タイトル	ディープラーニングの概要および北海学園大学工学部コンピュータ実習室における機械学習環境
著者	串山, 繁; Kushiyama, Shigeru
引用	工学研究 : 北海学園大学大学院工学研究科紀要(18): 11-17
発行日	2018-09-30

研究解説

ディープラーニングの概要および北海学園大学工学部コンピュータ実習室における機械学習環境

串　山　　繁*

Outline of Deep Learning and Circumstances of Machine Learning in the Computer Room in Faculty of Engineering, Hokkai-gakuen University

Shigeru Kushiyama*

要　旨

最近のニュースでは，AI（Artificial Intelligence：人工知能）に関する話題が頻繁に取り挙げられ，様々な分野において期待感が増大し，日本政府も若手データサイエンティストを育成するための予算を確保して漸く支援の手を広げようとしつつある．AI が現在注目されているのは，機械学習とりわけディープラーニング（深層学習）が画像処理などの知覚問題において様々な成功を収めていることに起因する．

本資料では，ディープラーニングの概要，本学工学部計算機実習室における機械学習環境を紹介すると共に TensorFlow を用いた簡単な回帰問題の事例を示す．

1．序

AI を支える核心的な技術がディープラーニングをはじめとする機械学習である．今日に至る経緯を振り返ると，大きく以下の３つのステージに分かれる様である．

第１ステージは，1980 年代末を終わりとする Symbolic AI の時代[1] である．それは人間が読める記号表現を用いてルールを定義し，プログラムにそれを組込み実行するものであるが，ファジーで複雑な画像分類，言語翻訳等には対応が困難であった．

第２ステージは，1990～2010 年にかけて研究された SVM（Suport Vector Machine）の時代[2] である．これはカーネル法を用いてクラス分け問題を高次元に写像して分離超平面を求めるもので，単純な分類問題では高い性能を示すが，画像分類などの知覚問題については不得手であった．

第３ステージは，2010 年以降開花したディープラーニング（Deep Learning, DNN：Deep Neural Network とも呼ぶ）である．ニューラルネットワーク自体は既に存在していたが，特徴量の重みやバイアスのパラメータ設定が専門家の芸術的なセンスに依存し，現実問題に適用するには困難であった．

ディープラーニングがブレークスルーした要因として，以下の３つ；ハードウェア，データセット，アルゴリズムの進化が指摘されている．参考文献[3] に拠れば，その詳細は以下の様である．

①1990 年代と 2000 年代のボトルネックは，ハードウェアとデータであるが，1990～2010 年の間に CPU は約 5000 倍に高速化，ゲーム市場が契機となって GPU（Graphics prossesing unit）と呼ばれる高速な超並列チップが開発された．②データ面ではストレージハードウェアが飛躍的に発展したことに加え，インターネットの台頭で機械学習用のデータセットの収集，配布が容易となった．③アルゴリズムの進化としては，誤差逆伝播法が再評価され，損失値の最小化が確率的勾配降下法（SGD：Stochastic Gradient Descent）を用いて効率良く成される様になったことである．併せて活性化関数やパラメータの調整を制御する最適化関数の改善もある．以上主に３つの要因が相乗効果の役割を果たし，今日のブレークスルーに至って

* 北海学園大学大学院工学研究科建設工学専攻（建築系）
Graduate School of Engineering (Architecture and Building Eng.), Hokkai-Gakuen University

いる．

応用事例としてはいずれも開発進展途上であるが，医療診断，監視カメラ画像の自動識別，知的なチャットロボット，Google 翻訳，Amazon の製品推奨システムなど多岐に亘る．

2．ディープラーニング（DNN）の概要

2.1　DNN の仕組み

機械学習は教師あり，教師なし学習およびその中間の強化学習に分けられるが，以下では教師あり機械学習を例に説明する．教師あり機械学習では，訓練用とテスト用のデータから，未知のデータに対する答えを導くルールを学習する．ディープラーニングのディープとは，図-1 に示す隠れ層（hidden layer）が連続し層数が多いことを意味する．図中の◯印はノードと呼ばれる．

