Deep Learning 輪読会 2017

第14章　自己符号化器

2017.12.11

松尾研究室
特任研究員　村上　遥

構成

第14章　自己符号化器

14.1 不完備な自己符号化器

14.2 正則化付き自己符号化器

14.3 表現力、レイヤーサイズ、および深さ

14.4 確率的な符号化器と復号化器

14.5 雑音除去自己符号化器

14.6 自己符号化器による多様体学習

14.7 縮小自己符号化器

14.8 予測スパース分解

14.9 自己符号化器の応用

14章　自己符号化器　の章について

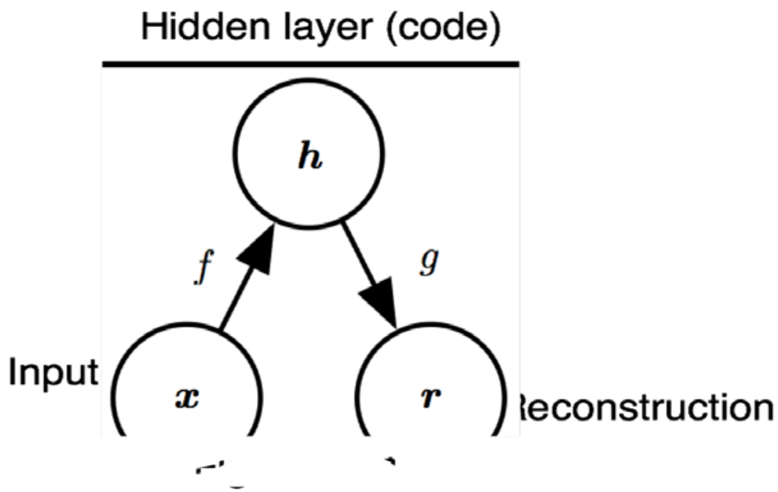
- この章は、14.1～14.4で例も交えながら自己符号化器の話が進みますが、
- 14.5～14.8でもう一度同じ項目が出てきます。
- 14.9は応用例。

- 生成モデルを知っている体で語られています。
- ページ順に話すと話がややこしくなるので順番を変えています
- 原文自体の構成がうろうろしているのと、細かい話がとても多くて載せきればいいので、読み直す時は項目で辞書的に読んだ方がいいです。

第14章　自己符号化器

- 入力をその出力にコピーするように学習させたニューラルネット
- 符号化器関数h=f(x)と復号化器r=g(h)から成る

- 用途：
- 従来ー　次元削減や特徴量の学習、最近ー　生成モデル
- 順伝播型ネットワークの特別な場合、普通と異なり、再循環（入力と再構成された入力の活性化の比較に基づく学習、機械学習には殆ど応用されていない）が可能



14.1 不完備な自己符号化器

- 入力を出力にコピーする意味
  - 実際は完コピはしないように設計
  - 優先順位を決めてコピーする過程で隠れ層hに有用な性質を持たせる

- hの次元をxより小さくなるように制限（不完備）
  - 訓練データの最も顕著な特徴量を捉えるようになる

- 符号化器と復号化器の容量が大きすぎるとただのコピータスクになる

14.2 正則化付き自己符号化器

- 符号化器および復号化器の符号次元および容量を,モデル化される分布の複雑さに基づいて選ぶことで、どのような構造の自己符号化器でもうまく学習させる
- 入力を出力にコピーする以外の性質もモデルが保持できるような損失関数を利用
- 非線形や過完備でOK
- 例：
  - 14.2.1 & (14.8)　スパース自己符号化器
  - 14.2.2 & 14.5　雑音除去自己符号化器
  - 14.2.3 & 14.7　縮小自己符号化器
  - 隠れ層でドロップアウトを行う自己符号化器

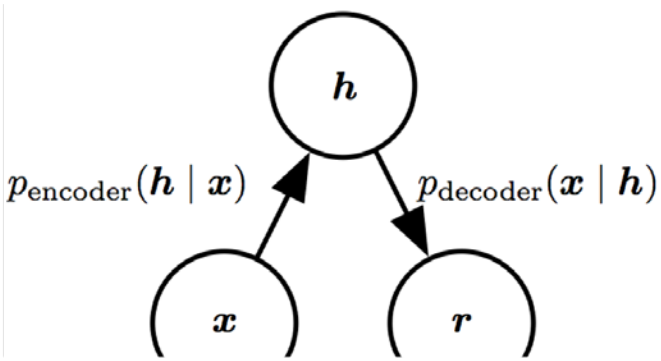
14.3 表現力,レイヤーサイズ、および深さ

- 自己符号化器は単層の符号化器と復号化器だけで学習することが多い
- 深さの利点：自己符号化器は順伝播型ネットワークなので、利点が共通
  - 十分な隠れユニットがあれば、任意の制度で関数の近似ができる
  - 計算コストと訓練データの量を指数関数的に減らせる

