規則性理解と外挿能力を評価するための足し算

Addition for Measuring the Ability to Understand Regularity and to Extrapolate

藤澤逸平^{1*} 金井良太¹
Ippei Fujisawa¹ Ryota Kanai¹

¹ 株式会社アラヤ

¹ Araya, Inc.

Abstract: In this paper, we discuss the possibility that elementary arithmetic, such as addition and other arithmetic operations, would be useful as a benchmark for measuring the ability to extrapolate which is considered important for the realization of general-purpose intelligence. Understanding addition can be regarded as performing addition of arbitrary digits correctly by memorizing and applying the rules of addition of single digits, and by learning the rules of carrying. We propose a benchmark in which we prepare a small training dataset that is considered sufficient to reveal the algebra of addition, and measure the accuracy using a test dataset that requires large multi-digit operations. Our benchmark has the following advantages over the datasets usually used in recognition tasks and reinforcement learning. Its simple structure makes it easy to generate datasets, adjust and extend the difficulty, identify inductive biases, and allows for discussion from a theoretical perspective in computer science like program synthesis. We hope that the elementary arithmetic benchmark will reveal functions missing in current AI systems, and also provide a good starting point for developing them. In particular, we speculate that the use of the knowledge may be required for a system to compute correctly for arbitrary digits. Finally, based on these insights, we propose a future direction for the development of systems with extrapolation capabilities.

1 はじめに

外挿 (extrapolation) とは、訓練データと異なる領域や異なる性質のデータに対する予測のことである。経済予測や車の運転時などにおいて、未来に対して人間が行う外挿はしばしば誤るものの、それでも多くの場面において、人間のそれはある程度機能する。適切な外挿能力は、汎用人工知能の機能の重要な要素の1つと言えるだろう。

近年、画像認識や自然言語処理などの高次元データに対する多くの認識課題において、GPT[1] や BERT[2] のように、多数のパラメタをもつニューラルネットワークを大量データで学習するというアプローチが一定の成功を収めている。しかしながら、この単純な延長上では、外挿機能は実装されないのではないか [3]。実際、例えば GPT-3 では 2 桁の足し算は zero-shot で 100% の精度だが、多数桁の足し算や 2 桁の掛け算など複雑

なタスクでは精度が低下する [4]。

多くの応用目的のために深層学習が急激に発展してきている最中、一方では深層学習単独の技術の限界も指摘されている [3] [5]。それらの限界を乗り越えるため、特にこの数年では、システム 1 とシステム 2 [6] を論拠とした議論と、それに関する研究が注目を集め始めている。標語的には、システム 1 が既存の AI システムに対応し、素早く無意識化で実行される認識のような役割を担い、システム 2 がよりゆっくりと動作する論理的・演繹的操作を担うとされている。システム 2 に対応するアルゴリズムの解明と実装、またはシステム 1 と 2 の融合は開かれた問題である。

こうした背景の下、近年の神経科学の知見を活用し、汎用人工知能に向けた研究の方向性として、いくつかの提案がなされている。例えばグローバルワークスペース理論 [7] は、マルチモーダルを扱うための有望なアーキテクチャとして検討され始めている [8] [9]。さらに、GLOM[10] や Perciever[11] など、新たなアーキテクチャも提案されている。マルチモーダルを扱うシステムの研究の進展は、シンボルグラウンディング問題を部分

*連絡先：株式会社アラヤ
東京都港区赤坂 1-12-32 アーク森ビル 24 階
E-mail: fujisawa@araya.org

的に解決するかもしれない。こうした試みは、必ずしも直接的に外挿能力を開発する方向性ではないかもしれないが、これらの提案モデルの汎用性の評価項目の1つとして、外挿能力の測定は有力な候補である。

