CS11-747 Neural Networks for NLP

Multi-task, Multi-lingual Learning

Graham Neubig

Site
https://phontron.com/class(nn4nlp2017/)
Remember, Neural Nets are Feature Extractors!

• Create a vector representation of sentences or words for use in downstream tasks

  this is an example →

  this is an example →

• In many cases, the same representation can be used in multiple tasks (e.g. word embeddings)
Types of Learning

- **Multi-task learning** is a general term for training on multiple tasks.

- **Transfer learning** is a type of multi-task learning where we only really care about one of the tasks.

- **Domain adaptation** is a type of transfer learning, where the output is the same, but we want to handle different topics or genres, etc.
When to Multi-task?
Plethora of Tasks in NLP

• In NLP, there are a plethora of tasks, each requiring different varieties of data

  • **Only text:** e.g. language modeling

  • **Naturally occurring data:** e.g. machine translation

  • **Hand-labeled data:** e.g. most analysis tasks

• And each in many languages, many domains!
**Rule of Thumb 1: Multitask to Increase Data**

- Perform multi-tasking when one of your two tasks has many fewer data

- **General domain $\rightarrow$ specific domain**
  (e.g. web text $\rightarrow$ medical text)

- **High-resourced language $\rightarrow$ low-resourced language**
  (e.g. English $\rightarrow$ Telugu)

- **Plain text $\rightarrow$ labeled text**
  (e.g. LM $\rightarrow$ parser)
Rule of Thumb 2:

- Perform multi-tasking when your tasks are related

- e.g. predicting eye gaze and summarization (Klerke et al. 2016)
Methods for Multi-task Learning
Standard Multi-task Learning

- Train representations to do well on multiple tasks at once

This is an example

In general, as simple as randomly choosing minibatch from one of multiple tasks

Many many examples, starting with Collobert and Weston (2011)
Pre-training

• First train on one task, then train on another

this is an example - Encoder → this is an example - Encoder

Initialize

• Widely used in word embeddings (Turian et al. 2010)

• Also pre-training sentence representations (Dai et al. 2015)
Examples of Pre-training Encoders

- Common to pre-train encoders for downstream tasks, common to use:
  - **Language models** (Dai and Le 2015)
  - **Translation models** (McCann et al. 2017)
  - **Bidirectional language models** (Peters et al. 2017)
Regularization for Pre-training  
(e.g. Barone et al. 2017)

• Pre-training relies on the fact that we won’t move too far from the initialized values

• We need some form of regularization to ensure this

  • **Early stopping**: implicit regularization — stop when the model starts to overfit

  • **Explicit regularization**: L2 on difference from initial parameters

\[
\theta_{adapt} = \theta_{pre} + \theta_{diff} \\
\ell(\theta_{adapt}) = \sum_{(X,Y) \in \mathcal{X}, \mathcal{Y}} -\log P(Y \mid X; \theta_{adapt}) + ||\theta_{diff}||
\]

• **Dropout**: Also implicit regularization, works pretty well
Selective Parameter Adaptation

- Sometimes it is better to adapt only some of the parameters
  - e.g. in cross-lingual transfer for neural MT, Zoph et al. (2016) examine best parameters to adapt

<table>
<thead>
<tr>
<th>Setting</th>
<th>Dev BLEU</th>
<th>Dev PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>No retraining</td>
<td>0.0</td>
<td>112.6</td>
</tr>
<tr>
<td>Retrain source embeddings</td>
<td>7.7</td>
<td>24.7</td>
</tr>
<tr>
<td>+ source RNN</td>
<td>11.8</td>
<td>17.0</td>
</tr>
<tr>
<td>+ target RNN</td>
<td>14.2</td>
<td>14.5</td>
</tr>
<tr>
<td>+ target attention</td>
<td>15.0</td>
<td>13.9</td>
</tr>
<tr>
<td>+ target input embeddings</td>
<td>14.7</td>
<td>13.8</td>
</tr>
<tr>
<td>+ target output embeddings</td>
<td>13.7</td>
<td>14.4</td>
</tr>
</tbody>
</table>
Soft Parameter Tying

• It is also possible to share parameters loosely between various tasks

• Parameters are regularized to be closer, but not tied in a hard fashion (e.g. Duong et al. 2015)
Domain Adaptation
Domain Adaptation

- Basically one task, but incoming data could be from very different distributions

  news text

  medical text

  spoken language

  Encoder

  Translation

- Often have big grab-bag of all domains, and want to tailor to a specific domain

- Two settings: **supervised and unsupervised**
Supervised/Unsupervised Adaptation

• **Supervised adaptation:** have data in target domain
  - Simple pre-training on all data, tailoring to domain-specific data (Luong et al. 2015)
  - Learning domain-specific networks/features

• **Unsupervised adaptations:** no data in target domain
  - Matching distributions over features
Supervised Domain Adaptation through Feature Augmentation

• e.g. Train general-domain and domain-specific feature extractors, then sum their results (Kim et al. 2016)

• Append a domain tag to input (Chu et al. 2016)

<news> news text
<med> medical text
Unsupervised Learning through Feature Matching

- Adapt the latter layers of the network to match labeled and unlabeled data using multi-kernel mean maximum discrepancy (Long et al. 2015)

- Similarly, adversarrial nets (Ganin et al. 2016)
Multi-lingual Models
Multilingual Inputs

• Often as simple as training a single (large) encoder
• Optionally: use adversarial objective to help ensure that information is shared (Chen et al. 2016)

• Quite successful in a number of tasks
Multilingual Structured Prediction/Multilingual Outputs

- Things are harder when predicting a sequence of actions (parsing) or words (MT) in different languages
- One simple method: add embedding of the expected output to your model (e.g. Tsvetkov et al. 2016)
Multi-lingual Sequence-to-sequence Models

• It is possible to translate into several languages by adding a tag about the target language (Johnson et al. 2016, Ha et al. 2016)
  
  <fr> this is an example → ceci est un exemple
  <ja> this is an example → これは例です

• Potential to allow for “zero-shot” learning: train on fr↔en and ja↔en, and use on fr↔ja
  
  • Works, but not as effective as translating fr→en→ja
Teacher-student Networks for Multilingual Adaptation (Chen et al. 2017)

- Use a better pivoted model to “teach” a worse zero-shot model to translate well
Multi-task Models
Types of Multi-tasking

- Most common: train on plain text or translated text, use information for syntactic analysis task
- Also, training on multiple annotation tasks
- Other examples:
  - Training with multiple annotation standards
  - Training w/ different layers for different tasks
Multiple Annotation Standards

- For analysis tasks, it is possible to have different annotation standards.

- Solution: train models that adjust to annotation standards for tasks such as semantic parsing (Peng et al. 2017), word segmentation.

- We can even adapt to individual annotators! (Guan et al. 2017)
Different Layers for Different Tasks (Hashimoto et al. 2017)

- Depending on the complexity of the task we might need deeper layers
- Choose the layers to use based on the level of semantics required
Questions?