HW1:

Taylor Series

Multivariate Calculus

Inner Product

Closed - form Solution (to a mathematical problem)

Lower / Upper bound of a function

CS7643: Deep Learning
Fall 2018
Homework 1

Instructor: Dhruv Batra
TAs: Harsh Agrawal, Michael Cogswell, Nirbhay Modhe, Erik Wijmans
Discussions: https://piazza.com/gatech/fall2018/cs48037643

Due: Tuesday, October 2, 11:55pm

Instructions

1. We will be using Gradescope to collect your assignments. Please read the following instructions for submitting to Gradescope carefully!

   - Each problem/sub-problem should be on one or more pages. This assignment has 6 total problems/sub-problems, so you should have at least 6 pages in your submission.
   - When submitting to Gradescope, make sure to mark which page corresponds to each problem/sub-problem.
   - For the coding problem (problem 6), please use the provided `collect_submission.sh` script and upload `hw1.zip` to the HW1 Code assignment on Gradescope. While we will not be explicitly grading your code, you are still required to submit it.

     Please make sure you have saved the most recent version of your jupyter notebook before running this script.
   - Note: This is a large class and Gradescope's assignment segmentation features are essential. Failure to follow these instructions may result in parts of your assignment not being graded. We will not entertain regrading requests for failure to follow instructions.

     Please read https://stats200.stanford.edu/gradescope_tips.pdf for additional information on submitting to Gradescope.

2. LaTeX'd solutions are strongly encouraged (solution template available at cc.gatech.edu/classes/AY2019/cs7643_fall/assets/sol1.tex), but scanned handwritten copies are acceptable. Hard copies are **not** accepted.

3. We generally encourage you to collaborate with other students.

   You may talk to a friend, discuss the questions and potential directions for solving them. However, you need to write your own solutions and code separately, and *not* as a group activity. Please list the students you collaborated with.

1 Gradient Descent

1. (3 points) We often use iterative optimization algorithms such as Gradient Descent to find $\mathbf{w}$ that minimizes a loss function $f(\mathbf{w})$. Recall that in gradient descent, we start with an initial value of $\mathbf{w}$ (say $\mathbf{w}^{(1)}$) and iteratively take a step in the direction of the negative of the gradient of the objective function *i.e.*

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \tag{1}$$

for learning rate $\eta > 0$.

In this question, we will develop a slightly deeper understanding of this update rule. Recall the first-order Taylor approximation of f at $\mathbf{w}^{(t)}$:

$$f(\mathbf{w}) \approx f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle \tag{2}$$

When f is convex, this approximation forms a lower bound of f. Since this approximation is a 'simpler' function than $f(\cdot)$, we could consider minimizing the approximation instead of $f(\cdot)$. Two immediate problems: (1) the approximation is affine (thus unbounded from below) and (2) the approximation is faithful for $\mathbf{w}$ close to $\mathbf{w}^{(t)}$. To solve both problems, we add a squared ℓ_2 *proximity term* to the approximation minimization:

$$\operatorname*{argmin}_{\mathbf{w}} \underbrace{f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle}_{\text{affine lower bound to } f(\cdot)} + \underbrace{\frac{\lambda}{2}}_{\text{trade-off}} \underbrace{\left\| \mathbf{w} - \mathbf{w}^{(t)} \right\|^2}_{\text{proximity term}} \tag{3}$$

Notice that the optimization problem above is an unconstrained quadratic programming problem, meaning that it can be solved in closed form.

What is the solution $\mathbf{w}^*$ of the above optimization? What does that tell you about the gradient descent update rule? What is the relationship between λ and η?