- 深層自己符号化器
  - 線形もしくは浅い自己符号化器に比べ、圧縮に優れることが実験的に示されている
  - 一般的な訓練方法は、浅い自己符号化器を積み重ねて学習させ、深層構造を貪欲に事前学習させるもの

14.4 確率的な符号化器と復号化器

- 自己符号化器は順伝播型ネットワークなので、損失関数と出力ユニットの種類は６章で紹介されたものと共通
- 順伝播型ネットワークの具体的な設計方針：
  - 出力分布p(y|x)を定義し、負の対数尤度-log*p*(y|x)を最小化
  - xは入力かつ目標、yはクラスラベルなどの目的となるベクトル
- 自己符号化器では：
  - 隠れ符号hが与えられると、復号化器が条件付き分布$p_{decoder}$(x|h)を出力すると見なせる
  - その後、-log$p_{decoder}$(x|h)の最小化によって学習させる



14.2.1 スパース自己符号化器

- 訓練基準として再構成誤差に加え符号層 h でのスパース性ペナルティ項 Ω(h) が含まれている自己符号化器

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h}).$$

復号化器の出力

- 分類などに利用する特徴量の学習に使われる
- Ω(h) は順伝播型ネットワークに追加された単なるデータ依存の正則化項

14.2.2 & 14.5 雑音除去自己符号化器 (DAE)

ディノイジングオートエンコーダー

- 破損したデータ点を入力として受け取り、元のデータ点を出力として予測するように訓練された自己符号化器
- 従来の自己符号化器は$L(x,g(f(x)))$を最小化
  - xから離れたg(f(x))にペナルティを課す損失関数
  - 容量が十分なら$g \circ f$に恒等関数の学習を促す
- DAEでは

$$L(\boldsymbol{x}, g(f(\tilde{\boldsymbol{x}}))).$$ を最小化

  - $\tilde{\boldsymbol{x}}$ は雑音により破損したxのコピー
  - DAEは入力のコピーではなく、破損を元に戻さなければいけない
  - 恒等関数の学習を妨げるようになっていれば、過完備で高容量のモデルでも自己符号化器として使用できる

14.5 雑音除去自己符号化器（DAE）

- 訓練手順：
  - xと $\tilde{\boldsymbol{x}}$ をサンプリング
  - (x, $\tilde{\boldsymbol{x}}$) を訓練事例として用い,自己符号化器の再構成分布 $p_{reconstruct}(x \mid \tilde{\boldsymbol{x}}) = p_{decoder}(x \mid h)$ を推定する
  - 勾配に基づく近似最小化を負の対数尤度$-\log p_{decoder}(x \mid h)$ に対して実行すればよい

