

TMVA DNN vs PyKeras Benchmarking

Input parameters

Nneurons = 100

Nlayers = 3

Activation = TANH

Lastactivation = SIGMOID

Initializer = XAVIERUNIFORM

Lossfunction = CROSSENTROPY

Transformations = "N,D"

Factorystring = "!V:!Silent:Color:DrawProgressBar:Transformations=I:
AnalysisType=Classification"

Learningrate = 0.1

Momentum = 0.0

Batchsize = 128

Convergencesteps = 100

Dropout = 0.0

Ntrainsignal = 50000

Ntrainbackground = 50000

Ntestsignal = 100000

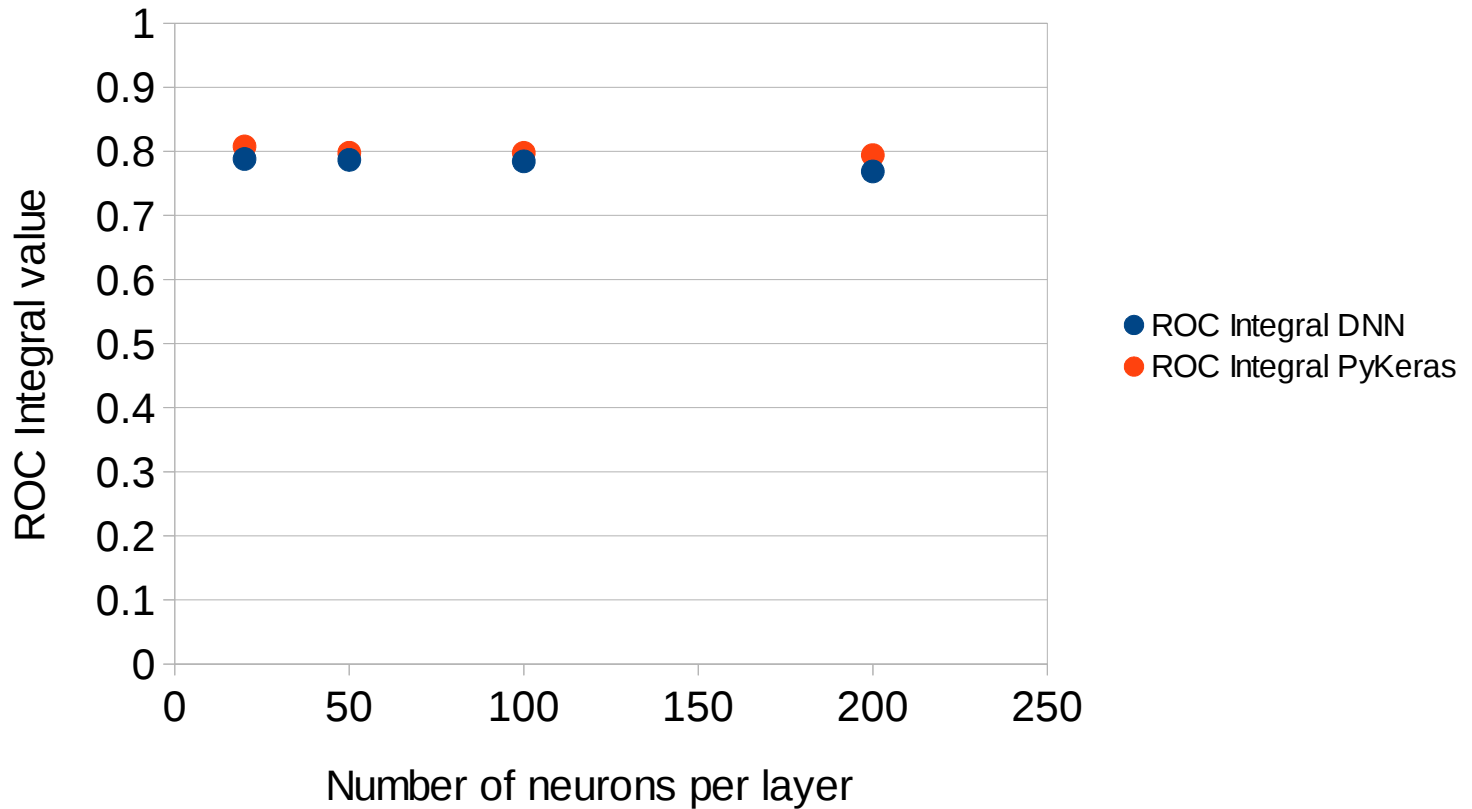
Ntestbackground = 100000

Scan over N neurons

	N	Batch size	Convergence steps	Dropout	ROC integral	CPU time	Real time
TMVAdnncpu	20	128	100	0	0.78814	27134.8	553.908
TMVAdnncpu	50	128	100	0	0.78673	24007	808.421
TMVAdnncpu	100	128	100	0	0.78442	29923.9	2075.95
TMVAdnncpu	200	128	100	0	0.76868	36029	5424.62
pykeras	20	128	100	0	0.80769	4174.37	1423.61
pykeras	50	128	100	0	0.79782	1510.73	595.66
pykeras	100	128	100	0	0.79772	2318.78	635.477
pykeras	200	128	100	0	0.79419	3074.4	574.525

