SEMINAR REPORT

ディープラーニング（深層学習）とハードウェア技術 ～富士通におけるディープラーニングへの取り組み～

富士通株式会社
ＡＩサービス事業本部
第一フロンティア事業部
事業部長　吉山　正治　氏

富士通の吉山と申します。今日は、「ディープラーニングとハードウェア技術～富士通におけるディープラーニングへの取り組み～」ということで、お話しさせていただきたいと思います。私は、富士通の AI サービスを提供する部門に所属しておりまして、実際にお客さまとお話させていただいて、業務へのディープラーニングの適用を行っております。

このディープラーニングについて説明しようとすると、どうしても微分とか偏微分とか行列とか、数式を出さないといけません。なるべく簡単に説明することに努めたのですが、一部数式が出てくるところはご容赦願いたく存じます。

さらにもう１つ。私はこの事業部に来て、まだ１年ほどです。ある意味、初心者に近いところもあるので、用語とか理解とかに皆さんと違うところがあるかもしれませんが、そこのところはちょっと目をつぶっていただければと思っております。

はじめに

AI への期待ということで、16 か国の経営層に「AI をどのように認識をしていますか?」というアンケートを取りました。図１をご覧いただきますと、「AI はいつか仕事を代替する」とか、「ビジネスプロセスに最大限活用する」とかのご意見をいただいています。スコアの低いところでは、「人が行う全ての仕事を代替する」というのがあります。ニュースとかメディアとかによると、AI が発展すると人のする仕事が全て無くなってしまうのではないか、全て AI に奪われてしまうのではないかと言われていますが、さすがビジネスリーダの方は冷静に見ていまして、全て代替することはないだろうと思っておられます。ここが１つ重要なポイントだと思っています。そうはいっても、他の項目は全てスコアが高いので、やはり AI の可能性に大きな期待を寄せているというのが現状ではないかと思っています。

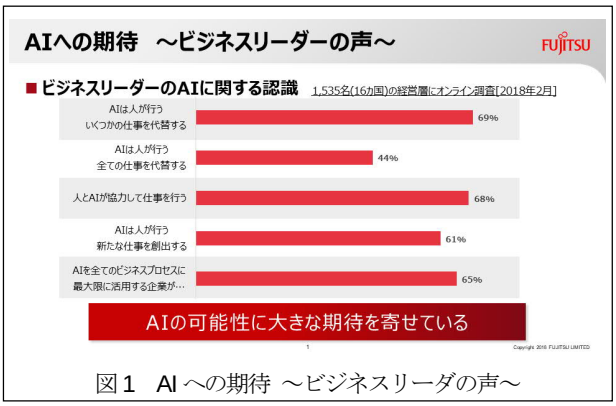


図１　AI への期待 ～ビジネスリーダの声～

人工知能（ＡＩ）とは

まず、「人工知能、AI とは何でしょうか」というところからお話しします。現在のテレビ CM を見ていても、事あるごとに AI という言葉が出てきていますが、実際問題として、AI とは何かについてきちんと押さえておく必要があると思っています。もっとも、AI という言葉には曖昧なところがあって、人工知能学会で定義されているのですが、正直なところ知性とか知能とかに係わる問題であって、そもそも知性・知能には定義がないのです。なので、AI は人によって色々な定義があるので、そこのところを踏まえた上で見ていただければよいと思います。

図２に示すように、人工知能の範疇には機械学習というものがあります。これは何かといいますと、機械で学習するというよりも、人が色々と情報を集めてきて、A が来たら B を返す、C が来たら D を返すというようなルールを決めて、それをコンピュータに覚え込ませる手法となります。それをさらに狭義で捉えるなら、ニューラルネットワークというものがあって、神経細胞をモデル化したものです。このモデルを使って学習させるもので、その中の１つがディープラーニングです。神経細胞がスマートフォンを見て「スマートフォンだ」と分かるのは、脳の中で何かの反応が起きていて、それでスマートフォンと判断しているのであって、それと同じようなモデルをコンピュー

タにやらせて、色々なもの、例えば、画像認識とか、音声認識とかをやらせようというのがディープラーニングです。

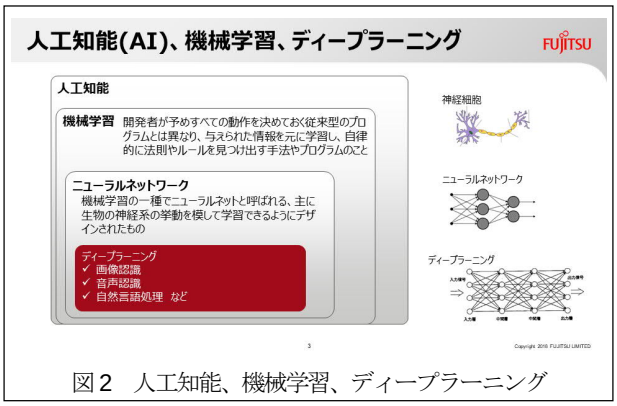


図２　人工知能、機械学習、ディープラーニング

AI は万能のように思われていますが、実は、ディープラーニングの世界は万能ではありません。普通のコンピュータ処理では、A の処理をさせて B の結果となって、必ず正しい答えが出せるのですが、ディープラーニングでは、常に正しい答えを出せるかというと、そうではなくて、8, 9 割方が正しいという世界です。

