§1 強化学習とQ学習

人が教えるのではなく、機械が自ら学ぶという機械学習の趣旨に最も合致している学習法の1つが**強化学習**でしょう。いろいろと挑戦させて、より大きな価値のある行動を探す方法を用いて機械が学習します。その「強化学習」の代表的な手法が**Q学習**です。

強化学習の代表がQ学習

AIを実現する手法の1つに**強化学習**があります。この強化学習の考え方を理解するために、例として子供の「水泳の学習」を考えてみます。

子供に泳ぎを学ばせるとき、マニュアルで理解させることはしません。実際にプールに連れて行き、水の中で訓練します。その中で、親や先生の言うことを参考にしながら、子供は水泳の能力を習得していきます。自分の「行動」から「状態」を把握し、長く泳げるようになれたなら嬉しいという「報酬」を得ます。これを繰り返すことで、泳げるようになるのです。

強化学習は、これと同じ学習法をコンピューターで実現します。行動と報酬を組み合わせて機械自らが学んでいくのです。

この強化学習には様々な方法が考え出されています。先に述べたように、その中で最も古典的で有名なのが**Q学習**です。古典的といっても、現在、様々な機械学習の基本として各方面で利用され、その有効性が確かめられています。

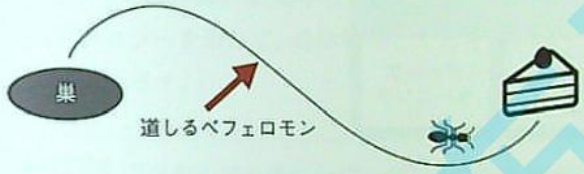
Q学習をアリから理解

Q学習は大変理解しやすい学習モデルです。本節では「アリが巣と餌場との最短経路を探す」という具体例で調べます。しくみがわかれば一般化は容易です。

注 実際のアリの動きは複雑です。以下の議論はアリの動きを単純化しています。

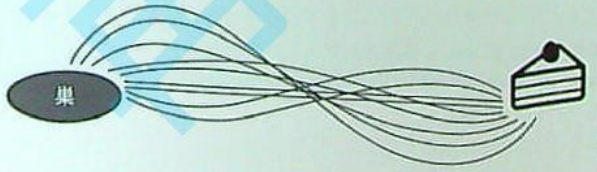
餌を探しに巣から出たアリが偶然に巨大なケーキに遭遇したとしましょう。このとき、ケーキを巣に運ぶために、何回も巣とケーキを往復することになります（アリは1匹だけとします）。アリも楽をしたいので、往復の中で最短のルートを発見していくことになります。このアリの立場になって考えてみましょう。

最初に留意すべき点は、アリは歩きながら「道しるべフェロモン」と呼ばれる匂いを道に付けることです。アリが迷わないのはこのためです。



アリは歩いた所に「道しるべフェロモン」と呼ばれる匂いを残す。

最初に来た道の匂いに従って往復すれば、アリはケーキを巣に運べます。しかし、楽をするために、アリはより短いルートを探したくなるはずです。そこで、最初のルートが最短ということは通常ありえないので、アリは最初のルートから少し外れた冒険ルートを探そうとします。この冒険心の御蔭で、往復回を重ねるごとに、最短ルートの近くで「道しるべフェロモン」の匂いは次第に濃くなることになります。結果として、強い匂いの方向に進めば、アリは最短ルートをたどることになるのです。

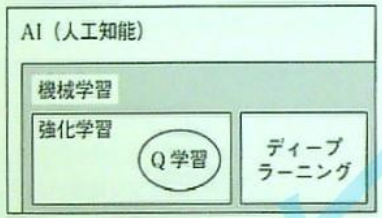


巣とケーキを往復するうちに、アリの付けた匂いが最短ルートで最強になる。

このように、「冒険心を持ちながら強い匂いの方向に進み、進みながら匂いを濃く書き換えていく」と仮定すると、往復を繰り返すうちに、アリは匂いの情報から最短ルートを歩くようになります。このアリの最短ルート探索のしくみを理想化したのがQ学習です。

機械学習と強化学習

強化学習は1980年頃から研究が盛んになってきました。現在話題のディープラーニングよりも先輩です。原理的には、ディープラーニングとは世界が異なります。下図で位置づけを見てみましょう。



AIにおけるQ学習の位置

Q学習はディープラーニングと融合し、さらに力を発揮します。碁や将棋で有力な棋士を圧倒したのも、この融合のもたらした結果です。この融合モデルがDQN（Deep Q Network）です。これについては次章で調べます。

MEMO Q学習とBellman最適方程式

人はある状態にいるとき、どのような行動をとるのが一番有益かを考えます。強化学習の基本もそこにあります。例えばロボットの学習を考えてみましょう。ロボットがある状態にあるとき、どのような行動をとるのが最も有益かを学ぶような学習アルゴリズムを作成するのです。このとき、「有益」という言葉は「価値」という言葉で表されます。現在、その価値の教え方として様々な方法が考え出されています。Q学習もその1つです。そして、この価値の満たすべき方程式はBellman最適方程式としてまとめられています。

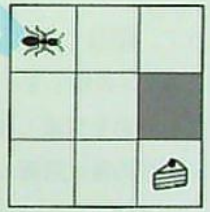
§2 Q学習のアルゴリズム

強化学習の代表例であるQ学習はわかりやすく、プログラミングも容易です。本節では、前節で調べたアリの動きを用いて話を進めましょう。

Q学習を具体例で理解

アリは歩きながら「道しるべフェロモン」と呼ばれる匂いを道に付けます。その匂いを頼りに、アリは巣穴から目的地までを往復できるのです。この匂いに導かれるアリの振る舞いは、Q学習を理解するうえで大変参考になります。そこで、このアナロジーを用いて、Q学習のしくみを調べましょう。具体的には、次の例題を考えます。

例題 正方形の壁の中に仕切られた8個の部屋が右図のようにあります。部屋と部屋の仕切りには穴があり、アリは自由に通り抜けられるとします。左上の部屋に巣があり、右下の部屋に報酬となるケーキがあります。アリが巣からケーキに行く最短経路探索の学習にQ学習を適用しましょう（右側中央の部屋には入れません）。

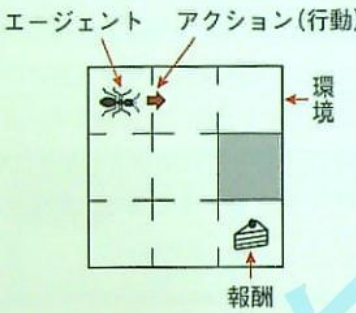


注 匂いは部屋の仕切りを通り抜けないとします。また、アリは記憶力をまったく持たないことを仮定します。

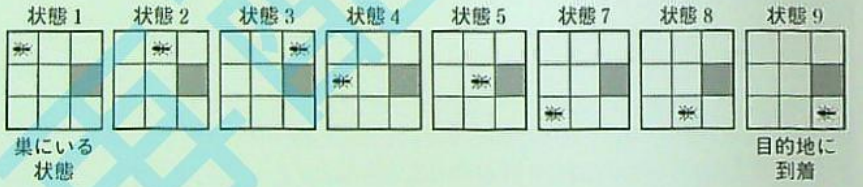
アリから学ぶQ学習の言葉

まずQ学習で利用される言葉を調べましょう。

例題 で調べるアリを、一般的に**エージェント**(agent)といい、アリの活躍する部屋全体を、一般的に**環境**と呼びます。また、アリは1つの部屋から隣の他の部屋に移りますが、この移る動作を**アクション**(action)と呼びます。アクションは、単純に**行動**とも呼ばれます。そして、目的地にあるケーキに与えられた数値を**報酬**(reward)といいます。

