Using genetic optimizer on method 4

```
num_generations = 100 # Number of generations.
num_parents_mating = 5 # Number of solutions to be selected as parents in the mating pool.
initial_population = keras_ga.population_weights # Initial population of network weights.

#Different Err func to work with the genetic optimizer
def errfunc(param, sol_idx):
    global data_inputs, data_outputs, keras_ga, globalModel

    predictions = pygad.kerasga.predict(model=globalModel,
                                        solution=param,
                                        data=data_inputs)
    mae = tensorflow.keras.losses.MeanAbsoluteError()
    abs_error = mae(data_outputs, predictions).numpy() + 0.00000001
    solution_fitness = 1.0/abs_error
    return solution_fitness
# MSE/MAE^2
def callback_generation(ga_instance):
    print("Generation = {generation}".format(generation=ga_instance.generations_completed))
    print("Fitness    = {fitness}".format(fitness=ga_instance.best_solution()[1]))

keras_ga = pygad.kerasga.KerasGA(model=globalModel,#Does the genetic optimization over existing Method 4 model
                                 num_solutions=10)

ga_instance = pygad.GA(num_generations=num_generations,
                       num_parents_mating=num_parents_mating,
                       initial_population=initial_population,
                       fitness_func=errfunc,
                       on_generation=callback_generation)
```