2. (3 points) Show that for a sequence of vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_T$ and $\mathbf{w}^\star$ that minimizes $f(\mathbf{w})$, an update equation of the form $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$ with $\mathbf{w}^{(1)} = 0$ satisfies

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{\left\|\mathbf{w}^\star\right\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \left\|\mathbf{v}_t\right\|^2 \tag{4}$$

3. (3 points) Let's now analyze the convergence rate of gradient descent *i.e.* how fast it converges to $\mathbf{w}^\star$. Show that for $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

$$f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \leq \frac{1}{T} \sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \nabla f(\mathbf{w}^{(t)}) \rangle \quad ??? \tag{5}$$

Further, use the result from part 2, with upper bounds B and ρ for $\|\mathbf{w}^\star\|$ and $\left\|\nabla f(\mathbf{w}^{(t)})\right\|$ respectively and show that for fixed $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, the convergence rate of gradient descent is $\mathcal{O}(1/\sqrt{T})$ *i.e.* the upper bound for $f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \propto \frac{1}{\sqrt{T}}$.

4. (2 points) Consider a objective function comprised of $N = 2$ terms:

$$f(w) = \frac{1}{2}(w-2)^2 + \frac{1}{2}(w+1)^2 \tag{6}$$

In order to find

$$k = \underset{w}{\operatorname{argmin}} f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle + \frac{\lambda}{2} \|w - w^{(t)}\|^2.$$

$$\hookrightarrow \sqrt{\langle w - w^{(t)}, w - w^{(t)} \rangle}$$

$$\frac{\partial k}{\partial w} = \nabla f(w^{(t)}) + \lambda . \frac{((w - w^{(t)})^2)^{1/2}}{\partial w}$$

$$= \frac{1}{2} \lambda . ((w - w^{(t)})^2)^{-\frac{1}{2}} . 2(w - w^{(t)})$$

$$= \lambda \frac{w - w^{(t)}}{\sqrt{(w - w^t)^2}} = \lambda \frac{w - w^t}{\|w - w^T\|}$$

$$\Rightarrow \frac{\partial k}{\partial w} = \nabla f(w^{(t)}) + \lambda \frac{w - w^t}{\|w - w^{(t)}\|}$$

$\Rightarrow$ min when $\nabla f(w^{(t)}) = -\lambda \frac{w - w^{(t)}}{\|w - w^{(t)}\|}$

$$\Rightarrow \nabla f(w^{(t)}) + \lambda (w - w^{(t)}) = 0$$

$$\Rightarrow w = w^{(t)} - \frac{1}{\lambda} . \nabla f(w^{(t)})$$

$\hookrightarrow$ The same as gradient descent

Seq $v_1, v_2, \ldots, v_t$ and w^* min $f(w)$

$$w^{(t+1)} = w^{(t)} - \mu v_t \quad \text{with } w^{(0)} = 0.$$

$$\underbrace{\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle}_{lp} \leq \frac{\|w^*\|^2}{2\mu} + \frac{\mu}{2} \sum_{t=1}^{T} \|v_t\|^2$$

$$lp = \sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \leq \sum_{t=1}^{T}$$

$$v_t = \frac{w^{(t)} - w^{(t+1)}}{\mu}$$

$$= \sum_{t=1}^{T} \left(\langle w^{(t)}, v_t \rangle - \langle w^*, v_t \rangle \right)$$

$$0 - w^*$$

$$0 - \mu v_{t+1} - w^*.$$

$=$

$$w^{(2)} = 0 - \mu v_1$$

$$w^{(3)} = 0 - \mu v_1 - \mu v_{2} - \mu.$$

$$w^{(4)} = \cancel{\#} - \mu \cdot \sum v_t$$

$$w^{(t)} - w^{*} = (0 - w^{*})(v_1)$$

$$+ (0 - \mu v_1 - w^{*})(v_2)$$

$$+ (0 - \mu v_1 - \mu v_2 - w^{*})(v_3)$$

$$\frac{\sqrt{\langle w^*, w^* \rangle}^2}{2\mu} + \frac{\mu}{2} \sum_{t=1}^{T} \|v_t\|^2 .$$

$$\frac{\|w^*\|^2}{2\mu} + \frac{\mu}{2} \sum_{T=1}^{T} \|v_t\|^2 \geq$$

$$\frac{1}{2} \sum_{t=1}^{T} \|v_t\| . \|w^*\|$$

Question 2:

$$w^{(t+1)} = w^{(t)} - \mu v_t \quad \text{with } w^{(1)} = 0$$

$$\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \leq \frac{\|w^*\|^2}{2\mu} + \frac{\mu}{2} \sum_{t=1}^{T} \|v_t\|^2$$