Scan over N neurons

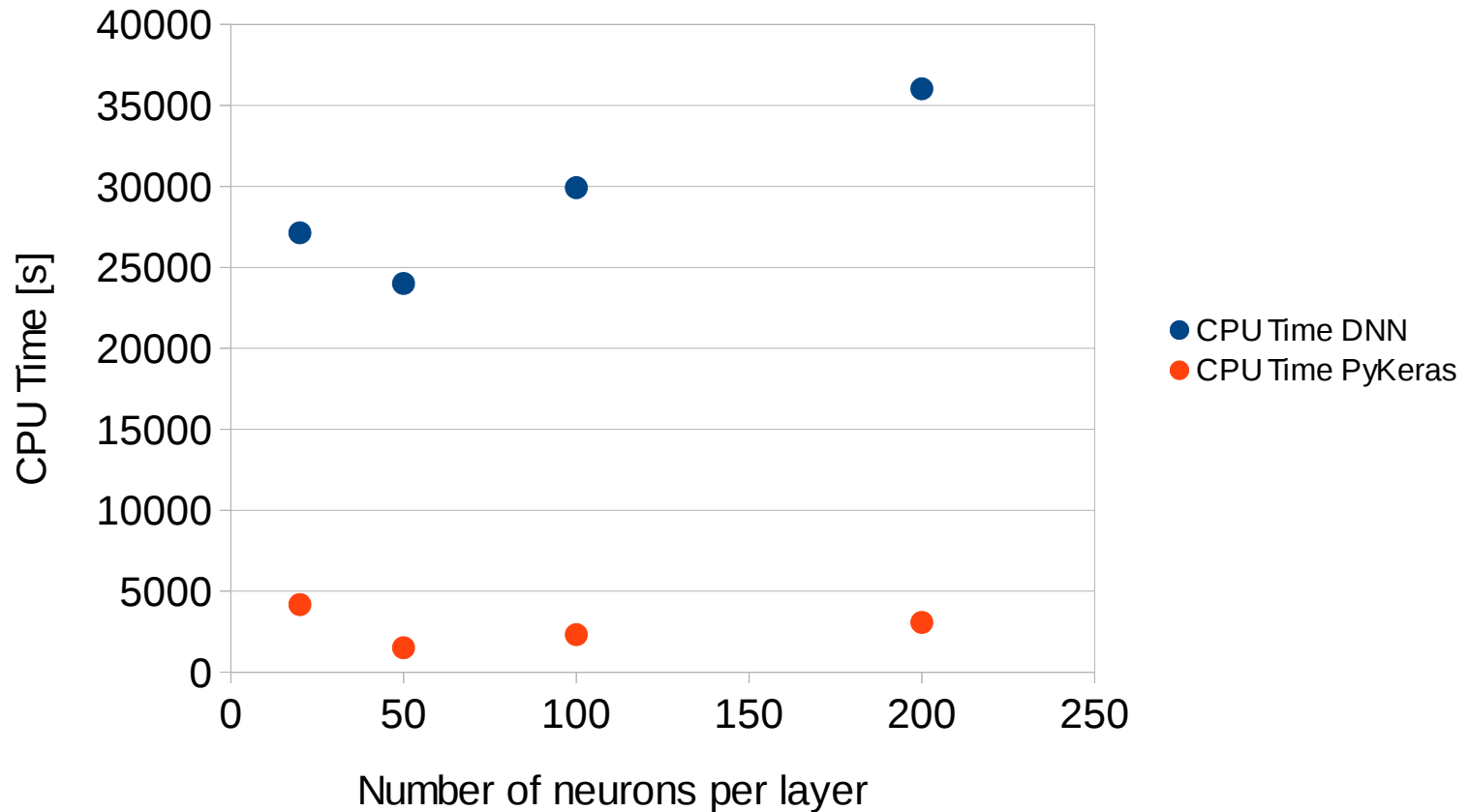
ROC Integral



Convergence steps = 100 , Batch size = 128, Dropout = 0

Scan over N neurons

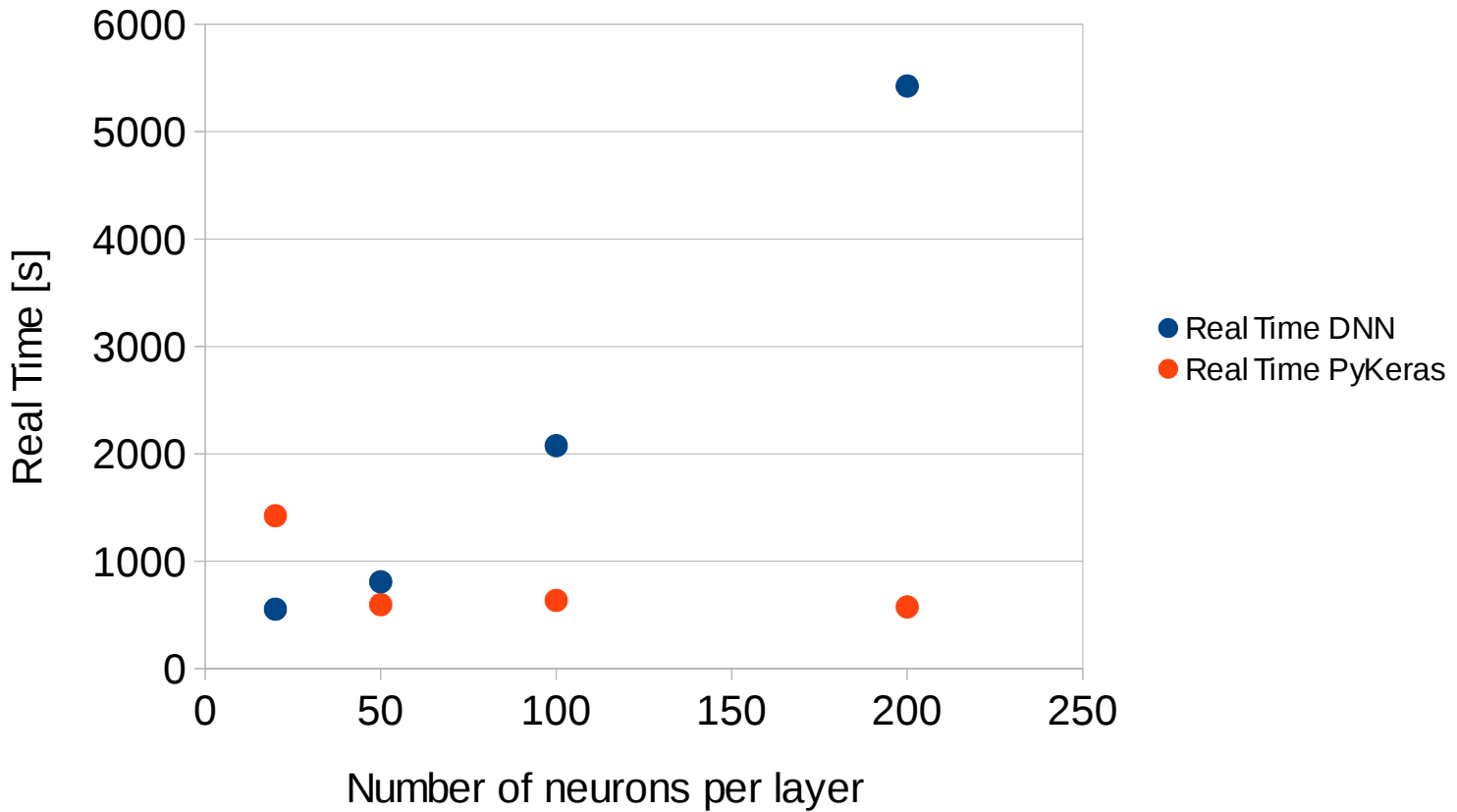
CPU Time



Convergence steps = 100 , Batch size = 128, Dropout = 0

Scan over N neurons

Real Time

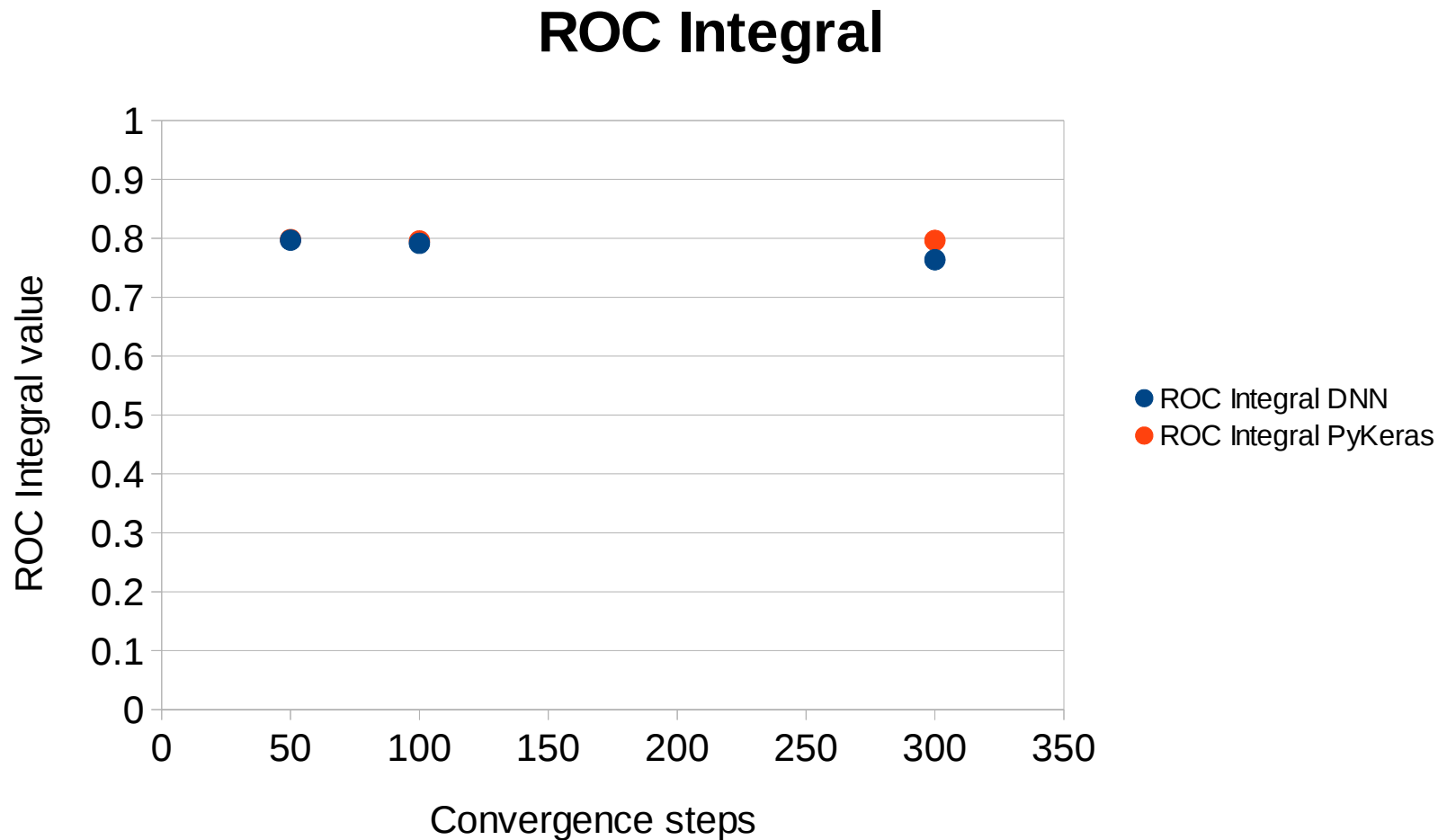


Convergence steps = 100 , Batch size = 128, Dropout = 0

Scan over Convergence steps

	N	Batch size	Convergence steps	Dropout	ROC integral	CPU time	Real time
TMVAdnncpu	100	128	50	0	0.79685	21230.2	1475.49
TMVAdnncpu	100	128	100	0	0.79141	27979.7	1945.84
TMVAdnncpu	100	128	300	0	0.76363	66933.7	4600.42
pykeras	100	128	50	0	0.7979	1441.19	397.065
pykeras	100	128	100	0	0.79556	2369.49	648.783
pykeras	100	128	300	0	0.79644	6001.73	1622.31

