

# ソフトウェア設計書

## 要件

要件番号	項目	概要
1	neural network	分岐のないニューラルネットワークを構築できること
2	deep-learning	ニューラルネットワークの学習ができること
3	save-load	ニューラルネットワークの情報を保存・呼び出しして再利用できること。
4	Flex-Initialize	ニューラルネットワークの構成を指定して(Layerの組み換え)初期化が可能であること。
5	teacher data file	外部で生成した教師データを読み込んで利用することができること
6	learning state	学習進捗を出力することができること。

## 設計

### 機能一覧

機能番号	項目	概要
1	foward calculation	ニューラルネットワークの順方向の計算を実施する。 入力の配列に対し計算結果の配列を出力する。
2	backward calculation	ニューラルネットワークの逆方向の計算を実施する。 ニューラルネットワークの出力と期待値の値を比較して、 ニューラルネットワーク変数の修正を行う。（誤差逆伝播）

機能番号	項目	概要
3	load teacher file	教師データの読み込み。外部ファイルから教師データを読み込む
4	save neural network	ニューラルネットワーク（構成・パラメータ）をファイルに保存する。
5	load neural network	ニューラルネットワーク(構成・パラメータ)のファイルを読み込み、再構成する。

# インターフェイスデザイン

## layerハンドラー

### H\_LAYER

#### 概要

ニューラルネットワークの1レイヤーの情報を保持する。  
フォワード、バックワードの計算を提供する。  
決まった入力ベクトルのサイズと出力ベクトルのサイズを持つ。

### 変数 S\_LAYER

型	変数名	概要
int	type	layerのタイプ。1:ReLuレイヤー 2:Sigmoidレイヤー 3:Affineレイヤー 4:Sofmaxレイヤー 5:Sofmax-with-Lossレイヤー
unsinged int	input_size	layerの入力要素数
unsinged int	output_size	layerの出力要素数
void *	pLayerParams	layer内部変数へのポインタ
void *	pForwardOutput	順伝播計算の出力値へのポインタ
void *	pBackwardOutput	逆伝播計算の出力値へのポインタ

pLayerParamsの要素

Layer Type	要素
LT_ReLU	NULL
LT_Sigmoid	NULL
LT_Affine	void *pAffineParams[2]={pW,pB}(ポインタ配列)
LT_Softmax	NULL
LT_SoftmaxWithLoss	void * pSWLParams[2]={pY,pT}(ポインタ配列)

各Layer設計

ReLU

- ・ 概要  
各要素についてReLU活性化関数を作用させ、出力する。
- ・ 入出力要素数

$$\text{inputsize} = \text{outputsize}$$

- ・ 順伝播関数

$$Y[i] = \begin{cases} 0 & \text{if } X[i] < 0 \\ X[i] & \text{if } X[i] > 0 \end{cases}$$

- ・ 逆伝播関数

$$\frac{\partial L}{\partial X[i]} = \begin{cases} 0 & \text{if } Y[i] < 0 \\ \frac{\partial L}{\partial Y[i]} & \text{if } Y[i] > 0 \end{cases}$$

Sigmoid

- ・ 概要  
各要素についてSigmoid活性化関数を作用させ、出力する。
- ・ 入出力要素数

$$\text{inputsize} = \text{outputsize}$$

- ・順伝播関数

$$Y[i] = \frac{1}{1 + \exp(-X[i])}$$

- ・逆伝播関数

$$\frac{\partial L}{\partial X[i]} = \frac{\partial L}{\partial Y[i]} Y[i](1 - Y[i])$$

## Affine

- ・概要

入力の配列に線形変換行列 $W$ と $B$ を作用させた結果を出力する。

- ・入出力要素数

$W$ のサイズは(outputsize,inputsize)

$B$ のサイズは(outputsize,1)

- ・順伝播関数

入力の配列を行列 $X$ (inputsize,1)

出力の配列を行列 $Y$ (outputsize,1)

とする。

$$Y = W \cdot X + B$$

つまり

$$Y[i, 1] = \sum_j W[i, j] \cdot X[j, 1] + B[i, 1]$$

- ・逆伝播関数

$$\frac{\partial L}{\partial X[i]} = \sum_j \frac{\partial Y[j]}{\partial X[i]} \frac{\partial L}{\partial Y[j]} = \sum_j W[i, j] \cdot \frac{\partial L}{\partial Y[j]}$$

また

$$\begin{cases} \frac{\partial L}{\partial B[i]} &= \frac{\partial L}{\partial Y[i]} \\ \frac{\partial L}{\partial W[i, j]} &= X[j] \cdot \frac{\partial L}{\partial Y[i]} \end{cases}$$

もしくは

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \cdot W^T$$

$$\frac{\partial L}{\partial W} = X^T \cdot \frac{\partial L}{\partial Y}$$

$$\frac{\partial L}{\partial B} = \frac{\partial L}{\partial Y}$$

## Softmax

- ・ 概要
- ・ 入出力要素数
- ・ 順伝播関数
- ・ 逆伝播関数

## SoftmaxWithLoss

- ・ 概要

入力配列にソフトマックスを作用させて正規化し、さらに出力配列と期待値配列を直行エントロピー損失関数をかけて計算する。

- ・ 入出力要素数

$$\text{outputsize} = 1$$

- ・ 順伝播関数

$X[\text{inputsize}] \rightarrow Y[\text{inputsize}] \& T[\text{inputsize}] \rightarrow L[1]$ と計算を進める。(Tは期待値配列)

$$Y[i] = \frac{\exp(X[i])}{\sum_j \exp(X[j])}$$

$$L = - \sum_i T[i] \ln(Y[i])$$

- ・ 逆伝播関数

$$\frac{\partial L}{\partial X[i]} = Y[i] - T[i]$$

機能

H\_LAYER create\_layer

- ・ 概要

layerハンドラーを生成する。  
layer内部変数を作成し、ハンドラーに割り当てる。  
・ 引数

型	引数名	概要
int	type	Layerタイプを指定する。
int	input_size	layerの入力要素数
int	output_size	layerの出力要素数

