

TR-O-0133

29

AVSを用いたニューラルネットワークの  
可視化シミュレーション

山賀 睦夫

1996. 3.15

ATR光電波通信研究所

# A V S を用いたニューラルネットワークの可視化シミュレーション

A T R 光電波通信研究所  
通信デバイス研究室

山賀睦夫

## はじめに

ニューラルネットワークは多数のニューロンが結合して構成されており、その状態を把握するのは簡単なことではない。しかし、ネットワークの状態の把握は新しいモデルの有効性を検証する場合や応用を考える場合、極めて重要である。そのための手段としては、

1. ある指標を設定し、その値を評価する
2. ネットワークの全体像を直観的に把握する

という2通りの方法が考えられる。計算機シミュレーションを主体とした研究を進める場合、前者の評価法は比較的簡単に実現できるが、後者は数字で表せる訳ではないので困難である。

そこで当所では、汎用の科学技術計算用可視化ツールである AVS を用いてネットワークの状態を可視化し、動作状態をリアルタイムに把握可能なシステムを構成して研究を効率的に進めてきた。本レポートでは、このシミュレーションシステムの概要について述べる。



# 目次

1	AVS とは	5
2	ニューラルネットワークモデル	7
2.1	相互結合型モデル	7
2.2	パターンの記憶と Pseudo-Inverse 法	7
2.3	カオスの発生	8
3	プログラムリスト	9
3.1	ニューラルネットワーク演算モジュール	9
3.1.1	定数、変数、関数名等の定義部	9
3.1.2	メインルーチン	11
3.1.3	変数の初期化	11
3.1.4	データ入出力用関数	12
3.1.5	初期パターンの設定	14
3.1.6	ニューロンの属性の初期化	14
3.1.7	ネットワークの状態の更新	15
3.1.8	ネットワークの状態のセット	16
3.1.9	AVS で扱うパラメータの定義	17
3.1.10	AVS の初期化	18
3.1.11	表示のクリア	18
3.1.12	コントロールパネルの読み取り	19
3.1.13	コントロールパネルの表示の更新	21
3.1.14	ネットワークの状態の描画	22
3.1.15	ネットワークの状態の評価と画面への表示	23
3.1.16	乱数の発生	25
3.2	グラフ表示モジュール	26
3.2.1	定義部	26
3.2.2	チョイスパネルの更新	28
3.2.3	保存データのクリア	28
3.2.4	データの登録	29
3.2.5	ファイルへの出力	30
3.2.6	グラフオブジェクトの生成	32
3.2.7	モジュール制御関数	38
3.2.8	モジュール記述関数	42
3.2.9	初期化関数	43
3.3	Makefile	44
4	モジュールの組み合わせによる AVS ネットワークの構成	45
4.1	モジュールの組み合わせ方	45
4.2	network ファイル	49
5	まとめ	55



## 第 1 章

### AVS とは

AVS(Application Visualization System) とは、Advanced Visual Systems Inc. が開発した科学技術計算用可視化ツールである。X-Window 上で動き、低価格のワークステーションからスーパーコンピュータまでの多くの機種で使うことができる。そのため、通常は手元のワークステーションで作業を行い、計算速度が要求される場合はスーパーコンピュータをレンタル使用する等の使い分けが可能である。

また、AVS はいくつかの‘モジュール’を組み合わせて目的の機能を実現する‘ネットワーク’を構成する様になっており(図 1.1)、様々な機能を持つモジュールがあらかじめ用意されている。したがって利用者は、ユーザインターフェースや表示等の手間のかかる部分は既存のモジュールを使用し、より本質的な計算部分のモジュールの作成に専念することができる。

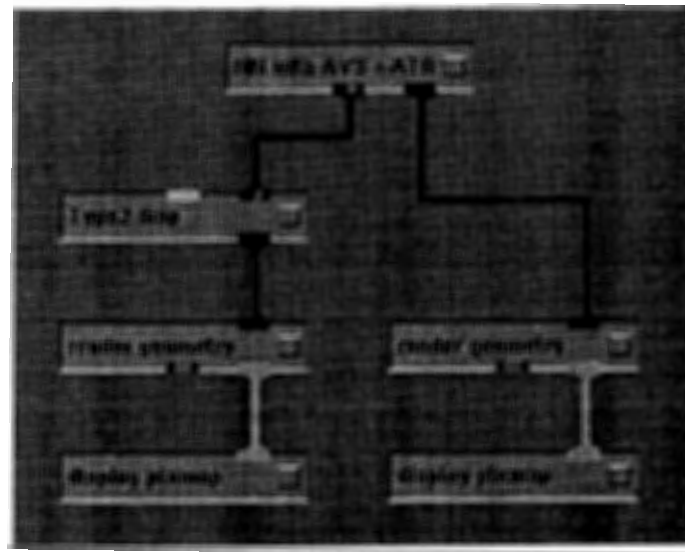


図 1.1: AVS のモジュールとネットワーク

本シミュレータでは、3次元空間上に並べた球の色をニューロンの状態によって変化させ、AVSにより表示させるという方法を取っている。我々の使用したモデルではニューロンを平面上に配置しているため、このように3次元空間上の球として扱うやり方は、必ずしも効率的な方法ではない。しかし、前項で述べた必要なモジュールのみを作成すれば良く、異なる計算機間での可搬性が良いこと、また、ニューロンの配置を3次的にした場合の対応の容易さ等を考えれば、AVSを使用するメリットは極めて大きいと言える。





## 第 2 章

### ニューラルネットワークモデル

本章では、当所で用いたニューラルネットワークモデルについて説明する。本レポートの目的はシミュレーションシステムの解説であるので、ここでの説明はそのために必要な最小限のものにとどめる。詳細が必要な場合は巻末の参考文献一覧を参照されたい。

#### 2.1 相互結合型モデル

ここでは、図 2.1に示すように各ニューロンの出力がある結合係数を介して全てのニューロンにフィードバックされる相互結合型のモデルを用いた。すなわち、ある時点でのニューロンの状態  $S(t+1)$  は、1 単位時間前のニューロンの状態  $S(t)$  と各ニューロン間の結合係数の集合である結合行列  $T$  によって決定される。(ニューロンの総数は  $N$  とする) (式 2.1)

$$S_i(t+1) = \text{sgn} \sum_{j=1}^N \varepsilon_{ij} T_{ij} S_j(t) \quad (2.1)$$

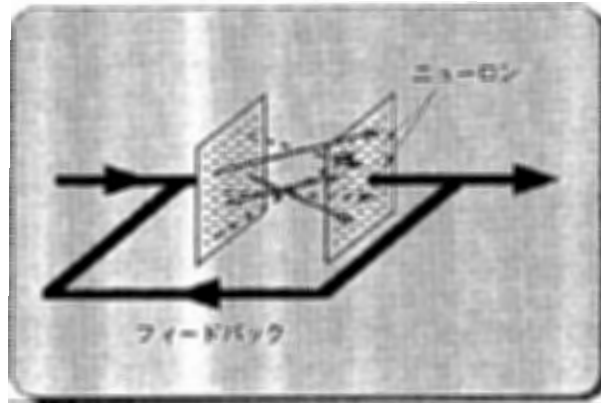


図 2.1: 相互結合型ニューラルネットワークモデル

#### 2.2 パターンの記憶と Pseudo-Inverse 法

このネットワークに図 2.2に示す 30 個の顔のパターンを記憶させた。ここでパターンを記憶させるには、ネットワークを動作させた時に記憶パターンが現れるように結合行列  $T$  を決めてやればよい。その際、これらのパターンを能率良く記憶させるため pseudo-inverse 法 [1] を用いて疑似的に直交化してある。(式 2.2)

$$T = \sum_{\mu=1}^K \sum_{\nu=1}^L \xi^{\mu, \nu+1} \otimes \xi^{\mu, \nu \dagger} \quad (2.2)$$



図 2.2: 記憶させた 30 個の顔のパターン

### 2.3 カオスの発生

前項の方法でパターンを記憶させた場合、ネットワークは極めて安定なダイナミクスを示す。すなわち、図 2.2 の横に並んだパターンが次々にあらわれる。

各ニューロンは他の全てのニューロンからの出力を入力として受け取って、次の時点での状態を決定している。この時ニューロンへの入力数を減らしていくとカオスが発生する。このカオスの発生の様子を可視化して研究の効率を上げるのが、このシミュレータ開発の目的である。

## 第 3 章

### プログラムリスト

本章では、

1. ATR で作成したニューラルネットワーク演算モジュール
2. 外注によりクボタコンピュータが作成したグラフ表示モジュール

のソースリストを示す。

AVS は C および FORTRAN に対応したインターフェースを持っているが、ここでは、C を用いてモジュールの作成を行なっている。

#### 3.1 ニューラルネットワーク演算モジュール

##### 3.1.1 定数、変数、関数名等の定義部

```
/*                      Chaotic Neural Network simulator                      */
/* --- ATR Optical and Radio Communications Research Laboratories --- */

#define NX 20                /* 横方向のニューロンの数 */
#define NY 20                /* 縦方向のニューロンの数 */
#define BALL NX*NY          /* ニューロンの総数 */
#define CYCLE 5              /* 記憶シーケンスの数 */
#define PATTERN 6           /* 1 シーケンスの中のパターンの数 */

#define TRUE 1
#define FALSE 0

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <avs/avs.h>         /* AVS 関連 */
#include <avs/geom.h>        /* AVS 関連 */
#include <avs/field.h>       /* AVS 関連 */
#include <time.h>            /* ファイル名に日付、時間を入れるため */
#include <sys/types.h>       /* ファイル名に日付、時間を入れるため */
#include <sys/time.h>        /* ファイル名に日付、時間を入れるため */

typedef float FLOAT3[3];    /* AVS に渡すための変数型定義 */

void init_param(void);
void read_weight(void);
void define_output(void);
```

```

void define_init(void);
void clear_lattice_param(void);
void take_next_step(int step);
void set_pattern(void);
void inits_AVS(int argc, char **argv);
void read_cntl_panel(void);
void update_cntl_panel(void);
void draw_lattice(void);
void eval(void);
float ran(int *);

int first_time; int last_time; int current_time; int reset;
int step_param; int next_param; int cont_param; int restart_param;
int chaos_param; int cycle; int pattern; int face_set; int inverse;
int set; int invset; int a_set; int a_invset; int iradii; int run;
int r; int xmin; int xmax; int ymin; int ymax;

char *datafile;
FLOAT3 vert[BALL], color[BALL];          /* ニューロンの空間位置、色 */
float radii[BALL];                       /* ニューロンの大きさ */
int read_panel, iend, at_first, at_reset, iset;
int rtime;
float w[BALL][BALL], c[BALL][BALL], x[BALL], face[CYCLE][PATTERN][BALL];
/* w: 結合行列、c: ニューロン間結合を切る際の優先順位、
   x: '現在の' ネットワークの状態、face: 記憶パターン */
int mr0;
int i;
int near_cycle, near_pattern;
float evaluation, face_mag[CYCLE][PATTERN];

GEOMedit_list editlist = NULL;
AVSfield_float *output = NULL;
int dims[1];

