

Graph Neural Networkサロゲートモデルを用いた ジェットエンジブラケットの解析事例

清野 多美子* 木村 春里*

Analysis of Jet Engine Brackets Using Graph Neural Network Surrogate Models: A Case Study

Tamiko Seino* and Harusato Kimura*

汎用の構造解析ソフトである Advance/FrontSTR を用いたジェットエンジブラケットの有限要素解析結果を基に、グラフニューラルネットワーク（GNN）によるサロゲートモデルを構築した。学習データには SimJEB[1]データセットのメッシュ情報を活用し、変位およびミーゼス応力の予測を行った事例を紹介する。

<https://doi.org/10.69290/j.001230-vol33>

Keywords: 機械学習、GAT サロゲートモデル、構造解析、GAT

1. はじめに

近年の開発現場では、有限要素法（FEM）を用いた構造解析が設計判断の不可欠な要素となっているが、モデルの複雑化や大規模化に伴い、計算コスト（計算時間およびメモリ消費量）の増大が大きな課題となっている。また最適な形状を探索する設計過程においては、形状変更のたびに膨大な時間を費やして解析を繰り返す必要があり、これが設計開発期間を長期化する要因となっている。この課題を解決する手法として、深層学習等の機械学習を用いて、従来の数値解析を近似的に高速代行する「サロゲートモデル」の構築が注目を集めている。GNN サロゲートモデルを活用するメリットとして、設計者の形状変更に伴う手間を大幅に削減し、従来手法では数時間から数日を要していた解析結果の予測を、複数の形状案に対し瞬時に比較検討できるようになり、製品開発期間の短縮やコスト削減が期待される。

こうした背景を踏まえ、Advance/FrontSTR を順解析に活用した事例として、グラフニューラルネ

ットワークによるサロゲートモデルを作成し、応力分布や変形の予測を行う事例を作成した。事例作成に当たっては Qiita の記事[2][3] を参考とし、ジェットエンジンのブラケットのデータセットである SimJEB[1]に含まれる FEA 用のメッシュデータを活用した。

2. 機械学習モデル

2.1. グラフニューラルネットワーク

GNN（Graph Neural Network）は、グラフデータを対象としたニューラルネットワークであり、点（ノード）とそれをつなぐ線（エッジ）からなるデータを扱う。通常の DNN（Deep Neural Network）や CNN（Convolutional Neural Network）では、入力の特徴量を特徴量とみなし、それを層ごとに次の特徴量に変換するため、入力する配列の形状を揃えることが一般的である。RNN（Recurrent Neural Network）やその派生形では再帰的に値を入力するため、時系列データのような可変長配列を扱うことができるが、有限要素法のメッシュデータのような複雑な構造の配列を可変長で扱うことはできない。これらに対し GNN ではグラフの各ノードにそれぞれ特徴量を与え、各ノードの特徴量

*アドバンスソフト株式会社 第2事業部

2nd Computational Science and Engineering Group,
AdvanceSoft Corporation

を周囲のノードの特徴量を使って更新する。そのため、グラフニューラルネットワークでは構造の異なるグラフを単一のモデルで扱うことができる。これら各ネットワークの特徴の比較を表 1 に示す。

表 1 機械学習モデルの特徴比較

モデル	主な入力データ	特徴
DNN	数値配列（ベクトル）	全ての層が全結合 データの並び順に意味がない場合に有効。
CNN	画像、動画など	局所的な特徴（エッジなど）を抽出する 「畳み込み」が得意。位置関係を重視。
RNN	時系列や連続的なデータ	過去の情報を「状態」として保持し、 次に活かす。データの順序が重要。
GNN	グラフ構造	ノード（点）とエッジ（線）の関係性を学習。 不規則な形状のデータに対応。