Let $D(w^{(t)}, w^*) = \langle w^{(t)} - w^*, w^{(t)} - w^* \rangle = (w^{(t)} - w^*)^T \cdot (w^{(t)} - w^*)$

$$\Rightarrow \Delta D = D(w^{(t+1)}, w^*) - D(w^{(t)}, w^*)$$

$$= D(w^{(t)} - \mu v_t, w^*) - D(w^{(t)}, w^*)$$

$$= (w^{(t)} - \mu v_t - w^*)^T \cdot (w^{(t)} - \mu v_t - w^*) - (w^{(t)} - w^*)^T (w^{(t)} - w^*)$$

$\times 2$ $\times 2$

$$= + \mu \|v_t\|^2 + 2\mu v_t \cdot w^* - 2\mu v_t \cdot w^{(t)}$$

$$= + \mu^2 \|v_t\|^2 - 2\mu \langle w^{(t)} - w^*, v_t \rangle$$

$$\Rightarrow \sum_{t=1}^{T} \Delta D = \boxed{\begin{aligned} & D(w^{(T)}, w^*) - D(w^{(1)}, w^*) \\ & = \sum_{t=1}^{T} -\mu^2 \|v_t\|^2 - \sum_{t=1}^{T} 2\mu \langle w^{(t)} - w^*, v_t \rangle \end{aligned}}$$

$$\Rightarrow D(w^{(T)}, w^*) - D(w^{(1)}, w^{(*)}) = \sum_{t=1}^{T} \mu^2 \|v_t\|^2 - \sum_{t=1}^{T} 2\mu \langle w^{(t)} - w^*, v_t \rangle$$

$$\Leftrightarrow \sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle = \frac{1}{2\eta} D(w^{(1)}, w^{(*)}) + \sum_{t=1}^{T} \frac{\eta}{2} \|v_T\|^2$$

As $w^{(1)} = 0$ and $D > 0$

$$\Leftrightarrow \sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \leq \frac{1}{2\eta} (0 - w^*)^T (0 - w^*) + \sum_{t=1}^{T} \frac{\eta}{2} \|v_t\|^2$$

$$\leq \frac{1}{2\eta} \|w^*\|^2 + \sum_{t=1}^{T} \frac{\eta}{2} \|v_t\|^2$$

<u>Question 3:</u> $\bar{w} = \frac{1}{T} \cdot \sum_{t=1}^{T} w^{(t)}$

$$f(\bar{w}) - f(w^*) \underset{=}{\leq} \frac{1}{T} \sum_{t=1}^{T} \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$$

$$\nabla f(w^{(t)}) . w^{(t)}$$

$$w^{(t+1)} = w^{(t)} - \eta \nabla f(w^{(t)})$$

$$\partial \quad f(w) = f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle$$

$$\boxed{f(\bar{w}) - f(w^*)} = \frac{1}{T} \sum_{t=1}^{T} \left(\nabla f(w^{(t)}) . w^{(t)} - \nabla f(w^{(t)}) . w^* \right)$$

$$f(\bar{w}) - f(w^*) \le \frac{1}{T}\sum_{t=1}^{T} \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$$

