AI 特許紹介(45)
AI 特許を学ぶ！究める！
～Reformer 特許～

2022 年 10 月 7 日
河野特許事務所
所長弁理士　河野英仁

「AI 特許紹介」シリーズは、注目すべき AI 特許のポイントを紹介します。熾烈な競争となっている第４次産業革命下では AI 技術がキーとなり、この AI 技術・ソリューションを特許として適切に権利化しておくことが重要であることは言うまでもありません。

AI 技術は Google, Microsoft, Amazon を始めとした IT プラットフォーマ、研究機関及び大学から毎週のように新たな手法が提案されており、また AI 技術を活用した新たなソリューションも次々とリリースされています。

本稿では米国先進 IT 企業を中心に、これらの企業から出願された AI 特許に記載された AI テクノロジー・ソリューションのポイントをわかりやすく解説致します。

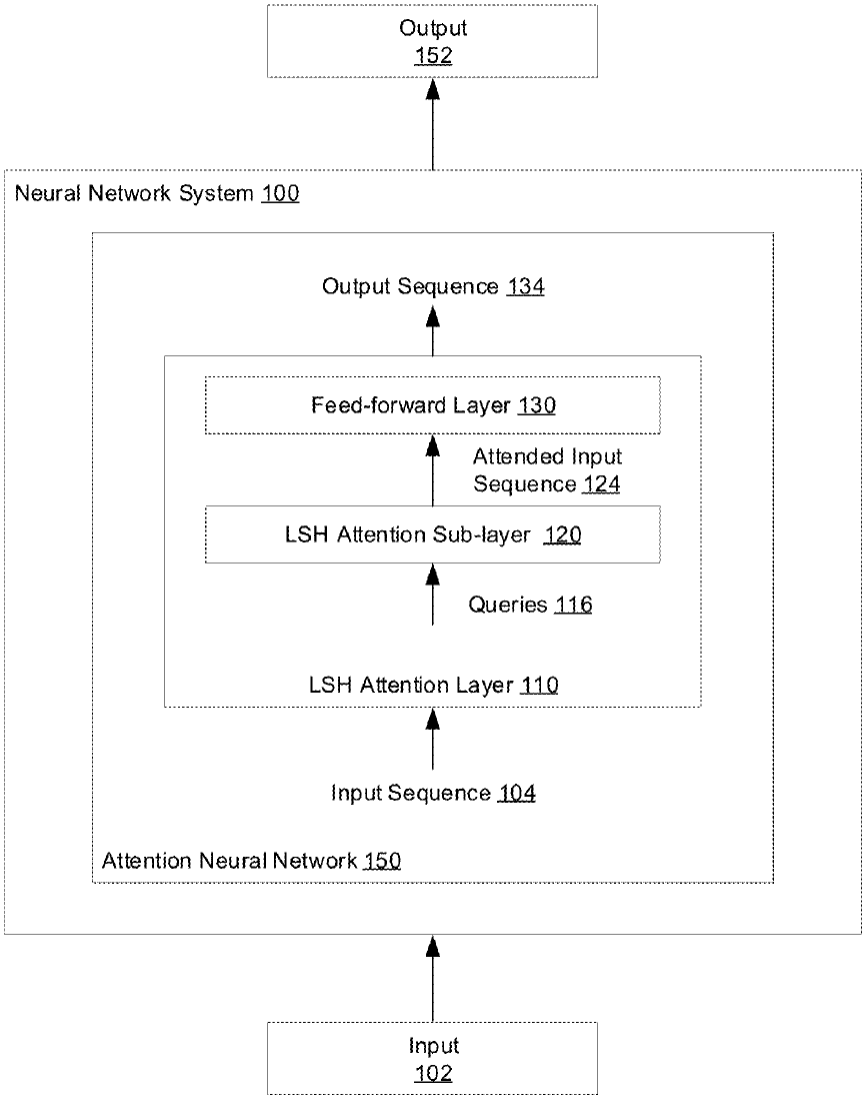
1.概要
特許権者　Google
出願日　2020 年 5 月 8 日
登録日　2021 年 2 月 2 日
登録番号 US10909461
発明の名称　局所性依存ハッシングを使用したアテンションニューラルネットワーク

961 特許は、局所性依存ハッシング（LSH：Locality-Sensitive Hashing）を用いることにより、Transformer の学習速度を向上させる Reformer 特許に関する。

