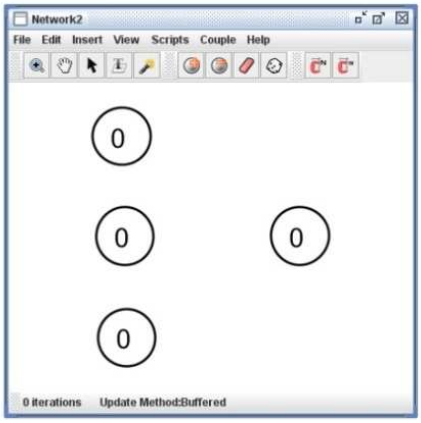
Net input



Now let us design an assembly of three neurons feeding an output neuron.

- For this go to **File --> Clear Workspace** and **Don't save changes**.
- Now click on the **network symbol** in the toolbar to open a fresh network window.
- Add four neurons by clicking four times on
- Arrange them according to this schema.
- Select the three neurons on the left side and press **1**, to make them source neurons.
- Now select the fourth (output neuron) and press 2, to connect all source neurons to this target.
- Press **N**, to select all neurons and right click **Edit 4 selected neurons**. Change the **upper bound** to **10**, the **lower bound** to **-10** and the **adjustment increment** to **1**.

Setting up the weights

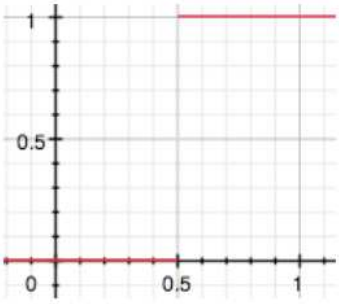
- Double click each synapse independantly. Assign the **strengths** to be **-1, 1** and **1** from top to bottom.
- Try to **iterate** the network. Why is there no output activation?
- Set the activations of the input neurons to **-1, -4** and **5** respectively by selecting each one of them and pressing **up/ down** accordingly
- What net input will this amount to? **Iterate** once to see the output.
- Now set the **bias** of the output neuron to **1** by double clicking it. Using the same activation pattern, what will the net input be now? **Iterate** once to see the output.
- Now set the activations all to **0**? What output can be expected?

Activation functions

An activation function is a method or updating rule for the activation of a node. Simbrain offers very different activation functions like linear, logistic, random, etc. (Double-click one neuron and check the drop-down box named **Update rule**).

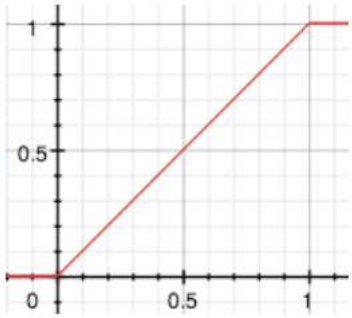
Binary activation functions

These were inspired by the function of actual neurons, which operate in a discrete (on-off) fashion. They either fire or don't fire an action potential.



- Double-click on the output neuron and make it to be **binary** with **upper bound 1** and **lower bound 0**.
- Set the threshold **θ** to be **0.5** and the bias to be **0**. When the net input exceeds the threshold, the output neuron will fire (i.e. be **1**).
- What will the output activation be now for the examples from before
  - **-1, -4, 5**
  - **-1, -4, 5** with bias **1** **(*)**
  - **0, 0, 0** without bias

Linear activation functions

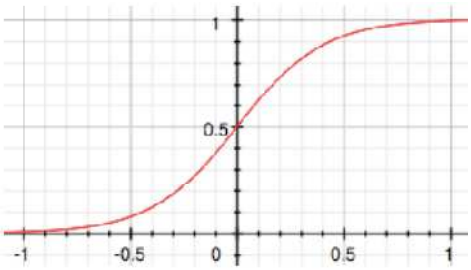


Linear activation functions compute activation as a linear function of net input. Thus net input is multiplied by a factor **m** corresponding to the slope of the linear function:

Often we want to impose limits on activation, to disallow certain very large or small activation values. That is, we "**clip**" the function at upper and lower bounds. If net input is greater than an upper bound, then activation is set to that upper bound. If net input is less than a lower bound, then activation is set to that lower bound. In *simbrain* clipping is turned on by default (to change it, go to the neuron dialog and adjust the fields upper and lower bound).

- Re-set the output neuron to have a **linear activation** function, where **m= 1**, **clipped** at **-2.5** and **2.5** respectively. What will the output activation be for activation of the known examples (*)?
- Was any result "clipped"? (What can you do to see this?)
- Now set the **slope** to be **-1**. What will the output activation be for (*)?
- Do the same now with **slope 0.5.**

Sigmoidal activation functions



"Sigmoidal" means S-shaped, and indeed the graph of this function is shaped like a stretched out "S". Sigmoidal activation functions are smooth function, which makes them desirable for certain purposes where it is useful to take the slope of the function. Assuming positive slope, as the net input of a sigmoidal grows the activation gets closer and closer to an upper asymptote, and as the net input diminishes the activation gets closer and closer to a lower asymptote. For net input near **0** the activation will be right around the middle of the upper and lower asymptotes. The activation of a sigmoidal changes rapidly around **0** – how rapidly depends on the slope of the sigmoidal. (**0** an "inflection point," where the slope of the function changes sign. The inflection point can be translated by using the **bias** term).

Again: With a net input of **0** the output of a sigmoidal will be **1/2** way between its upper and lower asymptote. As the net input grows or diminishes it goes towards its upper or lower asymptote, depending on the slope.

- Set the output neuron to have a **sigmoidal** update rule with **bias 0**, **upper bound 1** and **lower bound 0**.
- Consider the examples in (*). What will the activation of the output roughly be in each case?
- For what values of upper asymptote and lower asymptote will a sigmoidal neuron take the value **5** in response to net input **0**?
- Given net inputs of **2, 2.5,** and **0**, how could we make the first two net inputs produce much smaller values than **1**, but still have **0** produce **0.5**?
- What would happen if we made the slope **-1**? Again try out the examples in (*).