**using** NeuralLibrary.Neurons;

**namespace** NeuralLibrary

{

/// <summary>

/// The connection held between two neurons with a given

/// weight.

/// </summary>

**public** **class** Connection

{

/// <summary>

/// Initializes the connection.

/// </summary>

**public** Connection(**double** weightInitial,

Neuron anteriorNeuron, Neuron posteriorNeuron)

{

**this**.Weight = weightInitial;

**this**.AnteriorNeuron = anteriorNeuron;

**this**.PosteriorNeuron = posteriorNeuron;

}

/// <summary>

/// Feeds the product of output from the anterior

/// neuron and the weight of the connection forward

/// to the anterior neuron.

/// </summary>

**public** **void** FeedForward()

{

PosteriorNeuron.Net +=

AnteriorNeuron.Output \* Weight;

}

#region Fields

/// <summary>

/// The last delta weight (used for momentum)

/// </summary>

**protected** **double** lastDeltaWeight = 0;

#endregion Fields

#region Properties

/// <summary>

/// The anterior neuron within the connection.

/// </summary>

**public** Neuron AnteriorNeuron { **protected** set; get; }

/// <summary>

/// The posterior neuron within the connection.

/// </summary>

**public** Neuron PosteriorNeuron { **protected** set; get; }

/// <summary>

/// Updates the weight of the connection using the

/// weight update rule. dW = ERROR\_posterior \*

/// OUTPUT\_anterior

/// </summary>

**public** **virtual** **void** UpdateWeight(**double** learningRate,

**double** momentum)

{

**double** deltaWeight = -(Gradient \* learningRate) +

momentum \* lastDeltaWeight;

Weight += deltaWeight;

lastDeltaWeight = deltaWeight;

}

/// <summary>

/// Gets the gradient of the connection,

/// </summary>

**public** **double** Gradient

{

get

{

**double** output = 0;

**if** (AnteriorNeuron **is** BiasNeuron)

output =

(AnteriorNeuron **as** BiasNeuron).Output;

**else** **if** (AnteriorNeuron **is** InputNeuron)

output =

(AnteriorNeuron **as** InputNeuron).Output;

**else**

output = AnteriorNeuron.Output;

**return** PosteriorNeuron.Error \* output;

}

}

/// <summary>

/// The weight associated with a connection.

/// </summary>

**public** **double** Weight { set; get; }

#endregion Properties

}

}