コンピュータと AI では何が違うのかというと、コンピュータは人がすると難しいようなことを、とても速く処理してくれます。例えば、10 万行のデータの中からあるデータを探し出すとき、人はとても時間がかかるのに、コンピュータはすぐに見つけてくれる。逆に、AI 特にディープラーニングは、人のできることがようやくやれる程度です。そこのところが違っていると思います。先ほどのスマートフォンの話ではないですが、これはスマートフォンだと人の目にはすぐにわかりますが、これを AI に判断させようとすると、長い時間をかけて勉強させて、ようやく判断できるのです。そのような例を、次から具体的にお見せしたいと思います。

写真を見て犬と分かるのは何故か。古典的な機械学習では、犬の特徴を人が教え込みます。犬はどのような特徴があるかというと、簡単な例では、図３に示すように、①4 本足で、②鼻がシュッとしていて、③毛で覆われている。この３つぐらいが犬の特徴ではないかと思います。他に何か特徴があるのかとなって、特徴を次々と覚え込ませていかないといけないのが、従来の機械学習のロジックです。

図３　写真を見て犬と分かるには？

それで、犬は分かったけれども狐はどうなのか。犬とどこが違うのか。確かに狐も 4 本足で、鼻がシュッとしていて、毛で覆われています。なので、特徴を全て言葉にして覚え込ませるとなると非常に難しいです。機械学習では、昔はこれを一生懸命行っていました。プログラム全体が「If～then else～」の形で書き連ねられていて、最終的には分からないという経験をされた方もおられるかと思いますが、そういうことが起きます。機械学習で教え込ませることは非常に大変で、どうしても漏れが生じてしまい、認識率が低くなってしまうというのが現状です。

図３では、4 本足の話をしましたが、図４では、これは何本足ですかという話です。①飛んでいるとわからないし、2 本足で立たれると難しい。②鼻がシュッと言われても、そうでない犬もいる。③全身が毛で覆われているかとなると、最近は色々な犬がいて、覆われていないのもいる。人が全ての犬種、動き、状態を加味して特徴で教えることは、非常に難しいのです。なので、ディープラーニング手法が出現する前は、人が一生懸命覚えさせていたのが、ディープラーニングが出てきて、状況ががらりと変わってきたというのが現在の状況です。

図４　人が特徴量を教え込むことの限界

先ほどの犬の話では 4 本足、鼻がシュッというのが特徴であったと思いますが、人が教えなくても機械が画像から特徴量を自分で見つけ出すということが、ディープラーニングの特徴となります。なので、まずは入力として犬の画像をたくさん用意して、一生懸命覚え込ませます。そうすると、画像の中で特徴量を見つけて、これが犬だと分かる学習済のモデルを作ります。そうすると、次にこのモデルを使って、例えば、猫と入力して認識済モデルにかけると、これは犬ではないということで、犬の確率を 0.0001 とはじき出してきます（図 5）。これが、ディープラーニングの特徴です。なので、機械学習では人が特徴を覚え込ませていたものを、ディープラーニングでは画像を大量に用意すれば、何か知らない内に犬と分かるモデルを作り上げることができるのです。

図５　ディープラーニングの特長 － 特徴量を見つける

図 6 は、従来の機械学習とディープラーニングの違いを比較したものです。従来の機械学習では、人がデータを分析して特徴量を抽出して、学習方法を設計しています。なので、どのように学習しているかは、作っている人には分かっているのです。ディープラーニングでは何が違うのかというと、機械が自動抽出・設計して特徴量を見つけています。なので、抽出から学習のところはブラックボックスになっています。ここで何が起きているのか分からないが、結果は分かるということになります。

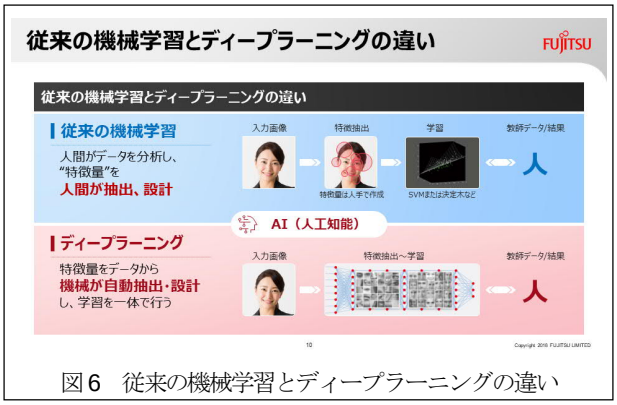


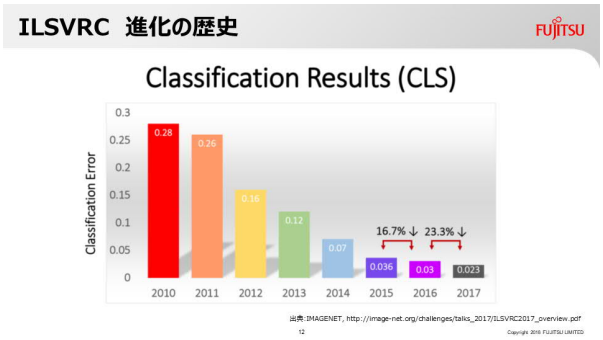
図 6　従来の機械学習とディープラーニングの違い

機械学習とディープラーニング、少し前までは機械学習の方が成績がよかったのですが、あるときブレークスルーが起きて、ディープラーニングが脚光を浴びるようになりました。ILSVRC（ImageNet large scale visual recognition challenge）という画像認識のコンテストがあります。2012 年に、トロント大の SuperVision というディープラーニングを使ったプログラムが出展されました。エラー率は 16%です。これは相当間違っているような気がしますが、2 位は 20%を超えています。20%以上間違っているというのは、人の能力と比較するとほど遠い世界です。人の間違い率は 5%ぐらいです。それでも、圧勝したのがディープラーニングということで、脚光を浴びたのです。確かにぶっちぎりでしたが、そうは言っても、16%というのが 2012 年のときの実力でした。

