Graph Convolutions

The convolutional algorithms we've shown you thus far expect rectangular tensors as their inputs. Such inputs could come in the form of images, videos, or even sentences. Is it possible to generalize convolutions to apply to irregular inputs?

The fundamental idea underlying convolutional layers is the notion of a local receptive field. Each neuron computes upon the inputs in its local receptive field, which typically constitute adjacent pixels in an image input. For irregular inputs, such as the undirected graph in Figure 6-11, this simple notion of a local receptive field doesn't make sense; there are no adjacent pixels. If we can define a more general local receptive field for an undirected graph, it stands to reason that we should be able to define convolutional layers that accept undirected graphs.

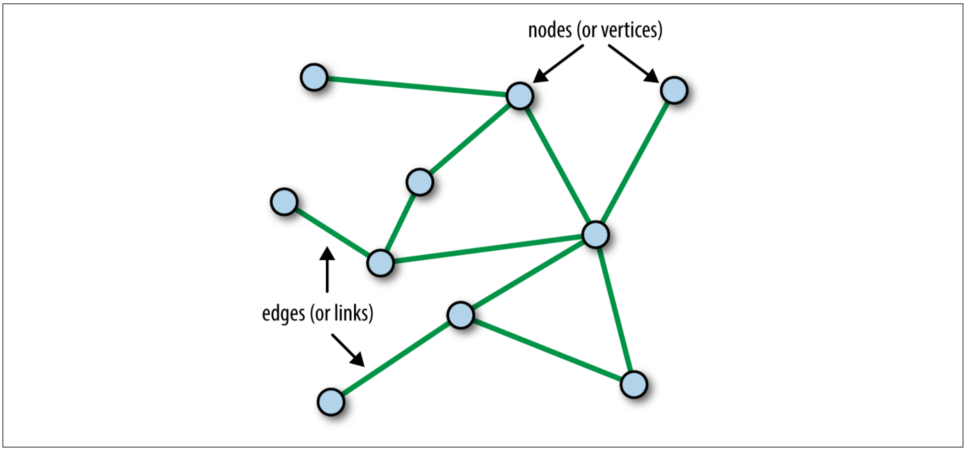

Figure 6-11. An illustration of an undirected graph consisting of nodes connected by edges.

As Figure 6-11 shows, a graph is made up of a collection of nodes connected by edges. One potential definition of a local receptive field might be to define it to constitute a node and its collection of neighbors (where two nodes are considered neighbors if they are connected by an edge). Using this definition of local receptive fields, it's possible to define generalized notions of convolutional and pooling layers. These layers can be assembled into graph convolutional architectures.

Where might such graph convolutional architectures prove useful? In chemistry, it turns out molecules can be modeled as undirected graphs where atoms form nodes and chemical bonds form edges. As a result, graph convolutional architectures are particularly useful in chemical machine learning. For example, Figure 6-12 demonstrates how graph convolutional architectures can be applied to process molecular inputs.

