top of pretrained features, the features must make the underlying classes linearly separable. These properties often occur naturally but do not always do so. This is another reason that simultaneous supervised and unsupervised learning can be preferable—the constraints imposed by the output layer are naturally included from the start.

From the point of view of unsupervised pretraining as learning a representation, we can expect unsupervised pretraining to be more effective when the initial representation is poor. One key example of this is the use of word embeddings. Words represented by one-hot vectors are not very informative because every two distinct one-hot vectors are the same distance from each other (squared L^2 distance of 2). Learned word embeddings naturally encode similarity between words by their distance from each other. Because of this, unsupervised pretraining is especially useful when processing words. It is less useful when processing images, perhaps because images already lie in a rich vector space where distances provide a low quality similarity metric.

From the point of view of unsupervised pretraining as a regularizer, we can expect unsupervised pretraining to be most helpful when the number of labeled examples is very small. Because the source of information added by unsupervised pretraining is the unlabeled data, we may also expect unsupervised pretraining to perform best when the number of unlabeled examples is very large. The advantage of semi-supervised learning via unsupervised pretraining with many unlabeled examples and few labeled examples was made particularly clear in 2011 with unsupervised pretraining winning two international transfer learning competitions (Mesnil *et al.*, 2011; Goodfellow *et al.*, 2011), in settings where the number of labeled examples in the target task was small (from a handful to dozens of examples per class). These effects were also documented in carefully controlled experiments by Paine *et al.* (2014).

Other factors are likely to be involved. For example, unsupervised pretraining is likely to be most useful when the function to be learned is extremely complicated. Unsupervised learning differs from regularizers like weight decay because it does not bias the learner toward discovering a simple function but rather toward discovering feature functions that are useful for the unsupervised learning task. If the true underlying functions are complicated and shaped by regularities of the input distribution, unsupervised learning can be a more appropriate regularizer.

These caveats aside, we now analyze some success cases where unsupervised pretraining is known to cause an improvement, and explain what is known about why this improvement occurs. Unsupervised pretraining has usually been used to improve classifiers, and is usually most interesting from the point of view of