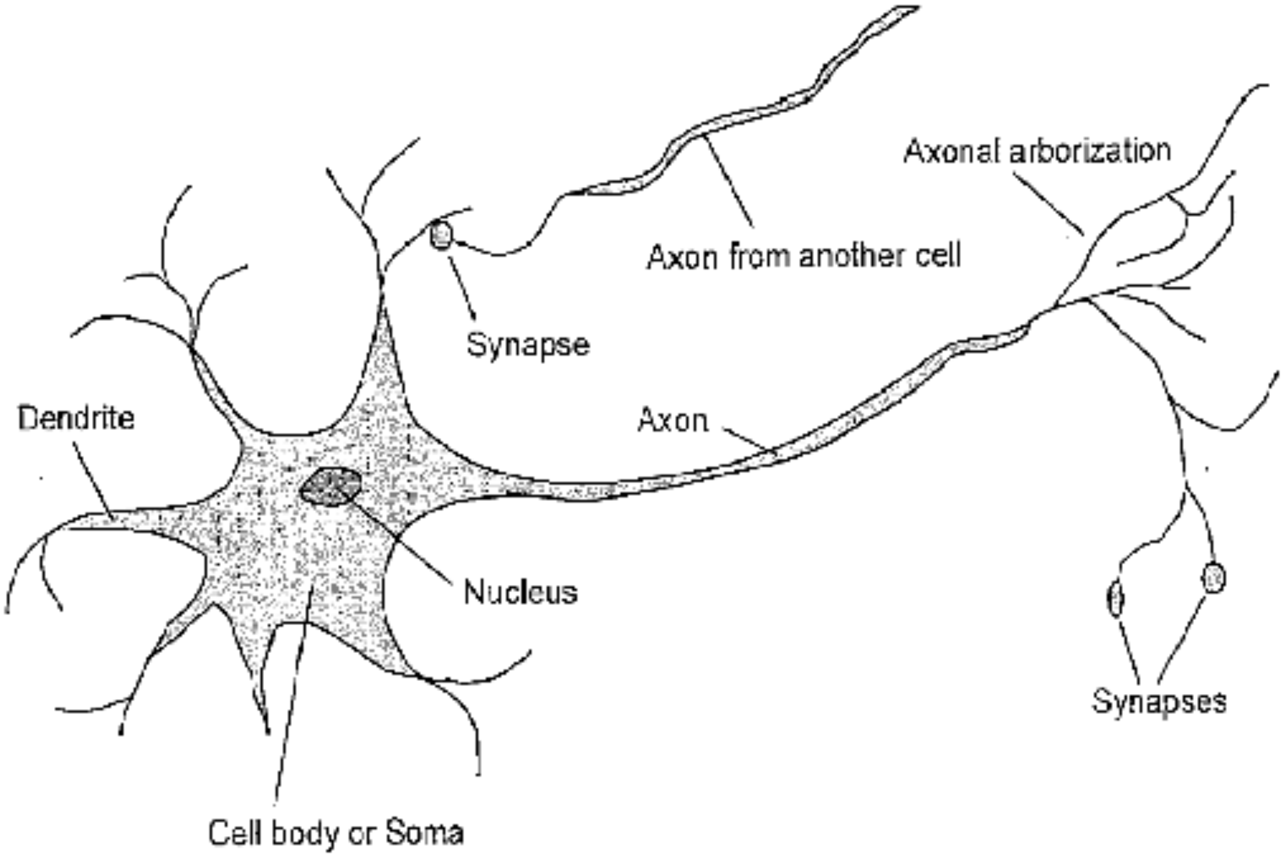
Artificial Intelligence and Machine Learning

Deep Learning

Big Idea

- Simulate human brain
- Typical human brain has
  - 10^11 neurons of 20+ types,
  - 10^14 synapses
  - 1ms to 10ms cycle time
- Signals are noisy “spike trains” of electrical potential.

