Learning the network

11-785 Introduction to Deep Learning

– lecture 5 –

TAVE Research DL001
Heeji Won

Contents

01. Computing the derivative

02. Computing the gradient

03. Special cases

04. Vector formulation

Contents

01. Computing the derivative

02. Computing the gradient

03. Special cases

04. Vector formulation

01. Computing the derivative

- Total training loss

$$Loss = \frac{1}{T}\sum_t Div(Y_t, d_t)$$

- Total derivative

$$\frac{dLoss}{dw_{ij}^{(k)}} = \frac{1}{T}\sum_t \frac{dDiv(Y_t, d_t)}{dw_{ij}^{(k)}}$$

Want!

✓ Chain Rule

– For any nested function $y = f(g(x))$

$$z = g(x) \Rightarrow \Delta z = \frac{dg(x)}{dx}\Delta x$$

$$y = f(z) \Rightarrow \Delta y = \frac{df}{dz}\Delta z = \frac{df}{dg(x)}\frac{dg(x)}{dx}\Delta x$$

- Distributed Chain rule

$$y = f(g_1(x), g_1(x), \dots, g_M(x))$$

Let $z_i = g_i(x)$

$$\Delta y = \frac{\partial f}{\partial z_1}\Delta z_1 + \frac{\partial f}{\partial z_2}\Delta z_2 + \cdots + \frac{\partial f}{\partial z_M}\Delta z_M$$

$$\Delta y = \frac{\partial f}{\partial z_1}\frac{dz_1}{dx}\Delta x + \frac{\partial f}{\partial z_2}\frac{dz_2}{dx}\Delta x + \cdots + \frac{\partial f}{\partial z_M}\frac{dz_M}{dx}\Delta x$$

$$\boxed{\frac{dy}{dx} = \frac{\partial f}{\partial g_1(x)}\frac{dg_1(x)}{dx} + \frac{\partial f}{\partial g_2(x)}\frac{dg_2(x)}{dx} + \cdots + \frac{\partial f}{\partial g_M(x)}\frac{dg_M(x)}{dx}}$$

