How transferable are features in deep neural networks?

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson

発表者　鈴木雅大

本論文について

- NIPS2014での論文
  - 被引用件数：15（去年発表された論文にしては多い？）
- Deep Learningで転移学習
  - ただし何かしらの新しいモデルを考案したという内容ではなく、みんな気になっていたことを色々実験・考察したという内容
    - 理解にそんな難しいことはありません
  - 論文内で扱っている手法について「転移学習」と言っているが、これまでの転移学習研究全体の中での位置付けなどについてはほとんど言及せず
    - そもそも転移学習について色々論じた論文ではない
    - 基本的には論文に従って説明します

DNNのgeneralとspecificについて

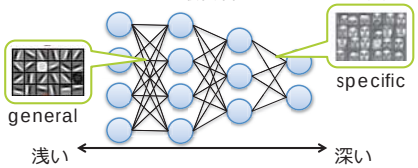
- 深いニューラルネットワークで画像を学習すると、次のような奇妙な現象が現れる

浅い層（1層目など）：

- ガボールフィルターのようなパターンが現れる
- この現象はデータセット、教師ありや教師なしなどの問題設定、損失関数などによらない（**general**）

深い層（最終層など）：

- データセットやタスクによって最終層の出力は大きく異なる（**specific**）



※特徴画像はhttp://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/CVPR2012-Tutorial_lee.pdfより

疑問点

generalとspecificについて次のような疑問が生じる

疑問1.
　ある層がgeneralかspecificかを計測することができるのか？
疑問2.
　general→specificの変化はある層で突然起こるのか？それとも何層にも渡って変わるのか？
疑問3.
　その変化はどの辺りの層で起こるのか？

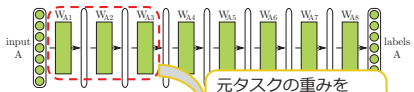
これらの疑問が解決すると・・・

DNNのgeneralな部分をみつけて転移学習に利用できる！

転移学習

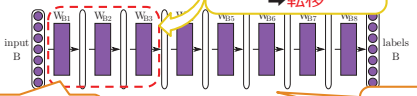
本論文での転移学習の流れ

元タスク（base task）



元タスクの重みを目標タスクにコピー
➡転移

目標タスク（target task）



転移する層がgeneralな方がうまくいく

残りの層はランダムに初期化&目標タスクのデータで学習

目標タスクに関するデータが少ないとき、転移によって**過学習を防ぐことができる**

（補足）ここでいう転移学習は[Pan et al. 2010]を参考にすると、帰納転移学習（inductive transfer learning　異なるタスク間の転移）にあたると思われる

fine-tuneとfrozen

転移学習では元タスクから目標タスクに転移したあと、2つの方法が選択できる

1. コピーした層を目標タスクで誤差逆伝播（fine-tune）
2. コピーした層の重みは変更しない（frozen）

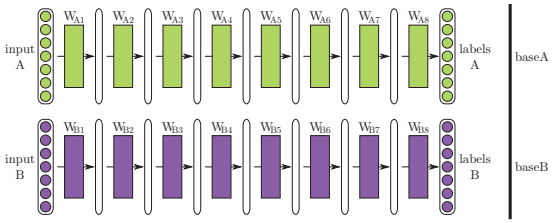
目標タスクのデータが少ない場合はfrozenの方がいい（過学習に陥るから）が、データが多いならfine-tuneの方がいい

→この２つの方法についてそれぞれ実験

モデルの設定

元タスクをA、目標タスクをBとする

- AもBも8層の畳込みニューラルネットワークとし、学習済みのネットワークをbaseA、baseBと呼ぶ（学習するデータセットは後述）
  - 8層なので、コピーできるn層は{1,2,...,7}となる
  - 「n層コピーする」とは一番浅い重みを1としてn番目までを全てコピーという意味

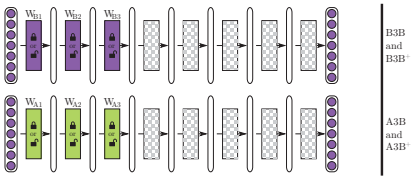


モデルの設定

n= 3 のとき

- B3B：baseBから3層をコピーしfrozen,残りの層をランダムに初期化しデータセットBで学習（つまり転移なし（selffer））
- A3B：baseAから3層コピーしfrozen,残りの層をランダムに初期化しデータセットBで学習（つまり転移あり（transfer））
- frozenではなく、fine-tuneの場合はB3B+、A3B+と表記する

※この図では
ロック= frozen
ロックが解除=fine-tune

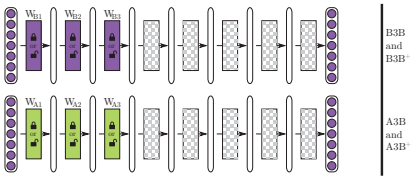


これを全てのnと両方向の転移（AnBとBnA）に関して実験する

モデルの設定

この設定をすることで、実験の結果によって次のことがいえる

- もしA3BがbaseBと同じくらいの性能ならば、**3層までは（少なくともBに関して）general**といえる
- もしA3BがbaseBに比べて性能が悪くなったら、**3層目の特徴はAについてspecific**といえる



