

自戒用
深層学習でよくみるネット
ワーク紹介

目的

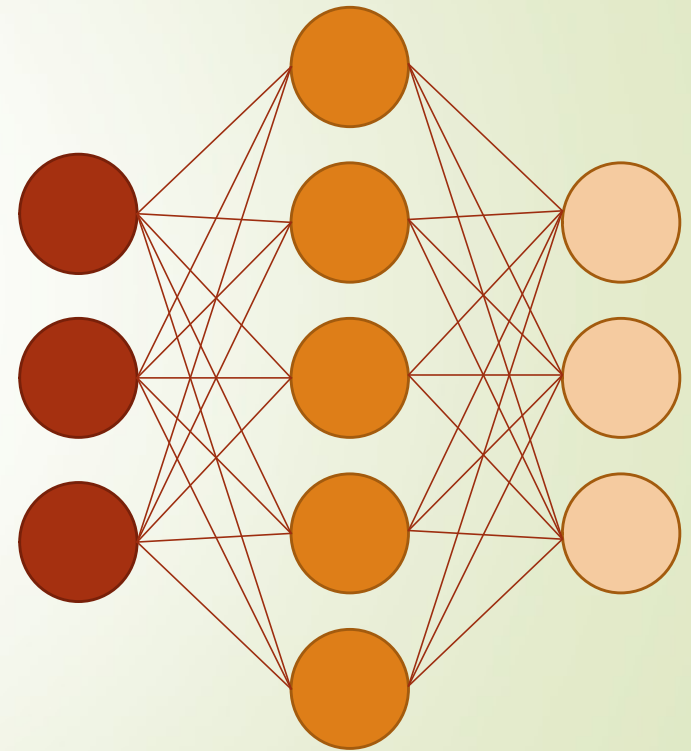
- 本スライドではこれからpytorch使う人も、Tensorflowを使う人も、kerasを使う人も、おそらく使うであろう**深層学習の関数**について軽く紹介
- 心底から仕組みを知りたい方は、
 - Good! : 論文を読む
 - Good! : 関連書籍を取り寄せる
 - Goodじゃないけど、わかりやすい説明があったりする：ネットで調べる