```

### 3.1.2 メインルーチン

基本的な構造は初期化した後、

1. ユーザからの指示の読み取り
2. 演算及び表示

の繰返しとなっている。

```
void main(int argc, char **argv) {
    inits_AVS(argc,argv);
    init_param();
    while(iend == 0) {
        do read_cntl_panel(); while(read_panel == 1);
        if (iend == 0){
            if (at_first == 1 || at_reset == 1) {
                if (at_first == 1) clear_lattice_param();
                clear_graph();
                draw_lattice();
                at_first = 0;
                at_reset = 0;
            }
            else {
                if (iset != 0) {
                    set_pattern();
                    draw_lattice();
                }
                else {
                    if (rtime + run <= last_time) take_next_step(run);
                    if ( rtime >= first_time) draw_lattice();
                }
            }
            update_cntl_panel();
        }
        read_panel = 1;
    }
}
```

### 3.1.3 変数の初期化

この関数ではグローバル変数の初期化を行なう。プログラムの起動時に1回だけ呼ばれる。

```
void init_param() {
    mr0 = 1073741824.0;
    rtime = 0;
    iend = 0;
    read_panel = 1;
    at_first = 1;
    iset = 0;
    define_output();
    read_weight();
}
```

### 3.1.4 データ入出力用関数

read\_weight() は、結合行列、結合の優先順位、記憶パターンの読み込みを行なう。

define\_output() は、計算結果のファイルへの出力を行なう。

define\_output() は、計算結果を出力するファイル名を定義する関数。ここでは、実行時刻を反映したファイル名を生成するようになっている。

```
void read_weight() {
    FILE *fp;
    char *filepath;
    char *mode;
    float buff;
    int i,j,k;
    filepath = "data/weight.data";      /* 結合行列の読み込み */
    mode = "r";
    fp=fopen(filepath,mode);
    for(j=1;j<=BALL;j++) {
        for(i=1;i<=BALL;i++) {
            (void)fscanf(fp,"%e",&buff);
            w[i-1][j-1] = buff;
        }
    }
    fclose(fp);

    filepath = "data/connect.data";     /* 結合の優先順位の読み込み */
    mode = "r";
    fp=fopen(filepath,mode);
    for(i=1;i<=BALL;i++) {
        for(j=1;j<=BALL;j++) {
            (void)fscanf(fp,"%e",&buff);
            c[i-1][j-1] = buff;
        }
    }
    fclose(fp);

    filepath = "data/face.data";        /* 記憶パターンの読み込み */
    mode = "r";                        /* 絶対値の計算も同時に... */
    fp=fopen(filepath,mode);
    for(i=1;i<=CYCLE;i++) {
        for(j=1;j<=PATTERN;j++) {
            face_mag[i-1][j-1] = 0;
            for(k=1;k<=BALL;k++) {
                (void)fscanf(fp,"%e",&buff);
                face[i-1][j-1][k-1] = buff;
                face_mag[i-1][j-1] += buff * buff;
            }
            face_mag[i-1][j-1] = (float)sqrt((double)face_mag[i-1][j-1]);
        }
    }
    fclose(fp);
}
```

```
void define_output() {
    float buff;
    struct tm *times;
    time_t tloc,tres;

    tres = time(&tloc);
    times = localtime(&tres);          /* 現在時刻の取得 */
    datafile = malloc(31);
    (void)strftime(datafile,26,"result/NN_%y-%m-%d_%H:%M",times);
}
```

### 3.1.5 初期パターンの設定

ネットワークの初期条件を乱数により決定するための関数。

```
void define_init() {
    int i;
    float r;

    for (i=1; i<=BALL; i++) {
        r = ran(&mr0);
        if (r < (float).5) x[i-1] = (float)-1.;
        else x[i-1] = (float)1.;
    }
}
```

### 3.1.6 ニューロンの属性の初期化

AVSでの表示に使用する変数、vert[ニューロンの数][3] (各ニューロンの3次元空間における位置座標)、radii[ニューロンの数] (各ニューロンの大きさ)、color[ニューロンの数][3] (各ニューロンの色、RGBで表現したもの)の初期化

```
void clear_lattice_param() {
    int i, j;

    define_init();
    for (i=1; i<=NX; i++) {
        for (j=1; j<=NY; j++) { /* ニューロンの位置の設定 */
            vert[(i-1)+(j-1)*NX][0] = (float) i;
            vert[(i-1)+(j-1)*NX][1] = (float) (NY + 1 - j);
            vert[(i-1)+(j-1)*NX][2] = (float)0.;
        }
    }
    for (i=1; i<=BALL; i++) { /* ニューロンの大きさ、色の初期設定 */
        radii[i-1] = (float) iradii / (float)10.;

        color[i-1][0] = ( x[i-1] + (float)1. ) / (float)2.;
        color[i-1][1] = color[i-1][0];
        color[i-1][2] = (float)1.;
    }
}
```



## 3.1.7 ネットワークの状態の更新

あらかじめ定義された規則に従ってネットワークの状態を更新するための関数。iirun ステップの状態更新を行う。各ニューロンの入力の数( $r$ )は、グローバル変数を使用。

```
void take_next_step(int iirun) {
    int jend, i, j, k;
    double y[BALL];
    int ii=1;

    while (ii <= iirun) {
        for (i=1; i<=BALL; i++) {
            y[i-1] = (double)0.;
            switch(chaos_param) {          /* 変数 chaos_param の値によって */
                case 1: jend=r;    break; /* ニューロンの入力の数を決 定 */
                default: jend=BALL; break;
            }
            for (j=1; j<=jend; j++) {      /* 結合行列とニューロンの状態の積 */
                k = c[i-1][j-1];
                y[i-1] += (double) w[i-1][k-1] * (double) x[k-1];
            }
        }
        for (i=1; i<=BALL; i++) {
            switch(y[i-1]>=(double)0.) { /* 内部状態を 1/-1 に変換 */
                case TRUE: x[i-1] = (float)1. ; break;
                default : x[i-1] = (float)-1.; break;
            }
            /* 新しい状態にしたがってニューロンの色を設定 */
            color[i-1][0] = ( x[i-1]+(float)1. ) / (float)2.;
            color[i-1][1] = color[i-1][0];
            radii[i-1] = (float) iradii / (float)10.;
        }
        ii++;
    }

    rtime += iirun;
}
```

### 3.1.8 ネットワークの状態のセット

ニューラルネットワークの状態を強制的にあるパターンにセットするための関数。グローバル変数 `iset` の値により、

1. 記憶パターンのいずれかをセットする
2. ある範囲のニューロンの状態を強制的に 1 にする
3. ある範囲のニューロンの状態を強制的に -1 にする

等の操作を行なう。

```
void set_pattern(void) {
    int i, j;
    float y[BALL];

    switch(iset) {
        case -3:                                /* パターンの反転 */
            for (i=1; i<=BALL; i++) x[i-1]=-x[i-1];
            break;

        case 3:                                /* 記憶パターンの1つにセット */
            for (i=1; i<=BALL; i++) x[i-1]=face[cycle-1][pattern-1][i-1];
            break;

        case 2: case -2:                       /* ある範囲の外側を 1/-1 にセット */
            for (i=1; i<=BALL; i++) {
                y[i-1] = x[i-1];
                x[i-1] = (float) (iset / 2);
            }
            for (i=xmin; i<=xmax; i++) {
                for (j=ymin; j<=ymax; j++) x[(i-1)+(j-1)*NX] = y[(i-1)+(j-1)*NX];
            }
            break;

        case 1: case -1:                       /* ある範囲内を 1/-1 にセット */
            for (i=xmin; i<=xmax; i++) {
                for (j=ymin; j<=ymax; j++) x[(i-1)+(j-1)*NX] = (float) (iset);
            }
            break;

        default:
            break;
    }

    for (i=1; i<=BALL; i++) {
        color[i-1][0] = ( x[i-1]+(float)1. ) / (float)2.;
        color[i-1][1] = color[i-1][0];
        radii[i-1] = (float) iradii / (float)10.;
    }
    iset = 0;
}
```

## 3.1.9 AVS で扱うパラメータの定義

出力ポートとしては、

1. ニューラルネットワークの状態を表示するためのポート
2. グラフ表示用のポート

の2つを定義している。

```
neural_net()
{
    int parm;

    AVSset_module_name("NN with AVS < ATR >", MODULE_DATA);
    AVScreate_output_port("Output Geometry", "geom");          /* ネットワーク表示用 */
    AVScreate_output_port("output",                             /* グラフ表示用          */
"field 1D 0-space 3-vector float irregular");
    parm = AVSadd_parameter("first time", "integer", 1, 1, 300000);
    AVSconnect_widget(parm, "typein_integer");
    parm = AVSadd_parameter("last time", "integer", 100000, 1, 300000);
    AVSconnect_widget(parm, "typein_integer");
    parm = AVSadd_parameter("current time", "integer", 0, 0, 300000);
    AVSconnect_widget(parm, "typein_integer");
    parm = AVSadd_parameter("Reset", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("step 1", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("next", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("continue", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("restart", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("chaos", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("CYCLE", "integer", 1, 1, CYCLE);
    AVSconnect_widget(parm, "typein_integer");
    parm = AVSadd_parameter("PATTERN", "integer", 1, 1, PATTERN);
    AVSconnect_widget(parm, "typein_integer");
    parm = AVSadd_parameter("set", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("inverse", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("set to 1", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("set to -1", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("set rest to 1", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("set rest to -1", "boolean", 0, 0, 1);
    parm = AVSadd_parameter("iradii", "integer", 4, 0, 10);
    parm = AVSadd_parameter("run", "integer", 1, 1, 30);
    parm = AVSadd_parameter("conectivity", "integer", 40, 1, 400);
    parm = AVSadd_parameter("x min", "integer", 1, 1, NX);
    parm = AVSadd_parameter("x max", "integer", NX, 1, NX);
    parm = AVSadd_parameter("y min", "integer", 5, 1, NY);
    parm = AVSadd_parameter("y max", "integer", 13, 1, NY);
}
```