上図を用いて，回帰を例に DNN の仕組みを説明する．

最初に学習のための特徴量（feature）および教師値（target）から成る観測データを抽出し，3 つの特徴量を入力層（input layer）に与える．次いで，後掲する線形回帰式に非線形の活性化関数（activation function：af）を作用させて 2 つの隠れ層，出力層（output layer）のノード値を求める．出力層の予測結果を教師値と比較して，誤差に相当する損失値（loss）を損失関数（loss function）を基に算出し，損失値を最小化するために，確率的勾配降下法に従って重み（weight）とバイアス（bias）を更新する．更新の制御は最適化関数（optimizer）で行う．図-1 の出力層へ向かう計算（forward pass）および逆向きの計算（back pass）を充分損失値が最小化するまで指定回数繰り返し，学習を終える．学習後，訓練データの損失値とテストデータの損失値の大小関係を比較し，訓練結果が過剰適合していないことを確認する．

DNN の目的，即ち学習内容は，重みとバイアスの最適なパラメータ値を得ることである．上記の学習結果を受けて未知の特徴量に対する予測値を得る．なお，中間の隠れ層についてはユーザーが層数と各隠れ層のノード数をソースコードで指定する．

以下に，図-1 を例に出力層へ向かう線形回帰式を示す．図中の線分は重みを表し，各ノードに 1 つのバイアスを考慮すると，第 1 層目の隠れ層は次式で表される．

$$_1H_1 = af(_1w_{11}*X_1 + _1w_{21}*X_2 + _1w_{31}*X_3 + _1b_1)$$
$$_1H_2 = af(_1w_{12}*X_1 + _1w_{22}*X_2 + _1w_{32}*X_3 + _1b_2)$$
$$_1H_3 = af(_1w_{13}*X_1 + _1w_{23}*X_2 + _1w_{33}*X_3 + _1b_3)$$
$$_1H_4 = af(_1w_{14}*X_1 + _1w_{24}*X_2 + _1w_{34}*X_3 + _1b_4)$$

同様に第 2 層目の隠れ層は次式となる．

$$_2H_1 = af(_2w_{11}*_1H_1 + _2w_{21}*_1H_2 + _2w_{31}*_1H_3 + _2w_{41}*_1H_4 + _2b_1)$$
$$_2H_2 = af(_2w_{12}*_1H_1 + _2w_{22}*_1H_2 + _2w_{32}*_1H_3 + _2w_{42}*_1H_4 + _2b_2)$$
$$_2H_3 = af(_2w_{13}*_1H_1 + _2w_{23}*_1H_2 + _2w_{33}*_1H_3 + _2w_{43}*_1H_4 + _2b_3)$$
$$_2H_4 = af(_2w_{14}*_1H_1 + _2w_{24}*_1H_2 + _2w_{34}*_1H_3 + _2w_{44}*_1H_4 + _2b_4)$$

最後の出力層については，次の通りとなる．

$$Y' = af(_3w_{11}*_2H_1 + _3w_{21}*_2H_2 + _3w_{31}*_2H_3 + _3w_{41}*_2H_4 + _3b)$$

