THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Week 13 Tutorial: Multi-task Learning

Anjin Liu
anjin.liu@sydney.edu.au

Tutorial Contents

THE UNIVERSITY OF SYDNEY

- Review (20min):
  - Lecture 12: Multi-task Learning
- Tutorial exercise & QA (40min):
- Announcement: Unit Study Survey
- This week lecture: Week 13 Review

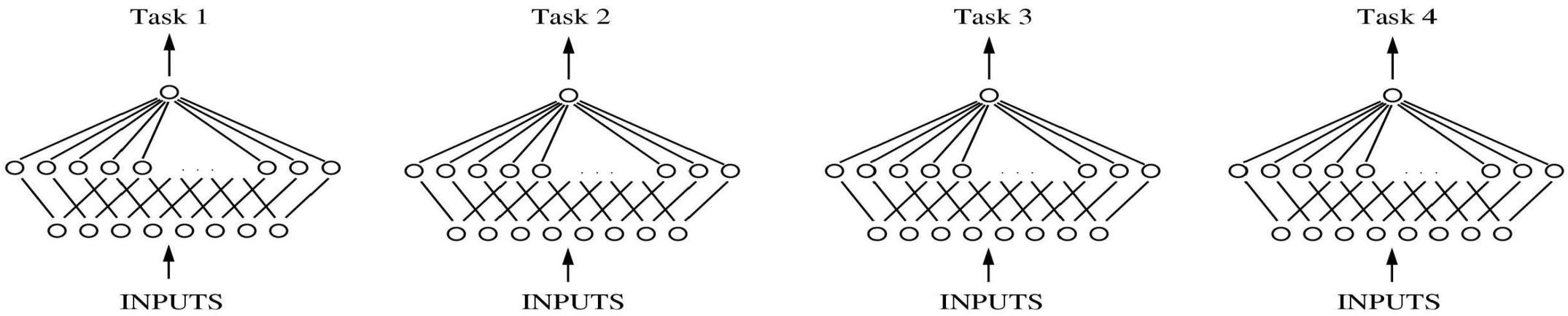
Key points

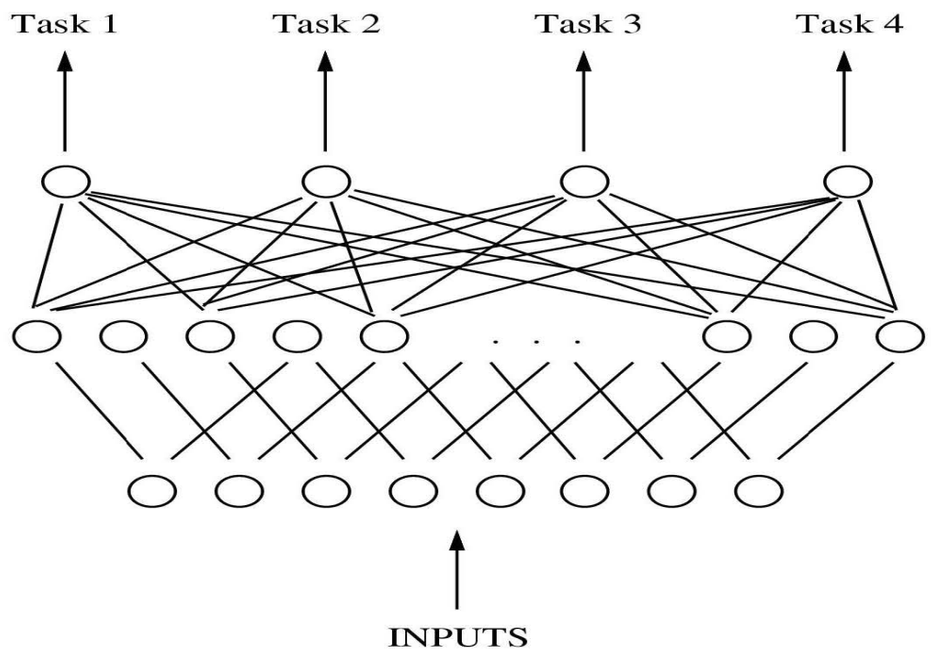
- Multi-task Learning
- Parameter-based MTL models
- Feature-based MTL models