Contents

01. Computing the derivative

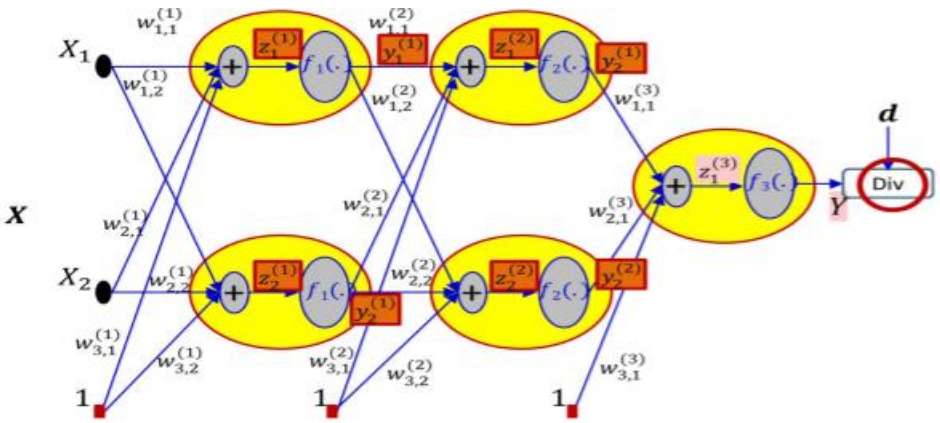
02. Computing the gradient

03. Special cases

04. Vector formulation

02. Computing the gradient

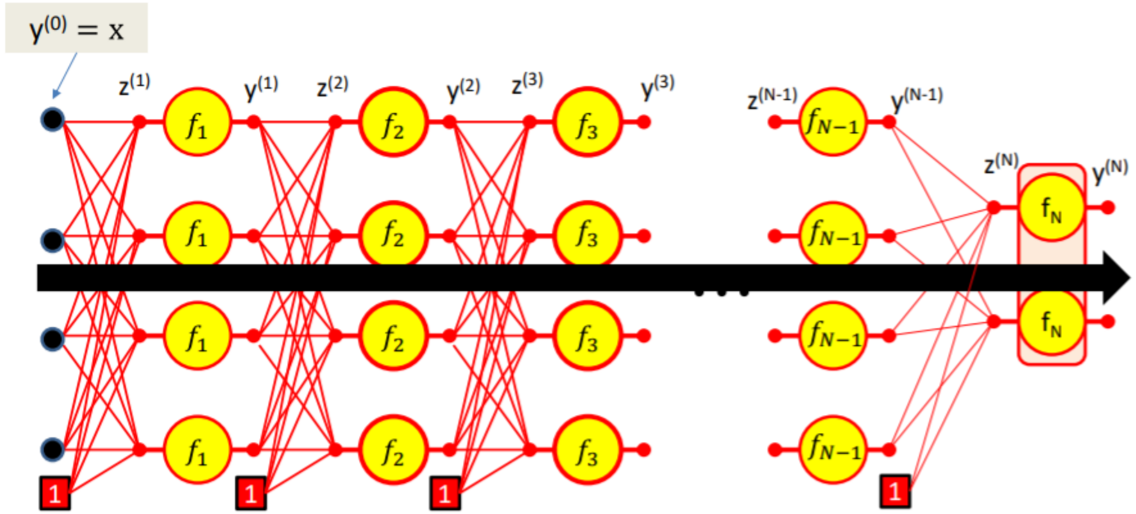
- computation of the derivative $\frac{dDiv(Y,d)}{dw_{ij}^{(k)}}$



✓ requires **intermediate and final output values** of the network in response to the input

- **The forward pass**

: the process of computing the output from an input as the forward pass



ITERATE FOR k = 1:N

for j = 1:layer-width

$$y_i^{(0)} = x_i$$

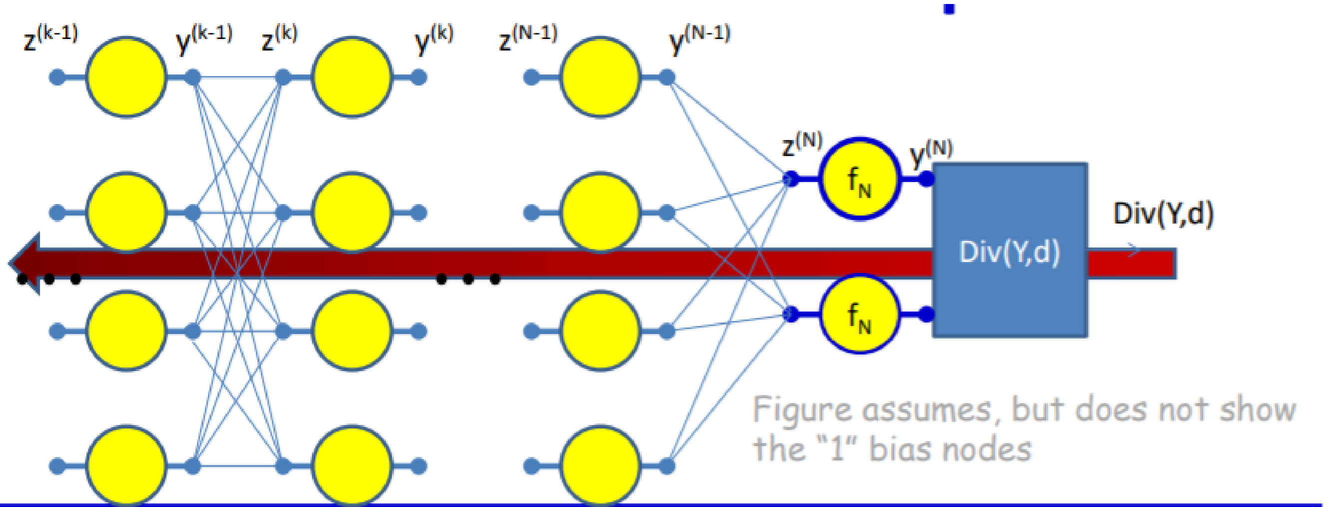
$$z_j^{(k)} = \sum_i w_{ij}^{(k)} y_i^{(k-1)}$$

$$y_j^{(k)} = f_k\left(z_j^{(k)}\right)$$

02. Computing the gradient

- The backward pass

: the process of computing the gradient from an output as the backward pass



Initialize: Gradient w.r.t network output

$$\frac{\partial Div}{\partial y_i^{(N)}} = \frac{\partial Div(Y,d)}{\partial y_i}$$

$$\frac{\partial Div}{\partial z_i^{(N)}} = f_k'\left(z_i^{(N)}\right)\frac{\partial Div}{\partial y_i^{(N)}}$$

