

factors. The model can generally be divided into two kinds of parts and associated parameters:

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 in figure 7.2.
2. Generic parameters, shared across all the tasks (which benefit from the pooled data of all the tasks). These are the lower layers of the neural network in figure 7.2.

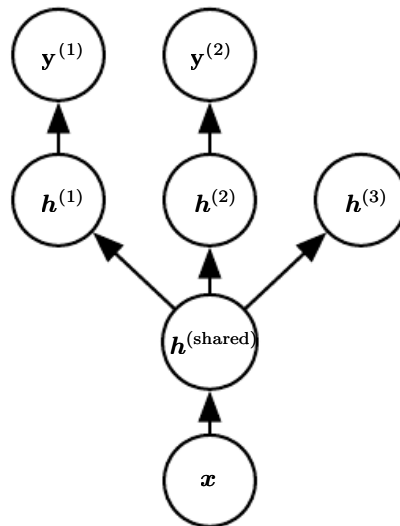


Figure 7.2: Multi-task learning can be cast in several ways in deep learning frameworks and this figure illustrates the common situation where the tasks share a common input but involve different target random variables. The lower layers of a deep network (whether it is supervised and feedforward or includes a generative component with downward arrows) can be shared across such tasks, while task-specific parameters (associated respectively with the weights into and from $h^{(1)}$ and $h^{(2)}$) can be learned on top of those yielding a shared representation $h^{(\text{shared})}$. The underlying assumption is that there exists a common pool of factors that explain the variations in the input x , while each task is associated with a subset of these factors. In this example, it is additionally assumed that top-level hidden units $h^{(1)}$ and $h^{(2)}$ are specialized to each task (respectively predicting $y^{(1)}$ and $y^{(2)}$) while some intermediate-level representation $h^{(\text{shared})}$ is shared across all tasks. In the unsupervised learning context, it makes sense for some of the top-level factors to be associated with none of the output tasks ($h^{(3)}$): these are the factors that explain some of the input variations but are not relevant for predicting $y^{(1)}$ or $y^{(2)}$.

Improved generalization and generalization error bounds (Baxter, 1995) can be achieved because of the shared parameters, for which statistical strength can be