THE UNIVERSITY OF SYDNEY

Multi-Task Learning

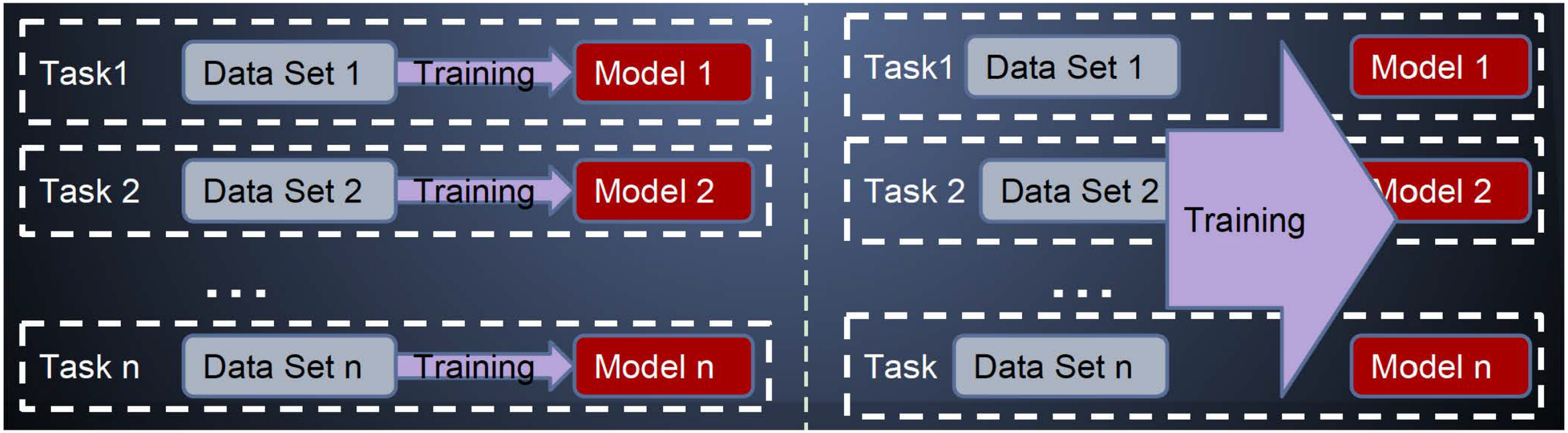
Motivation Examples: STL vs MTL





Concepts and General View

Multi-task Learning is an approach to learn multiple related problems at the same time, by exploiting the relatedness between problems.



Relatedness among Tasks

Learning tasks with the aim of mutual benefit.

Assumption : Tasks are related.

Example: Spam filtering - Everybody has a slightly different distribution over spam or non-spam emails, but there is a common aspect across users.

When tasks are independent to each other, multi-task learning will have no advantage to single task learning.

Big Data Interpretation

However, in some applications, large training examples are hard to collect, such as medical image analysis.

For this data insufficient problem, Multi-Task Learning (MTL) is a good solution when there are multiple related tasks each of which has limited training samples.

Problem Setup

Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^m$ where all the tasks or a subset of them are related, multi-task learning aims to help improve the learning of a model for $\mathcal{T}_i$ by using the knowledge contained in all or some of the m tasks.

The task $\mathcal{T}_i$ is accompanied by a training set $\mathcal{D}_i = \{\mathbf{x}_j^i, y_j^i\}_{j=1}^{n_i}$.

Our task is to learn hypotheses for $\{\mathcal{T}_i\}_{i=1}^m$.