2.特許内容の説明
従来のアテンションニューラルネットワーク内のアテンション層は、内積アテンションメカニズムを採用している。このメカニズムでは、指定されたクエリごとに、すべてのキーを使用してクエリのそれぞれの内積を計算する。ネットワークは、連続するネットワーク入力からキーまたはクエリを取得する。そのため、相当な長さのシーケンシャルデータに内積アテンションメカニズムを適用すると、計算コストが高くなる。

また、各ネットワーク層からのそれぞれのアクティベーションは、合計で数十または数百ギガバイトのメモリを必要とし、バックプロパゲーションされた勾配の計算結果を保存する必要もあるため、トレーニング時にも問題となる。

961 特許はこの問題を解消するためになされたものである。図 1 はニューラルネットワークシステム 100 を示す。



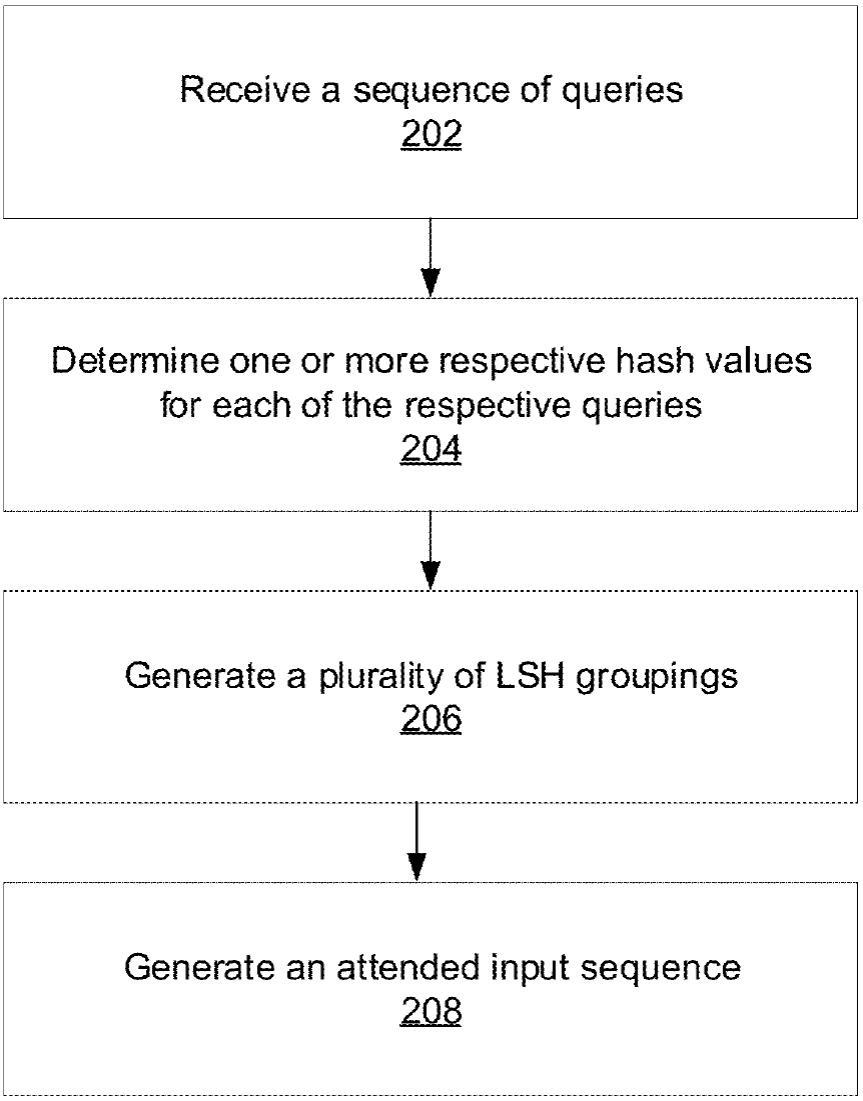
ニューラルネットワークシステム 100 は、入力 102 を受信し、入力 102 に対して機械学習タスクを実行して、出力 152 を生成する。ニューラルネットワークシステム 100 は、局所性依存ハッシング（LSH）アテンション層 110 を含むアテンションニューラルネットワーク 150 を有する。LSH アテンション層 110 は、入力シーケンス 104 上で動作し、対応する出力シーケンス 134 を生成する。