本稿では、足し算を始めとした四則演算のような初等算術が、外挿能力の測定として有用である可能性について議論する。より具体的には、任意の自然数に対する足し算を実行するのに十分と考えられる少数のデータを訓練データとして用意し、多数桁の演算を要求するテストデータでもって精度を測定するというタスクを提案する。初等算術は、教育を受けた人間の多くが習得できるほど単純ではあるものの、それを対処するシステムにとって非自明な問題をいくつか含んでいる。

上述した汎用人工知能に向けた試みの多くは、Atariやgrid worldといった簡易なゲームを環境とする強化学習の文脈で実験されることが多い。これらの多くのモデルは個別の環境で実験されるためにモデル間の比較が行いにくい。また、汎用性、特に外挿能力を獲得しているかを明示的に示す形にはなっているとは言い難い。我々の提案するベンチマークは、単純な設定のために明確になる。

ベンチマークの構成の単純さは、AIシステムに不足する機能についての見通しを良くするだけでなく、データの生成、難易度調整や問題の拡張、帰納バイアスの特定、新規タスクに対する知識の活用の評価を容易にするといった利点がある。また、プログラム生成のような計算機科学における理論的な視点からの議論も可能となる。

多くのプログラム言語では予め定義された関数によって、非常に容易に初等演算を実行できる。そのため初等算術をニューラルネットワークのような多数パラメタからなるシステムに解かせる直接の応用的な利点は少ない。しかし、小学生が習得するようにシステムが初等算術を学ぶということは、そのシステムの外挿能力の測定、またそれを通じた汎用知能の開発に重要な貢献をすると期待している。

本稿の構成について述べる。2節の関連研究では、先行研究として既に行われている初等算術に対する深層学習の適用、Out of Distribution (OOD), 関連するベンチマークとモデルについて述べる。3節では外挿について整理し、我々の使用方法と関連する仮説を述べる。4節では初等算術ベンチマークを提案する。5節の議論においては、強化学習と帰納プログラミングそれぞれとの関連、ベンチマークの脆弱性、訓練データの量についての課題について述べる。6節ではアーキテクチャの構想を示し、最後の7節では結論を述べる。

2 関連研究

2.1 初等算術に対する深層学習の適用

初等算術に対する深層学習の適用は、本稿が初出ではない。近年の初等算術をニューラルネットワークに解かせる試みの多く [12] [13] [14] [15] は、演算に特化したモジュールをレイヤーや活性化関数に組み込むことで精度改善を図っている。[16] では、これらのアーキテクチャを比較している。

我々がより注目したいのは、演算特化の帰納バイアスを排除し、汎用性を目的としたアーキテクチャ設計である。この方向性の議論はあまり多くは見られないが、例えば [17] では、新たなモジュールを導入するのではなく学習アルゴリズムを変更することで、複雑な演算に対応する方法を提案している。

MNIST の画像の系列が与えられたときにそれら全てを足し上げるタスク [12] や、2桁以下の数字を訓練データと使用し3桁から6桁のテストデータで精度を測るタスク [15] などが考えられている。

2.2 Out of Distribution (OOD)

OOD は、古典的な教師あり学習設定ではない、訓練データとテストデータが異なる分布から得られるという仮定の下で、テスト時にも機能するモデル構築、またはその評価を試みる分野といえる。

X を特徴空間、 Y をラベル空間としたときに、古典的な教師あり学習では、前提として $P_{train}(X, Y) = P_{test}(X, Y)$ 、つまり、訓練データとテストデータは同一の分布から独立にサンプリングされるという独立同分布 (i.i.d.) の仮定を置いている。しかしながら、機械学習モデルを現実の場面に適用する場合には $P_{train}(X, Y) \neq P_{test}(X, Y)$ とみなせる場面は少なくない。例えば、訓練データとして日本で取得された車載動画に対し、テストデータとして海外で取得されたものを用いる場合、大まかな構成や出現するものなどは変わらないものの、信号や看板の形状、建物の外観、車線などに異なる特徴が見られるだろう。訓練時とテスト時の分布のずれの定式化や、それをどのように特徴づけるかは未解決の課題とされている [18]。

