数理科学特論Ａ～画像数学～　第１１回

シリーズ４：ニューラルネットワークと最適化問題

(2) 画像フィルタとニューラルネットワーク

今日は，ニューラルネットワークと画像処理の関連についての話題で，画像フィルタとニューラルネットワークとの関連，画像フィルタのニューラルネットワークによる最適化について，私自身の学生時代の研究をもとに説明します．

加重メディアンフィルタとその最適化の問題

メディアンフィルタについては，モルフォロジーのところでもすでに説明しましたが，入力される画像の各画素について，その画素およびその近傍の画素値を使ってメディアン（中央値）を計算し，その結果を出力される画像のその位置の画素値とします．近傍の形・大きさはどの画素においても同じで，これをウィンドウ(window)とよびます．

加重メディアンフィルタはメディアンフィルタを拡張したもので，ウィンドウの各位置に「加重」(weight coefficient)というパラメータを設定します．ウィンドウ内の画素のメディアンを計算するには，ウィンドウ内の画素を大きさの順に並べ替え，中央の順位の値を求める必要があります．このとき，加重メディアンフィルタでは，各画素を対応する加重の数だけコピーした数列を作ってから大きさの順に並べ替え，中央の順位の値をフィルタの出力とします（図１）．

メディアンフィルタ，加重メディアンフィルタの特徴は，インパルス性ノイズの除去能力です．インパルス性ノイズとは，周辺の画素値から突出して大きい・小さい画素値が生じるようなノイズのことです．ウィンドウ内の平均値はこのような突出した画素値に大きく影響を受けるので，平均値フィルタではインパルス性ノイズを除くことができません．ところが，ウィンドウ内のメディアンは，画素値の大きさの順位にもとづいてウィンドウ内の画素から画素値を選ぶので，このような突出した値の影響を受けません．また，「ウィンドウ内の画素から新たな中間の画素値を生じるのではなく，ウィンドウ内の画素から画素値を選ぶ」ため，出力される画像にぼけを生じないという利点もあります．

加重メディアンフィルタはメディアンフィルタにさらに設計の自由度を与えたものです．大きな加重を与えられた位置の画素値は，ウィンドウ内の画素値を大きさの順に並べ替えた数列の中で多数を占めることになるので，メディアンとして出力されやすくなります．そこで，たとえばウィンドウの中央の画素に大きな加重を与えると，入力の画素値がそのまま変化されずに出力されやすくなります．また，たとえば縦方向の画素に大きな加重を与えると，入力画像の縦方向の線構造が保存されやすくなります．

しかし，線形フィルタリングがコンヴォリューションとみなせるので周波数空間で取り扱えるのに

加重

1	*1*	*1*
1	*3*	*1*
1	*1*	*1*

画素値

5	4	6
9	3	1
0	5	1

図１．加重メディアンフィルタ

画素値を加重の数だけ重複してならべる
5, 4, 6, 9, ***3, 3, 3,*** 1, 0, 5, 1

大きさの順にならべかえ，まん中の値を出力する
0, 1, 1, 3, 3, 3, 4, 5, 5, 6, 9 →出力3

対し, これらのメディアン系フィルタはそのような解析・設計手法がありません. 一般的な設計手法がないことは, 加重メディアンフィルタの大きな弱点です. すなわち, 加重の決めかたがわからなくては, いくら自由度のある一般的なフィルタといっても使いようがありません. これは, 加重メディアンフィルタだけの問題ではなく, 非線形フィルタが線形フィルタを上回る能力を持つといわれながらなかなか実用になりにくかった原因でもあります. そこで, その解決策として, 次節に述べるように加重メディアンフィルタをニューラルネットワークで表現して, その学習能力を使って入出力例からフィルタを最適化することにします.