$For\ k\ =\ N-1..0$

$For\ i\ =\ 1{:}\ layer\ width$

$$\frac{\partial Div}{\partial y_i^{(k)}} = \sum_j w_{ij}^{(k+1)}\frac{\partial Div}{\partial z_j^{(k+1)}}$$

$$\frac{\partial Div}{\partial z_i^{(k)}} = f_k'\left(z_i^{(k)}\right)\frac{\partial Div}{\partial y_i^{(k)}}$$

$$\forall j\ \frac{\partial Div}{\partial w_{ij}^{(k+1)}} = y_i^{(k)}\frac{\partial Div}{\partial z_j^{(k+1)}}$$

90

Contents

01. Computing the derivative

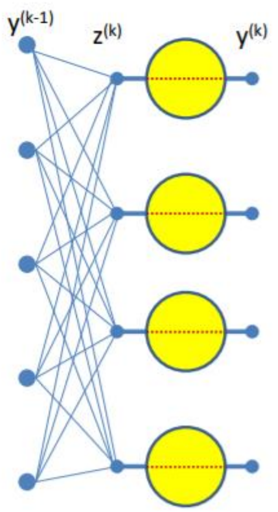
02. Computing the gradient

03. Special cases

04. Vector formulation

03. Special cases

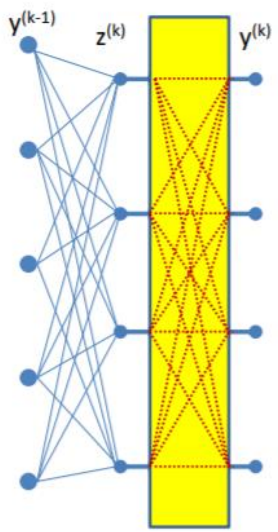
- Case 1. Vector activation



< Scalar activation >

- Each z_i
- influences one y_i

$$\frac{\partial Div}{\partial z_i^{(k)}} = \frac{\partial Div}{\partial y_i^{(k)}} \frac{dy_i^{(k)}}{dz_i^{(k)}}$$



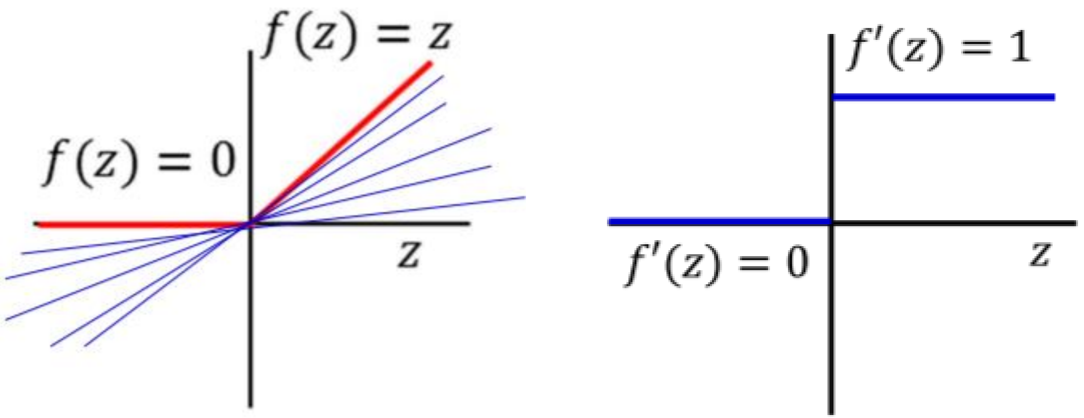
< Vector activation >

- Each z_i
- influences all, $y_1, \dots, y_M$

$$\frac{\partial Div}{\partial z_i^{(k)}} = \sum_i \frac{\partial Div}{\partial y_i^{(k)}} \frac{dy_i^{(k)}}{dz_i^{(k)}}$$