$f(w^{(t)})$ $\quad f(w^{(1)}) - f(w^{T+1}) = f(w^{(1)}) - f(w^{*})$

$$f(w) \ge f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle$$

$- \langle w^{(t+1)} - w^{(t)}, \nabla f$

$$\frac{1}{T}\sum_{t=1}^{T} f(w^{(t)}) - \frac{1}{T}\sum f(w^{(t+1)}) =$$

$$\langle w^{(t)} - w, \nabla f(w^{(t)}) \rangle \ge f(w^{(t)}) - f(w)$$

$\sum w^0$

$$\frac{1}{T}\sum_{t=1}^{T} f(w^{(t)}) - f(w) = \qquad f(w^0) - f(w^{T+1})$$

$f(0) - f(1) + f(1) - f(2) \quad 0$

$\approx f(w^0) - f(w^T) - \langle w^{(T+1)} - w^T, \nabla f(w^{(t)}) \rangle$

$\Rightarrow f(w^0) - f(w^T)$

$w^{(t+1)} = w^{(t)} - \mu \nabla f(w^{(t)})$

$w^{(t+2)} + w^{(t)}$

$w^{(t+2)} = \mu\nabla f(w^{(t)}) + w^{(t)}$

$= w^{(t)} - \mu \nabla f(w^{(t)}) \cancel{- \mu \nabla f(w^0)}$

$= T \cdot w^{(t)} +$

$= T \cdot w^{(t)}$

$w^{(t+1)} = w^{(t)} - \mu v_t$

$= w^{(t)} - \mu v$

$(w^{(t)}) - \mu v - \dots f(w$

$\nabla f(w^{(1)}) = w^{(t+1)} - w^{(1)}$

$\nabla f(w^{2}) = w^{(2)} - w^{(1)}$

$w^{(1)} = w^0 + \boxed{\nabla f(w^{(0)})}$

$w^{(2)} = w^{(0)} + \boxed{\nabla f(w^{(0)})} - \nabla f(w^{(1)})$

$w^{(0)}$ $\quad -\mu v$ $\quad w^0$

w_0

$w^{(n)} =$

$$\frac{1}{T} w^0 + \mu v_t$$

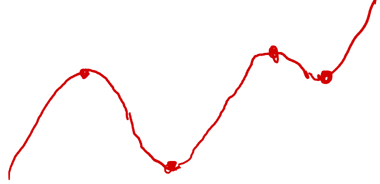
$$\frac{1}{T} \cdot w^0 + \mu$$

$w^{(1)} = w^{0} - \nabla f(w^{(0)})$ $\quad w' + w' + w$

$w^{(2)} = w^{(1)} - \nabla f(w^{(1)})$ $\quad -\nabla f(w^{0}) \nabla f(w^{(1)})$

$w^{(t)}$ $\quad w^{(1)} - w^{(0)} - w^{(2)} + w^{(2)}$

$f(\bar{w}) = f\left(\frac{1}{T}\right.$ $\quad \Leftrightarrow\; w^{(t)} - w^{(0)}$



Now consider using SGD (with a batch-size $B = 1$) to minimize this objective. Specifically, in each iteration, we will pick one of the two terms (uniformly at random), and take a step in the direction of the negative gradient, with a constant step-size of η. You can assume η is small enough that every update does result in improvement (aka descent) on the sampled term.

Is SGD guaranteed to decrease the overall loss function in every iteration? If yes, provide a proof. If no, provide a counter-example.

2 Automatic Differentiation

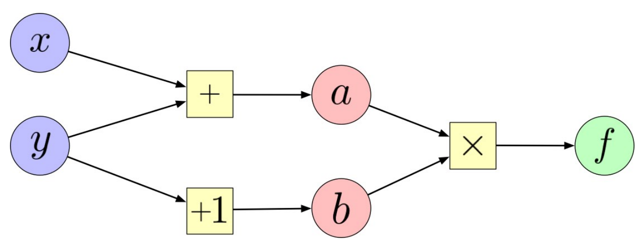
5. (4 points) In practice, writing the closed-form expression of the derivative of a loss function f w.r.t. the parameters of a deep neural network is hard (and mostly unnecessary) as f becomes complex. Instead, we define computation graphs and use the automatic differentiation algorithms (typically backpropagation) to compute gradients using the chain rule. For example, consider the expression

$$f(x, y) = (x + y)(y + 1) \tag{7}$$

Let's define intermediate variables a and b such that

$$a = x + y \tag{8}$$
$$b = y + 1 \tag{9}$$
$$f = a \times b \tag{10}$$

A computation graph for the "forward pass" through f looks like the following



We can then work backwards and compute the derivative of f w.r.t. each intermediate variable ($\frac{\partial f}{\partial a}$, $\frac{\partial f}{\partial b}$) and chain them together to get $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$.

