

本田財団レポート No. 179

第 40 回 本田賞授与式 記念講演（2019 年 11 月 18 日）

「ディープラーニング革命」

トロント大学 名誉教授
ベクター研究所 主任科学顧問
グーグルリサーチ 工学フェロー

ジェフリー・E・ヒントン 博士

公益財団法人 本田財団

ジェフリー・E・ヒントン博士

トロント大学 名誉教授
ベクター研究所 主任科学顧問
グーグルリサーチ 工学フェロー



■生まれ

1947年12月6日

■学歴・学位

1970 年：ケンブリッジ大学、実験心理学学士号

1978 年：エジンバラ大学、人工知能博士号

■職歴

1978-80 年：カリフォルニア大学サンディエゴ校、博士研究員

1980-82 年：ケンブリッジ MRC 応用心理学ユニット、
リサーチサイエンティスト

1982-87 年：カーネギーメロン大学、助教授&准教授

1987 年～：トロント大学、コンピューターサイエンス学部教授

1998-2001 年：ユニバーシティ・カレッジ・ロンドン、
ギャツビー計算論的神経科学ユニット長

2013 年～：グーグル・ブレイン・チーム、工学フェロー

2016 年～：ベクター研究所、主任科学顧問

■略歴

ジェフリー・ヒントン博士は、これまで 47 年間に渡り人工ニューラルネットワークの研究を続けています。ヒントン博士は、デビッド・ルーマルハートおよびロナルド・ウィリアムズ両氏と共に、ディープニューラルネットワークが複雑なデータの効果的な分散表現を学習できることを示しました。また、テリー・セジュノスキー氏と共に、ボルツマンマシンを開発しています。その他の業績としては、混合エキスパートモデル、ニューラルネットワークの変分推論・変分学習法、ディープ・ビリーフ・ネットワーク、そしてカプセルネットワークの考案が挙げられます。トロント大学の彼の研究グループは、音声認識（2009 年）および物体分類（2012 年）にディープラーニングを用いることで、両技術の革命的なブレイクスルーをもたらしています。

■栄典・受賞歴

1998年：王立協会フェロー

2001年：デビッド・E・ラメルハート賞（認知科学）

2005年：IJCAI人工知能研究優秀賞

2011年：ヘルツバーグゴールドメダル（カナダ、理工学）

2016年：米国工学アカデミー外国人会員

2016年：IEEE/RSEジェームズ・クラーク・マクスウェル・
ゴールドメダル

2016年：BBVA財団フロンティアーズ・オブ・ナレッジ賞
情報通信技術部門

2016年：NEC C&C賞

2018年：ACMチューリング賞（ヤン・ルカン氏および
ヨシュア・ベンジオ氏と同時授賞）

■主要出版物

A Learning Algorithm for Boltzmann Machines:

(with Ackley, D. H. & Sejnowski, T. J.), Cognitive Science, Elsevier, 9 (1): 147-169, 1985 (ボルツマンマシンの学習アルゴリズム)

Learning Representations by Back-Propagating Errors:

(with Rumelhart, D. E. & Williams, R. J.) Nature 323 (6088): 533-536, 9 October 1986 (誤差逆伝播法による表現学習)

A Fast Learning Algorithm for Deep Belief Nets:

(with Osindero, S. & Teh, Y.) Neural Computation, 18 (7): 1527-1554, July 2006 (ディープ・ビリーフ・ネットワークの高速学習アルゴリズム)

Reducing the Dimensionality of Data with Neural Networks:

(with Salakhutdinov, R.R.) Science, 313 (5786): 504-507, 28 July 2006 (ニューラルネットワークによるデータ次元削減)

ImageNet Classification with Deep Convolutional Neural Networks:

(with Krizhevsky, A. & Sutskever, I.), NIPS 2012.
Curran Associates Inc.: 1097-1105, 3 December 2012 (深層畳み込みニューラルネットワークによる ImageNet 分類)

The Deep Learning Revolution

Geoffrey Hinton

Google Brain Team
&
Vector Institute

初めに、このような栄誉ある賞を受賞できたことを大変嬉しく存じます。本田財団と選考委員会の皆様に感謝申し上げます。また、この式典にご来場いただいた多くの皆様にも感謝申し上げます。

本日はディープラーニングの歴史についてお話したいと思います。この分野における近年の成果についてお話するつもりはありません。私は教授ですので、ディープラーニングの仕組みについて主に解説させていただきます。

Two paradigms for Artificial Intelligence

The logic-inspired approach

The essence of intelligence is using symbolic rules to manipulate symbolic expressions.

We should focus on reasoning.

The biologically-inspired approach

The essence of intelligence is learning the strengths of the connections in a neural network.

We should focus on learning and perception.

図-1

〈図-1〉人工知能には2つのパラダイムがあります。知能とは論理的思考に基づくものであり、知能の本質とは推論の規則を用いてシンボリックな表現を操作することであると考える論理的アプローチ。もう1つは生物学的アプローチ。論理的アプローチとは概念がまったく異なります。生物学的アプローチでは、知能の本質とは学習、すなわち、ニューラルネットワーク内の結合強度を学ぶことであると考える、論理的思考ではなく学習と知覚に焦点を当てています。

Two views of internal representations

- **Internal representations are symbolic expressions.**
 - A programmer can give them to a computer using an unambiguous language.
 - New representations can be derived by applying rules to existing representations.
- **Internal representations are nothing like language.**
 - They are large vectors of neural activity.
 - They have direct causal effects on other vectors of neural activity.
 - These vectors are learned from data.

図-2

〈図-2〉人工知能におけるこれら 2 つのアプローチ。内部表現に関する概念もまったく異なります。シンボリックアプローチでは、内部表現を一連のシンボルのようなものであると考えます。それはプログラマがコンピュータに入力可能なものであり、規則を設けて操作すれば新たな表現を導き出すことができるというものです。生物学的アプローチにおける内部表現は言語のようなものではなく、人間の脳内で行われている膨大なニューロン活動のパターンであると考えます。言語とは単に脳内に情報をインプットし、脳内から情報をアウトプットする手段の一つです。しかし、脳内では言語という形で表現されているわけではなく、実際にはニューロン活動の大きなベクトルなのです。これらのベクトルは他のベクトルに対して直接的な因果効果を持ち、データから学習されます。プログラマがベクトルを入力するわけではありません。

Two ways to make a computer do what you want

- **Intelligent design:** Figure out consciously exactly how you would manipulate symbolic representations to perform the task and then tell the computer, in excruciating detail, exactly what to do.
- **Learning:** Show the computer lots of examples of inputs together with the desired outputs. Let the computer learn how to map inputs to outputs.
 - This requires a general purpose learning procedure that only has to be programmed once.

図-3

〈図-3〉コンピュータに何かを命令する場合、これら 2 つのアプローチはまったく異なる方法を用います。私が「インテリジェントデザイン」と呼ぶアプローチがあります。インテリジェントデザインでは、AI の教授がコンピュータの表現を決定し、それらを手作業で入力します。そしてタスクを完了させるためにその表現をどのように操作すればよいかを

正確に導き出し、コンピュータに詳細な実行方法を指示するのです。いわゆるプログラミングです。

このようなプログラミングとはまったく異なるアプローチがあります。まず、プログラミングを一度行い、学習方法を教えます。そして、各特定のタスクに関する複数の例を示すと、コンピュータは学習アルゴリズムを駆使して入力情報から出力情報を導き出す方法を理解するのです。つまり、一般的な学習手順をプログラムしたということです。そして、これがこれからお話するバックプロパゲーションと呼ばれるものです。この一般的なアルゴリズムは、音声認識や物体認識といった特定のタスク処理に採用されています。また、このアルゴリズムを異なるデータに適用すれば、異なるタスクを解決するための学習を行うのです。

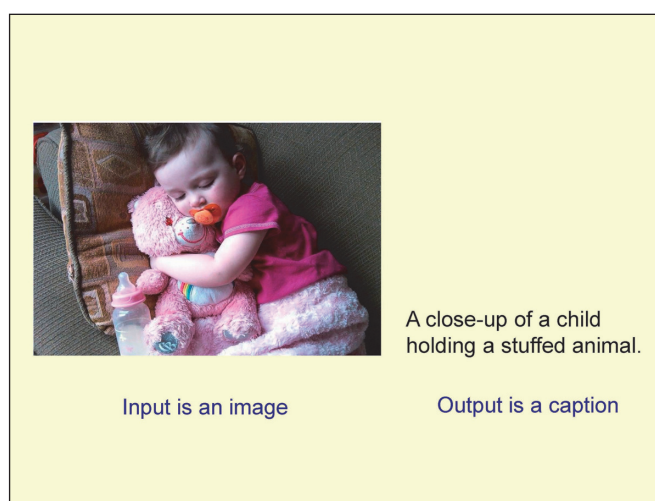


図-4

〈図-4〉シンボリック AI で解決することが非常に難しかったタスクの一例として、入力された画像の内容をコンピュータが一連の言葉を用いて表現することが挙げられます。コンピュータが認識している画像というのは、100 万ピクセルにも及ぶであろう赤、緑、青の値であり、その強さの度合いにしか過ぎません。この画像を説明する適切な一連の言葉は、「アップで撮影されたぬいぐるみを抱えている子供」といえるでしょう。この一連の言葉は実際にコンピュータによって生成されました。

The central question

- Large neural networks containing millions of weights and many layers of non-linear neurons are very powerful computing devices.
- But can a neural network learn a difficult task (like object recognition or machine translation) by starting from random weights and acquiring all of its knowledge from the training data?