加重メディアンフィルタのニューラルネットワークによる表現と学習による最適化

画像フィルタをニューラルネットワークで表現するために, 画素をニューロンで表すことにします. ところが, ニューロンは基本的に0, 1の２値をとるものとなっています. したがって, そのままでは画素が0, 1の２値である２値画像しか扱うことができません. ところが, メディアン系フィルタは次に述べる方法で多値画像のフィルタリングを２値画像のフィルタリングで表現することができます.

各画素が $0 \sim k-1$ の k 値をとる画像を扱うとします. このとき, $1 \le i \le k-1$ である i について, 「多値画像の対応する画素値が i 以上なら1, それ以外なら0」として $k-1$ 枚の２値画像を作ります. これらの２値画像をそれぞれメディアンフィルタリングして, 出力された２値画像をすべて重ね合わせて画素ごとに合計すると, もとの多値画像をメディアンフィルタリングしたのと同じ結果が得られます（図２）. この方法をしきい値分解(threshold decomposition)といいます.

なぜこのようなことができるのかは, 図３を見るとわかります. 図３の棒グラフは, ある場所のウィンドウに入った画素値を大きさの順にならべかえたものです. それに交差する横線は, 上の手順で作られた２値画像をつくるときのしきい値を表します. しきい値 i が「まん中の順位の画素値」より大きいときは, i 以上である画素, すなわちこのしきい値でつくられる２値画像でいえば1である画素の個数はウィンドウ内の画素の個数の半分より少なくなります. また, しきい値 i が「まん中の順位の画素値」以下のときは, このしきい値でつくられる２値画像で１である画素の個数はウィンドウ内の画素の個数の半分以上です.

