Lecture 7

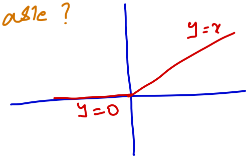
Math Foundations Team

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

e.g: is ReLU continuous? is it differentiable?

$$\text{ReLU}(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$



Consider at $x = 0$

$$\lim_{x \to 0^-} \text{ReLU}(x) = 0 \qquad \lim_{x \to 0^+} \text{ReLU}(x) = \lim_{x \to 0^+} x = 0$$

$$\therefore \lim_{x \to 0^-} \text{ReLU}(x) = \lim_{x \to 0^+} \text{ReLU}(x) = \text{ReLU}(0) \Rightarrow \text{Continuous}$$

Consider at $x = 0$:

$$f'(x) = \lim_{h \to 0^+} \frac{\text{ReLU}(x+h) - \text{ReLU}(x)}{h} = \lim_{h \to 0^+} \frac{h - 0}{h} = 1$$

$$f'(x) = \lim_{h \to 0^-} \frac{\text{ReLU}(x+h) - \text{ReLU}(x)}{h} = \lim_{h \to 0^-} \frac{0 - 0}{h} = 0$$

$\Rightarrow$ NOT Differentiable

Mid sem: 30 marks

90 mins

$\rightarrow$ 4 questions

8 $\leftarrow$

8 $\leftarrow$

8 $\leftarrow$

6 $\leftarrow$

e.g.: if $f(x)$ is represented using 1^{st} order Taylor Polynomial at $x = 1$, what is the approximation error at 1.1?

$f(x) = (x-2)^2 + 3$

$$f(x) = (x-2)^2 + 3$$

1^{st} approx: $f(x) = f(x)\big|_{x=a} + f'(x)\big|_{x=a}(x-a)$

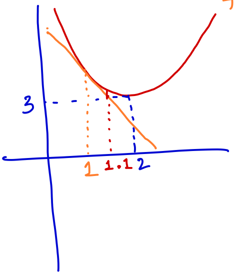
given $a = 1$

$$\tilde{f}(x) = f(1) + f'(x)\big|_{x=1}(x-1)$$

$$= (1-2)^2 + 3 + 2(x-2)\big|_{x=1}(x-1)$$

$$= 1 + 3 + 2(-1)(x-1) = 4 - 2(x-1)$$

$$= 4 - 2x + 2 = 6 - 2x$$



$$\tilde{f}(x)\big|_{1.1} = 6 - 2(1.1) = 3.8$$

$$f(x)\big|_{1.1} = (1.1-2)^2 + 3 = 0.81 + 3 = 3.81$$

error $= 0.01$

e.g. $f(x) = \begin{bmatrix} \frac{1}{2}(x - \tilde{x})^2 \\ x^2 \end{bmatrix}$

$\mathcal{L}$ = MSE + L2 Regularizer

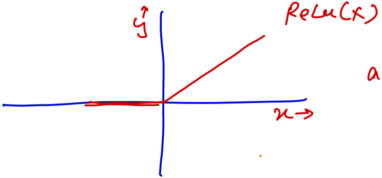
$f : R \rightarrow R^2$

$$\frac{df}{dx} = \begin{array}{cc} & \#\,i/p \\ \#\,o/p & \begin{bmatrix} \frac{df_1}{dx} \\ \frac{df_2}{dx} \end{bmatrix} \end{array} = \begin{bmatrix} \frac{d}{dx}\left(\frac{1}{2}(x - \tilde{x})^2\right) \\ \frac{d}{dx}\left(x^2\right) \end{bmatrix}$$

$$= \begin{bmatrix} (x - \tilde{x}) \\ 2x \end{bmatrix}$$

Normally x will be a vector in neural networks, as we will see.

x	o/p
3	3
2	2
-1	0
0	0
5	5
7	7
-100	0



activation fn
Neural networks

Introduction

- In last lecture, we discussed about differentiation of univariate functions, partial differentiation, gradients and gradients of vector valued functions.
- Now we will look into gradients of matrices and some useful identities for computing gradients.
- Finally, we will discuss back propagation and automatic differentiation.