2.2. 機械学習モデルの選定

グラフニューラルネットワーク（GNN）は、グラフデータを基本構造とする特徴量を扱うための手法で、現在主流である GNN の手法を高度に抽象化したものを Message passing フレームワークと呼ぶ。今回は、GNN において定番であるグラフ畳み込みネットワーク GCN (Graph convolutional network) と GCN にエッジ特徴量もサポートするアテンション機構を導入した GAT (Graph attention network) [4]を選定して機械学習を実施した。プログラム実装には Pytorch Geometric [5] を使用した。

各層でのノード特徴量は以下のように計算される。GCN を構成する GATConv 層では、ノード特徴量は式(1)で計算され、エッジ特徴量は扱わない。

$$\mathbf{x}_i^{(l+1)} = \mathbf{b} + \sum_{j \in N(i) \cup \{i\}} \frac{1}{\sqrt{\deg(i)}\sqrt{\deg(j)}} \mathbf{M}\mathbf{x}_j^{(l)} \quad (1)$$

$\mathbf{M}$: 学習可能な行列

$\mathbf{b}$: 学習可能なベクトル（バイアス）

$\deg(i)$: ノード $\mathbf{v}_i$ に隣接するノードの数+1

一方、GAT を構成する GATConv 層では、畳み込みの際にエッジ特徴量を考慮し、ノード特徴量は式(2)で計算される。

$$\mathbf{x}_i^{(l+1)} = \mathbf{b} + \sum_{j \in N(i) \cup \{i\}} \alpha_{ij} \mathbf{M}\mathbf{x}_j^{(l)} \quad (2)$$

ここで、

α_{ij}

$$= \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}_s^T \mathbf{M}\mathbf{x}_i + \mathbf{a}_t^T \mathbf{M}\mathbf{x}_j + \mathbf{a}_e^T \mathbf{M}_e \mathbf{e}_{ij})\right)}{\sum_{k \in N(i) \cup \{i\}} \exp\left(\text{LeakyReLU}(\mathbf{a}_s^T \mathbf{M}\mathbf{x}_i + \mathbf{a}_t^T \mathbf{M}\mathbf{x}_k + \mathbf{a}_e^T \mathbf{M}_e \mathbf{e}_{ik})\right)}$$

$$\text{LeakyReLU}(x) = \begin{cases} x & x \geq 0 \\ 0.2x & \text{otherwise} \end{cases}$$

$\mathbf{M}, \mathbf{M}_e$: 学習可能な行列

$\mathbf{a}_s, \mathbf{a}_t, \mathbf{a}_e$: 学習可能なベクトル

3. 教師データの作成

3.1. データセット

データセットは SimJEB[1]から作成した。SimJEB とは、ジェットエンジンのブラケットのデザインコンペティションである GE Jet Engine Bracket Challenge[6]で公開されている CAD データについて、メッシュを作成し、いくつかの载荷条件に対して有限要素解析を行った結果をデータセット化したものである。今回は SimJEB のメッシュファイルを Advance/FrontSTR 形式に変換し構造解析を実施することで新たなデータセットを作成した。データセット作成の流れを図 1 に、作成したメッシュの例を図 2 に示す。また作成したすべてのメッシュについて、その節点数の統計量は以下ようになっており、節点数の分布は図 3 のようになっている。

