

# 深層学習統合開発環境を用いた学習モデルの GUI 設計に関する研究

1190181 和田晋治 (集積システム研究室)

(指導教員 密山 幸男 准教授)

## 1. はじめに

機械学習を利用するためには Python や C++ などで複雑なプログラムを書かなければならず、技術的なハードルが高い。そこで、プログラムを書かずに容易に機械学習を行うことができ、開発効率を大幅に改善できるディープラーニング用 GUI ツールを SONY が開発した。

本研究では SONY のディープラーニング用 GUI ツール “Neural Network Console” を用いて、既存のニューラルネットワークを開発、評価できることを確認した。

## 2. 深層学習の概要

深層学習とは、機械学習の手法の 1 つである。深層学習の代表的な手法である畳み込みニューラルネットワークには、畳み込み層とプーリング層を繰り返す LeNet などがある。

## 3. Neural Network Console の概要

Neural Network Console では、コンポーネントの形でレイヤ (関数ブロック) を自由に配置することでニューラルネットワークを構築できる。ボタン 1 つで学習を開始し、学習の進捗状況や結果をリアルタイムに確認できる。学習した何十種類ものニューラルネットワークを履歴として一覧でき、過去に設計したニューラルネットワークの性能も一覧して比較できる。

一例として、設計したニューラルネットワークを用いて学習させた結果を図 3.1 に示す。グラフの横軸は Epoch であり、学習の試行回数を示す。青色のグラフの縦軸は Cost の出力であり、ニューラルネットワークの出力と実際の出力の差を示す。赤色の実線グラフは training error、赤色の破線グラフは validation error、青色の実線グラフはコスト関数 (損失関数) である。縦軸は training error と validation error の誤差を表している。training error は訓練データとの誤差 (訓練誤差) であり、validation error は評価データとの誤差 (汎化誤差) である。学習回数を増やすことによって訓練誤差は減少するが汎化誤差が大きくなり、未知のデータに対する予測性能が低下する過学習と呼ばれる現象が発生することがグラフからも確認できる。

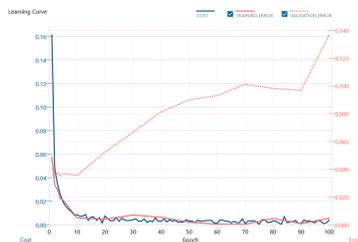


図 3.1 学習結果

## 4. Neural Network Console を用いた比較実験

過学習の対策として、Dropout や Batch Normalization といった処理を適用する方法が提案されている。

そこで、既存のニューラルネットワークである LeNet に Dropout や Batch Normalization を適用し、その有効性を比較評価した。手書き数字認識データセット MNIST のモノクロ手書き数字画像 (28x28 ピクセル) を学習用画像 60,000 枚と評価用画像 10,000 枚に分けた。LeNet の構成を図 1 に、その学習結果を図 5 に示す。Dropout を加えた LeNet の構成を図 2 に、その学習結果を図 6 に示す。Batch Normalization を加えた LeNet の構成を図 3 に、その学習結果を図 7 に示す。Dropout と Batch Normalization の両方とも加えた LeNet の

構成を図 4、その学習結果を図 8 に示す。また、表 1 に 100epoch 時の訓練誤差、汎化誤差、コスト関数を示す。

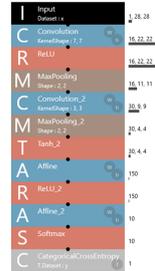


図 1 LeNet



図 2 Dropout を加えた LeNet

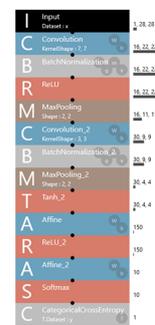


図 3 Batch Normalization を加えた LeNet

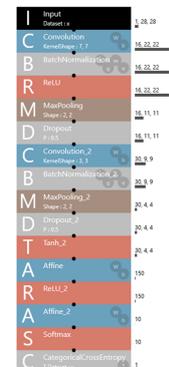


図 4 Dropout と Batch Normalization を加えた LeNet

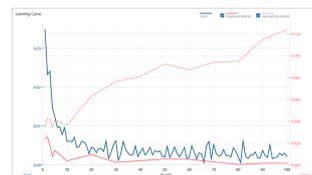


図 5 LeNet の学習結果

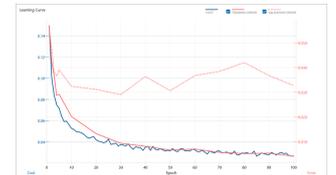


図 6 Dropout を加えた LeNet の学習結果



図 7 Batch Normalization を加えた LeNet の学習結果

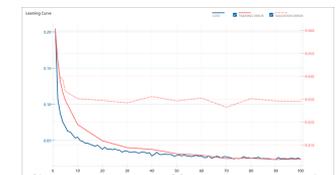


図 8 Dropout と Batch Normalization を加えた LeNet の学習結果

LeNet に Dropout を加えると汎化誤差が 0.121 から 0.033 まで減少したが、訓練誤差が増加した。Batch Normalization を加えると汎化誤差が 0.121 から 0.092 まで減少した。Dropout と Batch Normalization を加えると汎化誤差が 0.121 から 0.029 まで減少したが訓練誤差が増加した。

## 5. 結論

本研究では、Neural Network Console を使い Dropout や Batch Normalization の過学習を抑える効果を確認した。

今後の課題として、さらに過学習を抑え、分類の精度を向上させるなどが挙げられる。