MTL with Shared Knowledge

Multi-task Learning exploits the relatedness among tasks.
There are some shared knowledge across tasks.

Three questions in MTL with shared knowledge:
When to Share? What to Share? How to Share?

When to Share?

When there are relatednesses among tasks.

What to Share?

$$R_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

We consider two cases: share parameters and features.

How to Share?

Parameter-based MTL and feature-Based MTL models.

MTL models

Consider the linear hypothesis function, i.e., $h(x) = w^\top x$.

We have m different but related tasks, i.e., $\{\mathcal{T}_i\}_{i=1}^m$.

Denote w^i as the hypothesis for the i^{th} task, $i = 1, \ldots, m$.

The following empirical risk minimisation algorithm learns multiple tasks simultaneously, is it superior to learning the tasks individually?

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i).$$

MTL models

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, w^i).$$

VS

$$\min_{w^i} \frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, w^i), i = 1, \ldots, m.$$

Which group will have a better performance?

Parameter-based MTL models

We assume the multiple tasks are related by their parameters that

$$w^i = w_0 + \Delta w^i$$

where $i = 1, \ldots, m$.

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i).$$

VS

$$\min_{w^i} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i), i = 1, \ldots, m.$$

Which group will have a better performance?

Parameter-based MTL models

Can we improve the following MTL model?

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i).$$

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i) + \lambda \|\Delta W\|_F^2.$$

The latter model is better because it pushes the multi-task learning algorithm to have stronger relatedness.

Parameter-based MTL models

Given a matrix M, the **rank** of a matrix is the maximum number of linearly independent columns.

A rank 2 matrix:

$$S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Low-rank based MTL model (I):

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i) + \lambda \, \mathrm{rank}(W).$$

Parameter-based MTL models

Low-rank based MTL model (II):

Specifically, we assume that

$$w^i = u^i + \Theta^\top v^i,$$

where $i = 1, \ldots, m$, and $\Theta \in \mathbb{R}^{h \times d}$ is the shared low-rank subspace by multiple tasks. Then we have

$$W = U + \Theta^\top V.$$

Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." Journal of Machine Learning Research 6.Nov (2005): 1817-1853.

Parameter-based MTL models

Low-rank based MTL model (II):

$$\min_{U,V,\Theta} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, u^i + \Theta^\top v^i) + \lambda\|U\|_F^2,$$

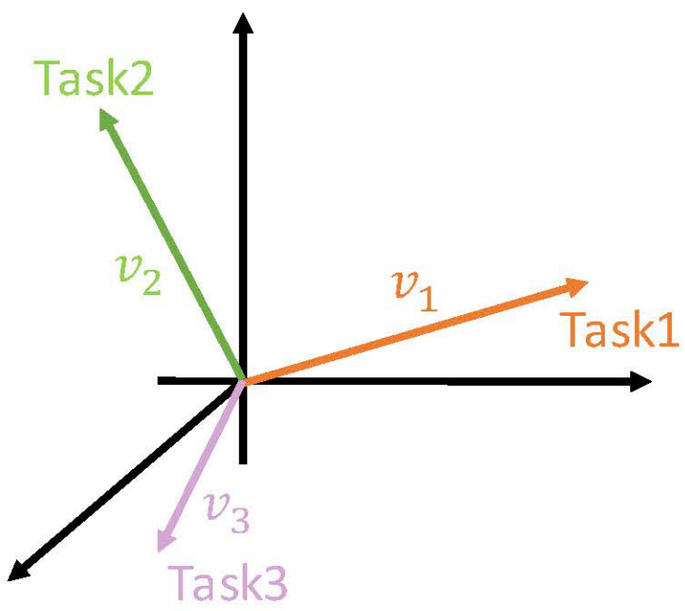
$$s.t.\ \Theta^\top\Theta = I.$$

Note that the orthogonal constraint makes the subspace non-redundant.

