CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Zoom link

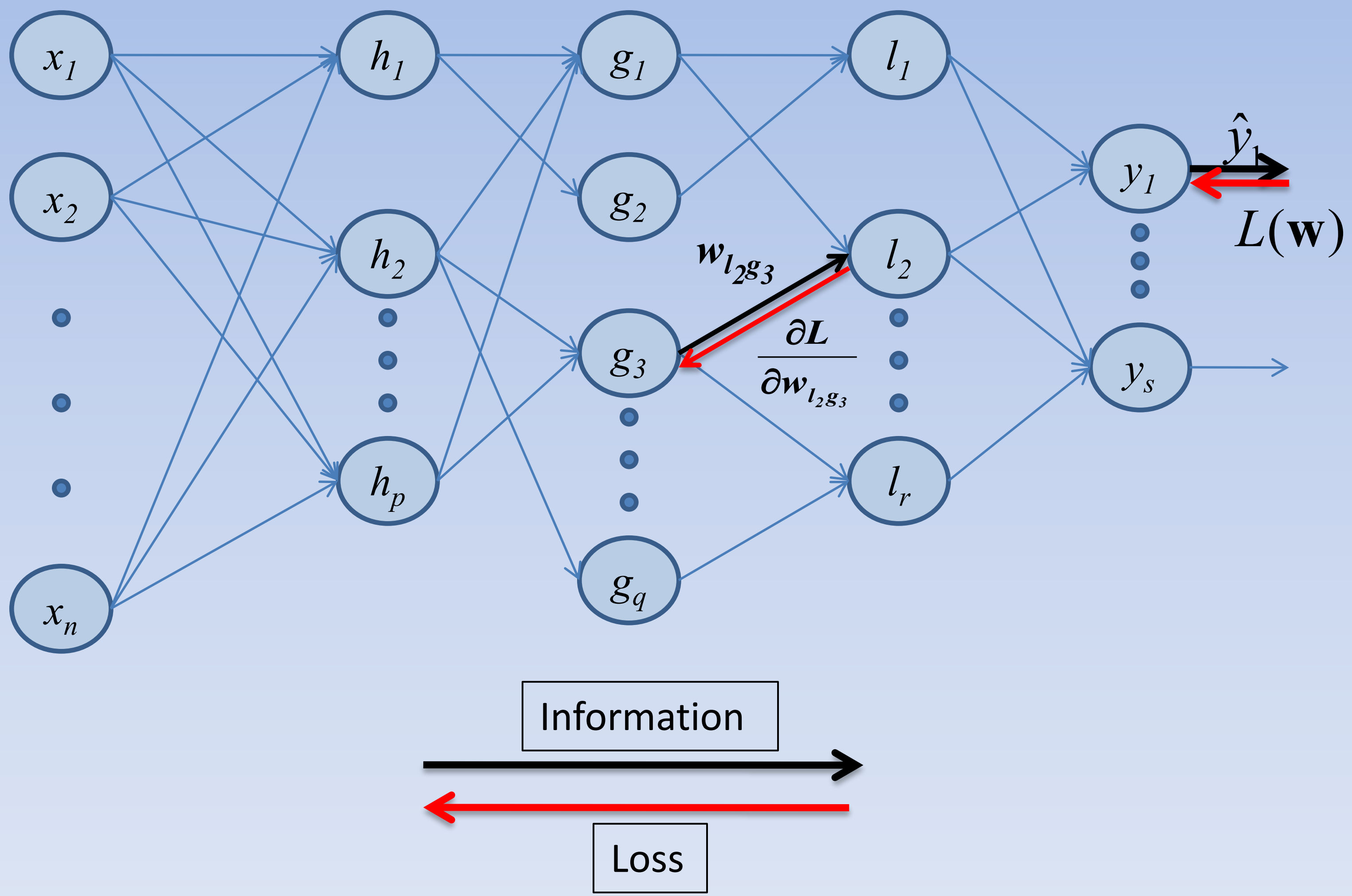
Recap

- To estimate perceptron parameters we define a l___ function that m___ the d___ between our e___ labels and the t___ labels.
- The gradient descent procedure will c_____ to a g____ m______ because _________.
- We can also use s______ gradient descent. This is different from regular GD because _____.
- SGD is useful if the function has multiple l____ o____. It can also be used during o____ l_____.
- The _____ function cannot be learned with a perceptron.
- In a general neural network, there are layers of h____ units between input and output.
- Every Boolean function can be represented by a network with ____ hidden layer.
- Every continuous function can be represented by a network with ____ hidden layers.
- However, the tradeoffs are (1) (2) (3).
- The activation functions in an ANN must be n__ l___ for learning.
- The sigmoid function outputs h(u)=1/(A + exp(B)).
- Backpropagation performs l____-w___ gradient descent. First, information flows f_____ through the network to compute the o____. Then, the l___ flows backward to compute the gradients.