Gradients of Matrices

The gradient of an $m \times n$ matrix A with respect to a $p \times q$ matrix B, the resulting Jacobian would be an $(m \times n) \times (p \times q)$, i.e., a four-dimensional tensor J, whose entries are given as

$$J_{ijkl} = \frac{\partial A_{ij}}{\partial B_{kl}}$$

Since, we can consider $\mathbb{R}^{m \times n}$ as $\mathbb{R}^{mn}$, we can shape our matrix into vectors of length mn and pq respectively. The gradient using mn vectors results in a Jacobian of size $mn \times pq$

$f: R^n \to R$ $\quad \nabla f \in R^{1 \times n}$ i/p o/p

$f: R \to R^m$ $\quad \nabla f \in R^{m \times 1}$

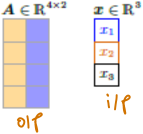
$f: R^n \to R^m$ $\quad \nabla f \in R^{m \times n}$ i/p o/p

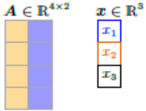
$f: \underset{i/p}{R^{(p \times q)}} \to \underset{o/p}{R^{(m \times n)}}$

$\nabla f \in R^{(mn) \times (pq)}$ o/p i/p

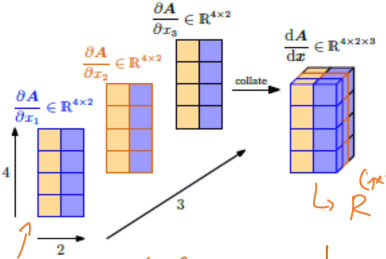
Gradients of Matrices

$f: \mathbb{R}^3 \rightarrow \mathbb{R}^{(4\times 2)}$

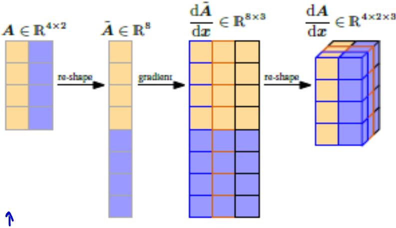




Partial derivatives:



$\hookrightarrow \mathbb{R}^{(4\times 2)\times 3}$



$\hookrightarrow$ 4x2 rows (o/p)
3 cols (i/p)

$\triangleleft$ o/p w.r.t first comp of i/p.

$f_1: \mathbb{R} \rightarrow \mathbb{R}^{4\times 2}$

$\triangleleft f_1 \in \mathbb{R}^{(4\times 2)\times 1}$

Gradients of Matrices

Let $f = Ax$ where $A \in \mathbb{R}^{m \times n}$, and $x \in \mathbb{R}^n$, then

$$\frac{\partial f}{\boxed{\partial A}} \in \mathbb{R}^{m \times \underbrace{(m \times n)}_{\text{grad is w.r.t } A}}$$

$[A]_{m \times n}$ $[x]_{n \times 1}$

$f : R^n \to R^m$

$\nabla_A f \in R^{m \times (m \times n)}$

dim of o/p

dim of A

By definition

$$\frac{\partial f}{\partial A} = \begin{bmatrix} \frac{\partial f_1}{\partial A} \\ \vdots \\ \frac{\partial f_m}{\partial A} \end{bmatrix}, \frac{\partial f_i}{\partial A} \in \mathbb{R}^{1 \times (m \times n)}$$

o/p $\in R^m$

$\therefore$ has m rows

each elem is grad wrt $A \in R^{m \times n}$

Consider 2x2 example:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

RECALL

$$f : R^2 \to R^2, \quad \nabla_x f = {}_{o/p}\overset{i/p}{\begin{bmatrix} \quad \end{bmatrix}} = \begin{bmatrix} \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} y_1 \\ \frac{\partial}{\partial x_2} y_2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial}{\partial x_1}(a_{11}x_1 + a_{12}x_2) & \frac{\partial}{\partial x_2}(a_{11}x_1 + a_{12}x_2) \\ \frac{\partial}{\partial x_1}(a_{21}x_1 + a_{22}x_2) & \frac{\partial}{\partial x_2}(a_{21}x_1 + a_{22}x_2) \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{A}$$