Let $\sigma(\cdot)$ denote the standard sigmoid function. Now, for the following vector function:

$$f_1(w_1, w_2) = e^{e^{w_1} + e^{2w_2}} + \sin(e^{w_1} + e^{2w_2}) \tag{11}$$
$$f_2(w_1, w_2) = w_1 w_2 + \sigma(w_1) \tag{12}$$

(a) Draw the computation graph. Compute the value of f at $\vec{w} = (1, 2)$.

(b) At this $\vec{w}$, compute the Jacobian $\frac{\partial \vec{f}}{\partial \vec{w}}$ using numerical differentiation (using $\Delta w = 0.01$).

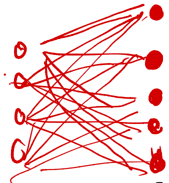
(c) At this $\vec{w}$, compute the Jacobian using forward mode auto-differentiation.

(d) At this $\vec{w}$, compute the Jacobian using backward mode auto-differentiation.

(e) Don't you love that software exists to do this for us?

$W_{c_2}[\quad]^{c_1}$

$\vec{h}^l = W \cdot \vec{h}^{(l-1)}$ 3 $\Rightarrow c_2[\quad]^1 = c_2 \times c_1 \cdot c_1[\quad]^1$



$W \cdot \vec{h}^{\,l-1}$

$[h^{l}_{(1)} \dots h^{l}_{(n)}] . W$

$1 \times C_1$

$h^{(l)}_1 = W_1 \cdot \vec{h}^{\,(l-1)}$

$\frac{\partial L}{\partial \vec{h}^{\,l-1}} = [h^{l-1}_{(1)} \dots] \underset{C_2}{\overset{C_1}{[\ \ ]}}$ $1 \times C_2$

$\frac{\partial L}{\partial \vec{h}^{\,l}} \cdot \frac{\partial \vec{h}^{\,l}}{\partial \vec{h}^{\,l-1}} = 1 \times C_2 \; C_2 \times C_1 = 1 \times C_1$

3 Implement and train a network on CIFAR-10

6. (Upto 26 points) In this homework, we will learn how to implement backpropagation (or backprop) for vanilla neural networks (or Multi-Layer Perceptrons) and ConvNets. You will begin by writing the forward and backward passes for different types of layers (including convolution and pooling), and then go on to train a shallow ConvNet on the CIFAR-10 dataset in Python. Next you will learn to use PyTorch, a popular open-source deep learning framework, and use it to replicate the experiments from before.

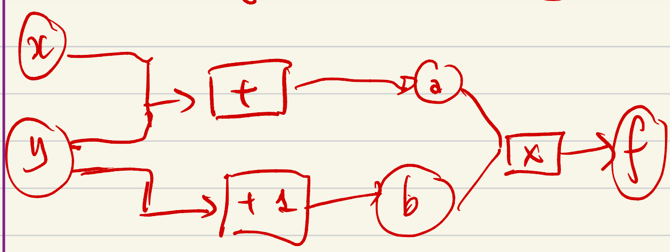
   Follow the instructions provided here: www.cc.gatech.edu/classes/AY2019/cs7643_fall/hw1-q6/

