コンパイル日：2020 年 1 月 9 日
慶應義塾大学理工学部物理学科
岡崎健人

畳み込みネットワーク

1　畳み込み処理

1.1　畳み込み処理の定義

実数を引数にとる実関数 $I(t)$ と重みの関数 $K(t)$ を用いて

$$S(t) = \int I(\tau)K(t-\tau)\,\mathrm{d}\tau$$

という関数 $S(t)$ を生成する。この処理を畳み込み（convolution）とよび、$S(t) = (I * K)(t)$ のように表記する。これらの関数の引数が離散値であるならば、離散畳み込みを

$$S(t) = (I * K)(t) = \sum_{\tau} I(\tau)K(t-\tau)$$

と定義する。畳み込みネットワークの文脈では I を入力（input）、K をカーネル（kernel）またはフィルター（filter）、S を特徴マップ（feature map）とよばれることが多い。

２変数に対する離散畳み込みは

$$S(i,j) = \sum_{m}\sum_{n} I(i-m, j-n)K(m,n)$$

となる。ただし畳み込みの可換性 $(I * K)(i,j) = (K * I)(i,j)$ が用いられている。一方で、ニューラルネットワークの実装においては相互相関（cross-correlation）とよばれる次の量も畳み込みとよび、よく用いられている。

$$S(i,j) = \sum_{m}\sum_{n} I(i+m, j+n)K(m,n).$$

1.2　２次元畳み込み処理の例

２次元のデータに畳み込み処理を行う例を図 1 に示す。簡単のため入力 I はサイズが 6×6 の行列であるとし、その成分の値をグレースケールで視覚化してある。２次元のフィルター K は 3×3 の行列である。フィルターはその中心を決められるように、行数と列数ともに奇数であると便利である。まず図 1 中の青色の四角形で示したように、入力 I の左上の位置にフィルター K の成分が全て含まれるように重ねる。同じ位置にある成分同士を掛け合わせ、総和をとると 4 になるので、その値が特徴マップ S の左上の成分になる。次に図 1 中のオレンジ色の四角形で示したように、重ね合わせていたフィルター K を右に 1 つずらして同様の計算をする。入力 I の右端までこの操作を繰り返したら 1 行下がる。この操作を入力 I の右下に達するまで繰り返す。

より一般に、2 次元の入力 I のサイズが $N \times M$, ２次元のフィルター K のサイズが $(2k+l)\times(2l+1)$ であるとする。ただし $N,\ M,\ k,\ l$ は 0 を含まない自然数である。(i,j) 成分の値をそれぞれ $I(i,j),\ K(i,\ j)$ としたとき、２次元畳み込み処理によって得られる値 $S(i,j)$ は

$$S(i,j) = \sum_{n=0}^{2l}\sum_{m=0}^{2k} I(i+m, j+n)K(1+m, 1+n)$$

となる。i と j の範囲は $i=1,\ldots,M-2k,\ j=1,\ldots,N-2l$ で、特徴マップ S のサイズは $(M-2k)\times(N-2l)$ となる。ただし Python では配列の要素番号は 0 から始まるので、その形式に合わせるには総和の下端を $n=0$ から $n=-1$ のように書き換えればよい。

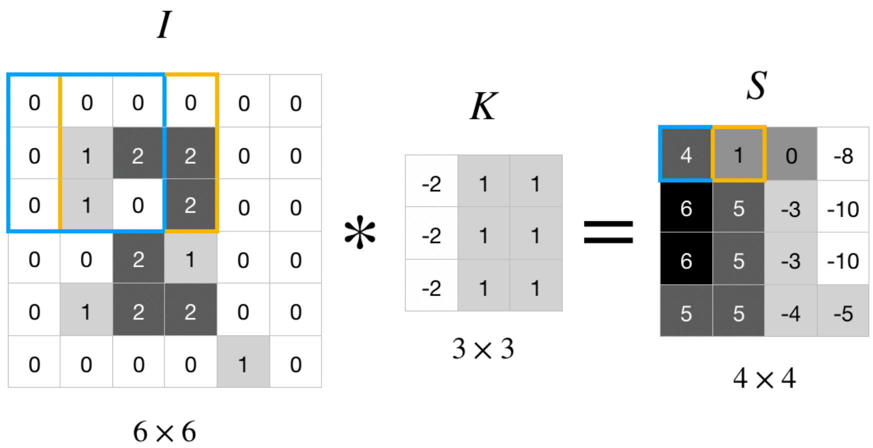

図 1　2 次元畳み込み処理の例

1.3　パディングとストライド