$$\nabla_A f = {}_{o/p}\overset{i/p}{\begin{bmatrix} \quad \end{bmatrix}} = \begin{bmatrix} \frac{df_1}{dA} \\ \frac{df_2}{dA} \end{bmatrix} = \begin{bmatrix} \frac{dy_1}{dA} \\ \frac{dy_2}{dA} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \frac{d}{da_{11}}(a_{11}x_1 + a_{12}x_2) & \frac{d}{da_{12}}(a_{11}x_1 + a_{12}x_2) \\ \frac{d}{da_{21}}(a_{11}x_1 + a_{12}x_2) & \frac{d}{da_{22}}(a_{11}x_1 + a_{12}x_2) \end{bmatrix} \\ \begin{bmatrix} \frac{d}{da_{11}}(a_{21}x_1 + a_{22}x_2) & \frac{d}{da_{12}}(a_{21}x_1 + a_{22}x_2) \\ \frac{d}{da_{21}}(a_{21}x_1 + a_{22}x_2) & \frac{d}{da_{22}}(a_{21}x_1 + a_{22}x_2) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x_1 & x_2 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ x_1 & x_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x^T \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ x^T \end{bmatrix} \end{bmatrix}$$

Gradients of Matrices

Now, we have

$$f_i = \sum_{j=1}^{n} A_{ij} x_j, \, i = 1, \cdots, m.$$

Therefore, by taking partial derivatives with respect to A_{iq}

$$\frac{\partial f_i}{\partial A_{iq}} = x_q.$$

Hence, i^{th} row becomes

$$\frac{\partial f_i}{\partial A_{i,:}} = x^T \in \mathbb{R}^{1 \times 1 \times n}$$

$$\frac{\partial f_i}{\partial A_{k,:}} = 0^T \in \mathbb{R}^{1\times 1\times n}, \text{for} k \neq i$$

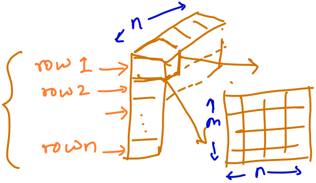
Hence, by stacking the partial derivatives, we get

$$\frac{\partial f_i}{\partial A_{k,:}} = \begin{bmatrix} 0^T \\ \vdots \\ x^T \\ \vdots \\ 0^T \end{bmatrix} \in \mathbb{R}^{1\times m\times n}$$

e.g. $\quad \underset{R^{n\times n}}{f} = \underset{R^{n\times m}}{B^T}\,\underset{R^{m\times n}}{B} = K \qquad f: R^{m\times n} \longrightarrow R^{n\times n} \qquad$ find $\dfrac{dK}{dB}$

$$\nabla_B K = \frac{dK}{dB} = \underset{(n\times n)}{\text{o/p}} \overset{\text{i/p }(m\times n)}{\left[\qquad\qquad\right]} \longrightarrow$$

n×n rows & each row element has m×n entries.



Consider a 2x2 matrix:

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \qquad K = \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$K = \begin{bmatrix} \overbrace{b_{11}^2 + b_{21}^2}^{K_{11}} & \overbrace{b_{11} b_{12} + b_{21} b_{22}}^{K_{12}} \\ b_{12}b_{11} + b_{22}b_{21} & b_{12}^2 + b_{22}^2 \end{bmatrix} = \underbrace{\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}}_{\text{o/p components}}$$

$$\nabla_B K = \begin{bmatrix} \begin{bmatrix} \frac{dK_{11}}{dB} & \frac{dK_{12}}{dB} \\ \frac{dK_{21}}{dB} & \frac{dK_{22}}{dB} \end{bmatrix} \end{bmatrix} =$$

$(2\times2)\times ?$

$$\begin{bmatrix} \overset{p=1 \quad q=1}{\begin{bmatrix} \underset{i=1\ j=1}{\frac{\partial K_{11}}{\partial b_{11}}} & \underset{i=1\ j=2}{\frac{\partial K_{11}}{\partial b_{12}}} \\ \underset{i=2\ j=1}{\frac{\partial K_{11}}{\partial b_{21}}} & \underset{i=2\ j=2}{\frac{\partial K_{11}}{\partial b_{22}}} \end{bmatrix}} & \overset{p=1 \quad q=2}{\begin{bmatrix} \underset{i=0\ j=0}{\frac{\partial K_{12}}{\partial b_{11}}} & - \\ - & - \end{bmatrix}} \\ \overset{p=2 \quad q=1}{\begin{bmatrix} \frac{\partial K_{21}}{\partial b_{11}} & - \\ - & - \end{bmatrix}} & \overset{p=2 \quad q=2}{\begin{bmatrix} \frac{\partial K_{22}}{\partial b_{11}} & - \\ - & - \end{bmatrix}} \end{bmatrix}$$