データセットについて

データセットはImageNet[Deng et al., 2009]を使用

- 1000クラス、訓練データ：1281167枚、テストデータ：50000枚
- AとBで学習するためにデータを半分（500クラス、約645000枚）に分ける

- できるだけ似ているデータセットに分ける
  - ImageNetは階層構造で、似ているクラスがクラスタになっているので、クラスタごとに均等になるようにランダムに分ける（例：ネコ科は13クラスあるので、6と7にランダムに分ける）
- 似ていないデータセットの分け方
  - 階層構造を利用してman-made（551クラス）とnatural （449クラス）に分ける

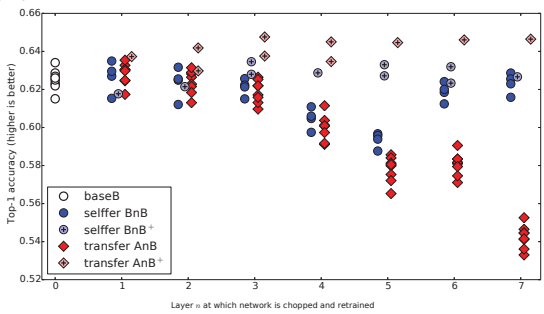
転移学習は似ている方がうまくいく（後の実験で検証）

実装について

- 実装はCaffe [Jia et al., 2014]を利用
  - 本研究ではよりよい精度を求める研究ではないので、多くの人に使われているCaffeを選択
  - 詳しい設定やパラメーターなどはhttp://yosinski.com/transferで公開されている
    - ipython notebookで本論文に載せてある図を描画できる

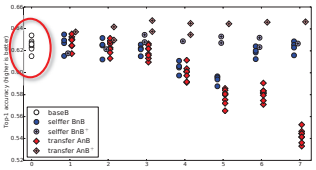
実験1：似ているデータセット

- 実験結果



平均正解率での評価
- AとBのそれぞれでランダムに分割（4回実行）
- AnAもBnBもBnBと表記（統計的に等しいから　AnBについても同じ）

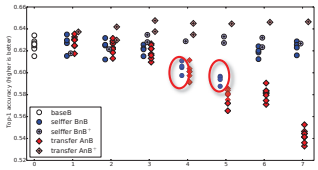
実験1の考察



考察1

- baseBの結果のエラー率は37.5%で、1000クラスのときのエラー率42.5%より低くなっている
  - 500クラスの方が1000クラスより間違え方（つまり分類先）も半分になっているため

実験1の考察

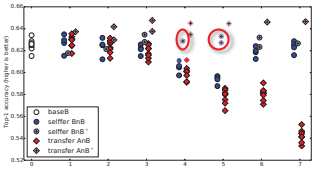


考察2

- 何故かBnBの正解率が4,5層辺りで特に落ちている（同じBだから落ちるはずがない）

  →元のNNがその層辺りに**壊れやすい共適応(co-adaptation)**の特徴を含んでいる証拠
  - それぞれの特徴が複雑で壊れやすく互いに作用しているため、frozenのように分離してしまうと、上の層までの誤差逆伝播だけでは**再学習できない**（一緒に学習しなければならない）
  - 6,7層目でだんだん正解率が戻っているのは、深い層に行けば行くほど、繋がり方もよりシンプルになって（共適応の特徴も少なくなって）、勾配法でもよい解を見つけられるから

実験1の考察

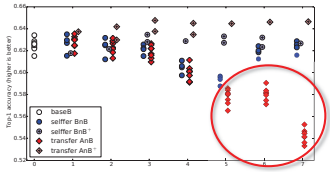


考察3

- BnB+をみると、どの層でもbaseBと大体同じくらいの正解率になっている

  →fine-tuningによってBnBでの問題が改善された

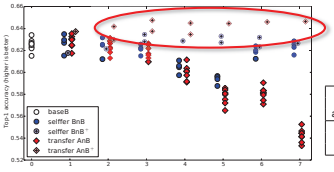
実験1の考察



考察4

- AnBをみると、1,2層は正解率はあまり低下せず、うまく転移できていることがわかる
  →1層だけでなく、2層の特徴もgeneralである証拠
- 3層ではわずかに低下し、4層~7層はかなり正解率が落ちている
  →考察2から
  3, 4層：共適応が失われたため
  5~7層：共適応が失われた&specificになったため
  であるとわかる