### 3.1.10 AVSの初期化

```
void inits_AVS(int argc,char **argv)
{
    AVScorout_init(argc,argv,neural_net);
}
```

### 3.1.11 表示のクリア

```
void clear_graph() {
    GEOMedit_list dummylist = NULL;

    if(output) AVSfield_free(output);
    dims[0] = 1;
    output = (AVSfield_float *)
        AVSdata_alloc("field 1D 0-space 3-vector float irregular",dims);
    output->data[0] = -1;
    output->data[1] = -1;
    output->data[2] = -1;
    AVSmark_output_unchanged("Output Geometry");
    AVScorout_output(dummylist,output);
}
```

## 3.1.12 コントロールパネルの読み取り

```

void read_cntl_panel() {
    AVScorout_input(&first_time, &last_time, &current_time, &reset, &step_param,
        &next_param, &cont_param, &restart_param, &chaos_param,
        &cycle, &pattern, &face_set, &inverse, &set, &invset, &a_set,
        &a_invset, &iradii, &run, &r, &xmin, &xmax, &ymin, &ymax);

    if (at_first == 0      && cont_param == 0 && next_param == 0
        && step_param == 0 && set == 0      && invset == 0
        && a_set == 0      && a_invset == 0  && reset==0
        && face_set ==0    && inverse == 0  && restart_param == 0) {
        read_panel = 1;
        return;
    }

    if (restart_param == 1 || reset == 1) {
        if (restart_param ==1) {
            restart_param = 0;
            at_first = 1;
            AVSmodify_parameter("restart", AVS_VALUE, restart_param, 0, 1);
        }
        if (reset == 1) {
            reset = 0;
            at_reset = 1;
            AVSmodify_parameter("Reset", AVS_VALUE, reset, 0, 1);
        }
        rtime = 0;
        AVSmodify_parameter("current time", AVS_VALUE, rtime, 0, 1);
        if (step_param == 1) {
            step_param = 0;
            AVSmodify_parameter("step 1", AVS_VALUE, step_param, 0, 1);
        }
        if (next_param == 1) {
            next_param = 0;
            AVSmodify_parameter("next", AVS_VALUE, next_param, 0, 1);
        }
        if (cont_param == 1) {
            cont_param = 0;
            AVSmodify_parameter("continue", AVS_VALUE, cont_param, 0, 1);
        }
        read_panel = 0;
        return;
    }

    if (step_param == 1) {
        run = 1;
        step_param = 0;
        AVSmodify_parameter("step 1", AVS_VALUE, step_param, 0, 1);
    }
    if (rtime + run > last_time) {
        if (step_param == 1) {

```

```
    step_param = 0;
    AVSmodify_parameter("step 1", AVS_VALUE, step_param, 0, 1);
}
if (next_param == 1) {
    next_param = 0;
    AVSmodify_parameter("next", AVS_VALUE, next_param, 0, 1);
}
if (cont_param == 1) {
    cont_param = 0;
    AVSmodify_parameter("continue", AVS_VALUE, cont_param, 0, 1);
}
read_panel = 1;
}
if (a_invset == 1) {
    iset = -2;
    a_invset = 0;
    AVSmodify_parameter("set rest to -1", AVS_VALUE, a_invset, 0, 1);
}
if (invset == 1) {
    iset = -1;
    invset = 0;
    AVSmodify_parameter("set to -1", AVS_VALUE, invset, 0, 1);
}
if (a_set == 1) {
    iset = 2;
    a_set = 0;
    AVSmodify_parameter("set rest to 1", AVS_VALUE, a_set, 0, 1);
}
if (set == 1) {
    iset = 1;
    set = 0;
    AVSmodify_parameter("set to 1", AVS_VALUE, set, 0, 1);
}
if (face_set == 1) {
    iset = 3;
    face_set = 0;
    AVSmodify_parameter("set", AVS_VALUE, face_set, 0, 1);
}
if (inverse == 1) {
    iset = -3;
    inverse = 0;
    AVSmodify_parameter("inverse", AVS_VALUE, inverse, 0, 1);
}
read_panel = 0;
}
```

### 3.1.13 コントロールパネルの表示の更新

```
void update_cntl_panel() {  
    if (next_param == 1) next_param = 0;  
    AVSmodify_parameter("next", AVS_VALUE, next_param, 0, 1);  
    AVSmodify_parameter("current time", AVS_VALUE, rtime, 1, 300000);  
}
```





## 3.1.15 ネットワークの状態の評価と画面への表示

```

void eval() {
    GEOMobj *obj_title=NULL, *obj_text=NULL;
    int i,j,k;
    FILE *dp;
    float product, mag;
    int text_label;
    float ref_point[3], offset[3], height, text_color[3];
    char *string;
    float max_product;
    int j_tmp;

    mag=0;
    for (k=1; k<=BALL; k++) mag += x[k-1] * x[k-1];
    mag = (float)sqrt((double)mag);
    near_cycle = 1;
    near_pattern = 1;
    evaluation = 0.0;

    if(output) AVSfield_free(output);
    dims[0]=CYCLE;
    output = (AVSfield_float *)
        AVSdata_alloc("field 1D 0-space 3-vector float irregular", dims);

    for (i=1; i<=CYCLE; i++) {
        max_product=0.0;
        for (j=1; j<=PATTERN; j++) {
            product=0.0;
            for (k=1; k<=BALL; k++) product += face[i-1][j-1][k-1] * x[k-1];
            product /= face_mag[i-1][j-1] * mag ;
            if ( fabs((double)product) > fabs((double)max_product) ) {
                max_product = product;
                j_tmp = j;
            }
        }
        if ( fabs((double)max_product) > fabs((double)evaluation) ) {
            evaluation = max_product;
            near_cycle = i;
            near_pattern = j_tmp;
        }
        output->data[(i-1) * 3 + 0] = i;
        output->data[(i-1) * 3 + 1] = rtime;
        output->data[(i-1) * 3 + 2] = max_product;
    }

    dp=fopen(datafile,"a+");
    fprintf(dp,"%5d%5d%10.5f", near_cycle, near_pattern, evaluation);
    if (chaos_param==1) {fprintf(dp,"%6d%10d\n",r,rtime);}
    else {fprintf(dp,"%6d%10d\n",BALL,rtime);}
    fclose(dp);
}
/* 現在の状態と記憶パターンとの内積の計算 */
/* および一番近い記憶パターンの決定 */
/* グラフ出力用変数の設定 */
/* 途中経過のファイル出力 */

```

```

string=malloc(32);

/* 位置のレファレンスを初期化 */
text_color[0] = 1.0; text_color[1] = 1.0; text_color[2] = 1.0;
ref_point[0] = (float) 0; ref_point[1] = (float) 0;
ref_point[2] = (float) 0;
offset[0] = (float) 0; offset[1] = (float) 0; offset[2] = (float) 0;

if(obj_title) GEOMdestroy_obj(obj_title);
text_label = GEOMcreate_label_flags(2, 0, 0, 1, GEOM_LABEL_LEFT, 0);
obj_title = GEOMcreate_label(GEOM_NULL, text_label); /* タイトル部の出力用オブジェクト */
ref_point[0] = (float)NX/4; ref_point[1] = (float)NY+10;
height = (float) 0.2;
GEOMadd_label(obj_title, "Chaotic Neural Network", /* タイトル設定 */
    ref_point, offset, height, text_color, text_label);
GEOMedit_geometry(editlist, "lattice", obj_title); /* editlist への追加 */

if(obj_text) GEOMdestroy_obj(obj_text);
text_label = GEOMcreate_label_flags(2, 0, 0, 0, GEOM_LABEL_LEFT, 0);
obj_text = GEOMcreate_label(GEOM_NULL, text_label); /* テキスト表示用オブジェクト */
ref_point[0] = (float)(NX+6); ref_point[1] = (float)(NY-3);
height = (float) 0.10;
sprintf(string, "Pattern      : (%d-%d)\n", near_cycle, near_pattern);
GEOMadd_label(obj_text, string, /* テキスト設定 Pattern...*/
    ref_point, offset, height, text_color, text_label);
ref_point[0] = (float)(NX+6); ref_point[1] = (float)(NY-6);
height = (float) 0.10;
sprintf(string, "Evaluation   : %6.3f", evaluation);
GEOMadd_label(obj_text, string, /* テキスト設定 Evaluation...*/
    ref_point, offset, height, text_color, text_label);
ref_point[0] = (float)(NX+6); ref_point[1] = (float)(NY-9);
height = (float) 0.10;
if (chaos_param==1) {sprintf(string, "Connectivity : %3d", r);}
else {sprintf(string, "Connectivity : %3d", BALL);}
GEOMadd_label(obj_text, string, /* テキスト設定 Connectivity...*/
    ref_point, offset, height, text_color, text_label);
ref_point[0] = (float)(NX+6); ref_point[1] = (float)(NY-14);
height = (float) 0.10;
sprintf(string, "    Step      : %2d", run);
GEOMadd_label(obj_text, string, /* テキスト設定 Step...*/
    ref_point, offset, height, text_color, text_label);
ref_point[0] = (float)(NX+6); ref_point[1] = (float)(NY-17);
height = (float) 0.10;
sprintf(string, "    Count     : %6d", rtime);
GEOMadd_label(obj_text, string, /* テキスト設定 Count...*/
    ref_point, offset, height, text_color, text_label);
GEOMedit_geometry(editlist, "lattice", obj_text); /* editlist への追加 */
free(string);
}

```

### 3.1.16 乱数の発生

```
float ran(int *i) {  
    float ret_val;  
  
    *i = *i * 843314861 + 453816693;  
    if (*i < 0) *i += 2147483648;  
    ret_val = (float)*i / 2147483648.;  
    return ret_val;  
}
```