2. AUTOMATIC DIFFERENTIATION

5) $f(x,y) = (x+y)(y+1)$

$a = x + y \qquad b = y + 1$

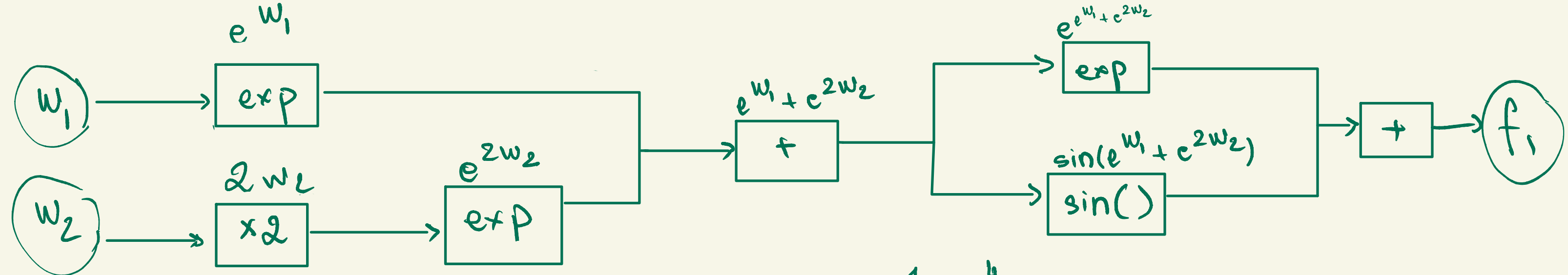
Example



$$f_1(w_1, w_2) = e^{e^{w_1} + e^{2w_2}} + \sin(e^{w_1} + e^{2w_2})$$

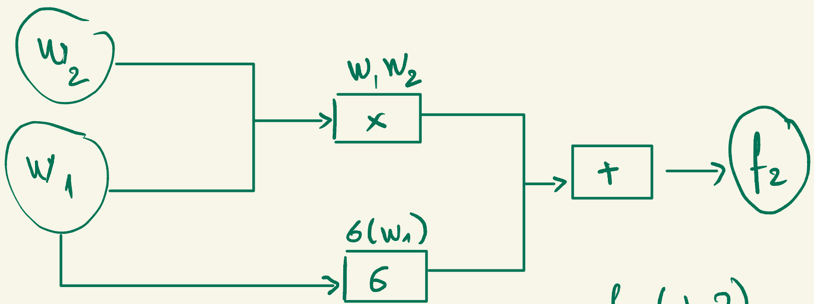
$$f_2(w_1, w_2) = w_1 \cdot w_2 + \sigma(w_1)$$

