として、ニューラルネットワークを組み込んだ場合を取り上げます。本サンプルで示す様な「ニューラルネットワーク演算要素モデル」等を、現実の 水理／水文等のモデルとして CommonMPの内部でどのように使用するかを検討するのは、モデル開発者が考えなければなりません。

此处では、一般的バックプロパゲーション学習によるニューラルネットワークのサンプルを作成し、それを、CommonMPの演算要素モデルに組み込む場合を例としてコーディングしています。

一般には、時系列データを扱う為に、リカレントタイプのニューラルネットワークを組み込む等の色々な工夫が必要ですが、本モデルはサンプルなので、そこまで深く検討したニューラルネットワークではありません。

学習のサンプルとしては、一般的にニューラルネットワークの問題で扱われることの多い、XOR(排他的論理和)学習を取り上げました。

### 2. モデル概要

ニューラルネットワーク関連のサンプルとしては、次の2つの演算要素モデルを作成しました。

- ①ニューラルネットワーク演算要素モデル ([NeuralNetworkModel](#))
- ②上記モデルテスト用、XOR学習／連想信号発生演算要素モデル ([DmyDataModel](#))

#### 2. 1 ニューラルネットワーク演算要素モデル ([NeuralNetworkModel](#))

図2. 1にニューラルネットワーク演算要素モデルの構造の模式図を示します。

ニューラルネットワーク演算要素モデルが、ニューラルネットワーク本体を、HasA 関係で保持しています。学習／連想の ニューラルネットワーク本来の機能は全て、ニューラルネットワーク本体部分が担当し、ニューラルネットワーク演算要素モデルは、他の演算要素モデルとの接続処理だけを行っています。

本コーディングサンプルにおいて、ニューラルネットワーク本体は、下記オブジェクトにより構成されています。

##### (1)個別ニューロンオブジェクト ([NN\\_Neuron](#))

個別のニューロン処理を行います。入力層では、本ニューロンより派生した 入力ニューロンオブジェクトが用いられています。閾値学習として バイアスニューロン(常に 1 を出力)オブジェクトも存在します。

##### (2)ニューロン層オブジェクト ([NN\\_Layer](#))

複数のニューロンオブジェクトからなり、一つの層を形成します。閾値学習のために、出力層を除く各層には、バイアスニューロンが1個組み込んであります。入力層以外は、上記個別ニューロンオブジェクト(+バイアスニューロン 但し、出力層はバイアスニューロン数は0)から成りますが、入力層は、入力ニューロン(+バイアスニューロン)から構成されます。