ただし，af：活性化関数，最終パラメータ総数：(3*4＋4) + (4*4＋4) + (4*1＋1) = 41

2.2　DNN の学習手順

以下に全訓練データをミニバッチに分けて学習する手順とそのフローを図-2 に示す．

学習手順

1．訓練データ X と教師値（即ち目的値）Y をバッチデータとして抽出

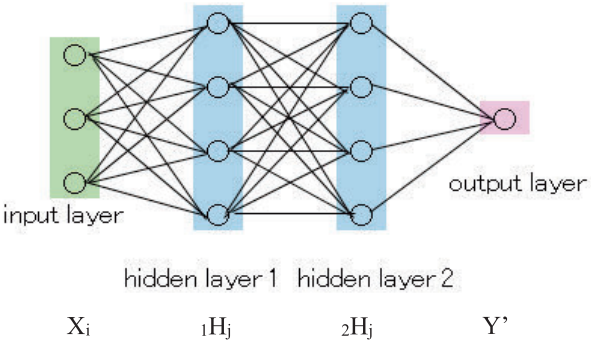


図-1　DNN 模式図

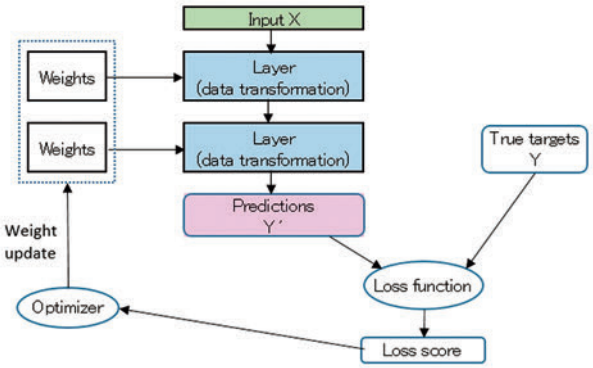


図-2　DNN 訓練手順のフロー（参考文献[3] 参照）

2．X を付与し DNN を実行．予測値 Y' を取得
3．損失関数を用いて予測値 Y' と目的値 Y から当該バッチの損失値を計算
4．確率的勾配降下法（stochastic gradient descent）を用いて損失関数の勾配（gradient）を計算
5．W＝W－step×gradient により勾配とは逆方向に少し移動し，当該バッチの損失値を低減する様，重みとバイアスを更新
6．各バッチ毎に訓練データの学習を指定回数（epoch 数）繰り返して重みとバイアスを最適化．この制御は最適化関数で指定

3．工学部コンピュータ実習室における機械学習環境

北海学園大学工学部計算機実習室には，2018 年 3 月にフリーでダウンロードできる以下の機械学習関連ライブラリをインストールした．

・Anaconda[1] を用いて，scikit-learn，Numpy，Scipy, matplotlib, Ipython, Jupyter Notebook
なお，Python は 3 系である．
・mglearn, TensorFlow, TensorBoard, pandas

上記は実習室Ⅰ～Ⅲの全コンピュータに配布する都合上，TensorFlow の仮想環境で動作する．なお，実習室においては，必要に応じて上記の各種ライブラリをインポートし，Python 3 系でソースコードを記述する環境であり，R 言語はインストールされていない．

scikit-learn は，機械学習初心者が最初に利用する良い学習材料と云われている．一方，TensorFlow は，Google Brain チームが開発し，2015 年 11 月に一般に公開した機械学習等に用いるソフトウェアライブラリで，アルゴリズムの実装を簡単にする高レベル API（Application Programming Interface）：Keras と組み合わせて用いることが多い．通常，Jupyter Notebook を用いてブラウザ上でプログラムを編集，分割して実行する．また，markdown 形式でソースコードの解説も必要箇所に自由に記載可能で，LaTeX の指定で数式も表記できる．なお，計算機実習室での Jupyter Notebook の起動手順は，表-1 に示す通りである．

[1] 大規模データ処理，予測解析，科学技術計算向けの Python ディストリビューション

表-1　実習室での Jupyter Notebook 起動手順

1）スタート／Anaconda Prompt を選択
2）Prompt 起動画面にて，>以下を入力
　# tensorflow の仮想環境に切り替え
　　>conda activate tensorflow
　# Jupyter Notebook を起動
　　>jupyter notebook
3）Jupyter Notebook 起動画面にて，
　新規ソースコード入力時：
　　画面右端 New を押下，Python 3 を選択
　作成済ソースコード訂正時：
　　一覧から. ipynb の該当ファイルを選択
4）編集済ソースコードは，File/Rename を選択して，適当なファイル名で保存
5）ファイル保存先は下記の通り
　デスクトップ/ユーザー名のフォルダ内