では、今はどうなっているかというと、図 7 の棒グラフ黄色がエラー率 16%ですが、1 回脚光を浴びると誰もがそれに群がってやり始めて、今はどこまで向上しているかというと、2017 年で 2.3%です。なので、ディープラーニングはほぼ間違えない、人ほど間違えないレベルまで到達しています。劇的に進化しているのです。2012 年より前までは、28%ぐらいの性能でけっこう機械学習が使われていたのですが、現在は、ディープラーニングに置き換わってきています。

図 8 は、AI と富士通の歴史を表わしたものです。富士通は昔から AI に取り組んでいます。日本初の AI 搭載コンピュータとか、知識情報システムとかを世に送り出しています。ディープラーニングには計算機パワーが必要ということで、昔からスパコン「京」に取り組んでいたので、AI コンピュータにも取り組んでいるのです。第1次AIブームがあって、冬の時代があって、第 2 次 AI ブームがあって、また冬の時代があって、今は第 3 次 AI ブームです。今のブームを牽引しているのは、やはりディープラーニングということになろうかと思います。

なにゆえ、このように脚光を浴びるようになったかというと、やはり、ビッグデータと計算機パワーとアルゴリズムの 3 点セットが揃って、ようやくできるようになったのではないかと思っています。1 つ目は、データが圧倒的に多数揃えられるようになったことです。第 2 次 AI ブームのときに失敗したのは、ビッグデータを自動的に集められなかったのが大きかったようです。極論すると、当時は町中歩いて、猫の写真を撮りまくるというような状況でした。今は Google で検索すれば、猫の画像がいっぱい出てきます。2 つ目は、計算機パワーが圧倒的に上がったことです。後ほどお話しますが、GPGPU（General purpose computing on graphics processing units）が出現して、計算能力が圧倒的によくなったことです。3 つ目が、アルゴリズムのニューラルネットワークの出現です。色々なことに対応できるようになったことが非常に大きいと思っています。



出典：IMAGENET, http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf

図 7　ILSVRC 進化の歴史

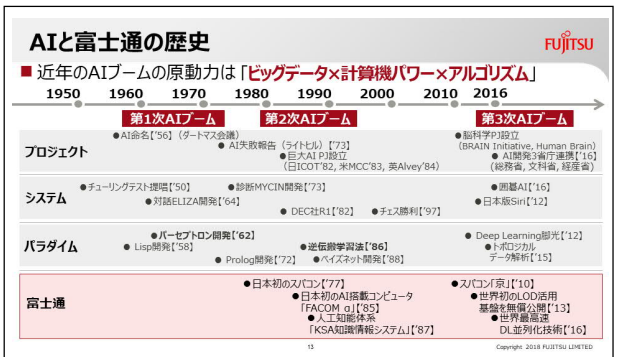


図 8　AI と富士通の歴史

アルゴリズム

それでは、3 点セットの内、ビッグデータはスキップして、アルゴリズムと計算機パワーについてお話させていただきたいと思います。

ニューラルネットワークは、人の脳の構造を模倣したネットワークをコンピュータ上で構築しようというのがスタートポイントです。シナプスが外部からの刺激を受けて違う刺激を出して次々と伝えていくことで、スマートフォンを認識したり、言葉を話したりできるというのが人の脳の構造です。

図 9 中央に示すように、これを模して考えられたのがパーセプトロンです。入力 x1, x2, x3 を何かしら入れて、それに重み付けをして計算した結果を出力します。この出力したものを、また次の入力にする。要するに、このような脳の単純なモデルの組み合わせと考えたのがパーセプトロンです。このパーセプトロンの出力は、0 と 1、真か偽かしかありません。これでは使いにくいということで、もう少し使いやすいものを考えよう

ということになって、脳の構造に近いもの、多層で組み合わせることにします。先ほどの図5の犬と猫ではないですが、犬は80%ぐらいで100%ではないということで、0/1がはっきりしないものは、ある関数を使って0から1までの確率分布で出力するようにします。このように組み合わせたものがニューラルネットワークになります。なので、それぞれの入力に対して重み付けした計算結果を踏まえて、次の出力に出すか出さないかを組み合わせてやるのが、このニューラルネットワークです。これを組み合わせれば、脳みそと同じような構造ができて、先ほどのような画像認識ができるのではないかというのが、ニューラルネットワークのコンセプトです。

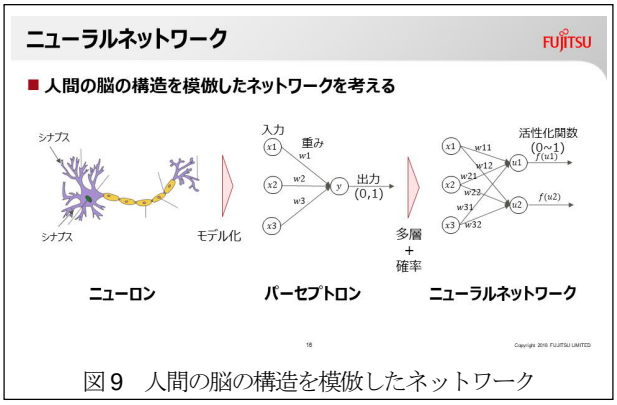


図9　人間の脳の構造を模倣したネットワーク

それで、計算は実際にはどのようにするかというと、例えばu1の出力は、それぞれの入力に対して重み付けして足していきます。f(u1)はw11×x1＋w12×x2＋w13×x3と計算します。同様にu2の出力f(u2)も同じような計算式となります。これを行列計算で表すと、図10のような行列計算式に置き換えられます。