- ・ 戻り値 H\_LAYER
- 正常：layerハンドラー  
異常：NULL

- ・ エラー判定
- 引数エラー  
typeが定義されていない値である  
inputsize<1  
outputsize<1  
各typeに対してinputsize,outputsizeが不正  
動的メモリ獲得エラー

int delete\_layer

- ・ 概要

layerハンドラーを削除する。  
ハンドラー変数のメモリも開放する。  
・ 引数

型	引数名	概要
H_LAYER	hLayer	削除するLayerハンドラーを渡す

・ 戻り値 int

正常 : 0

異常 : -1

・ エラー判定

引数がNULL

**int print\_layer**

・ 概要

layerの情報を標準出力に表示する。

id,type,入出力サイズ、計算パラメータを表示する。

・ 引数

型	引数名	概要
H_LAYER	hLayer	出力対象のlayerのハンドラー

・ 戻り値int

正常:0

異常:-1

・ エラー

引数がNULL

**int calc\_forward**

・ 概要

layerに順伝播の入力値を渡し、計算結果(pForwardOutput)を更新する。

・ 引数

型	引数名	概要
H_LAYER	hLayer	出力対象のlayerのハンドラー
double *	vInput	入力値の配列

・ 戻り値int

正常:0

異常:-1

・ エラー判定

引数が 1 つ以上NULL

計算エラー

int calc\_backward

・ 概要

layerに誤差逆伝播の入力値を渡し、計算結果(pBackwardOutput)を更新する。

・ 引数

型	引数名	概要
H_LAYER	hLayer	出力対象のlayerのハンドラー
double *	vInput	入力値の配列

・ 戻り値int

正常:0

異常:-1

・ エラー判定

引数が 1 つ以上NULL

計算エラー

matrixハンドラー

H\_MATRIX

概要

行列情報の保持と計算を行う。

行列のフォーマットは 行数:row 配列:column で指定する。

行列の要素数は  $size = row * column$  となる。

行列の要素は行: $i$  列: $j$ で  $i * column + j$ を指定する。

以上、以下の表で概念を表す。

i\j	0	1	*	j	*	column-1
0	[0]	[1]	*	[j]	*	[column-1]
1	[column+0]	[column+1]	*	[column+i]	*	[2*column-1]
*	*	*	*	*	*	*
i	[i*column+0]	[i*column+1]	*	[i*column+j]	*	(i+1)*column-1
*	*	*	*	*	*	*



<b>ij</b>	<b>0</b>	<b>1</b>	<b>*</b>	<b>j</b>	<b>*</b>	<b>column-1</b>
row-1	[(row-1)*column+0]	*	*	[(row-1)*column+j]	*	[row*column-1]

### 変数 S\_MATRIX

型	変数名	概要
double*	pElem	行列の値を持つ配列へのポインタ
unsigned int	row	配列の行数
unsigned int	column	配列の列数
unsigned int	size	行列の総要素数

### 機能

#### H\_MATRIX create\_matrix

- 概要  
matrixハンドラーを生成する。  
matrixの内部配列を作成し、ハンドラーに割り当てる。  
行列の要素数を計算し変数に保存する。
- 引数

型	引数名	概要
unsigned int	row_size	行列の行数
unsinged int	column_size	行列の列数

- 戻り値 H\_MATRIX  
正常：matrixハンドラー  
異常：NULL

#### int delete\_matrix

- 概要  
matrixハンドラーを削除する。  
割り当てられた配列を解放する。
- 引数

型	引数名	概要
H_MATRIX	hMatrix	削除する対象の行列ハンドラー

・戻り値 int

正常：0

異常：-1

## int add\_matrix

・概要

2つのmatrixの和を計算し計算結果を格納する。

※ 2つの入力matrixと1つの出力matrixの形(行数、列数)はすべて同じであること。

・引数

型	引数名	概要
H_MATRIX	hMatrix_IN1	入力の行列ハンドラー
H_MATRIX	hMatrix_IN2	入力の行列ハンドラー
H_MATRIX	hMatrix_OUT	出力先の行列ハンドラー

・戻り値 int

正常：0

異常：-1

## int product\_matrix

・概要

2つのmatrixの内積を計算し計算結果を格納する。

※入力1の列数=入力2の行数であり、出力の行数=入力 1 行数、出力の列数=入力 2 の列数であること。

・引数

型	引数名	概要
H_MATRIX	hMatrix_IN1	入力の行列ハンドラー
H_MATRIX	hMatrix_IN2	入力の行列ハンドラー
H_MATRIX	hMatrix_OUT	出力先の行列ハンドラー

・ 戻り値 int

正常 : 0

異常 : -1

**int print\_matrix**

・ 概要

matrixの要素を表示する。

・ 引数

型	引数名	概要
H_MATRIX	hMatrix_IN1	入力の行列ハンドラー

・ 戻り値 int

正常 : 0

異常 : -1

**固定値**

**Layerタイプ(LT\_\*)**

値	名前	概要
1	LT_ReLU	ReLUレイヤー
2	LT_Sigmoid	Sigmoidレイヤー
3	LT_Affine	Affineレイヤー
4	LT_Softmax	Softmaxレイヤー
5	LT_SoftmaxWithLoss	Sofmax-with-Lossレイヤー