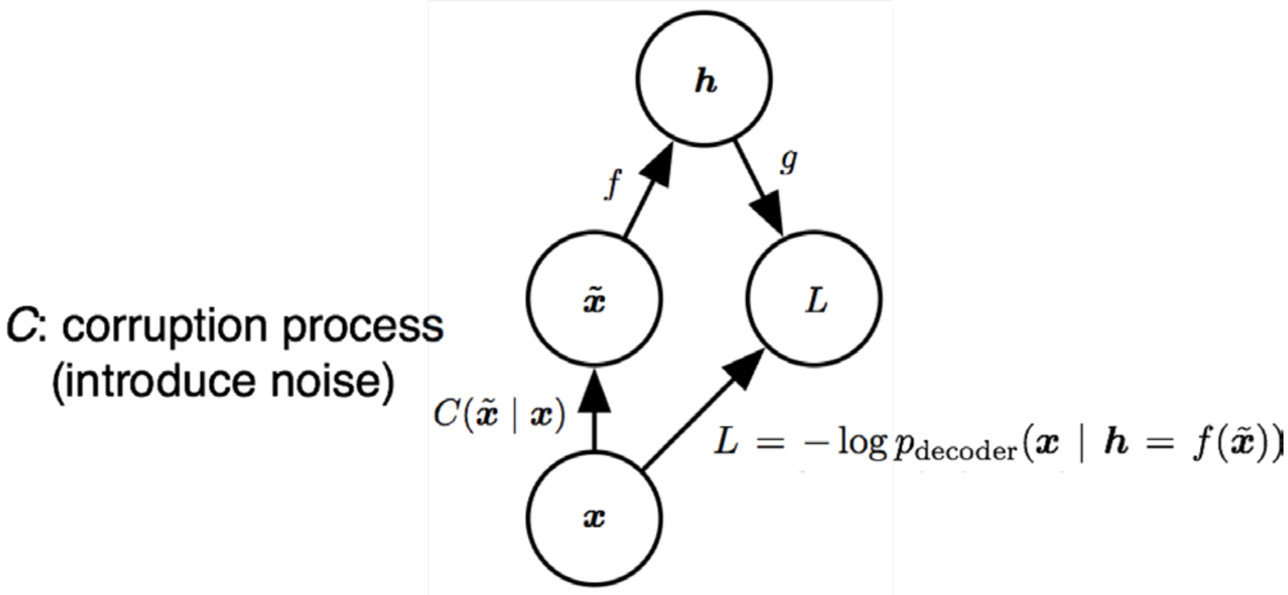


Figure 14.3

14.5.1　スコアの推定

- スコアマッチング：最尤法の代替手段　　(18.4,5、20.5.1節に詳細)
  - すべての訓練点xでモデルが訓練集合の経験分布と同じスコア

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}).$$

  - を持つようにすることで、確率分布の一致推定量を与える。
  - $\log p_{\mathrm{data}}$の勾配場の学習が,p_{data} 自体の構造を学習する方法の 1 つ

- DAEの非常に重要な性質：
  - 自己符号化器が訓練集合の経験分布の
  - スコアを推定するベクトル場を学習すること
- 訓練手順
  - $\tilde{\boldsymbol{x}}$ をxに写像するように訓練される
  - $\|g(f(\tilde{\boldsymbol{x}})) - \boldsymbol{x}\|^2$ の平均値を最小化するように訓練し
  - 結果的にベクトル場 g(f(x)) – x （緑↓）を学習

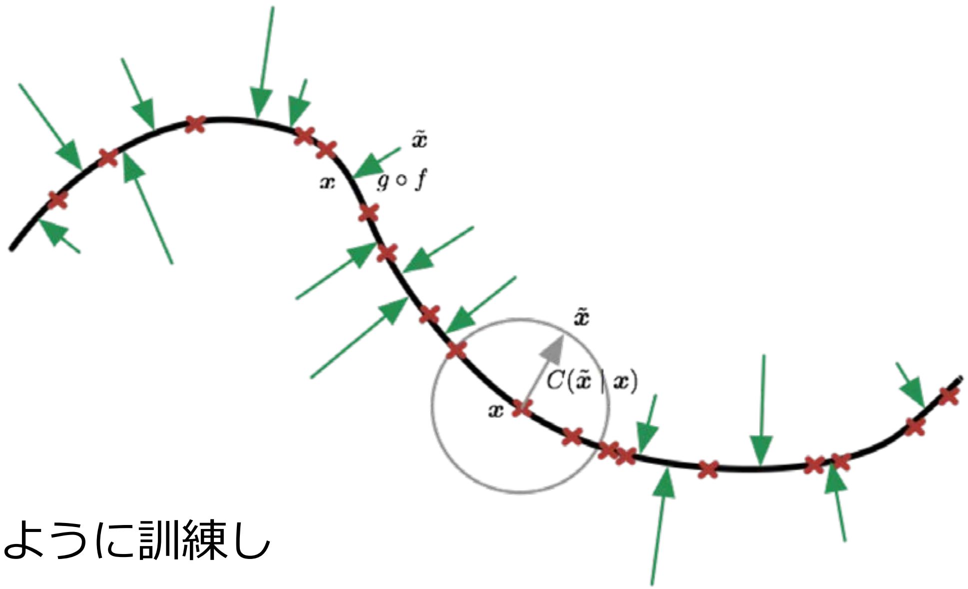


Figure 14.4

14.5.1.1　歴史的観点

- DAEは今は「単なる雑音除去を学習させた MLP (1987)」ではない
- 『雑音除去の学習の副作用として,良好な内部表現を学習する(2008)』という意図

