

1 1行単位のファイル入出力 234 頁

1.1 学習のポイント

1行単位のファイル入出力関数 `fgets`、`fputs` について学びます。

`fgets`(ファイルから 1 行入力)

書式

```
fgets(buf, n, fp)
```

機能

ファイル・ポインタ `fp` で示されるファイルから、バッファ `buf` に 1 行リードします。1 行の終わりは `'\n'` コードで判定されます。

`fgets` が正常におこなわれれば `buf` へのポインタが渡されますが、ファイル・エンドまたはエラーなら `NULL` を返します。

なお、MS-DOS の 1 行の終わりは `CR`(復帰)・`LF`(改行) の 2 文字で行なっていますが、`fgets` は `CR`・`FL` を `C` の行末記号 `'\n'` に変換し、文字列の終わりとして `'\0'` を付加します。

ファイル `fp` から 1 行単位でリードし、ディスプレイに出力していくプログラムは次のようになります。

```
while(fgets(buf, 256, fp) != NULL)
    printf("%s", buf);
```

`fputs`(ファイルへの 1 行出力)

書式

```
fputs(buf, fp)
```

機能

バッファ `buf` の文字列 (`'\0'` が終了マーク) をファイル・ポインタ `fp` で示されるファイルにライトします。`'\0'` はライトされません。`fget` とは逆に `'\n'` は `CR`・`LF` コードに変換されてライトされます。

fputs は最後に出力した文字を返しますが、エラーなら-1 を返します。

1.2 例題 52 reidai52.c 236 頁

fgets を用いて行番号付き TYPE コマンドを作りなさい。

```
/*
 例題 52
  fgets を用いて行番号付き TYPE コマンドを作りなさい。
  reidai52.c
*/

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int count = 0;
    char buf[256];
    FILE *fp;

    if((fp = fopen(argv[1], "r")) == NULL){
        printf("Can't open File\n");
        exit(1);
    }

    while(fgets(buf, 256, fp) != NULL){
        printf("%04d:\t%s", ++count, buf);
    }
    fclose(fp);

    return 0;
}
```

1.3 練習問題 52 rensyu52.c 238 頁

```
/*
 練習問題 52
  キーボードから文字列を 1 行単位で入力し、それをファイルに書き出して行く
*/
```

プログラムを作りなさい。行末を示す'\n' コードも含めてバッファに入力するためには、scanf ではなく gets(buf, 256, stdin) とします。

```
rensyu52.c
*/

#include <stdio.h>
#include <stdlib.h>

int main()
{
    FILE *fp;
    char buf[256];

    if((fp = fopen("abc.txt","w")) == NULL){
        printf("Can't open File\n");
        exit(1);
    }

    while(fgets(buf, 256, stdin) != NULL){
        fputs(buf, fp);
    }
    fclose(fp);

    return 0;
}
```

2 機械学習と深層学習

2.1 帰納的学習の例題-株価の予想-

簡単なモデルを用いた株価の予想を例題とします。

X 社の株価は、X 社に関連する企業である A 社～J 社の株価変動によって決定されるものとします。つまり、前日の A 社～J 社の株価変動があるパターンをとった時には当日の X 社の株価が上昇し、別のパターンでは株価が下降すると仮定します。そこで、過去の具体的な株価変動の事例を集めて帰納的な機械学習を行なうことでパターンを抽出し、X 社の株価の予測をする知識を得ることを考えます。