実験1の考察

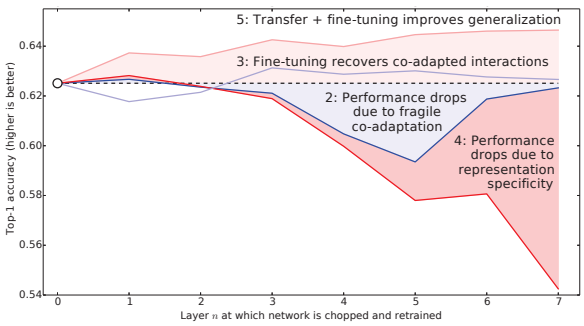


layers aggregated	mean boost over baseB	mean boost over selffer BnB+
1-7	1.6%	1.4%
3-7	1.8%	1.4%
5-7	2.1%	1.7%

考察5

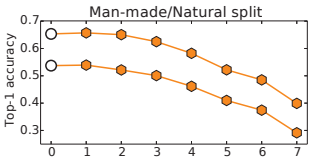
- AnB+をみると、正解率がbaseBやBnB+と比較してもよくなっている！
  - これまでは、目標タスクのデータが少ないときに過学習を防ぐため転移学習がいいとされていた
  - しかし今回の結果から、データが多いときでも転移学習によって汎化性能を高めることがわかった（これが訓練時間が長いからでないのはBnB+との比較で明らか）
  - fine-tuningしたあとでも元タスクの効果が残っている！
- 汎化による改善は、転移した層によらない
  - 転移する層を増やした方がわずかによくなっている（右上図）

実験1の考察のまとめ



実験2：似ていないデータセット

- 実験結果



baseAがman-made, baseBがnatural
- 白丸　　：baseA（下）とbaseB（上）の正解率
- オレンジ六角形：BnA（下）とAnB（上）の正解率

- 考察
  - naturalの方man-madeよりが正解率が高い
    - natural（449クラス）の方がman-made（551クラス）よりクラス数が少ない
    - naturalの方が簡単なタスク

    という理由が考えられる

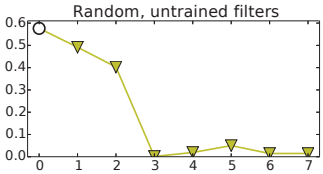
実験3：ランダムな重み

- [Jarrett *et al.* 2009]によると、畳込みフィルター、Rectificationなどをランダムにしても学習したのと同じくらいの精度となるらしい
  - 比較的小さいNN（2,3層）
  - 比較的小さいデータセット（Caltech-101）

→深いNN&大きなデータセットでも同じなのか？

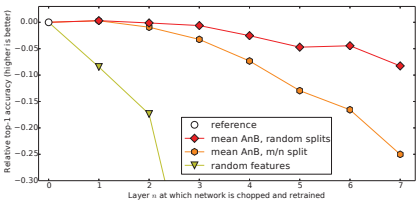
実験3の結果

- 実験結果



- 考察
  - 1,2層と落ち始め、3層以降が大きく正解率が下がっている
    - 小さいNNやデータセットのように単純ではない
  - ただし[Jarrett *et al.* 2009]とは色々設定が異なるので、単純に比較もできないことに留意

実験1,2,3の結果の比較



考察
- 層が深くなるほど、似ているデータセットと似ていないデータセットの差が開いている
- 似ていないデータセットでも転移した方が、ランダムよりもよい結果
  - [Jarrett *et al.* 2009]の結果と異なるのは、[Jarrett *et al.* 2009]では小さいデータセット（Caltech-101）で訓練していて過学習に陥っているからと考えられる（だから、ランダムな方がうまくいく）

まとめ

本研究ではNNのどの層が転移できるか、すなわちgeneralとspecificがどの層で起こっているかを明らかにする手法を提示した。そして実験によって次のことが明らかになった

- frozenの設定で転移した時、次の2つの原因によって性能が落ちる
  - 壊れやすい共適応の間を分離
  - specificな特徴を転移
- タスク間の距離が大きくなると、転移したあとの性能が下がる
- しかしランダムな場合と比べると、似ていないタスクでも転移した方がよい結果となった
- 転移したあとfine-tuningすると、どの層でも汎化性能が向上した

感想

- 実験＆考察がかなり大変そう
  - 相当の計算資源&時間が必要（GPUで9.5日とか書いてあった）
- DLの研究では、今回のような実験してみた系の研究が重要
  - NNやデータが大規模になってくると、どうなってるのかわからないことが多くなるので、今回のような論文は増えてきそう
- ソースコードの公開は重要
  - やはりDLでは再現性が問題になってくるので、大規模な実験の場合は、ソースコードの公開は必須？
  - パラメータとか論文に書ききれない

参考文献など

- 転移学習
  - R. Caruana, “Multitask learning,” Mach. Learn., vol. 28(1), pp. 41–75, 1997.
  - S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.
- 転移学習とDeep Learning
  - Y. Bengio, “Deep Learning of Representations for Unsupervised and Transfer Learning,” JMLR Work. Conf. Proc. 7, vol. 7, pp. 1–20, 2011.
- その他
  - 本実験の実装：http://yosinski.com/transfer
  - CVPR 2012 Tutorial : Deep Learning Methods for Vision（画像の引用）http://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/CVPR2012-Tutorial_lee.pdf