Feature-based MTL models

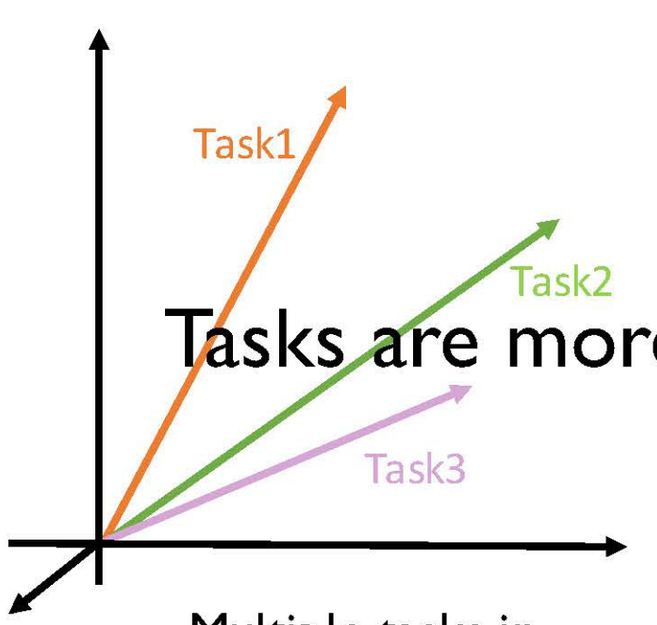
Note that the hypotheses are learned from training example.

By using $\mathcal{D}_i = \{x_j^i, y_j^i\}_{j=1}^{n_i}$ we have



Multiple tasks in original feature space

Can we map the features such that, $\mathcal{D}_i = \{P^\top x_j^i, y_j^i\}_{j=1}^{n_i}$



Multiple tasks in new feature space

Feature-based MTL models

Feature-based MTL model (I):

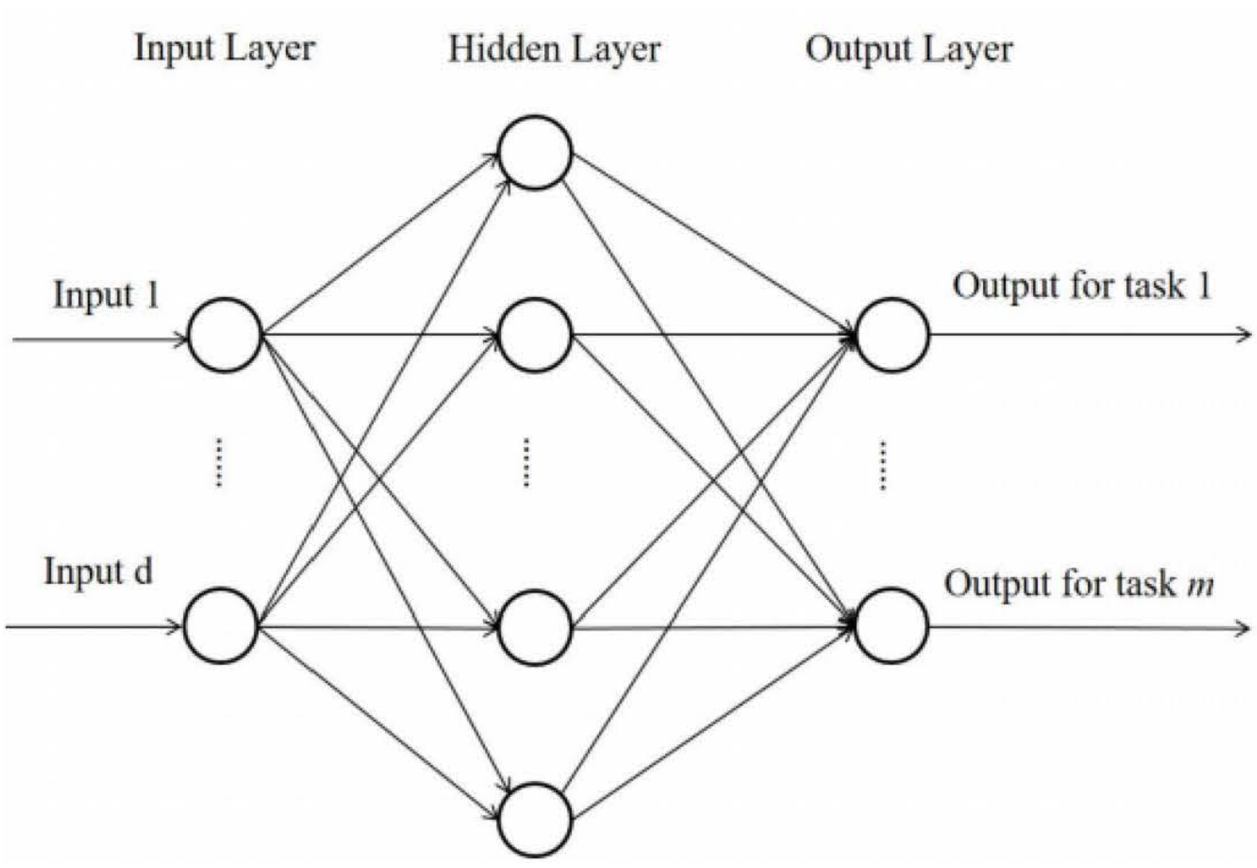
$$\min_{W,P} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(P^\top x_j^i, y_j^i, w^i) + \lambda \mathrm{rank}(W),$$