- Case 2. Non-differentiable activations

- ReLU



✓ At the differentiable points, we can use any sub-gradients

- Max

$$y = \max_j z_j \Rightarrow \frac{\partial y}{\partial z_i} = \begin{cases} 1, i = argmax_j z_j \\ 0, \quad otherwise \end{cases}$$

Contents

01. Computing the derivative

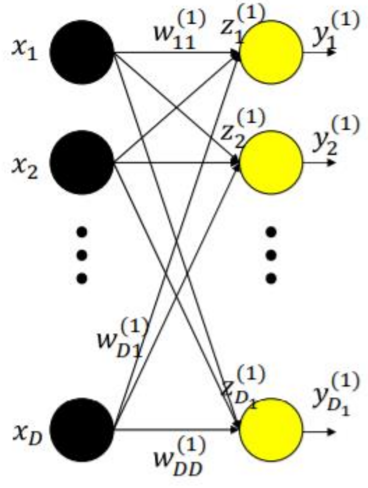
02. Computing the gradient

03. Special cases

04. Vector formulation

04. Vector formulation

- Vector formulation

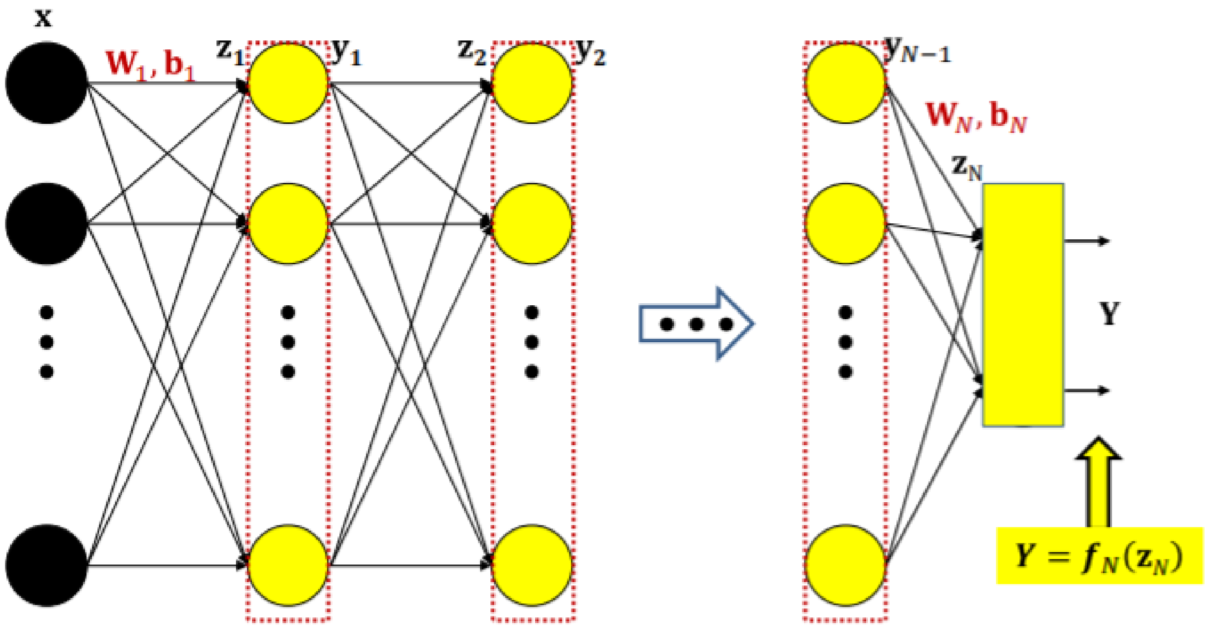


$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \qquad \mathbf{z}_k = \begin{bmatrix} z_1^{(k)} \\ z_2^{(k)} \\ \vdots \\ z_{D_k}^{(k)} \end{bmatrix} \qquad \mathbf{y}_k = \begin{bmatrix} y_1^{(k)} \\ y_2^{(k)} \\ \vdots \\ y_{D_k}^{(k)} \end{bmatrix}$$