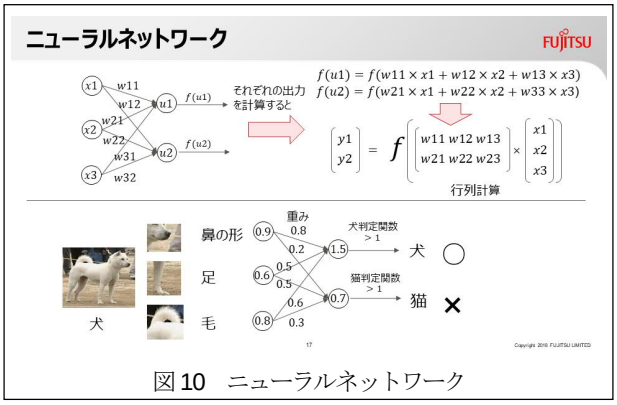


図10　ニューラルネットワーク

先ほど、図3で示した特徴、鼻の形、足、毛の組み合わせで犬と猫を判断します。犬は重み付けした計算結果が1.5になりました。1以上は犬であるというフラグが立ちます。猫判定関数は0.7となって、1以下なので猫ではない。ここでは単純な例で示したのですが、仕組み的にはこのような重み付け計算して、犬か猫かを判断していくことになります。

図11に示すように、最初は適当に重み付けしているので、実際に計算してみると、犬と猫を正しく判定してくれません。犬判定関数が1以上にならないので、犬判定にならない。判定が間違っているので、重み付けを見直さないといけない。重みの0.6, 0.2, 0.4を直していく、すなわち学習することになります。こうして重み付けの見直しを繰り返すことで、最終的に犬か猫かが正しく判定できるようになるというのが、ディープラーニングの重要なポイントとなります。

この重み付けの見直しを人がするのはとても無理で、プログラミングで自動的に行うのがディープラーニングです。ここの例では、入力が3つで出力が2つなので、重みは全部で6つしかないですが、層が深くなると、重みを次々と変えていかなければいけない。これがうまく調整できると、犬は犬、猫は猫と判断できるようになります。

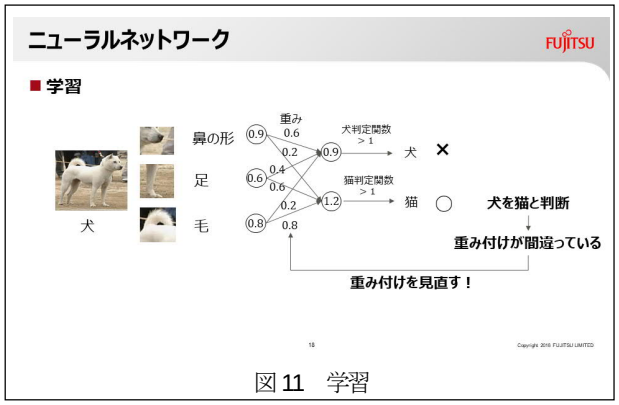


図11　学習

そこで、実際に図12のようなモデルを作ってみました。入力は28×28のデータなので、全部で784。これをビット列に直して、784入力のモデルを作りました。中間層に計算した結果を1回保存しておいて、もう1回計算して出力するモデルになっています。0から9までの数字を判定するもので、実際に計算してみました。デフォルト状態で計算して、80%の認識率で数字を認識することができました。家にあるパソコンで計算したもので、80%ならまずまずといったところです。計算時間は1分もかかっていないです。

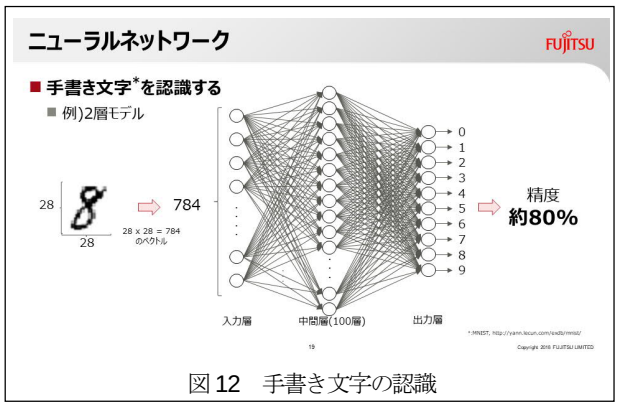


図12　手書き文字の認識

この80%の認識率はどうかというと、これでは容認できなくて、90%台までもって行かないと業務には使えないです。そこで次のアクションとしては、80%をいかにして上げるかがポイントとなります。ディープラーニングでいうところのチューニング作業になります。

色々とやり方はありますが、図13に例として書かせていただいたのは、中間層を変えてみることです。入力層784、中間層100を1回、出力層10にしているので、中間層が多い方が認識率が上がるのではないかと考えて、200にしてみたら81%になりました。それでは、減らすとどうなるかということで、50にしてみたら75%になりました。次に、活性化関数変えて