図－5

〈図－5〉数百万もの結合を有し、それぞれの結合が調整可能な重みを持ち、入力と出力の間に数多くのニューロン層が存在する膨大なニューラルネットワークは非常にパワフルな学習装置です。しかし、入力された画像を一連の言葉を使って説明するといった難しい問題を、ランダムな重みをかけたりシンプルな学習アルゴリズムを用いたりするだけで解決することができるかどうか、ニューラルネットワークの重要な課題なのです。多くの人にとっては、とても馬鹿げているように思えたでしょう。問題解決方法についての詳細をコンピュータにプログラミングしない限り、そのような難しい問題を大規模ニューラルネットワークに解決させるのは不可能だと彼らは考えていました。確かに問題解決方法をいくつかプログラミングする必要があるのですが、基本的にこれらのネットワークは問題を解決する方法を自己学習するのです。

The obvious learning algorithm

- Early researchers like Turing and Selfridge proposed that neural networks with initially random connections could be trained by reinforcement learning.
 - This is extremely inefficient.

図－6

〈図－6〉大規模ニューラルネットワークのための簡単な学習アルゴリズムは、まずネットワークにランダムな重みをかけ、その中の重みを 1 つ変更し、ネットワークの動作が良くなるかどうかを確認することです。進化論を学んだ人や、チューリング、オリバー・セルフリッジといった研究者には簡単に理解できるでしょう。つまり、ネットワークに入力の

例をいくつか与え、どのような出力が得られるかを確認した後、重みを 1 つ変更して、動作が改善されたかどうかを確認するのです。

今となっては非常に非効率的なアルゴリズムです。なぜなら、ネットワーク経由で例を与えなければならない、それがすべての結合に影響を及ぼすからです。数百もの例をネットワーク経由で与え、どのように動作するかを確認する。また 1 つ重みを変更する。その変更によって動作が改善されたとき、1 つの重みに対する適切な変更量のはじめてわかるということなのです。したがって、このアルゴリズムは最終的にはうまく動作し、機能的なネットワークを構築するのですが、極めて非効率的なのです。

Perceptrons

- ~1960: Rosenblatt introduced a simple, efficient learning procedure that could figure out how to weight features of the input in order to classify inputs correctly.
 - But perceptrons could not learn the features.
- 1969: Minsky and Papert showed that perceptrons had some very strong limitations on what they could do.
 - Minsky and Papert also *implied* that having deeper networks would not help.
- 1970s: The first neural net winter

図-7

〈図-7〉1960 年頃、ローゼンブラットは単純なニューラルネットで機能するはるかに効率的な学習アルゴリズムを発表しました。この学習アルゴリズムが学ぶことは、さまざまな特徴にどれだけの重みをかけるかを決定することです。いくつかの特徴はプログラマが設計しなければいけません。しかし、今ではニューラルネットワークが各特徴にどれだけの重みをかけるかを判断し、意思決定を行うのです。

1969 年、この種のニューラルネットの機能は非常に限定的であることがミンスキーとパパートによって指摘されました。これは比較的効率の良い学習アルゴリズムでしたが、手作業で設計された特徴が 1 層しかない場合、ほとんどのことができなかったのです。つまり、多くの特徴層が必要だということです。ミンスキーとパパートも多くの特徴層を設けた場合の機能についてまでは言及していませんでしたが、それらをどのようにトレーニングすればよいか誰もわからなかったため、ニューラルネットワークは役に立たないも同然でした。そして私たちはニューラルネットワークの冬の時代を初めて迎えることになったのです。

Back-propagation

- 1980s: The back-propagation procedure allows neural networks to design their own features and to have multiple layers of features.
 - Back-propagation created a lot of excitement.
 - It could learn vector embeddings that captured the meanings of words just by trying to predict the next word in a sequence.
 - It looked as if it would solve tough problems like speech recognition and shape recognition.

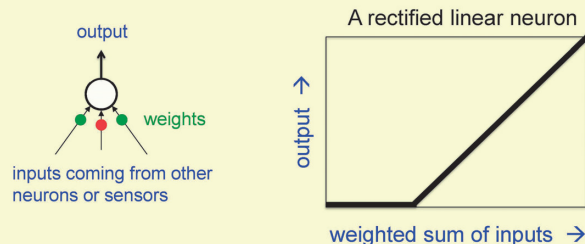
図－8

〈図－8〉1980年代になると、多くのグループによってバックプロパゲーションアルゴリズムが開発されました。バックプロパゲーションアルゴリズムの考案者の一人であるデビッド・ルーメルハートと共に研究できたことは幸運なことでした。ニューラルネットワーク内の重みを調整するこのアルゴリズムによって、ニューラルネットワークは多くの層からなる特徴を学習できるようになったのです。甘利さん（理化学研究所脳神経科学研究センター名誉研究員の甘利俊一博士）が先ほどのスピーチで仰っていたとおり、これは衝撃的なことでした。

初めの頃、入力された一連のシンボルの各シンボルをベクトルに変換する方法が学習可能であることを発表しました。一連のシンボルのうち、最初の2つのシンボルを表すベクトルから、3つめのシンボルを表すベクトルを予測することができたのです。すなわち、シンボリック入力を内部ベクトルに変換することを学習するニューラルネットワークの一例だったのです。当時、私たちはこの手法で多くの問題を解決できると思っていました。今日でもこの手法を用いて問題を解決することができます。しかし、どういうわけか当時はうまく機能しませんでした。

What is an artificial neuron?

- We make a gross idealization of a real neuron so that we can investigate how neurons can collaborate to do computations that are too difficult to program such as:
 - Convert the pixel intensity values of an image into a string of words that describe the image.



図－9

〈図－9〉バックプロパゲーションアルゴリズムとその仕組みについてもう少し詳しく説明

しましょう。まず、人工ニューロンとは何かを説明しなければなりません。脳内にあるニューロンはとても複雑なデバイスです。ニューロンを単純化し、ニューロンがどのように相互作用して難しい問題を解決しているかを解明することが私たちの目的でした。もちろん、何かを単純化しようとすればいくつかのディテールが失われてしまうというリスクが生じます。しかし、重要なのは研究に必要なディテールを残しつつ単純化することなのです。そして、この研究は人工ニューラルネットワークの希望でした。

まず、シグモイドニューロンと呼ばれる特定の種類のニューロンを使用しました。シグモイドニューロンは他のニューロンや感覚機能から送られてきた入力を受け取ります。それぞれの入力は重みを持ち、その重みが増加するとニューロンの動きが変わり、出力を発生させるのです。図内では重みをドットで示しています。一種類のニューロンを対象とした研究を始めてから約 20 年経った 2010 年頃、私たちは別種類のニューロン研究に着手しました。このニューロンは非常によく機能しました。このニューロンは入力が閾値未満であると出力を発生させませんが、ひとたび閾値を超えてしまえば、入力を増やすにつれて線形的に増加する出力を発生させるのです。

つまり、このようなニューロンを使って、入力線上のアクティビティに結合の重みを乗じ、それらの値の和をこの関数に当てはめるのです。和の値が大きければ、出力も大きくなります。もし和の値が小さければ、出力はゼロになるでしょう。これが基礎的な演算要素です。

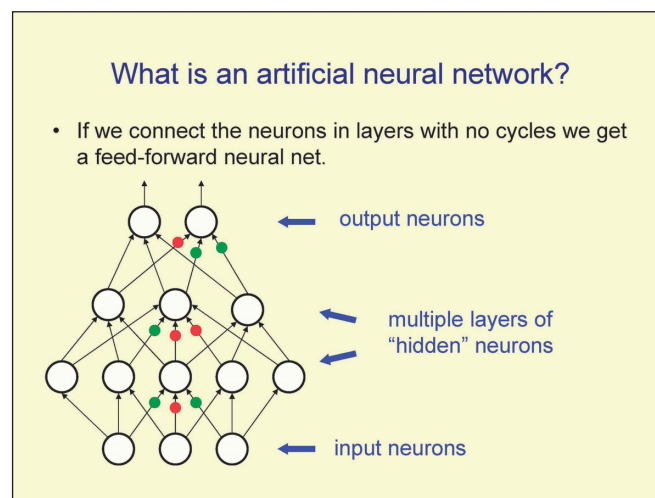


図-10

〈図-10〉次に、これらの要素をネットワーク内に組み込みます。最も単純なネットワークは、フィードフォワードネットワークです。この図は複数の層を形成するニューロンです。こちらは入力です。ピクセルの明度を示したりします。こちら側はアウトプットです。画像がある特定の 카테고리に属している確率を示したりします。例えば、ある何かが猫であるか犬であるかを判別したいとします。これは猫かもしれませんし、こちらは犬かもしれません。すべての結合には重みがあります。そのため、画像を入力すると、重みに応じて変化する一定の活動レベルがニューロン内で得られます。重みを変更するということは、