$$s.t.\ PP^\top = I.$$

Note that P is a projection matrix.

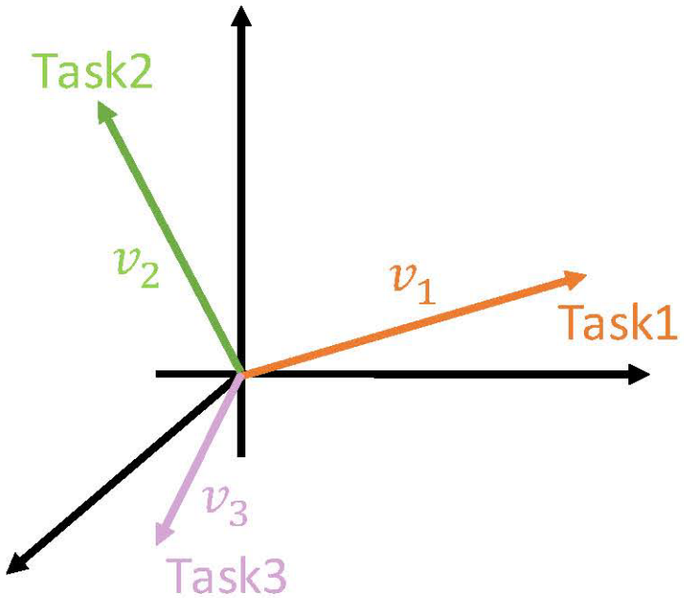
Feature-based MTL models

Feature-based MTL model (II):

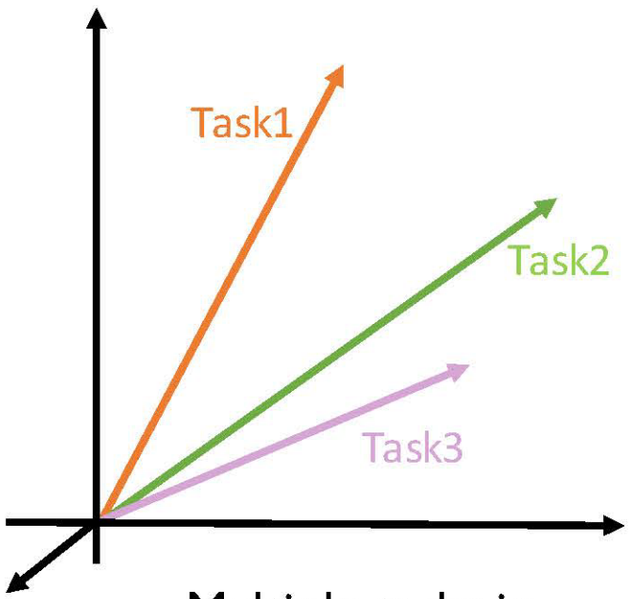


Shared Hidden nodes in a Neural Network. Note that neural network can be regarded as feature extractors.