使用されるデータセットは、実世界の画像データ [19] だけでなく、分布の変化を人工的に再現するために人工データ [20] が用いられることも多い。これら両方のデータを使って、OOD に対処する手法の有効性を評価することが重要であるとされている [18]。

2.3 関連するベンチマーク

[21] では、一般化スキルの測定にあたっては、開発者がモデルに入れ込む帰納バイアスの特定とその評価が重要であるという考察の下、アルゴリズム的情報理論に基づいた一般化スキルの定義を行っている。また、従来のスキル偏重のベンチマークを例に挙げ、ある特定のスキルの精度を測ることは必ずしも汎用知能の開発のためにならないと警鐘を鳴らしている。一般化スキルを測定するためのベンチマークとして Abstract Reasoning Corpus(ARC) を提案している。

Simplified version of the CommAI Navigation tasks (SCAN)[22] は、いくつかの特定の語のみを含む文が与えられたときに、対応する一連の行動を予測するタスクである。[23] では SCAN を分析し、その設計思想を補完するための付加的なデータセット NACS を提案している。人間であれば、全く見たことのない単語であっても、いくつかの例の提示により、それが動詞であることが分かれば、他の動詞と同じように用いることができる。これらのベンチマークは、既知の知識を活用することを要求することで、モデルの compositionality を評価することを目指している。compositionality とは、新たな表現を基本的な要素の組み合わせによって構成すること [24] であり、現在の深層学習技術をはじめとした AI システムに欠けていると言われる要素の 1 つである。

2.4 関連するモデル

[25] で提案された meta seq2seq は、SCAN を部分的に解決している。

DreamCoder[26] は、さまざまなドメインごとに、関数型プログラミング言語で記述される Domain Specific Language(DSL) の primitives を入力とすることで、それらを組み合わせたライブラリを構築する。

3 外挿とは何か

外挿とは、訓練データと異なる領域や異なる性質のデータに対する予測のことではあるものの、いくつかの異なる文脈で用いられている。本節ではそれらを整理するとともに、我々の使用する意味について述べる。

3.1 これまでの使用方法

最も素朴な使用方法は、内挿に対する対義語としての外挿だろう。得られたデータ点に対して関数フィッティングを行う際に、データ点の存在する区間で曲線を当てはめることが内挿であり、データ点の存在しな

い区間において関数を当てはめることが外挿である。例えば [27] では、この意味において、活性化関数として ReLU を用いたニューラルネットの外挿を評価している。

データ点が 1 次元空間上に分布しているときの区間の定義は明らかであるが、2 次元以上のときにデータ点が存在する区間（もしくは領域）をどのように定義するかは、データが密に分布していない限りは、あまり明らかではない。[28] では、訓練データの張る凸包の内部を内挿領域、それ以外を外挿領域と定義している。

Out of Distribution はその定義からいって、分布外のデータ、つまり訓練データにないデータ点に対する予測について議論していることから、これも外挿の一種と捉えられる。ここで扱われるデータは画像や音声、文章などの高次元データが念頭にあることが多い。画像については、1 つ 1 つの画像をデータ点とみなしたときに多様体仮説 [29] に基づいて連続的な空間を考えられそうではあるが、高次元データに対する内挿と外挿の区別はあまり明らかではない。

初等算術や SCAN には少数の離散的な記号のみが現れる。1 つのデータ点はそれらの組み合わせで表現される。例えば初等算術において、1 桁から 3 桁の数を訓練に使用した場合には、1 桁から 3 桁の数に対する予測は内挿、4 桁以上の数に対する予測は外挿として扱われている。

3.2 我々の使用方法と仮説

我々の使用方法においても、前項で挙げた例と同様、外挿を訓練データにない領域に対する予測のこととして用いる。

多くの議論はデータ点のみに焦点を当てており、それらを発生させる法則やメカニズムを含んだ議論はあまり見られない。しかし、外挿を成功させるには、データ点を発生させる背後にある法則の存在が重要ではないだろうか。もし仮に背後に全く法則がないのであれば、人間による外挿も適当な予測とはならないだろう。そこで我々は、背後に潜む法則（ルール）にも焦点を当てたい。

