留学報告書（2025 年 6 月）

1. はじめに

大古 一聡と申します。UC Berkeley の EECS で博士課程 1 年目を終えました。

2. 論文

こっちに来てから初めての論文を出しました。対照学習の理論解析です。ざっくり説明します：

まず対照学習について。画像を処理するモデル $f(x;\theta)$ と文章を処理するモデル $g(y;\psi)$ があるとします。特定のタスク（例：テキストとノイズの乗った画像を入力としノイズを除去した画像を出力する）用に訓練することもできますが、異なる訓練の方法を考えることができます。画像xとテキスト yのペアを大量に用意します。そして、xをモデル fに、yをモデル gに入力し、２つのモデルの出力の内積 $\langle f(x;\theta), g(y;\psi)\rangle$ を考えます。画像xにテキストyが対応していたら$\langle f(x;\theta), g(y;\psi)\rangle$の値が高く、そうでないならば低くなるようにモデル$f$と$g$を訓練（内部のパラメータ$\theta$と$\psi$を調整）します。言い換えれば、似ているものが高い値に異なるものが低い値になるようにモデルを訓練するということです。似ているかどうかの値を出して何が良いのと思うかもしれませんが、実はそのようにして訓練されたモデルは容易に（例：僅かなパラメータの変更）様々なタスク（テキストとノイズの乗った画像を入力としノイズを除去した画像を出力する）に転用することができます。それがこの話のすごいところで、この方法はモデルに画像とテキストの有用な特徴を抽出する能力を獲得させるのです。特定のタスクのためのデータを大量に用意するのは大変ですが、画像とテキストのペアなら Wikipedia に大量にありますし、~~質が低くても良いなら~~ Twitter から取ってくれば良い訳ですから、この方法で大規模に訓練したモデルを少量のデータで少しだけ調節して特定のタスクに使えるのは便利です。これ以上話すと長くなるので、より詳しくは昔の OpenAI のブログなどを見ると良いでしょう。

次に今回の研究について。対照学習の説明を聞くとなんとなく上手く行きそうな気もしますが、とは言え腑に落ちないところもあります。１つには損失関数の違い一つまり、似ているかどうかを判別させる際に用いる損失L_1と、転用先のタスクで対応する損失L_2は全く別物であり、モデルが前者の損失L_1を最小化する際に、転用先のタスクを解くために必要な情報を捨ててしまうのではないか、ということ。もっと言えば、モデルf, gとして用いるニューラルネットワークのパラメータの数は膨大であり、冗長性があります。損失という単なる１次元の指標を小さくするようなパラメータは複数あり、獲得される特徴はそもそも一意ではあり得ないのです。（非常に簡単な例を挙げれば、$f(x;\theta), g(y;\psi)$はそれぞれベクトルですが、片方に正則行列P、もう片方に逆行列P^{-1}をかけても内積$\langle f(x;\theta), g(y;\psi)\rangle$の値は変わりません。）

さて、こうした疑問に応えるために、L_1とL_2の中間地点Sを用意しました。これは$S \leq L_1$を満たすため、前者の損失L_1が小さければSも小さくなります。すると、このSで上から評価できるような損失L_2を使っているようなタスクであれば、損失L_1を小さくするだけで十分、つまり似ているかどうかを判別させることで獲得した特徴があれば解けるタスクだということになるのです。そのような損失L_2を使ったタスクの例として、テキストとノイズの乗った画像を入力と

しノイズを除去した画像を出力するタスクや、画像とテキストを入力としてテキストを出力するタスクなどが挙げられます。１つ目の例はテキストから画像生成をする際のサブタスクで、２つ目の例は画像検索が対応します。さらに、特定の分布を仮定した下で、Transformer というニューラルネットワークが実際にこれらの問題を効率的に（少ないサンプル数で）解くことができることを示しました。

このプロジェクトは博士学生３人でやって 100 ページ超えの論文になりました。日本では、特に理論分野において博士課程の学生が 3 人集まって共同研究を行う機会はあまり多くないため、アメリカで博士課程に在籍していることを強く実感した経験でもありました。

3. 授業

アメリカの(コンピューターサイエンスの)博士課程は 5 年あって日本に比べると長いです。その一つの理由は、最初の 1〜2 年は日本で言うところの修士課程と同じように、授業を取っていく必要があることです。多くの大学のウェブサイトには単位互換についてのページがあり、「これをこれに充てて…」という皮算用が捗りますが、実際は制約が厳しいです。特に、日本の大学の授業の多くが 2 時間×週 1 回なのに対し、アメリカは週 2 回（その分取る授業数は少ない）でより 1 つのトピックを掘り下げることに重点が置かれています。日本だと授業がレベル別に分けられていて、学部で入門、修士で発展を扱うということが典型的ですが、アメリカだと一気に入門から発展までやってしまうという文化のようです。内容的には全部知っているのですが、日本で履修した中で完璧に対応する授業を 1 つだけ挙げろと言われると挙げられない、という状況が発生します。

CS 専攻では、CS の各分野から授業をとるという requirement があります。各分野で複数のコースがあってどれかを取ればいいのですが、architecture の分野はそもそも馴染みがない上に、どの授業にも course project というのがあります。授業を受けるだけではなく実際に作ってペーパーにまとめるというかなりハードな演習が含まれており、どれを取るにもハードルが高いと（AI/機械学習専攻の学生の間で）有名です。

今期、僕は architecture の分野で並列計算の授業を取りました。グレードの半分を占める final project は、nearest neighbor search を GPU 上で高速化するというものでした。Nearest neighbor search についてまず説明すると、例えばデータが 100 万個とか 1 億個とかあったとします。これの中から該当するものを高速に検索したい、そしてそれを何回も繰り返したいのですが、毎回の検索の度に全部のデータが該当するか 1 つ 1 つ調べていると非常に非効率です。そこで、先にデータを”整理整頓”しておくことで、全てのデータを見ることなく検索ができるようにしようというのが nearest neighbor search の発想です。

例えば図書館をイメージしてもらうと、目当ての本を手に取る時に全ての本のタイトルを 1 つ 1 つ見たりしませんよね。大体分野で分けられていて、そこからは著者の名前で並んでいる、これが”整理整頓”のイメージです。しかし、データが高次元—100 とか 1000 とか—になると話が変わってきます。本を並べるには、「分野」と「著者の名前」という 2 つだけを見ていれば良

かったのですが、”0.25, 0.52, -0.78, … (以下 997 個続く)”のような数字の列を並べるには、何に注目すればいいでしょうか。最初の 1 つの数字だけを見て並べても、あまり意味のある並べ方にはならないでしょう。このような高次元のデータを扱うための方法はいろいろありますが、精度がいいとされているものはデータの特徴に合わせたグラフ(地図)を作るというものです。

Final project では 10^6 個のデータに対する前処理を 3.5 秒で行い、10^5 回/秒の検索ができるアルゴリズムを実装しました。これは Facebook の実装（別のアルゴリズム）に迫るくらい速いです。意外と学生だけでも公開ライブラリに迫るような実装が作れるという手応えを得ました。大変ではありましたが、165 人の中で最高評価を貰えました。これでコーディングに対する苦手意識が無くなった気がして、良い経験だったと思います。

4. おわりに

サンフランシスコ・ジャイアンツの試合を見に行きました。良い席！