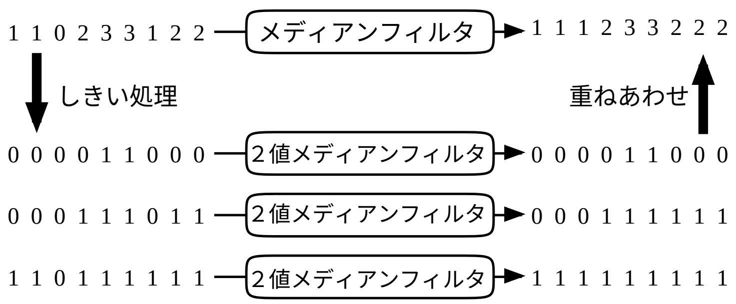

図２. しきい値分解

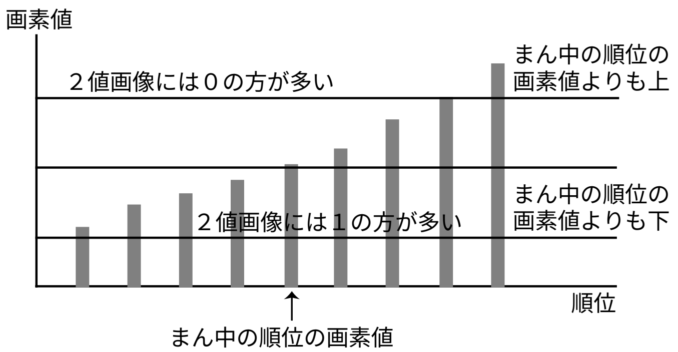

図３.
なぜしきい値分解
ができるのか？

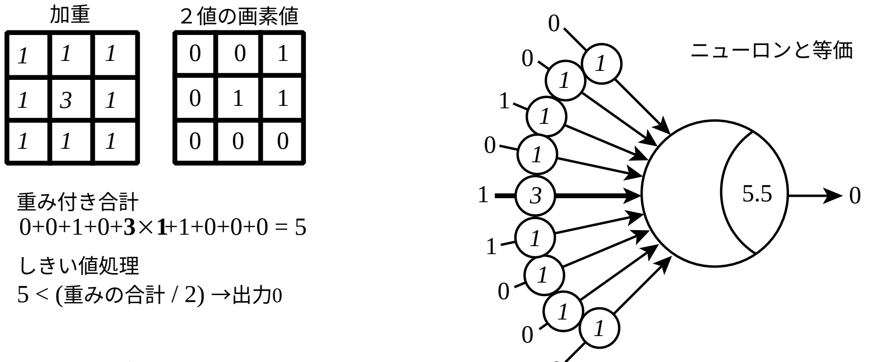

図４．２値加重メディアンフィルタとニューロン

ところで，２値画像に対するメディアンフィルタとは，結局「ウィンドウ内の画素のうち，1のほうが多ければ出力は1，そうでなければ出力は0」という「多数決」になりますから，上のことから．しきい値iが「まん中の順位の画素値」より大きいときは，このしきい値でつくられる２値画像に対するメディアンフィルタの出力は0，しきい値iが「まん中の順位の画素値」以下のときは，このしきい値でつくられる２値画像に対するメディアンフィルタの出力は1となります．したがって，これらの出力を足しあわせると，その値は「まん中の順位の画素値」すなわち多値画像に対するメディアンフィルタの出力と同じになります．加重メディアンフィルタの場合も，図３の棒グラフをつくるときに画素を対応する加重の数だけコピーするだけで，ほぼ同様です．

さて，「ウィンドウ内の画素のうち，1のほうが多ければ出力は1，そうでなければ出力は0」という「多数決」は，言い方を変えると「ウィンドウ内の画素をすべて合計して，画素の個数の半分よりも大きければ出力は1，そうでなければ出力は0」というしきい値処理ということができます．また，加重メディアンフィルタの場合も，図４のように，「各画素を対応する加重の数だけコピーしてから大きさの順に並べ替えてまん中の順位の画素を出力する」操作は，２値画像においては「ウィンドウ内の画素各々対応する加重をかけてからすべて合計して，重みの合計の半分よりも大きければ出力は1，そうでなければ出力は0」という重み付きしきい値処理になります．

これは，前回の講義で説明したニューロンの行う処理と全く同じです．すなわち，しきい値分解を行って加重メディアンフィルタを２値フィルタにおきかえることにより，フィルタのある１画素での操作をニューロンの動作で表現できたことになります．すなわち，加重メディアンフィルタが画像v^1の位置$\boldsymbol{Q}$の画素を処理するとき．ウィンドウ（これをDとします）内の相対画素位置を$\boldsymbol{q}$，その位置に対応する重みを$w(\boldsymbol{q})$とします．すると，２値の加重メディアンフィルタの処理は，画素値$v^1(\boldsymbol{Q}+\boldsymbol{q})$に重み$w(\boldsymbol{q})$を掛けて，それを$D$内の全ての$\boldsymbol{q}$について合計したものを，　重みの合計の半分で０または１にしきい値処理するということができますから，出力画像を$v^2(\boldsymbol{Q})$とすると，２値の加重メディアンフィルタの処理は

$$v^2(\boldsymbol{Q}) = H\left[\sum_{\boldsymbol{q}\in D} w(\boldsymbol{q})v^1(\boldsymbol{Q}+\boldsymbol{q}) - \frac{1}{2}\sum_{\boldsymbol{q}\in D} w(\boldsymbol{q})\right]$$
$$H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad (1)$$

と書くことができます．ここで，Dの外ではw($\boldsymbol{q}$)=0とし，$\boldsymbol{P} = \boldsymbol{Q} + \boldsymbol{q}$とすると，(1)式は

$$v^2(\boldsymbol{Q}) = H\left[\sum_{\forall \boldsymbol{P}} w(\boldsymbol{P}-\boldsymbol{Q})v^1(\boldsymbol{P}) - \frac{1}{2}\sum_{\forall \boldsymbol{P}} w(\boldsymbol{P}-\boldsymbol{Q})\right] \qquad (2)$$

と書き換えられます．(2)式での操作は，ニューロン$v^1(\boldsymbol{P})$の状態と結合重み $w(\boldsymbol{P}-\boldsymbol{Q})$によってニュー

