CS 189 Introduction to Machine Learning
Spring 2023 Jonathan Shewchuk

DIS10

1 Simple Gradients and their Interpretations

For each of the functions below, what are the partial derivatives with respect to each variable? For (a) only, what is ∇f? For parts (a)–(c), describe how changes in each variable affect f.

(a) $f(x, y) = xy$

(b) $f(x, y) = x + y$

(c) $f(x, y) = \max\{x, y\}$

(d) $\tanh x = \dfrac{\sinh x}{\cosh x} = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$. Hint: $\tanh x = \dfrac{1 - e^{-2x}}{1 + e^{-2x}} = s(2x) - (1 - s(2x)) = 2s(2x) - 1$.

1. a) $f(x,y) = xy$

$$\frac{\partial f}{\partial x} = y \qquad \frac{\partial f}{\partial y} = x$$

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} y \\ x \end{bmatrix}$$

The partial derivative of each variable tells us the sensitivity of the func. to that variable.

In this case, increasing x by dx will cause the func. to increase by $df = y\,dx$. Similarly, increasing y by dy will cause the func. to inc. by $df = x\,dy$.

ex. $x = 1$, $y = -2$

$f(x,y) = 1(-2) = -2$

$dx = 0.1$

$f(x+dx, y) = (1+0.1)(-2) = -2.2$

$df = f(x+dx, y) - f(x,y) = -2.2 - (-2) = -0.2$

Notice that $df = y\,dx$ as expected.

b) $f(x,y) = x + y$

$$\frac{\partial f}{\partial x} = 1 \qquad \frac{\partial f}{\partial y} = 1$$

For this case, increasing x by dx causes the func. to also inc. by $df = dx$. Similarly, increasing y by dy causes the func. to also inc. by $df = dy$. Now the rate of inc. does not depend on the initial x or y.

c) $f(x,y) = \max\{x,y\}$

$$= \begin{cases} x & \text{if } x \geq y \\ y & \text{o.w.} \end{cases}$$

indicator func.

$$\frac{\partial f}{\partial x} = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{o.w.} \end{cases} = \mathbb{1}\{x \geq y\}$$

$$\frac{\partial f}{\partial y} = \begin{cases} 0 & \text{if } x \geq y \\ 1 & \text{o.w.} \end{cases} = \mathbb{1}\{y \geq x\}$$

Now increasing x by dx will cause the func. to inc. by $df = dx$ if $x \geq y$ but will not cause any change if $x < y$. Note that the derivative only tells us the result of small changes to our variable.

d) $f(x) = \tanh(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \cdot \frac{e^{-x}}{e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$

Recall that the logistic func. is defined as

$S(r) = \frac{1}{1+e^{-r}}$

$S(2x) = \frac{1}{1+e^{-2x}}$

$1 - S(2x) = \frac{e^{-2x}}{1+e^{-2x}}$

$f(x) = S(2x) - (1 - S(2x)) = 2S(2x) - 1$

By the chain rule,

$f'(x) = 2S'(2x) \cdot 2 = 4S'(2x)$

Recall the derivative of the logistic func....

$S'(r) = \frac{e^{-r}}{(1+e^{-r})^2} = \left(\frac{1}{1+e^{-r}}\right)\left(\frac{e^{-r}}{1+e^{-r}}\right) = S(r)(1-S(r))$

$\therefore f'(x) = 4S(2x)(1-S(2x))$

2 Backprop in Practice: Staged Computation

Consider the function $f(x, y, z) = (x + y)z + x$.

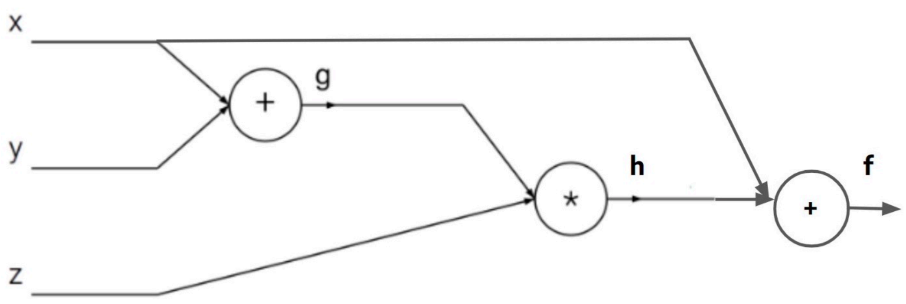
(a) Draw a directed acyclic graph (DAG)/circuit/network that represents the computation of f. Assign a variable name to each intermediate result.

(b) Write pseudocode for the forward pass and backward pass (backpropagation) in the network.