f_1



$$f_1(1,2) = e^{e^1+e^4} + \sin(e^1+e^4) \approx 7{,}802 \times 10^{24}$$

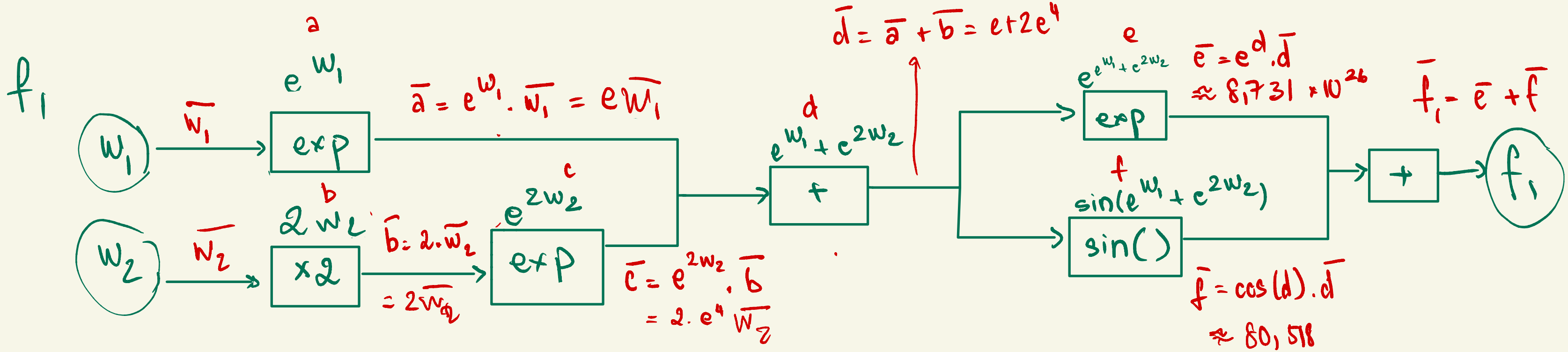
f_2

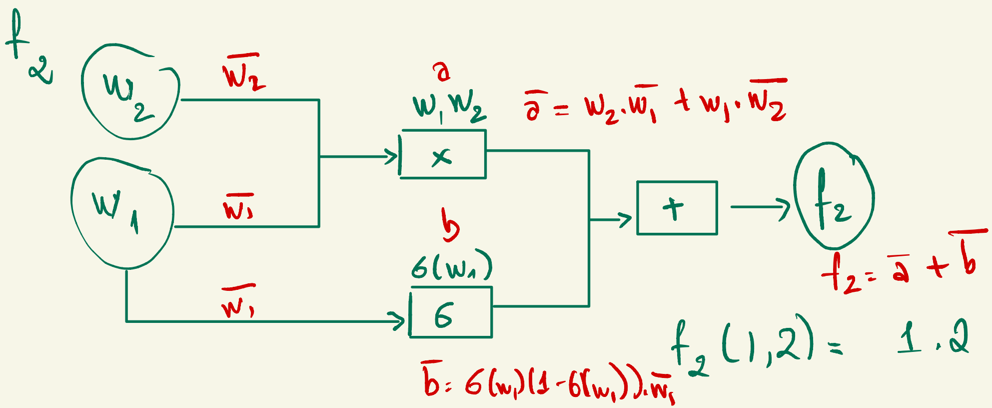


sigmoid: $\dfrac{1}{1+e^{-x}}$

$$f_2(1,2) = 1 \cdot 2 + \sigma(1) \approx 2{,}731$$

c)





sigmoid: $\frac{1}{1+e^{-x}}$

$f_2(1,2) = 1.2 + 6(1)$

b) Numerical differentation

$$\frac{\partial \vec{f}}{\partial \vec{w}} = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_2}{\partial w_2} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} \end{bmatrix}$$

$$\frac{\partial f_1}{\partial w_1} = \frac{f_1(\Delta w_1 + w_1) - f_1(w_1)}{\Delta w_1} =$$

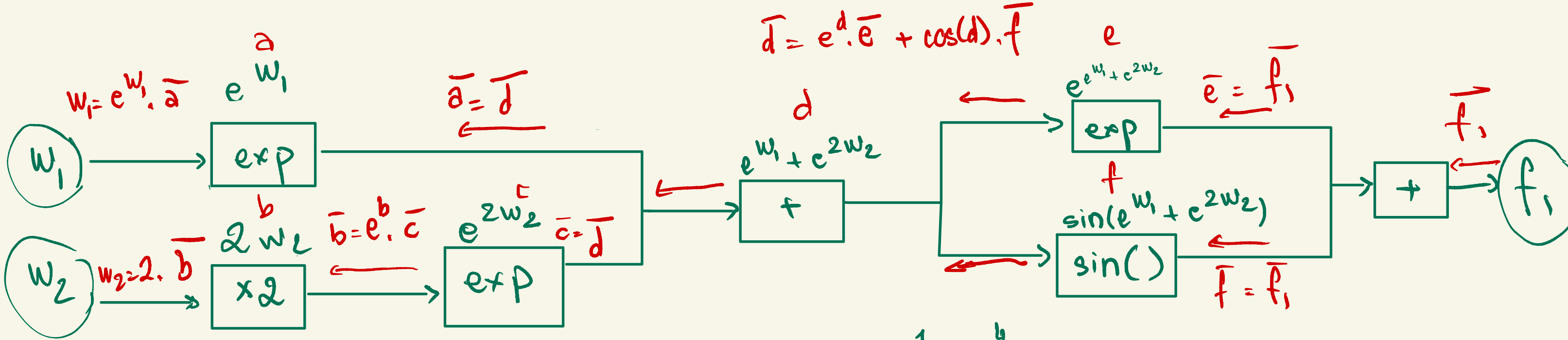
$$\frac{\partial f_1}{\partial w_2} = \frac{f_1(\Delta w_2 + w_2) - f_1(w_2)}{\Delta w_2} =$$

$$\frac{\partial f_2}{\partial w_1} = \frac{f_2(\Delta w_1 + w_1) - f_2(w_1)}{\Delta w_1} =$$

$$\frac{\partial f_2}{\partial w_2} = \frac{f_2(\Delta w_2 + w_2) - f_2(w_2)}{\Delta w_2} =$$

