

Psuedo code

Phase 1: ELM learning of original dataset

$[w_{inp_1}, w_{out_1}] = elm(h_1, t_1)$
 $free(h_1, t_1)$; *Original data is forgotten*

Phase 2: Generative Inertial learning

$h_g = rand([size_of_input, size_of_inertia])$; *Generate randomized inertia*
 $t_g = dot(feed_forward(w_{inp_1}, h_g), w_{out_1})$; *Calculate output inertia*
 $h_i = h_i.append(h_g)$
 $t_i = t_i.append(t_g)$; *Append inertia to incoming dataset*
 $[w_{inp_i}, w_{out_i}] = elm(h_i, t_i)$; *Learn new model from combined dataset*
 $free(h_i, t_i)$; *Purge the current dataset; Ready for incoming data*

Remark 1. *This algorithm should work with not only ELM but also any batch learning algorithm*

Remark 2. *The chunk size of the incoming data is irrelevant with the ability of the algorithm to derive correct estimation of the target function need prove*