TRAINING A MULTILAYER PERCEPTRON TO PREDICT A CAR SPEED IN A SIMULATOR

COMPARING RPROP, PSO, BFGS AND A MEMETIC PSO-BFGS HYBRID

Artur Henrique Gonçalves Coutinho Alves

Vinícius Veloso de Melo, PhD.

Federal University of São Paulo – Institute of Science and Technology

São José dos Campos – SP – Brazil

SBGames 2016

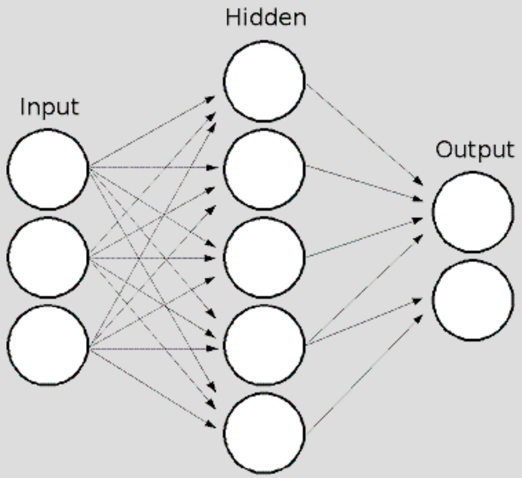
CONTEXT
PREDICTIVE CAR CONTROL IN A RACE

- The Open Racing Car Simulator (TORCS)
- Simulated Car Racing Championship
- Model-based predictive control (MPC)
- State estimation
- Approximation of physical behavior of a simulated car

In-game screenshot

NEURAL NETWORKS
TRAINING METHODS

- Feedforward Multilayer Perceptron (MLP)
- Backpropagation
- RPROP
- Bio-inspired computing
  - Swarm Intelligence



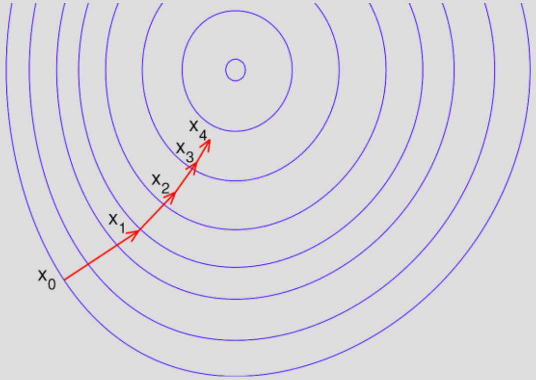
MLP example
(opencv.org)

NEURAL NETWORKS
PARTICLE SWARM OPTIMIZATION

- Flock behavior
- Particles represent sets of synaptic weights
- Particles explore the search space by “flying” over it
  - Velocity determined by personal and populational best positions
- Switches from global to local search
  - Not as effective after approaching local minima

NEURAL NETWORKS
HYBRIDIZATION

- Distinct methods have different pros and cons
- Combination of methods increases complexity
  - Successful hybridizations offer sufficiently increased quality as well
- Memetic approach
  - Population-based global search method combined with a local search method
  - PSO with quasi-Newton method BFGS – two approaches:
    - Full run of PSO followed by full run of BFGS
    - PSO and BFGS alternate after a number of iterations

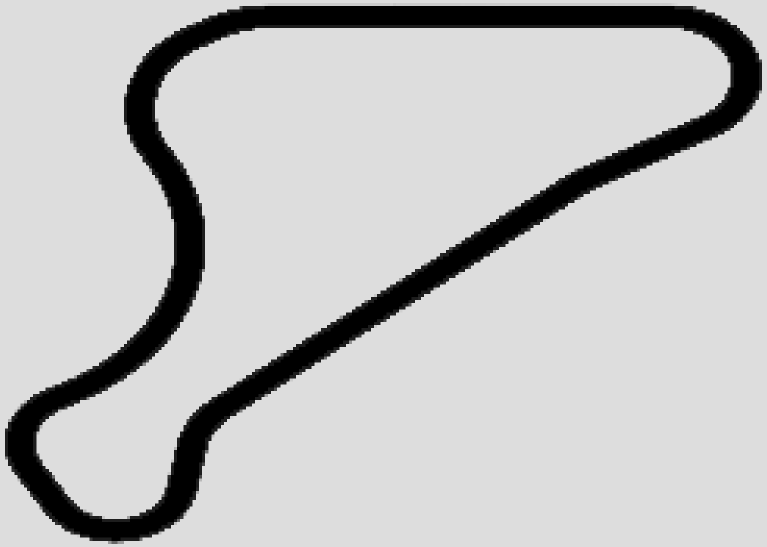


Local search by gradient descent

EXPERIMENT
DATA

- *SnakeOil* framework
- Data abstraction
  - Sensors: current longitudinal and transversal speeds
  - Actuators: throttle, brake, and steering wheel
- Model must predict variation of X and Y speeds for $t + 1$
- Simulator works at 50Hz
  - Data set subsampled to 5Hz to reduce correlation between current and future speeds

