The Mathemmatical Engineering of Deep Learning

Chapter 2 - Lecture 2 - Part (3/3)

B. Liquet[1,2] and S. Moka[3] and Y. Nazarathy[3]

[1] Macquarie University [2] LMAP, Université de Pau et des Pays de L'Adour' [3] The University of Queensland

Outline of Lecture

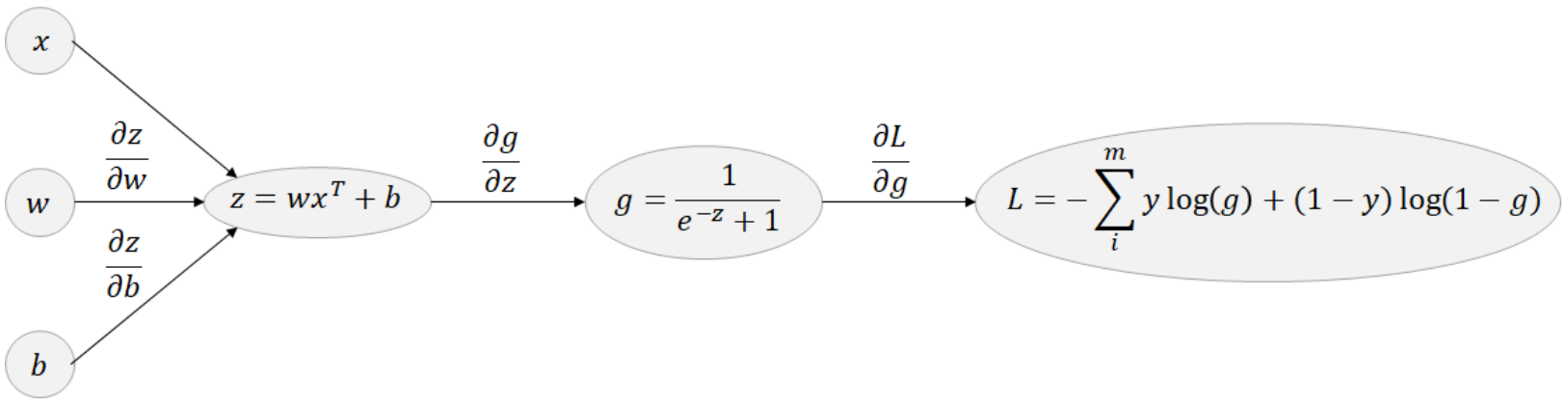
- Backpropagation
  - Chain rule
  - comptational graph
- Implementation

Backpropagation

Backpropagation

The key tool adopted by the neural network community to update the weights. This method exploits the derivative with respect to each weight *w* using the **chain rule** (univariate and multivariate rules)

The chain rule is used and generally illustrated through a **computation graph**



Chain rule

The univariate **chain rule** and the multivariate chain rule are the keys concept to calculate the derivative of cost with respect to any weight in the network.

- Univariate chain rule

$$\frac{\partial f(g(w))}{\partial w} = \frac{\partial f(g(w))}{\partial g(w)} \cdot \frac{\partial g(w)}{\partial w}$$

Example: univariate chain rule

Univariate logistic least squares model

$$
\begin{aligned}
z &= wx + b \\
y &= \sigma(z) \\
\mathcal{L} &= \frac{1}{2}(y - t)^2
\end{aligned}
$$