$$\mathbf{W}_k = \begin{bmatrix} w_{11}^{(k)} & w_{21}^{(k)} & \vdots & w_{D_{k-1}1}^{(k)} \\ w_{12}^{(k)} & w_{22}^{(k)} & \vdots & w_{D_{k-1}2}^{(k)} \\ \cdots & \cdots & \ddots & \vdots \\ w_{1D_k}^{(k)} & w_{2D_k}^{(k)} & \cdots & w_{D_{k-1}D_k}^{(k)} \end{bmatrix} \qquad \mathbf{b}_k = \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \\ \vdots \\ b_{D_{k+1}}^{(k)} \end{bmatrix}$$

$$\boxed{\mathbf{z}_k = \mathbf{W}_k \mathbf{y}_{k-1} + \mathbf{b}_k} \qquad \boxed{\mathbf{y}_k = \boldsymbol{f}_k(\mathbf{z}_k)}$$

- The forward pass



The Complete computation

$$Y = f_N(\mathbf{W}_N f_{N-1}(\dots f_2(\mathbf{W}_2 f_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots) + \mathbf{b}_N)$$

$$Div(Y, d) = Div(f_N(\mathbf{W}_N f_{N-1}(\dots f_2(\mathbf{W}_2 f_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots) + \mathbf{b}_N), d)$$

04. Vector formulation

- The Jacobian
  - The distributed chain rule

$$y = f\big(g_1(x), g_1(x), \dots, g_M(x)\big)$$

$$\downarrow$$

$$\Delta y = \frac{\partial f}{\partial z_1}\Delta z_1 + \frac{\partial f}{\partial z_2}\Delta z_2 + \cdots + \frac{\partial f}{\partial z_M}\Delta z_M$$

✓ What if y is a **vector**?

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = f\left(\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_D \end{bmatrix}\right) \longrightarrow J_{\mathbf{y}}(\mathbf{z}) = \begin{bmatrix} \frac{\partial y_1}{\partial z_1} & \frac{\partial y_1}{\partial z_2} & \cdots & \frac{\partial y_1}{\partial z_D} \\ \frac{\partial y_2}{\partial z_1} & \frac{\partial y_2}{\partial z_2} & \cdots & \frac{\partial y_2}{\partial z_D} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial y_M}{\partial z_1} & \frac{\partial y_M}{\partial z_2} & \cdots & \frac{\partial y_M}{\partial z_D} \end{bmatrix}$$

$$\Delta \mathbf{y} = J_{\mathbf{y}}(\mathbf{z})\Delta \mathbf{z}$$

- Chain rule
  - For vector functions of vector inputs

$$\boldsymbol{y} = \boldsymbol{f}\big(\mathbf{z}(\boldsymbol{x})\big) \;\rightarrow\; J_{\boldsymbol{y}}(\boldsymbol{x}) = J_{\boldsymbol{y}}(\mathbf{z}) J_{\mathbf{z}}(\boldsymbol{x})$$

  - For scalar functions of vector inputs

$$D = f\big(\mathbf{z}(\boldsymbol{x})\big) \;\rightarrow\; \nabla_{\boldsymbol{x}} D = \nabla_{\mathbf{z}}(D) J_{\mathbf{z}}(\boldsymbol{x})$$

$$(\because \Delta D = \nabla_{\mathbf{z}}(D)\Delta\mathbf{z}, \qquad \Delta\mathbf{z} = J_{\mathbf{z}}(x)\Delta x)$$