Neuron


Axonal arborization
Axon from another cell
Synapse
Dendrite
Axon
Nucleus
Synapses
Cell body or Soma

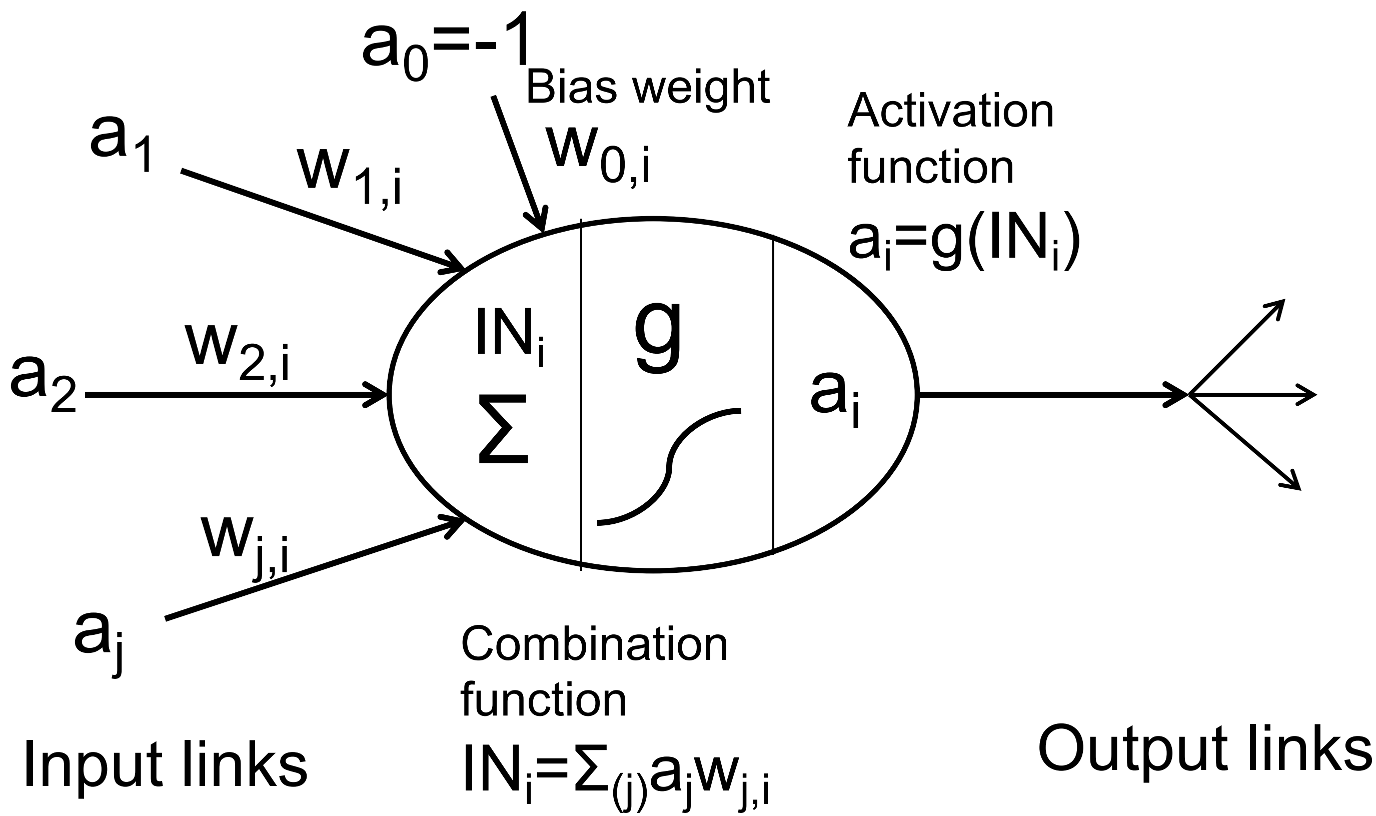
Neuron

- Neuron is an electrically excitable cell that processes and transmits information by electrical and chemical signaling.
- Dendrites receive an electric charge.
- The strengths of all the received charges are summed up. The aggregate value is then passed to the soma (cell body) to axon hillock. (Signal summation)
- If the aggregate input is greater than the axon hillock's threshold value, then the neuron fires, and an output signal is transmitted down the axon. (activation threshold)
- The strength of the output is constant, regardless of whether the input was just above the threshold, or a lot higher. (constant output)

Neuron Communication

- The signal of one neuron is transmitted to other neurons through synapses.
- The physical and neurochemical characteristics of each synapse determines the strength and polarity of the new input signal.

