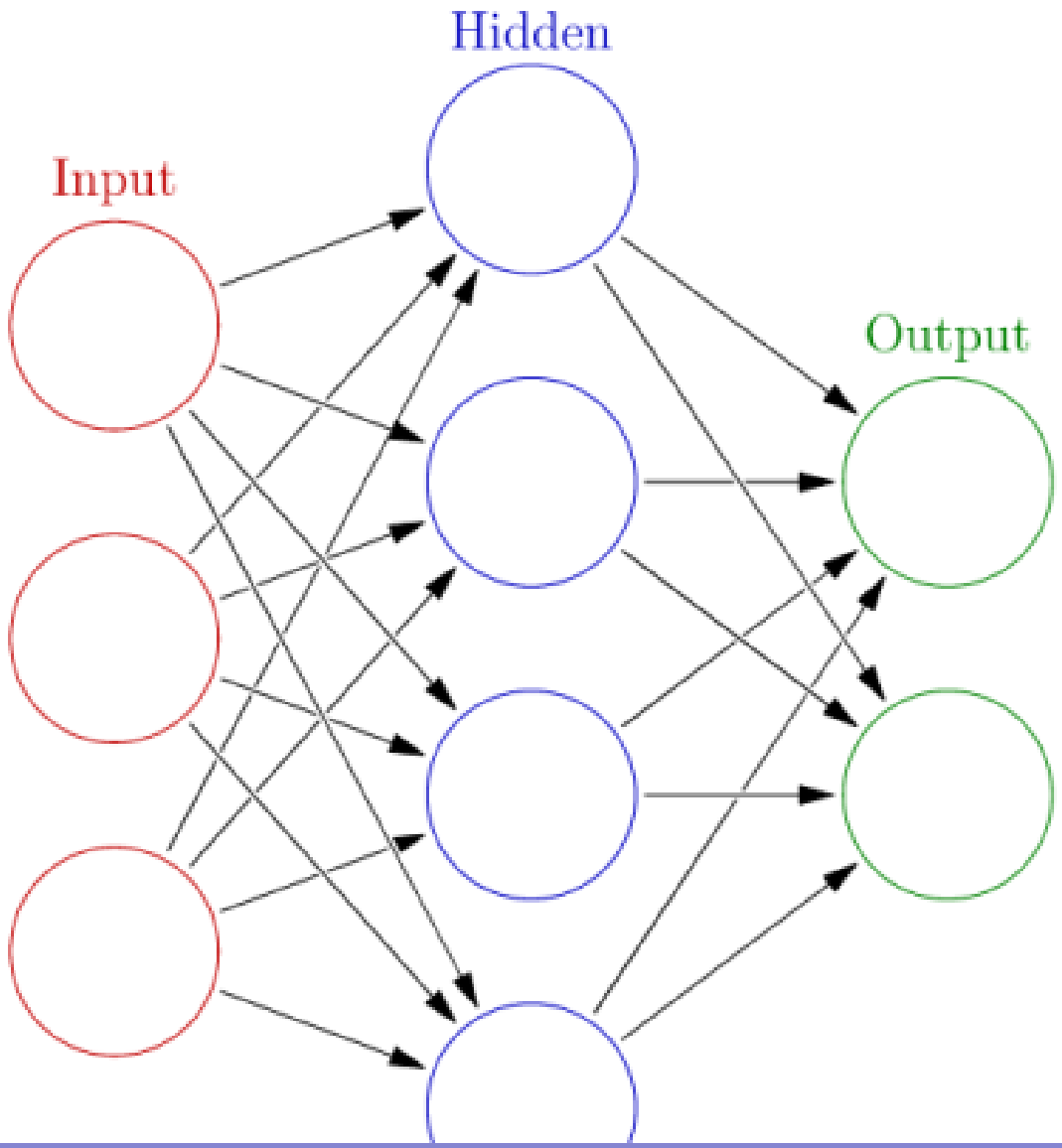
Neural Networks

Jan-Philipp Kolb

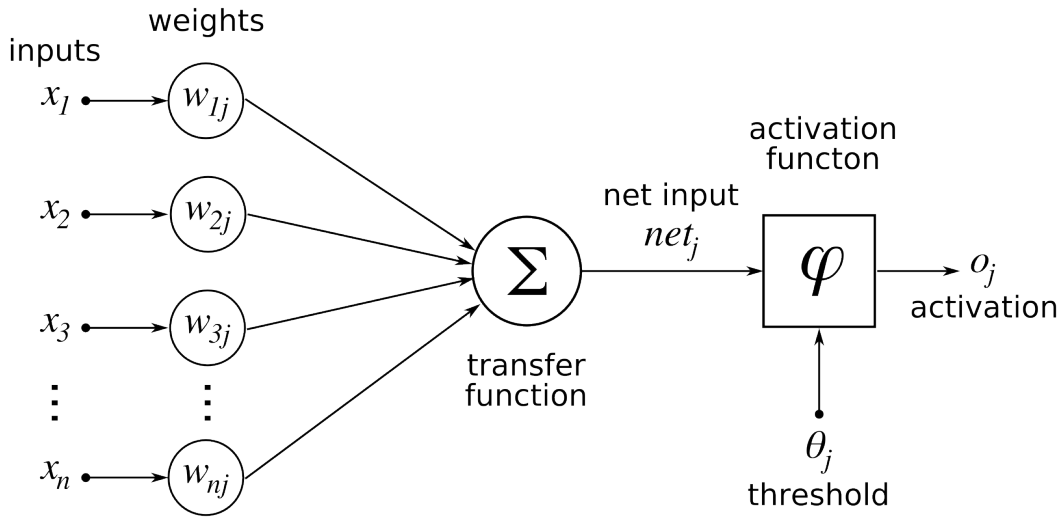
04 Juni, 2019

Examples of Multi-Neuron



Artifical Neuron

- Inputs correspond to raw data values
- The transfer function sums all the inputs together (cumulative inputs).
- If the summed input values reach a specified threshold, the activation function generates an output signal (all or nothing).
- The output signal then moves to a raw output or other neurons.
- This basic artificial neuron is combined with multiple other artificial neurons to create an ANN.