### 3.2 グラフ表示モジュール

このモジュールは外注により作成したものである。

演算モジュールよりデータを受け取り、グラフを表示する。規定のフォーマットでデータを出力するモジュールを作成すれば、他のプログラムからの利用も可能である。

#### 3.2.1 定義部

```
#if !defined(ignore_offset)
#if defined(mips)
#define ignore_offset
#endif
#endif

/*****
*
* 変更履歴
*   1994/07/08 : ラベルオフセットが有効でないレンダラー対応
*   1994/07/08 : デフォルトカラーマップの組み込み
*   1994/07/11 : グラフの縦軸（物理量）の負方向対応
*
*****/

#include <stdio.h>
#include <math.h>
#include <string.h>
#include <avs/avs.h>
#include <avs/geom.h>
#include <avs/colormap.h>
#include <avs/field.h>

#define REALLOC(p,s) (p?realloc(p,s):malloc(s))
#define MALLOC(s) malloc(s)

/*
* ===== 入力データ保存領域 =====
*/
static struct _store {
    int display;      /* 表示フラグ（大きい値順にソート = 出力順） */
    int line_no;      /* 線番号 */
    int nmember;      /* データ個数 */
    int idx;          /* ワーク */
    struct _member {
        float time; /* 時刻 */
        float value; /* 物理量 */
    } *member;
} *store = NULL;

static int nstore = 0; /* データ個数 */
static float max_time = 1.0; /* 最大時刻 */
static float min_time = 0.0; /* 最小時刻 */
static float max_value = 1.0; /* 最大物理量 */
```

```
static float min_value = 0.0; /* 最小物理量 */
static int nstep = 0;        /* ステップ数 */

/*
 *----- store 比較関数 -----
 */
int cmp_store(a, b)
    struct _store *a, *b;
{
    return(b->display - a->display);
}
```

### 3.2.2 チョイスパネルの更新

```

/*
 *----- チョイスパネルの更新 -----
 * 線番号選択ボタンの更新を行なう
 */
void modify_choice()
{
    int i;
    char *lines;

    qsort(store, nstore, sizeof(struct _store), cmp_store);
    lines = malloc((nstore + 1) * 10);
    strcpy(lines, "Line");
    for(i = 0; i < nstore; i++){
        if(store[i].display > 0) store[i].display ++;
        sprintf(lines + strlen(lines), "%c%3d",
            ((store[i].display > 0)?' ':'X'),
            store[i].line_no);
    }
    AVSmodify_parameter("lines", AVS_VALUE|AVS_MINVAL, "Line", lines, "#");

    free(lines);
}

```

### 3.2.3 保存データのクリア

```

/*
 *----- 全データの解放 -----
 * 保存データのクリア
 */
void data_free()
{
    int i;

    if(store != NULL){
        for(i = 0; i < nstore; i++) free(store[i].member);
        free(store);
        store = NULL;
    }
    nstore = 0;
    max_time = max_value = 1.0;
    min_time = min_value = 0.0;
    nstep = 0;
    modify_choice();

    return;
}

```

## 3.2.4 データの登録

```

/*
 * ----- データの登録 -----
 * 引数で指定されたデータを、保存します。
 *
 * 新しい線番号が入力された時は、1 を返します。
 */
int add_data(line_no, time, value)
    int line_no; /* 線番号 */
    float time; /* 時間 */
    float value; /* 物理量 */
{
    int i;

    nstep++;
    for(i = 0; i < nstore; i++){
        if(store[i].line_no == line_no) break;
    }
    while(time > max_time) max_time *= 2;
    if(time < min_time) min_time = time;
    if(value > max_value) max_value = value;
    if(value < min_value) min_value = value;

    if(i < nstore){
        /* 既に同じ線番号のデータが既存するとき */
        store[i].nmember ++;
        store[i].member = (struct _member *)REALLOC
            (store[i].member,
             sizeof(struct _member) * store[i].nmember);
        store[i].member[store[i].nmember - 1].time = time;
        store[i].member[store[i].nmember - 1].value = value;
        return(0);
    }
    else{
        /* 新たな線番号 */
        store = (struct _store *)REALLOC
            (store, sizeof(struct _store) * (nstore + 1));

        store[nstore].display = 1;
        store[nstore].line_no = line_no;
        store[nstore].nmember = 1;
        store[nstore].member = (struct _member *)MALLOC
            (sizeof(struct _member) * 1);
        store[nstore].member[0].time = time;
        store[nstore].member[0].value = value;
        nstore ++;
        return(1);
    }
}

```

## 3.2.5 ファイルへの出力

```

/*
 * ----- ファイルへの出力 -----
 */
void output_file(fp, start, end)
    FILE *fp;          /* 出力ファイル */
    int start, end; /* 開始/終了時刻 */
{
    int i, j, k, l;
    float itime;

    int nvalues;
    float *values;

    values = MALLOC(sizeof(float) * nstore);

    /*
     * ヘッダの書き込み と index の初期化
     */
    fprintf(fp, "time");
    for(i = 0; i < nstore; i++){
        if(store[i].display <= 0) continue;
        fprintf(fp, "\tline-%d", store[i].line_no);
        store[i].idx = 0;
    }
    fprintf(fp, "\n");

    itime = start;
    while(itime <= end){
        nvalues = 0;
        /*
         * itime に近い物理量の検索
         */
        for(i = 0; i < nstore; i++){
            if(store[i].display <= 0) continue;
            for(j = store[i].idx; j < store[i].nmember; j++){
                if(store[i].member[j].time >= itime) break;
            }
            store[i].idx = j;
            if(j < store[i].nmember){
                values[nvalues] = store[i].member[j].time;
                nvalues++;
            }
        }
        if(nvalues == 0) break;

        /*
         * 一番近い物理量をチェック
         */
        itime = values[0];
        for(i = 1; i < nvalues; i++){

```



```
        if(itime > values[i]) itime = values[i];
    }
    if(itime > end) break;

    /*
    * 行の出力
    */
    fprintf(fp, "%f", itime);
    for(i = 0; i < nstore; i++){
        if(store[i].display <= 0) continue;
        if(store[i].idx < store[i].nmember){
            if(store[i].member[store[i].idx].time == itime){
                fprintf(fp, "\t%f", store[i].member[store[i].idx].value);
                store[i].idx ++;
            }
            else{
                fprintf(fp, "\t");
            }
        }
        else{
            fprintf(fp, "\t");
        }
    }
    fprintf(fp, "\n");
}
free(values);
}
```

## 3.2.6 グラフオブジェクトの生成

```

/*
 * ----- グラフオブジェクトの生成 -----
 * 引数で指定された諸元で、グラフを出力します。
 * fp が NULL でない場合は、fp が示すファイルにテキストデータを出力します。
 */

#define WIDTH 2.0 /* 表示ジオメトリの幅 */
#define HEIGHT 1.0 /* 表示ジオメトリの高さ */

void create_graph(geom, label,
#if defined(ignore_offset)
    offset,
#endif
    cmap, area, minimum, maximum, fp)
    GEOMedit_list geom; /* 出力ポート */
    float label; /* ラベルサイズ */
#if defined(ignore_offset)
    float offset; /* ラベルサイズ */
#endif
    AVScolormap *cmap; /* カラーマップ */
    char *area; /* 表示範囲 */
    float minimum, maximum; /* 表示範囲 */
    FILE *fp; /* ファイルポインタ */
{
    float *colors;
    float *varts;
    GEOMobj *obj1, *obj2;
    int i, j, k;
    float r, g, b;
    float time_scale, time_offset;
    float value_scale, value_offset;

    static float def_hue[7] = {
        0.666667, 0.000000, 0.833333, 0.333333,
        0.500000, 0.166667, 0.500000 };
    static float def_saturation[7] = {
        1.000000, 1.000000, 1.000000, 1.000000,
        1.000000, 1.000000, 0.003922 };
    static float def_value[7] = {
        1.000000, 1.000000, 1.000000, 1.000000,
        1.000000, 1.000000, 1.000000 };
    static float def_alpha[7];

    static AVScolormap def_cmap = {
        7, 0.0, 0.0,
        def_hue, def_saturation, def_value, def_alpha };

    if(cmap == NULL) cmap = &def_cmap;
    /*
     * 表示範囲の設定

```

```

    */
    if(!strcmp(area, "none")){
        maxmum = max_time;
        minimum = min_time;
    }
    else if(!strcmp(area, "min")){
        maxmum = max_time;
    }
    else if(!strcmp(area, "max")){
        minimum = min_time;
    }
}

if(minimum >= maxmum){
    AVSwarning("illegal parameter minimum or maxmum");
    AVSmodify_parameter("area", AVS_VALUE, "none", 0, 0);
    return;
}

time_scale = WIDTH / (maxmum - minimum);
time_offset = -minimum;

if(min_value >= max_value){
    value_scale = 1.0;
    value_offset = 0.0;
}
else{
    value_scale = HEIGHT / (max_value - min_value);
    value_offset = -min_value;
}

/*
 * ファイルへの出力
 */
if(fp != NULL){
    output_file(fp, (int)minimum, (int)maxmum);
}

/*
 * 処理領域の確保
 */
colors = (float *)MALLOC(sizeof(float) * (nstep + 2) * 3);
varts = (float *)MALLOC(sizeof(float) * (nstep + 2) * 3);

/*
 * グラフオブジェクトの作成
 */
obj1 = GEOMcreate_obj (GEOM_POLYTRI, GEOM_NULL);
for(i = 0; i < nstore; i++){

    /* 表示対象でない線は飛ばす */
    if(store[i].display <= 0) continue;