EXPERIMENT
TRACKS AND TERRAINS



cgspeedway1
Tarmac
No elevations

Dirt 3
Dirt
Various jump sections

EXPERIMENT
CODING

- R programming language
- Neural network and RPROP from *neuralnet* package
  - Uses error derivative threshold as stopping criterion
- BFGS from *optim* function
- PSO and variants from scratch
  - Canonical PSO
  - Canonical PSO with population restart
  - Memetic PSO (MPSO)
  - Memetic PSO with Return (MPSOR)

EXPERIMENT SETTINGS

- Neural network and RPROP configuration
  - RPROP as baseline for best configuration and stopping criterion
    - Stopping criteria for all other methods
      - Best RPROP validation error (with derivative threshold 0.01)
      - Maximum number of iterations
  - Two linear output neurons (X and Y speeds)
  - Two logistic hidden neurons

EXPERIMENT
STEPS

- 30 independent runs of RPROP
  - Best validation error set as stopping criterion for other methods
- 30 independent runs of other methods
  - BFGS (1000 iterations through *optim*)
  - PSO & PSOr (5000 iterations)
  - MPSO (500 PSO + 1000 BFGS iterations)
  - MPSOR (500 PSO with 100 BFGS for each 50 PSO iterations without improvement)

EXPERIMENT RESULTS

- Most differences were significative for the Wilcoxon rank sum test ($\alpha = 5\%$)
- Detailed results are available in the paper

Table 3: Test errors in *cgspeedway1*. PSO with restart (PSOr), Memetic PSO (MPSO), and Memetic PSO with Return (MPSOR).

	RPROP	BFGS	PSO	PSOr	MPSO	MPSOR
Average	0.003996	0.263675	0.110572	0.023097	**0.003845**	0.003951
Std. Dev.	0.000764	0.319624	0.054253	0.021826	0.000823	**0.000662**
Median	0.004071	0.255030	0.094894	0.012584	**0.003704**	0.003856

Table 4: Training times (in seconds) in *cgspeedway1*.

	RPROP	BFGS	PSO	PSOr	MPSO	MPSOR
Average	**32.69**	110.15	310.46	306.75	92.31	90.41
Std. Dev.	7.98	19.69	**3.53**	4.95	16.97	25.66
Median	**33.57**	119.93	311.20	308.1	90.60	94.03

EXPERIMENT RESULTS

- Most differences were significative for the Wilcoxon rank sum test ($\alpha = 5\%$)
- Detailed results are available in the paper

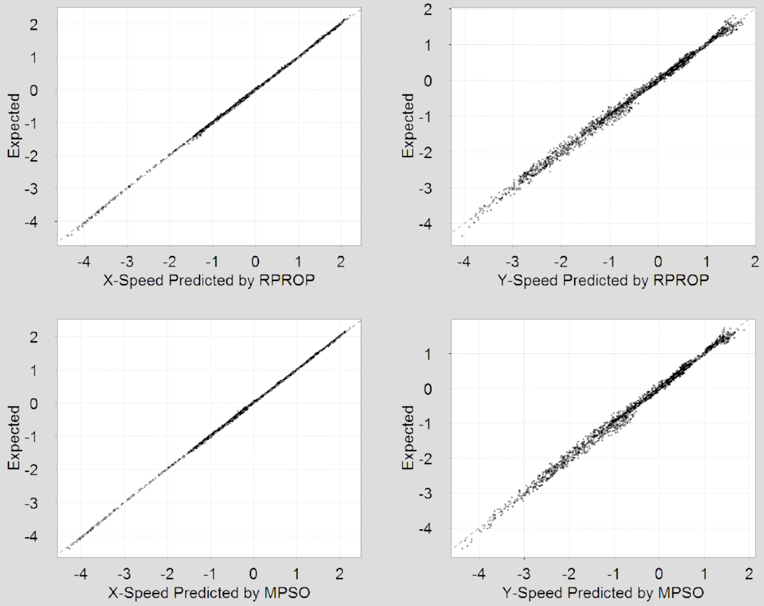
Table 5: Test errors in *Dirt 3*.