人間による法則理解には、ほとんど直感的に行われるものから、多少の論理的ステップを要するもの、科学的手法に基づく非常に高度なモデルまでいくつかの異なるレベルがある¹。自然言語や我々の思考プロセスが助けとなっていることを疑う余地は無い。しかし、低次のレベルであれば、自然言語の寄与する割合は必ずしも大きくないだろう。

¹ 低いレベルから順に例えば、直観に基づく投げられた鉄球や歩行者の軌道予測、複数桁の足し算や掛け算といった初等算術、物理学におけるモデルを利用した軌道予測など。

以上の背景を受けて、ルールを抽出し運用することができるアルゴリズムの構築を念頭に、初等算術ベンチマークを提案する。

²この設定は 4.2 で述べるように問題の複雑度についての容易な拡張が可能である。

表記に関する一般化や複雑化も可能である。 N 進法への変更や、使用する記号のシャッフルや変更による複雑化が考えられる。これらの変更は問題の本質的な難易度を変えるものではないが、ドメイン適応の課題の1つとみなせるだろう。

更には、SCAN における新たな動詞の導入に対応する問題設定も可能である。例えば、数字の3を表す記号として3を用いていたところに、全く同じ使用方法をされる別の記号 (例えば A など) を導入すればよい。

難易度の変更方法としては、上述した演算子の導入、オペランドの変更、 N 進法への拡張、表記のシャッフルや変更以外にも、項数を増やすこと、前置記法や後置記法などの記法の変更、訓練データの量を変化させること、これら全ての組み合わせが考えられる。

5 議論

5.1 強化学習との関連

人間が初等算術を解く場合、問題は通常、複数ステップの手続きによって解決される。その手法が筆算でも珠算でも何であれ、ある桁に注目し何らかの手続きを実行し記憶するという一連の手続きと、それらの手続きの積み重ねによって、最終的な答えに到達する点は同じである。もし我々が、筆算などに対応するような複数ステップを実行し得る何らかの環境を事前に与えるのであれば、強化学習の枠組みとみなせる。それらの環境として可能な状態や行動を予め定義することで、その制限の中でシステムを訓練できる。しかし、この場合でも、訓練時に現れないような多数桁の足し算を正確に実行するには、やはり既に現れた例に還元するなどの機構が必要となるだろう。

初等算術ベンチマークにおける挑戦的な課題の1つは、このような環境を所与とせず、システム自らが環境を構築することを要求することである。算術に限らないさまざまな問題に適用可能な内部モデルを、訓練によって獲得するような仕組みが必要なのかもしれない。

強化学習では高次元データである画像を入力とすることが多い。従って、システムの動作が期待したものとならないときに、高次元データの認識によって失敗しているのか、そのオブジェクトの配置などによる組み合わせに対処できずに失敗しているのかの切り分けは容易ではない。一方で、初等算術においては、独立な要素数は限られており、かつ、低次元なベクトルで表現することも可能であるために、組み合わせによるデータの複雑度のみを扱うことが可能である。

5.2 帰納プログラミングとの関連

提案のベンチマークでは初等算術を扱っており、かつ、有限個の例が教師データとして与えられる。一方で、帰納プログラミングは、有限個の入力と出力のペアが与えられたときに、それらを入出力とするアルゴリズムを求めるタスクである。これらの事実から、初等算術ベンチマークが単に初等算術を実行するプログラムを求めるタスクであるかのように思われるが、そうではない。違いは、特定のプログラミング言語とその実行環境が予め用意されていない点である。チューリングマシンを動作させることと、チューリングマシンである実行環境それ自体を内部に保持することは異なる。従って、アルゴリズムを求めることだけでなく、その実行環境も獲得されることが要求される。何らかのプログラミング言語を導入するか、ニューラルチューリングマシン [30] やチューリング完全であることが知られている RNN [31] のような機構を用いることが考えられる。