各種ライブラリがある中で TensorFlow を加えた理由は，Keras に加えて可視化を補完するツール：TensorBoard が付属し，実装モデルの計算グラフや様々なデータログを取得することに拠り，学習状況などを容易に可視化して実装モデルを検討できるからである．当然ながら隠れ層の設定も容易で最適化手法も各種取替え可能，パラメータは自動チューニングされ，利用者の負担は軽い．

大規模な画像処理問題を実装する場合には，先に述べた GPU がハードウエアに内蔵されている必要がある．本格的には，GPU を複数備えた Workstation，Ubuntu 上で機械学習関連アプリケーションを開発することが理想と指摘されている[3]．実習室のコンピュータは GPU を備えておらず，Windows 上の CPU で処理可能な範囲に制約されるが，scikit-learn の参考文献[4] に記載されている画像を扱うサンプル例の実行では殆ど支障は無かった．

なお，Jupyter Notebook で使用するブラウザを Internet Explorer とした場合，複数ページにまたがる編集・実行結果の印刷が 2 ページ以降出力されない．原因は不明であるが，他のブラウザを使用すればこれを回避できる．

4．回帰問題の事例

ここでは，TensorFlow と Keras を用いた回帰事例として，建物の定量的安全性評価への実装可能性についての検討事例を示す．

文献[5] では，実在 RC 造 11 層建物を例に地震動

を確定的，構造特性を不確定と仮定し，シミュレーションベースの subset MCMC[6] を用いて建物の破壊生起確率を定量的に評価することを試みたが，10^{-6} オーダーの破壊確率を求めるには subset MCMC を用いても 15 万回程度の応答計算を行う必要があり，計算負荷は非現実的なレベルに近かった．そこで本報では，上記応答計算のある一定割合を深層学習の結果で置き換えることの可能性について検討した．

4.1　解析仮定

入力地震動は，図-3 に示す JMA 神戸 NS の地表面最大速度：PGA（Peak Ground Acceleration）を 50 kine に基準化したレベル 2 地震動を用いた．

解析モデルは文献[5] に同じ RC 造 11 層，X 方向 2 スパン，Y 方向 2 スパン建物をせん断型架構に置換した表-2 に示す構造階高，層重量等を有する質点系モデルである．非線形応答解析を行う際の仮定は，スケルトンカーブ：tri-linear 型，履歴則：masing 型，構造物の粘性減衰：5 % の Rayleigh 減衰（2 次迄考慮）と先の文献に同じである．なお，構造物の破壊指標は，最大応答塑性率 μ_{max} とした．

以下の計算においては，建物各層の第 1，第 2 剛性：SK1，SK2 と層降伏せん断力 V_{jy} を不確定とした場合および最弱層（本計算例については 1 階）のみの SK1，SK2，V_{jy} を不確定とした場合について考慮した．なお，これら構造特性を変動させる際の変動係数はいずれも 5 %，サンプルは，あらかじめ発生総数 600 に指定して均等にサンプリングする Latin Hypercube Sampling 法で生成し，正規分布に従うと仮定した．