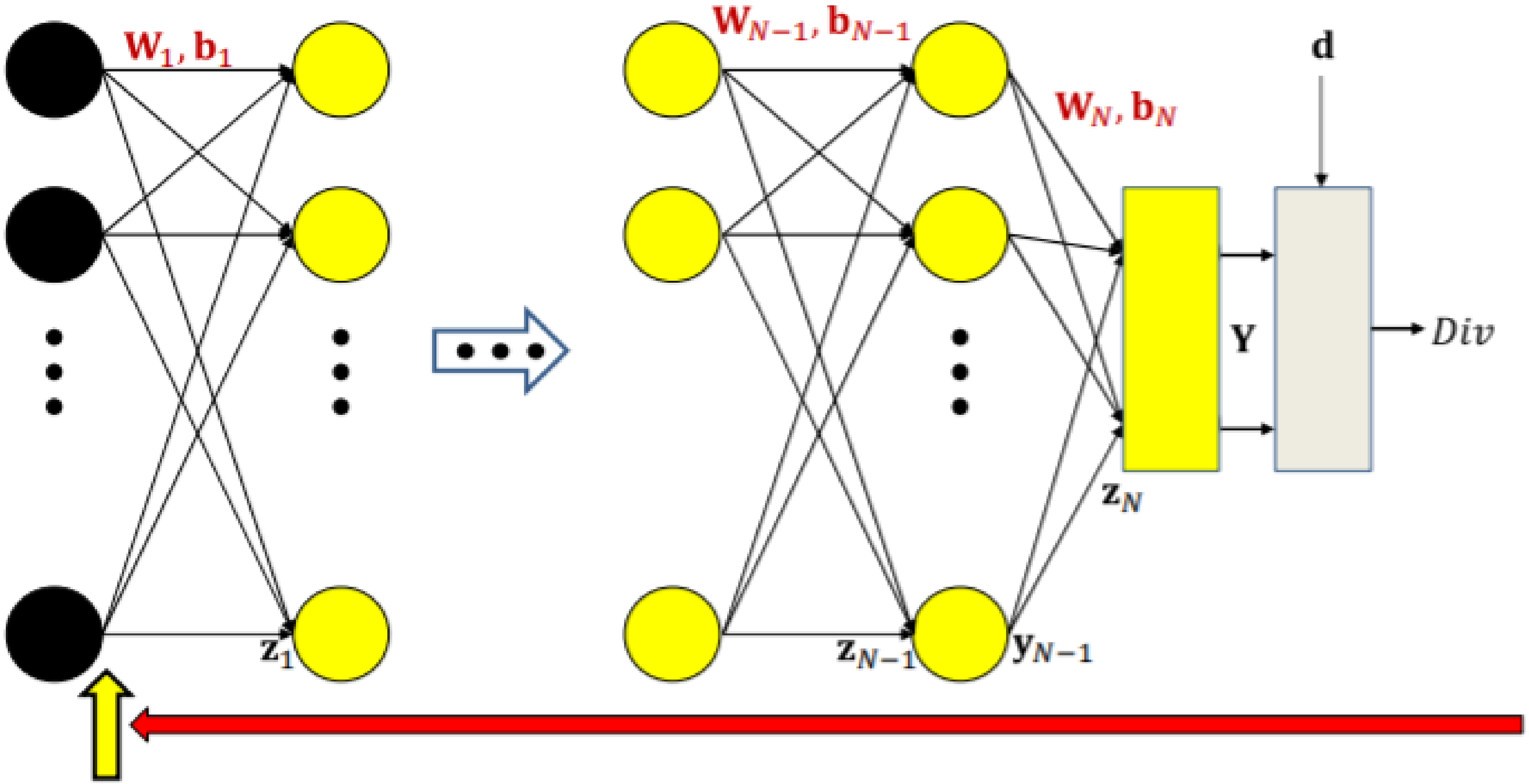
  - Affine functions

$$z = Wy + b \;\rightarrow\; J_z(y) = W$$

$$(\because W(y + \Delta y) + b = (Wy + b) + W\Delta y)$$
$$= z + \Delta \mathbf{z}$$

04. Vector formulation

- The backward pass



$$\nabla_{\mathbf{W}_1} Div = \mathbf{x} \nabla_{\mathbf{z}_1} Div$$

$$\nabla_{\mathbf{b}_1} Div = \nabla_{\mathbf{z}_1} Div$$

In some problems we will also want to compute the derivative w.r.t. the input

Thank you