- ・ 節点数の平均値：80967.9
- ・ 節点数の標準偏差：40113.5
- ・ 節点数の最小値：18141
- ・ 節点数の最大値：219357

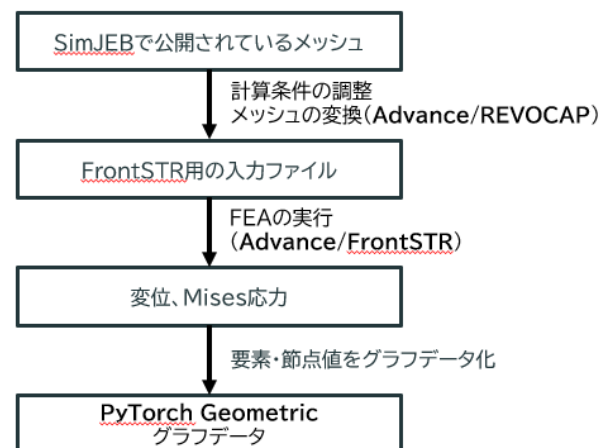


図 1 データセット作成の流れ

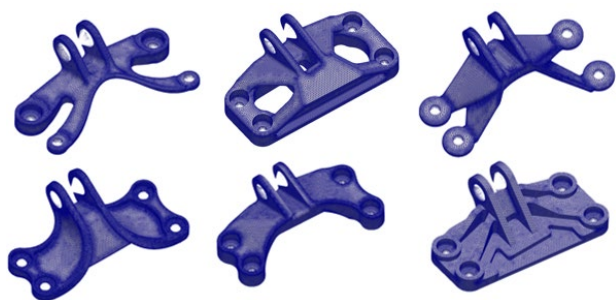


図 2 JEB の形状例

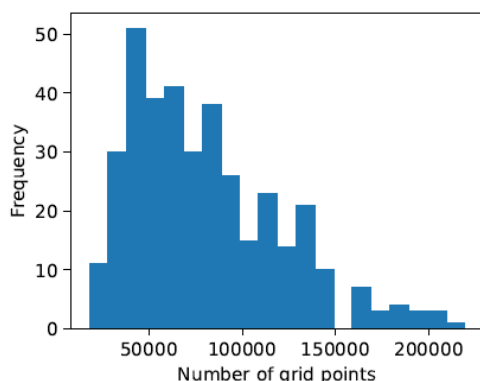


図 3 JEB モデルの総節点数の分布

3.2. 構造解析条件

SimJEB のメッシュは、四面体要素と 2 種類の剛体結合要素 (RBE2 および RBE3) で構成されており、全てのデータが同じ条件設定が可能なトポロジーを持つ。Advance/FrontSTR のメッシュへの変換では、四面体要素と剛体要素の代替としてトラス要素のヤング率を 100 倍にした材料特性を与えて使用した。

解析は線形静解析で、荷重条件は鉛直荷重または水平荷重を対象とし、単位荷重 (1 kN) の荷重とした。使用した境界・荷重条件を図 4 に示す。

ブラケット本体の材料物性は Ti-6Al-4V の値としてヤング率 113.8[GPa]、ポアソン比 0.342 を共通で使用した。

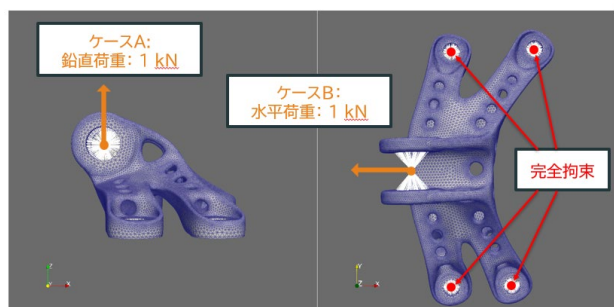


図 4 共通の境界・荷重条件

3.3. グラフデータへの変換

Advance/FrontSTR による構造解析実施後に、メッシュデータおよび解析結果である変位とミーゼス応力をグラフデータに変換した。グラフデータでは、節点をグラフのノードとした。グラフのエッジは同一の要素を構成する節点の間にあるものとして作成した。FEM メッシュをグラフデータに変換するイメージ図を図 5 に示す。

グラフデータに変換する際に、四面体要素とトラス要素を区別するためエッジ特徴量として二値変数 `is_ctetr4` を用意した。またノード特徴量として、座標(x, y, z) と荷重点であるか否かを表す二値変数である `is_input`、単点拘束をされているか否かを表す二値変数 `is_fixed` を与えた。予測対象の値としては、変位(`ux; uy; uz`) およびミーゼス応力 σ_{vm} (いずれも節点値) を与えた。これらのグラフデータを学習に用いるに先立ち、前処理として座標(x, y, z) については正規化を、予測対象の変位および Mises 応力に関しては標準化を行った。