みました。デフォルトはシグモイド関数ですが、ランプ関数に変えてみました。そうすると、認識率が 88%になりました。その次は、学習回数を変えてみようということで、2 万回にしてみたら 83%になりました。

チューニング作業でこれら 3 つを組み合わせてみたら、91%まで認識率が向上しました。デフォルトのままでは 80%なのが、チューニングを施すことで 91%まで上がりました。このような単純なモデルでも、チューニングすれば 91%まで上げられるということです。

このチューニングも、AI でできたらよいのですが、ここのところは人がしているのです。中間層の数を増やしてみるとか、中間層の回数をもう 1 つ増やしてみるとか、人が行って認識率を向上させているのが実情です。

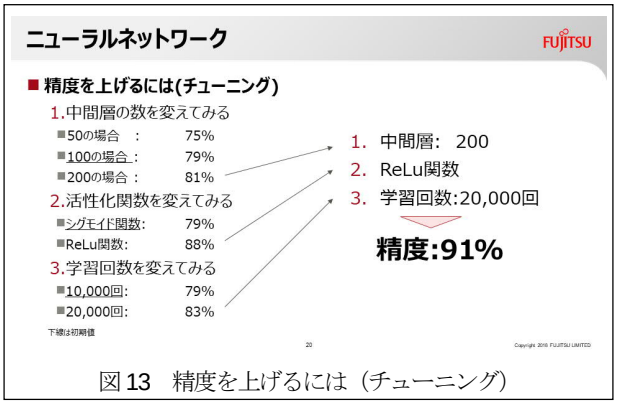


図 13　精度を上げるには（チューニング）

当事業部では、年間 40 件ぐらいの案件を扱っているのですが、先ほどのようなチューニングを繰り返し行っています。実ビジネスでは、顧客からデータをもらって、最初は 50～60%ぐらいだったものを 80～90%まで上げていく作業をしている状況です。これを 2～3 か月で終えなければいけないので、現状の AI は結構泥臭い作業だと思います。

図 14 は川崎地質様の実際の例です。今までは、撮ってきたレーザー画像を人が認識して、「ここは空洞だ」と見つけていましたが、これに AI を適用しました。認識率も非常に高く、探索時間を減らすことができたということです。

これも泥臭い作業で、1 回の学習に 10 時間以上かかっています。精度を上げていくとなると、数か月はかかります。モデルを作って夜中に動作させておくと、帰った後にエラーになっていることもあって、チューニングに結構時間がかかるのです。

図 14　道路陥没を防ぐ路面下空洞調査

実際のディープラーニングは、するべきことが非常に多くて、利用までの道のりはとても長いです。図 15 は、当社のプロセスモデルを示したものですが、まず、顧客の要求が結構曖昧です。とにかく AI を導入したいという顧客は多くいるのですが、課題設定、要件定義のところを誤ると後が大変です。例えば、カメラ画像でトラックを識別したいという話であれば、データの確認、データの準備、データの分析のところは非常に重要です。このトラックの写った画像にトラックというタグを付けていますが、これを誤ると違うものを認識してしまいます。データの数をいかに綺麗に揃えるかも非常に重要で、ここにも大そう時間がかかっています。

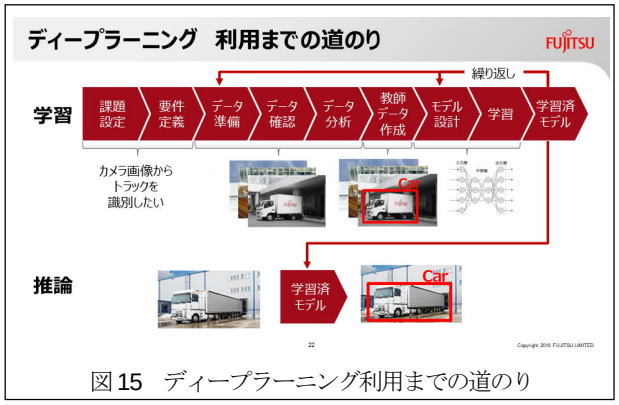


図 15　ディープラーニング利用までの道のり

これらのことを踏まえてモデル設計して、学習させる。結果がよくないなら、モデルを見直したりデータを見直したりして、ようやく学習済のモデルができ上ります。それで推論をして、実際にトラックの写った画像を見せて、これはトラックですと認識させる。このような手順を踏みます。ですので、通常のプログラミングでは、ロジックが決まったらそのとおりに組めばよく、後はバグがあるかないかの問題ですが、ディープラーニングはバグがあるかどうかも分からないブラックボックスなので、やってみないとうまくいくかどうかも分からない。やってみても、その正解率でよいのかどうかも分からないということで、判断が非常に難しいのです。

ニューラルネットワークには、様々なモデルがあります。RNN（Recurrent Neural Network）とか、LSTN（Long Short-Term Memory）とか、CNN（Convolution Neural Network）など色々なモデルがあって、これらのモデルの中から顧客の要望に沿うモデルを見つけてきて、組み合わせて構築する。これをベースにさらに改善をしていくのが、現状のビジネスではないかと思っています。

この中で、現在、画像認識でもっとも使われているのが CNN（畳み込みニューラルネットワーク）です。これは、ILSVRC のコンテストでも使われているモデルです。文字を識別するとき、人は全てを見なくても分かります。なぜなら、人は特徴を見つけて文字を判断しているのです。文字の特徴を抽出すれば、画像認識がうまくいくのではないかというのが、CNN のコンセプトになります。要するに、人の見方と同じような考え方を導入しようということです。