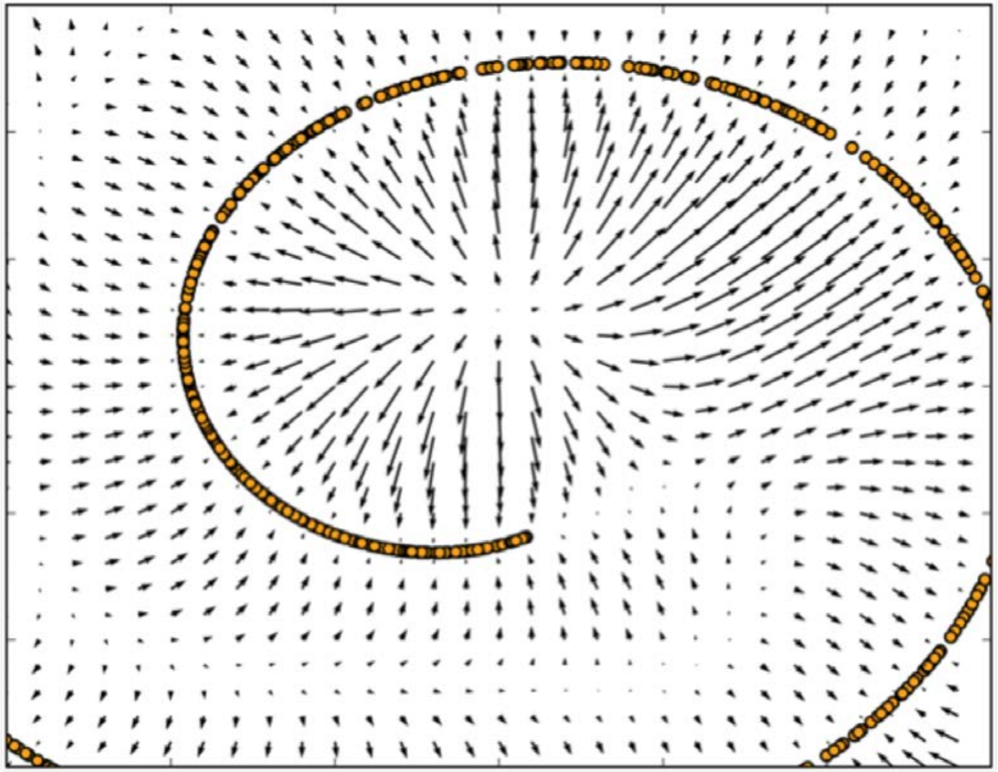


Figure 14.5

  – 湾曲した１次元多様体の周囲にある、DAEで学習されたベクトル場。
  – 低確率な点をより高確率な再構成へと写像し、確率が最大の場所の再構成は精密になるため矢印が短くなる

14.6　自己符号化器による多様体学習

- 自己符号化器は、低次元多様体やその多様体の小さい集合周辺にデータが集中しているという考え方を利用

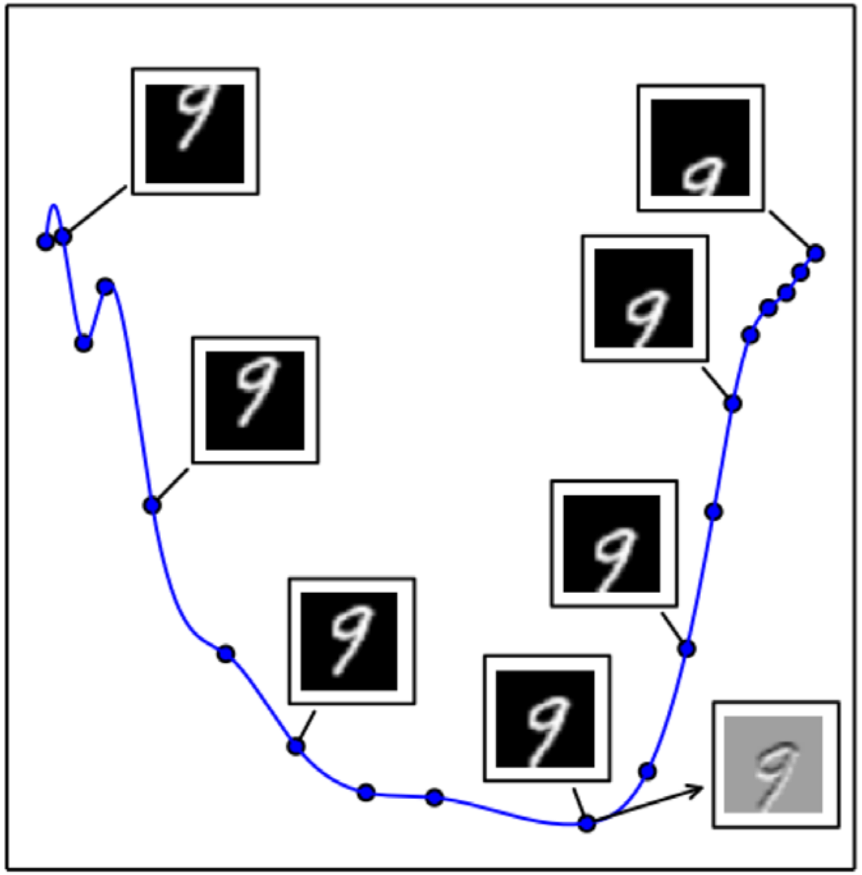


Figure 14.6

14.2.3 微分へのペナルティによる正則化

- スパース自己符号化器のようにペナルティ項Ωを訓練事例にのみ適用

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h}, \boldsymbol{x}).$$

- ただし $$\Omega(\boldsymbol{h}, \boldsymbol{x}) = \lambda \sum_i ||\nabla_{\boldsymbol{x}} h_i||^2.$$

- xの変化がわずかな時は、大きく変化しない関数をモデルに学習させる
- 訓練集合の経験分布に関する情報を捉える特徴量を学習する

- これを縮小自己符号化器（CAE：Contractive AE）という

14.7　縮小自己符号化器（CAE）

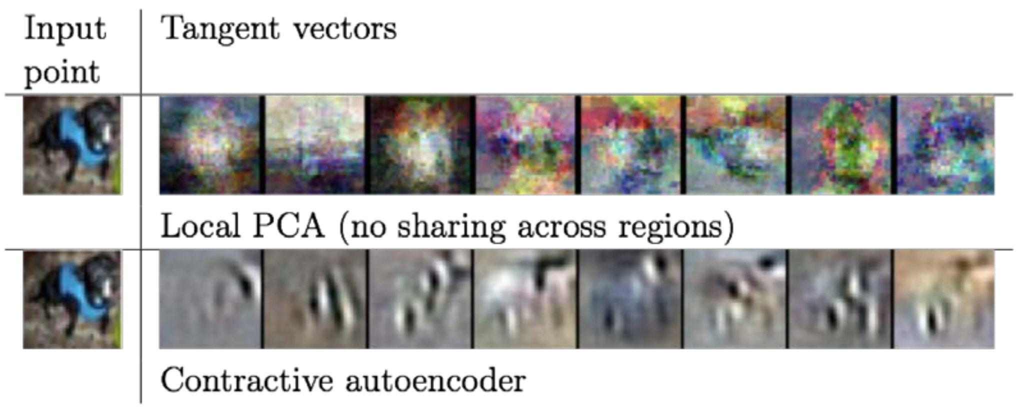
- 符号 h = f(x) に明示的な正則化

$$\Omega(\boldsymbol{h}) = \lambda \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_F^2 .$$

を導入してfの微分の最小化を図る

  - 符号化器関数に関する偏微分のヤコビ行列の二乗された要素の和

- 入力の近傍を出力のより小さな近傍に縮小していると考えられる
- ただし、局所的にしか縮小しない（xはすべてf(x)の近くに写像される）
- 目的：再構成誤差と縮小ペナルティ項Ω(h)のバランスを取ることで多様体を学習する
- 難点：隠れ層が複数にナルト計算コストが高くつく



推定された多様体の接ベクトルの例。
CAE の方が少ない訓練データでも、異なる位置間でのパラメータ共有を行なっているため正確な

14.8　予測スパース分解（PSD）

- スパース符号化とパラメトリック自己符号化器を組み合わせたモデル
- 画像や動画における物体認識のための、教師なし特徴量学習や、音声で使用
- 双方パラメトリックな符号化器f(x)と復号化器g(h)で構成されている
- 学習は $||\boldsymbol{x} - g(\boldsymbol{h})||^2 + \lambda|\boldsymbol{h}|_1 + \gamma||\boldsymbol{h} - f(\boldsymbol{x})||^2$ を最小化

- スパース符号化と同様、hに対する最小化⇔モデルパラメータに対する最小化で交互最適化を学習中にのみ行う

14.9 自己符号化器の応用

- 次元削減や情報検索のタスクに使われる
- 低次元表現は消費するメモリと実行時間が小さくなり、タスク性能改善に役立つ
- セマンティックハッシング：
  - 文章入力や画像に応用されている、次元削減とバイナリ化による情報検索手法
  - 低次元のバイナリの符号を生成するように次元削減アルゴリズムを訓練すれば、すべてのデータベースの登録事項を格納できる、バイナリ符号ベクトルを登録事項へと写像するハッシュテーブルが作れる。質問と同じバイナリ符号を持つすべてのデータベースの登録事項を返すことで情報検索ができる。
  - 質問が符号化してあるので、個々のビットを反転させれば少し似ていない登録事項も簡単に検索可

参考文献

- Deep Learning
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville
  - 日本語版
    https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-Ian-Goodfellow/dp/4048930621