

# Machine Learning: Introduction to Linear Regression, Logistic Regression, and Neural Networks

# Chapter 3 Review of Mathematical Concepts

# Review of Mathematical Concepts

Supervised Learning: Process of learning a function that maps input information to labelled output information. The labelled input/output information is called the training data. The learned function is then used to predict outputs when new input information is provided.

Approach and underlying mathematics

- Assume “Function Structure” with unknown function parameters
- Function structures for Linear Regression, Logistic Regression, and Neural Networks are based on matrices
- Training Algorithm uses optimization to find function parameters that minimize loss function
- Optimization algorithms involve computation of gradients, which requires the chain rule and matrix multiplication

# Review of Mathematical Concepts

Section	Title	Coverage
3.1	Linear Algebra	Vectors, Matrices, and Transposes Matrix Multiplication Broadcasting Multiplication and Transposes
3.2	Multivariable Calculus	Partial Derivatives Gradients Chain Rule
3.3	Optimization	Gradient Descent for Minimizing a Function
3.4	Notation	Notation to be used in Course

# 3.1 Review of Mathematical Concepts: Linear Algebra

# Review of Mathematical Concepts

- Linear Algebra
  - Vectors, Matrices, and Transposes
  - Dot Product and Matrix Multiplication
  - Broadcasting
  - Multiplication and Transpose

# Linear Algebra – Vectors and Matrices

- Example of a row vector of length  $d$  (start index at 0)

$$W = [W_0 \ W_1 \ \dots \ W_{d-1}]$$

Denote the  $i$ 'th entry as  $W_i$

- Example of a matrix with dimensions  $d \times m$  ( $d$  rows and  $m$  columns)  
(indices are  $i=0, \dots, d-1$ , and  $j=0, \dots, m-1$ )

$$X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

Denote the entry for row  $i$  and column  $j$  as  $X_{ij}$  (row index 1st, column index 2<sup>nd</sup>)

# Linear Algebra – Transpose

- Let  $W$  be a row vector of length  $d$  (dimensions  $1 \times d$ )
- The transpose of  $W$  denoted  $W^T$  ( $d \times 1$ ) is a column vector of length  $d$

$$W = [W_0 \ W_1 \ \dots \ W_{d-1}] \quad W^T = \begin{bmatrix} W_0 \\ \vdots \\ W_{d-1} \end{bmatrix}$$

$\xrightarrow{\text{d entries}}$ 
 $\downarrow \text{d entries}$

- Let  $X$  be ( $d \times m$ ) matrix. Transpose (dimensions  $m \times d$ ) denoted by superscript  $T$  defined by:  $X_{ij}^T = X_{ji}$

$$X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \vdots & \ddots & \vdots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix} \quad X^T = \begin{bmatrix} X_{00} & \dots & X_{d-1,0} \\ \vdots & \ddots & \vdots \\ X_{0,m-1} & \dots & X_{d-1,m-1} \end{bmatrix}$$

$\xrightarrow{\text{m entries}}$ 
 $\downarrow \text{d entries}$ 
 $\xrightarrow{\text{d entries}}$ 
 $\downarrow \text{m entries}$



# Linear Algebra – Transpose Example

$$W = [1 \ 2 \ 3 \ 4] \qquad W^T = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

- Example 2:

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}$$

# Linear Algebra – Dot Product

- Let  $W$  and  $X$  both be row vectors of length  $d$

$$W = [W_0 \ W_1 \ \dots \ W_{d-1}] \text{ and } X = [X_0 \ X_1 \ \dots \ X_{d-1}]$$

- Dot product of  $W$  and  $X$  given by:

$$Z = W_0X_0 + W_1X_1 + W_2X_2 + \dots + W_{d-1}X_{d-1} = \sum_{i=0}^{d-1} W_iX_i$$

- Can express this as a matrix multiplication (which is built upon dot product of row of first object and column of second)

$$Z = WX^T = [W_0 \ W_1 \ \dots \ W_{d-1}] \begin{bmatrix} X_0 \\ X_1 \\ \dots \\ X_{d-1} \end{bmatrix}$$

# Linear Algebra – Dot Product Example

- Example

$$W = [1 \ 2 \ 3 \ 4] \quad X = [5 \ 6 \ 7 \ 8]$$

$$Z = WX^T = [1 \ 2 \ 3 \ 4] \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} = (1)(5) + (2)(6) + (3)(7) + (4)(8) = 70$$

# Linear Algebra – Vector/Matrix Multiplication

- Let  $W$  be row vector of length  $d$ ,  $X$  be a matrix of dimension  $(d \times m)$

$$W = [W_0 \ W_1 \ \dots \ W_{d-1}] \quad X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

- Define  $Z_j$  to be the dot product of  $W$  and the  $j$ 'th column of  $X$

$$Z_j = W_0 X_{0j} + W_1 X_{1j} + W_2 X_{2j} + \dots + W_{d-1} X_{d-1,j} = \sum_{i=0}^{d-1} W_i X_{i,j} \quad \text{for } j=0, \dots, m-1$$

- $Z$  is a row vector of length  $m$

$$[Z_0 \ Z_1 \ \dots \ Z_{m-1}] = [W_0 \ W_1 \ \dots \ W_{d-1}] \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

or

$$Z = WX$$

# Linear Algebra – Matrix/Matrix Multiplication

- Let  $W$  be a matrix of dimension  $(n \times d)$ ,  $X$  be a matrix of dimension  $(d \times m)$

$$W = \begin{bmatrix} W_{00} & \dots & W_{0,d-1} \\ \dots & \dots & \dots \\ W_{n-1,0} & \dots & W_{n-1,d-1} \end{bmatrix} \quad X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

- Define  $Z_{ij}$  to be the dot product of row  $i$  of  $W$  and column  $j$  of  $X$

$$Z_{ij} = W_{i0}X_{0j} + W_{i1}X_{1j} + W_{i2}X_{2j} + \dots + W_{i,d-1}X_{d-1,j} = \sum_{k=0}^{d-1} W_{ik}X_{kj} \quad i=0,\dots,n-1, j=0,\dots,m-1$$

