Transfer Learning with Deep Tabular Models

Roman Levin, Valeriia Cherepanova, Avi Schwarzschild, Arpit Bansal, Bayan Bruss, Tom Goldstein, Andrew Gordon Wilson, Micah Goldblum

November 4, 2022
Introduction

- Gradient boosted decision trees (GBDT) are the traditionally dominant approach for tabular data

- However, recent tabular deep learning models demonstrate competitive performance and almost bridge the gap between DL and GBDT

- Accuracy aside, a major advantage of neural networks is that they are easily fine-tuned and learn reusable feature representations giving rise to transfer learning

- Transfer learning plays a central role in computer vision and NLP, but is underexplored in the tabular domain. One might wonder if representation learning is even useful for tabular data.

---

Introduction: Our Work

- We systematically study transfer learning with recent deep tabular models.
- We conduct experiments in a realistic medical diagnosis test bed with limited amounts of downstream data.
- We find that deep tabular models with transfer learning provide a definitive advantage over strong GBDT baselines, even those that also leverage upstream data.
- We compare supervised and self-supervised pre-training strategies and provide practical advice on transfer learning with tabular models.
- We propose a pseudo-feature method for cases where the upstream and downstream feature sets differ, addressing a common tabular-specific problem.
Medical data is often limited, especially for rare conditions. However, large related datasets – *upstream data* – with similar features (lab tests) could be available, e.g. for more common diseases.

MetaMIMIC\(^1\) repository
- Based on MIMIC-IV\(^2\) clinical database of ICU admissions
- 12 binary prediction tasks for different diagnoses – related tasks of varied similarity
- 34,925 patients, 172 features

\(^1\)https://github.com/ModelOriented/metaMIMIC
\(^2\)https://physionet.org/content/mimiciv/0.4/
### Experimental Setup: TL Pipeline, Upstream and Downstream Tasks

#### Upstream and downstream tasks
- simulate a scenario of diagnosing a rare disease
  - Reserve 11 targets for pretraining and 1 target for the downstream task
  - Limit the amount of downstream data to 4, 10, 20, 100, 200 samples
  - Obtain 12 upstream-downstream splits in total (60 transfer learning tasks)

---

#### Upstream Data

<table>
<thead>
<tr>
<th>Patient</th>
<th>Pulse</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>67</td>
<td>130</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>153</td>
</tr>
<tr>
<td>2000</td>
<td>78</td>
<td>165</td>
</tr>
</tbody>
</table>

#### Pre-training

- Multi-Label Classification

<table>
<thead>
<tr>
<th>Patient</th>
<th>Heart</th>
<th>.</th>
<th>Anemia</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>.</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Downstream Data

<table>
<thead>
<tr>
<th>Patient</th>
<th>Pulse</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>87</td>
<td>143</td>
</tr>
<tr>
<td>1</td>
<td>74</td>
<td>116</td>
</tr>
<tr>
<td>10</td>
<td>145</td>
<td>180</td>
</tr>
</tbody>
</table>

#### Fine-tuning

- New Target

<table>
<thead>
<tr>
<th>Patient</th>
<th>Purpura</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
Experimental Setup: Tabular Models and Transfer Learning Setups

- **Neural networks:**
  - FT-Transformer\(^1\)
  - Tabular ResNet\(^1\)
  - MLP\(^1\)

- **GBDT:**
  - CatBoost\(^2\)
  - XGBoost\(^3\)

---


Experimental Setup: Tabular Models and Transfer Learning Setups

- **Neural networks:**
  - FT-Transformer\(^1\)
  - Tabular ResNet\(^1\)
  - MLP\(^1\)

- **GBDT:**
  - CatBoost\(^2\)
  - XGBoost\(^3\)

- **Transfer learning setups:**
  - Linear head atop a frozen feature extractor
  - MLP head atop a frozen feature extractor
  - End-to-end fine-tuning with a linear head
  - End-to-end fine-tuning with an MLP head

- **Baselines:**
  - Neural model trained from scratch
  - GBDT

---


Experimental Setup: Tabular Models and Transfer Learning Setups

- **Neural networks:**
  - FT-Transformer\(^1\)
  - Tabular ResNet\(^1\)
  - MLP\(^1\)

- **GBDT:**
  - CatBoost\(^2\)
  - XGBoost\(^3\)

- **Transfer learning setups:**
  - Linear head atop a frozen feature extractor
  - MLP head atop a frozen feature extractor
  - End-to-end fine-tuning with a linear head
  - End-to-end fine-tuning with an MLP head

- **Baselines:**
  - Neural model trained from scratch
  - GBDT
    - Only on downstream data

---


Experimental Setup: Tabular Models and Transfer Learning Setups

- **Neural networks:**
  - FT-Transformer\(^1\)
  - Tabular ResNet\(^1\)
  - MLP\(^1\)

- **GBDT:**
  - CatBoost\(^2\)
  - XGBoost\(^3\)

- **Transfer learning setups:**
  - Linear head atop a frozen feature extractor
  - MLP head atop a frozen feature extractor
  - End-to-end fine-tuning with a linear head
  - End-to-end fine-tuning with an MLP head