ニューロンが活性化する環境を変化させていることであり、どのような特徴を検出するかを変化させているとも言えます。

例えば鳥を認識させたい場合、1 層目のニューロンには直線を認識させ、2 層目のニューロンには鋭角に繋がる直線を認識させます。もしそのような 2 つのニューロンが揃えば、つまり、2 本の直線が鋭角に繋がっていれば、それが鳥のくちばしであることを示すエビデンスとなるのです。さらに、目のような閉じた円形を認識するニューロンや、正しい空間関係を認識するニューロンがあれば、鳥を認識するヒントになるのです。

結合を学習させ、中間層にあるニューロンを特徴検出器に変化させるということが、これらのネットワークの考え方です。ネットワークを通過するにつれて、特徴検出器はより複雑な特徴を検出するようになります。「これは輪郭だ」とか、「これはくちばしのようだ」といった具合に検出するのです。そして、より複雑な特徴のエビデンスが十分に得られたら、すべてのクラスがアクティベートされます。当然、私たちが使っているニューラルネットワークはこれよりもずっと深くより複雑なものですが、これが基本的な原理です。

問題は、これらすべての結合強度をどのように学習するのかということです。先ほど、とてもシンプルなアルゴリズムについて説明しました。とてもシンプルなアルゴリズムを使う際は、初めにランダムな重みをかけ、多くの例を試します。そして予測の精度を確認し、重みを 1 つ変更します。

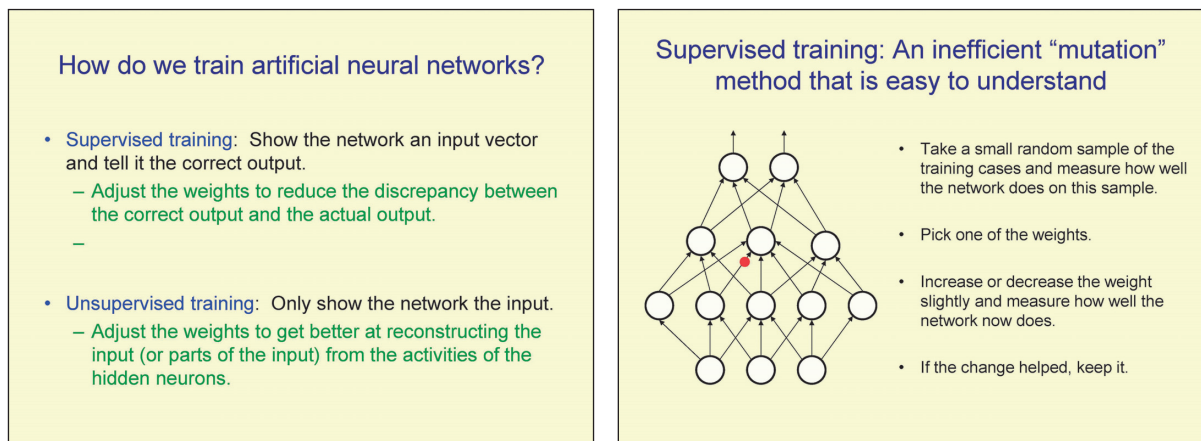


図-11

〈図-11〉例えば、この部分の重みを変更するとします。そうしたら、ネットワークの動作が良くなるか悪くなるかを確認するのです。もし動作が改善したのであれば、その変更はそのまましておきます。その後、より多くの例を試し、また重みを 1 つ変更します。このようにして、それぞれの重みの変更を延々と繰り返していきます。最終的には数十億個もの例をネットワークに入力しなければなりません。人の寿命は約数十億秒と言われているので、人間が作業をしても時間が足りないのです。

バックプロパゲーションアルゴリズムは、各結合強度の変更がどのように正解率を改善させるかを明らかにしますが、その計算をネットワーク内にあるすべての結合強度について同時に行うのです。

The backpropagation algorithm

- Backpropagation is just an efficient way of computing how a change in a weight will effect the output error.
- Instead of perturbing the weights one at a time and *measuring* the effect, use calculus to *compute* the error gradients for all of the weights at the same time. We can do this because we know how changing a weight will change the output.
 - With a million weights, this is more efficient than the mutation method by a factor of a million.

図-12

〈図-12〉 進化的アプローチのように 1 つずつ結合強度を変えて効果を測定するのではなく、ネットワークにフォワードパスを行なった後、バックワードパスを行います。情報を逆方向に送り、重みの変更がどのようにエラーを変化させるかということを、ネットワーク内にあるすべての重みに伝えるのです。つまり、導関数の計算です。重みを変更すると、エラーがどれだけ速く変化するかを求めます。バックプロパゲーションのポイントは、ネットワーク内にあるそれぞれの重みに対して、すべての導関数を同時に計算することができるということです。100 万個の重みを持つネットワークがあったとしても、進化的アルゴリズムやシンプルな突然変異アルゴリズムと比べて 100 万倍も効率的なこのアルゴリズムを使えばよいのです。非常に大きな差を生み出します。

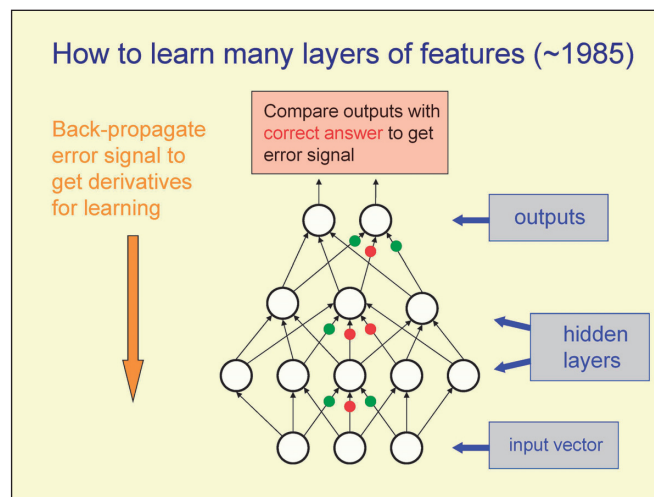


図-13

〈図-13〉 バックプロパゲーションでは、いくつかの入力を隠れ層に送ると、現状の重みで得られるさまざまな回答の確率が分かります。そして、その確率と正解を比較するのです。この入力猫かもしれないので、これが正解です。他の入力犬かもしれないので、それが正解でしょう。次に、出力された回答と出力してほしい回答との誤差を取ります。そして、同じ重みをかけた同ネットワークに信号を逆方向に送ることで、それぞれの重み

に対する計算を同時に行うことができ、重みの変化がどのような改善をもたらすのかを求めることができるのです。その後、ネットワークのパフォーマンスが改善するよう、それぞれの重みを少しずつ調整します。この作業を何度も繰り返せば、ネットワークはどのような指示にも対応できるようになるでしょう。十分に大きなネットワークを構築することができれば…の話ですが。

Stochastic gradient descent

- The main discovery of neural nets so far is that dumb stochastic gradient descent (SGD) works much better than anyone expected.
 - You don't need to get the gradient for a weight on the whole training set in order to make progress. A small "mini-batch" of data is sufficient.
 - Local optima are not a problem in practice.
 - Fancy second-order optimization methods are not needed.
- Its sufficient to use momentum and to decrease the learning rate for weights that have large gradients.