どのような処理をしているかというと、図 16 に示すように、例えば、「あ」という文字を 32×32 で区切ります。このデータは、単に飛び飛びの点で組み合わされているのではなく、結構線状に連続していたりして、それぞれの部分部分で特徴があるのです。なので、特徴を探し出すために、一定の区画ごとにフ

ィルタリングをかけて抽出します。例えば、5×5 で斜めの特徴を見つけたいなら、これで一旦フィルタリングをかけます。これ以外の特徴に対しても、フィルタリングをかけることで特徴だけを次々と抽出していきます。このような処理のことを畳み込みと言います。

色々な特徴をたくさん抽出して、そのままではデータが大きくなるので、さらにプーリング、すなわち圧縮処理を施します。このプーリングしたものをフィルタリングにかけて、特徴を導き出します。このように処理を繰り返して、最後に全結合のニューラルネットワークが書けるのです。CNN は、このように処理することで画像の特徴を抽出して、「あ」とか「い」とかの文字を抽出するアルゴリズムなのです。これらのモデルは一般に公開されているので、簡単に使うことができます。

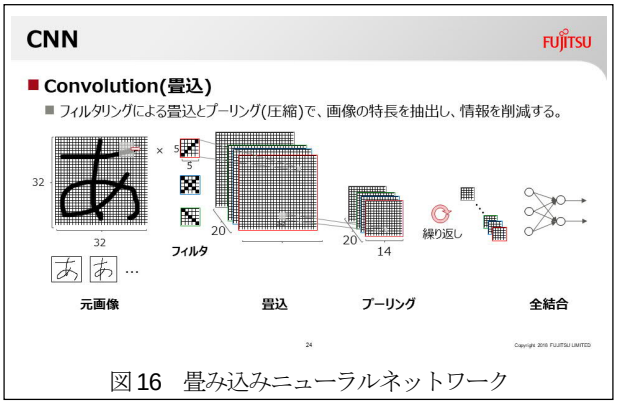


図 16　畳み込みニューラルネットワーク

当社は、DeepTensor というグラフデータ学習技術、人やモノのつながりを表すグラフデータから新たな知見を導くディープラーニングの新技術を提供しています。一般的な技術と当社独自技術を使って、顧客の様々なニーズに対応できるようにしています。

ディープラーニングをプログラミングするときには、フレームワークを使います。行列計算式を全て人が書いていたのでは大変なので、また、誤差が生じたときの重み付けを人が直すのも大変なので、これらを全てしてくれるフレームワークが用意されています。オープンソースになっていて、無償で使えます。なので、これらオープンソースを使うことで、簡単にプログラムが組めるようになっています。

たくさんあって、どれを使ったらよいのか悩みますが、図 17 はフレームワークの人気度[*1] を示したものです。一番人気が TensorFlow で次が Keras です。Keras は、TensorFlow をラッピングしているフレームワークで、さらに簡単に使えるモデルとなっていて、急激に伸びてきています。なので、TensorFlow を勉強すれば、比較的情報量も多いので、簡単にプログラムが組めるようになります。

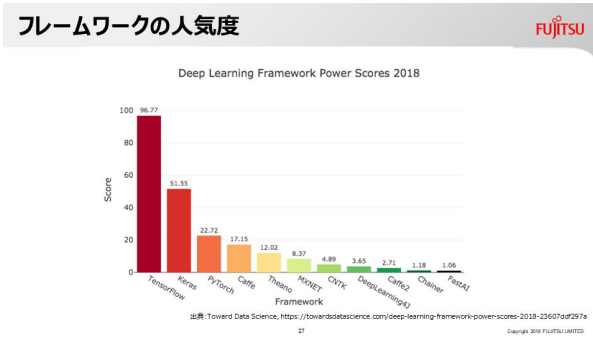
先ほどの図 12 のモデルを TensorFlow で書いてみると、複雑な行列演算とか、間違ったものの重み付けを直すとかも含めて、わずか 73 行で書けてしまいます。

TensorFlow はコンソールで書きますが、最近は GUI が用意されていて、もっと簡単にディープラーニングができます。当社は、Zinrai という名称で体系化していて、クラウドでこれを提供しています。他には、Sony、NVIDIA などが GUI で提供しています。

当社の Zinrai をご紹介します。画像を認識して、それが何かを当てるという例です。教師データは、24 種類 3 万枚用意しました。このデータを使って、このモノが何かを推論するプログラムを実際に作ってみたいと思います。

まず、フォルダにデータを用意します。教師データはインターネットから拾ってきます。次に、学習ウィザードを立ち上げて、プログラムの名前、ネットワーク種別、教師データ、最終的に学習済みにするモデル、パラメータなどを設定して、プログラムを走らせます。学習が完了すると、学習済みモデルができ上ります。これで画面上の認識用画像を選んで、推論できるようになります。

Zinrai の特徴は、エッジモデルも作ることができます。スマートフォンに学習結果を送って、カメラで撮って、その場で推論ができる仕組みも用意しています。例えば、お茶の葉の開き具合をその場で確かめたいのであれば、葉の状態を学習させておいて、現場で写真を撮って、摘み取るタイミングかどうかの判定モデルを簡単に作ることができます。



出典：Toward Data Science,
https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a

