در اینجا مدل مسئله تشکیل شده از یک عامل، وضعیت ها S و مجموعه از اقدامات A برای هر وضعیت.با انجام یک اقدام $a \in A$، عامل از یک وضعیت به وضعیت بعدی حرکت کرده و هر وضعیت پاداشی به عامل می دهد. هدف عامل حداکثر کردن پاداش دریافتی کل خود است. این کار با یادگیری اقدام بهینه برای هر وضعیت انجام می گردد. الگوریتم دارای تابعی است که ترکیب حالت/اقدام را محاسبه می نماید :

$$Q : S \times A \to \mathbb{R}$$

قبل از شروع یادگیری، Q مقدار ثابتی را که توسط طراح انتخاب شده برمی گرداند. سپس هر بار که به عامل پاداش داده می شود، مقادیر جدیدی برای هر ترکیب وضعیت/اقدام محاسبه می گردد. هسته الگوریتم از یک بروز رسانی تکراری ساده تشکیل شده است. به این ترتیب که بر اساس اطلاعات جدید مقادیر قبلی اصلاح می شود.

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\overbrace{\underbrace{R(s_t)}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})}_{\text{max future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right]$$

که $R(s_t)$ پاداش s_t و $\alpha_t(s, a)$ است. نرخ یادگیری $(0 < \alpha \leq 1)$ ممکن است برای همه زوج ها یکسان باشد. مقدار عامل تخفیف γ بگونه است که $0 \leq \gamma < 1$

فرمول فوق معادل عبارت زیر است: $Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1 - \alpha_t(s_t, a_t)) + \alpha_t(s_t, a_t)[R(s_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$

یک اپیزود الگوریتم وقتی s_{t+1} به وضعیت نهایی می سد پایان می یابد. توجه کنید که برای همه وضعیت های نهایی s_f و $Q(s_f, a)$ مربوطه هیچگاه بروز نمی‌شود و مقدار اولیه خود را حفظ می کند.

این نرخ تعیین می کند که تا چه میزان اطلاعات بدست آمده جدید بر اطلاعات قدیمی ترجیح داده شود. مقدار صفر باعث می‌شود عامل چیزی یاد نگیرد و مقدار یک باعث می‌شود عامل فقط اطلاعات جدید را ملاک قرار دهد.

عامل تخفیف اهمیت پاداش های آینده را تعیین می کند. مقدار صفر باعث می‌شود عامل ماهیت فرصت طلبانه گرفته و فقط پاداش های فعلی را مد نظر قرار می دهد. در حالی که مقدار یک عامل را ترقیب می کند برای یک دوره زمانی طولانی برای پاداش تقلا کند. اگر این عامل، یک یا بیشتر از یک باشد مقادیر Q واگرا می شود.

در ساده ترین شکل کیو-یادگیری از جداول برای ذخیره داده استفاده می شود. این روش با پیچیده شدن سیستم مورد نظر، به سرعت کارایی خود را از دست می دهد. یک راه حل استفاده از شبکه عصبی بعنوان تخمین گر تابع است. از این روش تسارو در بازی تخته نرد استفاده کرد.