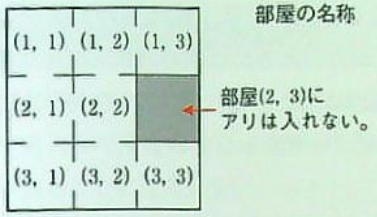


さて、例題 で規定する環境の下で、異なる様子が8個あります(下図)。この異なる8つの様子を、一般的に**状態**(state)と呼びます。以下では、次のように状態の名称を定義しましょう。「状態1」はアリが巣にいる状態です。「状態9」はアリが目的地に到着した状態です。



注 状態6は欠番ですが、プログラミング上、ダミーとして確保しています。

後の説明のしやすさのために、部屋には次の名称を付けることにします。



i行j列にある部屋を部屋(i, j)と表現する。

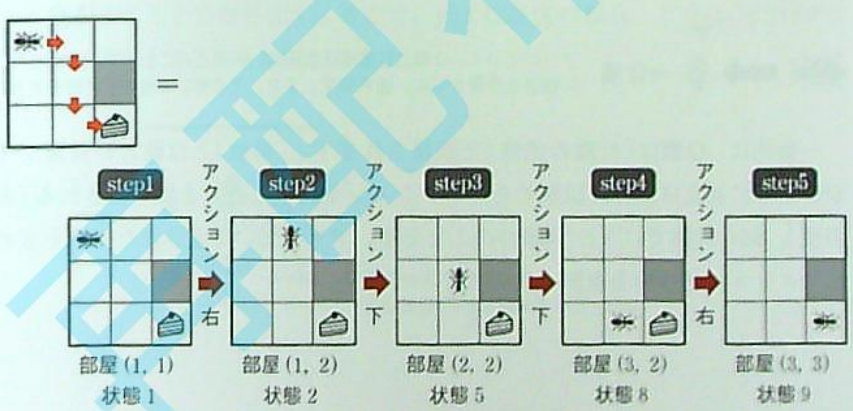
すると、i行j列にある「部屋(i, j)」と「状態番号s」は次の関係を持ちます。

$$s = 3(i-1)+j$$

注 状態とアリのいる部屋に「1対1」の対応があるので、この関係が成立します。

アリは左上の巣のある部屋(1, 1)からケーキのある部屋を(最短で)探しに行くことになります。その最初の部屋(1, 1)にアリがいる状態を最初の**ステップ**(すなわちステップ1番)と呼ぶことにします。そして、部屋を移動するたびにステップの番号を更新することにします。

例1 次の図は、状態1から4つの連続するアクション(右、下、下、右)で最終目標の状態9に達した場合を示しています。状態を変えるたびにステップ番号が更新されます。



本書ではステップ番号を「変数t」で表すことにします。

注 tはtimeの頭文字。段階(step)を時系列として捉えています。

この例1では、アリは部屋(1, 1)から目標の部屋(3, 3)に4回のアクション(5つのステップ)で到着できています。しかし、ときには決められた回数では到着できないときもあります。この到着の成否は別として、学習の一区切りのことを**エピソード**といいます。例1は1つのエピソードを示しています。

Q値

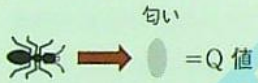
Q学習を式で表現するときに不可欠な値が**Q値**です。Q値とは「状態s」と「アクションa」によって決められる値です。すなわち、数学的に次のような多変数関数の形をしています。

$$\text{Q値} = Q(s, a) \cdots \boxed{1}$$

ここで、変数sはstate(状態)、aはaction(アクション)の頭文字です。

さて、このQ値とは何でしょうか。

例題のアリの場合、Q値とはアリを誘惑する匂いの強さです。「アリは道しるべフェロモン」の匂いを目安とし、進む道を探します。また、目的地にあるケーキの匂いにも誘惑されます。この匂いの強さがQ値の本質です。匂いの強さの大小、すなわちQ値の大小がアリの行動を決定するのです。



アリにとって、Q値とは誘惑される匂いの強さのこと。アリはこの匂いの強さを手掛かりに、道を探す。また、その匂いの強さを更新もする。

一般的に、Q値は「行動の価値」と表現されます。「価値」とは難しい言葉ですが、簡単に言えば、その状態でそのアクションを選択したときに期待される「魅力度」、別の言葉でいうと「報酬」のことです。アリは匂いで示された報酬を求めてアクション(行動)を選択するのです。

Q値が書かれる具体的な場所

例題において、アリのアクションとは部屋の出口を選択し、そこから部屋を移動することです。そこで、状態sにおけるQ値は図のように最大4つの出口に配置されることになります。

アリは「魅力度」、すなわち「報酬」を表すQ値の大きな出口を探してアクションを選択するはず。したがって、Q値は部屋の(最大)4か所の出口に書かれている必要がある。

状態sのとき、アリは最大4つのアクション(上、下、左、右)を選択できます。そこで、Q値は関数として次のように表現できます。

$$Q(s, \text{右}),\ Q(s, \text{上}),\ Q(s, \text{左}),\ Q(s, \text{下})$$

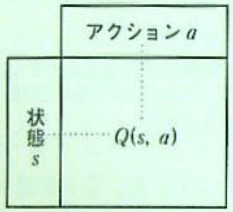
注 状態によって、アクションは制限されます。例えば$s=1$のとき、アクションは右と下の2つしかありません。

アリは原則として匂いの強い(すなわちQ値の大きい)値を目指してアクションを選択することになります。そこで、例えば次図の場合、アリは「下」のアクションを採用することを原則とします。

学習が済んでいるとき、アリはQ値の大きいアクション「下」を選択する。

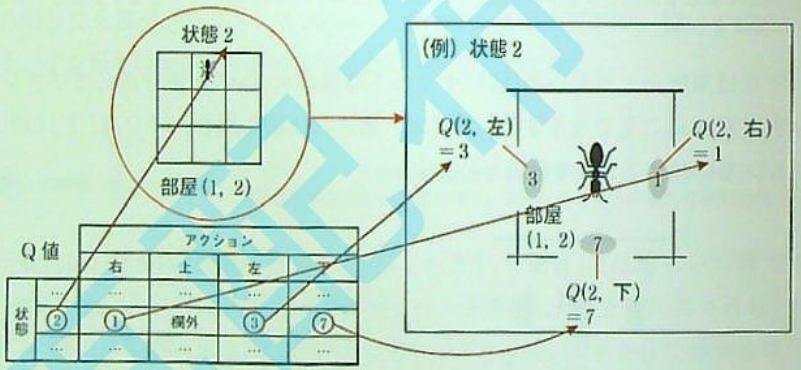
Q値の表とアリとの対応

式1に示すように、Q値は多変数関数として表されます。その多変数関数のイメージは表形式（すなわちテーブル）です。Q値の場合、表側が「状態」、表頭が「アクション」を表します。このように、Q値を表形式のイメージで理解しておくことは、Q学習の理解に大切です。また、後に調べるDQNを理解する上でも大切になります。



s、aが離散的な値をとるとき、多変数関数は表（すなわちテーブル）として表現できる。いまの例では、行動（アクション）aとして上、下、左、右の4種が存在する。状態sは1、2、3、4、5、7、8、9の8種。