Modelling a Neuron



Activation Functions

- Step function or threshold function

$$f(x) = \begin{cases} 0, & if\ x \leq 0 \\ 1, & otherwise \end{cases}$$

- Sigmoid/logistic function (continuous and differentiable)

$$f(x) = \frac{1}{1 + e^{-x}}$$

- Rectified Linear Units (ReLU) – Typically performs better than other activation functions for hidden layers

$$f(x) = \max(0, x)$$

Changing the bias weight $w_{0,i}$ moves the threshold location.

Modelling Neuron Networks

- Consists nodes and edges.
- Node takes input and triggers other nodes through connections.
- Each node has an activation function to decide whether to fire up.
- Each edge not only permits to transfer the value, but also has a weight.
- Artificial neural network simulates the brain.
- Artificial neural network is abstract and media independent. We can use parallel circuits or execute a program on a serial processor.

Compare Human Brain with ANN

- Brain is highly complex, non-linear and massively-parallel system.
- Brain has 1ms to 10ms response time
- Integrated circuit has 1ns response time
- If we construct parallel circuits to realize ANN, it’s expensive.
- If we execute a serial program on a computer to realize ANN, it is easy to create massive number of neurons. However, the simulated system would be slower by many orders of magnitude than a real neural network.

Example of Single Layer Perceptron

All of them use a single perceptron, and the activation function is a step function.

- And
  $W_0 = 1.5$, $W_1 = 1$, $W_2 = 1$
- Or
  $W_0 = 0.5$, $W_1 = 1$, $W_2 = 1$
- Not
  $W_0 = -0.5$, $W_1 = -1$

Perceptron Learning

- Learn by adjusting weights to reduce error on training dataset
- The error can be expressed as
$$E = \frac{1}{2} Err^2 = \frac{1}{2}(y - h_W(x))^2$$
- Perform optimization search by gradient descent:
$$\frac{\partial E}{\partial W_j} = Err \times \frac{\partial Err}{\partial W_j} = -Err \times g'(IN) \times x_j$$
- Simple weight update rule:
$$W_j = W_j + \alpha \times Err \times g'(IN) \times x_j$$
- Perceptron can learn for any linearly separable functions.

Multilayer Perceptrons

- Nodes are organized into layers. Edges are directed and carry weight.
- Layers are usually fully connected.
- No connections inside the layer.
- There are input layer and output layer.
- There can be may hidden layers.
- No direct connection between input and output layers.
- The number of hidden neurons is typically manually chosen.

Very Powerful Model

- With sigmoidal activation functions, it can be shown that a three-layer ANN can approximate any continuous function to arbitrary accuracy.
- Model learning => adjust weights
- Algorithm: backpropagation learning algorithm
  - Forward pass: feed forward propagation of input pattern signals through network
  - Backward pass: computes error signal, propagates the error backwards through network, adjusting the weight where the actual and desired output values are different

Back-propagation Learning

- Output layer:

$$\Delta_i = Err_i \times g'(IN_i)$$
$$W_{j,i} = W_{j,i} + \alpha \times a_j \times \Delta_i$$

- Hidden layer – back-propagate the error from the output layer:

$$\Delta_j = g'\left(IN_j\right) \times \sum_i (W_{j,i} \times \Delta_i)$$
$$W_{k,j} = W_{k,j} + \alpha \times a_k \times \Delta_j$$

- Most neuroscientists deny that back propagation occurs in the brain.

(Sequential) BP Learning Algorithm

Set learning rate α
Set initial weight values
Loop until stopping criteria satisfied

  select next training data item <input, target>
  present input pattern to input units
  compute forward from input to output
  present target response to output units
  compute error signal from output to input
  update all weights while propagate error backwards

End loop

Issues in Training Neural Networks

- Training parameters
  - Learning Rate
  - Momentum
  - Batch size
- Overfitting problem
  - Memorizing training data instead of learning knowledge
  - Problem: success rate won’t be as high on testing data as on training data
  - Overcome approach
    - Neuron dropout
    - Regularization

Self Organizing Maps

- Unsupervised training
  - Training data items don’t have pre-defined labels
- Similar to clustering
- But can apply to new data items that are not in the training dataset

Deep Learning

- Deep learning typically includes the following features:
  - Partially labeled data
  - Using Rectified Linear Units (ReLU) as the activation function for hidden layer neurons
  - Convolutional neural networks
    - Sparse connectivity
  - Applying neuron dropout techniques