図-14

〈図-14〉したがって、バックプロパゲーションで勾配を求めたら、つまり、ネットワーク内にある全重みの結合強度の変化が、どのような改善をパフォーマンスにもたらすかを計算したら、パフォーマンスが改善するよう同時並行的にすべての重みを更新するのです。そして、ネットワークのパフォーマンスを改善するには、トレーニングサンプルの小さなバッチに対してのみこれを行えばよいのです。

なぜなら、考え得るすべてのトレーニングサンプルをネットワークに入力する時間はないからです。まず、少量の例をランダムに選び、それらの例でネットワークがうまく機能するようトレーニングを行います。そうすることで、他の例に対してはパフォーマンスが悪化することもあります。ですがその後、また別の例をランダムに選び、この作業を繰り返していると、驚くことにネットワークはすべての例に対してとてもよく機能するようになるのです。皆さまはすべての例に対して勾配を計算しなければならないと思われたかもしれませんね。それが標準的な最適化方法なのだろうと。ですがそうではないのです。

しかし、そうは言ったものの、大きなトレーニングセットを使えるのであればその方がずっと効率的である、というのがニューラルネットワークの研究者たちの発見でした。先ほどの確率的勾配降下法ではいくつかの例を使い、重みの変更方法を決定します。さらにいくつかの例を使い、重みを再度変更します。特殊な例を与えたことにより、悪い方向に重みを変更してしまうと、他の例に対して一時的に状況が悪化するのですが、別の例を与えればその悪い変化を打ち消すことができ、良い方向に重みを再度変更することができます。

事実、この確率的勾配降下法は極めてうまく機能するので、とてもパワフルな学習アルゴリズムが手に入ったと当時は思いました。それは事実です。しかし、さまざまなものに

対して学習アルゴリズムを試してみたところ、かなりうまく機能するのですが、他の学習アルゴリズムと比べて特に優れているわけではなかったのです。同程度の機能を持つ学習アルゴリズムはすでに存在していたので、とても失望しました。

A big disappointment

- 1990s: Backpropagation plus SGD works pretty well, but underperforms the expectations of its proponents.
 - It is hard to train deep neural networks. But why?
- On modest-sized datasets some other machine learning methods work better than backpropagation.
 - The second neural network winter begins (in the Machine Learning community)
- Symbolic AI researchers claim that it is silly to expect to learn difficult tasks in big deep neural nets that start with random connections and no prior knowledge.

図－15

〈図－15〉 この手法をうまく機能させるためには、ビッグデータセットと大きなネットワークが必要であることを、当時の私たちはあまり理解していなかったのです。小さなデータセットや小さなネットワークを使っているのは、他の学習手法を凌駕することはできません。規模を拡大しなければ、真価を発揮することができなかったのです。80 年代当時、私たちはその規模を実現させることはできませんでした。ネットワークをトレーニングするための十分なデータセットもパワフルなコンピュータもなかったのです。

ニューラルネットワークの研究者たちは、非常に難しい問題を解決することができるようになるだろうと主張しましたが、当時の私たちにはそれが叶いませんでした。難しい問題を少しは解決することができましたが、あまりうまくいったとは言えません。当然、シンボリック AI の研究者たちは、ゼロからすべてを学習させようとする常軌を逸した信念を持っているからうまくいかないのだと言いました。ランダムな重みを持つニューラルネットワークにすべてを学習させるのであれば、知識をプログラミングする必要があると。難しい問題を解決させる唯一の方法は、知識をプログラミングすることだと。私のようなニューラルネットワークの研究者はそのような主張を信じませんでしたが、その主張が間違っていることを証明することもできませんでした。なぜなら、難しい問題を解決させることができなかったからです。

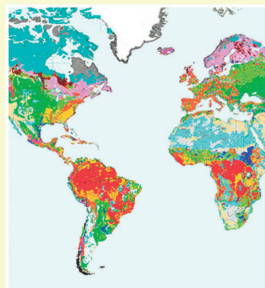
Some really silly theories

- The continents used to be connected and drifted apart! (geologists spent 40 years laughing)
- Great big neural nets that start with random weights and no prior knowledge can learn to perform machine translation.
- The more you dilute a natural remedy, the more potent it gets. (this one really is silly)

図－16

〈図－16〉ここで、本当に「馬鹿げている」理論を他にいくつか紹介しましょう。なぜなら、他の馬鹿げた理論を見ていると、たとえ自分の理論が馬鹿げていると言われても気持ちが落ち着くからです。最も馬鹿げた理論の 1 つに、1920 年代にウェゲナーが提唱した理論が挙げられます。かつて大陸はすべてつながっており、後に分かれていったという理論です。彼はとても有力なエビデンスをいくつか持っていました。

Continental Drift



In 1912, Alfred Wegener proposed that the continents had once been joined, and over time had drifted apart.

He had corroborating evidence: soil types, glacial scrapes, fossils, coal deposits.

His theory was dismissed as silly wishful thinking.

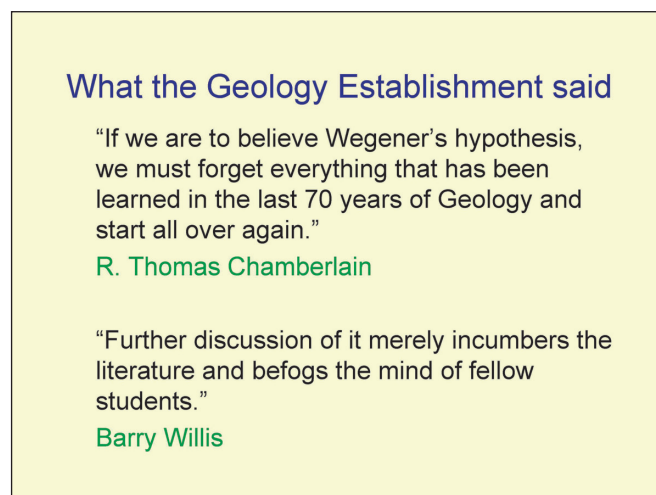
図－17

〈図－17〉南米の海岸とアフリカの海岸の土と岩を比べてみると、とてもよく似ています。同様に、ノルウェーとカナダでも似たような岩が確認されています。他にもあります。ある場所で発見されたとても古い化石と、別の場所で発見された化石を比べてみると、非常によく似ているのです。同じ種類のものが別々の場所で発見されるのは、かつて 1 つであった大陸が分かれ、当時生息していた動物が別離してしまったという以外に説明がつかないのです。

ウェゲナーには他にもたくさんのエビデンスがありました。熱帯地方の岩石層に大きく削られた跡があるというエビデンスです。それは氷河によって岩が引っ張られる際に生じるものなのです。つまり、熱帯地方に氷河が存在していたことを示すまぎれもないエビデンスだったのです。彼は北極に石炭鉱床があったというエビデンスも持っていました。

大陸が動かなければ、説明がつかないことです。

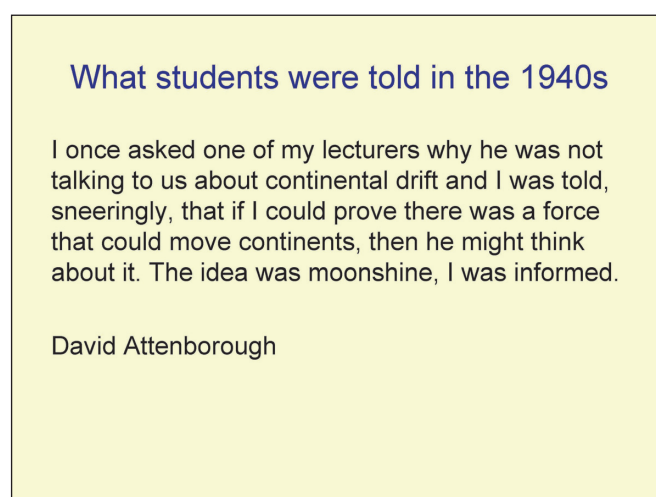
大陸が移動して分かれたと考える以外にこのデータを説明する方法はありません。しかし、地質学者たちは、大陸は不動であるという理論をもとに彼の大陸移動説を端から否定し、40 年もの間、完全にナンセンスな理論として扱っていました。



図－18

〈図－18〉地質学界がこのようなことを言っていたのです。このことを考えると私の心は落ち着きました。子供の頃、大陸移動説をめぐる議論に参加していた父は、生物学的見解に基づき大陸移動説を支持していたのです。さもないと化石の説明がつかなかったからです。とても馬鹿げていると否定され、後に正しいと証明された理論を目の当たりにした幼い頃の私。この体験は私に影響を与えたと思います。

ウェゲナーを信じるなら、地質学について学んできたことをすべて捨てなければならぬと人々は言いました。そして、実際その通りになったのです。彼らは、生徒が混乱をきたすので、このことを教科書に載せるべきではないとも言いました。



図－19

〈図－19〉地球の生物に関する数々の素晴らしい作品を残したデイビッド・アッテンボロー

という人物がいます。おそらく皆さんもご存知でしょう。学生の頃、彼は大陸移動説に興味を持っていました。しかし、大陸を移動させるものなど存在しないので、大陸移動説は完全にナンセンスだと教えられたのです。大陸移動のメカニズムは人々によってすでに提唱されていました。大陸は熱いマグマの上に浮かんでいるというメカニズムです。このメカニズムは 1930 年代にはすでに提唱されていましたが、地質学者たちは「地球は不動である」という当時の支配的な説を固持したため、この理論は真剣に受け取られませんでした。

同様に、人工知能の世界においても人々は知識をプログラミングしなければならないと固く信じていたので、コンピュータがすべてを学習するといった考えは、真剣に受け取られませんでした。私たちが実際に機能するものを提示することができなかったことも、信じてもらえなかった理由の 1 つだったと思います。機能するものを発表し始めた時でさえ、多くの反発がありました。

今世紀初頭、輝かしい科学的功績を残しているにも関わらず、科学会にナンセンスだと一蹴されてしまった人たちがたくさんいました。そのうちの 3 例を紹介したいと思います。1 枚のスライドだけ用意しました。少々お付き合いください。

"Further discussion of it merely incumbers the literature and befogs the mind of fellow students."