いま考えている**例題**でこの表の意味を確認しましょう。下図は状態2の場合において、アクションとそれに対するQ値を例示しています。

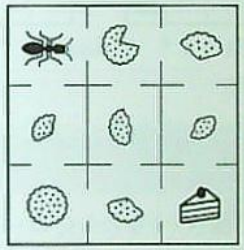


即時報酬

アリが目的の部屋への最短ルートを探しに行くとき、今いる部屋の隣に好物が落ちているかもしれません。アリは当然これを考慮してアクションを決定するはずです。このように、「隣の部屋に入る」という1アクションですぐに得られる「報酬」を**即時報酬**といいます。

注 即時報酬は負も可です。アリにとって不快に匂うものが部屋にある場合などです。

アリは即時報酬だけに魅了されてアクションを決定してはいけません。それでは目的地に到着できないからです。Q学習のアルゴリズムは、即時報酬だけにとらわれず、目標を目指すように作成しなければならないのです。



目的地の部屋に行く途中の部屋にクッキーの小片が落ちているとする。このクッキーもアリの好物。アリが目的の部屋にたどり着けるようにするには、このクッキーに惑わされないようなアルゴリズムをつくる必要がある。

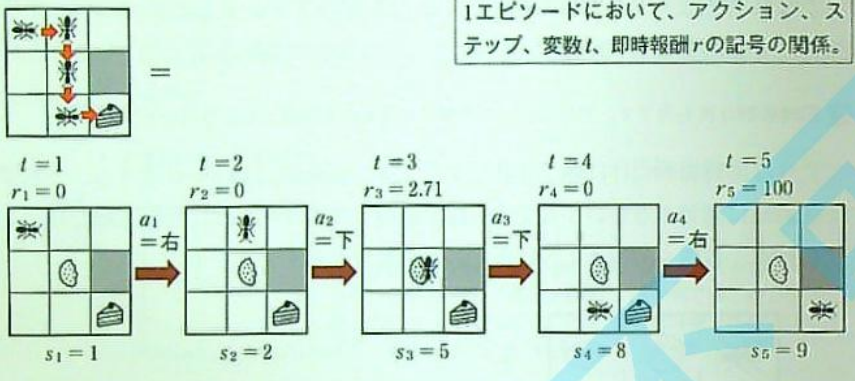
Q学習の数式で用いられる記号の意味

本書のQ学習で用いる記号の意味を表にしてまとめておきます。

〔表1〕

変数名	意味	アリの例
t	ステップ番号を表す変数	ステップ3のとき、$t=3$
s_t	ステップtにおける状態を表す変数	ステップ3の状態が5のとき、$s_3=5$
a_t	ステップtで選択するアクションを表す変数	ステップ3で選択したアクションが「右」のとき、$a_3=$「右」
r_t	ステップtにおいて、その場で受け取る即時報酬	ステップ3において、その場で受け取る即時報酬が10のとき、$r_3=10$

例2 先の例1で調べた様子を、ここで定義した記号で表現しましょう。ただし、部屋(2, 2)には新たにクッキー(即時報酬の値2.71)が置かれ、目的地の部屋(3, 3)にはケーキ(報酬の値100)があるとします。

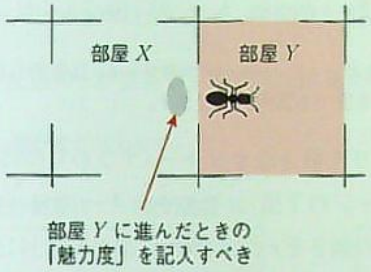


Q値の更新

アリは部屋を出るとき、その部屋の出口の匂いの強さ(すなわちQ値)を更新する必要があります。匂い情報を更新して、再訪時に最短の道を探しやすくするためです。

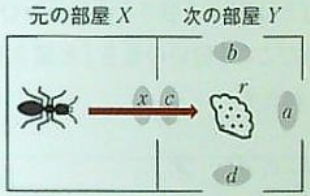
では、どのように更新するのでしょうか。

アリが「元の部屋」Xから「次の部屋」Yに進んだとします。このとき、Yに通じるXの出口に残すべき情報は、「次の部屋」Yに進んだときに得られる匂いの強さ(すなわちQ値)です。こうしておけば、部屋Xを再訪したとき、部屋Yについての的確な判断情報が得られるからです。部屋X再訪時に、Yに通じる出口情報を見るだけで、アリは部屋Yに行く「魅力度」(すなわち「価値」)がわかるわけです。



もう少し詳しく調べてみましょう。

「次の部屋」Yに通じる「元の部屋」Xの出口に記された匂いの強さ(Q値)をxとします。また、これから進む部屋Yの4つの出口の匂いの強さ(=Q値)をa、b、c、dとします。



匂いの強さx、a、b、c、dの位置関係。これらは部屋の出入口の足元に書かれている。部屋Yには匂いrを放つ好物=クッキーも置かれている。

注 部屋Yに4つの出口があるとします。状況に応じて適当に読んでください。

アリの気持ちになれば、次の部屋Yの「魅力度」はa、b、c、dの最大値で決まるはずです。部屋Yに入れば、その最大値が得られると期待されるからです。最大値(maximum)を表す記号maxを用いると、このことは次のように表現できます。

$$x\text{に設定する部屋}Y\text{の「魅力度」}=\max(a,\ b,\ c,\ d)$$

ところで、この魅力度を鵜呑みにするのは危険です。たとえば、匂いは時間とともに揮発し、減衰してしまうかもしれません。後から来るときには変化している可能性があるのです。そこで、多少割り引いた値を書き残さなければならないでしょう。その**割引率**をγとすると、「次の部屋」に行く魅力度は、現実には次の値になるはずです。

xに設定する部屋Yの「魅力度」$=\gamma \max(a, b, c, d)$ $(0<\gamma<1)$

注 γはギリシャ文字で「ガンマ」と読みます。γとr(ローマ字のアール)は区別しにくいのですが、多くの文献で採用されているので、本書でも慣例に従います。

また、これから進む部屋にはアリの好きなクッキー(すなわち即時報酬)が置かれていることもあります(前ページの下図)。このクッキーの匂いも魅力度に貢献します。そのクッキーの匂いの強さをrとすると、「次の部屋」に行く魅力度はさらに次のような式に変形されます。

xに設定する部屋Yの「魅力度」$=r+\gamma \max(a, b, c, d)$ … 2

学習率

アリにとってアクションを決める「魅力度」とは匂いの強さ(すなわちQ値)です。これまで「魅力度」と表現したことは、再びこの「匂いの強さ」と置き換えます。すなわち、上の式2は次のように表現されます。