2×2

$$= \begin{bmatrix} \begin{bmatrix} 2b_{11} & 0 \\ 2b_{21} & 0 \end{bmatrix} & \begin{bmatrix} b_{12} & b_{11} \\ b_{22} & b_{21} \end{bmatrix} \\ \begin{bmatrix} b_{12} & b_{11} \\ b_{22} & b_{21} \end{bmatrix} & \begin{bmatrix} 0 & 2b_{12} \\ 0 & 2b_{22} \end{bmatrix} \end{bmatrix}$$

$(2\times2)\times(2\times2)$

$K_{pq} \rightarrow (p,q)$ entry of o/p K

$b_{ij} \rightarrow (i,j)$ entry of i/p B

Gradients of Matrices with respect to Matrices

Let $B \in \mathbb{R}^{m\times n}$ and $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{n\times n}$ with

$$f(B) = B^T B =: K \in \mathbb{R}^{n\times n}$$

Then, we have

$$\frac{\partial K}{\partial B} \in \mathbb{R}^{(n\times n)\times(m\times n)}.$$

Moreover

$$\frac{\partial K_{pq}}{\partial B} \in \mathbb{R}^{1\times(m\times n)}, \text{for} p, q = 1, \cdots n$$

where K_{pq} is the $(p, q)^{th}$ entry of $K = f(B)$

Gradients of Matrices with respect to Matrices

Let i^{th} column of B be b_i, then

$$K_{pq} = r_p^T r_q = \sum_{l=1}^{m} B_{lp} B_{lq}$$

Computing the partial derivative, we get

$$\frac{\partial K_{pq}}{\partial B_{ij}} = \sum_{l=1}^{m} \frac{\partial}{\partial B_{ij}} B_{lp} B_{lq} = \partial_{pqij}$$

Clearly, we have

$$\begin{aligned} \partial_{pqij} &= B_{iq} \quad \text{if } j = p, p \neq q \\ \partial_{pqij} &= B_{ip} \quad \text{if } j = q, p \neq q \\ \partial_{pqij} &= 2B_{iq} \quad \text{if } j = p, p = q \\ \partial_{pqij} &= 0 \quad \text{otherwise} \end{aligned}$$

general form for the (2×2) solution

where $p, q, j = 1, \cdots, n \;\; i = 1, \cdots, m$

Useful Identities for Computing Gradients

- $\frac{\partial}{\partial X} f(X)^T = (\frac{\partial f(X)}{\partial X})^T$
- $\frac{\partial}{\partial X} tr(f(X)) = tr(\frac{\partial f(X)}{\partial X})$
- $\frac{\partial}{\partial X} det(f(X)) = det(f(X)) tr(f(X)^{-1} \frac{\partial f(X)}{\partial X})$
- $\frac{\partial}{\partial X} f(X)^{-1} = -f(X)^{-1} \frac{\partial f(X)}{\partial X} f(X)^{-1}$

Useful Identities for Computing Gradients

- $\frac{\partial a^T X^{-1} b}{\partial X} = -(X^{-1})^T a b^T (X^{-1})^T$
- $\frac{\partial x^T a}{\partial x} = a^T$
- $\frac{\partial a^T x}{\partial x} = a^T$
- $\frac{\partial a^T X b}{\partial X} = a b^T$
- $\frac{\partial x^T B x}{\partial x} = x^T (B + B^T)$
- $\frac{\partial}{\partial s}(x - As)^T W (x - As) = -2(x - As)^T W A$

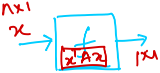
for symmetric W.