入力シーケンス 104 から出力シーケンス 134 を生成するために、LSH アテンション層 110 は、学習されたクエリ線形変換を各入力位置で各入力に適用して、各入力位置についてそれぞれのクエリQを生成し、学習されたキー線形変換を適用し、各入力位置の各入力に変換して、各入力位置のそれぞれのキー K を生成し、各入力位置の各入力に学習されたバリュー線形変換を適用して、各入力位置のそれぞれのバリュー V を生成する。クエリ Q、キー K、およびバリュー V はすべてベクトルである。

LSH アテンション層 110 は、LSH アテンションサブ層 120、および、フィードフォワード層 130 を含む。LSH アテンションサブ層 120 は、クエリのシーケンス 116 を受信し、次いでクエリ、キー、およびバリューを使用して LSH アテンション機構を適用し、入力シーケンス 104 のアテンドされた入力シーケンス 124 を決定する。アテンドされた入力シーケンス 124 は、一般に、各入力位置における各入力の各アテンドされたベクトルを含む。

各位置別フィードフォワード層 130 は、アテンドされた入力シーケンス 124 内の各位置で別々に動作する。特に、入力位置ごとに、フィードフォワード層 130 は、入力位置でアテンドされた層入力を受信し、入力位置でのアテンドされた層入力に一連の変換を適用して、入力位置の出力を生成する。

図２は、アテンドされた入力シーケンスを生成するためのプロセス 200 のフローチャートである。



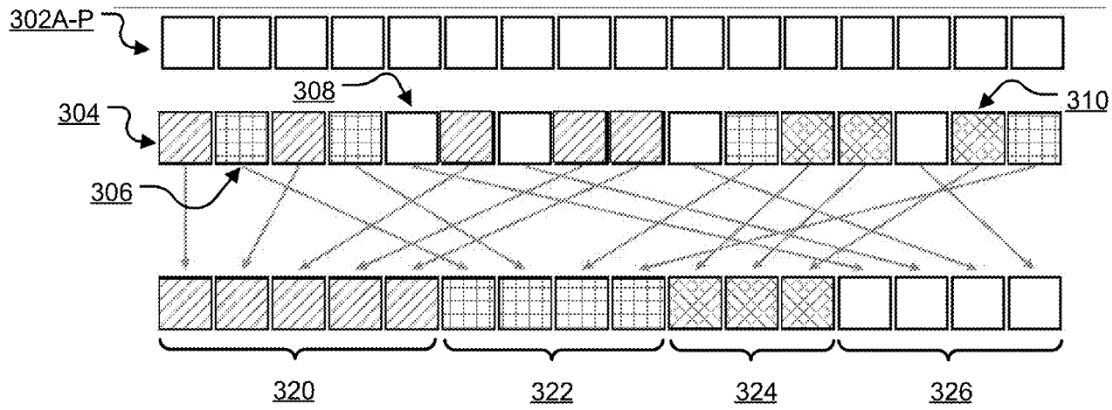
システムは、LSH アテンション層に含まれる LSH アテンションサブ層で、LSH アテンション層への入力シーケンスから導出される一連のクエリを受信する(202)。LSH アテンション層は、学習したクエリ線形変換を入力シーケンスの各入力位置で各入力に適用し、各入力位置のそれぞれのクエリを生成することにより、クエリのシーケンスを導出する。各クエリはベクトルである。

システムは、複数の入力位置のそれぞれで、各クエリのそれぞれについて、1つまたは複数の各ハッシュ値を決定する(204)。システムは、局所性依存ハッシュを使用して各クエリベクトルをハッシュベクトルにマッピングする。局所性依存ハッシュ（LSH)は、近くにあるクエリベクトル(例えば、ハミング、ユークリッド、またはコサイン距離に関して)を遠くにあるものよりも高い確率で同じハッシュベクトルにマッピングできる手法である。これにより、同様の入力ベクトルを同じ LSH グループ（ハッシュ バ

ケット）に高い確率でハッシュする。

LSH グループの数は、入力ベクトルの数よりも少なく、通常ははるかに少なくなる。ランダム射影、安定分布、ハミング距離のビットサンプリングなどのさまざまな方法を使用して、適切な LSH ハッシュ関数を決定できる。

図３Ａは、アテンションニューラルネットワーク内の LSH アテンションサブ層によって適用される LSH アテンションメカニズムの図である。



システムは、LSH アテンションサブ層で一連のクエリ 302A～P を受信し、異なる入力ベクトルを合計 4 つのハッシュベクトル 304-310 にマッピングできる所与の LSH ハッシュ関数に基づいて、特定の入力位置で各クエリのそれぞれのハッシュ値を決定する。