「次の部屋」の匂いの強さ$=r+\gamma \max(a, b, c, d)$ … 3

注 本書では、この式3の値を「期待報酬」と呼びます。その部屋に入ると手に入るであろうと思われる魅力度だからです。

ところで、上の図において、この式3の「匂いの強さ」を「元の部屋」の出口情報xの更新情報としてそのまま採用してよいでしょうか。答はNoです。「次の部屋」Yに正しい匂い情報が記録されている保証はないからです。アリの学習が完了していなければ、この式3の値を100%信じることはできないのです。

そこで、学習の進み具合として**学習率**αを導入しましょう$(0<\alpha<1)$。そして、以前の情報xと、新たに求めた値3とを次のように混ぜ合わせて更新値xとします。

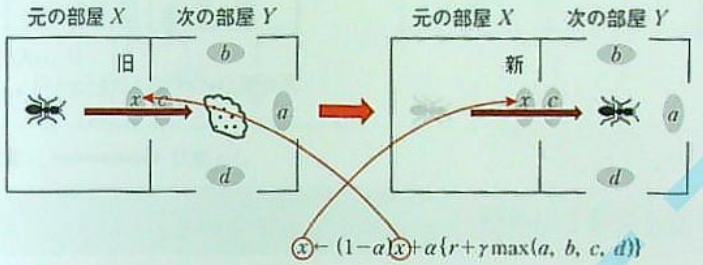
$x \leftarrow (1-\alpha)x+\alpha\{r+\gamma \max(a, b, c, d)\}$ … 4

変形すると、次のようにも表現できます。

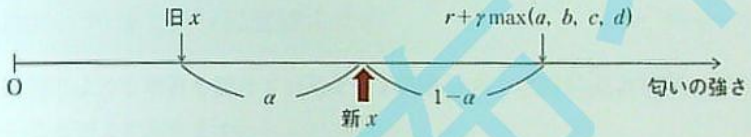
$x \leftarrow x+\alpha\{r+\gamma \max(a, b, c, d)-x\}$ … 5

ここで、左辺のxが更新値、右辺のxは更新前の値です。

注 αはモデル設計者が与えます。



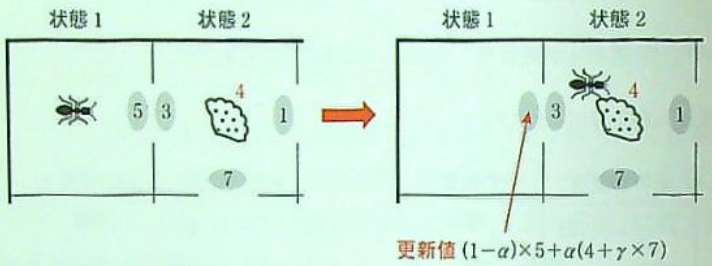
式4は数学では「内分の公式」として有名です。図で表すと、次のようになります。



この図が示すように、元の部屋の旧情報xと、これから進む次の部屋の新情報$r+\gamma \max(a, b, c, d)$を、式4は秤にかけているのです。

例3 アリが部屋(1, 1)から部屋(1, 2)に進むとします。各部屋には次ページの図右のように匂いの強さ(=Q値)が記されているとしましょう。アリが隣の部屋(1, 2)に進むとき、元の部屋(1, 1)の匂いの強さ5は、式4から次のように更新されます。

更新値$=(1-\alpha)\times 5+\alpha(4+\gamma\times 7)$



Q学習の記号で再表現

以上で得られた結論の式4（すなわち5）を、Q学習で利用される記号〔表1〕で表現してみましょう。これまで用いてきたxはQ値として、次のように表せます。

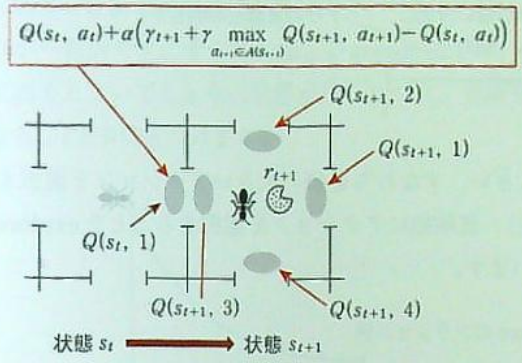
$$x = Q(s_t,\ a_t)$$

そこで、結論式5は次のように表現できます。

$$Q(s_t,\ a_t) \leftarrow Q(s_t,\ a_t)+\alpha\left(\gamma_{t+1}+\gamma \max_{a_{t+1}\in A(s_{t+1})} Q(s_{t+1},\ a_{t+1})-Q(s_t,\ a_t)\right) \cdots 6$$

この式6がQ学習の公式となります。この左辺の値は、アリが再訪したときに観測できる値です。その意味で左辺の値を**遅延報酬**と呼びます。遅延報酬を計算することがQ学習の原理となるわけです。

注 $a_{t+1}\in A(s_{t+1})$は数学の集合論の記号です。$A(s_{t+1})$はエージェントが状態s_{t+1}にあるとき選択できるアクションの集合を表します。そこで、$a_{t+1}\in A(s_{t+1})$は「a_{t+1}がそのアクションの集合Aの要素である」ことを示しています。



式6の各項の意味。この例では、$a_t=1$（すなわち右移動）と仮定。

ちなみに、式6の右辺（ ）の中の次式を「期待報酬」と呼ぶことは、先に調べました。

$$\text{期待報酬} = \gamma_{t+1}+\gamma \max_{a_{t+1}\in A(s_{t+1})} Q(s_{t+1},\ a_{t+1}) \cdots 7$$

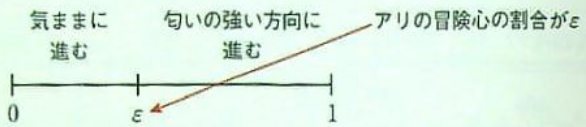
ε-greedy法でアリに冒険させる

人は同じような学習を続けていると、いつかスランプに陥り、目的地に達せられないことがよくあります。

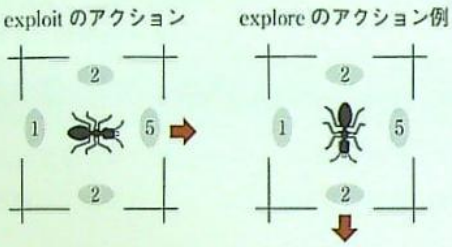
これはアリの経路学習についても同じです。現在の匂いの強さだけを頼りに進むべき部屋を選んでいると、迷路にはまり、アリは永遠に目的地にたどり着けないこともあるのです。そこで、これを回避し目的地にたどり着くようにするには、匂い情報だけに頼るのではなく、新しい道を探す冒険心が必要になります。この冒険心を取り入れる方法で有名なのがε-greedy法です。ときには冒険的になり、匂いの強さにかかわらず別の方向の部屋に進むことも許す方法です。