e.g: $\frac{\partial}{\partial x}(a^T x) = ?$ $\qquad [\quad]_{1\times n}[\quad]_{n\times 1} = [\quad]_{1\times 1}$

$$\nabla_x f = 1\left[\quad \overset{n\times 1}{} \quad\right] = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n}\right]$$

$$= \left[\frac{\partial}{\partial x_1}(a_1x_1 + a_2x_2 + \cdots) \quad \frac{\partial}{\partial x_2}(a_1x_1 + \cdots) \quad \cdots\right]$$

$$= \left[a_1 \quad a_2 \quad a_3 \quad \cdots\right] = a^T$$



e.g. $\frac{\partial}{\partial x}(x^T A x)$ $\qquad [x^T]_{1\times n}[A]_{n\times n}[x]_{n\times 1} = [1\times 1]$

$$\nabla_x f = 1\left[\quad \overset{n\times 1}{} \quad\right] = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n}\right]$$

Consider 2×2:

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix} = \begin{aligned} &x_1 x_1 a_{11} + \\ &x_1 x_2 a_{12} + \\ &x_2 x_1 a_{21} + \\ &x_2 x_2 a_{22}. \end{aligned}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j a_{ij}$$

$$\frac{\partial f}{\partial x_1} = 2x_1 a_{11} + x_2 a_{12} + x_2 a_{21} = 2x_1 a_{11} + x_2 (a_{12} + a_{21})$$

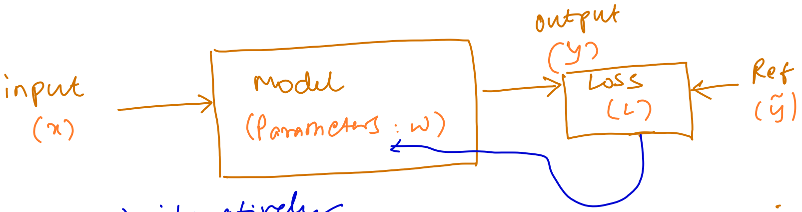
$$\frac{\partial f}{\partial x_2} = 2x_2 a_{22} + x_1 a_{12} + x_2 a_{21} = 2x_2 a_{22} + x_1 (a_{12} + a_{21})$$

$$\nabla_x f = \begin{bmatrix} 2x_1 a_{11} + x_2 (a_{12} + a_{21}) & 2x_2 a_{22} + x_1 (a_{12} + a_{21}) \end{bmatrix}$$

$$\Updownarrow$$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} + a_{11} & a_{12} + a_{21} \\ a_{21} + a_{12} & a_{22} + a_{22} \end{bmatrix}$$

$$= x^T (A + A^T)$$



Update w iteratively
So that arg Loss (L) is minimized

Back Propagation = Gradient Computation "Chain Rule" + Parameter update "Gradient Descent"

Backpropagation and Automatic Differentiation

Consider the function

$$f(x) = \sqrt{x^2 + exp(x^2)} + cos(x^2 + exp(x^2))$$

Taking dervatives

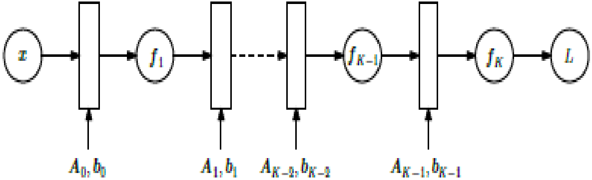
$$\frac{d}{dx}(f(x))^n = n.(f(x))^{n-1} . \frac{d}{dx}(f(x))$$

$$\begin{aligned}
\frac{df}{dx} &= \frac{2x + 2xexp(x^2)}{2\sqrt{x^2 + exp(x^2)}} - sin(x^2 + exp(x^2))(2x + 2xexp(x^2)) \\
&= 2x(\frac{1}{2\sqrt{x^2 + exp(x^2)}} - sin(x^2 + exp(x^2)))(1 + exp(x^2))
\end{aligned}$$

Motivation

- The implementation of the gradient could be significantly more expensive than computing the function, which imposes unnecessary overhead where we get such lengthy expressions.
- We need an efficient way to compute the gradient of an error function with respect to the parameters of the model.
- For training deep neural network models, the backpropagation algorithm is one such method.

In neural networks with multiple layers



$$f_i(x_{i-1}) = \sigma(A_{i-1}x_{i-1} + b_{i-1})$$

where x_{i-1} is the output of layer $i-1$ and σ is an activation function.

