知的情報処理特論　レポート

講義日：2015 年 4 月 23 日

作成者：T15M081 前田一磨

2.　ディープニューラルネットワーク

2.1　ディープニューラルネットワークの訓練

ディープニューラルネットワーク（DNNS）は、入力と出力の間に複数の隠されたユニットの層を持つ人工ニューラルネットワークモデルである。ヒントンらは、ディープモデルの学習目標である不適切な値を書き換えるための調整学習や Greedy Layer-wise の教師無し学習アルゴリズムを提案した [27]「ニューラルネットワークにおいてのデータの次元の低減化」。その後、マルテンスは、ディープネットワークの訓練のための 2 次元の最適化方法(ヘッセ行列を用いない最適化)を導入することによって新たなアプローチを提案した [28]「ヘッセ行列を用いない最適化によるディープラーニング」 。提案手法は、教師無し学習をすることなく、効率的に一般的なオプティマイザによってモデルを訓練した。ここでは、特徴ベクトルの自己組織化や時系列学習のために複数のオートエンコーダを最適化するマルテンスによって提案された学習方法を採用する。

2.2　ヘッセのない最適化

ヘッセ行列のないアルゴリズムは、よく知られている数値最適化技術であるニュートン法を用いて考案される。ニュートン法のような標準的な二次元最適化法は、勾配ベクトル演算 P による目的関数 F の反復更新パラメータと 学習パラメータ α を持つ$\theta_{n+1}=\theta_n+\alpha P_n$ としての更新パラメータ θ を考える。ニュートン法の核となるアイデアは、第二次までの各 θ 周辺の局地的に近似した F である二次方程式である。

$$M_{\theta_n}(\theta) \equiv f(\theta_n) + \nabla f(\theta_n)^T p_n + \frac{1}{2} p_n^T B_{\theta_n} p_n, \qquad \text{式(1)}$$

ここで、B_{θ_n} はθ_n での F の減衰ヘッセ行列である。H が無限大になることができる時、ヘッセ行列は$B_{\theta_n} = H(\theta_n) + \lambda I$ になるために再定義される。ここで λ≥0 は減衰パラメータであり、I は単位行列である。解決システムが$B_{\theta_N p_N} = -\nabla f(\theta_N)^T$である時、一般的なニュートン法は$M_{\theta_N}(\theta)$が N×N の行列$B_{\theta_N}$ を計算することによって最適化されて使用する。し

かし、この計算はひかえめなニューラルネットワークであってもNが大きいために非常に大規模になる状況である。この問題を克服するために、無限大ヘッセ行列の代わりにマルテンスによって開発されたヘッセ行列を用いない一種の方法を利用する。それは、半正定値ガウス･ニュートン曲率行列と、次の目的を最適化するための線形共役勾配（CG）アルゴリズムを利用する。ヘッセフリーという名前はCGは大規模で明確なヘッセ行列を必要としないことを示します。その代わりにヘッセ行列 H またはガウス-ニュートン行列Gと 勾配ベクトルP の 間の行列ベクトル積が用いられる。（具体的な実装の詳細については[28]、[30]「ヘッセ行列による高速かつ正確な乗算」と[31]「2次元最急降下法のための高速な曲率行列ベクトルの積」を参照してください。）

3. マルチモーダル時系列学習メカニズム

本稿では、次元圧縮による特徴抽出のためだけでなく、そのマルチモーダル時系列統合学習のためにもディープオートエンコーダを適用することを提案する。この研究で私たちの主な貢献は、私たちの提案したフレームワークが、クロスモーダルメモリ検索だけでなくその強力な汎化能力を利用し時系列予測器として機能することを証明することである。以下のサブセクションでは、まずオートエンコーダの基本的な仕組みについて解説し、その後そのマルチモーダル時系列学習とその機能についてどのように適用されるかを説明する。

3.1 特徴ベクトルの自己組織化

視覚的な画像や音声のスペクトルのような高次元の生の感覚入力は、小さな中央の層（すなわち、特徴抽出ネットワーク）との多層ネットワークによって低次元の特徴ベクトルに変換することができる[27]。この目的のために、ネットワークは次の式のように定義されている入出力マッピングと出力層での入力データを再構成することを目標に訓練されている。

$u_t = f(r_t)$ 式(2)

$\hat{r}_t = f^{-1}(u_t),$ 式(3)