Today

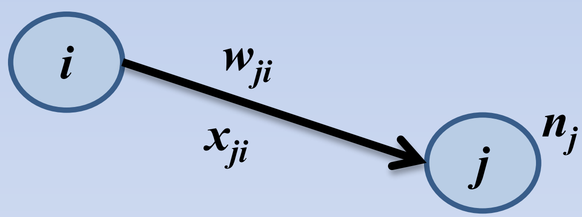
- Artificial Neural Networks (Ch 4, Mitchell)

Backpropagation



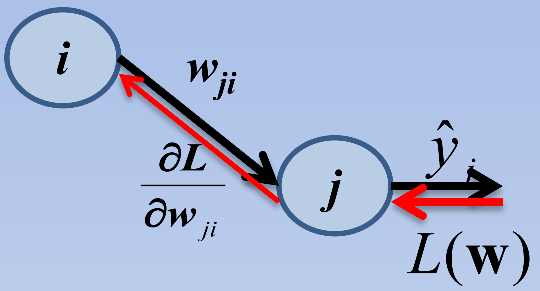
Backpropagation (SGD)

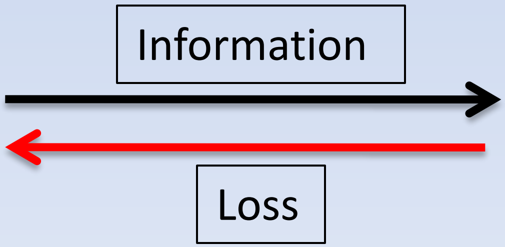
- Let x_{ji} be the i^{th} input to unit j
- Let w_{ji} be the parameter associated with x_{ji}
- Let $n_j = \sum_i w_{ji} x_{ji}$ be the "net input" to unit j



- Observe that $$\frac{\partial L}{\partial w_{ji}} = \frac{\partial L}{\partial n_j}\frac{\partial n_j}{\partial w_{ji}} = \frac{\partial L}{\partial n_j} x_{ji}$$

Output Layer





Derivation (output layer)

$$h(u) = \frac{1}{(1+e^{-u})}; 1 - h(u) = \frac{e^{-u}}{(1+e^{-u})}$$ (Sigmoid)

$$\frac{dh}{du} = \frac{e^{-u}}{(1+e^{-u})^2} = h(u)(1-h(u))$$ (Derivative of Sigmoid)

$$L(w_{ji}) = \frac{1}{2}(y_j - h(n_j))^2$$ (Squared Loss)

$$\frac{\partial L}{\partial n_j} = (h(n_j) - y_j)\frac{\partial h(n_j)}{\partial n_j}$$

$$\frac{\partial h(n_j)}{\partial n_j} = h(n_j)(1-h(n_j))$$ (Using Derivative of Sigmoid)