(3) 結合ネットワークオブジェクト ( `NN_Network` )  
各層内のニューロン同士を結合する重みを保持するオブジェクトです。

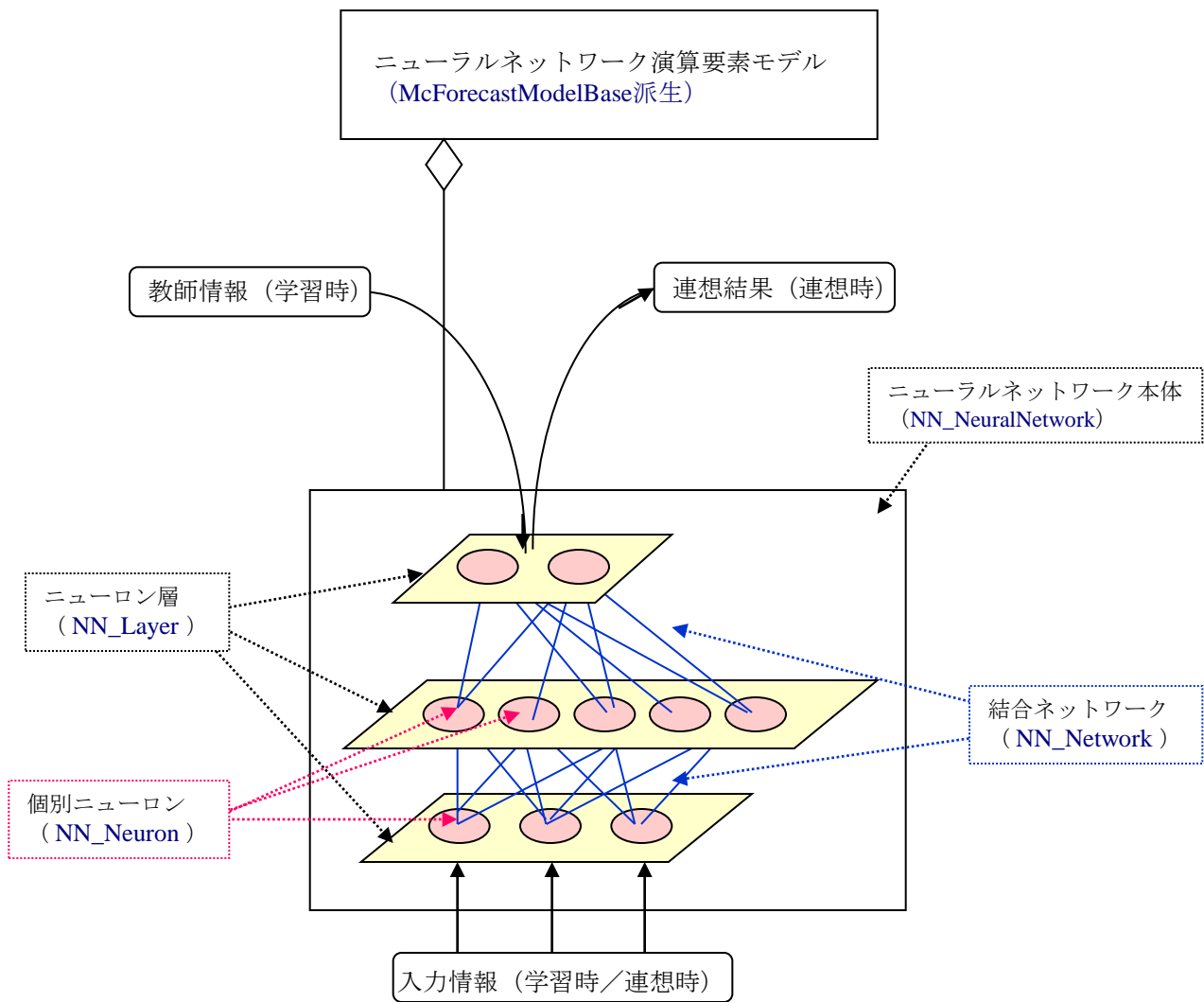


図 2. 1 ニューラルネットワーク演算要素モデル構造

ニューラルネットワーク本体は、単純なバックプロパゲーション学習を行います。

ニューラルネットワークの変数としては、

- ・ 各ニューロンのシグモイド関数の傾き(大きいほど、非線形性が強くなる)
- ・ 学習係数
- ・ 学習時の慣性係数

が設定できます。

ニューラルネットワーク本体オブジェクトは、ニューロン層の数は制限していませんが、ニューラルネットワーク演算要素モデル側で、3層または4層のネットワークを生成する様に規制を設けています。