図 17　フレームネットワークの人気度

計算機パワー

先ほどの図 14 の学習では 10 時間以上かかるという話をしましたが、これでは 1 日に 1 回しか学習できません。競合他社がいるので、この時間をいかに短くできるかが重要なポイントとなります。それには、計算機パワーが必要だということです。なぜ GPGPU を使うとディープラーニングができるかというと、ニューラルネットワークは行列の計算です。同じ計算をするなら、GPGPU にやらせた方が圧倒的に速いのではないかということで、GPGPU がニューラルネットワークの計算に使われるようになりました。

汎用 CPU でも計算はできます。図 18 に示すように、汎用 CPU と GPGPU でクロック速度も消費電力も似かよったものですが、明らかに構造が違います。汎用 CPU は連続で複雑な処理が得意で、GPGPU は単純で大量な処理が得意です。画像の点を一気に書き換えるには一気に計算しないといけないので、並列処理が得意でないといけないです。コア数を比べたら、汎用 CPU が 16 に対して GPGPU は 5,120 と数が大きく違います。なので、GPGPU は特定のことしかできないが、行列演算だけを考えたら、圧倒的に GPGPU の方が処理が速いです。それでは、普通の表計算を GPGPU にやらせたらどうかというと、それぞれ得手不得手があって、使い方に応じて選択するというのが一般的な考え方です。

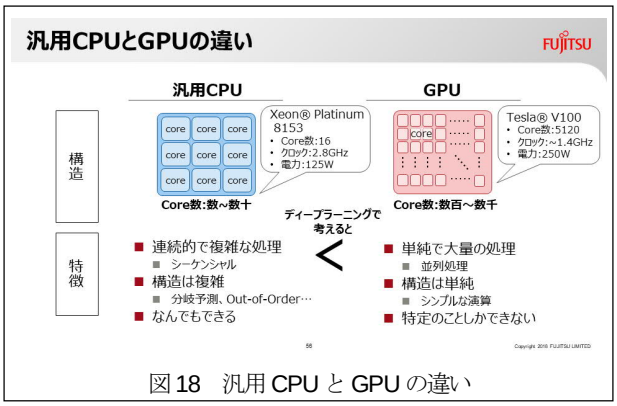


図 18　汎用 CPU と GPU の違い

計算時間は 27 倍ぐらい違っていて、圧倒的に GPGPU の方が速いです。汎用 CPU でもできないことはないですが、図 12 のモデルで 1 分ぐらいかかります。大きな画像を扱うとか、CNN で大きなネットワークを作るとなると、それなりの計算能力が必要になるということです。コンピューティングアーキテクチャは、汎用 CPU で何でも扱える時代は終わりを迎えていて、図 19 に示すように、ディープラーニング、スーパーコンピュータ、量子コンピューティングなどをそれぞれの用途に応じて使い分けるのが、今の流れだと思っています。

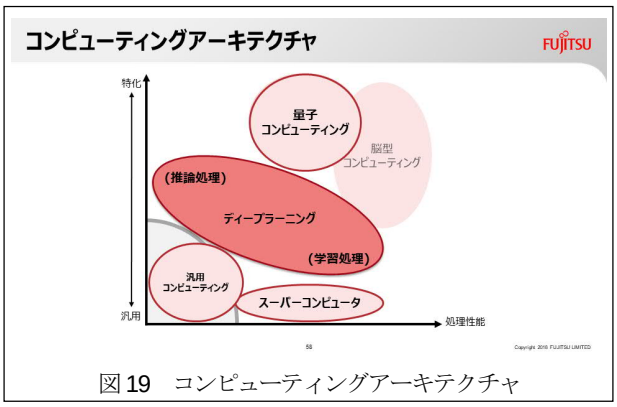


図 19　コンピューティングアーキテクチャ

ディープラーニング用プロセッサは、学習用と推論用で分けられます。学習には大きな CPU パワーが必要ですが、推論はそうでもないです。学習用で有名なところは、Google の TPU、Xeon の Phi、NVIDIA の Tesla®V100 です。今ほとんど、これらが使われていると思います。当社も参入しています。推論用は、FPGA チップに独自の回路を組み込んで使うのが、今の主力となりつつあります。

当社はメインフレームからスーパーコンピュータまで、ずっとプロセッサを独自で作っています。この技術を使ってディープラーニング用チップを作ろうということで、今は電力性能 10 倍を目指して取り組んでいるところです。

ＤＬＵ（Deep Leaning Unit）

当社の DLU の基本的な考え方は、小さいコアをたくさん並べて、同時に行列演算させようというものです。図 20 に示すように、チップ内に DPU というユニットを作って、その中に DPE というさらに細かい演算ユニットを作って、同時に計算できるプロセッサを構成します。1 チップでできることは限られているので、脳というすごいネットワークと同じような構成にするためには大規模にしなければいけないということで、「京」を作ったときのネットワークのように 6 次元のネットワークを組み込むことで、超並列処理ができるように今開発を進めているところです。

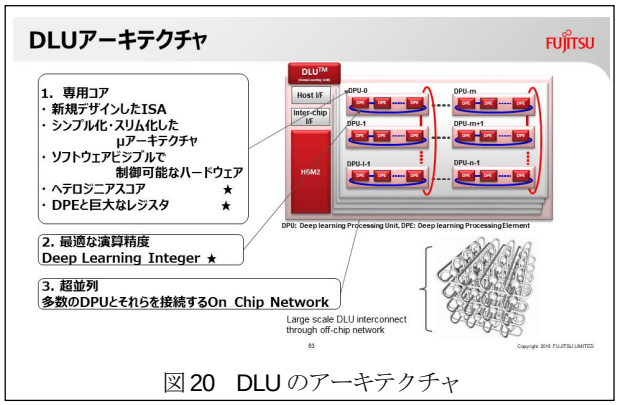


図 20　DLU のアーキテクチャ

同時に実行させるために Master コアを用意して、小さなコアである DPU に「計算して」と投げるようなアーキテクチャになっています。半導体チップ内にいかに多くの演算ユニットを詰め込めるかがポイントになると思っていて、不要なものを取り去るということで、ディープラーニングに特化した形でハードウェア設計を行っています。