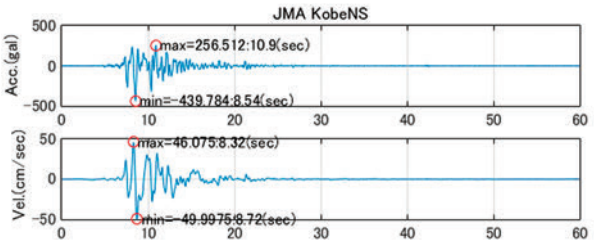


図-3　レベル 2 相当基準化入力地震動

4.2　解析結果

解析では，やや耐震性の劣る X 方向フレームについてのみ取り挙げ，先ず全層の構造特性が変動するとした場合について検討した．図-4 左上の（a）図は，変動パラメータが多数ある内の最弱層 1 階の第 2 剛性 SK2 を x 軸，同じく層降伏せん断力 V_{jy} を y 軸，最大応答塑性率 μ_{max} を z 軸にとり示したものである．他の 3 つ（b）～（d）図は，個々の構造特性と μ_{max} の関係を示したものであるが，これらの図から，μ_{max} の分布は面的ではなく膨らみを有していることが分かる．これは，（a）図で最弱層の x，y 座標値がほぼ同じであっても上層階の構造特性が異なる場合が多数あり，それに伴い異なる応答値 μ_{max} が生じたことに起因している．この様な分布形状の場合には，深層学習を実行しても高い精度の結果を期待出来ない．

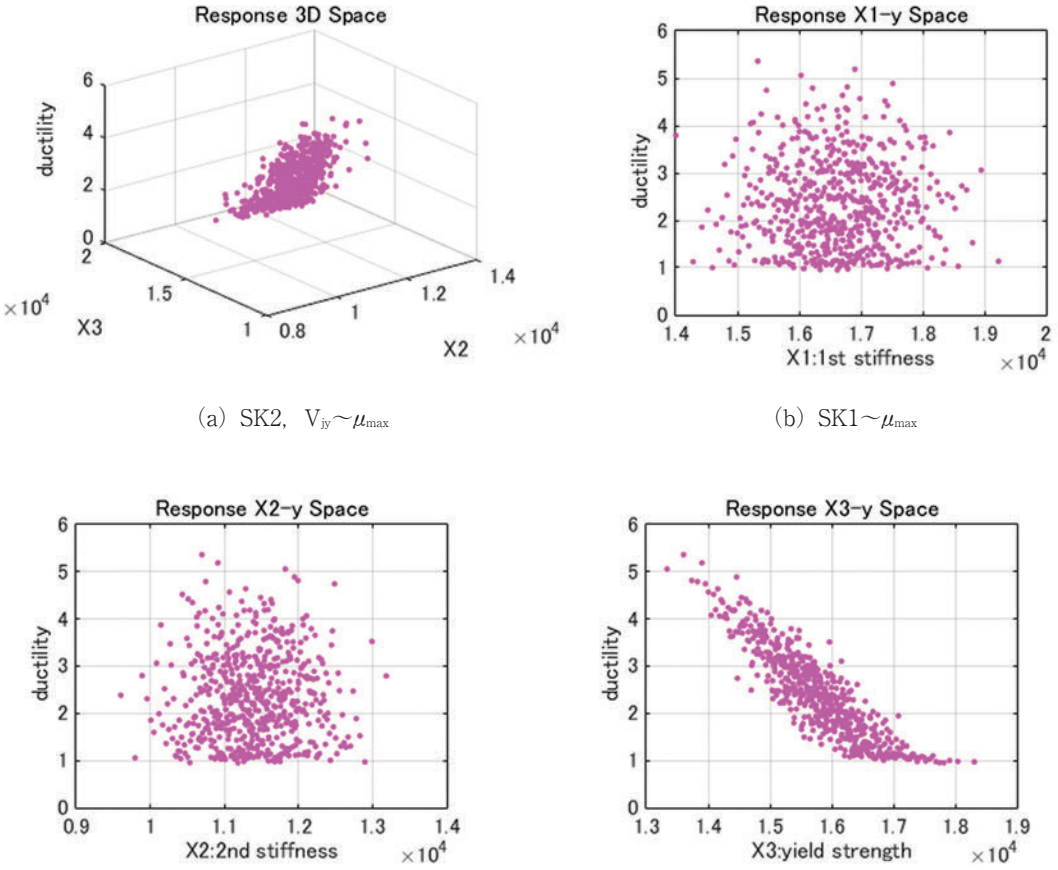
次に，最弱層 1 階のみ SK1，SK2，V_{jy} を変動すると仮定した結果について述べる．これは，最弱層にのみ損傷が集中して建物が崩壊に至る場合に該当する．なお，DNN を実行する際には，試行結果を踏まえた表-3 に示す前提条件で行った．