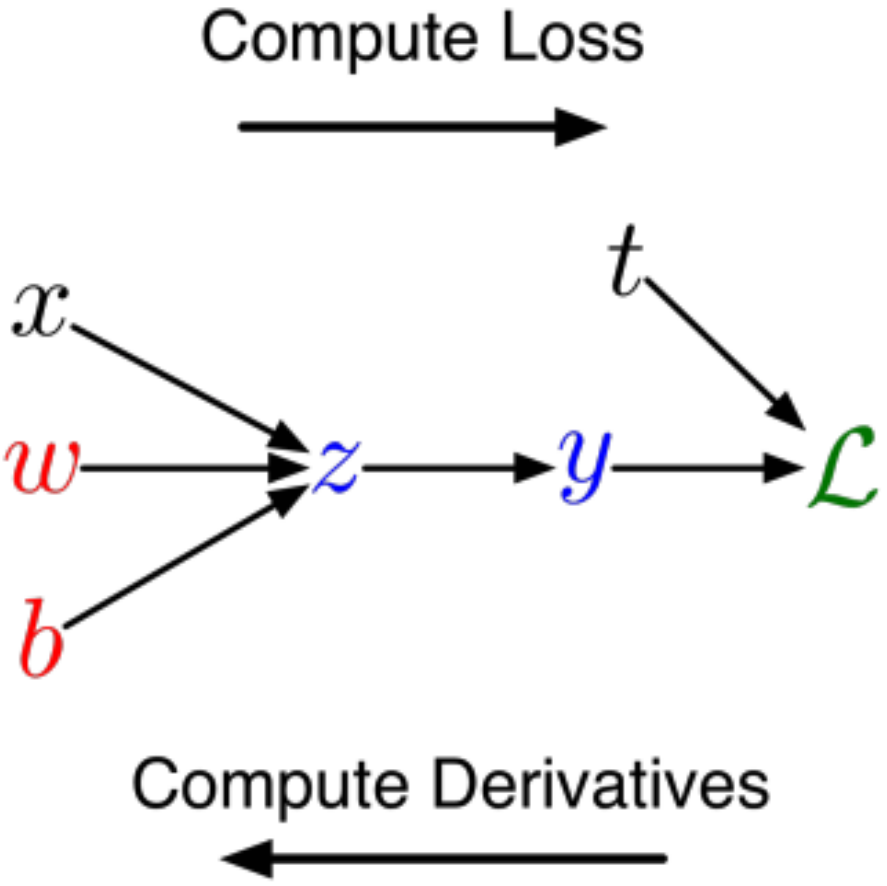
Computing the derivatives using chain rules

$$
\begin{aligned}
\frac{\partial L}{\partial y} &= y - t \\
\frac{\partial L}{\partial z} &= \frac{\partial L}{\partial y}\sigma'(z) \\
\frac{\partial L}{\partial w} &= \frac{\partial L}{\partial z}x \\
\frac{\partial L}{\partial b} &= \frac{\partial L}{\partial z}
\end{aligned}
$$

Goal: write a program that efficiently computes the derivatives and not get closed form

Computation graph.

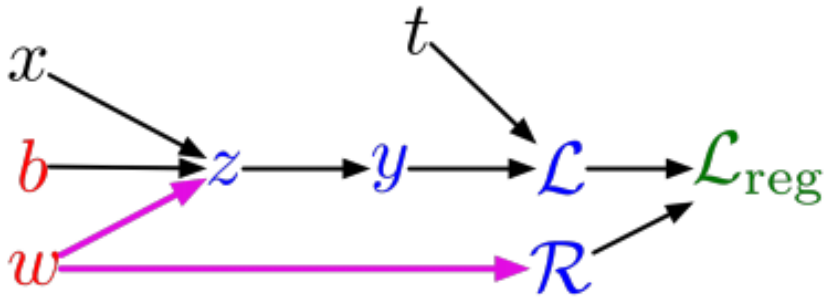
Help: Draw a diagram, Computation graph, to represent our computations



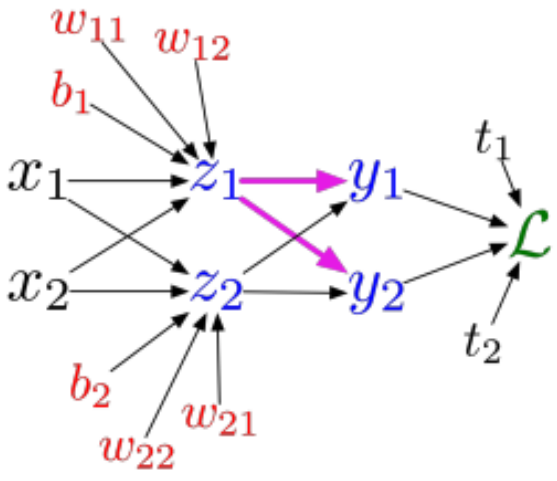
New Challenge

Need multivariate Chain Rule when computation graph has fan-out > 1.

