Neural Network Class Variables and Class Functions

	Class Variable	Type	Description
1	M	scalar	number of input nodes
2	N	scalar	number of hidden nodes
3	P	scalar	number of output nodes
4	num_patterns	scalar	number of input patterns
5	num_targets	scalar	number of targets - should equal num_patterns (checked)
6	nn_outputs	array - len num_targets	array of outputs from trained neural network to compare to targets
7	beta	scalar	activation function parameter
8	eta	scalar	learning rate
9	max_iterations	scalar	maximum number of iterations of backpropagation algorithm
10	typical_x	scalar	used for convergence test, typical value in weight matrix
11	typical_f	scalar	used fot convergence test, typical value of error function
12	gradient_tolerance	scalar	changes in relative gradient less than this value indicate convergence (code uses infinity norm over gradient)
13	step_tolerance	scalar	change in relative value of weights less than this value indicate convergence (uses infinity norm)
14	function_tolerance	scalar	change in error function less than this value indicate convergenc
15	inputnodes	array - len M+1	array of neural network input nodes of length M+1 - plus one for bias node
16	hiddennodes	array - len N+1	array of hidden nodes of length N+1 - plus one for bias node
17	ghiddennodes	array - len N+1	array of hidden nodes after application of activation function - activation function is not applied to the bias node
18	outputnodes	array - len P	array of output nodes of length P
19	goutputnodes	array - len P	array of output nodes after application of activation function | targets | array of targets for a pattern of length P
20	errors	array - len P	difference between targets and neural net output
21	square_errors	array - len P	square of errors
22	sum_square_errors	scalar	sum of square errors for all output nodes
23	total_error	scalar	total square error for all patterns
24	total_error_previous	scalar	total_error from previous iteration of backpropagation
25	change_in_error	scalar	change in error
26	w1	matrix - dim N x (M+1)	input to hidden neural network weight matrix
27	w2	matrix - dim P x (N+1)	hidden to output neural network weight matrix
28	deltaoutput	array - len P	deltas at output layer
29	deltahidden	array - len N+1	deltas at hidden layer
30	dw1	matrix - dim N x (M+1)	gradient matrix for w1 weights
31	dw2	matrix - dim P x (N+1)	gradient matrix for w2 weights
32	tmpw1	matrix - dim N x (M+1)	temporary w1 matrix for relative gradient and step calculations
33	tmpw2	matrix - dim P x (N+1)	temporary w2 matrix for relative gradient and step calculations
34	wk1	matrix - dim N x (M+1)	w1 weights from previous iteration, used for convergence testing
35	wk2	matrix - dim P x (N+1)	w2 weights from previous iteration, used for convergence testing
36	deltaw1	matrix - dim N x (M+1)	change in w1 weights, used for convergence testing
37	deltaw2	matrix - dim P x (N+1)	change in w2 weights, used for convergence testing
38	converged	boolean	used for convergence testing
39	interation_count	scalar	counts number of iterations of backpropagation algorithm

	Function	Description	Input	Output
1	copy_pattern_target_to_to_inputs	copies pattern to input nodes, copies target for subsequent error computations	input_patterns, input_targets	inputnodes, targets
2	copy_nnoutput_to_output	copies neural network output to nn_outputs for comparison to targets	goutput	nn_outputs
3	w1_matrix_vector_multiplication	computes w1*inputnodes	inputnodes	hiddennodes
4	hidden_activation	applies activation function to hidden nodes	ghiddennodes	hiddennodes
5	w2_matrix_vector_multiplication	computes w2*ghiddennodes	outputnodes	ghiddennodes
6	output_activation	applies activation function to output nodes	goutputnodes	outputnodes
7	activation	computes activation function used in hidden_actiovation and output_activation	a value, beta	returns computed result
8	compute_errors	computes difference between targets and nn output	target, goutputnodes	errors
9	compute_square_errors	computes square of errors at output nodes	square_errors	errors
10	compute_sum_square_errors	computes sum of squares over all output nodes	square_errors	sum_square_errors
11	accumulate_total_error	sums square error over all patterns	sum_square_errors	total_error
12	compute_output_deltas	computes deltas at output nodes	beta, goutputnodes, errors	deltaoutput
13	compute_hidden_deltas	computes deltas at hidden nodes	w2, deltaoutput, ghiddennodes	deltahidden
14	accumulate_gradient	computes gradient summed over all patterns	deltaoutput, ghiddenhodes, deltahidden, inputnodes	dw1, dw2
15	total_sq_error	performs steps above to compute error over all patterns		total_error
16	total_sq_error_and_gradient	performs steps above to compute error and gradient		total_error, dw1, dw2
17	compute_nn_outputs	computes neural network outputs for each pattern		nn_outputs
18	update_weights	computes new weights	w1, w2, eta, dw1, dw2	wi, w2
19	backup_weights	copies weights for testing convergence	w1, w2	wk1, wk2
20	back_up_total_error	copies total_error for testing convergence	total_error	total_error_previous
21	compute_change_in_weights	computes the change in weight values with iteration for testing convergence	w1, wk1, w2, wk2	deltaw1, deltaw2
22	compute_change_in_error	compute change in error with iteration for testing convergence	total_error, self.total_error_previous	change_in_error
23	init_weights_random	initializes nerual network weights randomly on a given interval	min, max	w1, w2
24	init_weights_one	initializes neural network weights to one, for testing		w1, w2
25	print_array	prints an array with a label, for testing	arrayname, label	
26	print_weight_matrix	print the values in a weight matrixs with a label, for testing	weight matrix name, label	
27	test_convergence	test relative gradient, relative step size, relative function value, and max iterations for testing convergence	tolerances, iteration_count	converged
28	rel_gradient	computes relative gradient for test_convergence function		relgrad
29	rel_step	computes relative step size in in weight change for test_convergence function		relstep