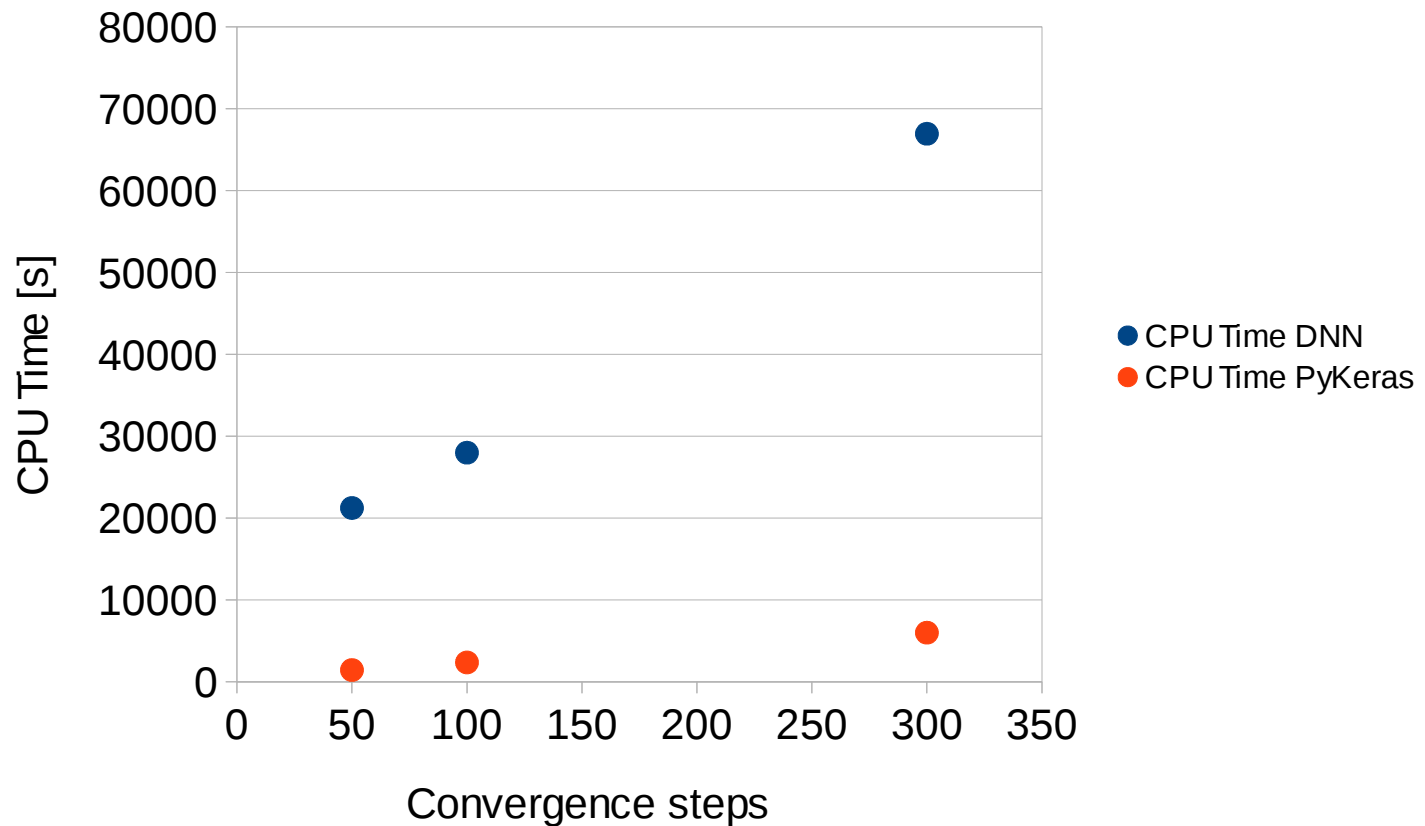
Scan over Convergence steps



N neurons = 100 , Batch size = 128, Dropout = 0

Scan over Convergence steps

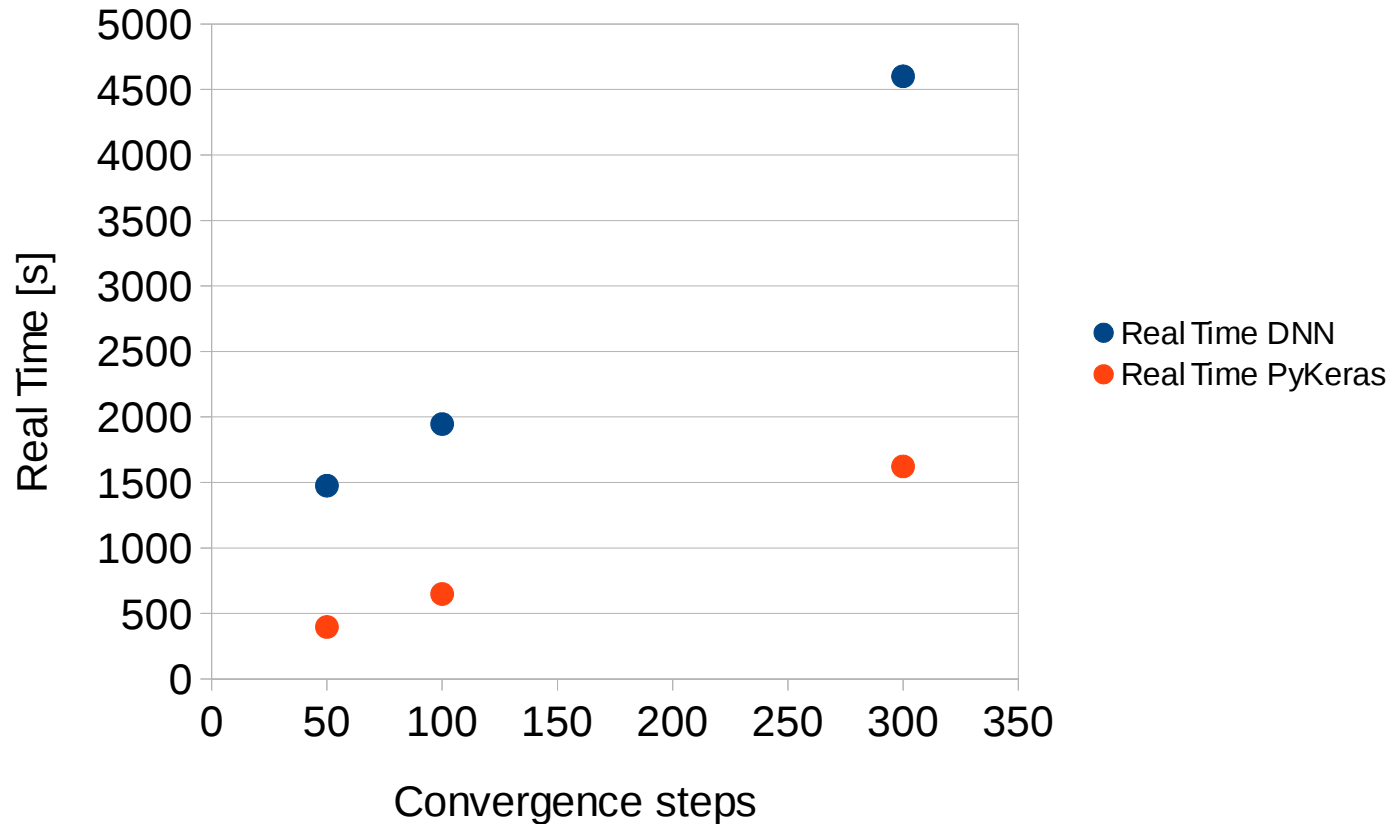
CPU Time



N neurons = 100 , Batch size = 128, Dropout = 0

Scan over Convergence steps

Real Time

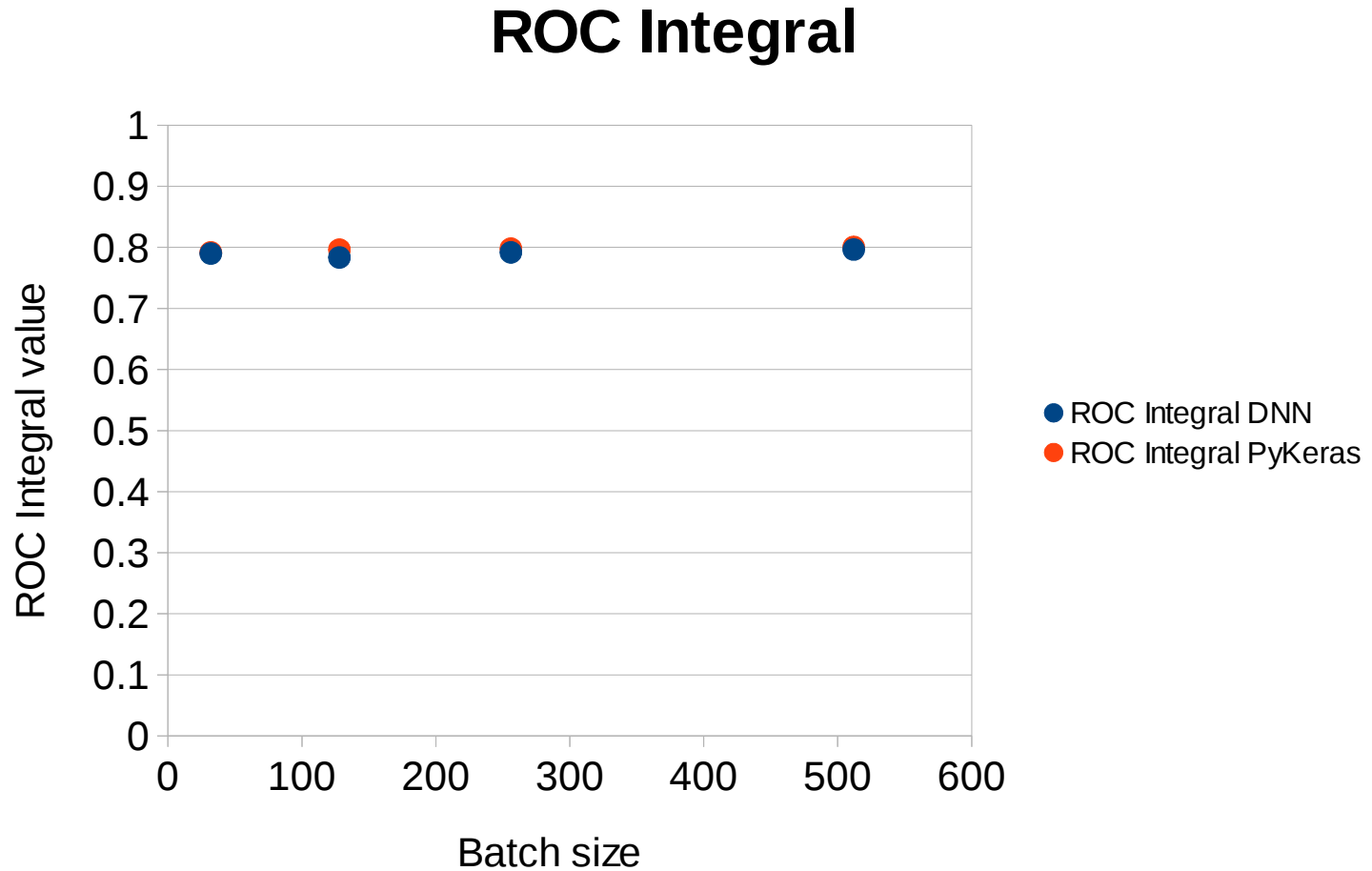