- 2007: NIPS program committee rejects a paper on deep learning by *al. et.* Hinton because they already accepted a paper on deep learning and two papers on the same topic would be excessive.
- ~2009: A reviewer tells Yoshua Bengio that papers about neural nets have no place in ICML.
- ~2010: A CVPR reviewer rejects Yann LeCun's paper even though it beats the state-of-the-art. The reviewer says that it tells us nothing about computer vision because everything is learned.

図-20

〈図-20〉これは大陸移動に関する文献からの引用です。こちらは同文献内での発言の一つです。

2007 年、私はある学生とともにディープラーニングに関する論文を NIPS（ニューラル情報処理システム会議）に提出しました。この論文は、NIPS に初めて提出されたディープラーニングに関する 2 つの論文のうちの 1 つです。しかし、この論文はカンファレンスに却下されました。私は却下の理由を尋ねました。彼らは、ディープラーニングに関する 1 本の論文がヨシュア・ベンジオによって提出されており、1 つのカンファレンスでディープラーニングに関する論文を 2 本も受理できないからだと答えました。多過ぎるだろうと。しかし現在、そのカンファレンスを訪れてみると、ほとんどの論文がディープラーニングに関するものです。

2009 年、ヨシュア・ベンジオは機械学習の主要カンファレンスの 1 つにニューラルネットに関する論文を提出しましたが、却下されました。査読担当者は、機械学習カンファレンスでニューラルネットワークを取り上げる余地はないため、査読をする必要などないと言いました。その学会は、ニューラルネットはナンセンスであると強く信じていたため、カンファレンスのために査読なんてしなくなかったのです。

もっとひどいケースがあります。2010 年、ヤン・ルカンは画像セグメンテーションのための畳み込みディープニューラルネットワークに関する論文を提出しました。この技術は当時の最先端技術を凌駕しており、コンピュータビジョンの研究者たちが開発したどんな技術よりも優れていたのです。しかし、論文は却下されました。この論文が却下された理由は、ニューラルネットがすべてを「学習している」からです。もしすべてを「学習している」のであれば、ニューラルネットがコンピュータビジョンについて何かを「教えてくれる」わけではないと、コンピュータビジョンの学会は言ったのです。つまり、コンピュータビジョンにおける唯一の課題は、どのような知識をプログラミングすべきかを「知る」ことだったのです。コンピュータビジョンのソリューションは多くの知識をプログラミングすることだと、彼らは端から決めつけていました。ヤン・ルカンの研究のほうが優れていたにもかかわらず、どのような知識をプログラミングすべきか「教えてくれない」という理由で、カンファレンスには受理されませんでした。より優れていたにもかかわらずです。コンピュータがすべてを学習する、つまり、知識をプログラミングする必要はないという論文の趣旨を、彼らは理解していなかったのです。

The return of backpropagation

- Between 2005 and 2009 researchers (in Canada!) made several technical advances that enabled backpropagation to work better in feed-forward nets.
- The technical details of these advances are very important to the researchers but they are not the main message.
- The main message is that backpropagation now works amazingly well if you have two things:
 - a lot of labeled data
 - a lot of convenient compute power (e.g. GPUs)

図-21

〈図-21〉研究者たち、特に私やヨシュア・ベンジオ、また、フランス人ですが私たちの中では「名誉カナダ人」という扱いのヤン・ルカンといったカナダの研究者たちは、2005 年から 2009 年頃の間にいくつかの技術的進歩を実現させ、バックプロパゲーションをより良く機能させることに成功しました。特筆すべきは、重みの初期化方法に関するより良い方法を発見したことです。ランダムな重みをかけ、いくつかの教師なし学習を行うことで重みを初期化し、バックプロパゲーションを実行する。この方法でシステム全体がより簡単に学習できるようになるのです。この作業の結果、バックプロパゲーションがとてもうまく機能するようになりました。ラベル付きデータがたくさんあり、処理能力の高いコンピュータがあればとてもうまく機能するのです。

Some of the technical tricks that made deep neural nets work better

- Unsupervised pre-training
 - Start by learning a layer of features that are good at reconstructing the input. Then learn a second layer that are good at reconstructing activities of the first layer etc.
 - After designing the features using unsupervised pre-training, use backpropagation to fine-tune them.
- Random dropout of units
 - Make each feature detector more robust by not allowing it to rely on other feature detectors to correct its mistakes.
- Rectified linear units
 - Rectified linear units are more powerful and easier to train than sigmoid units.

図-22

〈図-22〉技術的な要を担うのは、私たちが導入した教師なし事前学習です。ディープネットワークを用意し、出力のエラーをバックプロパゲーションすれば、特徴の層を学習することができます。しかし、最初にランダムな重みをかける場合、この方法はベストな学習方法とは言えません。1 度につき特徴の層を 1 つ学習し、これらの特徴から入力を再構築することができる特徴を学習しようとするのが、最初からより効率的に学習する方法なのです。1 層目の特徴を学習することで、特徴のアクティビティからピクセルの明度を再構築することができます。次に、2 層目の特徴を学習することで、2 層目の特徴のアクティビティから 1 層目の特徴のアクティビティを再構築することができます。これを繰り返していくのです。この方法で多くの特徴の層を学習することができます。このようにして特徴が初期化されれば、バックプロパゲーションを使ったネットワークのトレーニングがとても簡単になるのです。

また、ニューラルネットワークを使用する毎にニューロンのサブセットをランダムに取り除くという、ドロップアウトという手法も導入しました。結果、学習速度は遅くなるのですが、より良い一般化が可能になります。簡単に言えば、ニューロンが他のニューロンを頼ることを止めるのです。ニューロンは互いに少し独立性を保つ必要があります。別のニューロンに頼ってエラーを修正するのではなく、各ニューロンが役に立つ特徴を見つけなければならないのです。

最後に、先ほども言及しましたが、私たちは何年も使ってきたシグモイドユニットの使用をやめ、正規化線形ユニットを使用するようになりました。これは、入力が小さければ出力はゼロとなり、ひとたび閾値を超えれば入力に応じて出力が比例するといったものです。十分に非線形的です。

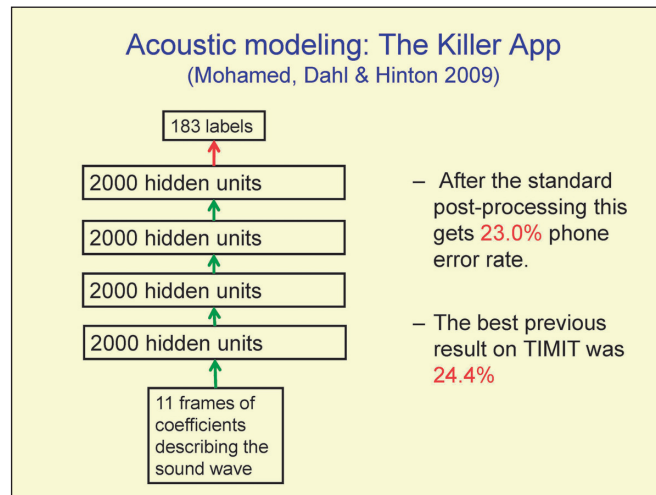


図-23

〈図-23〉これらのものが揃うと、私の研究室に所属する 2 人の学生は、この種の手法を音響モデリングに適用しました。従来の音声認識装置では、音声波形にフーリエ解析などを行い、各周波数にどの程度のエネルギーがあるかを求め、スペクトログラムのようなものを生成します。そのスペクトログラムの中央を見て、「中央のこのパターンの場合、話者はどの音素のどの部分を発声しようとしているのか？」ということを考えるのです。スペクトログラムの残りの部分のコンテキストとも照らし合わせて、どの音素のどの部分を話者が発声しているのかを判別します。当初はニューラルネットのほうが従来の技術と比べて若干優れている程度でしたが、後にニューラルネットの方がずっとうまく機能するようになりました。

まず始めに学生たちは、音波を表す係数のフレームを入力とするニューラルネットワークを開発しました。そのニューラルネットは複数の特徴検出層を備えており、最初はランダムな重みでスタートしましたが、その後は面白い特徴検出器に変化しました。そして話者が発声したと思われる 183 個の音素に対して 183 個のラベルを付けたのです。音波から音素の断片に変化させるため、これは音響モデルと呼ばれています。どの断片が実際に発声されたとみなすか、その決定は他のモデルが行います。それらを繋ぎ合わせて、もっともらしい英語を作ります。

What happened next

- As soon as Deep Neural Networks beat the previous technology, leading speech groups at MSR, IBM & Google developed them further.
- Navdeep Jaitly, a grad student from U of T, implemented our acoustic model at Google during an internship in 2011
 - By 2012, its was being used for voice search on the Android.
 - It gave a big decrease in the word error rate.
- Now the best speech recognition systems all use some form of neural net trained with backpropagation.