Backpropagation

To train these model, the gradient of the loss function L with respect to all model parameters $\theta_j = \{A_j, b_j\}, j = 1, \cdots, K$ and inputs of each layer needs to be computed. Consider,

$$\begin{aligned} f_0 &:= x \\ f_i &:= \sigma_i(A_{i-1} f_{i-1} + b_{i-1}), i = 1, \cdots, K. \end{aligned}$$

We have to find $\theta_j = \{A_j, b_j\}, j = 1, \cdots, K-1$ such that

$$L(\theta) = ||y - f_K(\theta, x)||^2$$

is minimum where $\theta = \{A_0, b_0, \cdots, A_{K-1}, b_{K-1}\}$

$$x \longrightarrow f_0 \longrightarrow \underbrace{\underset{A_0 f_0 + b_0}{\mathrm{Lin}(\cdot)} \rightarrow g_0 \rightarrow \underset{\frac{1}{1+e^{-g_0}}}{\sigma(\cdot)} \longrightarrow}_{} f_1 \rightarrow \underbrace{\underset{A_1 f_1 + b_1}{\mathrm{Lin}(\cdot)} \rightarrow g_1 \rightarrow \underset{\frac{1}{1+e^{-g_0}}}{\sigma(\cdot)}}_{} \rightarrow f_2$$

$$f_3 \rightarrow \underbrace{\quad\quad}_{} \rightarrow f_4 \quad \cdots\cdots \quad f_n \rightarrow \frac{1}{2}\big(\underbrace{f_n}_{\text{o/p of network}} - \underbrace{\hat{f}}_{\text{Label}}\big)^2 \longrightarrow \underset{\text{Loss}}{L}$$

$$\underbrace{\rightarrow f_1 \rightarrow A_1, b_1 \rightarrow g_1 \rightarrow f_2}_{} \qquad\qquad \rightarrow \underbrace{f_{n-1} \rightarrow f_n}_{} \rightarrow L$$

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial f_n} \cdot \frac{\partial f_n}{\partial f_{n-1}} \cdot \frac{\partial f_{n-1}}{\partial f_{n-2}} \cdots \underbrace{\frac{\partial f_2}{\partial f_1}} \cdot \frac{\partial f_1}{\partial f_0} \cdot \frac{\partial f_0}{\partial x}$$

$$\frac{\partial f_2}{\partial g_1} \cdot \frac{\partial g_1}{\partial A_1} \cdot \frac{\partial A_1}{\partial f_1} + \frac{\partial f_2}{\partial g_1} \frac{\partial g_1}{\partial b_1} \cdot \frac{\partial b_1}{\partial f_1}$$

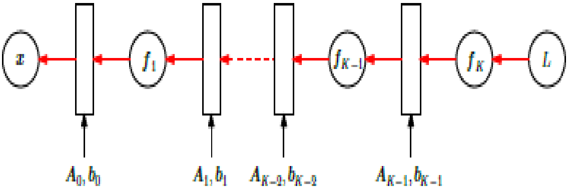
$$\frac{\partial L}{\partial A_1} = \underbrace{\frac{\partial L}{\partial f_n} \cdot \frac{\partial f_n}{\partial f_{n-1}} \cdots \frac{\partial f_3}{\partial f_2}}_{\frac{\partial L}{\partial f_2}} \underbrace{\frac{\partial f_2}{\partial A_1}} \quad \frac{\partial f_2}{\partial g_1} \cdot \frac{\partial g_1}{\partial A_1}$$

Backpropagation

Using the chain rule, we get

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_{K-1}} &= \frac{\partial L}{\partial f_K}\frac{\partial f_K}{\partial \theta_{K-1}}\\
\frac{\partial L}{\partial \theta_{K-2}} &= \frac{\partial L}{\partial f_K}\frac{f_K}{f_{K-1}}\frac{\partial f_{K-1}}{\partial \theta_{K-2}}\\
\frac{\partial L}{\partial \theta_{K-3}} &= \frac{\partial L}{\partial f_K}\frac{f_K}{f_{K-1}}\frac{\partial f_{K-1}}{\partial f_{K-2}}\frac{\partial f_{K-2}}{\partial \theta_{K-3}}\\
\frac{\partial L}{\partial \theta_{i}} &= \frac{\partial L}{\partial f_K}\frac{f_K}{f_{K-1}}\cdots\frac{\partial f_{i+2}}{\partial f_{i+1}}\frac{\partial f_{i+1}}{\partial \theta_{i}}
\end{aligned}
$$

Backpropagation



If the partial derivatives $\frac{\partial L}{\partial \theta_{i+1}}$ are computed, then the computation can be reused to compute $\frac{\partial L}{\partial \theta_i}$.