- $Z$  is an  $n \times m$  matrix

$$\begin{bmatrix} Z_{00} & \dots & Z_{0,m-1} \\ \dots & \dots & \dots \\ Z_{n-1,0} & \dots & Z_{n-1,m-1} \end{bmatrix} = \begin{bmatrix} W_{00} & \dots & W_{0,d-1} \\ \dots & \dots & \dots \\ W_{n-1,0} & \dots & W_{n-1,d-1} \end{bmatrix} \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

or

$$Z = WX$$

# Linear Algebra – Multiplication Example 1

$$W = [1 \ 2 \ 3] \quad X = \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

$$Z = WX = [1 \ 2 \ 3] \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

$$Z = [(1)(4) + (2)(5) + (3)(6) \quad (1)(7) + (2)(8) + (3)(9)]$$

$$Z = [32 \quad 50]$$

# Linear Algebra – Multiplication Example 2

$$W = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \quad X = \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

$$Z = WX = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

$$Z = \begin{bmatrix} (1)(4) + (2)(5) + (3)(6) & (1)(7) + (2)(8) + (3)(9) \\ (2)(4) + (3)(5) + (4)(6) & (2)(7) + (3)(8) + (4)(9) \end{bmatrix}$$

$$Z = \begin{bmatrix} 32 & 50 \\ 47 & 74 \end{bmatrix}$$

# Linear Algebra – Broadcasting

- Let  $W$  be row vector of length  $d$ ,  $X$  be a matrix of dimension  $(d \times m)$ , and  $b$  a scalar

$$W = [W_0 \ W_1 \ \dots \ W_{d-1}] \quad X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

- Define  $Z_j$  to be dot product of  $W$  and column  $j$  of  $X$  plus  $b$

$$Z_j = W_0 X_{0j} + W_1 X_{1j} + W_2 X_{2j} + \dots + W_{d-1} X_{d-1,j} + b = \sum_{i=0}^{d-1} W_i X_{ij} + b \text{ for } j=0, \dots, m-1$$

- $Z$  is a row vector of length  $m$

$$[Z_0 \ Z_1 \ \dots \ Z_{m-1}] = [W_0 \ W_1 \ \dots \ W_{d-1}] \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix} + b$$

or  $Z = WX + b$

- $b$  is a scalar, so it does not have the same dimensions as  $Z$ . It is added for each element of  $Z$ . This is an example of broadcasting (this is term used in numpy documentation)



# Linear Algebra – Broadcasting Example 1

$$W = [1 \ 2 \ 3] \quad X = \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix} \quad b = 7$$

$$Z = WX + b = [1 \ 2 \ 3] \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix} + 7$$

$$Z = [(1)(4) + (2)(5) + (3)(6) + 7 \quad (1)(7) + (2)(8) + (3)(9) + 7]$$

$$Z = [39 \quad 57]$$

# Linear Algebra – Broadcasting

- Let  $W$  be a matrix of dimension  $(n \times d)$ ,  $X$  be a matrix of dimension  $(d \times m)$ ,  $b$  be a column vector of length  $n$

$$W = \begin{bmatrix} W_{00} & \dots & W_{0,d-1} \\ \dots & \dots & \dots \\ W_{n-1,0} & \dots & W_{n-1,d-1} \end{bmatrix} \quad X = \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix} \quad b = \begin{bmatrix} b_0 \\ \dots \\ b_{n-1} \end{bmatrix}$$

- Define  $Z_{ij}$  to be the dot product of row  $i$  of  $W$  and column  $j$  of  $X$  plus  $i$ 'th entry of  $b$

$$Z_{ij} = W_{i0}X_{0j} + W_{i1}X_{1j} + W_{i2}X_{2j} + \dots + W_{i,d-1}X_{d-1,j} + b_i = \sum_{k=0}^{d-1} W_{ik}X_{kj} + b_i \quad i=0,\dots,n-1, j=0,\dots,m-1$$

- $Z$  is an  $n \times m$  matrix

$$\begin{bmatrix} Z_{00} & \dots & Z_{0,m-1} \\ \dots & \dots & \dots \\ Z_{n-1,0} & \dots & Z_{n-1,m-1} \end{bmatrix} = \begin{bmatrix} W_{00} & \dots & W_{0,d-1} \\ \dots & \dots & \dots \\ W_{n-1,0} & \dots & W_{n-1,d-1} \end{bmatrix} \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix} + \begin{bmatrix} b_0 \\ \dots \\ b_{n-1} \end{bmatrix}$$

or

$$Z = WX + b$$

- Example of Broadcasting - entry  $i$  of  $b$  is added to each entry of row  $i$  of  $Z$

# Linear Algebra – Broadcasting Example 2

$$W = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \quad X = \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix} \quad b = \begin{bmatrix} 11 \\ 12 \end{bmatrix}$$

$$Z = WX + b = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix} + \begin{bmatrix} 11 \\ 12 \end{bmatrix}$$

$$Z = \begin{bmatrix} (1)(4) + (2)(5) + (3)(6) + 11 & (1)(7) + (2)(8) + (3)(9) + 11 \\ (2)(4) + (3)(5) + (4)(6) + 12 & (2)(7) + (3)(8) + (4)(9) + 12 \end{bmatrix}$$

$$Z = \begin{bmatrix} 43 & 61 \\ 59 & 86 \end{bmatrix}$$

# Linear Algebra – Multiplication and Transpose

- Consider:

$$Z_{ij} = \sum_{k=0}^{d-1} W_{ik} X_{jk} \quad (\text{dot product of row } i \text{ of } W \text{ and row } j \text{ of } X)$$

- Rewrite as:

$$Z_{ij} = \sum_{k=0}^{d-1} W_{ik} X_{kj}^T \quad (\text{dot product of row } i \text{ of } W \text{ and column } j \text{ of } X^T)$$

- Can express this as a matrix multiplication:  $Z = WX^T$

- Consider:

$$Z_{ij} = \sum_{k=0}^{d-1} W_{ki} X_{kj} \quad (\text{dot product of column } i \text{ of } W \text{ and column } j \text{ of } X)$$

- Rewrite as:

$$Z_{ij} = \sum_{k=0}^{d-1} W_{ki}^T X_{kj} \quad (\text{dot product of row } i \text{ of } W^T \text{ and column } j \text{ of } X)$$