2.2 帰納学習による株価予想 learnstock.c

```
/******  
/*          learnstock.c          */  
/* 単純な帰納的学習の例題プログラム      */  
/*          パターン学習器          */  
/*100 個の学習データを読み込んで、      */  
/*適合する 10 桁の 2 進数パターンを答えます */  
/*使い方                                  */  
/*:\Users\odaka\dl\ch2>learnstock<ldata.txt */  
/******  
  
/*Visual Studio との互換性確保 */  
#define _CRT_SECURE_NO_WARNINGS  
  
/*ヘッダファイルのインクルード*/  
#include <stdio.h>  
#include <stdlib.h>  
  
/* 記号定数の定義          */  
#define OK 1  
#define NG 0  
#define SETSIZE 100 /*学習データセットの大きさ*/  
#define CNO 10 /*学習データの桁数 (10 社分)*/  
#define GENMAX 10000 /*解候補生成回数*/  
#define SEED 32767 /*乱数のシード*/  
  
/* 関数のプロトタイプの宣言 */  
void readdata(int data[SETSIZE][CNO],int teacher[SETSIZE]) ;  
    /*学習データセットの読み込み*/  
int rand012() ;/*0, 1 又は 2 を返す乱数関数*/  
int calcscore(int data[SETSIZE][CNO],int teacher[SETSIZE],  
    int answer[CNO]) ;  
    /*解候補パターンのスコア (0~SETSIZE 点) の計算*/  
  
/******
```

```

/* main() 関数 */
/*****/
int main()
{
    int i, j;
    int score = 0; /*スコア (0~SETSIZE 点)*/
    int answer[CNO]; /*解候補*/
    int data[SETSIZE][CNO]; /*学習データセット*/
    int teacher[SETSIZE]; /*教師データ*/
    int bestscore = 0; /*スコアの最良値*/
    int bestanswer[CNO]; /*探索途中での最良解*/

    srand(SEED); /*乱数の初期化*/

    /*学習データセットの読み込み*/
    readdata(data, teacher) ;

    /*解候補生成と検査*/
    for (i = 0; i<GENMAX; ++i) {
        /*解候補生成*/
        for (j = 0; j<CNO; ++j) {
            answer[j] = rand012();
        }

        /*検査*/
        score=calcscore(data, teacher, answer) ;

        /*最良スコアの更新*/
        if (score>bestscore) { /*これまでの最良値なら更新*/
            for (j = 0; j<CNO; ++j)
                bestanswer[j] = answer[j];
            bestscore = score;
            for (j = 0; j<CNO; ++j)
                printf("%1d ", bestanswer[j]);
            printf(":score=%d\n", bestscore);
        }
    }
    /*最良解の出力*/
    printf("\n 最良解\n");
    for (j = 0; j<CNO; ++j)

```

```

    printf("%1d ", bestanswer[j]);
    printf(":score=%d\n", bestscore);

    return 0;
}

/*****
/*          calcscore() 関数          */
/*解候補パターンのスコア(0~SETSIZE点)の計算*/
*****/
int calcscore(int data[SETSIZE][CNO],int teacher[SETSIZE],
              int answer[CNO])
{
    int score = 0; /*スコア(0~SETSIZE点)*/
    int point ; /*一致した桁数(0~CNO) */
    int i,j ;

    for (i = 0; i<SETSIZE; ++i) {
        /*一致度計算*/
        point = 0;
        for (j = 0; j<CNO; ++j) {
            if (answer[j] == 2) ++point; /*ワイルドカード*/
            else if (answer[j] == data[i][j]) ++point; /*一致*/
        }

        if ((point == CNO) && (teacher[i] == 1)) {
            ++score;
        }
        else if ((point != CNO) && (teacher[i] == 0)) {
            ++score;
        }
    }
    return score ;
}

/*****
/*          readdata() 関数          */
/*学習データセットの読み込み*/
*****/

```

```

void readdata(int data[SETSIZE][CNO],int teacher[SETSIZE])
{
    int i,j ;

    for (i = 0; i<SETSIZE; ++i) {
        for (j = 0; j<CNO; ++j) {
            scanf("%d", &data[i][j]);
        }
        scanf("%d", &teacher[i]);
    }
}

/*****
/*      rand012() 関数      */
/*  0, 1 又は 2 を返す乱数関数 */
*****/
int rand012()
{
    int rnd ;

    /*乱数の最大値を除く*/
    while((rnd=rand())==RAND_MAX) ;
    /*乱数の計算*/
    return (double)rnd/RAND_MAX*3 ;
}

```

2.3 learnstock.exe の使い方

```
./learnstock < ldata.txt
```