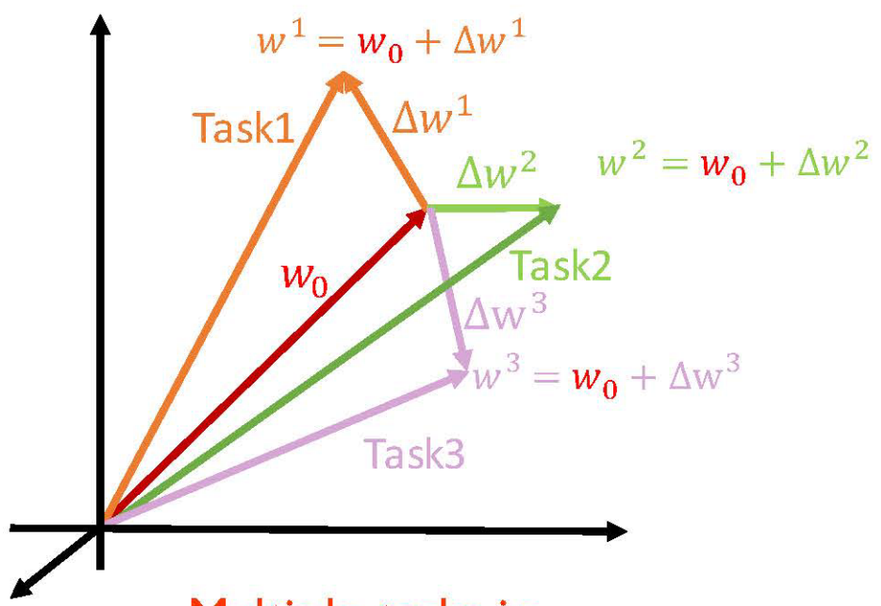
Feature- and Parameter-based MTL models



Multiple tasks in original feature space



Multiple tasks in new feature space



Multiple tasks in new feature space

Feature- and Parameter-based MTL models

$$\min_{w_0,\Delta W,P} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(P^\top x_j^i, y_j^i, w_0+\Delta w^i)+\lambda\|\Delta W\|_F^2,$$

$$s.t.\ PP^\top = I.$$

The above model learns the feature projection map P and the commonly shared parameter w_0 to enhance the relatedness among tasks.

Li, Ya, Xinmei Tian, Tongliang Liu, and Dacheng Tao. "Multi-Task Model and Feature Joint Learning." In IJCAI, pp. 3643-3649. 2015.

Why multi-task works?

Compared with single task learning, will all the tasks' performances be improved?

Is it possible that some task will be harmful to boost the performance? like a black sheep.

Relationship to Transfer Learning

In MTL, there is no distinction among different tasks and the objective is to improve the performance of all the tasks. However, in transfer learning which is to improve the performance of a target task with the help of source tasks, the target task plays a more important role than source tasks.

More about MTL

Online MTL, distributed MTL: handling large data sets.

A Survey on Mutli-Task learning: https://arxiv.org/pdf/1707.08114.pdf

An Overview of Multi-Task Learning in Deep Neural Networks: http://ruder.io/multi-task/

Matlab codes for many variants of MTL: http://jiayuzhou.github.io/MALSAR/

Key points

- Multi-task Learning
- Parameter-based MTL models
- Feature-based MTL models