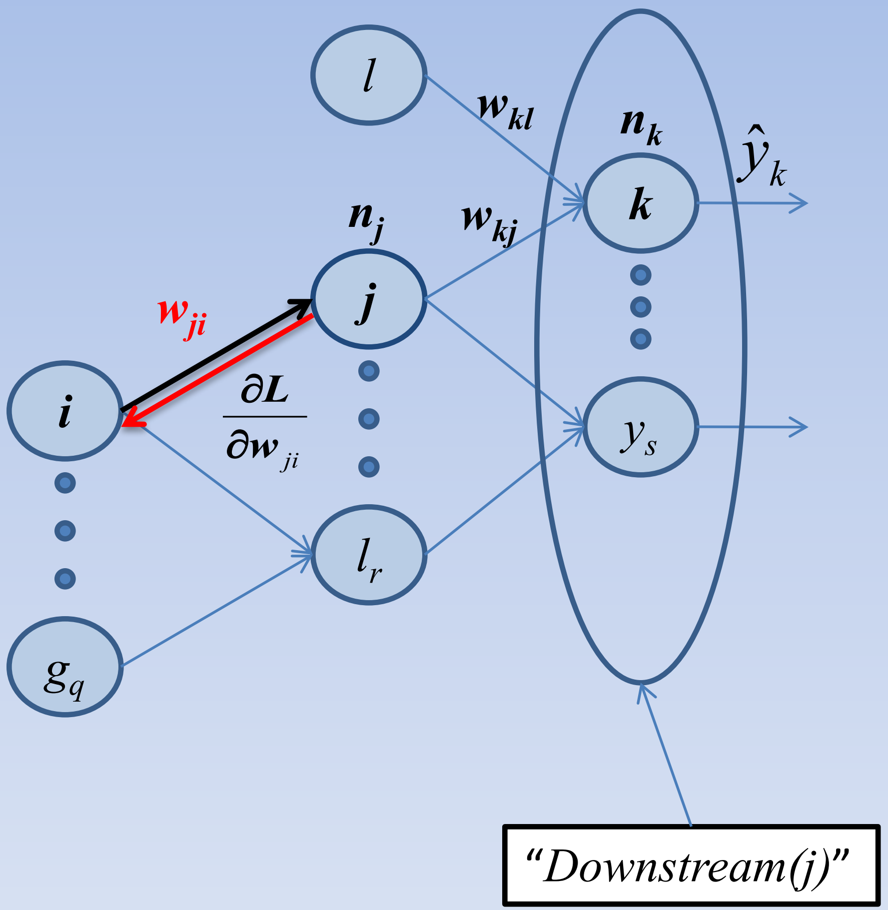
Derivation (output layer)

$$\frac{\partial L}{\partial n_j} = (h(n_j) - y_j)\frac{\partial h(n_j)}{\partial n_j}$$

$$\frac{\partial h(n_j)}{\partial n_j} = h(n_j)(1 - h(n_j))$$

$$\boxed{\frac{\partial L}{\partial w_{ji}} = (h(n_j) - y_j)h(n_j)(1 - h(n_j))x_{ji}}$$

Hidden Layer



Derivation (Hidden Layer)

- Since *j* affects the output only through *Downstream(j)*,

$$\frac{\partial L}{\partial n_j} = \sum_{k \in Downstream(j)} \frac{\partial L}{\partial n_k} \frac{\partial n_k}{\partial n_j}$$

Already calculated, next layer

$$n_k = \sum_l w_{kl} h(n_l); \frac{\partial n_k}{\partial n_j} = \frac{\partial \left(w_{kj} h(n_j) \right)}{\partial n_j}$$

$$= w_{kj} \frac{\partial h(n_j)}{\partial n_j} = w_{kj} h(n_j)(1 - h(n_j))$$

$$\frac{\partial L}{\partial n_j} = h(n_j)(1 - h(n_j)) \sum_{k \in Downstream(j)} \frac{\partial L}{\partial n_k} w_{kj}$$

Derivation (Hidden Layer)

$$\frac{\partial L}{\partial w_{ji}} = \frac{\partial L}{\partial n_j} x_{ji}$$

$$= h(n_j)(1-h(n_j))x_{ji} \sum_{k \in Downstream(j)} \frac{\partial L}{\partial n_k} w_{kj}$$

$$\boxed{= h(n_j)(1-h(n_j))x_{ji} \sum_{k \in Downstream(j)} \frac{\partial L}{\partial w_{kj}} \frac{w_{kj}}{x_{kj}}}$$

Review

- Consider a neural network with 2 input units, 2 hidden units and 1 output unit and all weights initialized to 1, with the bias set to zero. Using squared loss, show the weights after the first backprop update with these examples.

x_1	x_2	f
0	0	0
0	1	1

Updates

$$\frac{\partial L}{\partial w_{oh}} = h(n_o)(1 - h(n_o))x_{oh}(h(n_o) - y_o)$$

$$\frac{\partial L}{\partial w_{hi}} = h(n_h)(1 - h(n_h))x_{hi} \sum_{k \in Downstream(h)} \frac{\partial L}{\partial w_{kh}} \frac{w_{kh}}{x_{kh}}$$

Example notes

- Zeros as inputs
- SGD effects
- Vanishing gradients