N neurons = 100 , Batch size = 128, Dropout = 0

Scan over Batch size

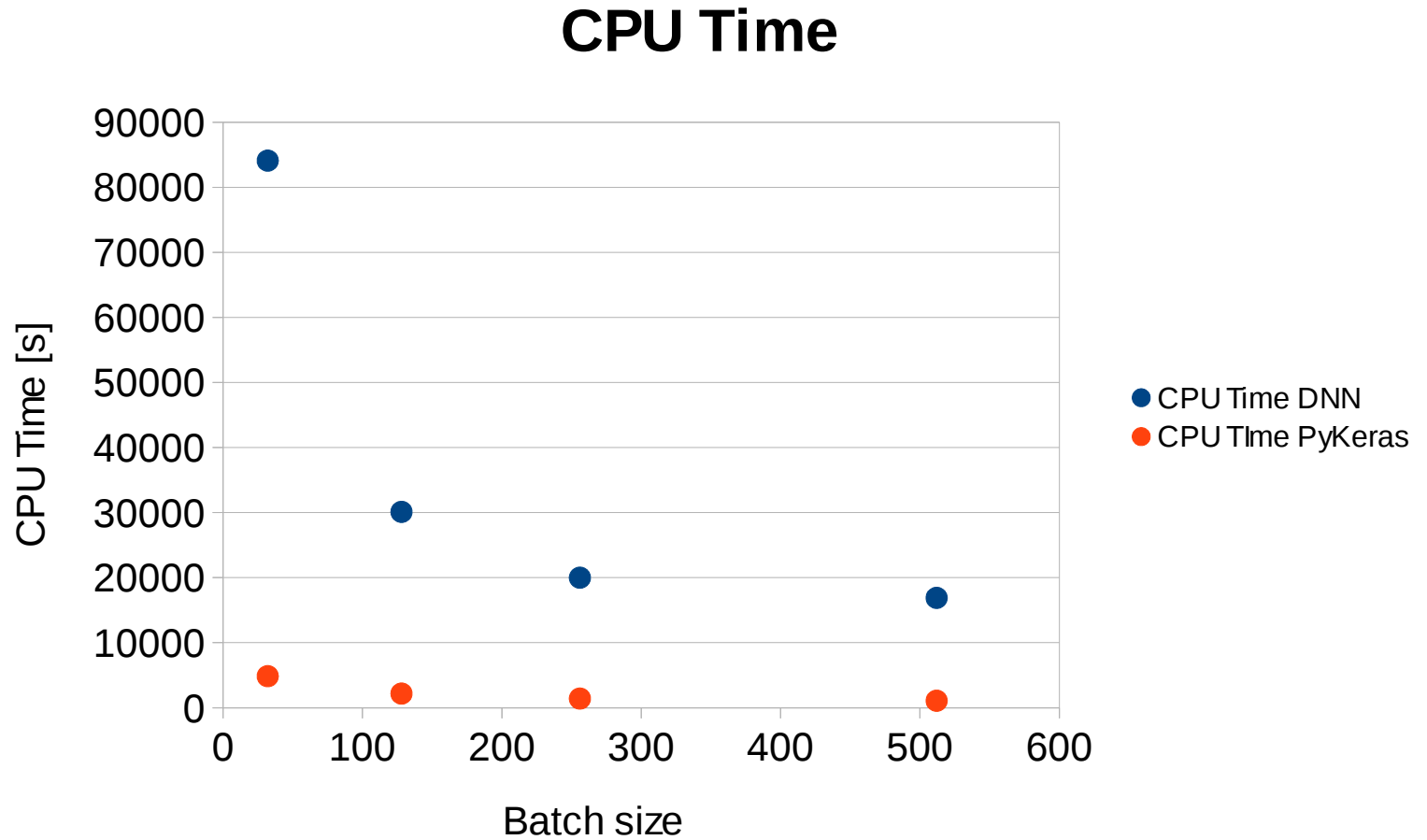
	N	Batch size	Convergence steps	Dropout	ROC integral	CPU time	Real time
TMVAdnncpu	100	32	100	0	0.78989	84098.4	2701.83
TMVAdnncpu	100	128	100	0	0.7835	30103.4	2084.95
TMVAdnncpu	100	256	100	0	0.7918	19988.1	2038.15