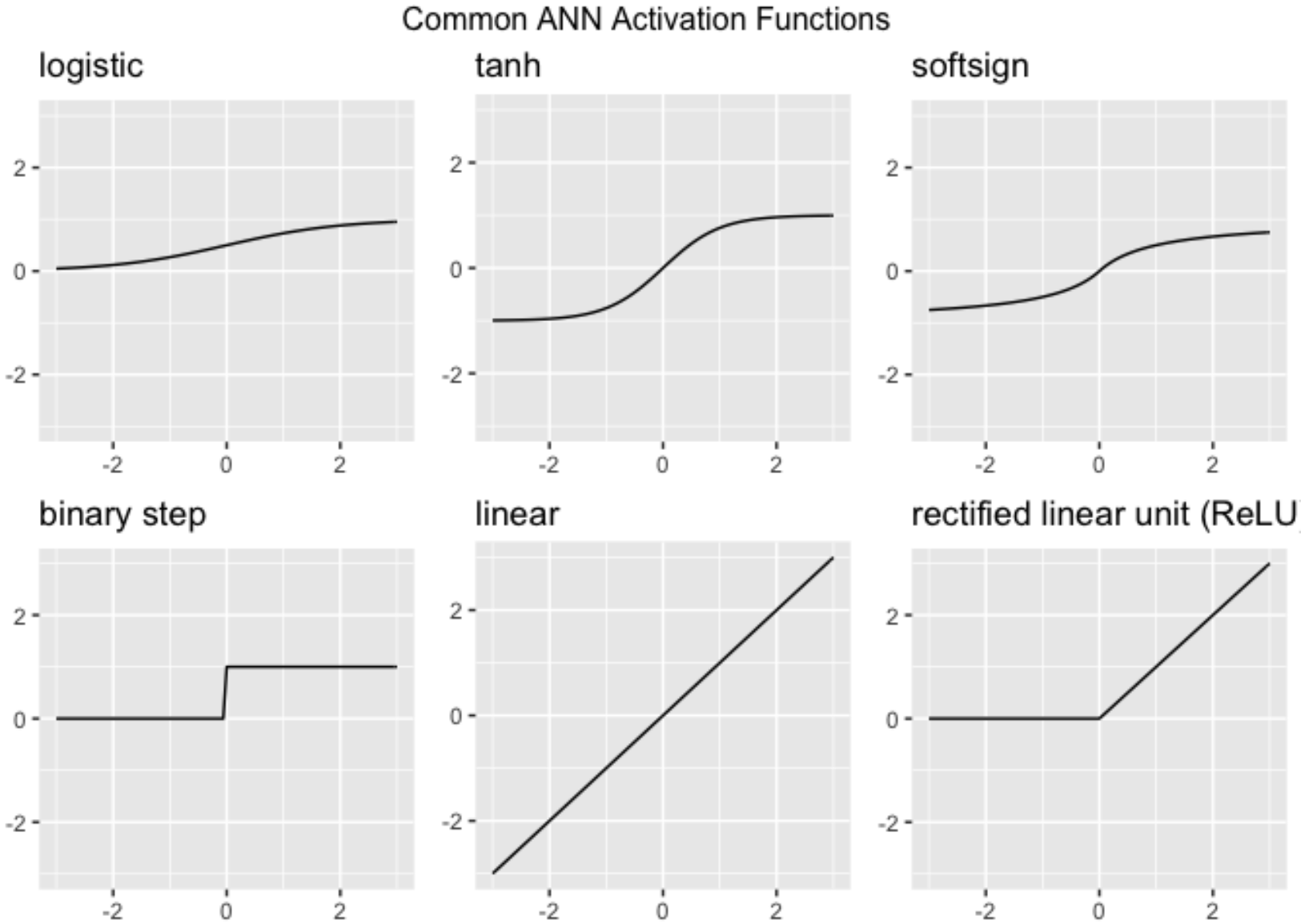


Activation Functions

- The capability of ANNs to learn any function, (given sufficient training data examples) are dependent on the appropriate selection of the **activation function(s)** present in the network.
- They enable the ANN to learn non-linear properties present in the data.
- The input into the activation function is the weighted sum of the input features from the preceding layer.
- Let o_j be the output from the jth neuron in a given layer for a network for k input vector features.

$$o_j = \Phi(b_j + \sum_{i=1}^{p} w_i x_i)$$

Common ANN Activation functions



The output (o_j)...

- ... can feed into the output layer of a neural network, or in deeper architectures may feed into additional hidden layers.
- The activation function determines if the sum of the weighted inputs plus a bias term is sufficiently large to trigger the firing of the neuron.
- No universal best choice for the activation function, researchers have provided information regarding what activation functions work well for ANN solutions to many common problems.
- The choice of the activation function governs the required data scaling necessary for ANN analysis.

How ANNs Learn

- We have some features (X) describing an output (y)
- To begin training our notional single-layer one-neuron neural network we initially randomly assign weights.
- We then run the neural network with the random weights and record the outputs generated.
- This is called a forward pass. Output values, in our case called y, are a function of the input values (X), the random initial weights (w) and our choice of the threshold function (T).

$$\hat{y} = f(X, w, T)$$

Choice of the performance function

- Once we have our ANN output values ($\hat{y}$) we can compare them to the data set output values (y).
- To do this we use a performance function P.
- The choice of the performance function is a choice of the analyst, we choose Sum of Squared Errors (SSE).