例えば、所与の LSH ハッシュ関数を使用することによって、システムは、クエリベクトル 302A をハッシュベクトル 304 に、クエリベクトル 302B をハッシュベクトル 306 に、クエリベクトル 302E をハッシュベクトル 308 に、クエリベクトル 302O をハッシュベクトル 310 にマッピングする。

システムは、複数の LSH グループを生成する(206)。手短に言えば、これは、(i)同様の、または場合によっては同じハッシュ値を有するそれぞれのクエリを同じ LSH グループに割り当てること、および(ii) 複数の LSH グルーピングにおける各 LSH グルーピングに対して、LSH グルーピングにおけるクエリの対応する入力位置に従ってそれぞれのクエリをソートすることを含む。特に、各入力位置 i について、システムは、それぞれのクエリベクトルが入力位置 i におけるクエリベクトルと同じハッシュ値にハッシュされる入力位置 $P_i=\{j:h(q_i)=h(q_j)\}$のセットを決定する。このようにして、システムは入力位置の異なるセットを決定し、各セットには、すべて同じハッシュ値を持つクエリベクトルのそれぞれの入力位置が含まれる。

次に、システムは、入力位置が同じセット内にあるクエリベクトル（つまり、すべて同じハッシュ値を持つクエリベクトル）を同じ LSH グループに割り当てることにより、複数の LSH グループを生成する。さらに、各 LSH グループについて、システムは、入力シーケンスの先頭にあるクエリベクトルが LSH グループの先頭にも配置されるように、それぞれのクエリを並べ替える。

図３Ａの例に示すように、システムは一連のクエリを４つの LSH グループ 320～326 に割り当てる。マルチラウンド LSH スキームが使用される場合、システムは、対応して、各入力位置 i について入力位置 $P_i^{(r)}$の異なるセットを決定する。

つまり、特定の入力位置にあるクエリごとに、システムは、それぞれのハッシュ値に基づいてクエリを異なる LSH グループに割り当てる。

次に、複数の LSH グループを生成するために、システムは、例えば入力位置 i ごとに異なる入力位置 $P_i^{(r)}$の集合を計算することによって結合し、入力位置 i におけるクエリがどの入力位置にアテンションを向けるべきかを決定する。たとえば、n 個の異なるハッシュ関数$\{h^{(1)}, h^{(2)}, \ldots h^{(n)}\}$を使用して複数回のハッシングを実行した後、システムは入力位置 i ごとに次のように計算する。

$$P_i = \bigcup_{r=1}^{n_{rounds}} P_i^{(r)},$$

ここで、マルチラウンド LSH スキームのラウンド r で、$P_i^{(r)}=\{j:h^{(r)}(q_i)=h^{(r)}(q_j)\}$ は、ハッシュ関数 h(r) を使用して、入力位置 i のクエリ ベクトルと同じハッシュ値にそれぞれのクエリ ベクトルがハッシュされる入力位置のセットを表す。

簡単な例として、ダブルラウンド LSH 中に、シーケンスの最初の入力位置（入力位置 1）がラウンド 1 でセット $P_1^{(1)}=\{1,2,3\}$ に割り当てられ、ラウンド 2 で $P_1^{(2)}=\{1,2,4\}$に割り当てられる。最初の入力位置について、システムは、$P_1^{(1)} \cup P_1^{(2)}$を計算することにより、$P_1=\{1,2,3,4\}$ のセットを決定する。