- Can express this as matrix multiplication:  $Z = W^T X$

# Linear Algebra – Jupyter Notebook Demo

- Open file IntroML/Examples/Chapter3/LinearAlgebra.ipynb
- Has examples of
  - Transpose
  - Matrix Multiplication
  - Matrix Multiplication with Broadcasting

## 3.2 Review of Mathematical Concepts: Multivariable Calculus

# Review of Mathematical Concepts

- Multivariable Calculus
  - Partial Derivatives
  - Gradients
  - Chain Rule

# Multivariable Calculus – Partial Derivatives

- Consider a function of  $m$  variables  $L = L(Z_0, Z_1, Z_2, \dots, Z_{m-1})$
- Partial derivative of  $L$  with respect to  $Z_j$  measures rate of change of  $L$  if only  $Z_j$  changes. By definition this partial derivative is:

$$\frac{\partial L}{\partial Z_j} = \lim_{\varepsilon \rightarrow 0} \frac{L(Z_0, Z_1, \dots, Z_j + \varepsilon, \dots, Z_{m-1}) - L(Z_0, Z_1, \dots, Z_j, \dots, Z_{m-1})}{\varepsilon}$$



# Multivariable Calculus – Gradient

- Consider a function of  $m$  variables  $L = L(Z_0, Z_1, Z_2, \dots, Z_{m-1})$
- The gradient is vector of length  $m$  defined by:

$$\nabla_Z L = \begin{bmatrix} \frac{\partial L}{\partial Z_0} & \frac{\partial L}{\partial Z_1} & \cdots & \frac{\partial L}{\partial Z_{m-1}} \end{bmatrix}$$

- Individual entries of gradient vector denoted by:

$$\nabla_Z L_j = \frac{\partial L}{\partial Z_j}, \quad j = 0, \dots, m - 1$$

# Multivariable Calculus - Gradient Notation

Generalize the gradient notation to scalars and matrices

- If  $L$  is a function of a single variable:  $L = L(b)$ . Gradient defined as:

$$\nabla_b L = \left[ \frac{\partial L}{\partial b} \right]$$

- If  $L$  is function of  $n \times m$  variables in matrix  $Z$  ( $Z_{ij}$  for  $i=0, \dots, n-1, j=0, \dots, m-1$ ),  $\nabla_Z L$  is a matrix with same dimensions as  $Z$

$$\nabla_Z L = \begin{bmatrix} \frac{\partial L}{\partial Z_{00}} & \cdots & \frac{\partial L}{\partial Z_{0,m-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial Z_{n-1,0}} & \cdots & \frac{\partial L}{\partial Z_{n-1,m-1}} \end{bmatrix}$$

$$\nabla_Z L_{ij} = \frac{\partial L}{\partial Z_{ij}} \quad i = 0, \dots, n-1, j = 0, \dots, m-1$$

# Multivariable Calculus – Chain Rule

- Consider a function of  $m$  variables  $L = L(Z_0, Z_1, Z_2, \dots, Z_{m-1})$ .
- Suppose:  $Z_j = W_0 X_{0j} + W_1 X_{1j} + W_2 X_{2j} + \dots + W_{d-1} X_{d-1,j}$  for  $j=0, \dots, m-1$
- In matrix terms:

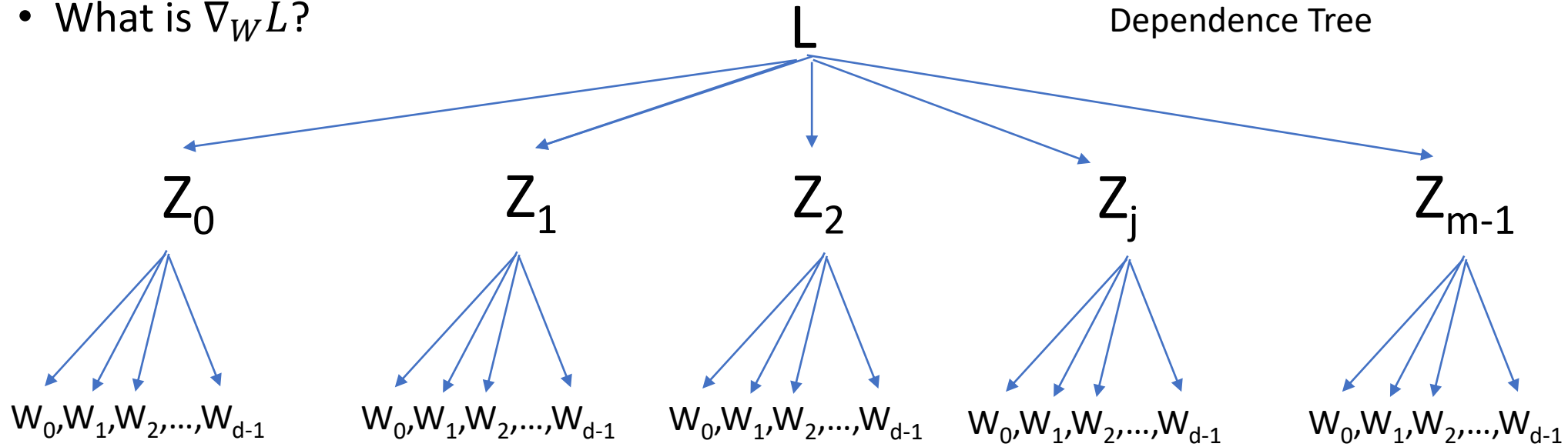
$$[Z_0 \ Z_1 \ \dots \ Z_{m-1}] = [W_0 \ W_1 \ \dots \ W_{d-1}] \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix}$$

$$Z = WX$$

- Question: what is  $\nabla_W L$ ?