全体のデータ数は、モデルサイズが大きくメモリ不足の原因となっていたケースを除き 370 データを使用し、そのうちの 80% に相当する 296 データを学習データセット、残りの 74 データを評価データセットとした。

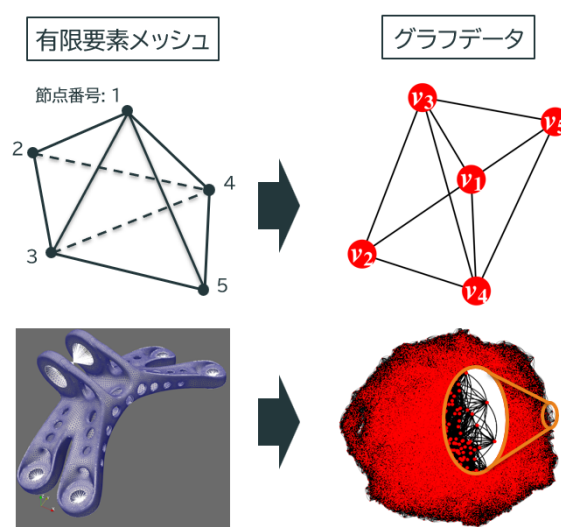


図 5 FEM メッシュからグラフデータへの変換

4. 学習と予測結果

4.1. 学習モデル

GNNモデルはグラフ畳み込み層と線形層が各3層ずつのモデル (Small) と 4層ずつのモデル (Medium) を用意し、サイズ、GNN 層後の活性化関数の有無、使用する GNN の種類、予測対象とする変数、载荷方向の5つの要素を変更して複数作成した。オプティマイザには Adam を用い、学習率は 1.0×10^{-3} 、MediumRelu のみ 1.0×10^{-4} とした。バッチサイズは 1 として 100–400 エポックの学習を行った。損失関数は平均自乗誤差 (MSE) を用いた。使用したネットワーク構造を表 2、表 3 に示す。

モデルサイズ : Small or Medium
 GNN 層後の活性化関数 : ReLU or なし
 GNN : GAT or GCN
 予測対象の変数 : (ux, uy, uz, σ_{um}) or (ux, uz)
 载荷方向 : 垂直 or 水平

表 2 使用したネットワーク構造 (Small)

層の種類別	入力サイズ	出力サイズ	活性化関数
Conv※	5	16	(ReLU)
Conv	16	32	(ReLU)
Conv	32	64	-
Dense	64	32	ReLU
Dense	32	16	ReLU
Dense	16	4(2)	-

※ConvはGATConvまたはGCNConv

表 3 使用したネットワーク構造 (Medium)

層の種類別	入力サイズ	出力サイズ	活性化関数
Conv	5	16	(ReLU)
Conv	16	32	(ReLU)
Conv	32	64	(ReLU)
Conv	64	128	-
Dense	128	64	ReLU
Dense	64	32	ReLU
Dense	32	16	ReLU
Dense	16	4(2)	-

4.2. 鉛直载荷ケース

4.2.1. 損失関数の推移

鉛直载荷のケースでモデルサイズと活性化関数についての検討を行った。図 6 に学習に伴う損失関数の推移を示す。Train が学習データセットを用いた場合で、Test が評価データセットを用いた場合である。いずれの場合でもロスが下がっており、過学習とはなっていないことが分かる。Small と Medium では僅差ながら Medium が優れるが、SmallRelu と Medium では SmallRelu が優れる結果となった。MediumRelu で学習率を 1.0×10^{-3} とすれば SmallRelu よりも性能が良いことが予想されるが、Medium は Small の倍以上の計算時間を要するため、SmallRelu を基準として用いることとした。目的変数を変位のみにする場合、MSE は小さくとも変位の推定値精度に顕著な変化はなかった。

