Multi-Task Learning (MTL) in Deep Neural Networks

Dr. Selva Kumar S(SCOPE)

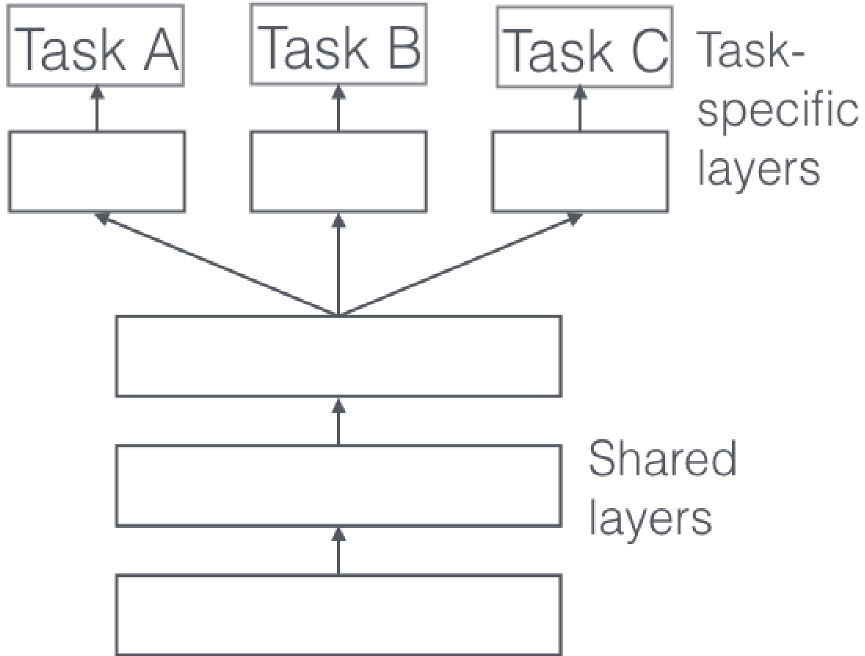
Multi-task Learning (MTL)

- Multi-task learning (MTL) is a field of machine learning in which models using data from multiple tasks are trained at the same time.
- This is done using shared representations to uncover the common ideas among a group of tasks that are connected.
- These shared representations help to overcome the well-known drawbacks of deep learning by increasing data efficiency.
- The goal of multi-task learning, as well as the allied fields of meta-learning, transfer learning, and continuous learning, should be the development of systems to facilitate this process. This process is critical to humans' ability to learn quickly and with a limited number of instances.

MTL Methods For Deep Learning

- Hard parameter sharing

Hard parameter sharing is the most commonly used approach to MTL in neural networks and goes back to. It is generally applied by sharing the hidden layers between all tasks, while keeping several task-specific output layers.

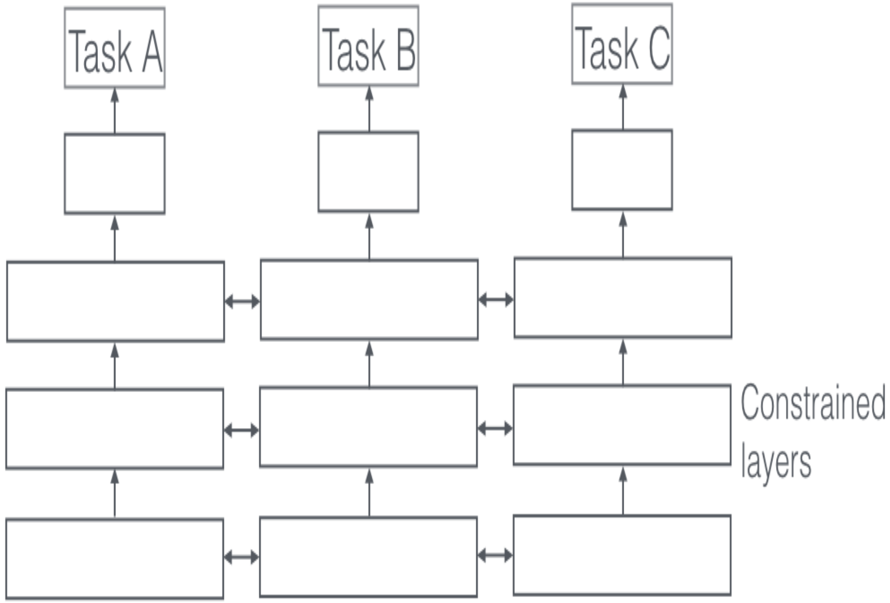


- Hard parameter sharing greatly reduces the risk of overfitting.

MTL Methods For Deep Learning

- Soft parameter sharing

In soft parameter sharing on the other hand, each task has its own model with its own parameters. The distance between the parameters of the model is then regularized in order to encourage the parameters to be similar.



- The constraints used for soft parameter sharing in deep neural networks have been greatly inspired by regularization techniques for MTL that have been developed for another model.

Why does MTL work

- Even though an inductive bias obtained through multi-task learning seems intuitively plausible, in order to understand MTL better, we need to look at the mechanisms that underlie it.

  - Implicit data augmentation
  - Attention focusing
  - Eavesdropping
  - Representation bias
  - Regularization

Why does MTL work

- Even though an inductive bias obtained through multi-task learning seems intuitively plausible, in order to understand MTL better, we need to look at the mechanisms that underlie it.

  - Implicit data augmentation
  - Attention focusing
  - Eavesdropping
  - Representation bias
  - Regularization

Why does MTL work

- Even though an inductive bias obtained through multi-task learning seems intuitively plausible, in order to understand MTL better, we need to look at the mechanisms that underlie it.

MTL mechanisms

- Implicit data augmentation
- Attention focusing
- Eavesdropping
- Representation bias
- Regularization

MTL Mechanisms

Implicit Data Augmentation

- MTL effectively increases the sample size that we are using for training our model.
- As all tasks are at least somewhat noisy, when training a model on some task A, our aim is to learn a good representation for task A that ideally ignores the data-dependent noise and generalizes well.
- As different tasks have different noise patterns, a model that learns two tasks simultaneously is able to learn a more general representation.
- Learning just task A bears the risk of overfitting to task A, while learning A and B jointly enables the model to obtain a better representation F through averaging the noise patterns.

MTL Mechanisms

Attention focusing

- If a task is very noisy or data is limited and high-dimensional, it can be difficult for a model to differentiate between relevant and irrelevant features.
- MTL can help the model focus its attention on those features that actually matter as other tasks will provide additional evidence for the relevance or irrelevance of those features.

Eavesdropping

- Some features G are easy to learn for some task B, while being difficult to learn for another task A.
- This might either be because A interacts with the features in a more complex way or because other features are impeding the model's ability to learn G. Through MTL, we can allow the model to *eavesdrop*, i.e. learn G through task B. The easiest way to do this is through *hints* , i.e. directly training the model to predict the most important features.

Applications of MTL

Representation bias

- MTL biases the model to prefer representations that other tasks also prefer. This will also help the model to generalize to new tasks in the future as a hypothesis space that performs well for a sufficiently large number of training tasks will also perform well for learning novel tasks as long as they are from the same environment.

Regularization

- Finally, MTL acts as a regularizer by introducing an inductive bias. As such, it reduces the risk of overfitting as well as the Rademacher complexity of the model, i.e. its ability to fit random noise.

Applications of MTL

- Machine learning in self-driving cars
- Predictions for the Stock Market
- Detection and recognition of objects