Linear関数

- いわゆる線形関数

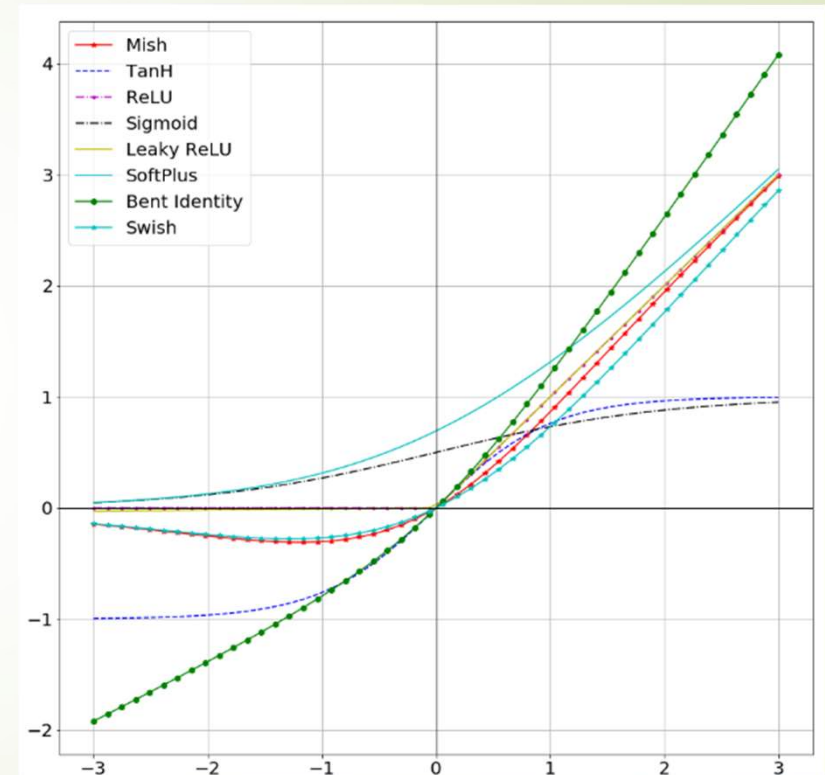
- $y_i = f(x_i, W) = \sum_k x_i w_{ik}$

- 1次元のフラットなデータをインプットし、1次元のアウトプットを線形変換を通じて返す



活性化関数

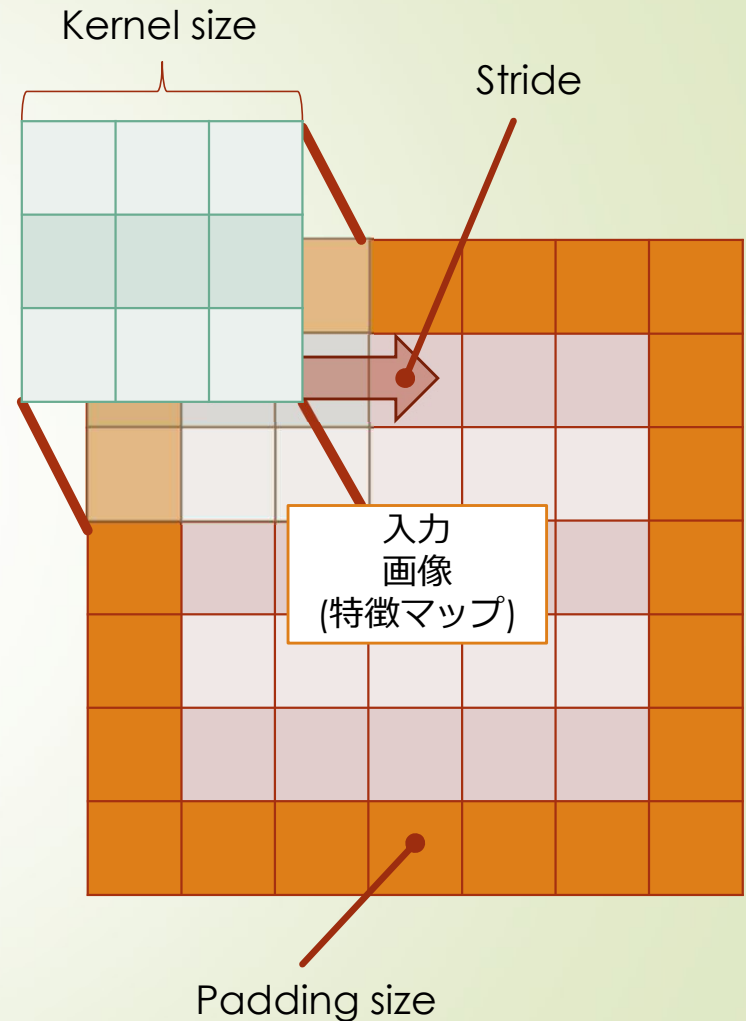
- 線形な関数を非線形に変換する関数
 - 線形な関数: 1次元関数, 2次元関数
 - 非線形な関数: sin関数, log関数
- 線形な関数は重ねても線形にしか変化できない
 - 複雑な表現ができない!
- Linear関数や畳み込み関数からの総和をインプットとし、変換を行う。
頻繁に使われている



