January 02, 2018

Few-shot learning using generative modeling

King Hong Leung

Alexander Toshev

Narayan Hegde

Yair Movshovitz-Attias

Few-shot learning using generative modeling

ABSTRACT

In many machine learning tasks, the available training data has a skewed distribution- a small set of training classes for which a large number of examples are available ("base classes"), and many classes for which only a limited number of examples are available (few-shot classes). This is known as the long-tail distribution problem. Few-shot learning refers to understanding new concepts from only a few examples. Training a classifier on these few-example classes is known as the few-shot classification task.

Techniques disclosed herein improve classification accuracy for few-shot classes by leveraging examples from the base classes. A generative machine-learning model is trained using the base class examples and learns essential properties of the base classes. These essential properties, representing the intersection between base and few-shot classes, are applied to few-shot classes to generate additional few-shot examples. The generated few-shot examples are used to train a machine classifier to achieve better classification of inputs from few-shot classes.

KEYWORDS

- Few-shot learning
- Variational auto-encoder
- Generative modeling
- Training data
- Machine learning
- Generative adversarial network

BACKGROUND

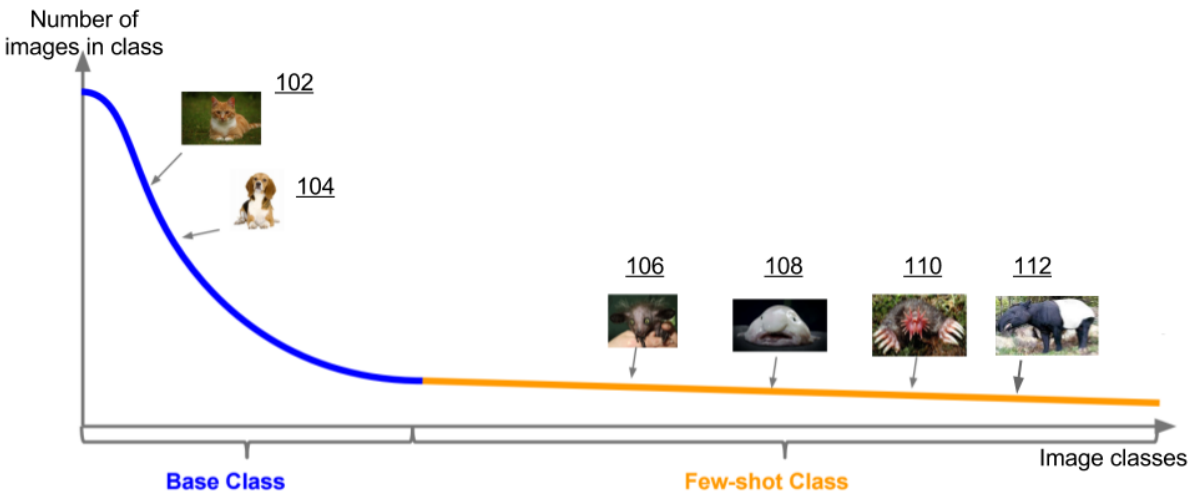

Fig. 1: Training data for machine classifiers follows a power-law distribution

Training data for machine classifiers often follows a power-law distribution. Fig. 1 illustrates this phenomenon: in training datasets, images of cats (102) and dogs (104) are a lot easier to find than images of aye-aye lemurs (106), blob-fishes (108), star-nosed moles (110), or Malayan tapirs (112). It follows that it is easier to train a machine classifier to recognize cats than to recognize, for example, star-nosed moles. The rare classes, e.g., 106-112 of Fig. 1, are referred to as few-shot classes, since there are few training examples available for these classes. The common classes, e.g., 102-104 of Fig. 1, are referred to as base classes. Typically, the accuracy of a machine classifier is low for few-shot classes, owing partly to the limited number of training examples that are available.

DESCRIPTION

Techniques of this disclosure enable accurate classification of few-shot classes. A generative machine-learning (ML) model, e.g., a variational auto-encoder (VAE) or a generative adversarial network (GAN), is trained to learn essential properties of the base classes. The essential properties are properties that translate to the few-shot classes. Having

learned such properties, the generative machine-learning model is utilized to generate a large number of few-shot examples. These newly generated few-shot examples, along with the original few-shot examples, are fed as training data to a machine classifier. In this way, a larger training set is made available to the machine classifier, enabling an improvement in classification for few-shot classes.