図 3B は、アテンションニューラルネットワーク内の LSH アテンションサブ層によって適用される LSH アテンションメカニズムの図である。

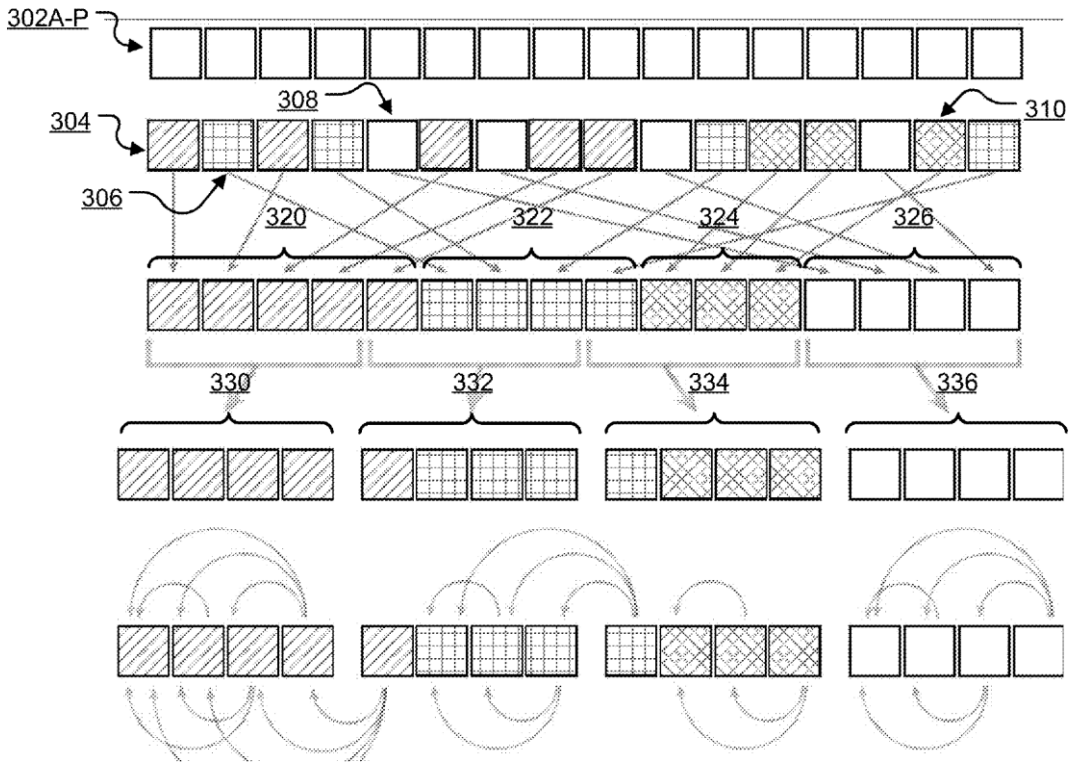


図３Ｂの例では、システムは、LSH グループ 320～326 のシーケンスから LSH セグメント 330～336 のシーケンスを生成する。 特に、LSH グループのそれぞれのサイズは異なる場合があるが、LSH セグメントはそれぞれ同じ固定サイズを持つ。システムは、複数の各 LSH グルーピング内のそれぞれのクエリに対してアテンションメカニズムを適用することに基づいて、アテンドされた入力シーケンスを生成する（208）。

このように、システムは各単一の LSH グループ内でのみアテンションすることができるため、各入力位置でのそれぞれのクエリがアテンションを向けることができる入力位置のセットが制限される。

アテンドされた入力シーケンスは、複数の入力位置のそれぞれにおける各アテンドされた層入力を含む。一般に、アテンションメカニズムは、クエリとバリューのセットを、バリューの加重合計として計算される出力にマップする。クエリとバリューは両方ともベクトルである。たとえば、内積アテンションメカニズムでは、特定のクエリについて、システムは、同じ LSH グループ内で特定のクエリに先行するすべてのクエリとのクエリの内積を計算し、各内積をスケーリング係数で割り（例えばクエリの次元の平方根によって）、次に、スケーリングされた内積にソフトマックス関数を適用して、バリューの重みを得る。次に、システムは、これらの重みに従って、バリューの重み付き合計を計算する。

図３Ｂの後続のクエリから先行するクエリに向かう各曲線矢印は、機械学習タスクを実行する際、対応する後続の入力位置で層入力を処理するときに、バリューの重みなど、いくつかのアテンションの尺度が、システムによって、対応する前の入力位置での層入力に支払われることを示す。

アテンションニューラルネットワーク内の LSH アテンション層ごとに、システムはプロセス 200 を繰り返し実行して（つまり、LSH アテンション層に含まれる 1 つまたは複数の LSH アテンションサブ層のそれぞれで)、LSH アテンション層への入力シーケンスから派生した一連のクエリを処理し、1 つまたは複数のアテンドされた入力シーケンスを生成して、そこから出力シーケンスを決定する。

すなわち、プロセス 200 は、所望の出力、すなわち入力シーケンスに対してシステムによって生成されるべき出力シーケンスが未知である入力シーケンスの出力シーケンスを予測する。

プロセス 200 はまた、アテンションニューラルネットワークをトレーニングして、アテンションニューラルネットワークのパラメーターのトレーニング値を決定するために、トレーニングデータのセットから導出された入力シーケンス、すなわち、システムによって生成されるべき出力が既知である入力のセットから導出された入力シーケンスを処理する。