図－24

〈図－24〉このようなニューラルネットは他の技術よりもうまく機能したため、生徒たちはさまざまな研究所を訪れ、その使い方を披露しました。彼らは IBM やマイクロソフトも訪れました。ある学生は Google を訪れました。Google を訪れたその学生は、トロントで開発された音響モデルを Google の音声認識装置に組み込んだのです。結果、驚異的な改善がもたらされました。2012 年にトロントで行われた研究では、効率を上げるためにいくつかのエンジニアリング上の変更が加えられ、その成果は Google の音声認識、音声検索、アンドロイドに搭載されました。Siri よりもはるかにうまく機能したため、大きな音声認識研究団体はニューラルネットワークを活用すべきであることを認めざるを得ませんでした。

今では、高性能な音声認識装置のすべてにニューラルネットワークが使用されています。音波を変換して何の音素であるかを示す確率を導き出す前処理工程のためだけに使用されているわけではありません。システム全体が大きなニューラルネットワークなのです。片方から音波を取り込み、反対側から文字に書き直されたものが得られますが、そこにニューラルネットワーク以外のものは存在しません。そしてそれは、端から端まですべてがトレーニングされているのです。

Object Recognition

- The 2012 ImageNet object recognition challenge has about a million high-resolution training images taken from the web.
 - There are 1000 different classes of object.
 - The task is to get the “correct” class in your top 5 bets.
- Some of the best existing computer vision methods were tried on this dataset by leading computer vision groups from all over the world.

図－25

〈図－25〉音声認識機能が Android に搭載された 2012 年、大企業はニューラルネットワー

クが製品に使えることを認めざるを得ませんでした。そのちょうど同時期、私の生徒であるアレックス・クリジェフスキーとイリヤ・サスケバーの2人は ImageNet を使った物体認識コンペティションに参加しました。このコンペティションでは十分に大きなトレーニングセットが用意されていたため、ようやくニューラルネットワークの真価を発揮することができたのです。それまで、ほとんどの物体認識コンペティションでは小さなトレーニングセットしか用意されていませんでした。そのため、これらの学習手法は手作業でプログラミングされた技術に勝ることができなかったのです。十分なデータがなかったのです。しかし、実際に十分なデータを獲得することで、ニューラルネットは驚くべきことを成し遂げることに成功しました。

このコンペティションでは、非公開のテストセットがありました。テストデータのみで機能するアルゴリズムを構築するといった不正を防ぐために、このコンペティションの開催者は非公開のテストセットを用意したのです。そして、彼らが参加者のアルゴリズムをテストデータ上で実行し、性能を確認します。ネットワークやアルゴリズムを渡し、彼らがテストデータ上でそれらを実行し、性能を確認するのです。

Error rates on the ImageNet-2012 competition	
• 2017 very deep neural nets (beats people!)	• 3%
• University of Toronto (Krizhevsky <i>et al</i> , 2012)	• 16%
• University of Tokyo	• 26%
• Oxford University (Zisserman <i>et al</i>)	• 27%
• INRIA (French national research institute in CS) + XRCE (Xerox Research Center Europe)	• 27%
• University of Amsterdam	• 29%

図-26

〈図-26〉コンペティションの結果、従来の物体認識装置の中で最も優秀な成績を残したものは東京大学が開発したものでした。それはオックスフォード大学が開発したものよりも若干良い結果を残しました。同様に、フランスの国立研究機関とゼロックス・リサーチセンター・ヨーロッパが共同で開発したものよりも若干良い結果を残しました。ご覧の通り、従来手法のエラー率は約 25%に近づいていることがわかりますが、私が協力したアレックスとイリヤがマークしたエラー率はたったの 16%でした。これは劇的な改善でした。このことを受けて、コンピュータビジョン学会はこれらの手法が実際に機能するものであることを認めざるを得ませんでした。この2年前に、「学習している」という理由でヤンの論文を却下したコンピュータビジョン学会。彼らは科学学会らしい振る舞いをするようになりました。「これは我々の研究よりも優れている。これからはこちらを使うようにしたい」と彼らは言うのです。学会のすべてがそのようになったわけではありませんし、

2 年もかかりました。ですが、たった 2 年の間に学会は完全に変わり、今では物体認識の研究者であれば誰もがニューラルネットワーク、特に畳み込みニューラルネットワークを採用するようになったのです。

2012 年にエラー率を 26%から 16%に大幅に改善させたことは画期的なブレイクスルーでした。それ以来、人々はニューラルネットワークをさらに発展させました。学会の多大なる努力の甲斐もあり、現在は 3%にまでエラー率が低下しています。このデータセットにおける人間のエラー率は約 5%です。つまり、このデータセットにおいては人間を凌駕しているのです。総合的に見れば人間には敵いません。ただし、特定のデータセットにおいては人間にも勝るのです。今では、皮膚ガンの判別といった特定の問題においては、最も優秀な人間にも匹敵するようになりました。そして近い将来、人間を超えるでしょう。CAT スキャンの解析においても、もうすぐ人間を超えるでしょう。極めて専門的なデータが膨大にあれば、畳み込みニューラルネットワークは大きな改善をもたらすでしょう。

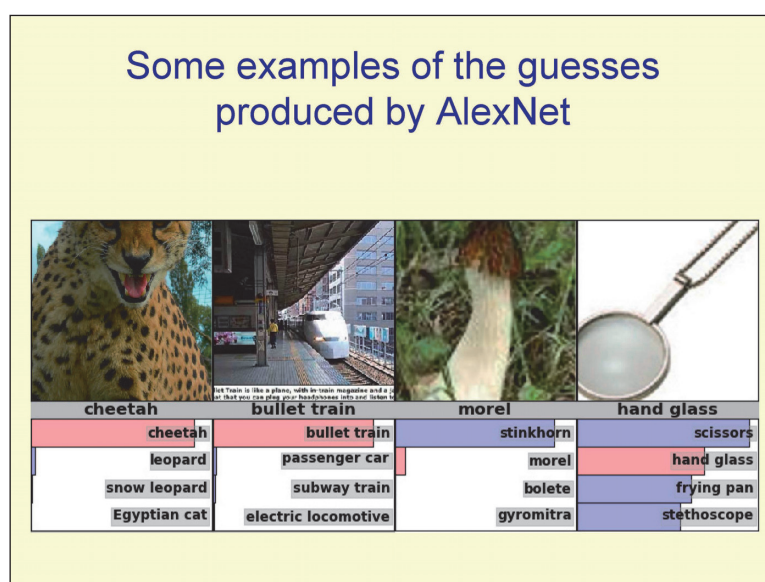


図-27

〈図-27〉 ImageNet コンペティションの例をいくつかご紹介します。画像に関して興味深いことは、画像はウェブから集められたものであり、普通の見え方の画像ばかりではなかったということです。これは、アップ気味に撮影されたチーターの写真です。ちょっと近すぎるかもしれませんね。ピンクはニューラルネットワークによって得られた回答を示しています。ImageNet コンペティションでは、5 つの回答を挙げることができます。5 つの回答のうちの 1 つでも正解していれば、正解としてカウントされるのです。1 番目の回答が正解なのですが、他の回答も正解に近いことがわかりますね。

ここでも、画像が新幹線であると回答していますが、新幹線は画像のほんの一部でしかありません。新幹線よりはるかに大きな建物があり、ここには人も写っています。学習にあたっての要点として、中央にあるものにラベル付けをする必要があります、新幹線に着目する必要がある、といったことを学習させる必要がありました。電気機関車や客車といった

他の回答も比較的正解に近いですね。

ここではキノコの種類の判別をととても正確に行なっています。私よりもずっと正確ですね。そして正しい回答も得られています。あ、違いますね、2 番目であると回答していますね。画像はアミガサタケですが、スッポンタケであると回答しています。

これはカタログに掲載されている画像の一例です。最初にハサミであると回答していますが、これは不正解です。なぜハサミであると判断したかという、下の丸い部分がハサミの柄に見え、上の部分が刃に見えたからです。フライパンであると判断した理由も同様に説明できます。つまり、視覚はそれほど良いわけではありませんが、エラーが示している通り、視覚の概念をある程度理解しているということです。よって、正解にかなり近いエラーを出力するのです。

