Identifying beneficial task relations for multi-task learning in deep neural networks

Author: Joachim Bingel, Anders Sogaard

Presenter: Litian Ma

Background

- **Multi-task learning** (MTL) in deep neural networks for NLP has recently received increasing interest due to some compelling benefits
- It has potential to efficiently **regularize models** and to **reduce** the need for **labeled data.**
- The main driver has been empirical results pushing **state of the art** in various tasks.
- In NLP, multi-task learning typically involves very **heterogeneous** tasks.

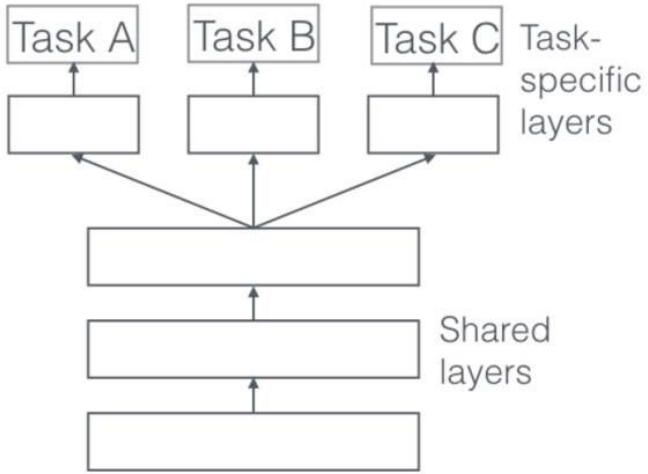
However ...

- While great improvements have been reported, results are also often **mixed**.
- **Theoretical guarantees** no longer apply to the overall performance.
- Little is known about the conditions under which MTL leads to gains in NLP.
- Want to answer the question:

What task relations guarantee gains or make gains likely in NLP?

Multi-task Learning -- Hard Parameter Sharing

- Extremely popular approach to multi-task learning.
- Basic idea:
  - Different tasks **share some of the hidden layers**, such that these learn a joint representation for multiple tasks.
  - Is considered as **regularizing** target model by doing model interpolation with auxiliary models in a dynamic fashion.



MTL Setup

- Multi-task learning architecture: **Sequence labeling with recurrent neural networks**
- With a **bi-directional LSTM** as a single hidden layer of 100 dimensions that is shared across all tasks.
- Input ot the hidden layer: 100-dimensional word vectors pre-trained by GloVe embeddings.
- Generates predictions from the bi-LSTM through task-specific dense projections.
- The model is **symmetric** in the sense that it does not distinguish between main and auxiliary tasks.

MTL Training Step

- A **training step** consists of:
  - Uniformly drawing a training task
  - Sampling a random batch of 32 examples from the task's training data.
- Each training step works on exactly one task, and optimizes the **task-specific projection** and the **shared parameters** using Adadelta.
- Hyper-parameters are fixed across single-task and multi-task settings.
  - Making our results only applicable to the scenario where one wants to know whether MTL works in the current parameter setting.

Ten NLP Tasks

- CCG Tagging (**CCG**)
- Chunking (**CHU**)
- Sentence Compression (**COM**)
- Semantic frames (**FNT**)
- POS tagging (**POS**)
- Hyperlink Prediction (**HYP**)
- Keyphrase Detection (**KEY**)
- MWE Detection (**MWE**)
- Super-sense tagging (**SEM**)
- Super-sense Tagging (**STR**)

Experiment Setting

- Train single-task bi-LSTMs for each of the ten tasks.
- Trained **25000** batches.

- One multi-task model for each of the pairs between the tasks, yielding 90 directed pairs of the form. $\langle \mathcal{T}_{main}, \{\mathcal{T}_{main}, \mathcal{T}_{aux}\}\rangle$
- Trained **50000** batches to account for the uniform drawing of the two tasks at every iteration.