先ず，3 個の構造特性パラメータ（即ち，深層学習における特徴量）を k 分割交差検証の分割（fold）毎に表-3 に示す様に訓練とテストデータに分割後，scikit-learn の StandardScaler で標準化する TensorFlow のプログラムを記述，実行した．

図-5(a) は，x，y，z 軸に夫々第 2 剛性，層降伏せん断力，μ_{max} をとり，4 つの fold を平均した平

表-2　解析建物の構造階高，層重量および構造特性

階	11F	10F	9F	8F	7F	6F	5F	4F	3F	2F	1F
構造階高（cm）	295	295	295	295	295	295	295	295	295	295	490
層重量（kN）	4022	4176	4282	4282	4311	4330	4426	4446	4486	4550	4724
初期剛性（kN/cm）	7193	9609	10604	11296	12083	13175	14441	15372	16617	18916	27463
第 2 剛性（kN/cm）	3205	4119	4368	5010	6010	8755	9629	10634	11393	10287	13560
層降伏せん断力（kN）	4089	6237	8296	9891	11267	12862	14122	15025	15824	16389	16869



（a）SK2，V_{jy}～μ_{max}　　（b）SK1～μ_{max}

（c）SK2～μ_{max}　　（d）V_{jy}～μ_{max}

図-4　最弱層 1 階の構造特性～応答量関係（全層構造特性変動）

表-3　DNN 実行時の条件（試行結果考慮済）

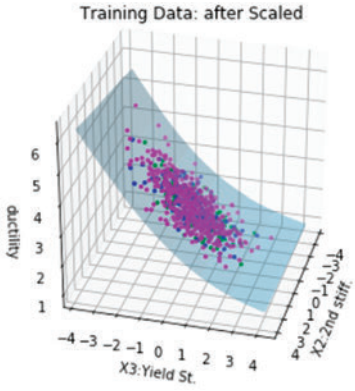
- 特徴量 3，観測データポイント数 600
  （＝訓練データ 450＋テストデータ 150）
  別に検証データ 100 個用意
- 隠れ層 2，ノード数 8
  ∴パラメータ総数＝(3*8＋8)＋(8*8＋8)＋(8＋1)＝113
- バッチサイズ：10，epoch 数：60
- 活性化関数：隠れ層：'softplus'，出力層：適用無し
- 損失関数：'mean_squared_error'
- 最適化関数：'adam'
- k 分割交差検証（k＝4）を採用

均回帰曲面（構造特性～応答量関係）と最終 fold の各データ（即ち，ピンクの訓練データ，緑のテストデータ，青の検証データ）を重ねたものである．(a) 図から概ね全てのデータが回帰曲面近傍にあることが分かる．なお，個別の fold の回帰曲面形状は，データが入れ替わるので，互いに微妙に変化する．

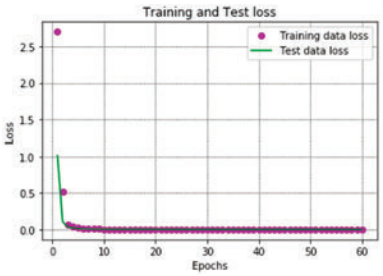
一方，(b)，(c) 図は，夫々訓練とテストデータの損失値（loss）の推移とテストデータの平均絶対誤差（MAE）の全 fold 平均値の推移である．(b) 図から損失値の推移は単調減少し，ロスの大小関係はテストロス＜訓練ロスであること，(c) 図から全 fold の MAE 平均値の推移が単調減少していることが分かる．これより，過剰適合は認められないと判断できる．なお最終 epoch＝60 の MAE 平均値は 0.09454 であった．この値は，回帰曲面に対する各点縦軸最大応答塑性率の平均的な差を表し，分類問題の精度に該当する．