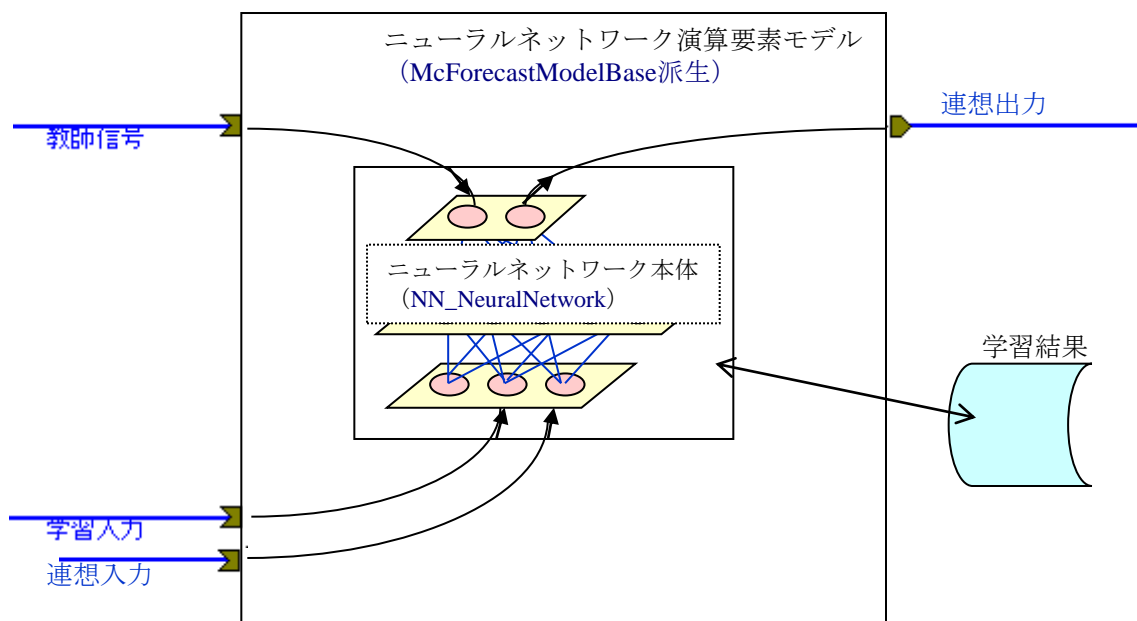


図 2. 2 本サンプルにおける  
ニューラルネットワーク演算要素モデルへの入出力接続

図2. 2は、本サンプルにおける、ニューラルネットワーク演算要素モデルへの入出力と、内部のネットワークへの入出力の関係を示します。

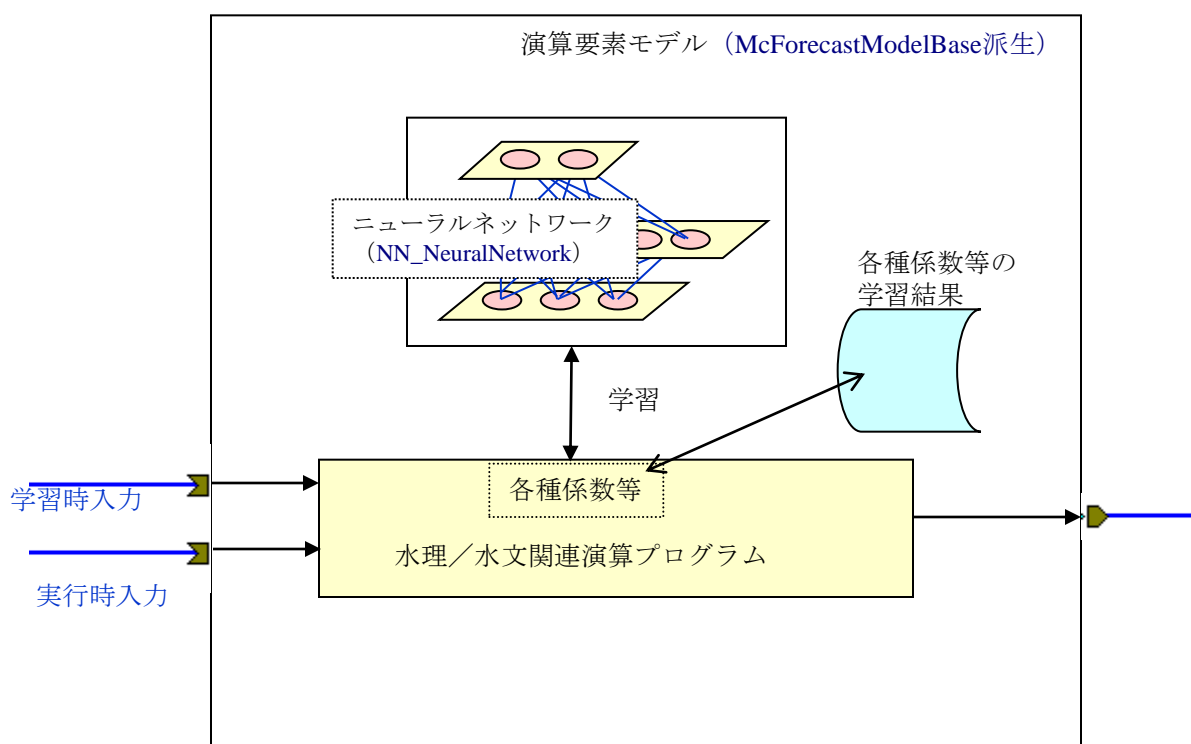


図 2. 3 ニューラルネットワーク学習機能を利用した  
水理／水文演算要素モデルの一例

実際の運用としては、図2. 3に示すように、水理／水文モデルの各種係数等の学習にニューラルネットワークを使用する場合等が 考えられますが、本サンプルではそこまで深く考慮しておりません。

2. 2 ニューラルネットワーク演算要素テスト用データ発生モデル  
(DmyDataModel)

XOR学習用の入力／及び教師信号として下記入力パターンを発生させます。

パターン番号	入力パターン		教師パターン
	第1ニューロン	第2ニューロン	出力ニューロン
①	0.0	0.0	0.0
②	1.0	0.0	1.0
③	0.0	1.0	1.0
④	1.0	1.0	0.0

連想時には、 学習時の入力パターンと同じパターンを発生させます。

本サンプルにおいては、学習時のパターン数は、4個であるため、 伝送データのセル内の変数配列に、 各パターンを設定しました。(図2. 4図参照)  
各ニューロンには、伝送データの各セルが対応しています。  
但し、この学習パターン設定は、本サンプル用に限ってコーディングしたもので、ニューラルネットワークの使用方法に応じて、 パターンの与え方は、検討する必要があります。