<https://github.com/digantamisra98/Mish>
の図より引用

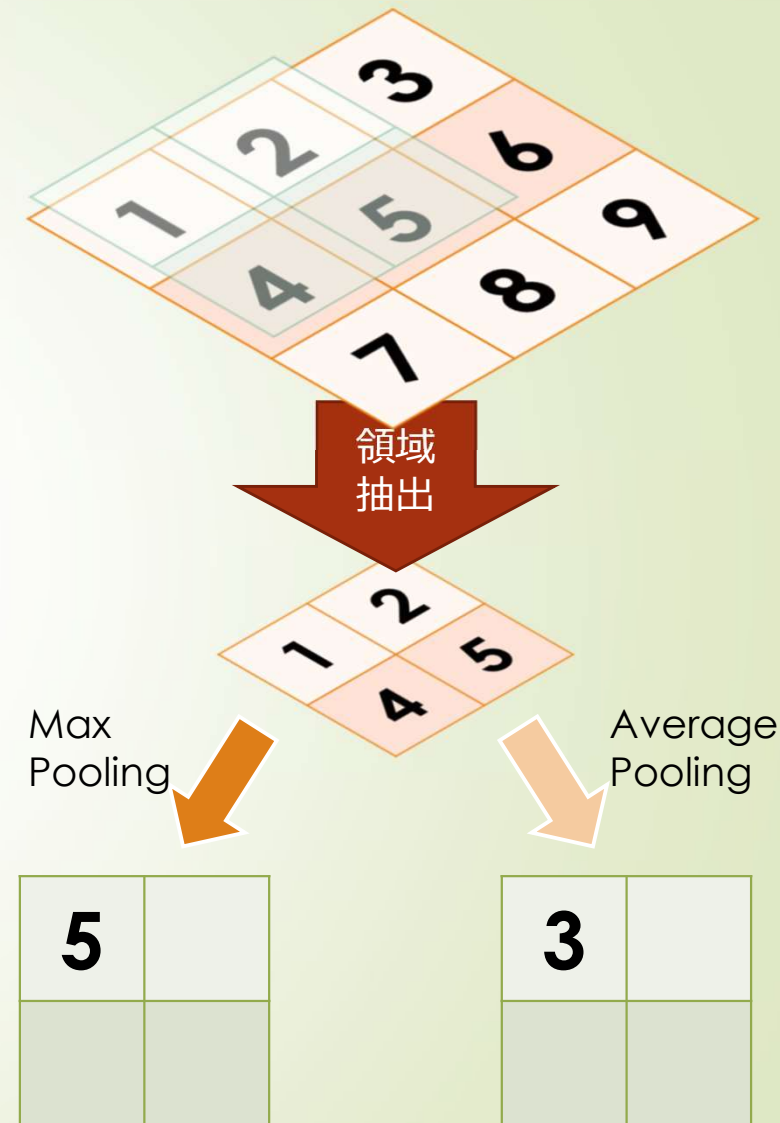
畳み込み関数

- 畳み込みを通じて特徴を抽出する関数
 - フィルタをもちいて、フィルタをずらしながら畳み込み計算を行う
 - 主なパラメータとして、
 - kernel size: フィルタの大きさ
 - stride: フィルタをどれくらい移動させるか
 - padding size: 入力の周辺を囲むピクセルのパディングサイズ
 - 入力の端も畳み込めるようになる
 - 出力のサイズを調整できる
- フラットなデータを扱うLinear関数と比べて、チャンネルごと空間を扱うので位置ずれなどに強い
 - 1次元畳み込み: 音声データ
 - 2次元畳み込み: 画像データ
 - 3次元畳み込み: CT画像、動画データ