(c) On your network drawing, write the intermediate values in the forward and backward passes when the inputs are $x = -2$, $y = 5$, and $z = -4$.

2. $f(x,y,z) = (x+y)z + x$

a)



b) <u>forward</u>:

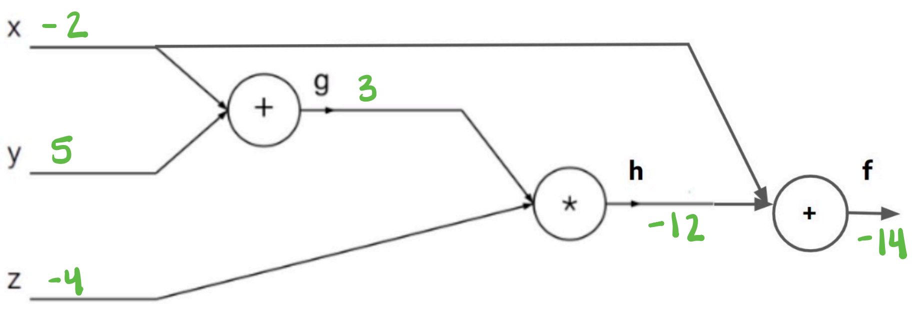
1) $g = x + y$
2) $h = gz$
3) $f = h + x$

<u>backward</u>:

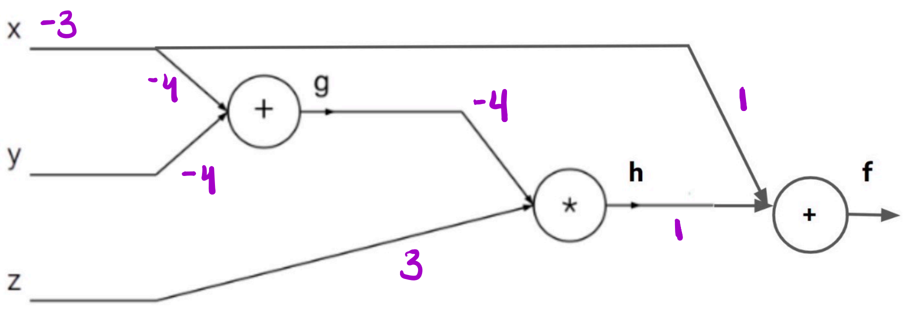
3) dfdh = 1
   dfdx = 1
2) dfdg = dfdh • dhdg = dfdh • z
   dfdz = dfdh • dhdz = dfdh • g
1) dfdx += dfdg • dgdx = dfdg • 1
   dfdy = dfdg • dgdy = dfdg • 1

c) $x=-2,\ y=5,\ z=-4$

forward:



backward:

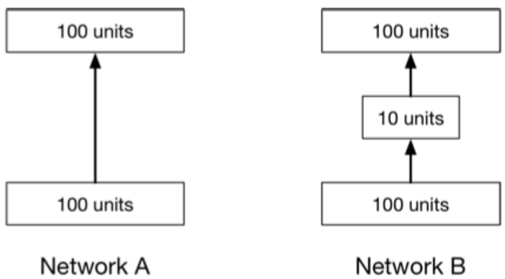


3 Model Intuition

(a) What can go wrong if you just initialize all the weights in a neural network to exactly zero? What about to the same nonzero value?

(b) Adding nodes in the hidden layer gives the neural network more approximation ability, because you are adding more parameters. How many weight parameters are there in a neural network with architecture specified by $n = \left[n^{(0)}, n^{(1)}, ..., n^{(\ell)}\right]$, a vector giving the number of nodes in each of the $\ell + 1$ layers? (Layer 0 is the input layer, and layer ℓ is the output layer.) Evaluate your formula for a network n = [8, 10, 10, 3].

(c) Consider the two networks in the image below, where the added layer in Network B has 10 units with **linear activation**. Give one advantage of Network A over Network B, and one advantage of Network B over Network A.

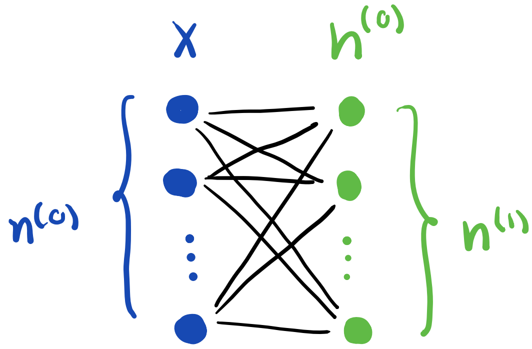


(a) What can go wrong if you just initialize all the weights in a neural network to exactly zero? What about to the same nonzero value?