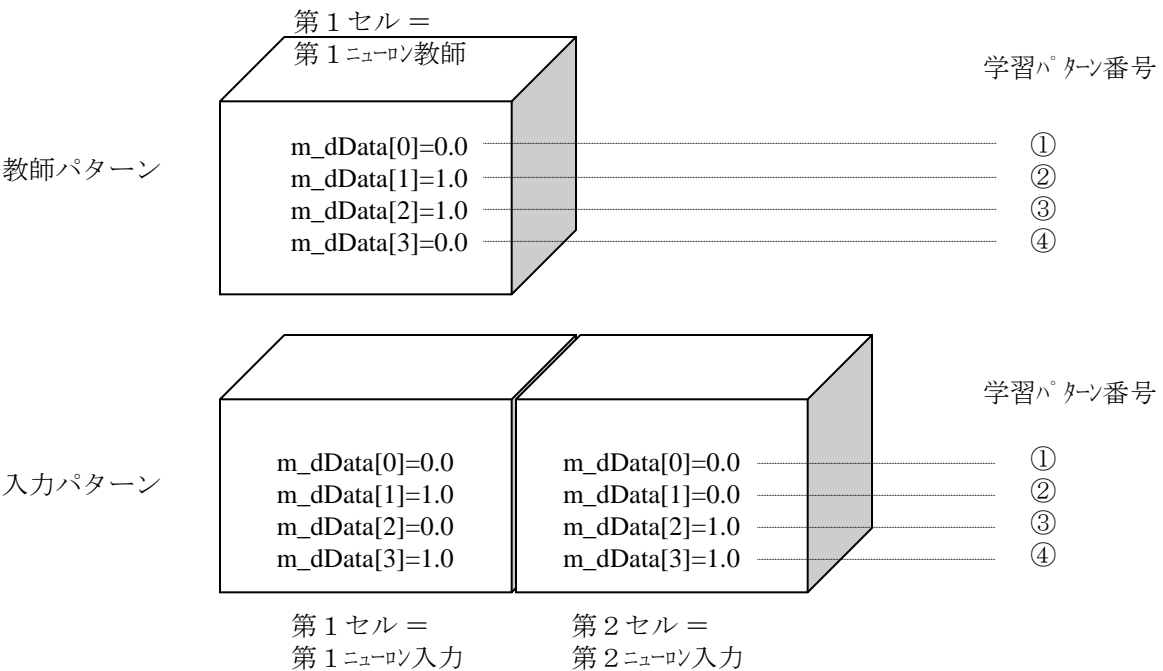


図2. 4 学習時の各パターンの設定

連想時のパターンの設定も、学習時のパターンの設定と同様としております。  
 (図2. 5図参照)  
 各ニューロンには、伝送データの各セルが対応しています。

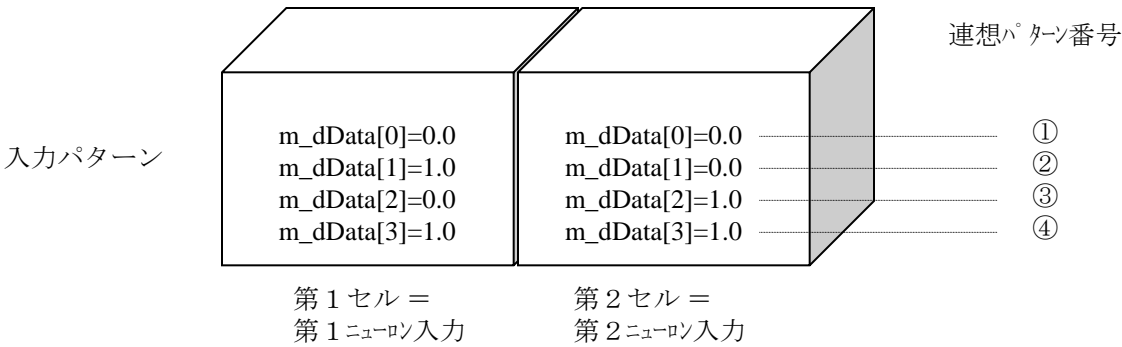


図 2. 5 連想時の各パターンの設定

### 3. モデル接続例

(構造定義ファイル: NeuralNetModelSample.xml)

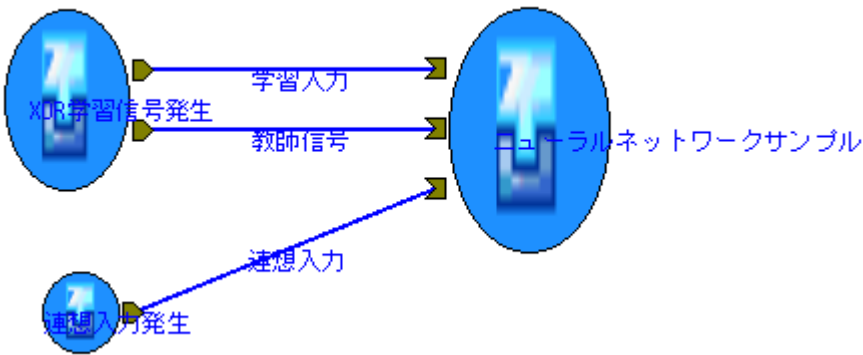


図 3. 1 ニューラルネットワークサンプル接続

モデルの接続例を 図3. 1に示します。 構造定義ファイル: NeuralNetModelSample.xml を読み込めば、上記モデルが生成されます。

本サンプルでは、ニューラルネットワークの動作の確認は、全て、 ディバッグ出力で行って  
います。

¥conf¥HymcoModelDebug.cfg に NeuralNetworkModel と追加すれば  
そのディバッグ用出力が モデル管理画面の「演算ログ」部に 表示されます。