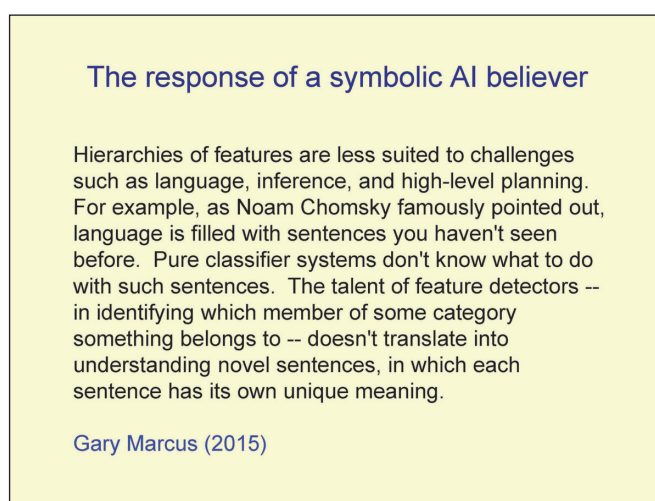


図-28

〈図-28〉私たちはバックプロパゲーションでトレーニングされたニューラルネットワークや、ビッグデータセットでトレーニングされた大きなニューラルネットワークが、非常にうまく音声認識や物体認識を学習することができることを発表しました。すると、シンボリック AI 研究者たちの反応が見られました。彼らの中には考え方を改めた人もいます。しかし、ゲイリー・マーカスのようなシンボリック AI を心から信じている人もいます。彼は多くの著書を発行しているのですが、2015 年に発行した著書の中で、「この種の学習アルゴリズムはクラス認識のために特徴を捉えるといったことには役に立つかもしれないが、未知なる文章を処理することはできないため、言語処理においては役に立たないだろう」と主張しました。

その昔、学習するという概念をまったく信じていなかったチョムスキーは、「言語は未知なる文章に溢れているため、言語処理に統計的手法を用いることはできない」と主張しました。この主張は完全に間違っているのですが、この考えはシンボリック AI の学会に広く浸透しており、ゲイリー・マーカスも未だに信じているようです。つまり、未知なる文章を処理しなければならないため、特徴検出器はまったく役に立たないというのです。

A radically new way to do machine translation

(Suskever, Vinyals and Le, 2014)

- For each language we have an encoder neural network and a decoder neural network.
- The encoder reads in the sequence of words in the source sentence.
 - Its final hidden state represents the thought that the sentence expresses.
- The decoder expresses the thought in the target language.

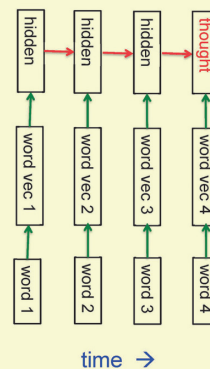
図－29

〈図－29〉 その本が出版される 1 年前、Google とモンリオールにあるヨシュア・ベンジオのラボの人たちは、未知なる文章の処理を可能にする全く新しい手法を用いた機械翻訳を発表しました。そして 2015 年頃には、他の手法を用いた機械翻訳に匹敵するようになりました。今では、ニューラルネットを用いた手法のほうが他の手法よりもはるかに優れており、Google 翻訳にも採用されています。Google 翻訳が 5 年前と比べてずっと良くなったのはこのおかげなのです。

今でははるかに複雑になっているのですが、当初のアイデアは各言語に対して単語を 1 つ 1 つ読み取る回帰型ニューラルネットワークを使い、これらの単語を内部ベクトルに変換するというものでした。一つの文章を読み終えたら、ネットワーク内で一連の言葉を大きなベクトルに変換します。「思考ベクトル」と私が呼ぶこのベクトルは、その文章で表現された思考を表すニューロン活動のベクトルです。

An encoder recurrent neural network (RNN)

- The English encoder RNN converts a sequence of English words to a “thought vector”



図－30

〈図－30〉 エンコーダーの仕組みはこうです。単語を入力すると、その単語はシンボルになります。そのシンボルをベクトルに変換します。それは大きなニューロン活動の集合体です。そのベクトルは他のニューロンに入力を送ります。そして、文の 1 番目の単語がこ

れであることを表すニューロン活動のベクトルが得られます。次に、2 番目の単語をベクトルに変換します。ここでの入力、2 番目の単語から得られた情報に文中ですでに得られた情報を加えたものですので、1 番目と 2 番目の単語がわかっている文を表すベクトルが得られます。文の最後までこの作業を繰り返します。文の最後に至る頃には、文に込められた思考を表すベクトルが得られます。それが特定の言語の思考を表すのです。これを行うには、すべての結合強度が正しいことが前提となりますが、一度正しい結合強度を学習してしまえば、あとは簡単に実行することができます。別の言語を処理したい場合、別のネットワークをトレーニングし、その言語を処理できるようにします。したがって、英語のネットワークは英語における単語の順序や、形容詞が名詞の前にどのように置かれるかということをたくさん学習します。フランス語のネットワークは別のことを学習するでしょう。これが、ある言語の文章を思考に変換する方法です。

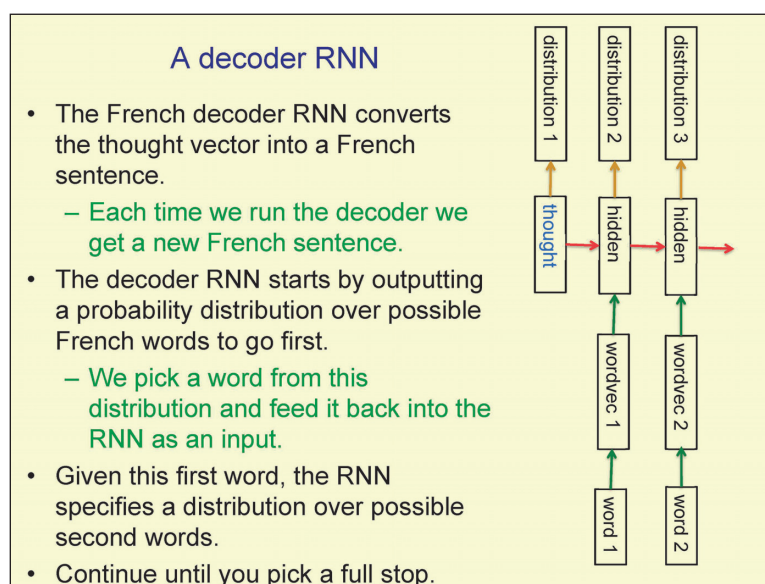


図-31

〈図-31〉 そうすれば、その思考を出力言語の文章に変換することができるのです。フランス語のデコーダーがあるとしましょう。フランス語のデコーダーが何をするかという、与えられた思考をもとに、文章の最初の単語が何であるかを考えるのです。そして、最初の単語の確率を計算するのです。最初の単語は 40%の確率で「le」、30%の確率で「la」、10%の確率で「chat」である、といった具合に考えるのです。つまり、この確率論的なネットワークは単語の確率が何パーセントであることを示すのです。

翻訳文を生成する場合、これらの確率を参考に単語を一つ選びます。「le」である確率が高ければ、おそらく「le」を選びます。選んだ「le」をここに入力としてフィードバックすると、「le」がベクトルに変換されます。そのベクトルに思考を加え、「この思考と『le』という単語がある時、他に何を表現する必要があるだろうか？」ということを考えます。それはこの部分で表されます。そして、残りの文を選択し、何を表現する必要があるかを考え、次にどの単語を生成するかを決定するのです。さまざまな単語に対して確率

を計算し、その確率をもとに単語を選択し、入力にフィードバックする。この時点で文の1番目と2番目の単語がわかります。「le chat」となったかもしれません。これを踏まえて、ここに表現を生成します。この表現は、すでに「le chat」が使用されていることと、生成しようとしていた思考をわかっており、次にどの単語を選ぶべきかを考えます。そしてこの作業を繰り返すのです。

これら2つのネットワークをトレーニングするには、ネットワーク同士を繋げます。最初はランダムに重みがかけられているので、意味が通らないフランス語を生成します。英語の翻訳とは呼べない一連の文章を生成します。入力した英文に対する正しいフランス語の翻訳をあなたは知っています。この時点で「chat」という単語を生成する確率が非常に低かったとしましょう。しかし、実際の翻訳に必要な正しい単語は「chat」なので、「この結合と、この結合と、この結合のすべてを変更し、『chat』という単語が生成される確率を上げよう」となるのです。これがネットワークをバックプロパゲーションすることであり、重みの変更方法を導き出す方法なのです。「chat」の確率を上げましたが、「chat」の確率を上げるためには、思考ベクトルをどのように変更するかも導き出す必要があります。

思考ベクトルの変更方法がわかったら、エンコーダーネットワークに戻ります。ネットワークは「この思考をどのように変更したいかがわかったので、この部分と、この部分と、この部分の重みの変更方法を求めることができ、思考を変更することができる」と考え、フランス語で正しい翻訳を生成するようになるのです。数百万の英文と仏文の組み合わせに対してこの作業を続けます。十分な時間をかけてこの作業を行えば、最初はランダムな重みがかけられていたネットワークが翻訳を行えるようになるのです。

現在、システム自身が生み出す翻訳は質が高いとは言えませんが、それほど悪いわけでもありません。ただ翻訳は行います。それ以来、翻訳の精度を上げるために多くの研究が行われてきました。ですがそれらの研究について触れるつもりはありません。

How the encoder and decoder RNNs are trained

- Given a sentence pair, use back-propagation through time to maximize the probability of producing the specified translation.
 - We backpropagate through the decoder to train the encoder.
 - All we require is lots of pairs of an English sentence and its French translation.
 - There is no innate linguistic knowledge.

