internal representation across tasks but also by learning the learning algorithm itself (Hospedales *et al.*, 2021). Meta-learning can be used to facilitate generalization of, for example, a classification model to new classes when there are very few labelled examples of the new classes. This is referred to as *few-shot learning*. When only a single labelled example is used it is called *one-shot learning*.

6.3.5 Contrastive learning

One of the most common and powerful representation learning methods is *contrastive learning* (Gutmann and Hyvärinen, 2010; Oord, Li, and Vinyals, 2018; Chen, Kornblith, *et al.*, 2020). The idea is to learn a representation such that certain pairs of inputs, referred to as positive pairs, are close in the embedding space, and other pairs of inputs, called negative pairs, are far apart. The intuition is that if we choose our positive pairs in such a way that they are semantically similar and choose negative pairs that are semantically dissimilar, then we will learn a representation space in which similar inputs are close, making downstream tasks, such as classification, much easier. As with other forms of representation learning, the outputs of the trained network are typically not used directly, and instead the activations at some earlier layer are used to form the embedding space. Contrastive learning is unlike most other machine learning tasks, in that the error function for a given input is defined only with respect to other inputs, instead of having a per-input label or target output.

Suppose we have a given data point $\mathbf{x}$ called the *anchor*, for which we have specified another data point $\mathbf{x}^+$ that together with $\mathbf{x}$ makes up a positive pair. We must also specify a set of data points $\{\mathbf{x}_1^-, \ldots, \mathbf{x}_N^-\}$ each of which makes up a negative pair with $\mathbf{x}$. We now need a loss function that will reward close proximity between the representations of $\mathbf{x}$ and $\mathbf{x}^+$ while encouraging a large distance between each pair $\{\mathbf{x}, \mathbf{x}_n^-\}$. One example of such a function, and the most commonly used loss function for contrastive learning, is called the *InfoNCE* loss (Gutmann and Hyvärinen, 2010; Oord, Li, and Vinyals, 2018), where NCE denotes 'noise contrastive estimation'. Suppose we have a neural network function $\mathbf{f}_\mathbf{w}(\mathbf{x})$ that maps points from the input space $\mathbf{x}$ to a representation space, governed by learnable parameters $\mathbf{w}$. This representation is normalized so that $||\mathbf{f}_\mathbf{w}(\mathbf{x})|| = 1$. Then, for a data point $\mathbf{x}$, the InfoNCE loss is defined by

$$E(\mathbf{w}) = -\ln \frac{\exp\{\mathbf{f}_\mathbf{w}(\mathbf{x})^{\mathrm{T}}\mathbf{f}_\mathbf{w}(\mathbf{x}^+)\}}{\exp\{\mathbf{f}_\mathbf{w}(\mathbf{x})^{\mathrm{T}}\mathbf{f}_\mathbf{w}(\mathbf{x}^+)\} + \sum_{n=1}^{N}\exp\{\mathbf{f}_\mathbf{w}(\mathbf{x})^{\mathrm{T}}\mathbf{f}_\mathbf{w}(\mathbf{x}_n^-)\}}. \tag{6.20}$$

We can see that in this function, the cosine similarity $\mathbf{f}_\mathbf{w}(\mathbf{x})^{\mathrm{T}}\mathbf{f}_\mathbf{w}(\mathbf{x}^+)$ between the representation $\mathbf{f}_\mathbf{w}(\mathbf{x})$ of the anchor and the representation $\mathbf{f}_\mathbf{w}(\mathbf{x}^+)$ of the positive example provides our measure of how close the positive pair examples are in the learned space, and the same measure is used to assess how close the anchor is to the negative examples. Note that the function resembles a classification cross-entropy error function in which the cosine similarity of the positive pair gives the logit for the label class and the cosine similarities for the negative pairs give the logits for the incorrect classes. Also note that the negative pairs are crucial as without them the