

機械学習における人工知能を用いた平面図形の定理の発見法の研究

東京工科大学 コンピュータサイエンス専攻
前田 充輝

1. 背景

幾何学における定理はシムソンの定理やその拡張系であるターナーの定理や清宮の定理など様々な定理が存在するが、現在それらの定理はひらめきや既存の定理の拡張によって人間の手により発見されている。また、幾何学における定理の模索は古くから行われているが、近代においても新たな定理が発見されていることを見ると、見落とされている定理が存在する可能性は極めて高い。

現在、機械学習の分野においてはディープラーニングによる学習が主流になってきている。ディープラーニングによる学習は入力と出力がまとめられた大量のビッグデータを教師データとして扱い、それらを参考に予測を行っていく。また、ディープラーニングでは計算アルゴリズムとしてニューラル、ネットワークが活用されている。このニューラル、ネットワークは、人間の脳機能に見られるいくつかの特徴を模倣したアルゴリズムであり、入力層、中間層、出力層の3層に分類される。ニューラル、ネットワークはこの中間層を多く重、ることによって精度を向上することができる。ディープラーニングは質の高い教師データと量さえあればかなり精度の高い予測が期待できるため、現在多くの研究者によって研究されている分野である。

2. 目的

既存の平面図形の定理を人工知能に学習させ、そこから定理を導かせることを考える。
学習させる既存の定理は、シムソンの定理やその拡張である清宮の定理といったそれぞれ似ているものを使用する。このようにシムソンの定理や清宮の定理のような似かたよった定理を学習させることによって、図形を学習させるためのノウハウを発見し、新定理発見への足掛かりとする。

使用する人工知能は日本の機械学習系ベンチャー企業である Preferred Networks によって開発されたニューラル、ネットワークを実装するためのライブラリである Chainer を利用して実装する。これによって、まず入力データと正解データから二乗平均誤差を求め、誤差逆伝播(Backpropagation)を行って正しい値に近くなるまで学習を繰り返し最適化する。これによって出力された値をより精度の高いものへと改良していく。

3. 研究内容

既存の平面図形の定理を複数人工知能に学習させ、そこから人工知能に定理を導かせる。
学習させる既存の定理はシムソンの定理やその拡張である清宮の定理といったそれぞれ似ているものを使用する。使用する人工知能は Preferred Networks によって開発されたニューラル、ネットワークを実装するためのライブラリである Chainer を利用して実装する。これによってまず入力データと正解データから二乗平均誤差を求め、誤差逆伝播(Backpropagation)を行い、これによって出力された値をより精度の高いものへと改良していく。このようにシムソンの定理や清宮の定理のような似かたよった定理を学習させることによって、図形を学習させるためのノウハウを発見し、新定理発見への足掛かりとする。

・シムソンの定理

$\triangle ABC$ の外接円上の点 P から 3 辺 BC, CA, AB に下した垂線の足 X, Y, Z は同一直線上にある。

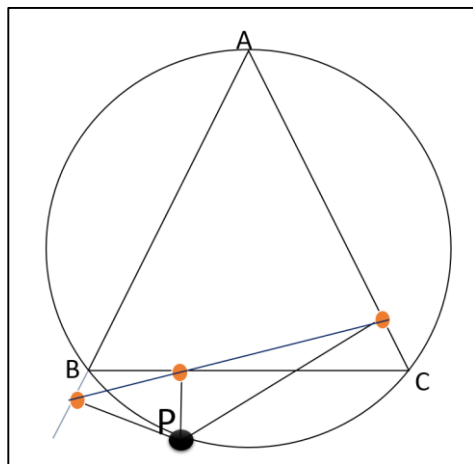


図1 シムソンの定理

・シムソンの定理の chainer による予測

三角形 ABC の三頂点 $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ を決めて固定する。次に点 P の X 座標 x_P と Y 座標 y_P を乱数で生成し、これを入力とする。

点 P が外接円内にあれば折れ線 ZXY は上曲がり、外接円外にあれば下曲がりとなるため、点 P の位置をこれにより判別することが可能となる。点 P から 3 辺 AB, BC, CA に下した垂線の足 X, Y, Z の座標を $(x_X, y_X), (x_Y, y_Y), (x_Z, y_Z)$ とすると XY, YZ の直線の方程式は次のようになる。

まず辺 BC の方程式は

$$\left\{ \begin{array}{l} a = \frac{y_B - y_C}{x_B - x_C} \\ y = a(x - x_B) + y^B \end{array} \right.$$

である。点 P から BC に下した垂線の方程式を

$y = kx + b$ とすると

$k = -1/a$

$b = y_B - kx_B$

ゆえにこれと BC の交点の X 座標 x_H と Y 座標 y_H は

$$x_H = \frac{y_P - k \times x_P - b}{a - k}$$

$$y_H = kx_A + k \times y_P$$

次に 3 つの交点を左から Y, X, Z とし、直線 YX と XZ

の傾きを求め、YX の傾きを α , XZ の傾きを β とする。そして $\gamma = \frac{\alpha - \beta}{1 + \alpha\beta}$ を求め、 $\gamma > 0$ ならば上曲がり、 $\gamma < 0$ ならば下曲がりであると判別でき、これを出力とすることで AI に点 P が外接円の外にあるか中にあるかを判断できる。

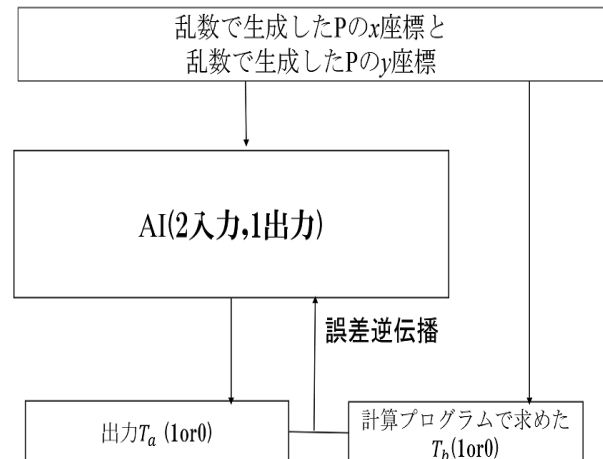


図3. プログラム全体の構成図

5. 参考文献

[1] Credo

<http://credo.asia/2015/07/21/what-is-deep-learning/>

[2] IT media ビジネス

http://www.bizmakoto.jp/makoto/articles/1507/20/news006_2.html

[3] 斎藤康毅(著), ”ゼロから作る Deep Learning —Python で学ぶディープラーニングの理論と実装”, オライリー・ジャパン, 2016 年

[4] 清水亮(著), ”はじめての深層学習プログラミング”, 反術評論社, 2017 年