図－32

〈図－32〉ある言語から別の言語へ翻訳するためにバックプロパゲーションを使うことは、

比較的シンプルなネットワークのトレーニング方法であるということ、また、ネットワークは言語の知識を一切必要としないということが、このデモンストレーションの主なポイントです。ある言語の単語を与え、別の言語の単語を生成するように命令する。始めにランダムな重みをかける。ネットワークにプログラミングしなければならない言語の知識はゼロです。

Combining vision and language
(a simplified account of work by Vinyals *et. al.*)

- The activity vector in the last hidden layer of a deep neural net trained on ImageNet is a “percept”.
 - The percept encodes what is in the image in terms of objects rather than pixels.
- Map this percept to the initial hidden state of an English decoder RNN.
- Train the decoder RNN to say what it sees in the image.
 - Use an additional set of 200,000 images that each come with several captions (MS-COCO).

図－33

〈図－33〉次に、最初にお見せした画像と画像キャプションの例に戻り、私たちが現在どのようにして問題を解決しているのかを説明しましょう。問題の解決には、機械翻訳のシステムと ImageNet の認識システムを結合させます。

まず、画像を ImageNet の認識システムに入力し、さまざまな物体のカテゴリに関する確率を出力します。出力の直前にある最後の層には、画像のピクセルをもとに画像の中にあると思われる物体の種類に関する情報を生成した、ニューロン活動の大きなベクトルがあります。その大きなベクトルを知覚として扱います。次に、その知覚と言語デコーダーを使います。英語をエンコードした思考を入力する代わりに、その知覚を入力します。そして、この知覚を言語に置き換えるよう命令するのです。そのためにはトレーニングセットが必要ですが、画像とキャプションが含まれている素晴らしいトレーニングセットを Microsoft が提供してくれました。画像認識を行えるシステムはすでにあります。そのシステムのニューロンの最終層を言語デコーダーへ入力し、それが何であることを言わせるのです。重みを学習するので、画像の中のものを答えられるようになります。これがうまく機能するのです。

Neural net machine translation

- It has evolved a lot since 2014.
- Use soft attention to words in the source sentence when producing the target sentence.
- Pre-train the word embeddings by trying to fill in the blanks using transformer networks.
- Neural nets are now used by Google to do machine translation.
 - They are much better than the previous method.
 - They learn everything from data using random initial weights.

図－34

〈図－34〉ある言語における一連の単語を思考に変え、その思考を他の言語で表現するというシンプルなアイデアを初めて導入した2014年以来、ニューラルネットを用いた機械翻訳は多くの進化を遂げてきました。そして、その大きな要因の一つが「アテンション」です。通訳者であればわかることですが、通訳の際には元言語の文章をすべて聞き取るわけではありません。そして元言語に立ち返って再考することもなく、訳した後の文章を述べます。ですが今行われていることは、通訳にあたり、元言語の単語、むしろ単語の隠れ表現に立ち返って、それらのどの部分に着目して訳すべきかを決定し（ソフトアテンション）、その決定に基づいて内容を発する前に修正を加えているようなものなのです。そうすることでより良い翻訳に繋がります。

翻訳改善に繋がるもう一つの方法は、事前学習をさせることです。翻訳のためにすべてをトレーニングするわけではありません。ある言語における一連の文章を用意し、いくつかの単語を削除します。そして、ネットワークに完全な文になるよう再構築させるのです。単語が削除された文を与えているにもかかわらずです。隣り合う単語のセットを少し削除した方がよりうまくいきます。文章中に小さな空白を作るのです。ネットワークが学習しなければならないことは空白を埋めることです。翻訳はせずに、単に英語で空白を埋めるのです。その学習が完了したら、機械翻訳の学習をさせます。空白の埋め方を学習することで、単語の表現に必要な非常に良いベクトルを学習するのです。

例えば、「may」という単語が含まれる英文を与えます。すると、「may」をベクトルに変えると、5月の「May」なのか、「he may do that」といった法助動詞の「may」なのかわかりません。すなわち、当初の「may」を表すために使ったベクトルは、5月を表すベクトルと法助動詞を表すベクトルの2つの折衷でした。幸い、大型の高次元ベクトルは他のものに近すぎることなく、5月にも法助動詞にもかなり近かったので、機能しました。しかし現在は、複数の単語を与えると、各単語に対するベクトルが得られ、得られたベクトルは近い単語のベクトルを確認します。同じ文の中で近い単語が「June」であったとしましょう。「May」という単語があり、近い単語が「June」であれば、「May」の意味を明確にすることができ、5月を表していると判断するのです。近い単語が「might」や「could」

であれば、法助動詞の「may」であると判断するでしょう。このような単語の曖昧さを解消する方法によって、これらの言語モデルがはるかに良く機能するようになりました。

今ではとても良く機能します。実のところ、とても良く機能するので、今ではニューラルネットに大規模な言語セットのトレーニングをさせることで、機械翻訳ではなく、単に次の単語を予測させるといったことに利用できるようになったのです。トレーニングが完了したら、一つの文章をネットワークに与え、次の単語を予測させます。ネットワークが予測した単語が何であったとしてもそれが正解であると教え、入力にフィードバックします。ネットワークはまた次の単語を予測します。それが正解であると教えます。そうすると、大きなネットワークは驚くような物語を生み出すのです。筋の通った長編の物語を生み出すのです。まったく同じ手法を使って作曲を行うこともできます。ニューラルネットワークが人間の作ったような音楽を作曲するのです。ニューラルネットワークによる作曲なのか、人による作曲なのかを聞き分けることはかなり難しいです。

The lesson of neural net machine translation

- This is the final nail in the coffin of pure symbolic AI.
 - Machine translation is an ideal task for symbolic AI because the input is symbols and the output is symbols.
- But to make it work we need **vectors inside**.
- The insights of symbolic AI researchers need to be used to design better architectures for neural nets.
 - Let stochastic gradient descent learn how to make it all work.

図－35

〈図－35〉ニューラルネットを用いた機械翻訳の教訓ですが、機械翻訳は本来であればシンボリック AI にうってつけの問題だということです。なぜなら、入力はある言語の文章という一連のシンボルであり、出力も別の言語の文章という一連のシンボルだからです。そして機械翻訳を行うための簡単な方法というのも、出力シンボルを生成するために入力シンボルを操作することです。だからこそ、一連のシンボルを操作するルールが必要であると考えたのかもしれませんが、かつての翻訳はそれで機能していました。ですが実際のところ、それはベストな手法ではなかったのです。入力シンボルを取り込み、ニューロン活動の膨大なベクトルに変換する。これらのベクトル間の相互作用を利用して出力シンボルを生成するのです。

入力も出力もシンボルです。しかし、内部にあるものはすべてベクトルであり、これらのベクトルは他のベクトルに対して因果効果を持つのです。

今では、シンボリック AI は時間の無駄ではなかったと思います。シンボリック AI の研究者たちは、人々がどのように論理的思考を行うのかについて多くの見識を持っていました。

彼らはその見識を従来のコンピュータに結び付けようとしたのですが、それは正しい方法ではありませんでした。私たちが今やらなければならないことは、より良いニューラルネットワーク設計のために、シンボリック AI の研究者たちが持つ見識を活用することです。ただし、手作業ですべてをプログラミングしようとする必要はありません。すべての重みを求めるといった大変な作業は、ニューラルネットワークの学習アルゴリズムにさせればよいのです。

もし、音声認識や物体認識、翻訳を行うために知的システムを構築したい場合、始めにランダムな重みをかけた膨大なニューラルネットワークを使うべきであることは、今のところ疑いの余地がないと思います。どのように問題を解決するかによって、構築されるネットワークの種類は異なるでしょう。しかし、基本的にはネットワークがデータからすべてを学ぶのです。

私からの話は以上です。

■ このレポートは本田財団のホームページに掲載されております。
講演録を使用される場合は、事前に当財団の許可を得てください。

発 行 所 公益財団法人 **本田財団**
104-0028 東京都中央区八重洲2-6-20ホンダ八重洲ビル
Tel.03-3274-5125 Fax.03-3274-5103
<https://www.hondafoundation.jp>
発 行 者 亀 岡 晃 浩