ロン$v^2(\boldsymbol{Q})$の状態が変化される過程と考えることができます．この操作は，２層の階層型ニューラルネットワークと等価です．図５にこのネットワークのモデルを示します．第１層は入力２値画像を表し，第２層は処理後画像を表します．それぞれのニューロンは，画素に１対１に対応します．

(2)式で，$w(\boldsymbol{P}-\boldsymbol{Q})$は$\boldsymbol{P}$と$\boldsymbol{Q}$の相対位置によってのみ決まります．このことは，このモデルが移動不変(translation-invariant)な結合重みを持っていることを示しています．ネットワークの結合重み$w(\boldsymbol{P}-\boldsymbol{Q})$はフィルタの加重$w(\boldsymbol{q})$と同じものであるから，ネットワークの学習を行うことによって加重の最適化が実現されます．

結合重み$w(\boldsymbol{P}-\boldsymbol{Q})$の学習の目標は，フィルタの出力を理想出力にできるだけ近づけることです．ここでは，前回説明したδルールを応用して学習を行います．学習の繰り返しの数をtで表します．入力される雑音の重畳した画像を$v^{1,0}$とし，理想出力を$v^{2,I}$とする．また，学習途中のネットワークの出力を$v^{2,t}$とし，[0,1]の連続値をとるものとします．このとき，学習過程は修正項$\Delta^t(\boldsymbol{P}-\boldsymbol{Q})$を用いて次のように表されます．

$$w^{t+1}(\boldsymbol{P}-\boldsymbol{Q}) = w^t(\boldsymbol{P}-\boldsymbol{Q}) + \Delta^t(\boldsymbol{P}-\boldsymbol{Q}) \tag{3}$$

修正項は次のように定められます．

$$\Delta^t(\boldsymbol{P}-\boldsymbol{Q}) = \frac{1}{N}\sum_{\forall \boldsymbol{P}} \varepsilon\left(v^{2,I}(\boldsymbol{Q}) - v^{2,t}(\boldsymbol{Q})\right)v^{1,0}(\boldsymbol{P}) \tag{4}$$

ここでNは画像中の全画素数で，εは正の学習係数です．移動不変な重みを修正するために，修正項は全画面に対して平均されています．なお，学習過程では，ニューロンの状態を連続値にするため，しきい関数のかわりにシグモイド関数（前回の講義の(3)式）を用いています．

以下，この手法を適用して加重メディアンフィルタを最適化した例を示します．この実験では，６４×６４画素の２値画像を用いました．図６に原画像を示し，図７に生起確率5％のごましお雑音（salt-pepper noise，画素がある確率で白または黒におきかわる雑音）が重畳した画像を示します．これらの画像が理想的入出力例で，図７の画像が入力されたとき図６の画像が出力されるようにフィルタを最適化します．計算能力の制限から，第２層の１画素に接続している第１層の画素は9×9画素の範囲に限定しました．結合重みの初期値は$w(0)$のみ１とし，他は０としました．また，シグモイド関数のパラメータ$\varepsilon = 0.5$とし，$\lambda = \lambda + 0.01t$として学習が進むほど本来のメディアンフィルタを表現するためのしきい関数に近付くようにしました．2000回学習した後の重みを図８に示します．加重メディアンフィルタの重みは正しか許されないので，重みが学習途中で負になった場合は０としました．多値画像に対する加重メディアンフィルタの定義に合うように，図８の重みを整数化したものが図９です．この重みを用いた加重メディアンフィルタを，別の雑音が重畳した画像（図１０）に適用しました．この画像には図６と同じ統計的性質を持つ雑音が重畳しています．この画像を学習後のフィルタで処理したのが図１１です．また，同じ画像を従来の3×3のウィンドウを持つメディアンフィルタで処理したのが図１２です．図９の原画像を図１３に示します．これらの結果から，本手法が学習によって最適化したフィルタは，類似の画像の処理に高い性能を示すことがわかります．