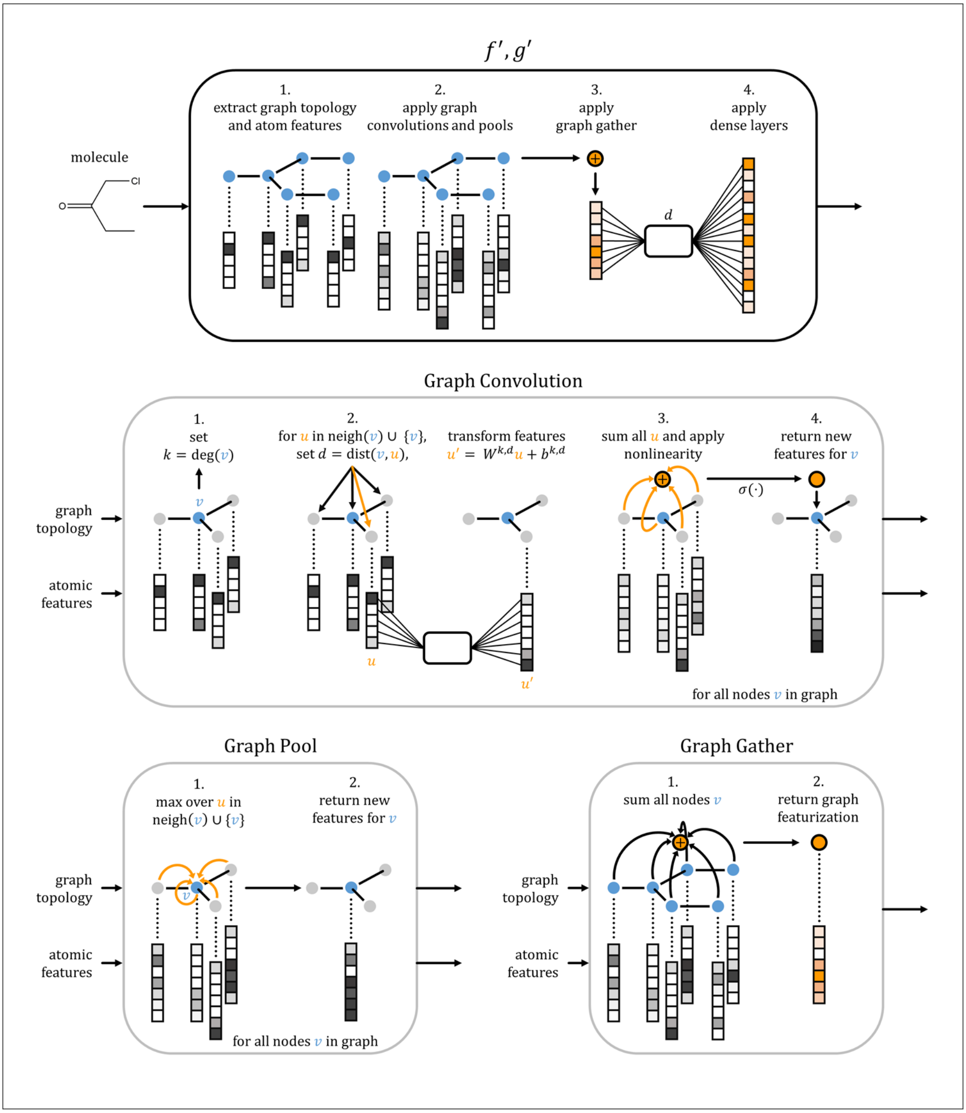

Figure 6-12. An illustration of a graph convolutional architecture processing a molecular input. The molecule is modeled as an undirected graph with atoms forming nodes and chemical bond edges. The "graph topology" is the undirected graph corresponding to the molecule. "Atom features" are vectors, one per atom, summarizing local chemistry. Adapted from "Low Data Drug Discovery with One-Shot Learning."

Generating Images with Variational Autoencoders

The applications we've described thus far are all supervised learning problems. There are well-defined inputs and outputs, and the task remains (using a convolutional network) to learn a sophisticated function mapping input to output. Are there unsupervised learning problems that can be solved with convolutional networks? Recall that unsupervised learning requires "understanding" the structure of input datapoints. For image modeling, a good measure of understanding the structure of input images is being able to "sample" new images that come from the input distribution.

What does "sampling" an image mean? To explain, let's suppose we have a dataset of dog images. Sampling a new dog image requires the generation of a new image of a dog that *is not in the training data*! The idea is that we would like a picture of a dog that could have reasonably been included with the training data, but was not. How could we solve this task with convolutional networks?

Perhaps we could train a model to take in word labels like "dog" and predict dog images. We might possibly be able to train a supervised model to solve this prediction problem, but the issue remains that our model could generate only one dog picture given the input label "dog." Suppose now that we could attach a random tag to each dog—say "dog3422" or "dog9879." Then all we'd need to do to get a new dog image would be to attach a new random tag, say "dog2221," to get out a new picture of a dog.

Variational autoencoders formalize these intuitions. Variational autoencoders consist of two convolutional networks: the encoder and decoder network. The encoder network is used to transform an image into a flat "embedded" vector. The decoder network is responsible for transforming the embedded vector into images. Noise is added to ensure that different images can be sampled by the decoder. Figure 6-13 illustrates a variational autoencoder.

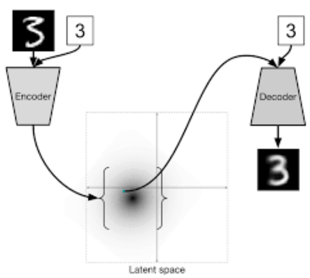

Figure 6-13. A diagrammatic illustration of a variational autoencoder. A variational autoencoder consists of two convolutional networks, the encoder and decoder.

There are more details involved in an actual implementation, but variational autoencoders are capable of sampling images. However, naive variational encoders seem to generate blurry image samples, as Figure 6-14 demonstrates. This blurriness may be