# Multivariable Calculus – Chain Rule

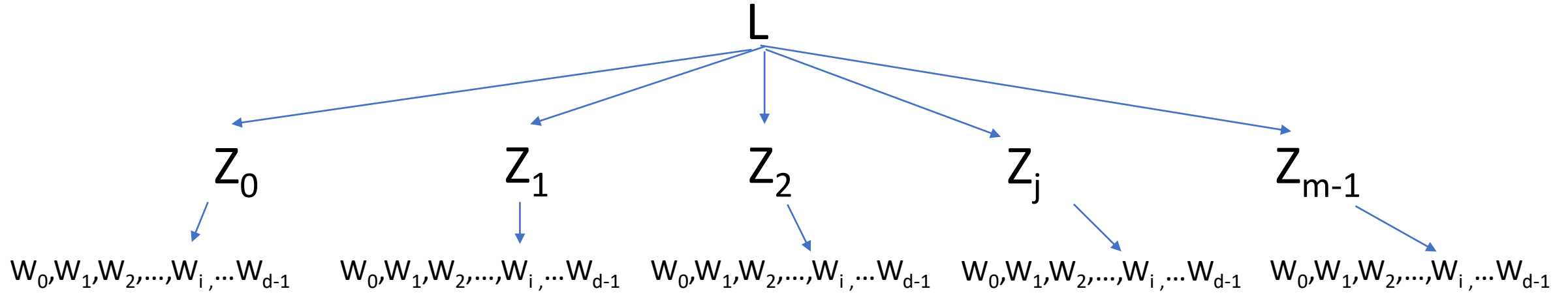
- What is  $\nabla_W L$ ?



Applying the chain rule of multivariable calculus:

- (1) To determine partial derivative of  $L$  with respect to  $W_i$ , find all paths from  $L$  to  $W_i$
- (2) Take partial derivatives along each path and multiply
- (3) Sum over all the paths

# Multivariable Calculus – Chain Rule



1. For  $\frac{\partial L}{\partial W_i}$  there are  $m$  paths:  $L \rightarrow Z_j \rightarrow W_i$   $j=0, \dots, m-1$ :
2. Product of derivatives on each path:  $\frac{\partial L}{\partial Z_j} \frac{\partial Z_j}{\partial W_i}$
3. Sum over all paths:  $\frac{\partial L}{\partial W_i} = \sum_{j=0}^{m-1} \frac{\partial L}{\partial Z_j} \frac{\partial Z_j}{\partial W_i}$  (for  $i=0, \dots, d-1$ )

• By definition:  $\frac{\partial L}{\partial Z_j} = \nabla_Z L_j$

• From the last pages:  $Z_j = W_0 X_{0j} + W_1 X_{1j} + W_2 X_{2j} + \dots + W_{d-1} X_{d-1,j}$  so  $\frac{\partial Z_j}{\partial W_i} = X_{ij}$

$$\frac{\partial L}{\partial W_i} = \sum_{j=0}^{m-1} \nabla_Z L_j X_{ij} = \sum_{j=0}^{m-1} \nabla_Z L_j X_{ji}^T$$

In matrix form:  $\nabla_W L = \nabla_Z L X^T$  dimensions:  $1 \times d = (1 \times m) \text{ times } (m \times d)$

# Multivariable Calculus – Chain Rule

- Consider more general case:  $L = L(Z)$
- $Z$  is an  $(n \times m)$  matrix, so  $L$  is a function of  $n \times m$  variables
- $Z$  defined by  $Z = WX + b$ :  $W$  is  $n \times d$ ,  $X$  is  $d \times m$  and  $b$  is  $n \times 1$

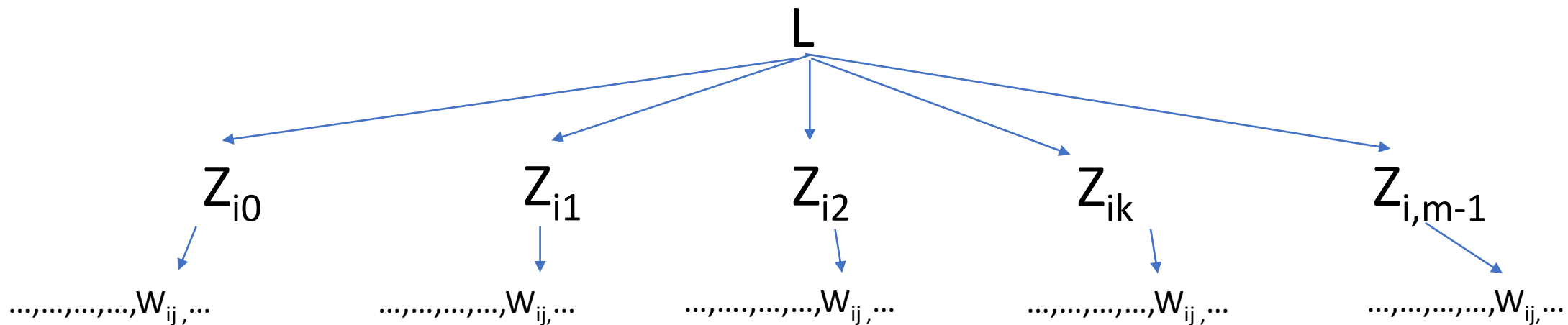
$$\begin{bmatrix} Z_{00} & \dots & Z_{0,m-1} \\ \dots & \dots & \dots \\ Z_{n-1,0} & \dots & Z_{n-1,m-1} \end{bmatrix} = \begin{bmatrix} W_{00} & \dots & W_{0,d-1} \\ \dots & \dots & \dots \\ W_{n-1,0} & \dots & W_{n-1,d-1} \end{bmatrix} \begin{bmatrix} X_{00} & \dots & X_{0,m-1} \\ \dots & \dots & \dots \\ X_{d-1,0} & \dots & X_{d-1,m-1} \end{bmatrix} + \begin{bmatrix} b_0 \\ \dots \\ b_{n-1} \end{bmatrix}$$

- What are:  $\nabla_W L$ ,  $\nabla_b L$ ,  $\nabla_X L$ ?
- Note: these gradients come up in machine learning training algorithms

# Multivariable Calculus – Chain Rule

- For  $\nabla_W L_{ij} = \frac{\partial L}{\partial W_{ij}}$  ( $i=0,\dots,n-1, j=0,\dots,d-1$ )

$$Z=WX+b \quad \begin{bmatrix} \dots & \dots & \dots \\ Z_{i0} & \dots & Z_{i,m-1} \\ \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & W_{ij} & \dots \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \dots & \dots & \dots \\ X_{j0} & \dots & X_{j,m-1} \\ \dots & \dots & \dots \end{bmatrix} + \begin{bmatrix} \dots \\ \dots \\ \dots \end{bmatrix}$$

