Equations to implement for temporal different learning for tetris. The equations are for a linear feature-based approximation architecture using approximate and optimistic λ-policy iteration.

<u>Variable:</u>

w : wall width

r : A feature vector of (2w + 2) entries. The first entry is a constant. (2w + 1) entries are values of features are listed below.

i : A board state in the game

<u>Features:</u>

h_k : The height of the kth column of the wall, k is from 1 to w

$|h_k - h_{k+1}|$: The absolute different between the heights of the kth and the (k + 1) column

$max_k h_k$: The maximum wall height

L : The number of holes in the wall

<u>Utility function:</u>

$$J(i,r) = r(0) + \sum_{k=1}^{w} r(k)h_k + \sum_{k=1}^{w-1} r(k+2)|h_k - h_{k+1}| + r(2w)max_k h_k + r(2w+1)L$$

Vector r can be assigned with random values first. For each iteration of the algorithm, we play M (in the order of 100) games. Suppose after t iterations the weights vector is r_t. We can find r_{t+1} by the following equation:

$$r_{t+1} = argmin_r \sum_{m=1}^{M} \sum_{k=0}^{N_m} \left(J(i_{m,k}, r) - J(i_{m,k}, r_t) - \sum_{s=k}^{N_m - 1} \lambda^{s-k} d(i_{m,s}, i_{m,s+1}) \right)^2$$

where
$(i_{m,0}, i_{m,1}, ..., i_{m,N_m - 1}, i_{m,N_m})$ is the sequence of board states comprising the mth game in the iteration and i_{m,N_m} is the terminal state

$d(i_{m,s}, i_{m,s+1}) = g(i_{m,s}, \mu_t(i_{m,s}), i_{m,s+1}) + J(i_{m,s+1}, r_t) - J(i_{m,s}, r_t)$ is the temporal difference

where $g(i_{m,s}, \mu_t(i_{m,s}), i_{m,s+1})$ is the number of rows clear of going from board state $i_{m,s}$ to board state $i_{m,s+1}$ using action $\mu_t(i_{m,s})$
$J(i_{m,N_m}, r_t) = 0$
λ is a chosen learning factor
μ_t is the policy function

The vector *r* in the equation above can be found my using a number of methods: stochastic gradient descent, which is great for parallel and distributed platforms but is only available in libraries as part of a supervised learning routine. Other optimization methods are readily available in libraries: Nelder-Mead Simplex algorithm, Broyden-Fletcher-Goldfarb-Shanno algorithm, Newton-Conjugate-Gradient algorithm.