**スーパーコンピューター上での高速化に向けたコード最適化技術の検討**

Investigation of code optimization techniques for acceleration on supercomputers

物理情報工学科　渡辺研究室　学籍番号：61713173　内藤翔太

Abstract:

We trained two types of images in an all-coupled three-layer neural net and investigated the decision criteria of the generated models. We found that the models judge by the total number and position of points in the input images, not by the shape of the figures. We trained the model with normalized data so that the model could not utilize the total number or position of points. Then, when the input image was inferred, it was suggested that the model did not recognize two types of images, but only one of them.

Keywords: Machine Learning, Vulnerabilities

1. 研究背景・目的

先日、理化学研究所のスーパーコンピューター「富岳」が性能を評価する４部門で首位となり、総合的な性能の高さを証明した。スーパーコンピュータを利用するにあたり、重要視されるのは「計算速度」であり、これには「コード最適化」が欠かせない。この「コード最適化」手法の一つに「SIMD化（single-instruction-multiple-data）」という「一つの命令で複数の計算を行うもの」があるが、これは職人技を必要とする。そこで、SIMD化を含むコードの自動生成を行うことでスーパーコンピュータの普遍化を目的とする。このコード自動生成にあたり機械学習を用いるが、機械学習には危険がはらんでいる。この最たる例が「adversarial-example[1]」である。これは、分類器に対する脆弱性攻撃で、分類器が正確な分類をしていた画像に、人間に見えない程度のノイズをかけると恣意的に分類器の判断を誤らせることができるといったものである。そこで、

機械学習への理解を深めるため、機械が何を特徴量として学習しているのかを調べる。

2. 方法

まず、マルとバツの正解データを用意し、その画像を学習させた分類器を作成した。続いて、この分類器が「何を根拠にマルもしくはバツと判断しているのか」を調査するため、以下の実験を行った。

（実験１）「点の個数」依存性：入力データの点の個数の違いによって、分類器がマルまたはバツと判断する可能性が変化するかを調べた。

（実験２）「点の位置」依存性：入力データのどこに点があるかによって、分類器がマルまたはバツと判断する可能性が変化するかを調べた。

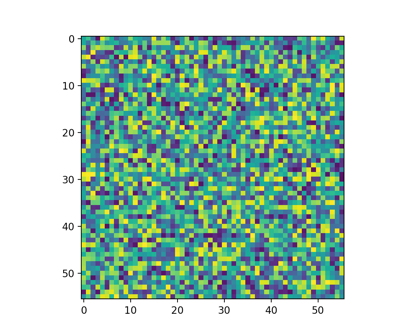
（実験３）「点の個数」・「点の位置」により判別できないように正規化したデータで分類器を学習させ、理想的なマル・バツを逆推定させた。

3.　結果

（実験１）ランダムな点を入力としているため、点の個数によらずマル・バツの確率は0.5になると期待されたが、実際には点の個数が増えることでマルの確率が高くなった。これは機械が、「点の個数が多いとマルである確率が高い」と判断していることを示唆する。

（実験２）点の位置によって、マル・バツの確率が変化した。これは機械が、「入力データに存在する点の位置」によってマル・バツを判別していることを示唆する。

（実験３）「点の個数」・「点の位置」により判別できないように正規化したデータで分類器を作成した。この分類器によるマル・バツの逆推定をFig. 1、Fig. 2に示す。

** パソコンの画面

自動的に生成された説明**Fig. 1 Circle inverse estimation Fig. 2 Cross inverse estimation

Fig. 1,2は、分類器がバツの逆推定は行えているが、マルの逆推定を行えていないことを示唆する。

4　結論と今後の展望

本研究では、学習済みの分類器に、工夫したデータを入力することで、分類器が何を基準に判定しているのかを調べた。今後は、機械学習を用いたSIMDコードの自動生成に、本研究で得られた知見を適用し、スーパーコンピューター上での効率的なコード生成を目指す。

参考文献

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy,

International Conference on Learning Representations (2015).