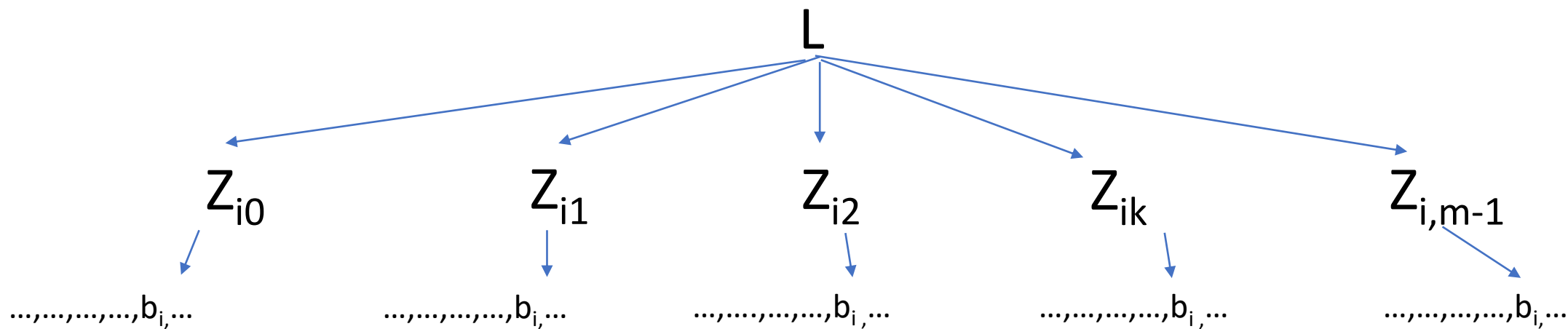


- Applying chain rule:  $\nabla_W L_{ij} = \sum_{k=0}^{m-1} \frac{\partial L}{\partial Z_{ik}} \frac{\partial Z_{ik}}{\partial W_{ij}} = \sum_{k=0}^{m-1} \nabla_Z L_{ik} X_{jk} = \sum_{k=0}^{m-1} \nabla_Z L_{ik} X_{kj}^T$
- This is matrix/matrix multiplication:  $\nabla_W L_{ij} = (\nabla_Z L X^T)_{ij}$
- Hence:  $\nabla_W L = \nabla_Z L X^T$

# Multivariable Calculus – Chain Rule

- For  $\nabla_b L_i = \frac{\partial L}{\partial b_i}$  ( $i=0, \dots, n-1$ )

$$Z=WX+b \quad \begin{bmatrix} \dots & \dots & \dots \\ Z_{i0} & \dots & Z_{i,m-1} \\ \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix} + \begin{bmatrix} \dots \\ b_i \\ \dots \end{bmatrix}$$



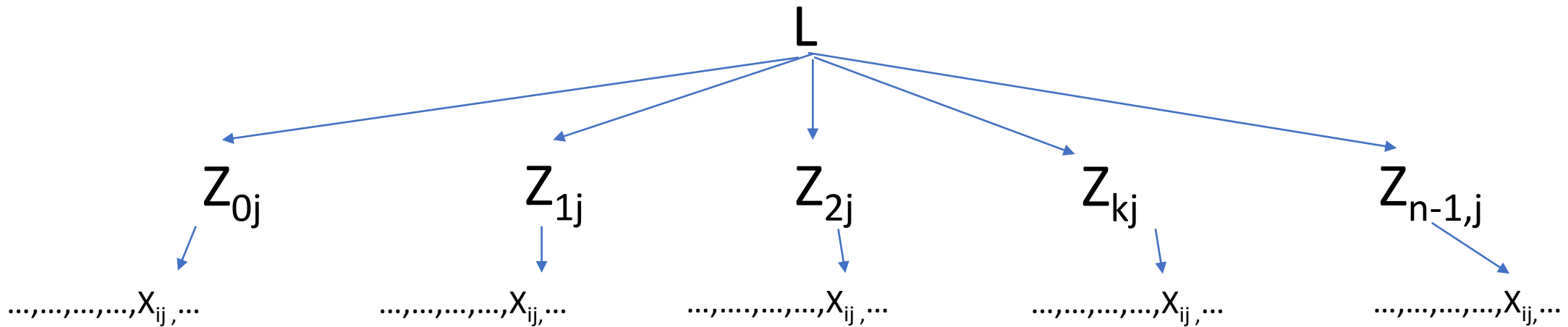
- Applying chain rule:  $\nabla_b L_i = \sum_{k=0}^{m-1} \frac{\partial L}{\partial Z_{ik}} \frac{\partial Z_{ik}}{\partial b_i} = \sum_{k=0}^{m-1} \nabla_Z L_{ik} * 1 = \sum_{k=0}^{m-1} \nabla_Z L_{ik}$
- This is simply a sum along the  $i$ 'th row of the gradient  $\nabla_Z L$



# Multivariable Calculus – Chain Rule

- For  $\nabla_X L_{ij} = \frac{\partial L}{\partial X_{ij}}$  ( $i=0,\dots,d-1, j=0,\dots,m-1$ )

$$Z=WX+b \quad \begin{bmatrix} \dots & Z_{0j} & \dots \\ \dots & \dots & \dots \\ \dots & Z_{n-1,j} & \dots \end{bmatrix} = \begin{bmatrix} \dots & W_{0i} & \dots \\ \dots & \dots & \dots \\ \dots & W_{n-1,i} & \dots \end{bmatrix} \begin{bmatrix} \dots & \dots & \dots \\ \dots & X_{ij} & \dots \\ \dots & \dots & \dots \end{bmatrix} + \begin{bmatrix} \dots \\ \dots \\ \dots \end{bmatrix}$$



- Applying chain rule:  $\nabla_X L_{ij} = \sum_{k=0}^{n-1} \frac{\partial L}{\partial Z_{kj}} \frac{\partial Z_{kj}}{\partial X_{ij}} = \sum_{k=0}^{n-1} \nabla_Z L_{kj} W_{ki} = \sum_{k=0}^{n-1} W_{ik}^T \nabla_Z L_{kj}$
- This is matrix/matrix multiplication:  $\nabla_X L_{ij} = (W^T \nabla_Z L)_{ij}$
- Hence:  $\nabla_X L = W^T \nabla_Z L$