システムは、初期ニューラルネットワーク層をトレーニングするための従来の機械学習トレーニング技術（確率的勾配降下法、RMSprop、または Adam オプティマイザー等）の一部として、トレーニング データのセットから選択された入力に対してプロセス 200 を繰り返し実行する。

3.クレーム

961 特許のクレーム 1 は以下の通りである。

1. ネットワーク入力に対して機械学習タスクを実行してネットワーク出力を生成するシステムにおいて、該システムは、1 つまたは複数のコンピュータと、1 つまたは複数のコンピュータによって実行されると、1 つまたは複数のコンピュータに以下を実装させる命令を記憶する 1 つまたは複数のストレージデバイスを含み、

機械学習タスクを実行するように構成されたアテンションニューラルネットワークを備え、アテンションニューラルネットワークは 1 つ以上の局所性依存ハッシュ

(LSH：locality-sensitive hashing) アテンション層を含み、各 LSH アテンション層は、1 つ以上の LSH サブ層を含み、各 LSH サブ層は、以下のように構成され、

LSH アテンション層への入力シーケンスから導出された一連のクエリを受け取り、クエリのシーケンスは、複数の各入力位置にそれぞれのクエリを有し、

複数の各入力位置で、各クエリのそれぞれについて、1 つまたは複数のハッシュ値を決定し、

複数の LSH グループを生成し、これには、(i) 同様のハッシュ値を持つそれぞれのクエリを同じ LSH グループに割り当てること、及び、(ii)複数の LSH グルーピングにおける各 LSH グルーピングに対して、LSH グルーピングにおけるクエリの対応する入力位置に従ってそれぞれのクエリをソートすることを含み、

複数の各入力位置でそれぞれのアテンドされた層入力を含むアテンドされた入力シーケンスを生成し、これには、複数の LSH グルーピングにおける各 LSH グルーピングに対して、LSH グルーピング内の各クエリに対してアテンション機構を適用することを含む。

4. 本特許に関連する論文

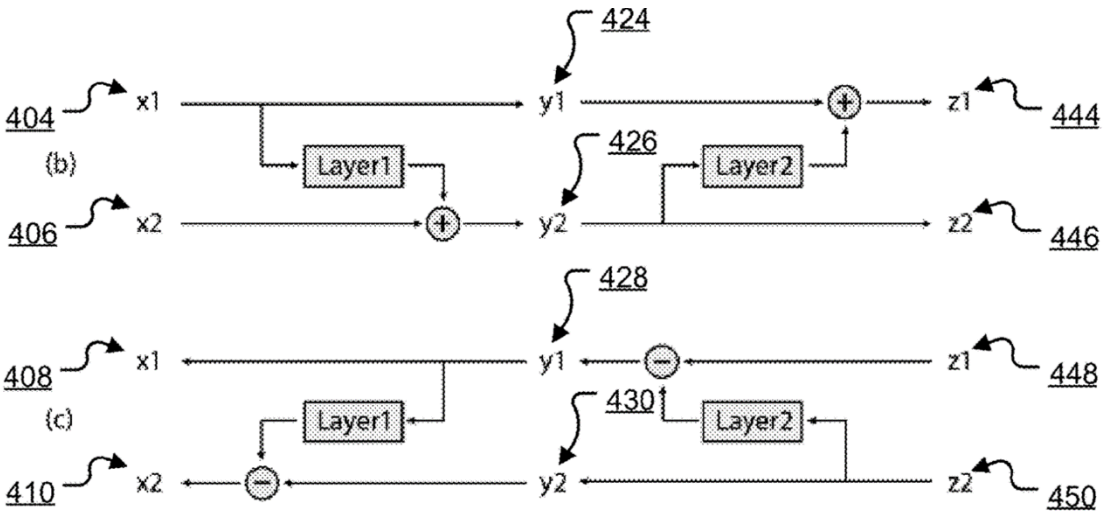
本特許に関する論文“REFORMER: THE EFFICIENT TRANSFORMER”[1]が、Google の Nikita Kitaev 氏らにより公表されている。

大規模な Transformer モデルは、多くのタスクで最先端の結果を達成するが、これらのモデルのトレーニングは、特に長いシーケンスでは非常にコストがかかる。本論文では、Transformer の効率を向上させる 2 つの手法を紹介している。

1 つは、内積のアテンションを、局所性依存ハッシュを使用するものに置き換え、その複雑さを O(L2) から O(Llog L) に変更する。ここで、L はシーケンスの長さである。