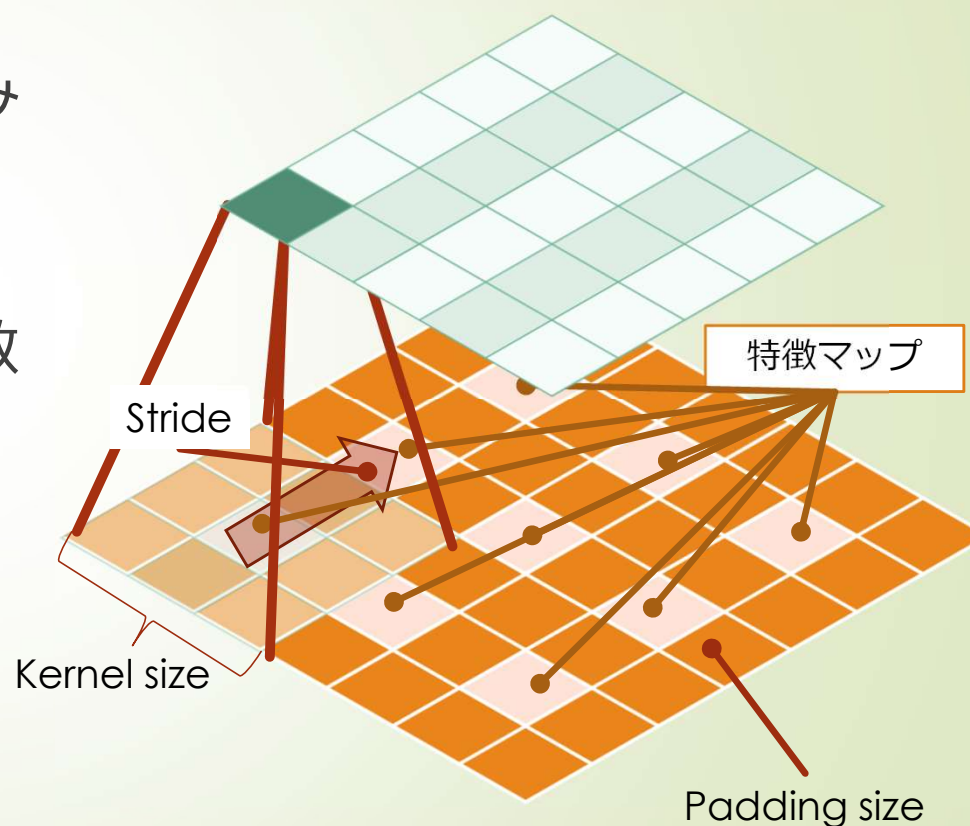
Pooling関数

- 入力された特徴量の強さをより強調することを目的とした関数
 - 使用するパラメータは主に"Kernel size"と"Stride"
- 主なものとして
 - Max Pooling: カーネルから抽出した領域内の最大値をとる
 - Average Pooling: 領域内の値の平均をとる



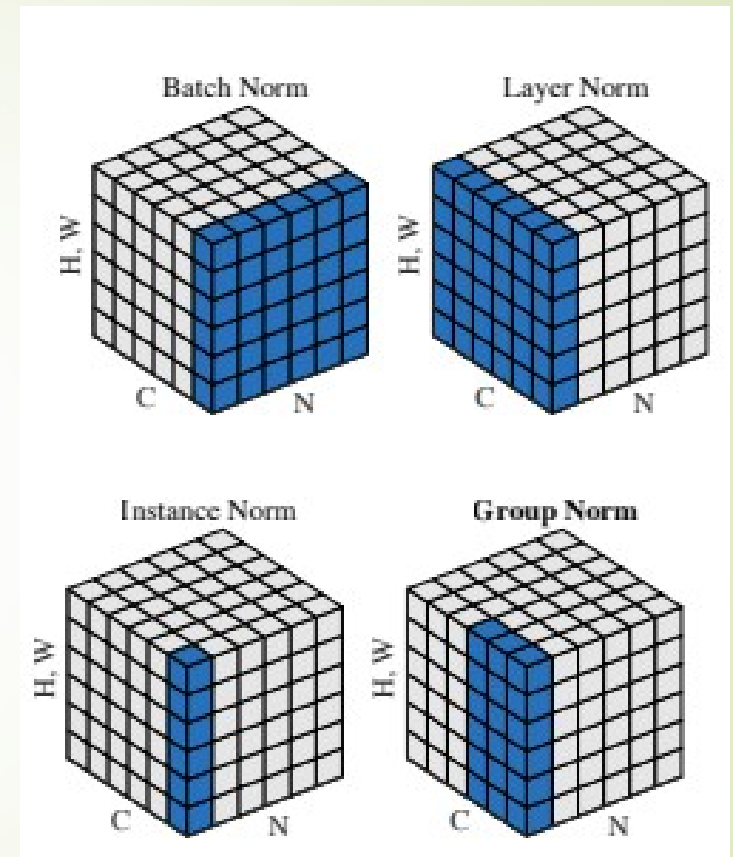
逆畳み込み関数

- もととなるインプット(特徴マップ)をパディング等で拡大してから畳み込む関数
 - 転置畳み込みともいったりする
- 使用するパラメータは畳み込み関数と大方同じ
 - Kernel size
 - Stride
 - Padding size



Normalization関数

- 畳み込みや逆畳み込みを通じて出力される特徴マップには、極端なばらつきもあり得る
 - ばらつきがあると勾配消失や過学習といった学習の障害に
- そんなばらつきを上手く調節するのがNormalization関数
 - ばらつきを上手く調整するためには？
 - 平均=0, 分散=1となるように正規化する
 - 学習時のミニバッチごとに平均=0, 分散=1となるように正規化
- 学習の安定性を高める効果があるとか



Normalization関数の種類
(引用: Yuming Wu et al,
"Group Normalization" (2018).
Figure2より)