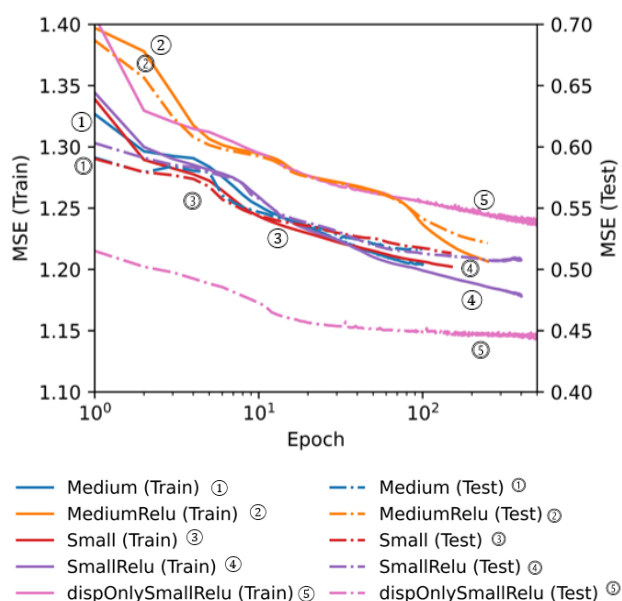


図 6 学習に伴う損失関数の推移 (垂直载荷)

4.2.2. 予測結果

鉛直载荷のケースのうち、GAT、SmallRelu を利用した予測結果の比較を行った例を示す。Advance/FrontSTR による解析結果を正解値とし、GNN による予測結果と比較し、予測対象の変数について予測値と正解値の差分を出力し可視化を行った。それぞれの値を可視化した図を図 7 に示す。

鉛直载荷の荷重負荷方向である変位の z 成分の比較においては、GNN による予測値は、傾向と値の双方で FEA による結果とおおむね整合した。変

位の x 成分の比較についても予測値と正解値は傾向と値の双方において数値解析による正解値とおおむね整合したが、予測値のほうの一部大きな値を予測している箇所も見られた。変位の y 成分の比較では予測値は正解値と部分的に整合しているが、一部において予測値と正解値で正負が合致しない現象が見られた。ただし、これは結果を標準化しているため出現しているが、そもそもの振幅が小さいためとも考えられる。ミーゼス応力の比較においては、大まかな応力集中位置の出現については予測可能であるが、全体的に差分の大きいところが現れる結果となった。特に FEA で応力集中箇所で予測値の値が小さくなる傾向が見られた。



図 7 正解値と予測値の比較例（鉛直载荷）
（上から u_x , u_y , u_z , ミーゼス応力）

4.3. 水平载荷ケース

4.3.1. 損失関数の推移

水平载荷のケースで GCN と GAT の比較を行った。学習データ (Train) ではそれほど大きな差はないものの、評価データ (Test) では GAT のほうが良い結果を示した。

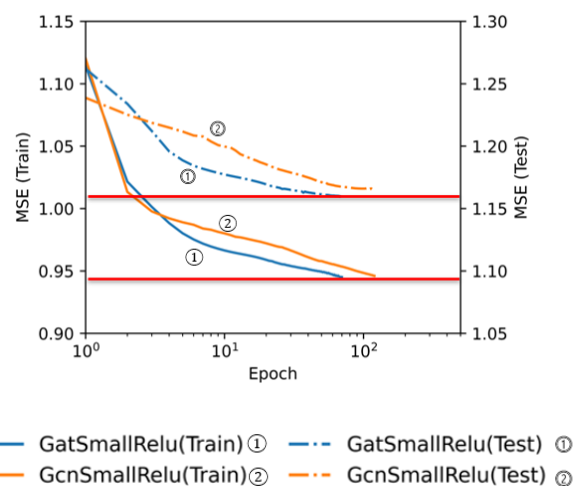


図 8 学習に伴う損失関数の推移（水平载荷）

4.3.2. 予測結果

水平载荷のケースのうち、GAT、SmallReLu を利用した予測結果の比較を行った例を示す。荷重負荷方向の変位の x 成分と、 z 成分の比較を実施したところ、全体的な変位分布の出現については正解値と予測値で同様の傾向を示すものの、荷重負荷位置付近では、差分が大きく出現する傾向が見られた。