1 つ特徴的なのは、普通のプロセッサは演算器があって、レジスタがあって、キャッシュメモリがあるのですが、DLU にはキャッシュメモリがありません。なぜかというと、重み付けの値はすぐ変えてしまうので、キャッシュに覚える必要がないのです。なので、キャッシュを全て取り払って、図 21 に示すように、大きなレジスタファイルを置いて、レジスタと演算器をなるべくたくさん並べるというのが、ディープラーニングユニットの特徴となっています。

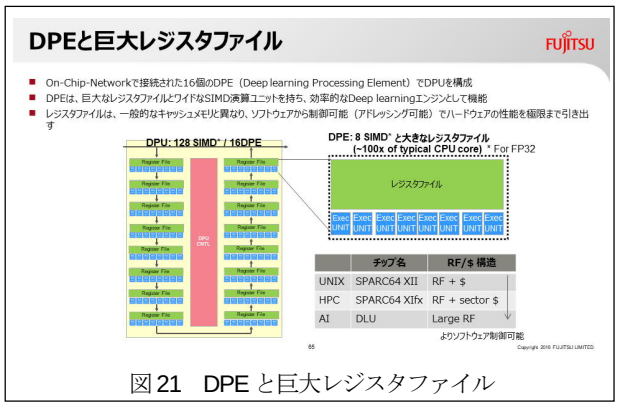


図 21　DPE と巨大レジスタファイル

2 つ目の特徴は演算精度です。GPU で 3D 演算するには、8bit 計算では画像が崩れてしまうので、32bit 倍精度の浮動小数点演算で計算しています。そうすると、計算が重くなってしまいます。ディープラーニングでは、それほどの精度はいらないということで、精度を落とした計算ができる仕組みを取り入れています（図 22）。

演算精度を落として、最終的に 8bit で計算するようにすると、32bit の 4 倍の計算能力になります。もっとも、8bit 計算で表せるのは 256 とおりしかないので、演算精度がガクッと落ちてしまいます。そこで、演算精度は 16bit にしておいて、統計情報

をとって数値が集中している中央付近の数値を返すようにして、数値があまりない端の方は捨ててしまって、8bit 内で表せるように工夫しています。評価試験をして、数値を返せる範囲を決めて、そこの数値を返すようにすることで、演算精度が落ちないようにしています。

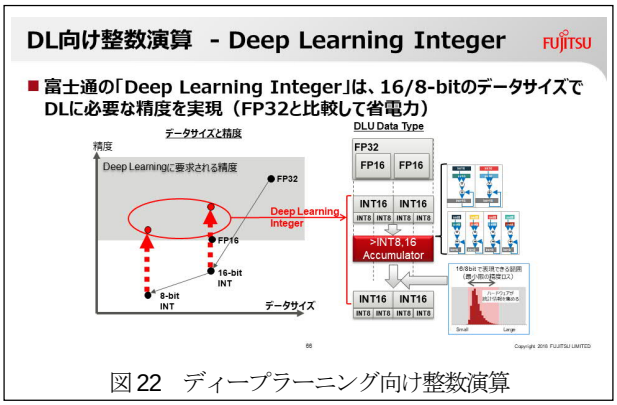


図 22　ディープラーニング向け整数演算

通常、8bit では学習精度が落ちます。しかし、図 22 のようなディープラーニング用 8bit 整数演算器を使うと、従来の 32bit 演算と遜色ないようにできることが実験データから分かっています。今は、こういう仕組みを取り入れながら、より速い計算を目指しているところです。まだ完成形には至っていませんが、最終的にはサーバの PCI スロットに差し込めるような製品を考えています。今年度中には製品化できるように、今開発を進めています。他社ができないような大規模ネットワークを構築できるよう、設計しているところです。

まとめ

ディープラーニングの技術は急速に進化しています。今日できないことが、もしかしたら、来週にはできるようになっているかもしれません。そして、より複雑な処理ができるようになっていくのだろうと思っています。そのことは、より高い計算機パワーが必要になると認識しています。例えば、工場で製品の傷を見るのに、すごく細かいカメラで撮ります。5,000×5,000 pixel のカメラで撮ると、その分データも大きくなります。それを処理するためには、大きな計算機パワーが必要になります。当社は、今の DLU の後継機を開発して、より処理能力の高い、より省電力なものを供給できるように進めていきたいと思っています。

開発環境もずいぶん整備されました。7～8 年前とは格段に違います。テキストベースで組んでいたプログラムは、GUI が使えるようになって、Keras を使って組むと短く書けます。ディープラーニングの環境は、ソフトウェア／ハードウェアどちらも改善されていくと思っています。これからより簡単に使えるようになって、ディープラーニングが市場にますます広がっていくものと思っています。

後半、駆け足ぎみになりましたが、話は以上です。ありがとうございます。

本講演録は、平成 30 年 12 月 7 日に開催された SCAT 主催「第 103 回テレコム技術情報セミナー」のテーマ、「AI のトレンドとソフト・ハード両面の最新の取組」の講演内容です。