別の留意点は、帰納バイアスの導入についてである。帰納プログラミングにおいては、ある特定のタスクをうまく記述することのできる言語である DSL を導入することによって探索効率を向上させる。しかし前述の通り、DSL として初等算術に特化したものを採用することは望ましくない。

帰納バイアスに関連する事実として、有限個の入出力関係を満たすアルゴリズムは無限に存在することが知られている。実際、例えば、足し算を実行するアルゴリズムとして、小さい桁から足し上げと繰り上げ処理を順次実行していくものが考えられるが、有限個の例に対処するためにはループ処理の回数は無限である必要はなく、十分大きな整数としておけばよい。最大桁に到達したら実行を止める、という条件は必ずしも必要ない。従って、帰納バイアスを全く導入しないときに、人間が期待するアルゴリズム、人間にとって理解しやすいアルゴリズムが得られるとは限らない。無限に存在する解³に対して優劣をつけるには、アルゴリズムの記述長さ、実行時のステップ回数といった何らかの効率性を訓練時に加味することが考えられる。

5.3 ベンチマークの脆弱性

通常のベンチマークでは、人間のあらゆる知識やアルゴリズムなどを導入したとしても困難な課題が提案される。そこで設定された精度指標を向上させる限り、モデルにどんな帰納バイアスを入れることも許される。この観点からすると、我々のベンチマークは対極に位置する。つまり、開発者が直接的に四則演算を実行す

³有限個のデータ点に対する多項式フィッティングと同様。

るためのプログラムを導入するなど、演算に関わる特定の知識を入れ込むことによって容易に解かれる。難読化は、この問題を部分的に緩和するが、本質的な解決策とは言えない。モデルの是非を評価するには、モデルの挙動だけでなく、初等算術のための帰納バイアスが入っていないことをアルゴリズムの上でも確認する必要があるだろう。

この点については、我々がどんなデータに対しても機能する外挿能力の実装を目指しているのだとすると、他のベンチマークを利用しても同様の事情となる。通常は、特定スキルの実行に主眼が置かれており、そのために開発者がどんな知識を導入することも許されるからである。一般に、課題の複雑さに応じて解法となるアルゴリズムも複雑となることで、どのような帰納バイアスが入っているかを見極めるのは難しくなる。一方で、初等算術であれば、足し算や四則演算ならではのドメイン知識がどれくらい含まれているのかを反省することは比較的容易である。

直接的に演算に関わる帰納バイアスを意識的に排除していくと初等算術をシステムに解かせることは十分難しくなる。初等算術ベンチマークは単純な設定であるために、抽象化、新たな例の既知への還元、既に得た知識の明示的な利用といった機能が今の AI システムに欠如していることを明確にしてくれる。

5.4 必要な訓練データの量

小学生による足し算の習得は、直接的な例示だけでなく、自然言語による教示や、日常生活における環境との相互作用を通じた事前の知識の活用によって達成されると考えられる。例えば、通常は筆算などのテクニックを伝授されるし、どのような順序で注目するかという指導も、板書における文字を書く中で暗に行われているだろう。つまり、最終的な答えだけではなく、思考プロセスの各ステップも教示されている。

本提案では、機械に対する自然言語による教示は難問であるため、これを回避するために具体的な等式のみを教師データとして採用した。提案タスクのように、具体的な等式のみからなる教示であらゆる演算を達成可能かは明らかではない。特に、より複雑な算術において、どの程度のデータ量が訓練データとして必要であるかは明らかではない。しかし、訓練データの量を変更するのは容易であるため、いくつかのバリエーションを用意し評価すればよい。

6 アーキテクチャの構想

足し算や掛け算の問題を解くために必要な計算ステップは、桁が大きくなるにつれて増加する。1 桁同士の演

算を 1 ステップとみなすのであれば、ともに n 桁の二項の足し算の計算量は $O(n)$ 、掛け算は $O(n^2)$ となる。固定の演算回数ではなく、問題の複雑度に応じて演算回数が変わるようなシステム (例えば [32]) が望ましいのではない。

