



Figure 7.8: A demonstration of adversarial example generation applied to GoogLeNet (Szegedy *et al.*, 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet’s classification of the image. Reproduced with permission from Goodfellow *et al.* (2014b).

to optimize. Unfortunately, the value of a linear function can change very rapidly if it has numerous inputs. If we change each input by ϵ , then a linear function with weights \mathbf{w} can change by as much as $\epsilon \|\mathbf{w}\|_1$, which can be a very large amount if \mathbf{w} is high-dimensional. Adversarial training discourages this highly sensitive locally linear behavior by encouraging the network to be locally constant in the neighborhood of the training data. This can be seen as a way of explicitly introducing a local constancy prior into supervised neural nets.

Adversarial training helps to illustrate the power of using a large function family in combination with aggressive regularization. Purely linear models, like logistic regression, are not able to resist adversarial examples because they are forced to be linear. Neural networks are able to represent functions that can range from nearly linear to nearly locally constant and thus have the flexibility to capture linear trends in the training data while still learning to resist local perturbation.

Adversarial examples also provide a means of accomplishing semi-supervised learning. At a point \mathbf{x} that is not associated with a label in the dataset, the model itself assigns some label \hat{y} . The model’s label \hat{y} may not be the true label, but if the model is high quality, then \hat{y} has a high probability of providing the true label. We can seek an adversarial example \mathbf{x}' that causes the classifier to output a label y' with $y' \neq \hat{y}$. Adversarial examples generated using not the true label but a label provided by a trained model are called **virtual adversarial examples** (Miyato *et al.*, 2015). The classifier may then be trained to assign the same label to \mathbf{x} and \mathbf{x}' . This encourages the classifier to learn a function that is