

# Training the ANN with Stochastic Gradient Descent

---

**STEP 1:** Randomly initialise the weights to small numbers close to 0 (but not 0).



**STEP 2:** Input the first observation of your dataset in the input layer, each feature in one input node.



**STEP 3:** Forward-Propagation: from left to right, the neurons are activated in a way that the impact of each neuron's activation is limited by the weights. Propagate the activations until getting the predicted result  $y$ .



**STEP 4:** Compare the predicted result to the actual result. Measure the generated error.



**STEP 5:** Back-Propagation: from right to left, the error is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much we update the weights.



**STEP 6:** Repeat Steps 1 to 5 and update the weights after each observation (Reinforcement Learning). Or:  
Repeat Steps 1 to 5 but update the weights only after a batch of observations (Batch Learning).



**STEP 7:** When the whole training set passed through the ANN, that makes an epoch. Redo more epochs.