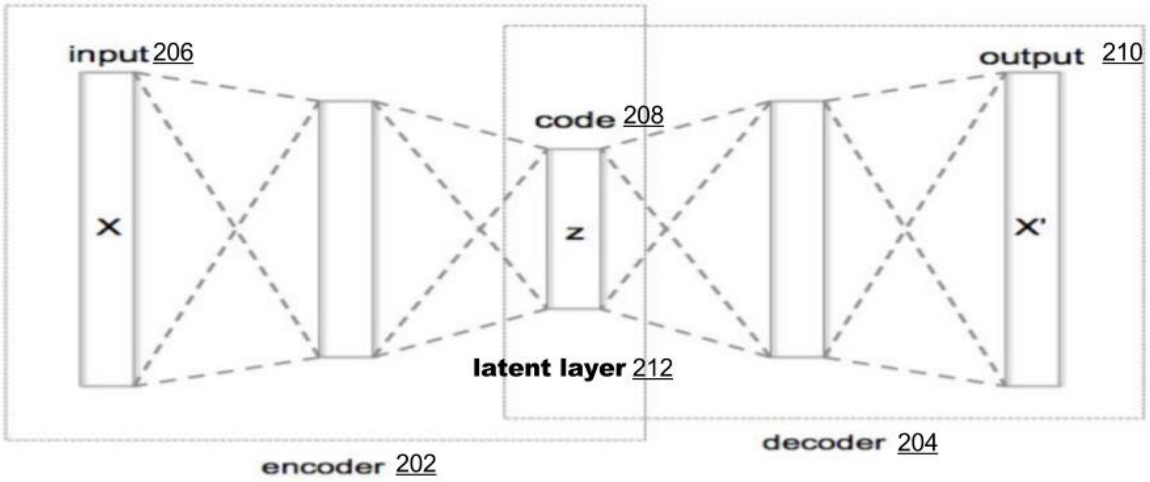

Fig. 2: Variational auto-encoder

Variational auto-encoders (VAE) are a type of generative machine-learning model that are capable of learning the essential properties of classes and generating new examples from those classes. As illustrated in Fig. 2, a variational auto-encoder comprises an encoder (202) and a decoder (204) connected by a latent layer (212). The encoder is a machine-learning model that takes input (206) and creates embedding or code (208) that comprises essential information relating to the input. The decoder is a machine-learning model that takes the code and generates as output (210) new examples of the class not included in the input. The code is stored in the latent layer (212), also known as the information bottleneck layer.

Example: A variational auto-encoder is fed with a large number of images of cats. It encodes as code certain generalizations about the input images. These generalized observations, that are

derived from training based on the input images apply to almost any cat (or four-legged animals), and are condensed by the encoder as code (208). The code may on some occasions have human-recognizable features (furry creature, four legs, etc.). However, mostly it comprises patterns in mathematical spaces that are not immediately interpretable by humans. The decoder takes these observations and generates new examples that were not part of the input. For example, after training on input images of several sitting cats and several standing cats, the VAE may take a particular input image of a sitting cat and transform its pose.

In this manner, the VAE generalizes from the input in an unsupervised manner, e.g., it generates plausible yet previously unseen output images. The VAE distills complex input (e.g., thousands of cat images, each with thousands of pixels) into a low dimensional space (e.g., a code vector that has only 100 entries). The VAE effectively learns the input distribution and then samples it to create multiple valid instances from this distribution.

Per techniques of this disclosure, a generative ML model, e.g., a VAE, is used to create new few-shot class examples that can be used to train a machine classifier. A loss function, e.g., the semantic similarity between input and output, is optimized to train the VAE to re-generate images that are presented at its input as output. Another loss function, e.g., the L2-norm (Euclidean distance) between output and input, is optimized to train the VAE to transform images of a certain base class into new and valid images within that same class. In this manner, the VAE is trained over all base class examples along with the limited number of real few-shot examples that are available. Having been trained, the VAE can accept as input real few-shot examples, and generate at its output new and valid few-shot examples.

Example: A variational auto-encoder is fed with images of animals of different classes including a large number of images of common animals like cats and rare images of uncommon

animals like Malayan tapirs. It generates code and is trained to transform, e.g., the pose of a cat. The VAE is presented with a real image of a Malayan tapir. It generates at its output a valid image of the Malayan tapir in a different pose. The newly generated image of the Malayan tapir, not previously available in the corpus of images, serves to augment the training set for a machine classifier.