- **Baselines:**
  - Neural model trained from scratch
  - GBDT
    - Only on downstream data
    - Leveraging upstream data through stacking

---
## Results: Transfer Learning with Deep Tabular Models vs GBDT

The results of 30000 experiments, aggregated as average rank across all 12 downstream tasks at each data level:

<table>
<thead>
<tr>
<th>Num samples</th>
<th>FT-Transformer</th>
<th>ResNet</th>
<th>MLP</th>
<th>Catboost</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>11.1</td>
<td>15.2</td>
<td>11.2</td>
<td>9.8</td>
<td>11.3</td>
</tr>
<tr>
<td>100</td>
<td>11.9</td>
<td>13.1</td>
<td>12.2</td>
<td>10.2</td>
<td>9.2</td>
</tr>
<tr>
<td>20</td>
<td>11.6</td>
<td>13.2</td>
<td>13.5</td>
<td>13.9</td>
<td>13.4</td>
</tr>
<tr>
<td>10</td>
<td>11.8</td>
<td>13.2</td>
<td>15.7</td>
<td>13.4</td>
<td>12.2</td>
</tr>
<tr>
<td>4</td>
<td>10.9</td>
<td>13.9</td>
<td>14.7</td>
<td>11.5</td>
<td>15.2</td>
</tr>
</tbody>
</table>

### Takeaways:
- Deep tabular models with transfer learning outperform strong GBDT baselines at all data levels (note the color pattern).
- Representation learning with deep tabular models is more powerful than leveraging knowledge transfer through stacking.
- Simpler models such as MLP with transfer learning are very competitive in extremely low data regimes. However, more complex architectures like FT-Transformer offer consistent performance gains over GBDT across all data levels.
- On specific tasks, transfer learning with deep tabular models offers up to 5-7% accuracy improvement.
Self-Supervised Learning (SSL) vs Supervised Pre-training

- SSL - another way to use the upstream data (without labels)
- SSL learns the intrinsic structure of the data in an unsupervised way
- In NLP and vision, SSL pre-training produces more transferable features than supervised pre-training
- We use recent tabular SSL methods to investigate this in tabular domain
  - MLM pre-training – masks a random feature for each sample and predicts it from the other features
  - Contrastive pre-training – forms positive pairs using data augmentation, maps positive pairs close and negative pairs far in the feature space
Self-Supervised Learning (SSL) vs Supervised Pre-training

- SSL - another way to use the upstream data (without labels)
- SSL learns the intrinsic structure of the data in an unsupervised way
- In NLP and vision, SSL pre-training produces more transferable features than supervised pre-training
- We use recent tabular SSL methods to investigate this in tabular domain
  - MLM pre-training – masks a random feature for each sample and predicts it from the other features
  - Contrastive pre-training – forms positive pairs using data augmentation, maps positive pairs close and negative pairs far in the feature space

Takeaways:
- With a linear head, SSL works better than training from scratch
- However, supervised pre-training leads to more transferable features than SSL
Pseudo-Features: Aligning Upstream and Downstream Feature Sets

- What if the upstream and downstream features are similar, but misaligned? E.g. an additional lab test?
- We propose a pseudo-feature method to address this
Pseudo-Features: Aligning Upstream and Downstream Feature Sets

- What if the upstream and downstream features are similar, but misaligned? E.g. an additional lab test?
- We propose a pseudo-feature method to address this

Suppose the upstream data \((X_u, Y_u)\) is missing a feature \(x_{\text{new}}\) present in \((X_d, Y_d)\)

1. Pre-train \(f: X_u \rightarrow Y_u\) on \((X_u, Y_u)\) without \(x_{\text{new}}\)
2. Fine-tune \(f\) to predict \(x_{\text{new}}\) and get \(\hat{f}: X_d \setminus \{x_{\text{new}}\} \rightarrow x_{\text{new}}\)
3. Use \(\hat{f}\) to get \(\hat{x}_{\text{new}} = \hat{f}(X_u)\) and \((X_u \cup \{\hat{x}_{\text{new}}\}, Y_u)\)
4. Pre-train a feature extractor on \((X_u \cup \{\hat{x}_{\text{new}}\}, Y_u)\) and fine-tune on \((X_d, Y_d)\) with \(x_{\text{new}}\)
Pseudo-Features: Results

- We compare imputing the missing feature with discarding the feature from upstream or downstream data.

- Pseudo-feature method offers appreciable performance boosts over discarding the missing feature.
Does Tabular Transfer Learning Work Beyond Medical Data?

- Yeast functional genomics data
- Emotions in music data
We demonstrated that representation learning in tabular data is useful and that transfer learned deep tabular models definitively outperform strong GBDT baselines with stacking.

We showed that supervised pre-training produces more transferable features than SSL methods.

We presented a pseudo-feature method to enable effective transfer learning in cases where the upstream and downstream feature sets differ.

We hope that this work serves as a guide for practitioners.
Thank you!