```

```

/* 線色の割り当て */
if(cmap == NULL || cmap->size < store[i].line_no){
    /* カラーマップが無い/カラーマップが足りない時は白 */
    r = g = b = 1.0;
}
else{
    /* カラーマップの HSV を RGB に変更 */
    FILTERhsv_to_rgb (&r, &g, &b,
                      cmap->hue[store[i].line_no - 1],
                      cmap->saturation[store[i].line_no - 1],
                      cmap->value[store[i].line_no - 1]);
}

k = 0;
for(j = 0; j < store[i].nmember; j++){
    /* 時刻が表示範囲外のデータは無視する */
    if(store[i].member[j].time < minimum) continue;
    if(store[i].member[j].time > maxmum) continue;

    colors[k * 3 + 0] = r;
    colors[k * 3 + 1] = g;
    colors[k * 3 + 2] = b;
    varts[k * 3 + 0] = store[i].member[j].time;
    varts[k * 3 + 1] = store[i].member[j].value;
    varts[k * 3 + 2] = 0.0;
    k++;
}

if(k > 1){
    /* 二点以上の場合はオブジェクト作成 */
    for(j = 0; j < k; j++){
        varts[j * 3 + 0]
            = (varts[j * 3 + 0] + time_offset) * time_scale;
        varts[j * 3 + 1]
            = (varts[j * 3 + 1] + value_offset) * value_scale;
    }
    GEOMadd_polyline(obj1, varts, colors, k, GEOM_COPY_DATA);
}

{
    int flagx, flagy;
    char work[256];
    float y0;
    y0 = (0.0 + value_offset) * value_scale;

    /* 座標軸の生成 */
    colors[0] = colors[1] = colors[2]
        = colors[3] = colors[4] = colors[5] = 1.0;
    varts[0] = 0.0;
    varts[1] = y0;

```

```

    varts[2] = 0.0;
    varts[3] = WIDTH;
    varts[4] = y0;
    varts[5] = 0.0;
    GEOMadd_disjoint_line(obj1, varts, colors, 2, GEOM_COPY_DATA);
    varts[0] = 0.0;
    varts[1] = 0.0;
    varts[2] = 0.0;
    varts[3] = 0.0;
    varts[4] = HEIGHT;
    varts[5] = 0.0;
    GEOMadd_disjoint_line(obj1, varts, colors, 2, GEOM_COPY_DATA);

    /* 座標メモリの生成 */
    flagx = GEOMcreate_label_flags(1, 0, 1, 0, GEOM_LABEL_CENTER, 1);
    flagy = GEOMcreate_label_flags(1, 0, 1, 0, GEOM_LABEL_RIGHT, 1);
    obj2 = GEOMcreate_obj(GEOM_LABEL, GEOM_NULL);

    colors[0] = colors[1] = colors[2]
        = colors[3] = colors[4] = colors[5] = 1.0;

    for(i = 0; i <= 10; i++){

        /* y 軸メモリ */
        varts[0] = 0.0;
        varts[1] = i * (HEIGHT / 10.0);
        varts[2] = 0.0;
        varts[3] = 0.01;
        varts[4] = i * (HEIGHT / 10.0);
        varts[5] = 0.0;
        GEOMadd_disjoint_line(obj1, varts, colors, 2, GEOM_COPY_DATA);

        if(!(i % 2)){
            #if defined(ignore_offset)
                varts[0] = -0.01;
                varts[1] = i * (HEIGHT / 10.0) - offset / 2.0;
                varts[2] = 0.0;
                varts[3] = 0.0;
                varts[4] = 0.0;
                varts[5] = 0.0;
            #else
                varts[0] = -0.01;
                varts[1] = i * (HEIGHT / 10.0);
                varts[2] = 0.0;
                varts[3] = 0.0;
                varts[4] = -label / 2.0;
                varts[5] = 0.0;
            #endif

            sprintf(work, "%2.1f",
                min_value + (max_value - min_value) / 10.0 * i);
            GEOMadd_label(obj2, work, varts + 0, varts + 3,

```

```

        label, colors, flagy);
    }

    /* x 軸メモリ */
    varts[0] = i * (WIDTH / 10.0);
    varts[1] = y0;
    varts[2] = 0.0;
    varts[3] = i * (WIDTH / 10.0);
    varts[4] = y0 + 0.01;
    varts[5] = 0.0;
    GEOMadd_disjoint_line(obj1, varts, colors, 2, GEOM_COPY_DATA);

    if(!(i % 2)){
#ifdef ignore_offset
        varts[0] = i * (WIDTH / 10.0);
        varts[1] = y0 - 0.01 - offset;
        varts[2] = 0.0;
        varts[3] = 0.0;
        varts[4] = 0.0;
        varts[5] = 0.0;
#else
        varts[0] = i * (WIDTH / 10.0);
        varts[1] = y0 - 0.01;
        varts[2] = 0.0;
        varts[3] = 0.0;
        varts[4] = -label * 1.2;
        varts[5] = 0.0;
#endif

        sprintf(work, "%2.1f",
            minimum + (maximum - minimum) / 10.0 * i);
        GEOMadd_label(obj2, work, varts + 0, varts + 3,
            label, colors, flagx);
    }
}

free(varts);
free(colors);

GEOMedit_geometry (geom, "graph", obj1);
GEOMdestroy_obj(obj1);
GEOMedit_geometry (geom, "graph", obj2);
GEOMdestroy_obj(obj2);
}

/* 画面表示エリアの設定 */
{
    static float extent[6] = {
        -WIDTH * 0.1, WIDTH * 1.1,
        -HEIGHT * 0.1, HEIGHT * 1.1,
        0.0, 0.0};
    GEOMedit_window (geom, "graph", extent);
}

```

```
    }  
  
    return;  
}
```

## 3.2.7 モジュール制御関数

```

/*
 * ----- モジュール制御関数 -----
 */
type2o_compute(in_data, in_color,
               out_graph,
               param_reset, param_area, param_minimum, param_maxmum,
               param_label,
#ifdef ignore_offset
               param_label_offset,
#endif
               param_file_output, param_write,
               param_display_all, param_display_none,
               param_lines)

/* モジュールのポート/パラメータ変数 */
AVSfield_float *in_data;
AVScolormap *in_color;
GEOMedit_list *out_graph;
int param_reset;
char *param_area;
float *param_minimum;
float *param_maxmum;
float *param_label;
#ifdef ignore_offset
float *param_label_offset;
#endif
char *param_file_output;
int param_write;
char *param_lines;
{
    GEOMobj *obj;

    /*
     * 入力データの受け付け
     */
    if(AVSinput_changed("data", 0)) {
        int i;
        int line_no;
        float time, value;
        int result;

        result = 0;
        for(i = 0; i < MAXX(in_data); i++){
            line_no = in_data->data[i * 3 + 0];
            time     = in_data->data[i * 3 + 1];
            value     = in_data->data[i * 3 + 2];

            if(line_no < 0){
                /* 負の線番号があった時は、全てのデータをクリアする */
                data_free();
            }
        }
    }
}

```



```

        continue;
    }

    result += add_data(line_no, time, value);
}
if(result){
    modify_choice();
}

}

/*
 * リセットボタン
 */
if(AVSparameter_changed("reset")){
    if(param_reset){
        char *result;
        result = AVSmessage(NULL, AVS_Warning, NULL, "Question",
                             "Yes!No", "really?");
        if(!strcmp(result, "Yes")) data_free();
    }
}

/*
 * 表示エリア
 *
 * minimum < maxmum でなければ範囲指定を許さない。
 */
if(AVSparameter_changed("area")){
    if(!strcmp(param_area, "min&max")){
        if(*param_minimum >= *param_maxmum){
            AVSwarning("illegal parameter minimum or maxmum");
            AVSmodify_parameter("area", AVS_VALUE, "none", 0, 0);
        }
    }
}

/*
 * 表示線番号
 */
/* 全て表示 */
if(AVSparameter_changed("display all")){
    if(param_display_all){
        int i, j;
        for(j = i = 0; i < nstore; i++){
            if(j < store[i].line_no) j = store[i].line_no;
        }
        for(i = 0; i < nstore; i++){
            store[i].display = j - store[i].line_no + 1;
        }
        modify_choice();
    }
}

```

```

    }
}
/* 全て非表示 */
if(AVSparameter_changed("display none")){
    if(param_display_none){
        int i;
        for(i = 0; i < nstore; i++){
            store[i].display = -store[i].line_no;
        }
    }
    modify_choice();
}
}
/* 線番号が選択されたとき */
if(AVSparameter_changed("lines")){
    int i, line_no;

    if(strcmp(param_lines, "Line")){
        line_no = atoi(param_lines + 1);
        for(i = 0; i < nstore; i++){
            if(store[i].line_no == line_no){
                if(store[i].display > 0) store[i].display = 0;
                else store[i].display = 1;
                break;
            }
        }
        modify_choice();
    }
}

/*
 * グラフの描画
 */
if(AVSinput_changed("data", 0)
|| AVSinput_changed("color", 0)
|| AVSparameter_changed("reset")
|| AVSparameter_changed("area")
|| AVSparameter_changed("minimum")
|| AVSparameter_changed("maxmum")
|| AVSparameter_changed("label size")
|| AVSparameter_changed("write")
|| AVSparameter_changed("display all")
|| AVSparameter_changed("display none")
|| AVSparameter_changed("lines")){

    FILE *fp;

    if(param_write){
        /* 出力ファイルのオープン */
        if(param_file_output == NULL) fp = NULL;
        else{
            fp = fopen(param_file_output, "w");

```

```
        if(fp == NULL){
            AVSwarning("file %s can not create.", param_file_output);
        }
    }
    else fp = NULL;

    *out_graph = GEOMinit_edit_list(*out_graph);
    create_graph(*out_graph, *param_label,
#ifdef ignore_offset
        *param_label_offset,
#endif
        in_color, param_area,
        *param_minimum, *param_maxmum, fp);
    /*
    GEOMedit_visibility(*out_graph, "graph", -1);
    */
    if(fp != NULL) fclose(fp);
}

return (1);
}
```

## 3.2.8 モジュール記述関数

```

/*
 * ----- モジュール記述関数 -----
 */
type2o()
{
    int param;

    AVSset_module_name("Type2 disp", MODULE_RENDER);

    AVScreate_input_port("data",
                        "field 1D 0-space 3-vector float irregular",
                        REQUIRED);
    AVScreate_input_port("color", "colormap", OPTIONAL);

    AVScreate_output_port("graph", "geom");

    /* リセット */
    param = AVSadd_parameter("reset", "oneshot", 0, 0, 1);
    AVSadd_parameter_prop(param, "width", "integer", 4);
    AVSconnect_widget(param, "oneshot");

    /* 表示範囲 */
    param = AVSadd_parameter("area", "choice",
                        "none", "none#min#max#min&max", "#");
    AVSconnect_widget(param, "radio_buttons");
    AVSadd_parameter_prop(param, "width", "integer", 2);
    AVSadd_parameter_prop(param, "columns", "integer", 2);
    AVSadd_parameter_prop(param, "title", "string", "area");
    param = AVSadd_float_parameter("minimum",
                        0.0, FLOAT_UNBOUND, FLOAT_UNBOUND);
    AVSadd_parameter_prop(param, "width", "integer", 4);
    AVSconnect_widget(param, "typein_real");
    param = AVSadd_float_parameter("maximum",
                        0.0, FLOAT_UNBOUND, FLOAT_UNBOUND);
    AVSadd_parameter_prop(param, "width", "integer", 4);
    AVSconnect_widget(param, "typein_real");

    /* ラベル */
    param = AVSadd_float_parameter("label size",
                        0.05, FLOAT_UNBOUND, FLOAT_UNBOUND);
    AVSadd_parameter_prop(param, "width", "integer", 4);
    AVSconnect_widget(param, "typein_real");

#ifdef ignore_offset
    param = AVSadd_float_parameter("label offset",
                        0.05, FLOAT_UNBOUND, FLOAT_UNBOUND);
    AVSadd_parameter_prop(param, "width", "integer", 4);
    AVSconnect_widget(param, "typein_real");
#endif
}