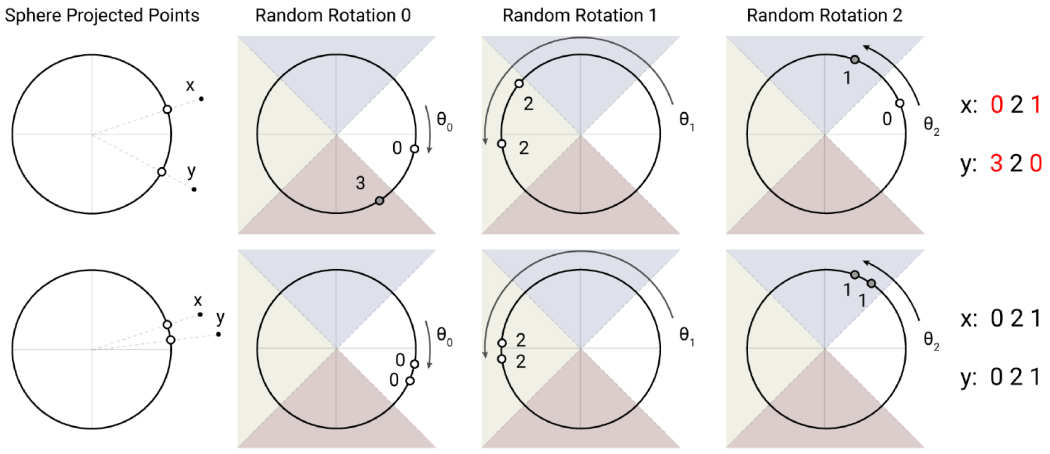
さらに、本稿では解説を省略したが、下記図に示すように、標準の残差層の代わりに可逆的な残差層を使用する。これにより、トレーニング プロセスで活性化を N 回ではなく 1 回だけ保存する (N は層の数)。

[1] Nikita Kitaev, Łukasz Kaiser, Anselm Levskaya “REFORMER: THE EFFICIENT TRANSFORMER” arXiv:2001.04451v2 [cs.LG] 18 Feb 2020

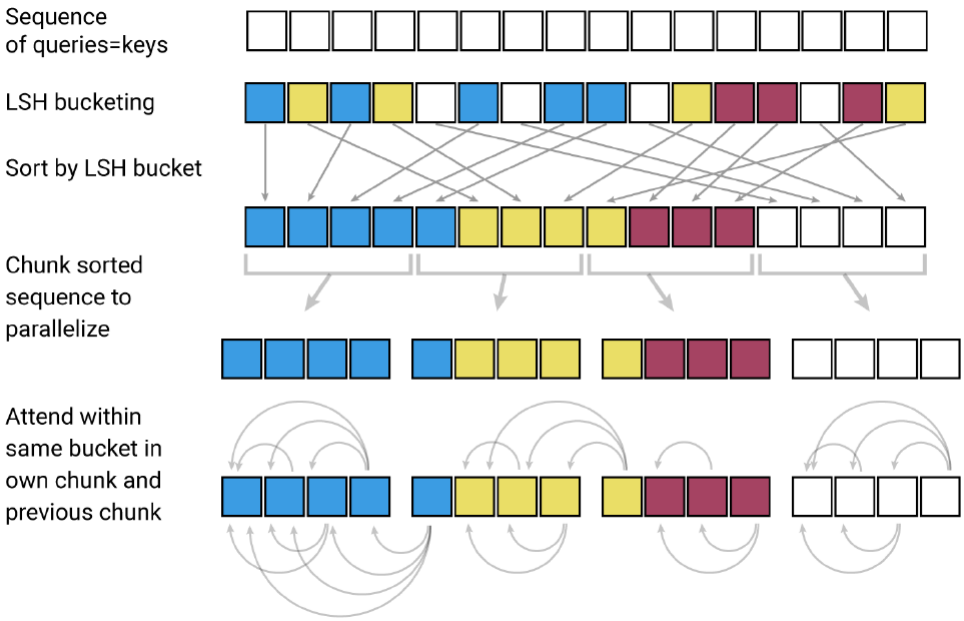


結果として得られるモデルである Reformer は、Transformer モデルと同等のパフォーマンスを発揮すると同時に、メモリ効率が大幅に向上し、長いシーケンスでより高速となる。