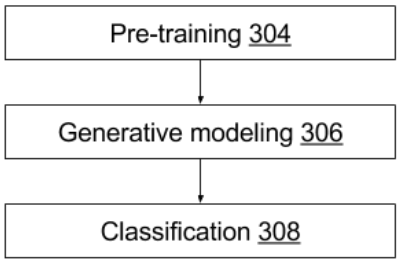

Fig. 3: The learning architecture comprises three machine-learning components

Fig. 3 shows the steps in augmenting a training set, per techniques of this disclosure. Three machine-learning components work in sequence in order to augment few-shot training data. In a pre-training phase (304), an inception network is trained using base-class examples to generate features. For example, the inception network may be a convolutional neural network. The features generated by the inception network are known as pre-trained embeddings. Per techniques of this disclosure, the inception network ascertains relevant features of the input images in a self-trained manner. For example, the inception network accepts as input raw pixels of images, rather than features derived out of hand-coding or by mathematical transformations of the images. The inception network accepts images of a variety of dimensions, e.g., images of size 32x32. By distilling features of the input image to pre-trained embeddings, which generally are smaller in size than the image itself, the inception network serves as a dimension reducer.

The inception network preserves class-distinguishable features, and enables the next phase to train for complex image datasets.

In a generative modeling phase (306), the pre-trained embeddings determined by the inception network of the first phase are used to generate new, previously unseen embeddings or images of few-shot cases. The generative modeling phase uses generative ML models, e.g., VAE or generative adversarial network (GAN). In effect, the generative modeling phase learns the distribution that fits the complex high-dimensional image dataset used as input, and samples this distribution in order to generate previously unseen yet plausible images. The generative model is a core part of the pipeline that generates image with greater accuracy and diversity in order to improve the classification phase, which comes next.

In the classification phase (308), a classifier is trained over the base and augmented few-shot classes such that it can achieve accurate classification of instances from either class.

The accuracy of the learning architecture described herein is measured on the image classifier network of the classification phase using a set of few-shot examples. When tested on standard image datasets such as MNIST and CIFAR100, the techniques of this disclosure provide improvement over a standard classification scheme.

As discussed above, techniques of this disclosure use, for example, a VAE in the generative modeling phase. The VAE comprises an encoder and decoder deep neural network, connected by an information-bottleneck (also known as latent) layer. Per techniques of this disclosure, the latent layer is made stochastic.

For example, this layer is modeled as parameters, e.g., mean and standard deviation, of a Gaussian distribution. The Kullback-Leibler divergence between the latent-layer's distribution and the unit Gaussian distribution is optimized to enforce on the latent layer a close adherence

to Gaussianity. It also acts as regularization. The encoder network learns to capture enough information to represent just enough class features required to regenerate the image. The decoder network comprises de-convolutional networks in order to render back the image.

The final layer of decoder network models the parameter of another distribution. This layer can be used to sample more images from the same class. Base and few-shot examples are fed in such proportions that the latent layer can model the mean and variance of each class. Instances from the latent layer are sampled using Monte Carlo sampling techniques and fed into the decoder. Monte Carlo sampling is used to simulate (estimate) Kullback-Leibler divergence loss, parameter clipping of the latent layer's distribution, etc. Semantic similarity between images, which is a loss function that drives the training of the VAE, is measured by using another network pre-trained and well-performing on base classes.

The training-set augmentation techniques of this disclosure are usable in a standalone manner and can be combined with other few-shot learning or data-augmentation techniques. The techniques can use any suitable generative machine-learning model, e.g., GAN, pixel-RNN, etc. Loss functions other than L2-norm or semantic similarity can also be used to train the generative ML model. Aside from augmenting the training set, the techniques provide insight into the mechanism of transfer learning, for example, by observing at the output of the generative model those base-class properties that transfer to few-shot classes.

CONCLUSION

Techniques of this disclosure address the problem of scarce training data for machine classifiers. Generative machine-learning models are used to generate additional training data for a machine classifier. Per techniques described herein, the generative models are trained on both base classes (classes that have abundant examples) and few-shot classes (classes with scarce

training data) and learn essential properties of the base classes that transfer to the few-shot classes. In this manner, the generative models create plausible yet different and previously unseen examples of few-shot classes. The training set for the machine classifier is augmented with the newly generated few-shot examples. The classification accuracy of the machine classifier is improved by use of the generated examples, with better generalization performance for few-shot classes.