L_2-Regularized regression



Multiclass logistic regression



Multivariate Chain Rule

Part I

Let $z = f(x, y)$, $x = g(t)$ and $y = h(t)$, where f, g and h are differentiable functions. Then $z = f(x, y) = f(g(t), h(t)))$ is a function of t, and

$$\begin{aligned} \frac{dz}{dt} = \frac{df}{dt} &= f_x(x, y)\frac{dx}{dt} + f_y(x, y)\frac{dy}{dt} \\ &= \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}. \end{aligned}$$

Multivariate Chain Rule

Part II

a) Let $z = f(x, y)$, $x = g(s, t)$ and $y = h(s, t)$, where f, g and h are differentiable functions. Then z is a function of s and t, and

$$\frac{\partial z}{\partial s} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial s}$$

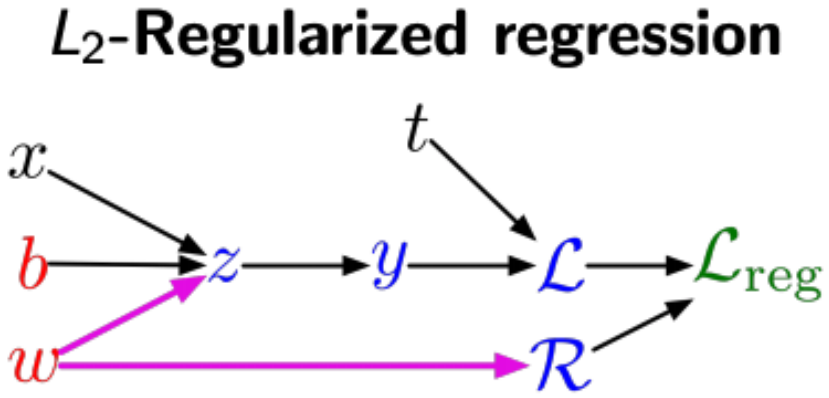
and

$$\frac{\partial z}{\partial t} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t}$$

b) Let $z = f(x_1, x_2, \ldots, x_m)$ be a differentiable function of m variables, where each of the x_i is a differentiable function of the variables $t_1, t_2, \ldots, t_n$. Then z is a function of the t_i, and

$$\frac{\partial z}{\partial t_i} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t_i} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t_i} + \cdots + \frac{\partial f}{\partial x_m}\frac{\partial x_m}{\partial t_i}.$$

Case of Regularized regression

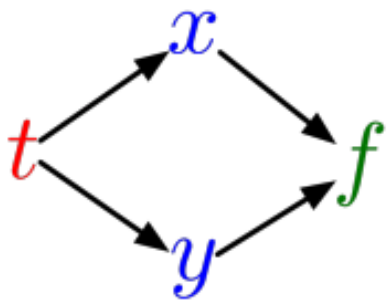


Forward pass:

$$
\begin{aligned}
z &= wx + b \\
y &= \sigma(z) \\
\mathcal{L} &= \frac{1}{2}(y - t)^2 \\
R &= \frac{1}{2}w^2 \\
\mathcal{L}_{reg} &= L + \lambda R
\end{aligned}
$$

Multivariate Chain Rule: understand

- Consider a function $f(x, y)$ and functions $x(t)$ and $y(t)$. All the variables are scalar-valued.



$$\frac{d}{dt} f(x(t), y(t)) = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$

Multivariate Chain Rule: understand

- Example

$$\begin{aligned} f(x,y) &= y + e^{xy} \\ x(t) &= cost \\ y(t) &= t^2 \end{aligned}$$

- Using the chain rule

$$\begin{aligned} \frac{df}{dt} &= \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \\ &= (ye^{xy}).(-sint) + (1 + xe^{xy}).2t \end{aligned}$$

