Neural Network documentation

neuralNet module contains 4 classes, baseNetwork and a trainNetwork method.

NeuralNetwork class

+ *NeuralNetwork(dict networkRepresentation)*

    Constructor. Takes dictionary representing the neural network topology as an argument.

+ *getRepre()*

    Returns representation of the neural network topology as a dictionary.

+ *getCompactRepre()*

    Returns compact representation of the neural network topology as a dictionary. Attributes with default values are omitted.

+ *getLayers()*

    Returns list of **Layer** objects in the neural network.

+ *getLearnParam()*

    Returns learning parameter (gamma) specified in the settings of the network.

+ *sigmoid(float* x)

    Calculates the sigmoid function for x.

+ *dsigmoid(float* x)

    Calculates the sigmoid derivative for x.

+ *dtanh(float* x)

    Calculates the tanh for x.

+ *activate(float* x)

    Returns a result of activational function with x as an input.

+ *derivate(float* x)

    Returns a result of derivative of an activational function with x as an input.

+ *out(float* x)

    Returns a result of an output function. (returns x).

+ *eval(list* inputs)

    Calculates the output of the network for specific inputs. Outputs are returned as a list.

+ *calcErrors(list* result, list expected)

    Calculates and sets the errors for every neuron in the network based on the results and expected output. Returns nothing (used internally).

+ *train(list* trainingSet)

Calculates the errors and adjusts weights based on the training set. Training set is a list of rows, where every row contains 2 lists: inputs and outputs.

Example of a training set: [[[1, 1], [1]], [[1, 0], [1]], [[0, 1], [1]], [[0, 0], [0]]]

Layer class

+ *Layer(object network, list layerRepre)*

Constructor. Object **network** is the NeuralNetwork instance the layer is part of. **LayerRepre** is a list of dictionaries, where every dictionary is a neuron with specific attributes.

+ *getRepre(), getCompactRepre()*

Analogous to NeuralNetwork functions.

+ *getNetwork()*

Returns the instance of a network the layer is part of.

+ *getNeurons()*

Returns list of **Neuron** objects the layer contains.

+ *eval(list outputs)*

Calculates the output of the neurons in the layer based on the outputs of the previous layer. Outputs are returned as a list.

+ *propagateError()*

Function backpropagates the error on a layer to neurons connected to the layer with synapses.

+ *modifyWeights()*

Function modifies weights of connections on neurons (or biases) based on the error on neurons.

Constant class

+ *Constant(dict constRepre)*

Creates an input with constant value and a name specified in **constRepre** dictionary.

+ *getRepre(), getCompactRepre()*

Analogous functions.

+ *setName(str name), getName()*

Sets the name of a neuron to a specific string, or returns it as a string.

+ *getValue()*

Returns the constant value of Constant input.

Variable(Constant) class

Subclass of Constant. Provides a way for neural network to specify it's input.

+ *setValue(float value)*

Sets the value on the input to a **value** rounded to 5 digits.

Neuron class

+ *Neuron(object layer, dict neuronRepre)*

Constructor. Creates a neuron in a **layer** with attributes specified in the **neuronRepre** dictionary.

+ *getRepre(), getCompactRepre()*

Analogous functions.

+ *getLayer()*

Returns the instance of a **Layer** the neuron is part of.

+ *setName(str name), getName()*

Analogous functions to **Layer**.

+ *setSynapses(dict synapses), getSynapses()*

Returns a dictionary of neuron connection or connects a neuron to a neurons in specific layers with a weight based on the key:value attributes in the dictionary.

Example: setSynapses({“L0N0”:0.75}) – connects the neuron to the neuron 0 in layer 0 with a weight of 0.75

+ *setThreshold(float threshold), getThreshold()*

Sets or gets the threshold of the neuron. Threshold is added to the input of the neuron during evaluation.

+ *setBias(dict bias), getBias()*

Similar to synapses. Returns a dictionary of bias connections or connects a bias to the neuron with specific value and a weight.

Example: setBias({1:null}) – connects a bias with a value of 1 and a constant weight of 1 (null) to the neuron

+ *setInput(float value), getInput()*

Sets or gets the input to the neuron. Used internally during evaluation.

+ *setValue(float value), getValue()*

Sets or gets the output of the neuron. Used internally during evaluation.

+ *setError(float error), getError()*

Sets or gets the error on the neuron. Used internally during evaluation.

Module functions

+ *baseNetwork()*

Returns a dictionary representing a base network with 2 inputs, 2 hidden layers (3 & 2 neurons respectivelly) and one output. The network uses sigmoid function with alpha = 0.5.

+ *trainNetwork(object network, str setPath, int epochs = 5)*

Trains a **network** with training set located at **setPath**. Repeats for n-**epochs** (default = 5)