

1.3 なぜディープラーニングなのか、 なぜ今なのか

17T4028N 菊田尚樹

1.3 なぜディープラーニングなのか、なぜ今なのか

畳み込みNN、誤差逆伝播法→1989には理解されていた

長短期記憶(DLにおける根本的なアルゴリズム)→1997年に開発

しかし、ディープラーニングが注目されだしたのは2012年ごろ

なぜか…

機械学習の進化は以下の3つの技術的要素に依存

- ・ハードウェア
- ・データ
- ・アルゴリズムの進化

新しいアイデアを試してみるには適したハードウェアとデータが必要

1.3.1 ハードウェア

計算機の高速化を後押ししたのはゲーム市場

2000年代

NVIDIA,AMDなどの半導体メーカーがビデオゲームのグラフィックス処理のためにGPU(高速な並列計算が可能)の開発に多額の投資

2007年 NVIDIAがCUDAをリリース

CUDA：GPU製品のプログラミングインターフェース

→高度な並列計算が可能なアプリケーションにおいてCPUがGPUに置き換わる 大量の行列演算を行うDLは恩恵を受ける

さらにDL専用のチップの効率化に投資が進む

2016年 GoogleがDL専用のテンソル処理ユニットTPUを開発

2018に出た三世代目は100ペタFLOPS

1.3.2 データ

データは機械学習における貴重なエネルギー資源
インターネットの台頭によって機械学習用の大規模なデータセットの収集と配布が可能に

- (例) 画像：Flickr (個人が取った写真の公開、共有を行うサービス)
タグ付けすることで分類し検索性を上げている
→写真とタグは画像認識にもってこい
- 動画：YouTube
文書：Wikipedia

1.3.2 データ

画像認識用データセット

- **ImageNet** : 現在 1400万枚以上の画像を収録 2万以上の分類
INLSVRC(ImageNet Large Scale Visual Recognition Challenge)という画像認識大会が毎年開催されている
研究者の競い合いがディープラーニングの進歩につながる
- **MNIST** : 手書き数字(0-9)画像を収録(学習用6万枚 テスト用1万枚)
- **CIFAR-10** : 飛行機、犬など10クラスの画像を収録
1クラスあたり6000枚の合計6万
(学習用5万 テスト用1万枚)

1.3.3 アルゴリズム

2000年代後半まで深いNNの確実な訓練法は発見されていなかった

↑ 多層における勾配の伝播が困難

2009~2010年に勾配の伝播の問題を改善

- ・ 活性化関数の改善
 - ・ 重みの初期化法の改善
 - ・ 最適化手法の改善
- 10層以上のモデルで訓練可能に

2014~2016年にはさらに高度な手法が発見

- ・ バッチ正規化
 - ・ 残差接続
 - ・ dw畳み込み
- 現在は数千もの層のモデルで訓練可能に

1.3.4 新しい投資の波

2012から2013年にかけてディープラーニングは最先端のアルゴリズムとなり、業界のリーダーに注目され、AIの歴史上最大の投資が進んでいく

- 2013年 Googleは5億ドルでDeepMind(AlphaGoを作成)を買収
- 2014年 Baidu(中)がDLの研究プロジェクトに3億ドル投資
- 2016年 Intelが4億ドルでNervana Systems(DL用のプラットフォームを提供)を買収

1.3.5 ディープラーニングの大衆化

当初はC++とCUDAに精通した人でなければディープラーニングの研究はできなかったが現在はPythonが書ければできてしまう

[Pythonにおけるディープラーニングのフレームワークの例]

- **TensorFlow** : Google製 2015年～ 多次元配列を用いた処理が特徴
Define-and-Run方式
 - **Keras** : TensorFlow上のライブラリ 初心者向け
- **PyTorch** : Facebook製 2017年～ Numpyに似ている
Define-by-Run方式
- **Chainer** : 日本製 2015年～ 日本語のサイトが多くあり勉強しやすい
Define-by-Run方式

1.3.6 これは続くのか？

ディープラーニングの3つの特性

- 単純さ

特徴エンジニアリングが不要、エンドツーエンドの訓練可能モデル

- スケーラビリティ

GPU・TPUによる並列処理に適している

→ムーアの法則(1.5年でトランジスタ数が二倍)にそって発展？

- 多様性と再利用可能性

一から作るのではなく追加での学習が可能
作成済みのモデルを別の目的で利用可能

今後もDLとその基本概念は継承されると予想