Recent Advances in Language Model Fine-tuning

Author: SEBASTIAN RUDER
Overview

• Fine-tuning a pretrained language model (e.g., BERT, mBERT, XLM-R) has become a new norm for training a model on an NLP downstream task.

Source: https://ruder.io/recent-advances-lm-fine-tuning/
# Overview

- Methods discussed in this presentation:

<table>
<thead>
<tr>
<th>Category</th>
<th>Methods</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Fine-tuning</td>
<td>Domain/task/language adaptive fine-tuning</td>
<td>Specialise to target domain</td>
</tr>
<tr>
<td>Behavioural Fine-tuning</td>
<td>Intermediate-task training, self-supervised, frame as MLM</td>
<td>Specialise to target task</td>
</tr>
<tr>
<td>Parameter-efficient Fine-tuning</td>
<td>Adapters, sparse parameter permutations, pruning</td>
<td>Reduce space of fine-tuned models</td>
</tr>
<tr>
<td>Text-to-text Fine-tuning</td>
<td>Frame as text-to-text, prompt engineering, controllable NLG</td>
<td>Effectively use large autoregressive pre-trained LMs</td>
</tr>
<tr>
<td>Mitigating Fine-tuning</td>
<td>Stop runs early, use a small lr, regularisation, avoid random init</td>
<td>Reduce variance of fine-tuning runs</td>
</tr>
</tbody>
</table>

Source: [https://ruder.io/recent-advances-lm-fine-tuning/](https://ruder.io/recent-advances-lm-fine-tuning/)
Adaptive fine-tuning

• Pretrained language models can get in trouble with downstream data that is very different from the pretraining data in terms of .

• Adaptive fine-tuning is a way to bridge such a shift in distribution by fine-tuning the model on relevant unlabeled data:
  - Domain-adaptive: Unlabeled data is close to the training data of the downstream task.
  - Task-adaptive: Unlabeled data is the label-removed version of the downstream training data.

Source: https://ruder.io/recent-advances-lm-fine-tuning/
Behavioural fine-tuning

- Behavioural fine-tuning focuses on learning behaviours that are useful for the target task by fine-tuning the model on labeled data of the relevant tasks.
- E.g.: This paper (https://arxiv.org/abs/2101.11038) pre-finetunes the model on 50 labelled datasets (classification, summarization, common sense, MRC) in a massively multi-task setting and obtains significant improvement for Natural Language Inference, Question Answering, …

Source: https://ruder.io/recent-advances-lm-fine-tuning/
Parameter-efficient fine-tuning

- Using adapters which are small networks injected between transformer layers. During fine-tuning, original weights of the pretrained language model are fixed while adapter and task-specific weights are updated.

- Removing some last few layers of the pretrained language models and performing the fine-tuning with the remaining ones might still give similar performance.

Source: MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer
Text-to-text fine-tuning

- Casting the downstream task to a generative extraction task.
Text-to-text fine-tuning

- Casting the downstream task to a text generation task.

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"sts1 sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."
Mitigating fine-tuning instabilities

• Performance for fine-tuning on a small downstream dataset can vary drastically between different runs.
• Previous work shows that weight initialization of the output layer and the order of the training data contribute to variation in performance.
• To avoid instabilities, avoid randomly initialized output layers. We can use behavioural or text-to-text fine-tuning.