確率的にこの気まぐれを取り入れれば、新たな道を探せるチャンスが生まれます。この冒険的の確率をεで表します。確率εの割合で、勝手なアクションを許すわけです$(0<\varepsilon<1)$。

注 εはギリシャ文字で、イプシロンと読まれます。ローマ字のeに対応します。



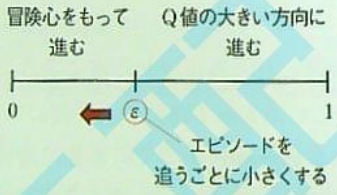
Q学習では、匂いの強さの大きい、すなわちQ値の大きいアクションを選択することを**exploit**(利用し尽くす)、冒険的にアクションを選択することを**explore**(探検する)と英語で表現しています。



この図のように匂いの強さが記されているとする。この場合、左の図がexploit、右の図がexploreの行動例。

ちなみに、exploit的な行動を**グリーディ**(greedy(欲張りな))と表現します。

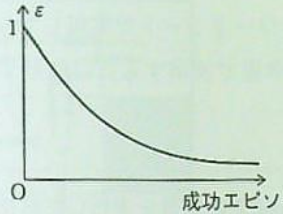
さて、ε-greedy法では、冒険の確率εが固定されています。そのεを、最初は大きく、学習が進むにつれて次第に小さくすると、学習の速さは向上することが知られています。この工夫を取り入れたのが**修正ε-greedy法**です。



アリの冒険心の割合がε。成功したエピソードが増えるにつれ、すなわち学習が進むにつれ、このεの値を次第に小さくするのが修正ε-greedy法。

この考え方は日常の経験にマッチします。何かを学ぶとき、最初はやみくもに努力しますが、学習が進むにつれてコツがわかり、次第に定型的な学習になります。この経験を取り入れるのです。

通常、Q値の初期値は不明なので、学習の初めには適当に値を割り振っておくのが一般的です。そこで、修正ε-greedy法では、Q学習の最初でεを1に設定しておくとよいでしょう。学習が進むにつれ、冒険をする必要が少なくなってきたなら、εを0に近づけます。



修正ε-greedy法におけるεの設定のイメージ。

学習の終了条件

学習が終了したと判断される条件は、Q値が学習によって一定値に収束することです。それは人の学習と同じです。いくら学習を積んでも成績が変わらなくなれば、その学習を打ち切ることになるでしょう。

Q値が収束するということは、Q値が学習によって変わらなくなることです。式6でそれを見ると、次のように表現できます。

$$\gamma r_{t+1} + \gamma \max_{a_{t+1} \in A(s_{t+1})} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \to 0 \cdots 8$$

すなわち、学習の終了条件は次のように表現できます。

$$\gamma r_{t+1} + \gamma \max_{a_{t+1} \in A(s_{t+1})} Q(s_{t+1}, a_{t+1}) \to Q(s_t, a_t) \cdots 9$$

式9の左辺を「期待報酬」と呼びました(式7)。現在のQ値と期待報酬が同じになれば飽和状態であり、それ以上は学習の必要はないということを、式9は意味しているわけです。

さて、学習が終了したと判断された場合、explore的アクションは不要になります。行動は「Q値の大きいアクションを選択する」というexploit的(すなわちグリーディな処理)に徹すればよいわけです。

§3 Q学習をExcelで体験

これまでに調べてきたQ学習をExcelのワークシートで実現してみましょう。前節で用いた例題を具体例とします。Q学習で実演するには簡単すぎますが、しくみを理解するには最適です。

> **演習** ▶§2で調べた例題について、ExcelでQ学習を実行してみましょう。なお、目的地の部屋に到着したとき、その報酬値は100とします。また、各部屋の即時報酬は−1とします。

注 本節のワークシートは、ダウンロードサイト（▶244ページ）に掲載されたファイル「7.xlsx」にあります。

▶ワークシート作成上の留意点

ワークシートに実装する際の注意点を調べます。

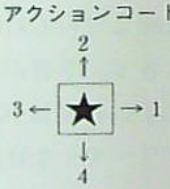
■（i）アリとケーキの表現

表記の簡略化のために、アリの表現には★を用い、目的地にあるケーキは「終」と表記します。

■（ii）アクションコード

アクションについては、コード化しておくと便利なときもあります。そこで、「アクションコード」として、次のように約束しておきます。

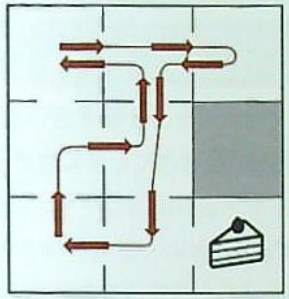
移動	右	上	左	下
アクションコード	1	2	3	4



注 コードは左回転（数学の正の向き）の順に付けられています。

■（iii）最大ステップ数・最大エピソード数

簡単な例なので、1エピソード中の最大ステップ数は10とします。そして、10回ステップを繰り返して目的地に到着しない場合は、そのエピソードは無視することにします。



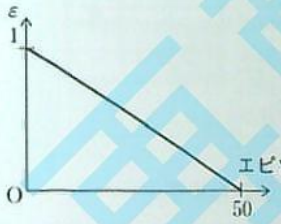
10回のステップを処理しても、目的地にたどり着かない例。このような場合には、そのエピソードは無視。

実験するエピソード数は50回とします。単純な演習なので、これくらい繰り返せば、十分学習が進むことが期待されるからです。

■（iv）修正ε-greedy法のεの値

本書では修正ε-greedy法を用いることにします（▶§2）。ここでは、εを次のように可変にします。分母の50は最大エピソード数のことです。

$$\varepsilon = 1 - \frac{\text{到着エピソード数}}{50} \quad \cdots \boxed{1}$$



式[4]のグラフ。最初のエピソードでは、全ステップがexploreのアクションとなる。最後のエピソードでは、ほぼexploitのアクションになる。

■(v)割引率と学習率の設定

割引率γは0.7、学習率αは0.5としました。通常、割引率γは0.9以上、学習率αは0.1程度に設定しますが、本演習は単純であり、収束を早くさせたいので、この値を利用します。

▶ExcelでQ学習

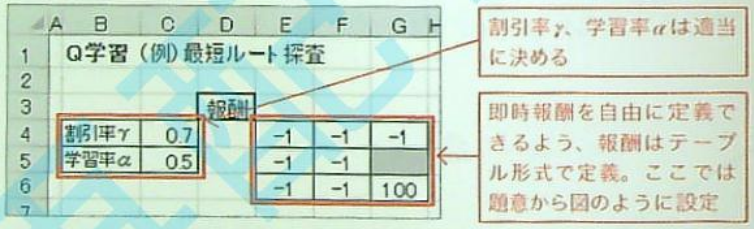
以上で準備が整いました。実際にワークシートでQ学習を、段階を追いながら実行してみましょう。