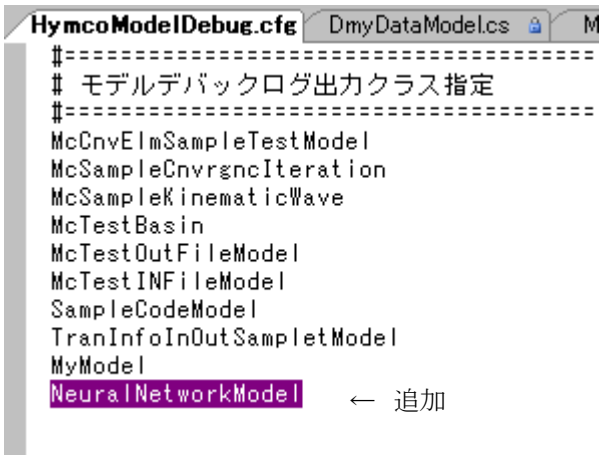


図 3. 2 HymcoModelDebug.cfg へのモデル追加

先ず、サンプルニューラルネットワークの プロパティ画面の初期設定から  
学習モード(0)に指定してください。（図3. 3参照）

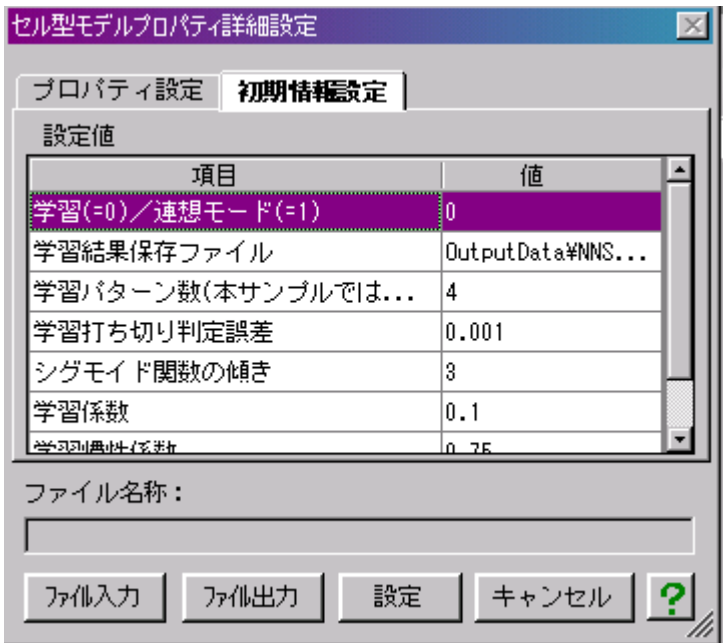
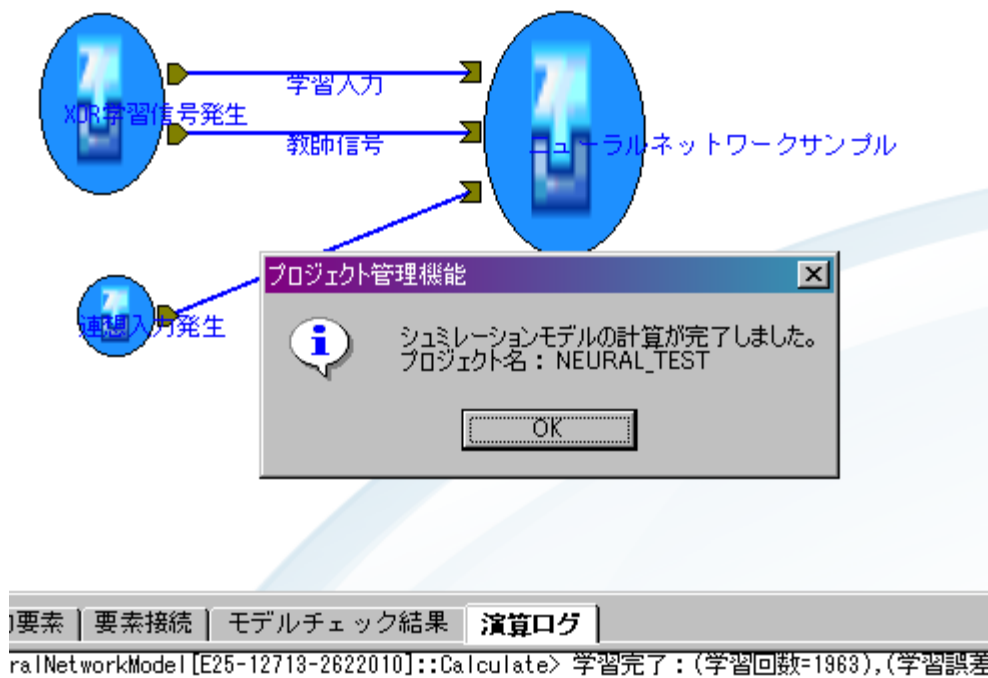


