Multi-task Learning

Sargur N. Srihari
srihari@buffalo.edu

Regularization Strategies

1. Parameter Norm Penalties
2. Norm Penalties as Constrained Optimization
3. Regularization and Under-constrained Problems
4. Data Set Augmentation
5. Noise Robustness
6. Semi-supervised learning
7. Multi-task learning
8. Early Stopping
6. Parameter tying and parameter sharing
7. Sparse representations
8. Bagging and other ensemble methods
9. Dropout
10. Adversarial training
11. Tangent methods

Sharing parameters over tasks

- Multi-task learning is a way to improve generalization by pooling the examples out of several tasks
  - Examples can be seen as providing soft constraints on the parameters
- In the same way that additional training examples put more pressure on the parameters of the model towards values that generalize well

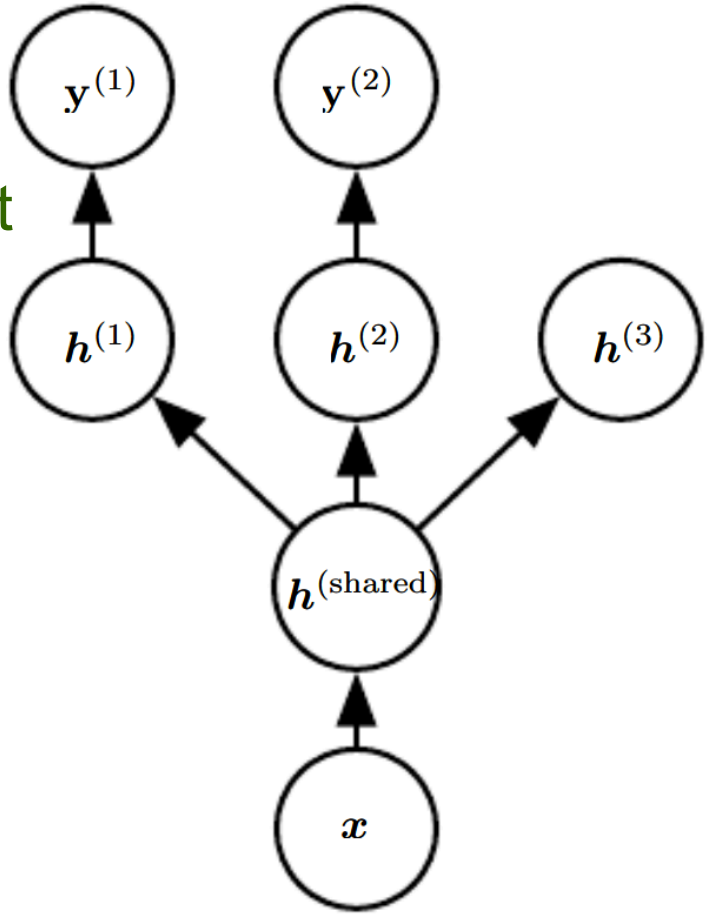
Common form of multitask learning

- Different supervised tasks, predicting $\mathrm{y}^{(i)}$ given $\mathbf{x}$, share the same input $\mathbf{x}$, as well as some intermediate representation $\boldsymbol{h}^{(\mathbf{shared})}$ capturing a common pool of factors

Common multi-task situation

- Common input but different target random variables
  - Lower layers (whether feedforward or includes a generative component with downward arrows) can be shared across such tasks.
    - Task-specific parameters $h^{(1)}$, $h^{(2)}$ can be learned on top of those yielding a shared representation $h^{(\text{shared})}$

Common pool of factors explain variations of Input x while each task is associated with a Subset of these factors



  - In the unsupervised learning context some of the top level factors are associated with none of the output tasks $h^{(3)}$

These are factors that explain some of the input variations but not relevant for predicting $h^{(1)}$, $h^{(2)}$

Model can be divided into two parts

1. Task specific parameters
   - Which only benefit from the examples of their task to achieve good generalization
     - These are the upper layers of the neural network
2. Generic parameters
   - Shared across all tasks
     - Which benefit from the pooled data of all tasks
     - These are the lower levels of the neural network

Benefits of multi-tasking

- Improved generalization and generalization error bounds
  - achieved due to shared parameters
    - For which statistical strength can be greatly improved
      - In proportion to the increased no. of examples for he shared parameters compared to the scenario of single-task models
- From the point of view of deep learning, the underlying prior belief is the following:
  - Among the factors that explain the variations observed in the data associated with different tasks, some are shared across two or more tasks