The standard methodology in machine learning

- learning one task at a time
- Large problems are broken into small, reasonably independent subproblems that are learned separately and then recombined
Motivation

- A net with a 1000x1000 pixel input retina is unlikely to learn to recognize complex objects in real-world scenes
- But what if we simultaneously train a net to recognize object outlines, shapes, edges, regions, subregions, textures, reflections, highlights, shadows, text, orientation, size, distance, etc.,
According to Wikipedia, multi-task learning is an approach to learn a problem together with other related problems at the same time, using a shared representation.
Relatedness

-Learning tasks with the aim of mutual benefit

-Assumption: All tasks are related

-Example 1: Different classification tasks

Spam filtering - Everybody has a slightly different distribution over spam or not-spam emails but there is a common aspect across users.

Idea: Learning together can be a good regularizer
Relatedness

Example 2: Image Categorization
Relatedness

Other examples:
- Web Page Categorization [chen et al ICML 09]
  Page categories can be related
- Movie Ranking [Yu et. al NIPS 06]
  similar tastes between users
Learning simultaneously

- Inductions of multiple task are performed simultaneously to capture intrinsic relatedness

- The main question : How to learn ?
Learning Methods

- Joint feature learning: the simplest idea
- Mean-regularized MTL: Penalizes the deviation of each task from the mean
- Shared parameter gaussian process
- Low rank regularized
- Alternating structural optimization
- ... [will discuss later]
Shared Representation

- Shared Hidden node in a Neural Network:
The simplest one can be a neural network shared hidden units among tasks.

- Shared Parameter:
  Like Gaussian process

- Regularization-based:
  Mean, Joint feature table, …
Shared Representation

Sharing Hidden Nodes in Neural Network
- A set of hidden units are shared among multiple tasks. (goal: improving generalization)
Shared Representation

-Joint Feature Learning
creating a common set of features
MTL with Joint Feature learning

- Using Group Sparsity

$l_1/l_2$-norm regularization
An Application In NLP

- A unified architecture for Natural Language Processing deep neural net with multi task learning (by Collobert and Watson)
- Tasks: POS, NER, Chunking, Semantic Roles, ...
- Relatedness: Are these tasks related?
- Shared Representation: NN layers
- Training: Joint training using weight sharing
- Tasks:
  1. POS (Part of Speech Tagging): labeling each word with a unique tag that shows its tactic roles, ex. adverb, noun,…
  2. Chunking: labeling segments of a sentence with syntactic constituents
3. Named Entity Recognition: Labeling atomic elements in the sentence into categories such that “Location”, “Person”
4. Semantic Role Labeling: Giving a semantic role to a syntactic constituent of a sentence. Example: [John]Arg0 [ate]Rel [the apple]Arg1
An Application In NLP - Regular approaches

Rich Hand-Designed Features → Shallow Classification (Algorithm like SVM) → Model for a certain task

Selecting features by empirical process (trial and error)
Task-based algorithm selection
An Application In NLP 1 - new approach

- Deep Neural Network
- Feature extraction in several layers using back propagation
An Application In NLP 2 - new approach

- First Layer: features for each word
- Second Layer: features for the input sentence (sequenced based)
- Following layers: Classical NN layers
An Application In NLP 3-

Look up tables layer

- for word $i$ in the Dictionary considering a d-dimensional space

$L_{Tw}(i) = W_i$

- $W$ : parameters to be learnt

- For solving variable sentence length: Considering fixed size window size around each word.
An Application In NLP

- Time Delay Neural Network: perform linear operation over the input words.
- Max Layer: Captures the most relevant features over the sentence.
An Application In NLP 5 - Output and Algorithm

- Using softmax for joint learning
- Algorithm (training in the stochastic manner):
  1. select the next task
  2. select a random training example for this task
  3. Use gradient for updating NN
  4. go to step 1
<table>
<thead>
<tr>
<th></th>
<th>wsz=15</th>
<th>wsz=50</th>
<th>wsz=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRL</td>
<td>16.54</td>
<td>17.33</td>
<td>18.40</td>
</tr>
<tr>
<td>SRL + POS</td>
<td>15.99</td>
<td>16.57</td>
<td>16.53</td>
</tr>
<tr>
<td>SRL + Chunking</td>
<td>16.42</td>
<td>16.39</td>
<td>16.48</td>
</tr>
<tr>
<td>SRL + NER</td>
<td>16.67</td>
<td>17.29</td>
<td>17.21</td>
</tr>
<tr>
<td>SRL + Synonyms</td>
<td>15.46</td>
<td>15.17</td>
<td>15.17</td>
</tr>
<tr>
<td>SRL + Language model</td>
<td>14.42</td>
<td>14.30</td>
<td>14.46</td>
</tr>
<tr>
<td>SRL + POS + Chunking</td>
<td>16.46</td>
<td>15.95</td>
<td>16.41</td>
</tr>
<tr>
<td>SRL + POS + NER</td>
<td>16.45</td>
<td>16.89</td>
<td>16.29</td>
</tr>
<tr>
<td>SRL + POS + Chunking + NER</td>
<td>16.33</td>
<td>16.36</td>
<td>16.27</td>
</tr>
<tr>
<td>SRL + POS + Chunking + NER + Synonyms</td>
<td>15.71</td>
<td>14.76</td>
<td>15.48</td>
</tr>
<tr>
<td>SRL + POS + Chunking + NER + Language model</td>
<td>14.63</td>
<td>14.44</td>
<td>14.50</td>
</tr>
</tbody>
</table>
What if tasks are not totally related

- If the tasks have a group structures
  => Clustered Multi-task learning

  e.g. tasks in the yellow group are predictions of heart related diseases and in the blue group are brain related diseases.

  more information : Bakker and Heskes JMLR 2003
What if tasks are not totally related

- If the tasks have a tree structures
  => Multi-task Learning with Tree Structures

more information:
Tree-Guided Group Lasso
(Kim and Xing 2010 ICML)
What if tasks are not totally related

- If the tasks have a graph structures

  => Multi-task Learning with Graph Structures

more information:
Graph-guided Fused Lasso (Chen et. al. UAI11)
Connection to other ML topics

Learning Methods

- **Transfer Learning**
  - Define source & target domains
  - Learn on the source domain
  - Generalize on the target domain

- **Multi-task Learning**
  - Model the task relatedness
  - Learn all tasks simultaneously
  - Tasks may have different data/features

- **Multi-label Learning**
  - Model the label relatedness
  - Learn all labels simultaneously
  - Labels share the same data/features

- **Multi-class Learning**
  - Learn the classes independently
  - All classes are exclusive
Software Packages

MALSAR: Multi-tAsk Learning via StructurAl Regularization
-Implemented by Biodesign Institute of Arizona State University
Main References


Thanks for you attention

Any Question ????