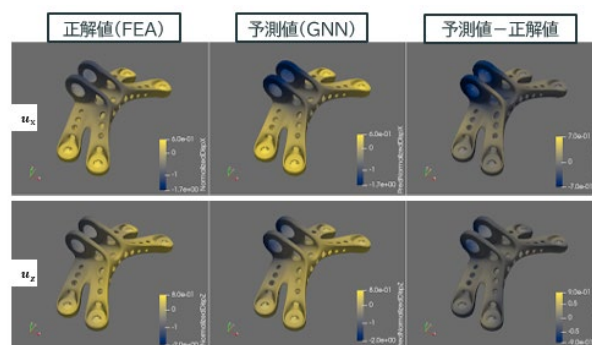


図 9 正解値と予測値の比較例（水平载荷）
（上から u_x , u_z ）

5. 実行時間

5.1. データセットの FEA の計算時間

解析実行は以下の条件で実施し、全体のデータセット数 370 に対し、計算は数時間で実行できた。ただしこれはモデル作成を含まない、最も計算負荷の小さい線形静解析のみの時間であり、実際の設計現場では、モデル作成に要する時間が課題となることが多いと考えられる。

- ・ 線形静解析
- ・ ソルバー：直接法
- ・ Windows WSL2 上で実行
- ・ 逐次計算

5.2. GNN 学習に要した計算時間

計算は Windows の WSL2 上で実施したが、グラフのサイズが大きく Medium では GPU のメモリでは容量が不足したため、CPU 上で学習を実施し、計算時間はおおよそ表 4 のようになった。

表 4 GNN 学習に要した計算時間

モデル	プロセッサ	計算時間 (min/epoch)
Medium、GAT	CPU	40
Small、GAT	CPU	20
Small、GAT	GPU	15
Small、GCN	GPU	10

6. まとめ

グラフニューラルネットワーク (GNN) のサロゲートモデルを用いたジェットエンジンブラケット(JEB)の予測解析の事例を作成した。作業にあたっては、SimJEB データセットからメッシュを変化し、Advance/FrontSTR による構造解析を実施し、メッシュと計算結果をグラフデータに変換することでデータセットを作成することができた。作成したデータセットを用いて、GATConv 層を含むようなネットワークで学習を行った。学習させたモデルは、変位や Mises 応力について、正解値と同様の傾向を示す結果を推定することができた。ただし十分な精度良い推論とするには、モデルの構成やノーマライズの方法を調整するなどの課題が考えられる。また学習においては、FEM のモデル規模が大きくなり、学習モデルの選定方によっては計算機に大きなメモリが必要となるなど、学習時の計算機のスペックに対する課題もあると考えられる。

参考文献

- [1] E. Whalen, A. Beyene, and C. Mueller. SimJEB: Simulated Jet Engine Bracket Dataset. Version

V4.2021. doi: 10.7910/DVN/XFUWJG.

- [2] S. Takaramoto. 3 次元物理シミュレーションのサロゲートモデルを作りたい. 2023. url: <https://qiita.com/rm0063vpedc15/items/70b5719353cbe4bf22b8> (閲覧 2025/07)
- [3] S. Takaramoto. 3 次元物理シミュレーションのサロゲートモデルを作りたい. 2024. url: <https://qiita.com/rm0063vpedc15/items/1679a5c6cd77c0a4ef40>. (閲覧 2025/07)
- [4] P. Veličković et al. Graph Attention Networks. 2018. url: <https://arxiv.org/abs/1710.10903>.
- [5] M. Fey and J. E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019.
- [6] K. Kiis et al. GE Jet Engine Bracket Challenge. 2013. url: <https://grabcad.com/challenges/gejet-engine-bracket-challenge>.

※ 技術情報誌アドバンスシミュレーションは、それぞれの文献タイトルの下に記載した DOI から、PDF ファイル (カラー版) がダウンロードできます。また、本雑誌に記載された文献は、発行後に、JDREAMIII (日本最大級の科学技術文献情報データベース) に登録されます。