① Q学習のための全体のパラメーターを設定します。

▶§2で調べた「割引率」、「学習率」を設定します。これらの値は設計者が適当に決めます。また、アリが目標の部屋に到着したとき、報酬は100とします。本ワークシートを改変しやすいように、即時報酬を定義できる欄も用意しています。

なお、各部屋の即時報酬を−1としたのは、長い経路にペナルティーを課しダラダラ探すのを排除するためです。

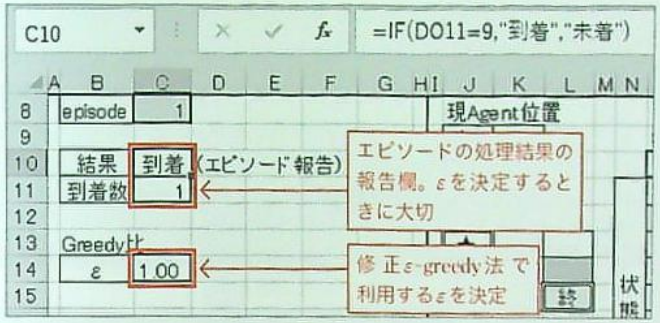
注 多くの文献では目的地の報酬を1にしますが、ここでは結果の数値の見やすさを優先します。



② 該当エピソードで利用する修正ε-greedy法のεを決定します。

修正ε-greedy法を利用するときに必要な確率εの値を設定します（式[1]）。また、Q学習の大きな単位はエピソードなので、そのエピソードの処理結果をまとめます。

注 本書では、印刷の都合上、到達しない場合はカットしています。



③ 該当ステップのアリの状態とそのときのQ値の表を設定します。

現ステップにおけるアリ（Agent）の状態を確認します。また、学習の開始時（すなわち最初のエピソードの最初のステップ）では、現Q値の表を乱数を用いて作成します。

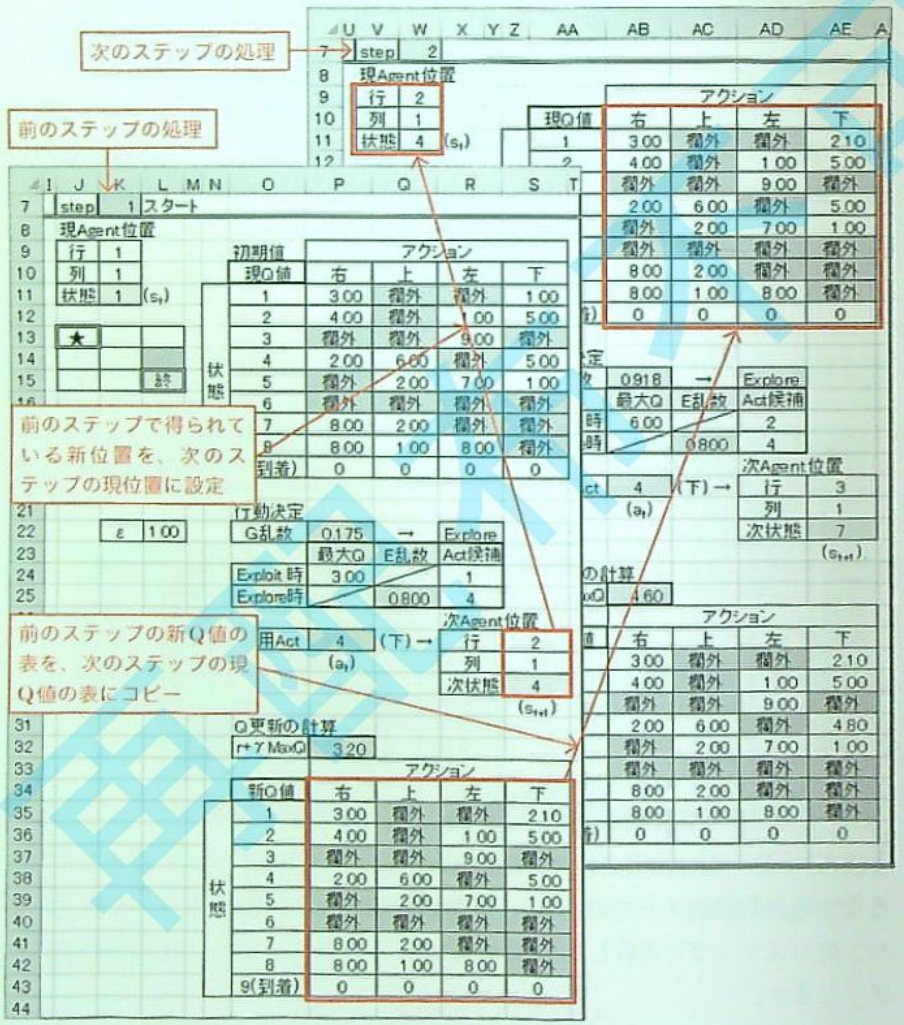
注 エピソードの最初のステップでは、アリは部屋(1, 1)（＝状態1）にいます。

該当エピソードの2番目以降の新ステップでは、前のステップで求められている「次Agent位置」と「次状態」を、「現Agent位置」と「状態」にセットします。また、前のステップで更新した「新Q値」の表を、新ステップの「現Q値」の表にコピーします。