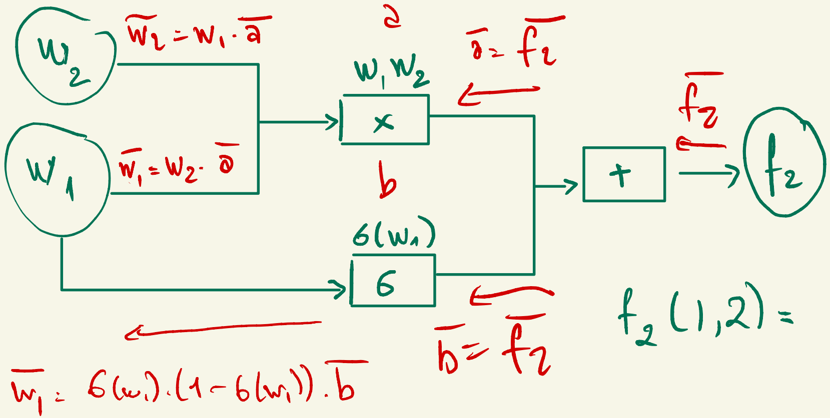
d)

f_1



$$f_1(1,2) = e^{e^1 + e^4} + \sin(e^1 + e^4) \approx 7{,}802 \times 10^{24}$$

f_2



sigmoid: $\frac{1}{1 + e^{-x}}$

$$f_2(1,2) = 1 \cdot 2 + \sigma(1)$$

$$\frac{\partial b}{\partial w_1}$$

Binomial:

forward mode

$\vec{h}^{(l-1)}$

$g(.)$

$\vec{h}^{(l)} = g(\vec{h}^{(l-1)})$

Input $\boxed{\dfrac{\partial h^{(l-1)}}{\partial \vec{x}}}$

$\dfrac{\partial h^{(l)}}{\partial \vec{x}} = \partial h^{(l)}$

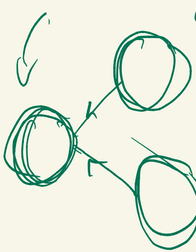
$\dfrac{\partial L}{\partial x} = \dots$

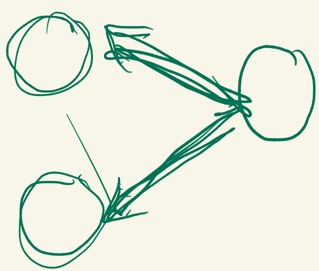
$\dfrac{\partial L}{\partial x}$

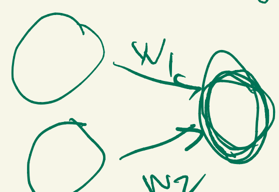
$\dfrac{\partial h}{\partial w_1} + \dfrac{\partial h}{\partial w_2}$

$W = x_1 - x_2$

d_h

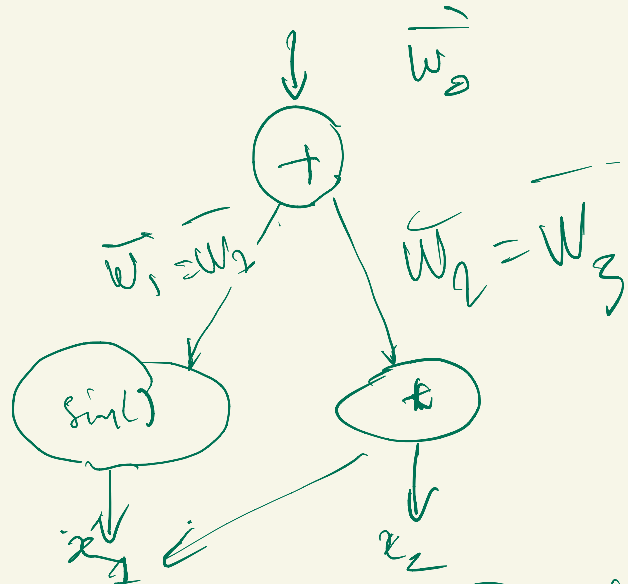






$\dfrac{\partial h}{\partial w_1 + w_2} \cancel{+} \dfrac{\partial h}{\partial w_2}$

$$f(x_1, x_2) = \sin(x_1) + x_{\cancel{2}} \cdot x_2$$



$$\bar{w}_3 \cdot \cos(x_1) + \bar{w}_3 \cdot x_2$$

$$\bar{w}_3 \cdot (x_1)$$