# Multivariable Calculus – Chain Rule Example

- Consider:

$$L = 2Z_0 + Z_1$$

- Gradient is

$$\nabla_Z L = \left[ \frac{\partial L}{\partial Z_0} \quad \frac{\partial L}{\partial Z_1} \right] = [2 \quad 1]$$

- Now assume:

$$Z = WX + b \quad \text{and} \quad W = [W_0 \quad W_1], X = \begin{bmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{bmatrix}$$

so

$$Z = [Z_0 \quad Z_1] = [W_0X_{00} + W_1X_{10} + b \quad W_0X_{01} + W_1X_{11} + b]$$

- Substituting the expressions for  $Z_0$  and  $Z_1$  into  $L$ , we get:

$$L = 2W_0X_{00} + 2W_1X_{10} + 2b + W_0X_{01} + W_1X_{11} + b$$

# Multivariable Calculus – Chain Rule Example

Start with:

$$L = 2W_0X_{00} + 2W_1X_{10} + 2b + W_0X_{01} + W_1X_{11} + b \quad \nabla_Z L = \begin{bmatrix} \frac{\partial L}{\partial z_0} & \frac{\partial L}{\partial z_1} \end{bmatrix} = [2 \quad 1]$$

$$\text{Direct Calculation: } \nabla_W L = \begin{bmatrix} \frac{\partial L}{\partial w_0} & \frac{\partial L}{\partial w_1} \end{bmatrix} = [2X_{00} + X_{01} \quad 2X_{10} + X_{11}]$$

$$\text{Chain Rule: } \nabla_W L = \nabla_Z L X^T = [2 \quad 1] \begin{bmatrix} X_{00} & X_{10} \\ X_{01} & X_{11} \end{bmatrix} = [2X_{00} + X_{01} \quad 2X_{10} + X_{11}]$$

$$\text{Direct Calculation: } \nabla_X L = \begin{bmatrix} \frac{\partial L}{\partial x_{00}} & \frac{\partial L}{\partial x_{01}} \\ \frac{\partial L}{\partial x_{10}} & \frac{\partial L}{\partial x_{11}} \end{bmatrix} = \begin{bmatrix} 2W_0 & W_0 \\ 2W_1 & W_1 \end{bmatrix}$$

$$\text{Chain Rule: } \nabla_X L = W^T \nabla_Z L = \begin{bmatrix} W_0 \\ W_1 \end{bmatrix} [2 \quad 1] = \begin{bmatrix} 2W_0 & W_0 \\ 2W_1 & W_1 \end{bmatrix}$$

$$\text{Direct Calculation: } \nabla_b L = \left[ \frac{\partial L}{\partial b} \right] = [3]$$

$$\text{Chain Rule: } \nabla_b L = \sum_{j=0}^{m-1} \nabla_Z L_j = [3]$$

# Gradient Formulas Summary

Component		Description
Function	Z is nxm matrix W is nxd matrix X is dxm matrix b is nx1 matrix	$L = f(Z)$ $Z = WX + b$
Gradients		$\nabla_W L = \nabla_Z L X^T$ $\nabla_b L_i = \sum_{j=0}^{m-1} \nabla_Z L_{ij}$ (sum along i'th row of gradient) $\nabla_X L = W^T \nabla_Z L$

## 3.3 Review of Mathematical Concepts: Optimization

# Review of Mathematical Concepts

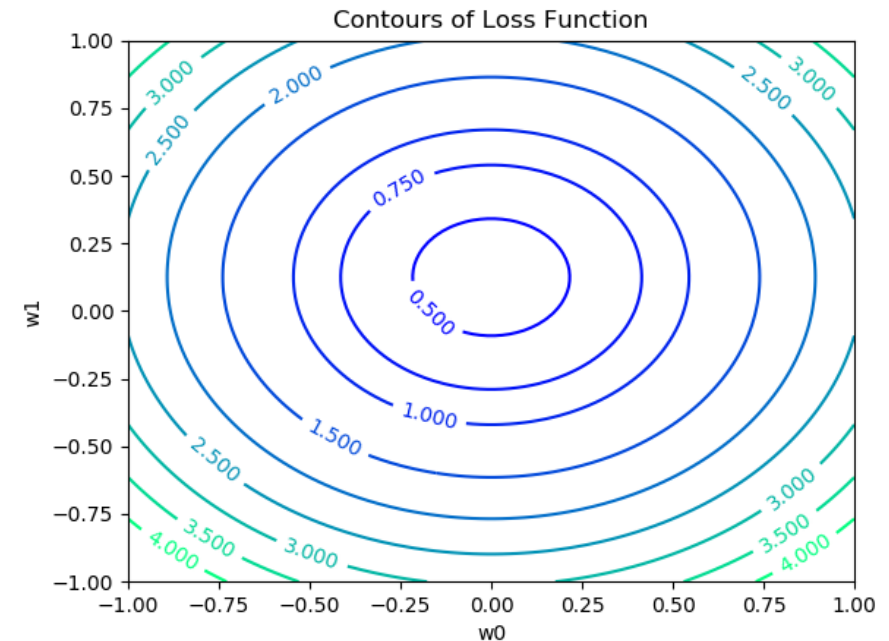
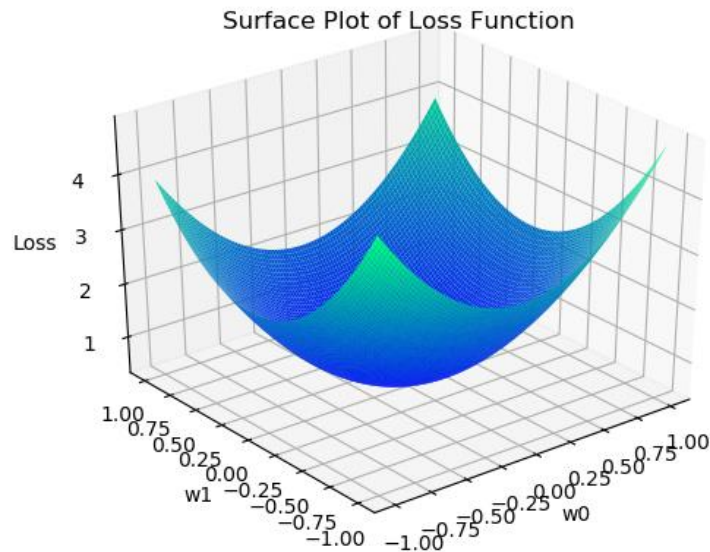
- Optimization
  - Gradient Descent for Minimizing a Function