If two hidden units connected to the same inputs w/ the same activation funcs. are initialized w/ the same params., then the deterministic learning alg. will constantly update both of these units the same way.

This means that all the neurons in a hidden layer will have the same weights at every iteration of the learning alg. Each neuron in a layer then represents the same "feature" + there is no advantage of including multiple neurons per layer. We then lose the expressiveness of our model.

(b) Adding nodes in the hidden layer gives the neural network more approximation ability, because you are adding more parameters. How many weight parameters are there in a neural network with architecture specified by $n = \left[n^{(0)}, n^{(1)}, ..., n^{(\ell)}\right]$, a vector giving the number of nodes in each of the $\ell + 1$ layers? (Layer 0 is the input layer, and layer ℓ is the output layer.) Evaluate your formula for a network n = [8, 10, 10, 3].



The # of weight parameters b/t the input & first hidden layer is $n^{(0)} n^{(1)}$.

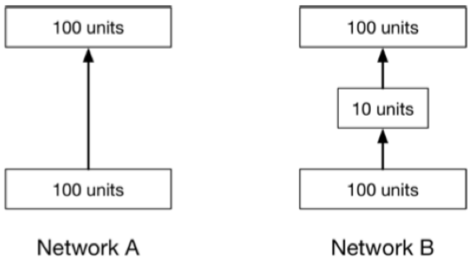
The total # of weight params. is then

$$\sum_{i=0}^{\ell-1} n^{(i)} n^{(i+1)}$$

For a network w/ layers n = [8, 10, 10, 3], the total # of weight params. is

$8(10) + 10(10) + 10(3) = 80 + 100 + 30 = 210$

(c) Consider the two networks in the image below, where the added layer in Network B has 10 units with **linear activation**. Give one advantage of Network A over Network B, and one advantage of Network B over Network A.



Adding a hidden layer w/ linear activation actually reduces the # of weight params.

A : # of params = 100 (100) = 10,000
B : # of params = 100 (10) + 10 (100) = 2,000

Therefore, A is more expressive than B, meaning that it can learn funcs. B can't. However, it is also more computationally expensive b/c it requires a much larger matrix operation.

4 More Backprop in Practice: Staged Computation

Consider the function $f(w, x) = \dfrac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$.

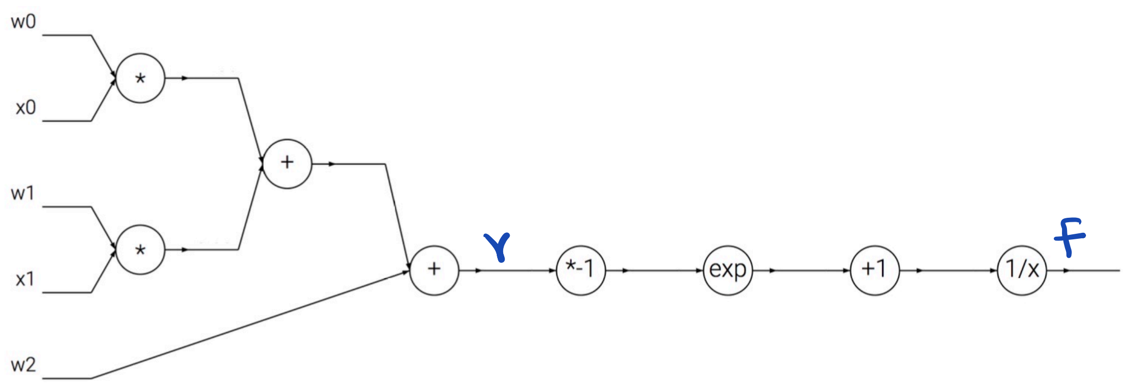
(a) Draw a network that represents the computation of f.

(b) Write pseudocode for the forward pass and backward pass (backpropagation) of the network.

(c) With the weights $w = [2, -3, -3]$ and inputs $x = [-1, -2]$, write the intermediate values in the forward and backward passes on your network diagram.

(d) Now consider a network that computes the function $f(x, y) = \dfrac{x + s(y)}{s(x) + (x + y)^2}$. Write pseudocode for the forward and backward passes of the network.