TMVAdnncpu	100	512	100	0	0.79648	16887.9	2367.45
pykeras	100	32	100	0	0.79178	4849.21	1629.51
pykeras	100	128	100	0	0.79631	2177.14	619.642
pykeras	100	256	100	0	0.79796	1409.67	337.002
pykeras	100	512	100	0	0.80075	1078.29	207.365

Scan over Batch size



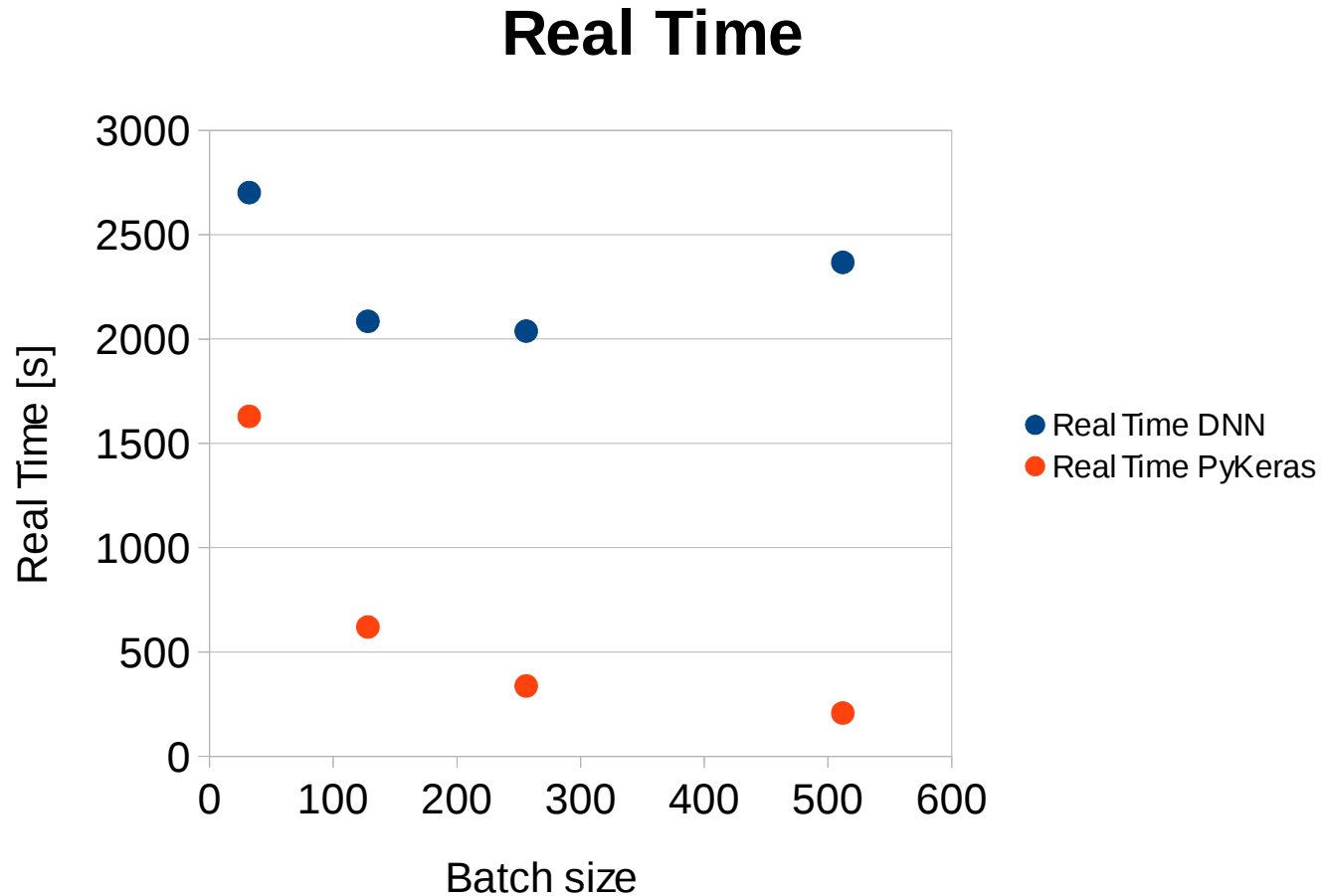
N neurons = 100 , Convergence steps = 100 , Dropout = 0