ここで、r_t、u_t、$\hat{r}_t$はそれぞれ、生の入力データ、対応する特徴、再構成されたデータを表すベクトルである。関数f（.）とf^{-1}（.）はそれぞれ、ネットワークの入力層から隠れた中央層へと隠れた中央層から出力層への変換マッピングを表す。オートエンコーダは、入力層から隠れた中央層へのノード数を減少させることにより、入力の次元数を圧縮する。したがって、隠れた中央層のノード数は、特徴ベクトルの次元を決定する。対称的に、元の入力は、最終的には隠れた中央層から出力層へのノード数を増やすことによって特徴ベクトルから再構成される。次元圧縮の仕組みについて、単純で一般的に利用される手法は、主成分分析(PCA)である。しかしながら、ヒントンらは、ディープオートエンコーダは画像

再構成と圧縮特徴の獲得についてはPCAを上回っていることを実証した[27]。彼らの功績を参照して私たちは、従来の分類器を用いて、行動認識タスクを容易にするためにクロスモーダルメモリ検索の精度特徴のまばらさを優先しているのである。我々の次元圧縮フレームワークのためにディープオートエンコーダを利用した。

3.2　時間遅れネットワークを使用した時系列のマルチモーダル学習

時間遅れニューラルネットワーク（TDNN）は多次元時系列学習のためのフィードフォワードニューラルネットワークを利用する方法である　[29]「異質な言語認識のための時間遅れニューラルネットワーク構造」。TDNNが動機となって、私たちは時系列の学習のためのディープオートエンコーダを利用して新たな計算の枠組みを提案する。単一の時間ステップにおける時系列学習ネットワークへの入力は関節角度ベクトル、画像特徴ベクトル、および音声特徴ベクトルの時間領域によって定義され、以下のように構成される。

$s_t = (a_{\mathbf{t}}, u^i_{\mathbf{t}}, u^s_{\mathbf{t}})$　　式(4)

$\{\mathbf{t} | t - T + 1 \leq \mathbf{t} \leq t\},$　　式(5)

ここでS_t、a_t、u^i_t、u^s_tはそれぞれ時間tにおけるネットワークへの入力、関節角度、画像特徴、および音声特徴を表すベクトルであり、Tはタイムウィンドウの長さである。ここで、$\mathbf{t}$はtからTの時間領域の段階より前を表し、添字$\mathbf{t}$のベクトルはベクトルの時系列を示す。時系列の学習ネットワークの入出力マッピングは、以下のように定義されている。

$v_t = g(s_t)$　　式(6)

$\hat{s}_t = g^{-1}(v_t),$　　式(7)

ここで、v_tと$\hat{s}_t = (\hat{a}_{\mathbf{t}}, \hat{u}^i_{\mathbf{t}}, \hat{u}^s_{\mathbf{t}})$は、それぞれマルチモーダル特徴ベクトルと復元マルチモーダル時系列配列である。関数$g(.)$と$g^{-1}(.)$はそれぞれ、ネットワークの入力層から隠れた中央層と隠れた中央層から出力層への変換マッピングを表す。マルチモーダル時系列学習にニューラルネットワークを適用することのメリットの1つは、それらの汎化能力である。ネットワークは、入力データの不足を補うことができるので、時系列学習ネットワークは、2つの異なる方法で使用することができる。

（1）1つのモデルから現在の配列を取得して他のモデルに利用できること。

（2）過去の配列から将来の配列を予測すること。

このように、時系列学習ネットワークは、クロスモーダルメモリ検索または空間的または時間のいずれかの方法で、ネットワーク外部からの入力データをマスクすることにより時系列予測器として機能します。これにより、時系列学習ネットワークは、クロスモーダルメモリ検索ができ、さらに、入力データにマスクすることにより、時系列予測器として機能します。これらの機能の実用的な実装は、以下のサブセクションで説明されている。

知的情報処理特論　レポート

講義日：2015 年 4 月 30 日

作成者：T15M081 前田一磨

3.3　クロスモーダルメモリ検索

クロスモーダルメモリ検索はネットワーク外から他のモダリティに対応する配列を提供することにより、ネットワーク内のモダリティのための自己生成配列によって実現される。検索したモダリティについては、入力ノードと出力ノードから周期ループが用意されている。したがって、動き・音配列から画像配列を生成する場合のネットワークへの入力は以下のように定義される。

$$s_t = (a_t, \hat{u}_t^i, u_t^s). \qquad \text{式(8)}$$

図 1 に示すように繰り返し入力の時間領域は、(1)最も古いタイムステップの出力を破棄し、(2)出力から取得した最新の値を持った最新のタイムステップを満たすことによって、一段階の方向へのネットワークの前の出力を移動することで生成される。(つまり、今回の例では、動き・音の配列繰り返し入力することによって足りない画像配列する。)

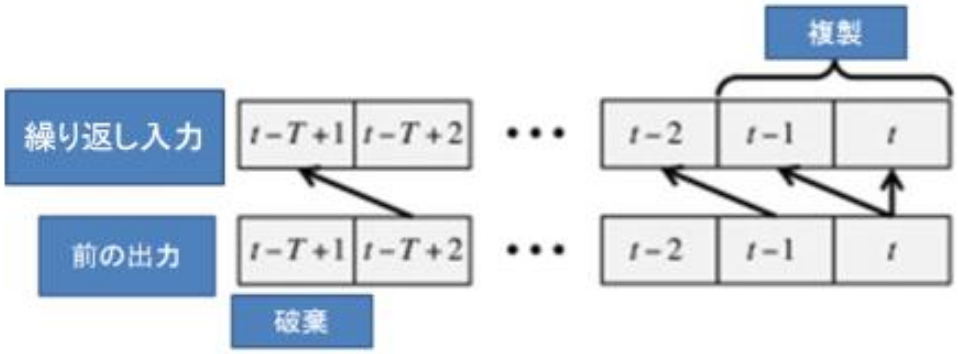

図 1　繰り返し入力のバッファシフト

3.4　時系列予測

同様に、時系列予測は、入力層、出力層から周期ループを構成することによって実現される。違いは、全てのタイムウィンドウ T のステップの中で入力データで満たされる両方のモダリティの最初のT_{in}ステップ(すなわち、過去のT_{in}が現在の t のステップに移動す

る)だけである。残り（すなわち、未来の $T\text{-}T_{in}$ は、予測された時間ステップに移行する）は、前の時間ステップからの出力で満たされる。したがって、ネットワークへの入力は以下のように定義される。

$$s(t) = (a_{\mathbf{t}_1}, \hat{a}_{\mathbf{t}_2}, u^i_{\mathbf{t}_1}, \hat{u}^i_{\mathbf{t}_2}, u^s_{\mathbf{t}_1}, \hat{u}^s_{\mathbf{t}_2}), \qquad \text{式（ 9 ）}$$

$$\{\mathbf{t}_1 | t - T_{in} + 1 \le \mathbf{t}_1 \le t\}, \qquad \text{式（ 10 ）}$$

$$\{\mathbf{t}_2 | t + 1 \le \mathbf{t}_2 \le t + (T - T_{in})\}. \qquad \text{式（ 11 ）}$$

図 2 に示すように、繰り返し入力の予測領域は、一段階の時間方向へのネットワークに相当する前の出力を移動することで生成される。（つまり、動き・音・画像それぞれの情報を繰り返し入力することによって足りない情報を予測して生成する。）

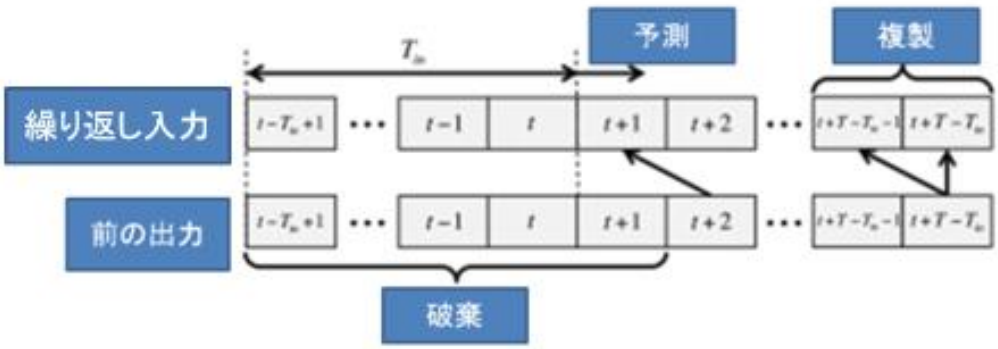

図 2　時系列予測の繰り返し入力のバッファシフト