```

```

/* ファイル出力 */
{
    char result[256];
    if(getenv("PWD") == NULL) strcpy(result, "./");
    else{
        strcpy(result, getenv("PWD"));
        strcat(result, "/");
    }

    param = AVSadd_parameter("file output", "string",
                             result, "", ".txt");
    AVSconnect_widget(param, "browser");
}

param = AVSadd_parameter("write", "oneshot", 0, 0, 1);
AVSadd_parameter_prop(param, "width", "integer", 4);
AVSconnect_widget(param, "oneshot");

/* 表示線番号 */
param = AVSadd_parameter("display all", "oneshot", 0, 0, 1);
AVSadd_parameter_prop(param, "width", "integer", 4);
AVSconnect_widget(param, "oneshot");
param = AVSadd_parameter("display none", "oneshot", 0, 0, 1);
AVSadd_parameter_prop(param, "width", "integer", 4);
AVSconnect_widget(param, "oneshot");
param = AVSadd_parameter("lines", "choice", "Line", "Line", "#");
AVSconnect_widget(param, "radio_buttons");
AVSadd_parameter_prop(param, "width", "integer", 1);
AVSadd_parameter_prop(param, "columns", "integer", 4);

AVSset_compute_proc(type2o_compute);
}

```

### 3.2.9 初期化関数

```

/*
 * ----- 初期化関数 -----
 */
AVSinit_modules()
{
    AVSmodule_from_desc(type2o);
}

```

### 3.3 Makefile

Makefile の一例を示す。演算モジュールのソースは `cnn.c` という一つのファイルに、グラフ表示モジュールは `type2o.c` というファイルに納められているものとする。

```

INC_FILE=$(ROOT)/usr/avs/include/Makeinclude
include $(INC_FILE)

AVS_LIBS = $(ROOT)/usr/avs/lib
BASELIBS=-lgeom -lutil -lm
FLOWLIBS=-L$(AVS_LIBS) -lflow_c $(BASELIBS) $(LASTLIBS)
CSIMLIBS=-L$(AVS_LIBS) -lsim_c $(BASELIBS) $(LASTLIBS)
MODLIBS=-L$(AVS_LIBS) -lmdata -lmfilt -lmmapp -lmrend -lmconvex -ldf -lrf
F77_FLOWLIBS=-L$(AVS_LIBS) -lflow_f $(BASELIBS) $(LASTFLIBS)
F77_SIMLIBS=-L$(AVS_LIBS) -lsim_f $(BASELIBS) $(LASTFLIBS)
F77_BIND = $(ROOT)/usr/avs/bin/f77_binding
AVS_INC = -I$(ROOT)/usr/avs/include
F77_INC = $(ROOT)/usr/avs/include
CFLAGS=$(APARCFLAGS) $(AVS_INC)
F77FLAGS=$(AFFLAGS)
FFLAGS=$(F77FLAGS)

all: type2o cnn

clean:
rm -f *.o $(EXAMPLES) avs examples.lint io_funs_f77.c qix_rand.c

avs:
ln -s $(F77_INC) avs

type2o: type2o.o
$(CC) $(CFLAGS) -o type2o type2o.o $(FLOWLIBS)

cnn: avs cnn.o
$(CC) $(CFLAGS) -o cnn  cnn.o $(CSIMLIBS)

```

## 第 4 章

# モジュールの組み合わせによる AVS ネットワークの構成

### 4.1 モジュールの組み合わせ方

前章のソースを make することによって、実行ファイル `cnn`、`type2o` の 2 つができる。この両者の存在するディレクトリで AVS を実行すると、図 4.1 の様な初期画面があらわれる。

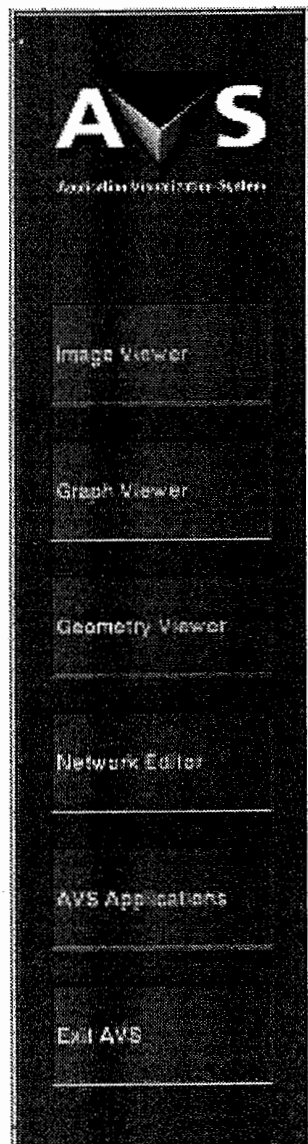


図 4.1: AVS の初期画面

シミュレーションに必要な機能を得るためには、ネットワークエディタ上で作成したモジュールと既存のモジュールを組み合わせ、ネットワークを構成する必要がある。ネットワークエディタを使用するには、上から4つ目の‘Network Editor’をクリックする。すると、図4.2の様なネットワークエディタが現れる。

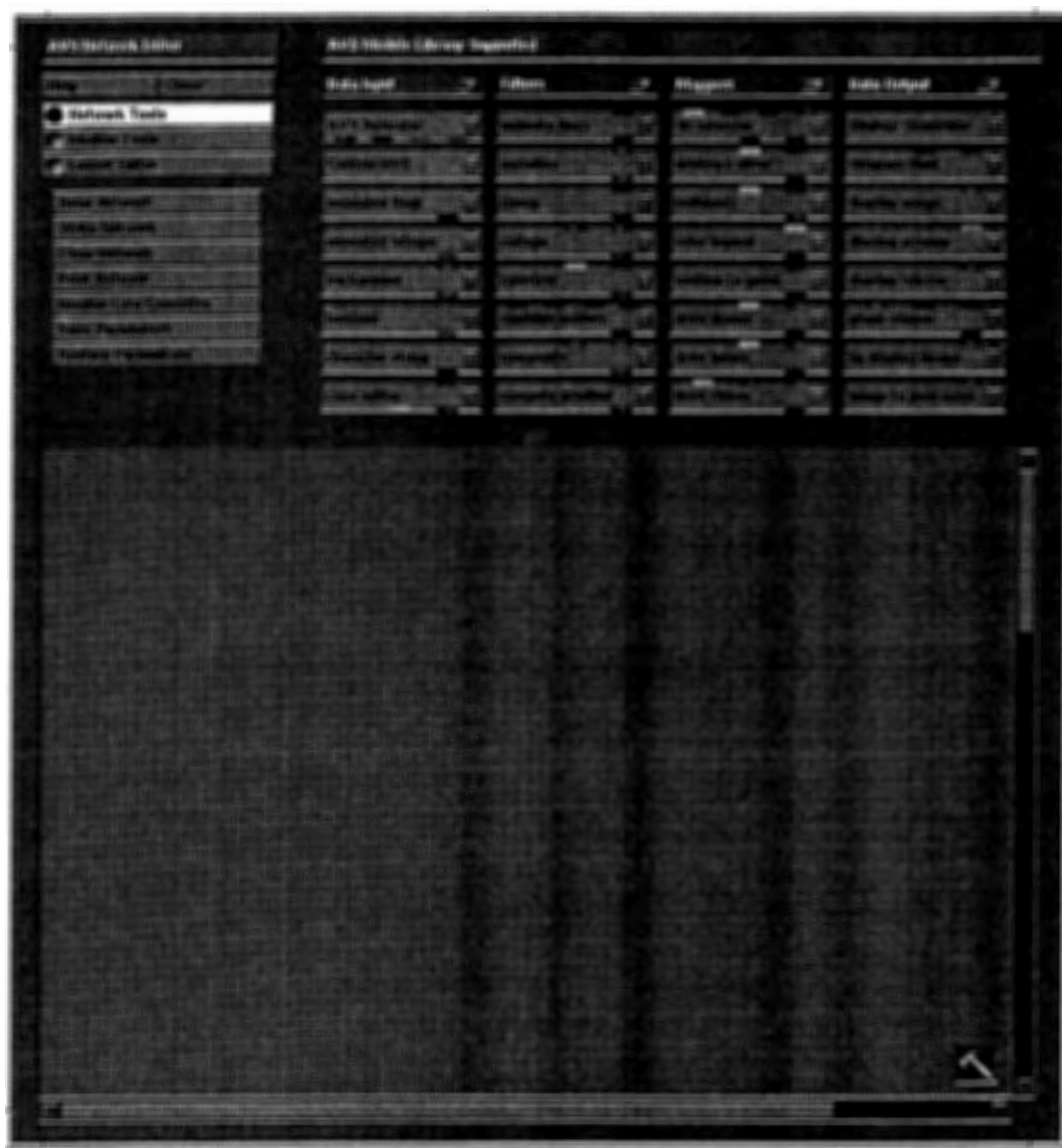


図 4.2: ネットワークエディタ

まず、この左上の部分の‘Module Tools’をクリックし、さらに‘Read Module(s)’をクリックすると、ファイルを選択するダイアログボックスが現れるので、先ほど make した‘cnn’を選ぶ。すると、右側のモジュール名が並んだ部分に、‘NN with AVS <ATR>’という名前のモジュールが現れる。このモジュールをマウスでドラッグしてネットワークエディタ(図4.2)の画面下部に持っていく。同様にファイル‘type2o’を選択すると、‘Type2 disp’というモジュールが現れる。こちらも同様にドラッグして画面下部に持っていく。