Scan over Batch size



N neurons = 100 , Convergence steps = 100 , Dropout = 0

Scan over Batch size

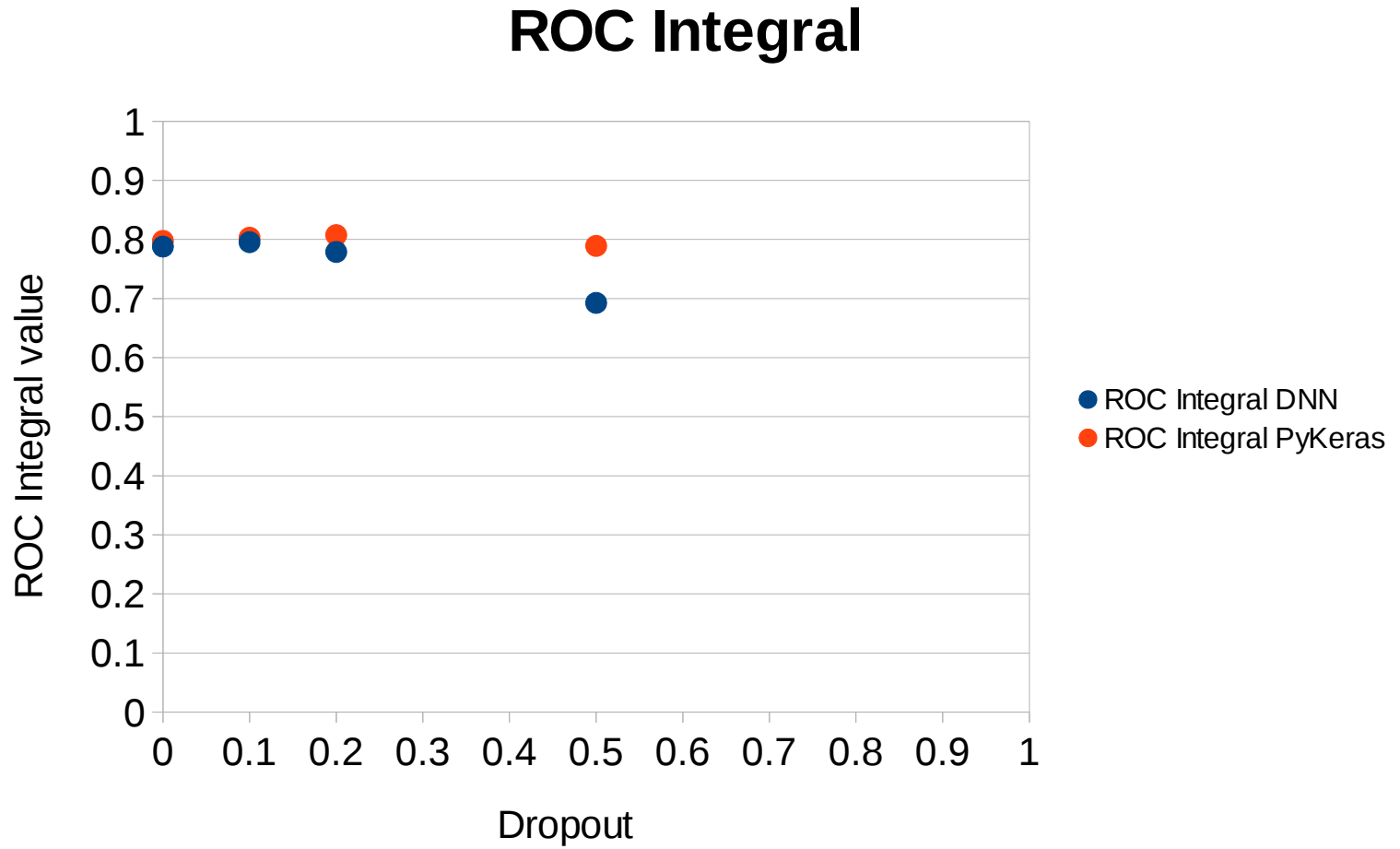


N neurons = 100 , Convergence steps = 100 , Dropout = 0

Scan over Dropout

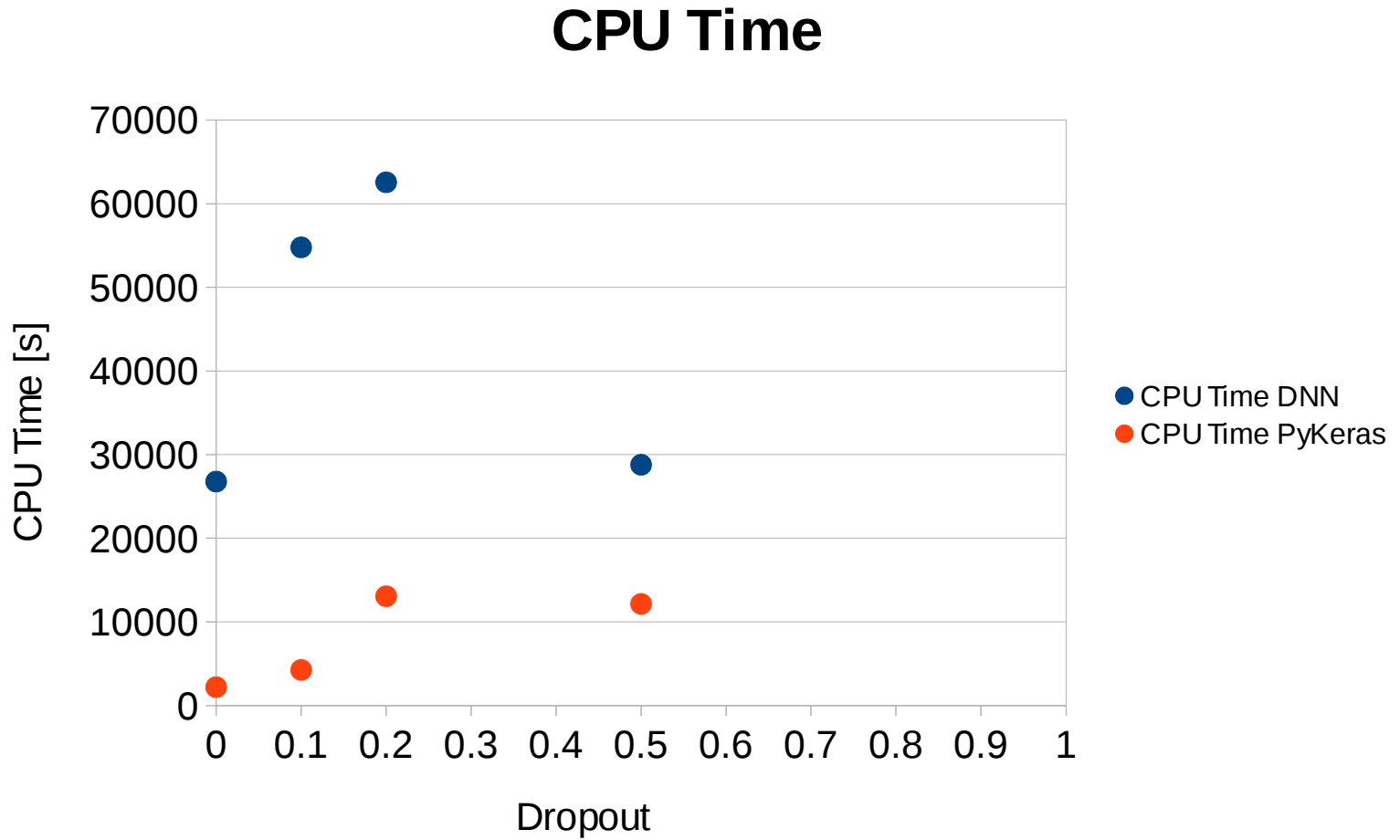
	N	Batch size	Convergence steps	Dropout	ROC integral	CPU time	Real time
TMVAdnncpu	100	128	100	0	0.78795	26770.3	1837.87
TMVAdnncpu	100	128	100	0.1	0.79546	54772.9	3548.32
TMVAdnncpu	100	128	100	0.2	0.77889	62546.3	4032.41
TMVAdnncpu	100	128	100	0.5	0.69264	28777.6	1862.06
pykeras	100	128	100	0	0.79759	2195.14	599.519
pykeras	100	128	100	0.1	0.80295	4252.65	1149.3
pykeras	100	128	100	0.2	0.80738	13060.7	3500.09
pykeras	100	128	100	0.5	0.78908	12143.4	3252.2