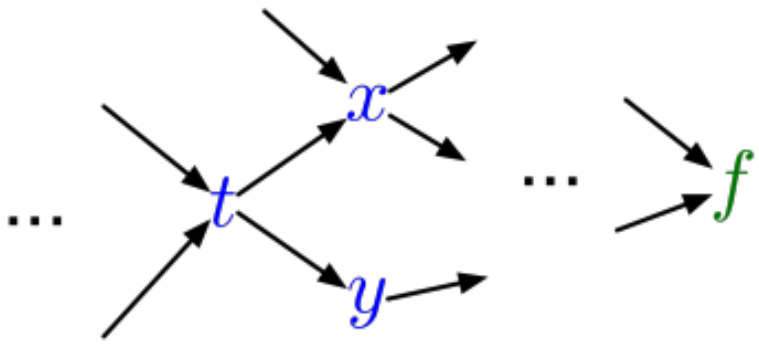
Multivariate Chain Rule: general example

- Backpropagation framework

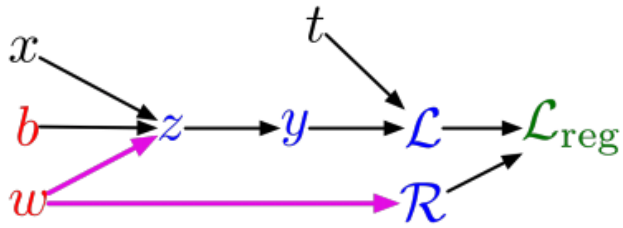
Mathematical expressions to be evaluated

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}$$

Values already computed by our program



L_2-Regularized regression



Forward pass

$$z = wx + b \rightarrow y = \sigma(z) \rightarrow \mathcal{L} = \frac{1}{2}(y - t)^2 \rightarrow \mathcal{R} = \frac{1}{2}w^2 \rightarrow \mathcal{L}_{reg} = L + \lambda\mathcal{R}$$

Backward pass

$$\boxed{\frac{\partial \mathcal{L}_{reg}}{\partial \mathcal{R}} = \lambda} \rightarrow \boxed{\frac{\partial \mathcal{L}_{reg}}{\partial \mathcal{L}} = 1} \rightarrow \boxed{\frac{\partial \mathcal{L}_{reg}}{\partial y} = \frac{\partial \mathcal{L}_{reg}}{\partial \mathcal{L}} \times \frac{\partial \mathcal{L}}{\partial y} = 1 \times (y - t)}$$

$$\rightarrow \boxed{\frac{\partial \mathcal{L}_{reg}}{\partial z} = \frac{\partial \mathcal{L}_{reg}}{\partial y} \times \frac{\partial \dagger}{\partial z} = \frac{\partial \mathcal{L}_{reg}}{\partial y} \times \sigma'(z)}$$

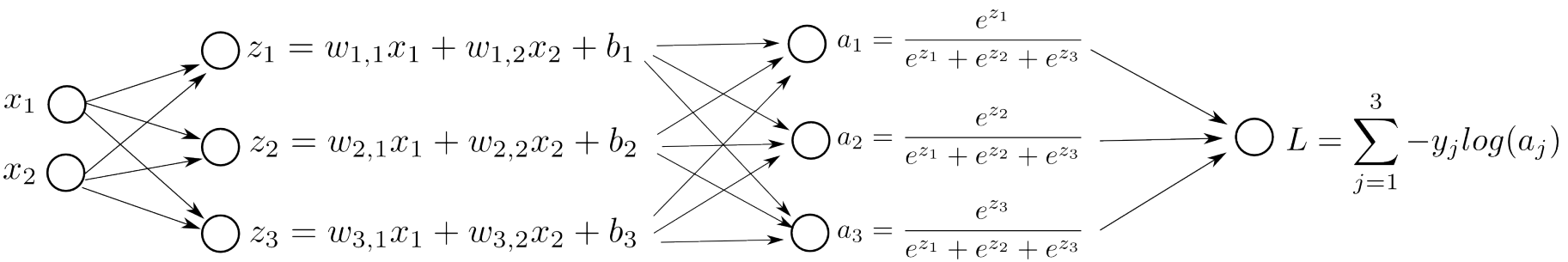
$$\rightarrow \boxed{\frac{\partial \mathcal{L}_{reg}}{\partial w} = \frac{\partial \mathcal{L}_{reg}}{\partial z} \times \frac{\partial = z}{\partial w} + \frac{\partial \mathcal{L}_{reg}}{\partial \mathcal{R}} \times \frac{\partial \mathcal{R}}{\partial w} = \frac{\partial \mathcal{L}_{reg}}{\partial z}x + \frac{\partial \mathcal{L}_{reg}}{\partial \mathcal{R}}w}$$