この他、‘render geometry’、‘displar pixmap’というモジュールを2つずつ画面下部に持っていく。<sup>1</sup>

モジュールの上下に付いているのが、入出力用のポートである。このポートをマウスの中ボタンで押さえて適当なラインを選択して離すことによって、モジュール同士をつなぐことができる。(切る時は右ボタン)

<sup>1</sup>‘NN with AVS <ATR>’は左端の Data Input の領域に、‘Type2 disp’、‘render geometry’、‘displar pixmap’は右端の Data Output の領域にあらわれる。



これらのモジュールを図 4.3 の様に結合させる。すなわち作成した2つのモジュールの出力は、'render geometry' モジュールによって geometry データに変換され、'display pixmap' によって表示される。なお AVS のバージョンによっては、'render geometry'、'display pixmap' の両者の機能を兼ね備えた 'geometry viewer' が使用可能である。

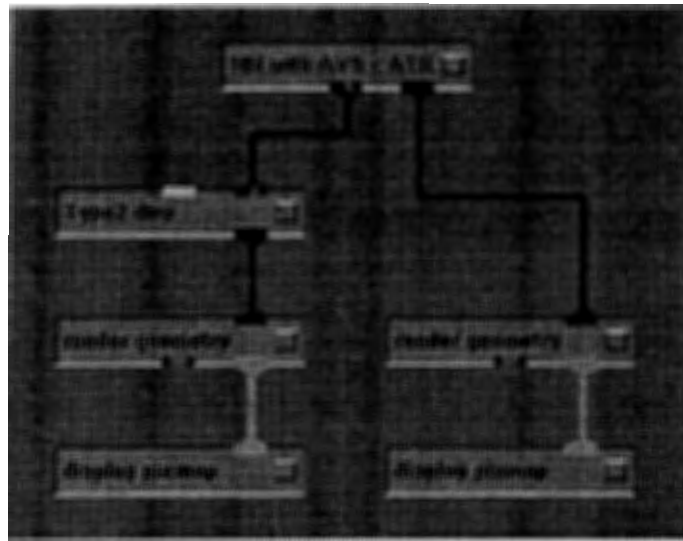


図 4.3: モジュール同士をつなげてネットワークにしたところ

表示された画像はマウスの右ボタンで表示位置を、シフトキー + 中ボタンで大きさを、それぞれ変更可能である。<sup>2</sup> またマウスの中ボタンのみを使用すれば、画像を3次元空間中で回転させることができる。

ネットワークエディタ (図 4.2) 左上の 'Layout Editor' をクリックすると、各モジュールのボタン、ダイアル、ウィンドウサイズ等をマウスを使って自由に再配置することが可能である。図 4.4 にこの機能を使って、使いやすい様に再配置した後の画面の様子を示す。

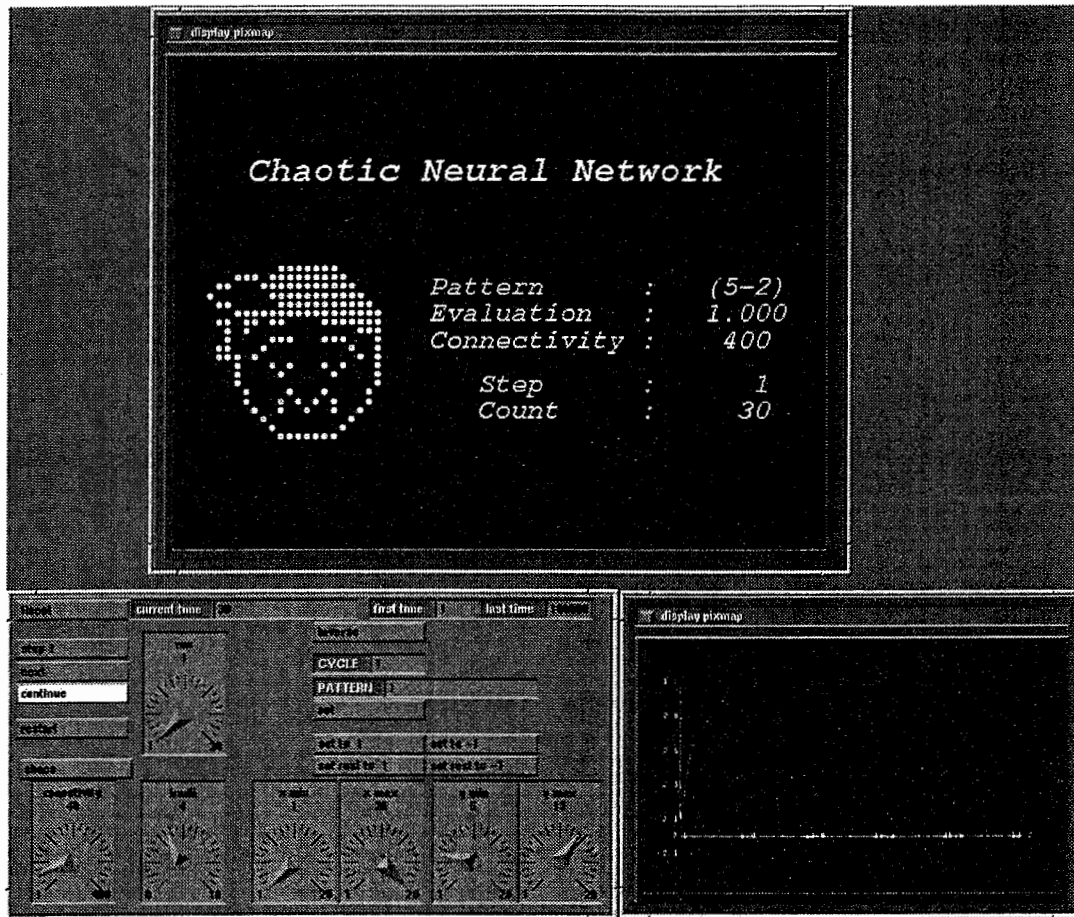


図 4.4: 動作中のシミュレータ

この完成したネットワーク (ボタン等の位置も含めて) は network ファイルとして保存可能である。network ファイルについては次節で述べる。

<sup>2</sup> 画像のウィンドウ中での位置をある程度自動的に最適化するには、画面左側のウィンドウ上部の 'Data Viewers' から 'Geometry Viewer' を選び、さらに 'Normalize' ボタンをクリックすればよい。

## 4.2 network ファイル

ネットワークエディタ (図 4.2) 左上の 'Network Tools' を選択し、さらに 'Write Network' をクリックすることにより作成したネットワークを保存できる。このようにして一度作成した network ファイルは、ネットワークエディタから呼び出すことができる。('Read Network' をクリック)

また network ファイルは、AVS 用のスクリプトファイルとなっているため、直接コマンドラインから起動することも可能である。

以下に、前節で作成したネットワークの network ファイルを示す。ただし、通常ユーザは network ファイルの中身を意識する必要は無い。

```
#!/usr/bin/avs -network
version 3.9 (25.58 convex)
module "display pixmap.user.0" -xy 383,425
module "render geometry.user.1" -xy 383,357
module "NN with AVS < ATR >.user.2" -xy 293,202 -ex ./cnn
module "Type2 disp.user.3" -xy 202,284 -ex ./type2o
module "display pixmap.user.4" -xy 202,425
module "render geometry.user.5" -xy 202,357
port_connect "render geometry.user.1":0 "display pixmap.user.0":0
port_connect "NN with AVS < ATR >.user.2":0 "render geometry.user.1":0
port_connect "NN with AVS < ATR >.user.2":1 "Type2 disp.user.3":0
port_connect "Type2 disp.user.3":0 "render geometry.user.5":0
port_connect "render geometry.user.5":0 "display pixmap.user.4":0
mod_render "render geometry.user.1"
view AVS 602x446+297+57 0.001000 0.001000 0.990909 {

  no_depth_cue
  zbuffer
}
light directional 1 {
}
light ambient 16 {
  set_matrix    0.000000    0.000000    0.000000    0.000000
                0.000000    0.000000    0.000000    0.000000
                0.000000    0.000000    0.000000    0.000000
                0.000000    0.000000    0.000000    0.000000
}
set_matrix 0.134611 0.000000 0.000000 0.000000
           0.000000 0.134611 0.000000 0.000000
           0.000000 0.000000 0.134611 0.000000
           0.000000 0.000000 0.000000 1.000000
set_position -4.588745 -2.130273 0.000000
}
mod_render "render geometry.user.5"
view Untitled 379x222+734+593 0.000000 0.000000 0.000000 {

  no_depth_cue
  zbuffer
}
light directional 1 {
}
light ambient 16 {
```

```

set_matrix    0.000000    0.000000    0.000000    0.000000
              0.000000    0.000000    0.000000    0.000000
              0.000000    0.000000    0.000000    0.000000
              0.000000    0.000000    0.000000    0.000000
}
set_matrix 4.166667 0.000000 0.000000 0.000000
           0.000000 4.166667 0.000000 0.000000
           0.000000 0.000000 4.166667 0.000000
           -4.166667 -2.083333 0.000000 1.000000
set_window -0.200000 2.200000 -0.100000 1.100000 0.000000 0.000000
}
shell "ui" shell
panel Application -w app_panel -p ui -xy 0,0 -wh 260,1024
panel "Top Level Stack" -w master_stack -p Application -xy 2,100 -wh 257,603
panel "display pixmap.user.0" -w panel -p "Top Level Stack" \
    -xy 0,63 -wh 257,257
manipulator "display pixmap.user.0:Store Frames" -w toggle -p "display pixmap.user.0" \
    -xy 10,10 -wh 118,21
manipulator "display pixmap.user.0:Append Frame" -w oneshot -p "display pixmap.user.0" \
    -xy 10,28 -wh 118,21
manipulator "display pixmap.user.0:Delete Current" -w oneshot -p "display pixmap.user.0" \
    -xy 10,53 -wh 118,21
manipulator "display pixmap.user.0:Replay" -w radio_buttons -p "display pixmap.user.0" \
    -xy 10,75 -wh 118,64
manipulator "display pixmap.user.0:Current Frame" -w islider -p "display pixmap.user.0" \
    -xy 10,136 -wh 235,31
manipulator "display pixmap.user.0:Max Frames" -w typein_integer -p "display pixmap.user.0" \
    -xy 10,169 -wh 236,21
manipulator "display pixmap.user.0:Replay Speed" -w islider -p "display pixmap.user.0" \
    -xy 10,193 -wh 235,31
manipulator "display pixmap.user.0:Save Image" -w typein -p "display pixmap.user.0" \
    -xy 10,225 -wh 236,21
panel "render geometry.user.1" -w panel -p "Top Level Stack" \
    -xy 0,63 -wh 136,61
manipulator "render geometry.user.1:Output Image" -w toggle -p "render geometry.user.1" \
    -xy 10,10 -wh 118,21
manipulator "render geometry.user.1:High Quality" -w toggle -p "render geometry.user.1" \
    -xy 10,28 -wh 118,21
panel "Type2 disp.user.3" -w panel -p "Top Level Stack" -xy 0,63 -wh 257,452
manipulator "Type2 disp.user.3:reset" -w oneshot -p "Type2 disp.user.3" \
    -xy 10,10 -wh 236,21
manipulator "Type2 disp.user.3:area" -w radio_buttons -p "Type2 disp.user.3" \
    -xy 10,28 -wh 236,43
manipulator "Type2 disp.user.3:minimum" -w typein_real -p "Type2 disp.user.3" \
    -xy 10,75 -wh 236,21
manipulator "Type2 disp.user.3:maximum" -w typein_real -p "Type2 disp.user.3" \
    -xy 10,94 -wh 236,21
manipulator "Type2 disp.user.3:label size" -w typein_real -p "Type2 disp.user.3" \
    -xy 10,118 -wh 236,21
manipulator "Type2 disp.user.3:file output" -w browser -p "Type2 disp.user.3" \
    -xy 10,136 -wh 235,195