Scan over Dropout probability



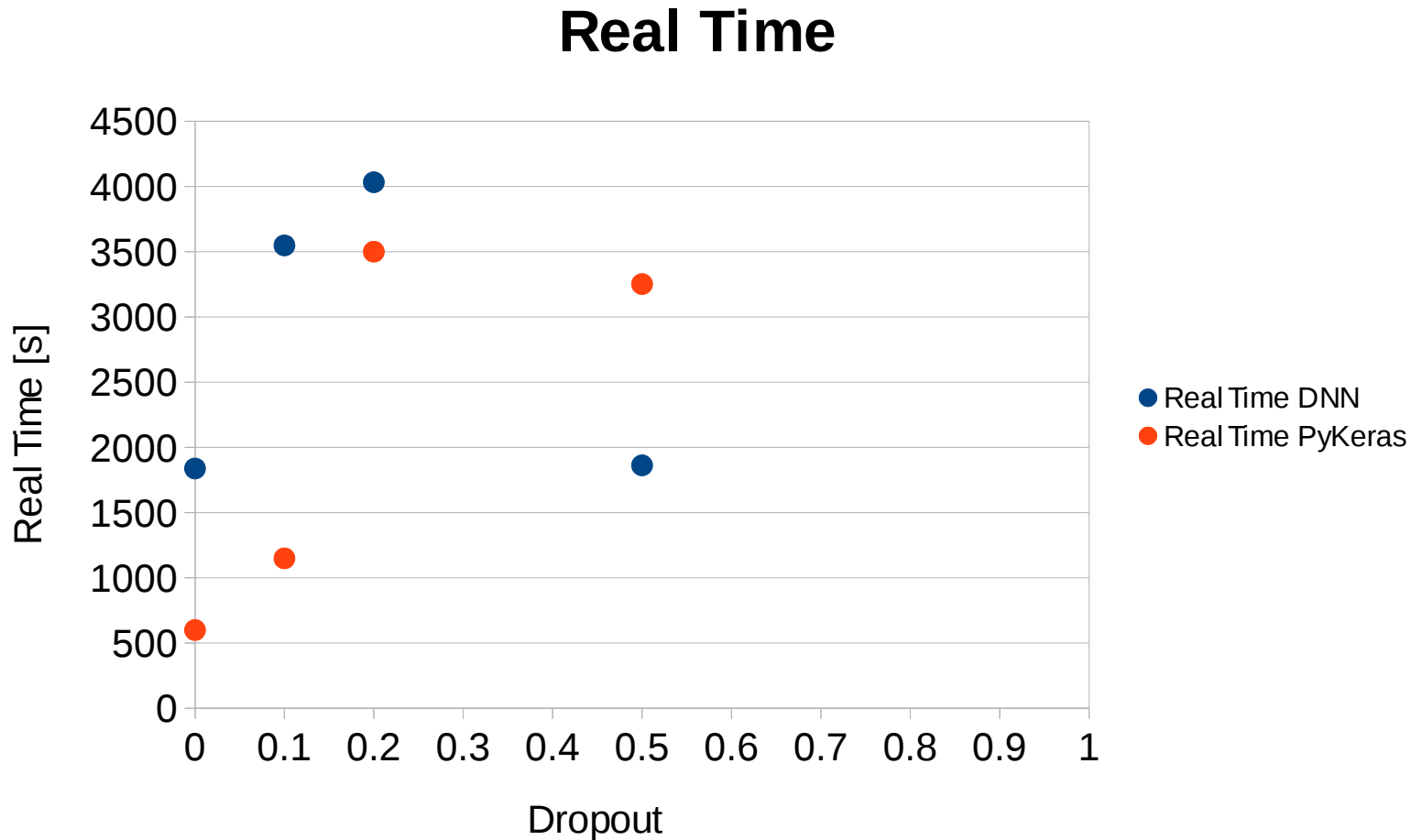
N neurons = 100 , Convergence steps = 100 , Batch size = 128

Scan over Dropout probability



N neurons = 100 , Convergence steps = 100 , Batch size = 128

Scan over Dropout probability



N neurons = 100 , Convergence steps = 100 , Batch size = 128

Further steps

Run a stable configuration also in native Keras

Run on GPU

Test also new neural network implementation