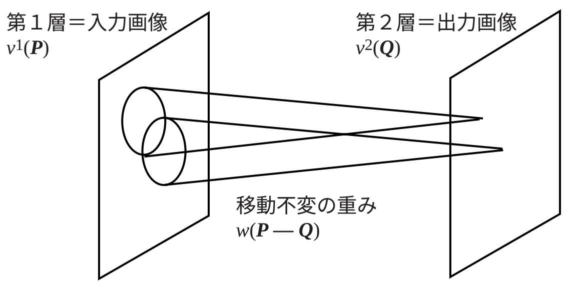

図５．加重メディアンフィルタを表すネットワーク

今日の参考文献

浅野　晃，博士学位論文，大阪大学(1992).

A. Asano et al., "Optimization of the weighted median filter by learning," *Opt. Lett.*,**16,** 3, 168-170 (1991).

図６．理想的入出力例における出力画像

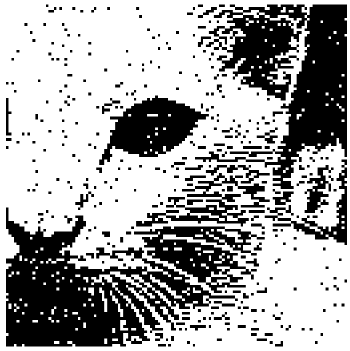

図７．理想的入出力例における入力画像

0.00	0.00	0.00	0.00	0.00	1.83	0.02	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.42	0.00	0.00	0.00
0.36	0.61	0.00	0.00	0.00	000	0.00	0.00	0.00
0.00	0.00	0.34	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.83	1.70	13.46	36.01	14.31	3.39	0.00	0.28
0.00	0.00	0.00	0.00	1.35	0.80	0.00	0.00	0.00
0.40	0.00	0.00	0.00	1.39	0.00	0.00	0.00	0.00
0.29	0.00	0.00	2.39	0.00	0.00	0.55	0.00	0.00
0.00	0.19	0.00	0.22	0.00	0.00	0.00	0.00	0.00

図８．学習によって得られた重み

0	0	0	0	0	4	0	0	0
0	0	0	0	0	3	0	0	0
1	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	3	27	72	29	7	0	1
0	0	0	0	3	2	0	0	0
1	0	0	0	4	0	0	0	0
1	0	0	5	0	0	1	0	0
0	0	0	0	0	0	0	0	0

図９．整数化された重み

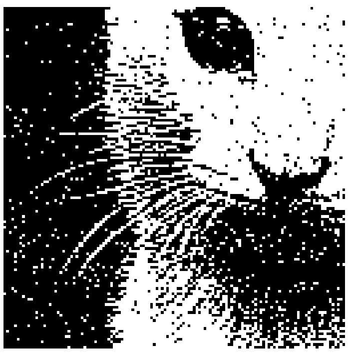

図10．テスト用入力画像

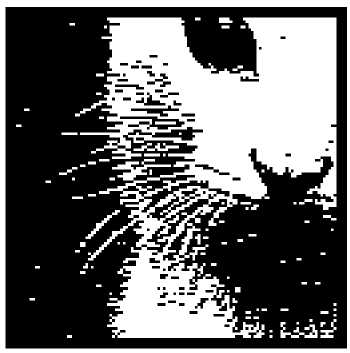

図11．最適化されたフィルタの出力

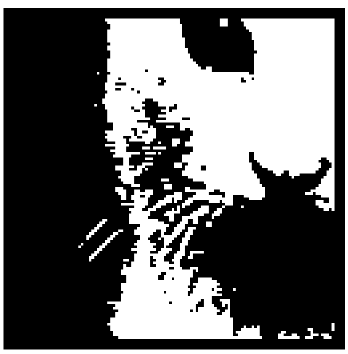

図12．通常のメディアンフィルタの出力

図13．図10 の画像のノイズ重畳前の画像