Relative Gains and Losses

- 40 out of 90 cases show improvements
- **Chunking** and **high-level semantic tagging** generally contribute most to other tasks, while hyperlinks do not significantly improve any other task.
- **Multiword** and **hyperlink detection** seem to profit most from several auxiliary tasks.
- **Symbiotic relationships** are formed
  - e.g., by POS and CCG-tagging, or MWE and compression.

Figure 1: Relative gains and losses (in percent) over main task micro-averaged F_1 when incorporating auxiliary tasks (columns) compared to single-task models for the main tasks (rows).

Predict gains from MTL

- Dataset-inherent features + learning curve feature.
- **Learning curve feature**:
  - Gradients of the loss curve at 10, 20, 30, 50, and 70 percent of 25000 batches.
  - Steepness of the Fitted log-curve (parameter a and c): $L(i) = a \cdot \ln(c \cdot i + d) + b$
- Each of 90 data points is described by 42 features.
  - 14 features each task.
  - main, auxiliary, and **main/auxiliary ratios**.
- **Binarize** the experiment results as labels.
- Use **logistic regression** to predict benefits.

Data features	
Size	Number of training sentences.
# Labels	The number of labels.
Tokens/types	Type/token ratio in training data.
OOV rate	Percentage of training words not in GloVe vectors.
Label Entropy	Entropy of the label distribution.
Frobenius norm	$\|\|X\|\|_F = [\sum_{i,j} X_{i,j}^2]^{1/2}$, where $X_{i,j}$ is the frequency of term j in sentence i.
JSD	Jensen-Shannon Divergence between train and test bags-of-words.
Learning curve features	
Curve gradients	See text.
Fitted log-curve	See text.

Table 2: Task features

Experiment Results

- A strong signal in **meta-learning features.**
- The features derived from the single task inductions are the most important.
    - Only using data-inherent features, F1 score is worse than the majority baseline.

	Acc.	F_1 (gain)
Majority baseline	0.555	0.615
All features	0.749	0.669
Best, data features only	0.665	0.542
Best combination	0.785	0.713

Table 3: Mean performance across 100 runs of 5-fold CV logistic regression.

Experiment Analysis

Feature	Task	Coefficient
Curve grad. (30%)	Main	-1.566
Curve grad. (20%)	Main	-1.164
Curve param. c	Main	1.007
# Labels	Main	0.828
Label Entropy	Aux	0.798
Curve grad. (30%)	Aux	0.791
Curve grad. (50%)	Main	0.781
OOV rate	Main	0.697
OOV rate	Main/Aux	0.678
Curve grad. (20%)	Aux	0.575
Fr. norm	Main	-0.516
# Labels	Main/Aux	0.504
Curve grad. (50%)	Aux	-0.099
Curve grad. (50%)	Main/Aux	0.076
OOV rate	Aux	0.061
Curve grad. (30%)	Main/Aux	-0.060
Size	Main	-0.032
Curve param. a	Main	0.027
Curve grad. (10%)	Main/Aux	0.023
JSD	Main	0.019
JSD	Main/Aux	-0.015
Curve grad. (10%)	Main	$6 \cdot 10^{-2}$
Size	Main/Aux	$-6 \cdot 10^{-3}$
Curve grad. (70%)	Main/Aux	$-4 \cdot 10^{-4}$

Experiment Analysis

- Features describing the **learning curves** for the main and auxiliary tasks are the **best** predictors of MTL gains.
- The **ratios** of the learning curve features seem **less** predictive, and the gradients around **20-30%** seem most important.
- If the main tasks have flattening learning curves (small negative gradients) in the 20-30% percentile, but the auxiliary task curves are still relatively steep, MTL is more likely to work.
    - Can help tasks that get stuck early in **local minima**.

Key Findings

- MTL gains are **predictable** from dataset characteristics and features extracted from the single-task Inductions
- The most predictive features relate to the single-task learning curves, suggesting that MTL, when successful, often helps target **tasks out of local minima**.
- **Label entropy** in the auxiliary task was also a good predictor; but there was little evidence that dataset balance is a reliable predictor, unlike what previous work has suggested.

Thanks!