```

```

manipulator "Type2 disp.user.3:write" -w oneshot -p "Type2 disp.user.3" \
  -xy 10,334 -wh 236,21
manipulator "Type2 disp.user.3:display all" -w oneshot -p "Type2 disp.user.3" \
  -xy 10,357 -wh 236,21
manipulator "Type2 disp.user.3:display none" -w oneshot -p "Type2 disp.user.3" \
  -xy 10,375 -wh 236,21
manipulator "Type2 disp.user.3:lines" -w radio_buttons -p "Type2 disp.user.3" \
  -xy 10,400 -wh 59,21
panel "display pixmap.user.4" -w panel -p "Top Level Stack" \
  -xy 0,63 -wh 257,257
manipulator "display pixmap.user.4:Store Frames" -w toggle -p "display pixmap.user.4" \
  -xy 10,10 -wh 118,21
manipulator "display pixmap.user.4:Append Frame" -w oneshot -p "display pixmap.user.4" \
  -xy 10,28 -wh 118,21
manipulator "display pixmap.user.4:Delete Current" -w oneshot -p "display pixmap.user.4" \
  -xy 10,53 -wh 118,21
manipulator "display pixmap.user.4:Replay" -w radio_buttons -p "display pixmap.user.4" \
  -xy 10,75 -wh 118,64
manipulator "display pixmap.user.4:Current Frame" -w islider -p "display pixmap.user.4" \
  -xy 10,136 -wh 235,31
manipulator "display pixmap.user.4:Max Frames" -w typein_integer -p "display pixmap.user.4" \
  -xy 10,169 -wh 236,21
manipulator "display pixmap.user.4:Replay Speed" -w islider -p "display pixmap.user.4" \
  -xy 10,193 -wh 235,31
manipulator "display pixmap.user.4:Save Image" -w typein -p "display pixmap.user.4" \
  -xy 10,225 -wh 236,21
panel "render geometry.user.5" -w panel -p "Top Level Stack" \
  -xy 0,150 -wh 136,61
manipulator "render geometry.user.5:Output Image" -w toggle -p "render geometry.user.5" \
  -xy 10,10 -wh 118,21
manipulator "render geometry.user.5:High Quality" -w toggle -p "render geometry.user.5" \
  -xy 10,28 -wh 118,21
panel "NN with AVS < ATR >.user.2" -w panel -p ui -xy 161,613 -wh 630,332
manipulator "NN with AVS < ATR >.user.2:first time" -w typein_integer -p "NN with AVS < ATR >.user.2" \
  -xy 373,4 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:last time" -w typein_integer -p "NN with AVS < ATR >.user.2" \
  -xy 491,4 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:current time" -w typein_integer -p "NN with AVS < ATR >.user.2" \
  -xy 125,4 -wh 236,21
manipulator "NN with AVS < ATR >.user.2:step 1" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 4,44 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:next" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 4,68 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:continue" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 4,91 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:restart" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 4,127 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:chaos" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 9,169 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:CYCLE" -w typein_integer -p "NN with AVS < ATR >.user.2" \
  -xy 315,61 -wh 118,21

```

```

manipulator "NN with AVS < ATR >.user.2:PATTERN" -w typein_integer -p "NN with AVS < ATR >.user.2" \
  -xy 315,86 -wh 236,21
manipulator "NN with AVS < ATR >.user.2:set" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 315,109 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:set to 1" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 315,144 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:set to -1" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 433,144 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:set rest to 1" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 315,167 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:set rest to -1" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 433,167 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:iradii" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 135,192 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:run" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 136,37 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:conectivity" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 20,191 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:x min" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 251,192 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:x max" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 342,192 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:y min" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 440,192 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:y max" -w idial -p "NN with AVS < ATR >.user.2" \
  -xy 530,192 -wh 89,129
manipulator "NN with AVS < ATR >.user.2:Reset" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 4,4 -wh 118,21
manipulator "NN with AVS < ATR >.user.2:inverse" -w toggle -p "NN with AVS < ATR >.user.2" \
  -xy 315,28 -wh 118,21
panel "display pixmap.user.0!display" -w container -p ui -xy 309,4 -wh 728,582\
  -P zoom_coords string "0 0 0 0 0 <$NULL> 0 0 0 0"
panel "display pixmap.user.4!display" -w container -p ui -xy 806,614 -wh 474,327\
  -P zoom_coords string "0 0 0 0 0 <$NULL> 0 0 0 0"
manipulator "render geometry.user.1":width -w none
manipulator "render geometry.user.1":height -w none
manipulator "render geometry.user.1":object transform -w none
manipulator "render geometry.user.1":add to object transform -w none
manipulator "render geometry.user.1":light transform -w none
manipulator "render geometry.user.1":color -w none
manipulator "render geometry.user.1":mode -w none
manipulator "render geometry.user.1":transparency -w none
manipulator "render geometry.user.1":object -w none
manipulator "render geometry.user.1":line radius -w none
manipulator "render geometry.user.1":point radius -w none
manipulator "render geometry.user.5":width -w none
manipulator "render geometry.user.5":height -w none
manipulator "render geometry.user.5":object transform -w none
manipulator "render geometry.user.5":add to object transform -w none
manipulator "render geometry.user.5":light transform -w none
manipulator "render geometry.user.5":color -w none

```

```
manipulator "render geometry.user.5":mode -w none  
manipulator "render geometry.user.5":transparency -w none  
manipulator "render geometry.user.5":object -w none  
manipulator "render geometry.user.5":line radius -w none  
manipulator "render geometry.user.5":point radius -w none  
# End of file
```





## 第 5 章

### まとめ

本レポートではニューラルネットワークの可視化を実現するために、汎用の科学技術計算用可視化ツールである AVS を用いた例について述べてきた。この AVS を用いたシミュレーションは、当所でニューラルネットワーク研究を進めていく上で大きな力となった。その成果については、参考文献 [3]-[8] を参照されたい。

本システムの利点としては、ユーザインターフェース、画面出力などのモジュールが標準で用意されているため、比較的簡単に必要な機能が実現できることが挙げられる。逆に問題点としては、ニューラルネットワークの規模によっては、一般的なワークステーションで実行する場合は計算速度が、また計算を高速な計算機で行なわせた場合はネットワークの転送速度がネックとなり十分な動作速度が得られない可能性があることが挙げられる。したがって大規模なネットワークのシミュレーションを行なう場合には、高速な計算機と高速なネットワークが要求される。

本レポートが、ニューラルネットワークに限らず AVS を用いたシステムを作成する場合の参考 (ひな型) になれば幸いである。

## 謝辞

本レポート発表の機会を与えて下さった ATR 光電波通信研究所 猪股英行社長、渡辺敏英 通信デバイス研究室長に深謝いたします。本シミュレータ作成にあたって新上和正主任研究員、Peter Davis 客員研究員を初めとする計算物理グループの方々の助言を頂きました。感謝いたします。

## 参考文献

- [1] S. Amari, Neural Theory of Association and Concept-formation, Biol. Cybernetics, vol.26, pp.175-185 (1977).
- [2] S. Nara, P. Davis, M. Kawachi and H. Totsuji, Chaotic Memory Dynamics in a Recurrent Neural Netwrk with Cycle Memories Embedded by Pseudo-Inverse Method, International Journal of Bifurcation and Chaos, vol. 5, No. 4 , pp.1205-1212 (1995).
- [3] M. Yamaga, P. Davis, S. Nara and Y. Iino, Hierarchical Chaotic Memory Dynamics in a Recurrent Neural Circuit, International Journal of Bifurcation and Chaos, submitted.
- [4] M. Yamaga, P. Davis and S. Nara, Hierarchical Chaotic Transitions among Memory Sequences in a Recurrent Neural Circuit, Proceedings of The 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, p.613-614 (1994).
- [5] 山賀陸夫, P. Davis, 奈良重俊, シーケンシャルな記憶をもつ神経回路網におけるカオス, テレビジョン学会年次大会, pp.149-150 (1994).
- [6] 山賀陸夫, P. Davis, 奈良重俊, 神経回路網における記憶シーケンス間のカオス, テレビジョン学会技術報告, vol.19, No.25, pp.1-5, IPU 95-21 (1995).
- [7] 山賀陸夫, P. Davis, 連続パターンを記憶させた神経回路網におけるカオスと引き込み現象, 電子情報通信学会 基礎・境界ソサイエティ大会, p.36 (1995).
- [8] M. Yamaga and P. Davis, Locking of Memory Sequences and Chaos in a Recurrent Neural Network, Proceedings of NOLTA'95, pp.199-202 (1995).