$$\rightarrow \boxed{\frac{\partial \mathcal{L}_{reg}}{\partial b} = \frac{\partial \mathcal{L}_{reg}}{\partial z} \times \frac{\partial z}{\partial b} = \frac{\partial \mathcal{L}_{reg}}{\partial z}}$$

Example: softmax neural network

Let consider a simple example with

- $K = 3$ ($y \in \{1, 2, 3\}$) and two features (x_1 and x_2).
- $w_j \in \mathfrak{R}^2$ and $b_j \in \mathfrak{R}$, $j = 1, 2, 3$.



Let's write as an example for $\frac{\partial L}{\partial w_{2,1}}$:

$$\frac{\partial L}{\partial w_{2,1}} = \sum_{i=1}^{3} \left(\frac{\partial L}{\partial a_i}\right)\left(\frac{\partial a_i}{\partial z_2}\right)\left(\frac{\partial z_2}{\partial w_{2,1}}\right)$$

$$= \left(\frac{\partial L}{\partial a_1}\right)\left(\frac{\partial a_1}{\partial z_2}\right)\left(\frac{\partial z_2}{\partial w_{2,1}}\right) + \left(\frac{\partial L}{\partial a_2}\right)\left(\frac{\partial a_2}{\partial z_2}\right)\left(\frac{\partial z_2}{\partial w_{2,1}}\right) + \left(\frac{\partial L}{\partial a_3}\right)\left(\frac{\partial a_3}{\partial z_2}\right)\left(\frac{\partial z_2}{\partial w_{2,1}}\right)$$

When we change $w_{2,1}$; a_1 a_2 and a_3 changes as a result. Then, the change of a_1 a_2 and a_3 affects L. We sum up all the changes $w_{2,1}$ produced over a_1 a_2 and

Updating weights using Backpropagation

For neural network framework, the weight is updated using gradient descent concepts

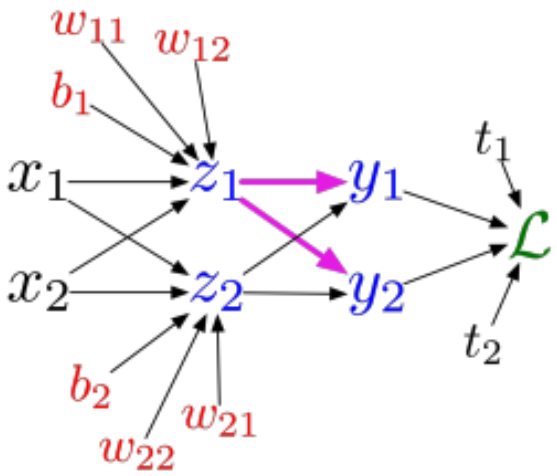
$$w = w - \alpha \frac{\partial J(w, b)}{\partial w}$$

The main steps for updating weights are

1. Take a batch of training sample
2. Forward propagation to get the corresponding cost
3. Backpropagate the cost to get the gradients
4. update the weights using the gradients
5. Repeat step 1 to 4 for a number of iterations

Your turn to play

Multiclass logistic regression



$$z_\ell = \sum_j w_{\ell j} x_j + b_\ell$$

$$y_k = \frac{e^{z_k}}{\sum_\ell e^{z_\ell}}$$

$$\mathcal{L} = -\sum_k t_k \times \ln y_k$$

TO DO: Forward pass and Backward pass

Take Home Message

- Forward propagation: input x is propagated through the network (graph) to reach the output $\widehat{y}$. Then the cost could be computed $\mathcal{J}(\theta)$
- Back-propagation: process to compute the gradient by using repeatedly the chain rule coupled with a memory that allows to reuse computations.