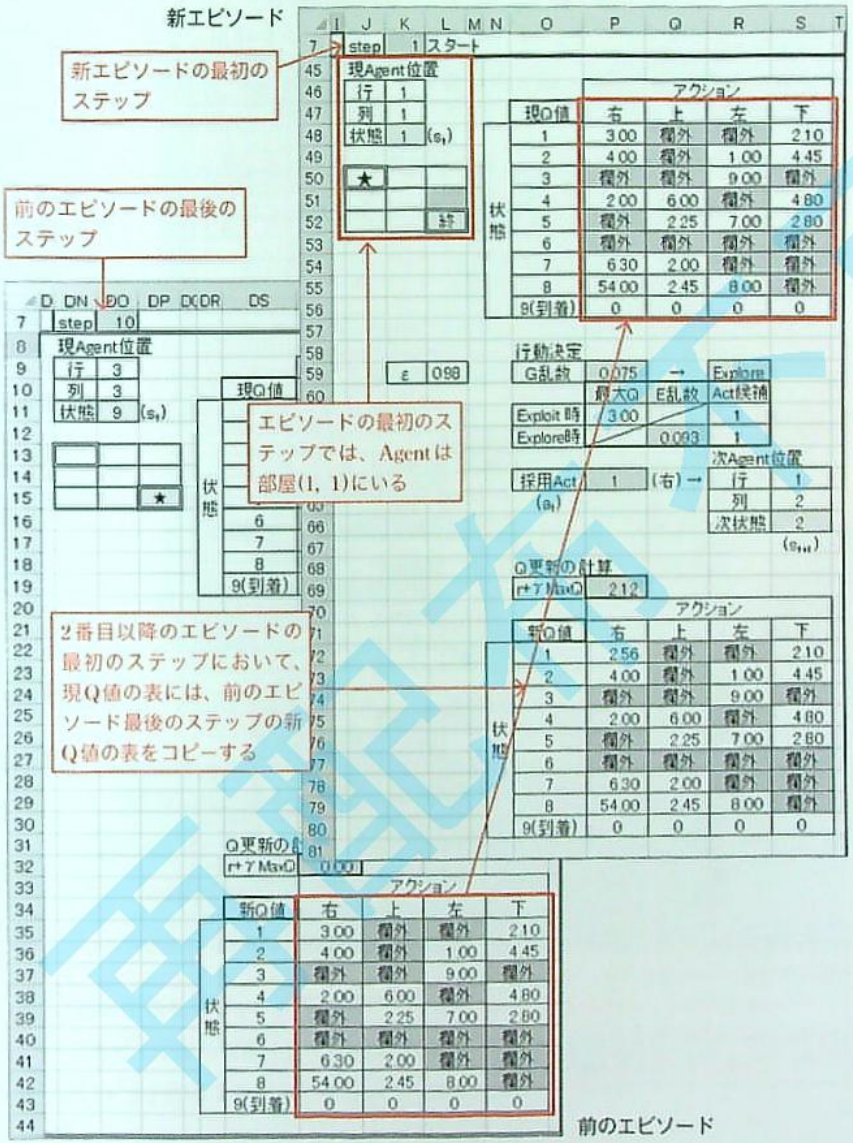
MEMO　**1ステップQ学習**

Q学習にも様々なバリエーションがあります。本書で採用した方法は1ステップQ学習と呼ばれる方法です。次ステップ$t+1$の期待報酬値を計算し、すぐに元ステップtのQ値の更新を行います。

これとは別に、アクションを一連の時系列動作として捉え、過去にさかのぼってQ値の更新をいっきに行う方法も有名です。



2番目以降のエピソードの最初のステップの「現Q値」の表には、前のエピソードの最後のステップ(ステップ10)で求められている「新Q値」の表を採用します。

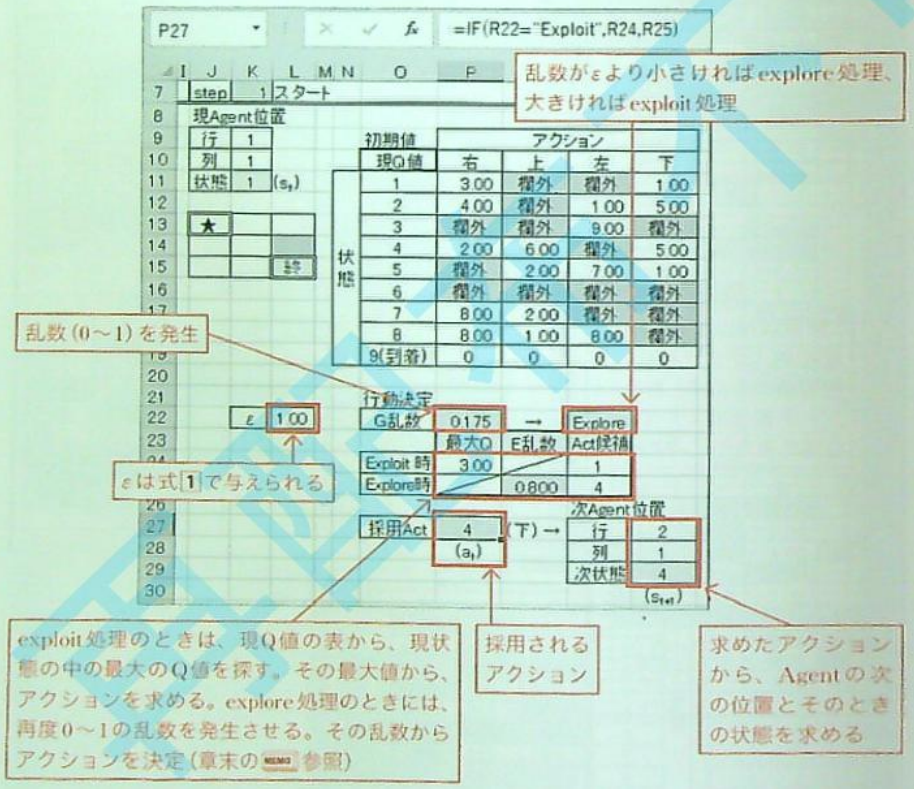


④ 採用するアクションがexploitか、exploreかを判断し、Agentの次の位置と状態を求めます。

ε-greedy法では、冒険的なアクション(explore)をとるか否かは0～1の乱数とεとの大小で判断します。

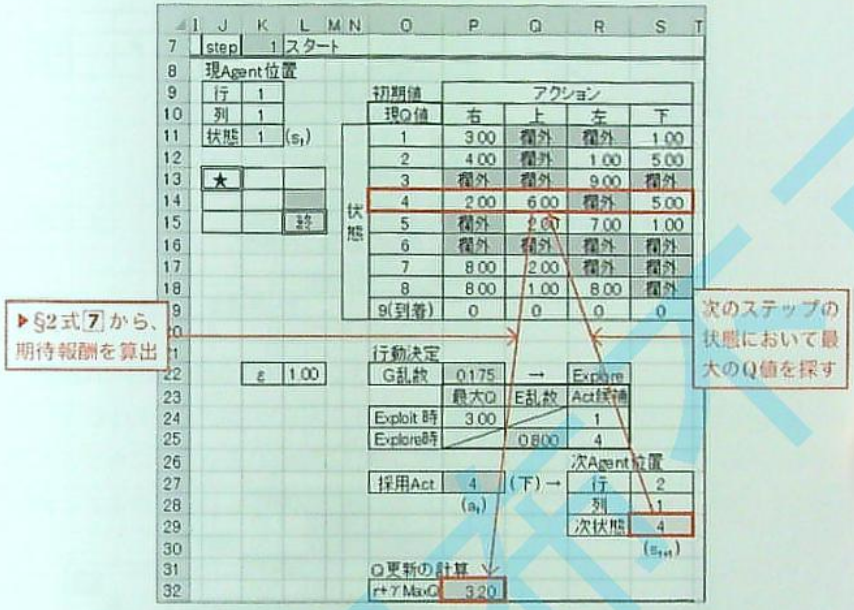
乱数がεより大きいときには、exploitのアクションを採用します。このとき、現Q値の表の該当状態で、最大のQ値を持つアクション(「上」「下」「左」「右」の移動)を採用します。

乱数がεより小さいときには、冒険的なアクション(explore)をとります。このとき、再度乱数を発生させ、その乱数の大きさに応じて次のアクション(「上」「下」「左」「右」の移動)を選択します。



⑤ Agentが得られる期待報酬値を算出します。

④で得られた次の状態から、現Q値の表を用いて、期待報酬値(▶§2式7)を算出します。ちなみに、この演習では、即時報酬の値r_{t+1}は−1としています($t+1$番目のステップで目的地に到着しない場合)。