# Optimization

- Let  $L(W_0, W_1, W_2, \dots, W_{d-1})$  denote a Loss function of  $d$  parameters
- Machine Learning Training Algorithm involves finding the parameters  $W_0, W_1, W_2, \dots, W_{d-1}$  that minimize Loss function
- From multivariable calculus, gradient  $\nabla_W L = 0$  at minimum
- $\nabla_W L$  has  $d$  components, so setting gradient to 0 involves solving  $d$  equations for  $d$  unknowns
- In general it is not feasible to solve these equations exactly
- Optimization is a process of finding the minimum (or getting close to minimum) using an iterative process making use of the gradient

# Optimization - Surface and Contour Plots

- Use example of Loss function of 2 variables  $\text{Loss} = L(W_0, W_1)$  to illustrate Gradient Descent optimization algorithm
- Surface plot: Loss function (3 dimensions)
- Contour plot: lines of constant Loss in  $(W_0, W_1)$  plane (2 dimensions)



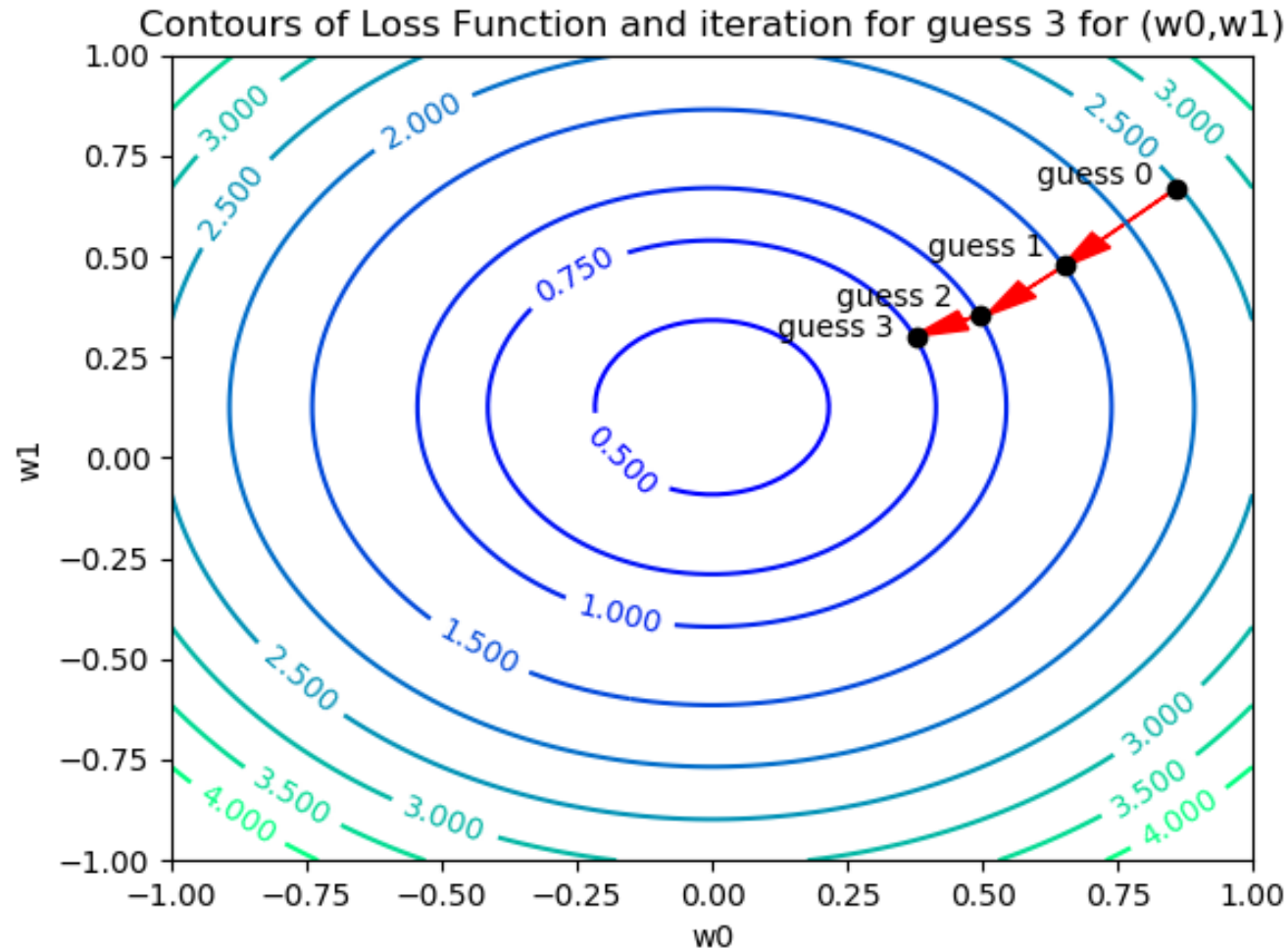


# Optimization - Gradient Descent Algorithm

## Key facts to be used in Gradient Descent Algorithm

- At each point in  $(W_0, W_1)$  plane one can compute gradient vector  $\nabla_W L$ , which points in direction of most rapid increase of  $L$
- Gradient orthogonal to contour lines
- To minimize the  $L$ , take step in opposite direction of gradient
- Gradient Descent algorithm applies this process repeatedly:
  - Make guess for  $(W_0, W_1)$
  - Loop: compute gradient at current point and take step in opposite direction
- Size of step depends on parameter  $\alpha > 0$  called Learning Rate
  - If too large, then may overshoot minimum
  - If too small, then many steps may be required to get near minimum

# Optimization - Gradient Descent Animation



# Optimization: Gradient Descent Algorithm

Let  $W = [W_0 \ W_1 \ W_2 \ \dots \ W_{d-1}]$  denote parameter vector of variables. Let  $L(W) = L(W_0, W_1, W_2, \dots, W_{d-1})$  be a function of these parameters