図-6(a)，(b) は図-5(a) 図を Matlab で表示し直したものである．真横から見た (b) 図から，平均回帰曲面上に訓練，テストおよび検証データが概ね綺麗に乗っていることが分かる．

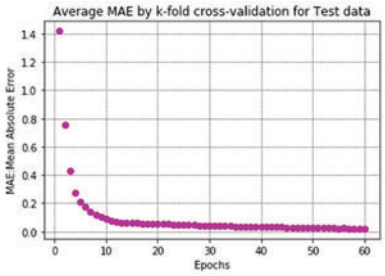
図-7 は，TensorBoard に拠る本例の計算グラフである．図中の文字が小さく細部の視認が困難であるが，そこには実装モデルの定義が示されている．以上より，地震時応答計算の一定割合を学習結果へ置き換えることは可能と推察される．



注）最終 fold：訓練データ–ピンク
テストデータ–緑，検証データ–青
（a）平均回帰曲面

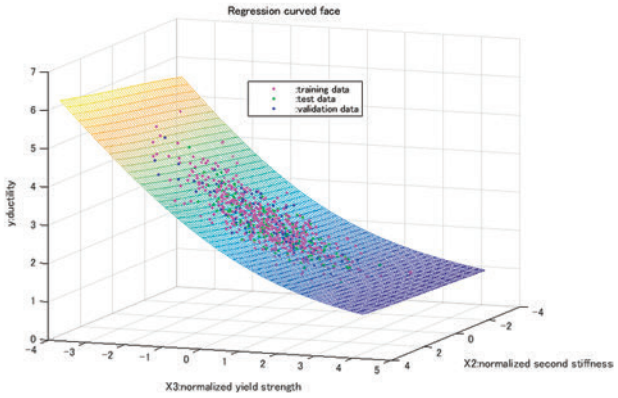


最終 fold
（b）訓練，テスト損失値の推移

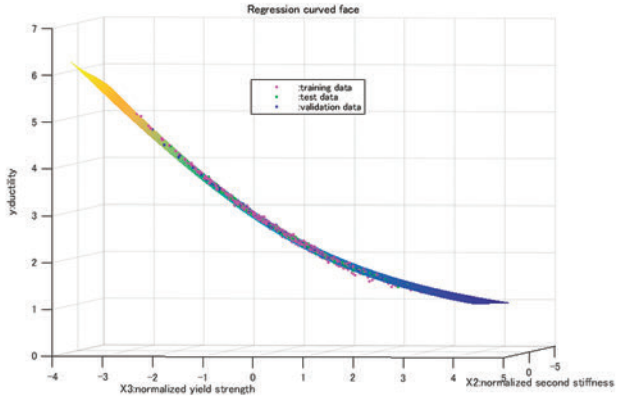


MAE：Mean Absolute Error
for test data
（c）全 fold の MAE 平均値

図–5　構造特性～応答量関係，epoch～loss，MAE 関係（最弱層のみ構造特性変動）

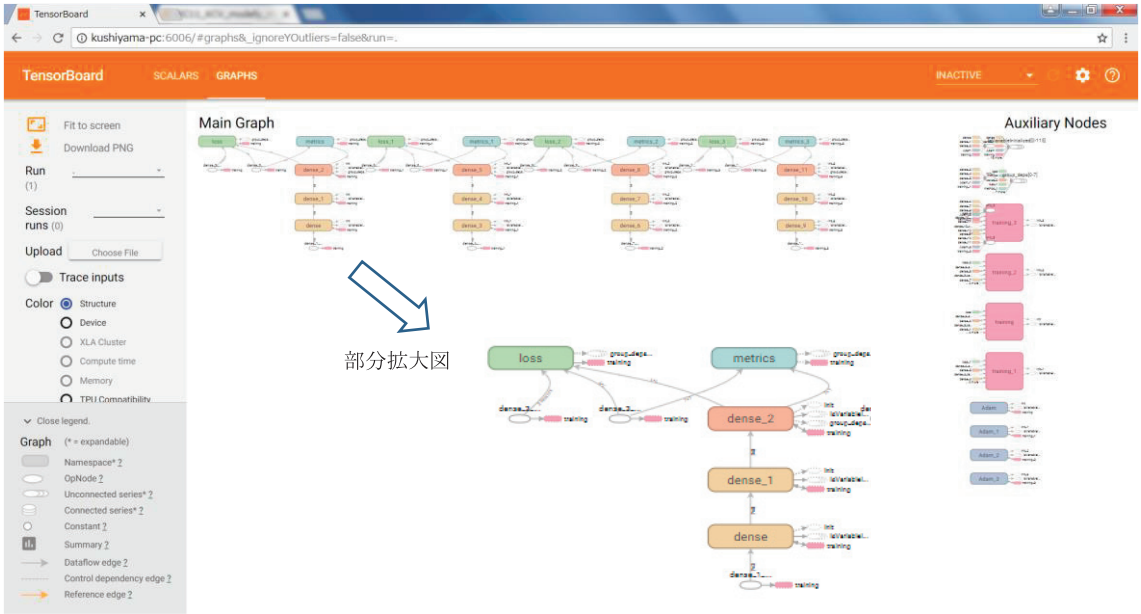


（a）Matlab による表示



注）ピンク：訓練データ，緑：テストデータ，青：検証データ
（b）視点を変えた Matlab による表示

図–6　構造特性～応答量関係（最弱層のみ構造特性変動）



図–7　TensorBoard による計算グラフの表示

5．結語

現在急速に進化しつつある AI（人工知能）の中核的な技術である Deep Learning（深層学習）の概要と本学工学部計算機実習室の機械学習環境について紹介した．大規模な問題を処理するには充分な設備とは云えないが，機械学習を専攻するデータサイエンス系学科の学生でなくとも scikit-learn，TensorFlow のサンプルコードを打ち込み気軽に体験できる環境が整備されている．サンプルコードも GitHub 上に沢山用意されており，また個人所有の PC にも上記同様のライブラリを容易にインストールできることを付記する．