4. $f(w,x) = \dfrac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$

a) 

b) <u>forward</u>:

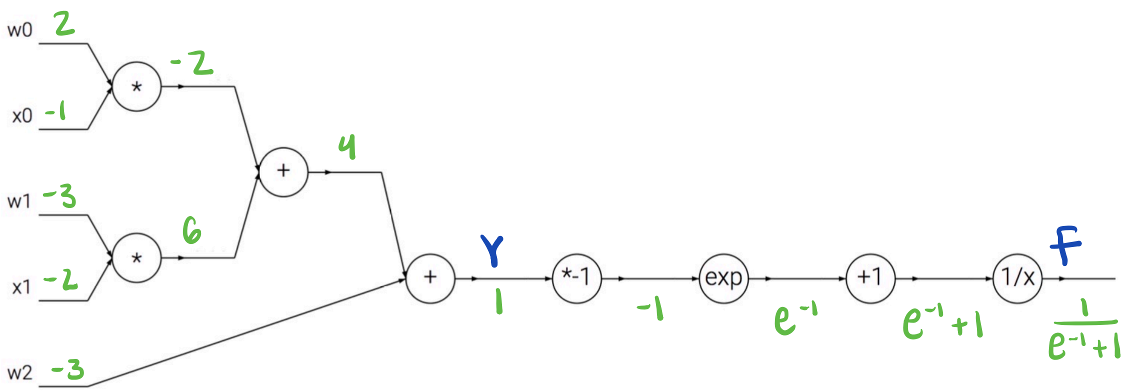
1) $Y = w_0x_0 + w_1x_1 + w_2$
2) $F = S(Y)$, where S is sigmoid func.

<u>backward</u>: deriv. of sigmoid func.

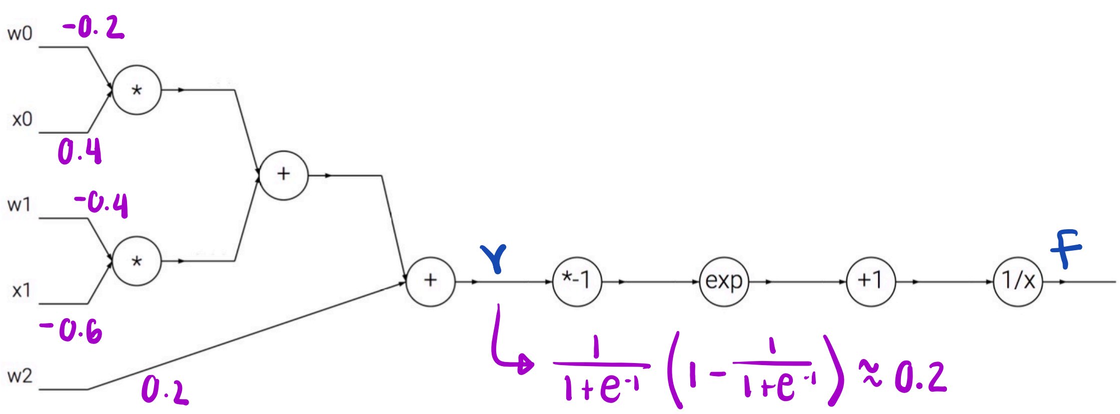
2) $dfdY = F(1-F)$
1) $dfdx_0 = dfdY \cdot dYdx_0 = dfdY \cdot w_0$
   $dfdx_1 = dfdY \cdot dYdx_1 = dfdY \cdot w_1$
   $dfdw_0 = dfdY \cdot dYdw_0 = dfdY \cdot x_0$
   $dfdw_1 = dfdY \cdot dYdw_1 = dfdY \cdot x_1$
   $dfdw_2 = dfdY \cdot dYdw_1 = dfdY \cdot 1$

c) $w = [2, -3, -3]$, $x = [-1, -2]$

Forward:



backward:



d) $F(x,y) = \dfrac{x + s(y)}{s(x) + (x+y)^2}$

<u>forward</u>:

1) $s(y) = 1/(1+e^{-y})$
2) $num = x + s(y)$
3) $s(x) = 1/(1+e^{-x})$
4) $sum = x + y$
5) $sqr = sum^2$
6) $denom = s(x) + sqr$
7) $inv = 1/denom$
8) $f = num \cdot inv$

Note that we broke up the fwd pass in this way so that we can easily compute the deriv. w.r.t. each intermed. variable.

<u>backward</u>:

8) $dnum = inv$

   $dinv = num$

7) $ddenom = dinv \cdot (-1/denom^2)$
6) $ds(x) = ddenom \cdot 1$

   $dsqr = ddenom \cdot 1$

5) $dsum = dsqr \cdot 2sum$
4) $dx = dsum \cdot 1$

   $dy = dsum \cdot 1$

3) $dx \mathrel{+}= ds(x) \cdot [s(x)(1-s(x))]$

2) dx += $dnum \cdot 1$

$ds(y) = dnum \cdot 1$

1) $dy = ds(y) \cdot [s(y)(1 - s(y))]$