Equation 16-6. Q-Learning using an exploration function

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \cdot \max_{a'} f(Q(s',a'), N(s',a'))\right)$$

- $N(s', a')$ counts the number of times the action a' was chosen in state s'.
- $f(q, n)$ is an *exploration function*, such as $f(q, n) = q + K/(1 + n)$, where K is a curiosity hyperparameter that measures how much the agent is attracted to to the unknown.

Approximate Q-Learning

The main problem with Q-Learning is that it does not scale well to large (or even medium) MDPs with many states and actions. Consider trying to use Q-Learning to train an agent to play Ms. Pac-Man. There are over 250 pellets that Ms. Pac-Man can eat, each of which can be present or absent (i.e., already eaten). So the number of possible states is greater than $2^{250} \approx 10^{75}$ (and that's considering the possible states only of the pellets). This is way more than atoms in the observable universe, so there's absolutely no way you can keep track of an estimate for every single Q-Value.

The solution is to find a function that approximates the Q-Values using a manageable number of parameters. This is called *Approximate Q-Learning*. For years it was recommended to use linear combinations of hand-crafted features extracted from the state (e.g., distance of the closest ghosts, their directions, and so on) to estimate Q-Values, but DeepMind showed that using deep neural networks can work much better, especially for complex problems, and it does not require any feature engineering. A DNN used to estimate Q-Values is called a *deep Q-network* (DQN), and using a DQN for Approximate Q-Learning is called *Deep Q-Learning*.

In the rest of this chapter, we will use Deep Q-Learning to train an agent to play Ms. Pac-Man, much like DeepMind did in 2013. The code can easily be tweaked to learn to play the majority of Atari games quite well. It can achieve superhuman skill at most action games, but it is not so good at games with long-running storylines.

Learning to Play Ms. Pac-Man Using Deep Q-Learning

Since we will be using an Atari environment, we must first install OpenAI gym's Atari dependencies. While we're at it, we will also install dependencies for other OpenAI gym environments that you may want to play with. On macOS, assuming you have installed Homebrew, you need to run:

```
$ brew install cmake boost boost-python sdl2 swig wget
```