**Model Details:**

**For simplicity and model accuracy, we fit the model to the concatenated session for each rat.**This method is better than fitting the model to each session since each session has few data and therefore prompts to produce a inaccurate estimate. We use MATLAB fmincon with multiple starting points to encourage global minimum of negative loglikelihood of free-choice trials. For the model, we have 2 models: model 1 assumes rewarded and unrewarded trials shares the same learning rate, thus symmetric learning, and model 2 assumes different learning rates for rewarded and unrewarded trials, thus allowing asymmetric learning such as loss aversion.

**Parameter:**

(learning rate), between 0 and 1. and **nr** for 2 rates model.

(inverse temperament), sensitivity to the difference in subjective reward values, > 0

**gamma**(memory decay rate or temporal discounting rate of action value function), between 0 and 1, 1 means no decay at all, 0 means total forget.

**b**(bias), negative means putting negative bias(averse) to sub-optimal choice, postive means giving extra value(prefer) to sub-optimal choice.

**Variables:**

**R(t)** = reward at trial t.

**Q(A)(t)** = action value function for action A at t.

**Action Value Function Update:**

**Model 1(4 parameters) - single learning rate:**

for all trial\_type:

Q(A)(t+1) =Q(A)(t) + \*(R(t) - Q(A)(t))

Q(~A)(t+1) = (1-gamma)\*Q(~A)

**Model 2(5 parameters) - 2 learning rate for rewarded and unrewarded trials:**

= rewarded trial learing rate, = unrewarded trial learing rate

for all trial\_type:

If R(t) == 1:

Q(A)(t+1) =Q(A)(t) + \*(R(t) - Q(A)(t))

Q(~A)(t+1) = (1-gamma)\*Q(~A)(t)

Else:

Q(A)(t+1) =Q(A)(t) + \*(R(t) - Q(A)(t))

Q(~A)(t+1) = (1-gamma)\*Q(~A)(t)

**For all action A1 and A2, the final output of A1 at trial t for example, is:**

P(A1) = Softmax(Q(A1)(t), Q(A2)(t),) =

Where P(A1) is the probability of choosing actions A1, given the action-value estimates Q(A1)(t) and Q(A2)(t) at time t.

Maximizing loglikelihood is same as minizing negative loglikelihood. Thus, the target function is given by minimizing mean **negative loglikelihood(NLL)** based on choice history. Forced trials are excluded **so we are only optimizing NLL with respect to free choice trials:**

**sf\_sub** = softmax prediction of informative choice probability conditioned on free choice

**choice2\_sub** = informative choice conditioned on free choice

**NLL** = -**Mean**(choice2\_sub .\* log(sf\_sub) + (1 - choice2\_sub) .\* log(1 - sf\_sub));