図 3. 3 学習モード設定

此处で演算を実行し、学習が完了すると、図3. 4 に示すように 計算完了メッセージが表示され、演算ログに 学習回数等の情報が表示されます。 学習完了時に プロパティで指定したファイル  
( ¥CommonMP¥CommonMPData¥SampleModelDevelop¥OutputData¥NNSample ) に学習した結果(各ニューロン間の結合値)が保存されます。



学習誤差表示を「表示ON」にしておく事で、図3. 5に示すように、学習状況を表示させることができます。

次に 図3. 6に示すように プロパティ画面の初期設定から 連想モード(1)に指定してください。この状態で演算を開始すると、図3. 7に示すように

入力パターン	→	連想結果
0.0, 0.0		0.026.... (粗 0 を連想)
1.0, 0.0		0.969.... (粗 1 を連想)
0.0, 1.0		0.969.... (粗 1 を連想)
1.0, 1.0		0.042.... (粗 0 を連想)

という結果が得られ、 XORの学習通りの 連想を行っていることがわかります。

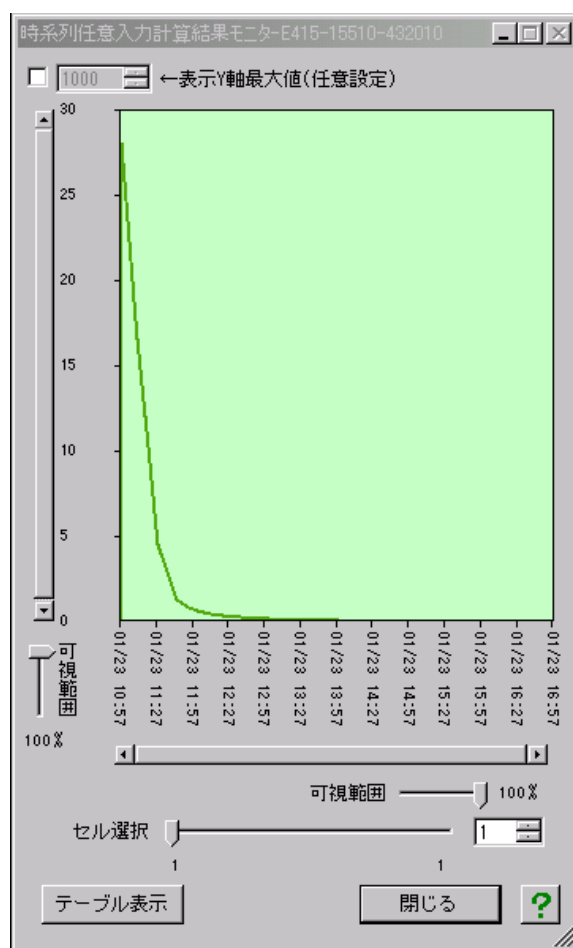


図 3. 5 学習曲線

セル型モデルプロパティ詳細設定

プロパティ設定 初期情報設定

設定値

項目	値
学習(=0)/連想モード(=1)	1
学習結果保存ファイル	OutputData\NNS...
学習パターン数(本サンプルでは...)	4
学習打ち切り判定誤差	0.001
シグモイド関数の傾き	3
学習係数	0.1
学習識別性係数	0.75

ファイル名称:

ファイル入力 ファイル出力 設定 キャンセル ?

図 3. 6 連想モード設定





図 3. 7 連想結果

既に、ご承知の通り、ニューラルネットワークは、学習回数を多くすれば、誤差が減少するというものではなく、ネットワーク構造(層の数、各層内のニューロン数等)や、各種係数の調整が必要です。これらの調整は、どのようなモデルにニューラルネットワークを使用するか等により、逐次行われるものであり、本サンプル内では触れません。

この様に、一見 水理水文とは関係のないモデルも工夫により、CommonMP上で動作させる事が可能です。しかし、CommonMPVer1.0では、それらのを容易に実現する事が難しく、適切な評価画面も制約が多くあります。今後、ユーザーからのご要求により、色々な局面でもCommonMPを使用して頂ける事を目指します。