ニューラルネットは、入力を複雑な非線形の幾何的変換によって出力へと写像する関数とみなせる [5]。この写像はそれぞれの入出力を対応付けているものの、複数のデータ点を明示的に関係づけるようなものではなく、データ点の入出力をそのまま記憶しているようなものだ。これに対して、人間のようにこれまでに見たことのない多数桁の数に対して正しく計算できるシステムは、分析することによって未知の例を既知へと還元しているのではない。これまでに見たことのない数であっても、部分に注目するなどして他の数や既に知っている知識へと関係づけることができているのではない。そこで、さまざまなタスクに適用可能な分析のためのモジュールとして、注目と、注目した対象に対する操作を帰納バイアスとして導入してはどうか。操作とは例えばルールの適用である。還元できない場合には新たなルールとして記憶する。これは思考プロセス、もしくは、タスクのための内部モデルと呼べるようなものだろう。

5.2 で述べたように、有限個の例から導ける可能な解は無数にあるために、内部モデルを構成できたからといって、直ちに外挿領域での予測が確からしくなるわけではない。しかし、内部モデルなしに外挿領域での予測を成功させることは難しいだろう。背後にある法則は、画像認識における特徴量設計が困難であるように、必ずしも言語化し尽くせるものではないかもしれない。しかし、それが有意義なタスクであれば、そこには何らかの秩序はあるはずだ。内部モデルが背後にある法則を反映したものであれば、たとえ予測を誤っていたとしても、何らかの秩序だったふるまいが認められるのではない。

さまざまなタスクに応じて有効に働く内部モデルをシステムが構築するのは、全くの仮定なしには探索空間が広すぎるために困難であろう。人間同士の教示においては、例の提示だけでなく言語や動作なども用いられるが、それらの活用によって初めてタスク固有のモデルが構築されているのかもしれない。そうだとすると、具体的なデータの提示のみからでは、タスクのための良い内部モデルの特定や獲得は望めない。この場合は、近年の言語モデルの活用か、言語や意味理解を含めたこれまでにない教示の仕方が必要となるであろう。一方で、計算量に任せた探索方法を活用することで、初等算術のようにあまり複雑でないようなタスクであれば、例の提示のみからでも期待する動作をするシステムは得られるかもしれない。

7 むすび

本稿では、外挿について整理した後に、初等算術ベンチマークの構想を提案した。提案のベンチマークは規則性理解や外挿能力を測るのに適しており、難易度の変更が容易であり、さまざまなタスクを生成できるなどのいくつかの利点を有している。また、足し算を学んだあとに、掛け算タスクがどの程度容易に解決されるかを評価することで、知識の系統的活用の評価に使用可能である。

足し算は一見すると非常に簡単な問題に感じられるかもしれない。しかし、人間のように無限を認識することができる知能、既に得られた知識の利用でもって新たな問題を解決する知能を探索する上では、格好の問題設定になると期待できる。複雑なデータにたよって何を達成したかを見誤ってしまわないように、なるべく不要な要素を削ぎ落とす方向に特化した初等算術ベンチマークが、人工知能の新規モジュール開発の一助となれば幸いである。