下記図は、局所性依存ハッシュの生成イメージを示す説明図である。

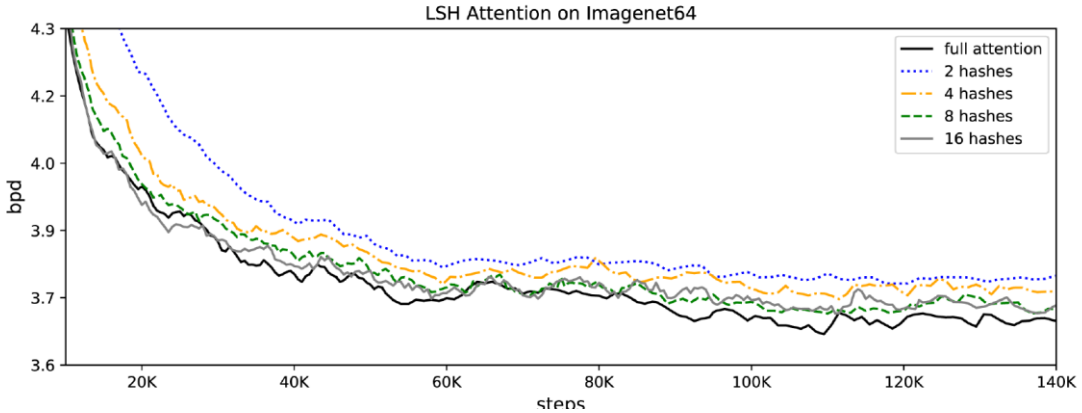


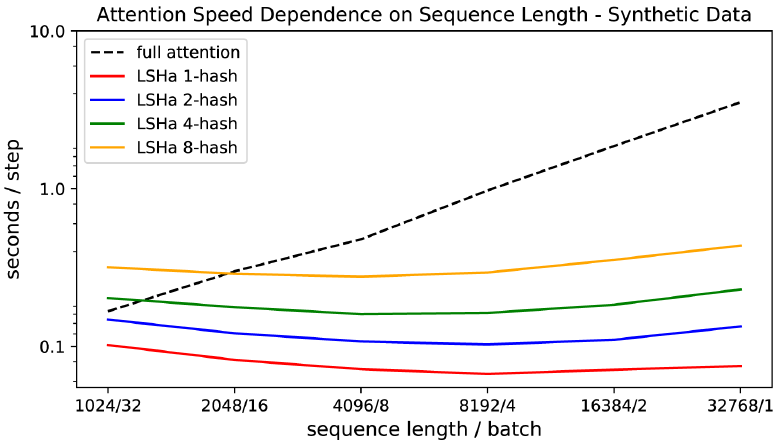
角度局所性依存ハッシュは、球状に投影されたポイントのランダムな回転を使用して、サインされた軸の投影に対する argmax によってバケットを確立する。このシンプル化された 2D の例では、2 つのポイント x と y は、それらの球投影が互いに近くない限り（下側）、3 つの異なる角度ハッシュに対して同一ハッシュバケットに属する可能性は低くなる（上側）。



上述した通り、ハッシュバケット化、並べ替え、チャンク化が行われ、因果的アテンションが生成される。

下記図はフルアテンションおよび LSH アテンションの入力長の関数としてのアテンション評価の速度を評価したグラフである。





上側のグラフに示すように、LSH のアテンションは、完全なアテンションの近似値であり、ハッシュの数が増えるほど正確になる。 $n_{rounds} = 8$ では、すでにフルアテンションにほぼ一致している。モデルの計算コストはハッシュの数に応じて増加するため、このハイパーパラメーターは利用可能な計算コストに応じて調整する。

下側の図は、トークンの総数を固定したまま、さまざまなアテンションタイプの速度とシーケンスの長さをプロットしている。通常のアテンションはシーケンスの長さが長くなると遅くなるが、LSH のアテンション速度は横ばいであることがわかる。

以上

著者紹介
河野英仁
河野特許事務所、所長弁理士。立命館大学情報システム学博士前期課程修了、米国フランクリンピアースローセンター知的財産権法修士修了、中国清華大学法学院知的財産夏季セミナー修了、MIT(マサチューセッツ工科大学)コンピュータ科学・AI 研究所 AI コース修了。
AI 特許コンサルティング、医療 AI 特許コンサルティングの他、米国・中国特許の権利化・侵害訴訟を専門としている。著書に「世界のソフトウエア特許(共著)」、「FinTech 特許入門」、「AI/IoT 特許入門 2.0」、「ブロックチェーン 3.0(共著)」がある。