Make initial guess  $W_{\text{epoch}=0}$  and choose learning rate  $\alpha > 0$

1. Loop epoch  $i = 1, 2, 3, \dots$

- Compute gradient vector  $\nabla_W L_{\text{epoch}=i-1}$  at  $W_{\text{epoch}=i-1}$
- Compute new guess using formula:  
$$W_{\text{epoch}=i} = W_{\text{epoch}=i-1} - \alpha \nabla_W L_{\text{epoch}=i-1}$$
- Compute  $L(W_{\text{epoch}=i})$

Loop for fixed number of epoch (or if  $L(W)$  reduced sufficiently)

Note: this process may not lead to convergence to minimum or may converge to a local minimum

# Optimization – Gradient Descent Example

- Consider simple function (has minimum at [0 0])

$$L(W) = L(W_0, W_1) = 2W_0^2 + W_1^2 \text{ with gradient } \nabla_W L = [4W_0 \quad 2W_1]$$

- Choose  $\alpha=0.1$  and initial guess

$$W_{epoch=0} = [2 \ 2]$$

$$L(W_{epoch=0}) = 2 * 2^2 + 2^2 = 12$$

- Epoch 1:

$$\nabla_W L(W_{epoch=0}) = [4W_0 \quad 2W_1] = [8 \ 4]$$

$$W_{epoch=1} = W_{epoch=0} - \alpha \nabla_W L(W_{epoch=0}) = [2 \ 2] - 0.1 * [8 \ 4] = [1.2 \ 1.6]$$

$$L(W_{epoch=1}) = 2 * 1.2^2 + 1.6^2 = 5.44$$

- Epoch 2:

$$\nabla_W L(W_{epoch=1}) = [4W_0 \quad 2W_1] = [4.8 \ 3.2]$$

$$W_{epoch=2} = W_{epoch=1} - \alpha \nabla_W L(W_{epoch=1}) = [1.2 \ 1.6] - 0.1 * [4.8 \ 3.2] = [0.72 \ 1.28]$$

$$L(W_{epoch=2}) = 2 * 0.72^2 + 1.28^2 = 2.6752$$

# Optimization – Gradient Descent Example

```
In [1]: # import numpy and matplotlib
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: # define Loss and gradient functions
def loss(W):
    return 2*W[0]**2 + W[1]**2

def grad(W):
    return np.array([4*W[0], 2*W[1]])
```

```
In [3]: # initialization
W = np.array([2,2])
alpha = 0.1
nepoch = 30

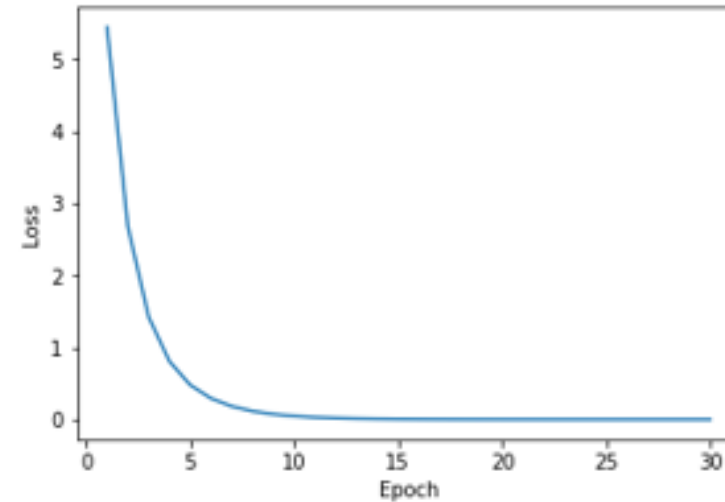
# iteration
loss_history = []
for epoch in range(nepoch):
    gradW = grad(W)
    W = W - alpha*gradW
    loss_history.append(loss(W))
print("After {} epochs".format(nepoch))
print("W: {}".format(W))
print("Loss: {}".format(loss_history[-1]))

plt.figure()
epoch_list = list(range(1, nepoch+1))
plt.plot(epoch_list, loss_history)
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.show()
```

After 30 epochs

W: [4.42147839e-07 2.47588008e-03]

Loss: 6.129982554452983e-06



# General Formula for Optimization Algorithm

- Optimization algorithms use iterative approach to approximate  $W$  that minimizes loss
- Algorithms differ in update formula - in general formula is:

$$W_{epoch=i} = W_{epoch=i-1} + Update$$

- Discuss additional optimization approaches later in course

# Optimization – Jupyter Notebook Demo

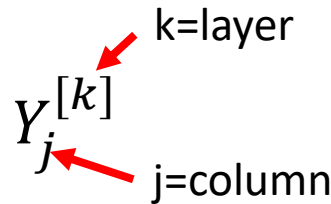
- Open file IntroML/Examples/Chapter3/Optimization.ipynb
- Has examples of
  - Optimization using 2 epochs
  - Optimization using loop for 30 epochs

# 3.4 Notation



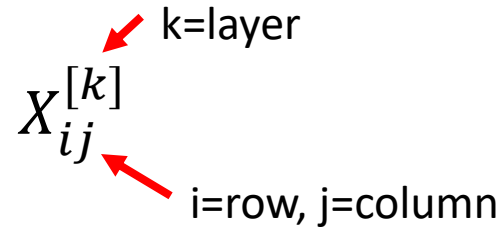
# Notation

- Throughout course we will be dealing with vectors and matrices
- For Neural Networks we will be dealing with multiple layers
- Vectors will typically be row vectors:

$$Y_j^{[k]}$$


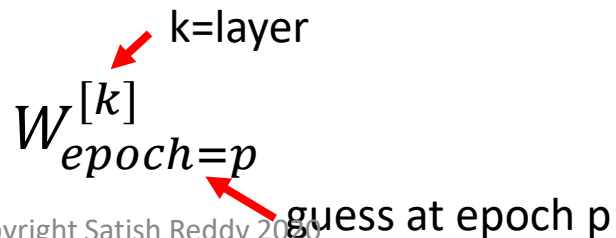
$k=\text{layer}$   
 $j=\text{column}$

- Matrix:

$$X_{ij}^{[k]}$$


$k=\text{layer}$   
 $i=\text{row}, j=\text{column}$

- Optimization (subscript epoch=p) corresponds to guess for p'th epoch. Here  $W$  could be a scalar, vector, or matrix

$$W_{\text{epoch}=p}^{[k]}$$


$k=\text{layer}$   
guess at epoch  $p$