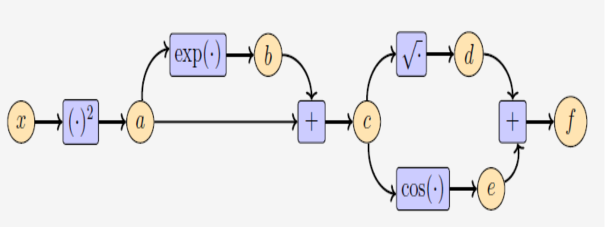
Example

Consider the function

$$f(x) = \sqrt{x^2 + exp(x^2)} + cos(x^2 + exp(x^2))$$

Let

$a = x^2, b = exp(a), c = a + b, d = \sqrt{c}, e = cos(c) \Rightarrow f = d + e$



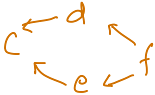
Example

$$
\begin{aligned}
\Rightarrow \frac{\partial a}{\partial x} &= 2x \\
\frac{\partial b}{\partial a} &= exp(a) \\
\frac{\partial c}{\partial a} &= 1 = \frac{\partial c}{\partial b} \\
\frac{\partial d}{\partial c} &= \frac{1}{2\sqrt{c}} \\
\frac{\partial e}{\partial c} &= -sin(c) \\
\frac{\partial f}{\partial d} &= 1 = \frac{\partial f}{\partial e}
\end{aligned}
$$

Example

Thus, we have

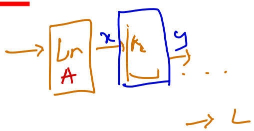
$$
\begin{aligned}
\frac{\partial f}{\partial c} &= \frac{\partial f}{\partial d}\frac{\partial d}{\partial c} + \frac{\partial f}{\partial e}\frac{\partial e}{\partial c}\\
\frac{\partial f}{\partial b} &= \frac{\partial f}{\partial c}\frac{\partial c}{\partial b}\\
\frac{\partial f}{\partial a} &= \frac{\partial f}{\partial b}\frac{\partial b}{\partial a} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial a}\\
\frac{\partial f}{\partial x} &= \frac{\partial f}{\partial a}\frac{\partial a}{\partial x}
\end{aligned}
$$



Example

Substituting the results, we get

$$
\begin{aligned}
\frac{\partial f}{\partial c} &= 1.(\frac{1}{2\sqrt{c}} + 1).(-sin(c)) \\
\frac{\partial f}{\partial b} &= \frac{\partial f}{\partial c}.1 \\
\frac{\partial f}{\partial a} &= \frac{\partial f}{\partial b}exp(a) + \frac{\partial f}{\partial c}.1 \\
\frac{\partial f}{\partial x} &= \frac{\partial f}{\partial a}2x
\end{aligned}
$$



$$\frac{\partial L}{\partial A} = \dots \underbrace{\frac{\partial y}{\partial x}}_{ReLU} \dots$$

Thus, the computation for calculating the derivative is of similar complexity as the computation of the function itself.

Formalization of Automatic Differentiation

Let $x_1, \cdots, x_d$: input variables.
$x_{d+1}, \cdots, x_{D-1}$: intermediate variables.
x_D : output variable, then we have,

$$x_i = g_i(x_{Pa(x_i)})$$

Note that g_is are elementary functions and are also called as forward propagation function and $x_{Pa(x_i)}$ is the set of parent nodes of variable x_i in the graph.

Formalization of Automatic Differentiation

$R^m \to R^n$ (Rev mode Auto diff)

$R^m \to R$ (Back Prop)

o/p is scalar (Loss)

Now,

$$f = x_D \Rightarrow \frac{\partial f}{\partial_D} = 1$$

For other variables, using chain rule, we get

$$\frac{\partial f}{\partial x_i} = \sum_{x_j : x_i \in Pa(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i} = \sum_{x_j : x_i \in Pa(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial g_i}{\partial x_i}$$

The last equation is the back propagation of the gradient through the computation graph. For neural network training, we back propagate the error of the prediction with respect to the label.