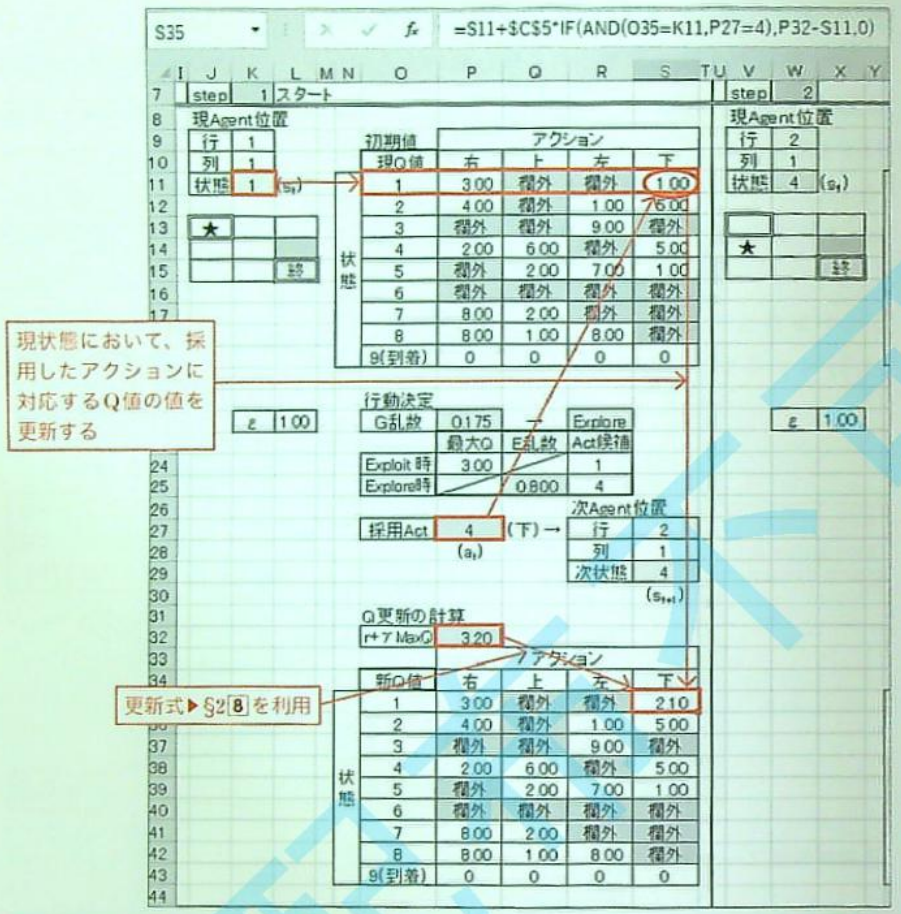
注 期待報酬はr＋γMaxQと表示しています。

MEMO　値のコピーにも配列数式が便利

Q学習ではステップを進めるごとに、前のステップで更新したQ値の表を次のステップのQ値の表にコピーする必要があります。この際に便利なのが配列数式の方法です。一気にコピーできるとともに、コピーミスがなくなります。

⑥ Q値を更新します。

⑤で求めた「期待報酬」を、新Q値の表の該当欄の更新値とします。それには、更新式(▶§2式6)を利用します。



以上でQ学習の1ステップとその流れは完成です。

ここで調べた1ステップのモジュールを10個分右にコピーし、1エピソード分を作成します(10は1エピソードの中の最大ステップ数)。さらにそれを50エピソード分コピーします(50はここで調べる学習回数です)。こうして、Q学習のワークシートが完成します。

⑦ 以上の学習で得られたQ値を利用して、学習したアリがどのように行動するか調べてみましょう。

得られた最終のQ値の表を見てみましょう。

		アクション			
	Q値	右	上	左	下
状態	1	31.96	欄外	欄外	32.11
	2	10.41	欄外	18.46	47.30
	3	欄外	欄外	26.90	欄外
	4	47.30	20.52	欄外	47.29
	5	欄外	31.81	21.62	69.00
	6	欄外	欄外	欄外	欄外
	7	69.00	24.85	欄外	欄外
	8	100.00	46.43	47.29	欄外
	9(到着)	0	0	0	0

部屋(1, 1)から出たアリ(すなわちAgent)は、このQ値の表に従って行動します。すなわち、「状態」が与えられたとき、この表の行に書かれた最大Q値に対応するアクションを選びながら行動します。このルールに従って、実際にアリに行動してもらいましょう(下図)。

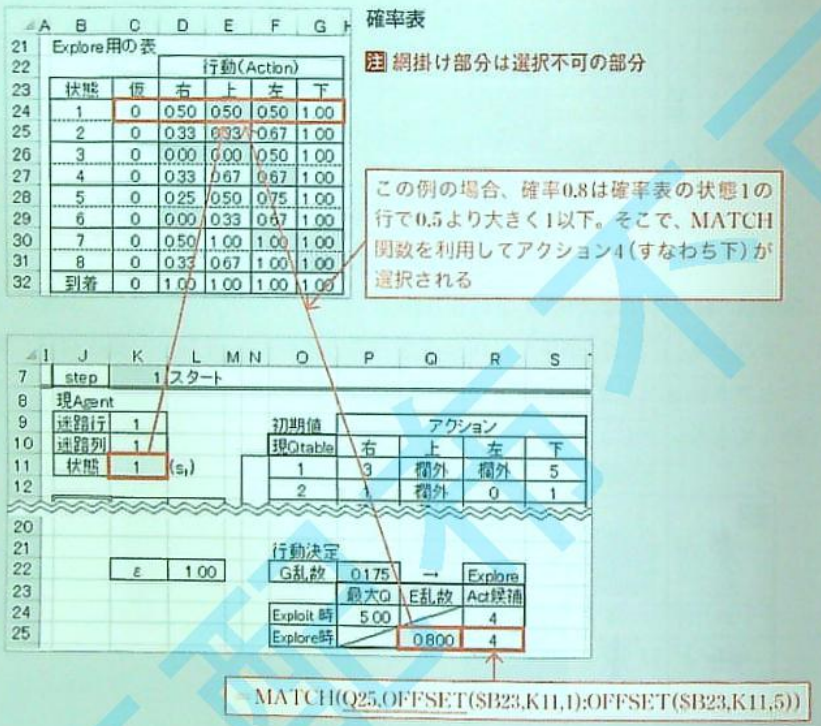
注 上記Q値について、小数部を四捨五入しているので、一部大小が不明の部屋があります。

Q学習の甲斐あって、最短ルートで目的地に到着しています。

以上の例は簡単なもので、エピソード回数は50と小さい数で済みました。実用上は、このような数では収まり切れないことに留意してください。

MEMO explore のアクションに確率を割り当てる方法

「exploit」の行動を選択すると、アクションを確率的に選択することになります。このとき、迷路や経路の問題では、選択に条件が付けられます。本節の例でいうと、たとえばある部屋では右に行けず、またある部屋では下には行けません。このとき、確率をアクションに簡単に割り当てるには、下図のような確率表を用意するとよいでしょう。この表とMTACH関数とを組み合わせることで、explore処理のアクションが選択できます。



8章 DQN

Q学習で用いられるQ値をニューラルネットワークで表現しようとする技法がDQNです。ニューラルネットワークには複雑な関数や表を整理してくれる性質があります。それをQ学習の結果の表現に応用するのです。