謝辞

山口貴史氏には、有益な議論の機会を頂きましたことを深く感謝いたします。

参考文献

- [1] T. S. Alec Radford, Karthik Narasimhan and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.
- [3] G. Marcus, “Deep learning: A critical appraisal,” *arXiv preprint arXiv:1801.00631*, 2018.
- [4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.
- [5] F. Chollet, *Deep Learning with Python*. Manning, Nov. 2017.
- [6] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011.
- [7] G. A. Mashour, P. Roelfsema, J.-P. Changeux, and S. Dehaene, “Conscious processing and the global neuronal workspace hypothesis,” *Neuron*, vol. 105, no. 5, pp. 776–798, 2020.
- [8] Y. Bengio, “The consciousness prior,” *arXiv preprint arXiv:1709.08568*, 2017.
- [9] R. VanRullen and R. Kanai, “Deep learning and the global workspace theory,” *Trends in Neurosciences*, vol. 44, no. 9, pp. 692–704, 2021.
- [10] G. E. Hinton, “How to represent part-whole hierarchies in a neural network,” *arXiv preprint arXiv:2102.12627*, 2021.
- [11] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, “Perceiver: General perception with iterative attention,” in *ICML*, 2021.
- [12] Y. Hoshen and S. Peleg, “Visual learning of arithmetic operations,” in *AAAI*, 2016.
- [13] A. Neelakantan, Q. V. Le, and I. Sutskever, “Neural programmer: Inducing latent programs with gradient descent,” *arXiv preprint arXiv:1511.04834*, 2016.
- [14] A. Trask, F. Hill, S. E. Reed, J. Rae, C. Dyer, and P. Blunsom, “Neural arithmetic logic units,” in *NeurIPS*, 2018.
- [15] A. Madsen and alexander rosenberg johansen, “Neural arithmetic units,” *arXiv preprint arXiv:2001.05016*, 2020.
- [16] B. Mistry, K. Farrahi, and J. S. Hare, “A primer for neural arithmetic logic modules,” *arXiv preprint arXiv:2101.09530*, 2021.
- [17] B. Nollet, M. Lefort, and F. Armetta, “Learning arithmetic operations with a multistep deep learning,” in *IJCNN*, 2020.
- [18] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, “Towards out-of-distribution generalization: A survey,” *arXiv preprint arXiv:2108.13624*, 2021.

- [19] N. Ye, K. Li, L. Hong, H. Bai, Y. Chen, F. Zhou, and Z. Li, “Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms,” *arXiv preprint arXiv:2106.03721*, 2021.
- [20] B. Aubin, A. Slowik, M. Arjovsky, L. Bottou, and D. Lopez-Paz, “Linear unit-tests for invariance discovery,” *arXiv preprint arXiv:2102.10867*, 2021.
- [21] F. Chollet, “On the measure of intelligence,” *arXiv preprint arXiv:1911.01547*, 2019.
- [22] B. M. Lake and M. Baroni, “Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks,” in *ICML*, 2018.
- [23] J. Bastings, M. Baroni, J. Weston, K. Cho, and D. Kiela, “Jump to better conclusions: SCAN both left and right,” in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 47–55, 2018.
- [24] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” *arXiv preprint arXiv:1604.00289*, 2016.
- [25] B. M. Lake, “Compositional generalization through meta sequence-to-sequence learning,” in *NeurIPS*, 2019.
- [26] K. Ellis, C. Wong, M. I. Nye, M. Sablé-Meyer, L. Cary, L. Morales, L. B. Hewitt, A. Solar-Lezama, and J. B. Tenenbaum, “Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning,” *arXiv preprint arXiv:2006.08381*, 2020.
- [27] K. Xu, M. Zhang, J. Li, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka, “How neural networks extrapolate: From feedforward to graph neural networks,” in *ICLR*, 2021.
- [28] R. Balestriero, J. Pesenti, and Y. LeCun, “Learning in high dimension always amounts to extrapolation,” *arXiv preprint arXiv:2110.09485*, 2021.
- [29] L. Cayton, “Algorithms for manifold learning,” *Univ. of California at San Diego Tech. Rep.*, pp. 1–17, 2005.
- [30] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” *arXiv preprint arXiv:1410.5401*, 2014.
- [31] H. Siegelmann and E. Sontag, “On the computational power of neural nets,” *Journal of Computer and System Sciences*, vol. 50, no. 1, pp. 132–150, 1995.
- [32] A. Banino, J. Balaguer, and C. Blundell, “Pondernet: Learning to ponder,” *arXiv preprint arXiv:2107.05407*, 2021.