本資料では回帰問題の事例として，Tensor Flow を用いて地震時応答計算の置き換え可能性を調べ，特徴量の数を減らした特定層集中崩壊型建物の応答計算に対して適用を見込めることを確認した．最終目標は，建物の定量的安全性評価の確率を得るための計算負荷低減である．

今後の課題としては，Matlab で作成した質点系の動的非線形応答計算プログラムの必要箇所で，Python3 系で記述された回帰問題処理コードを呼び出し，連結・実装することが挙げられる．

参考文献

1）John Haugeland, “Artificial Intelligence: The Very Idea, Cambridge”, MIT Press, 1985.

2）Vladimir N. Vapnik, “Statistical Learning Theory”, John Wiley & Sons, Inc., 1998.

3）François Chollet, “Deep Learning with Python”, Manning Publication, 2017.
（邦訳）株式会社クイープ　訳，“Python と Keras によるディープラーニング”，マイナビ，2018.

4）Andreas C. Müller and Sarah Guido, “Introduction to Machine Learning with Python”, O’REILLY, 2017.
（邦訳）中田秀基　訳，“Python ではじめる機械学習”，O’REILLY Japan, 2017.

5）Shigeru Kushiyama, “Seismic Damage Estimation of an Actual Reinforced Concrete Structure Using subset MCMC”, Open Journal of Civil Engineering, September 2013, Vol.3, No.3.

6）S. K. Au and J. L. Beck: “Subset Simulation and its Application to Seismic Risk Based on Dynamics Analysis”, Journal of Engineering Mechanics, 2003, pp. 901–917.