	RPROP	BFGS	PSO	PSOr	MPSO	MPSOR
Average	0.004057	0.307323	0.123548	0.027718	**0.003827**	0.100128
Std. Dev.	0.003257	0.329420	0.032478	0.029732	**0.002423**	0.2112
Median	0.002779	0.288398	0.120994	0.017652	0.003061	**0.002431**

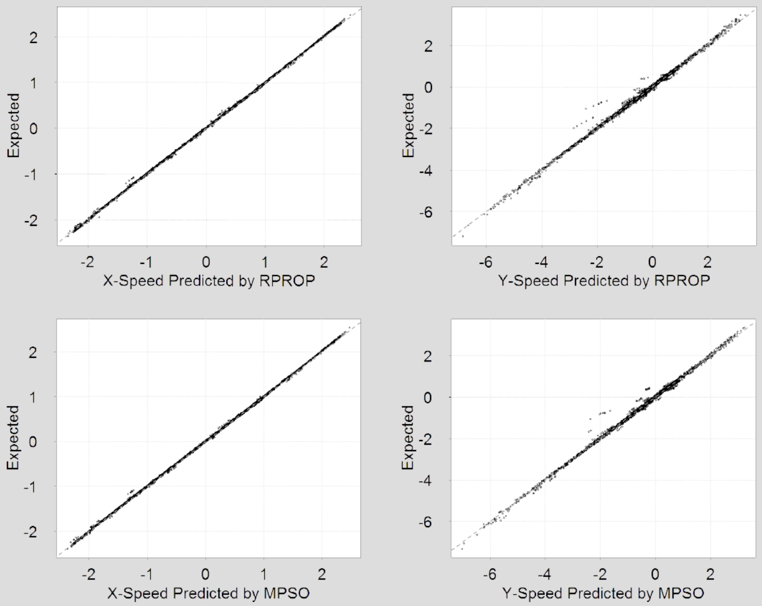
Table 6: Training times (in seconds) in *Dirt 3*.

	RPROP	BFGS	PSO	PSOr	MPSO	MPSOR
Average	**119.68**	180.62	458.54	467.56	152.48	130.55
Std. Dev.	60.72	16.70	**3.07**	4.35	25.42	47.78
Median	**103.08**	186.43	459.12	468.70	150.54	142.72

EXPERIMENT RESULTS



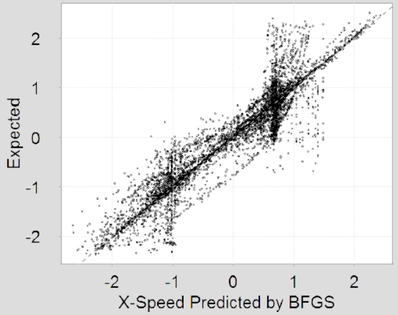
cgspeedway1

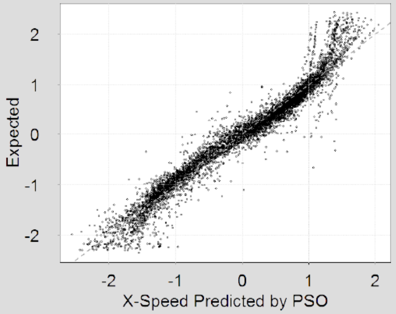


Dirt 3

EXPERIMENT RESULTS

- Prediction issues
  - Vertical lines and clusters in BFGS; may be caused by small weights and a dominance of such values in the data set
  - PSO and PSOr present curvilinear plots; may be limited outputs because of insufficient weights





Dirt 3

EXPERIMENT
CONCLUDING REMARKS

- The hybrid approach is able to compete result-wise with RPROP
  - It’s slower, but this may be improved by using a more efficient implementation in C, for example
  - It’s better and faster than its parts
- PSO never reached the main stopping criterion before reaching its iteration limit
  - Populational purge, which improved its results, was not enough
- Feedforward MLP was able to approximate the behavior of the vehicle
- Future works will focus on improving both global and local search methods and using faster programming languages

THANK YOU!

UNIFESP

UNIVERSIDADE FEDERAL DE SÃO PAULO

1933

Federal University of São Paulo

Institute of Science and Technology

This work has been supported by CNPq grant #486950/2013-1 and a CAPES scholarship

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES