第 32 回　Weekly Report

深層学習を用いた動画像からの危険認知手法のための基礎的研究

2 月　2 日　(火)

小松　大起

1　はじめに

1.1　研究背景

近年, 情報処理技術として知的処理技術の一つである深層学習が様々な分野で用いられている. 深層学習とは, ニューロンの層が多段に組み上げられたニューラルネットワークのことを指す.[1] ニューラルネットワークとは人間の脳の仕組みから着想を得たものであり, 神経回路網をコンピュータ上で表現しようと作られた数理的モデルである. 深層学習で用いられる分野としては株価予想や人物認識や表情認識, 擬似的なデータを生成するアルゴリズムである GAN を用いた画像生成などに挙げられる画像処理, 話し言葉や書き言葉など我々が普段使うような自然言語を対象として, それらの言葉が持つ意味を解析する自然言語処理などがある.

1.2　研究目的

人間の認知は時間経過による視覚世界の変化の予測が可能である. 従来の深層学習においては静止画の画像処理が中心であったが, 本研究では動画像から未来フレームの画像生成へと拡張を行い, 危険認知や行動予測などへの予測画像応用を行うための予測画像生成及び, その生成画像の評価方法を明らかにすることである.

2　PredNet

PredNet は Deep Recurrent Convolutional Neural Network の 1 種で 神経科学の概念である Predictive Coding を組み込んで作られたモデルである. 2016 年に William Lotter, Gabriel Kreiman, David Cox の 3 氏によって公開された.

2.1　PredNet の層構造

PredNet の各層には 4 つの素子が存在しておりそれぞれ、Target, Representation, Prediction, Error と呼ぶ. Target は下層からの出力である誤差信号をエンコード, 符号化する. Representation は Recurrent unit で, 上層からの出力, 側方からの誤差信号, 1 ステップ前の自分の出力を受け取る. Representation unit は Target の予測をする Prediction unit に投射し, 入力の予測が出力される. Error は Prediction と Target の誤差であり, Error は上層に送られる. この Error が小さくなるように学習を進めていく. また, 層構造の例を図 1 に示す.

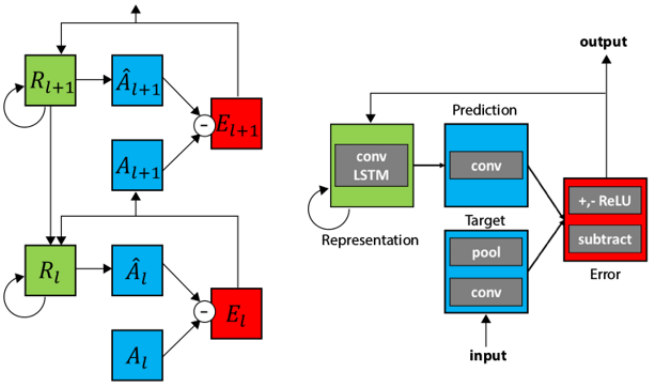

Fig. 1: PredNet の層構造の例

3　使用した動画像

3.1　KITTI

ドイツの都市環境を運転している車の屋根に取り付けられたカメラによってキャプチャされたデータセットである.City, Residential, Road のカテゴリに分かれており, それぞれ都市街, 住宅地, 高速道路というようなシーン分けがされている. それぞれのカテゴリに

は City には 28 シーン, Residential には 21 シーン, Road には 12 シーン存在している. 合計 61 のシーンがあるが, それぞれのカテゴリから 10 フレームがテストデータとしてサンプリングされ予測画像生成に用いられ, 57 シーンが訓練に用いられ, 4 シーンが検証に用いられる. また, 用いられる画像は中央でトリミングされて,128 × 160 ピクセルの画像となっている. 1 フレームは 0.1 秒である. 最大 5 フレーム先の画像を生成することが可能.

4 オートエンコーダ

2006 年に hinton らによって提案された, 自己符号化器とも呼ばれるオートエンコーダはニューラルネットワークのモデルの一つである. 画像や音声データなどを符号化と呼ばれる圧縮作業を行い, 意味のある特徴量を残した後にそれを用いて復号化と呼ばれる次元の復元を行うことを目的として用いられることが多い.

4.1 本実験で用いるオートエンコーダの構造

使用する画像のサイズは 100 * 100 を RGB の 3 次元で入力とした. 中間層での素子の値 z_i 及び, 出力層の素子の値 y_k は, 入力のフィドフォワードで求められる. それぞれ, z_i の値と y_k の値は以下の式で与えられる.

$$z_j = \sum_i f_{ji} x_i \tag{1}$$

$$y_k = \sum_j w_{ji} z_i \tag{2}$$

また、中間層の素子での値を $\vec{z}$, 出力層の素子での値を $\vec{y}$ のベクトルで表す. f_{ji} を (j, i) 成分に持つ行列 F とした時, 以下のように表すこともできる.

$$\vec{z} = F\vec{x} \tag{3}$$

$$\vec{y} = W\vec{z} \tag{4}$$

また、オートエンコーダの概略図を図 2 に示す. 最適化関数には Adam を用いて, 損失関数には binary crossentropy を用いた. 中間層は 32 次元にして圧縮を行っている. 現段階での出力の結果を図 3 に示す.

5 活性化関数

活性化関数とは, ニューロン間の移動に伴い入力値を別の数値に変換して出力するための関数のことである.

5.1 ステップ関数

ステップ関数は, 入力が 0 未満の場合には常に出力値が 0 となり, 0 以上の場合には常に出力値が 1 となるような関数を指す. ステップ関数は, パーセプトロンから用いられている関数であり入力 0 を起点として階段状のグラフを示す. この起点を閾値と呼ぶ. 入力を x として $f(x)$ を出力とすると数式は以下の式で表される.

$$f(x) = \begin{cases} 0, & (x < 0) \\ 1, & (x \geq 1) \end{cases} \tag{5}$$

6 物体検出

一般的に物体検出においては検出したい物体の訓練データを, 検出モデルに学習させた上で行う. そのため学習していない物体は物体として認識することができず, 検出ができない. 一般的な物体検出の方法を以下に説明し, 実際に訓練データなしでの物体検出がどの程度行えるのかを OpenCV を用いて実験する. その結果を図 3 に示す. 結果からわかるように影を検出したり, 車を検出できていなかったり実用に耐えうるものではないことがわかった.

7 今週の作業

- OpenCv を用いた訓練データを用いない物体認識
- 本選考エントリー

8 来週以降の作業

- オートエンコーダで中間層を扱えるようにする

9　参考文献

[1] 浅川伸一．python で体験する深層学習．コロナ社, 2016.

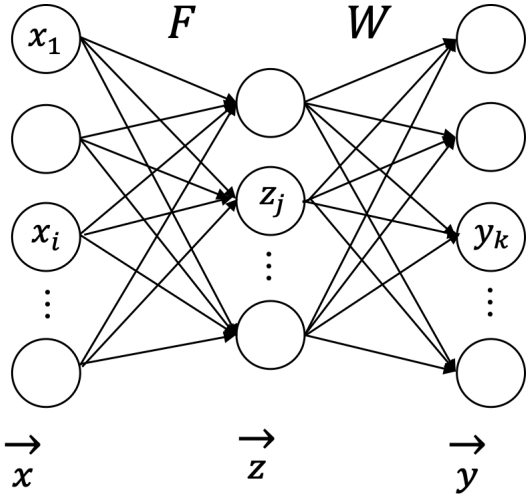

Fig. 2: autoencoder の概略図

Fig. 3: 訓練データなしの物体認識