# Neural Network – Back-propagation

HYUNG IL KOO

# Hidden Layer Representations

- Backpropagation has an ability to discover useful intermediate representations at the hidden unit layers inside the networks which capture properties of the input spaces that are most relevant to learning the target function.
- When more layers of units are used in the network, more complex features can be invented.
- But the representations of the hidden layers are very hard to understand for humans.

# Basic Math

# Optimization

- **Find $x$ that minimizes $f(x)$**

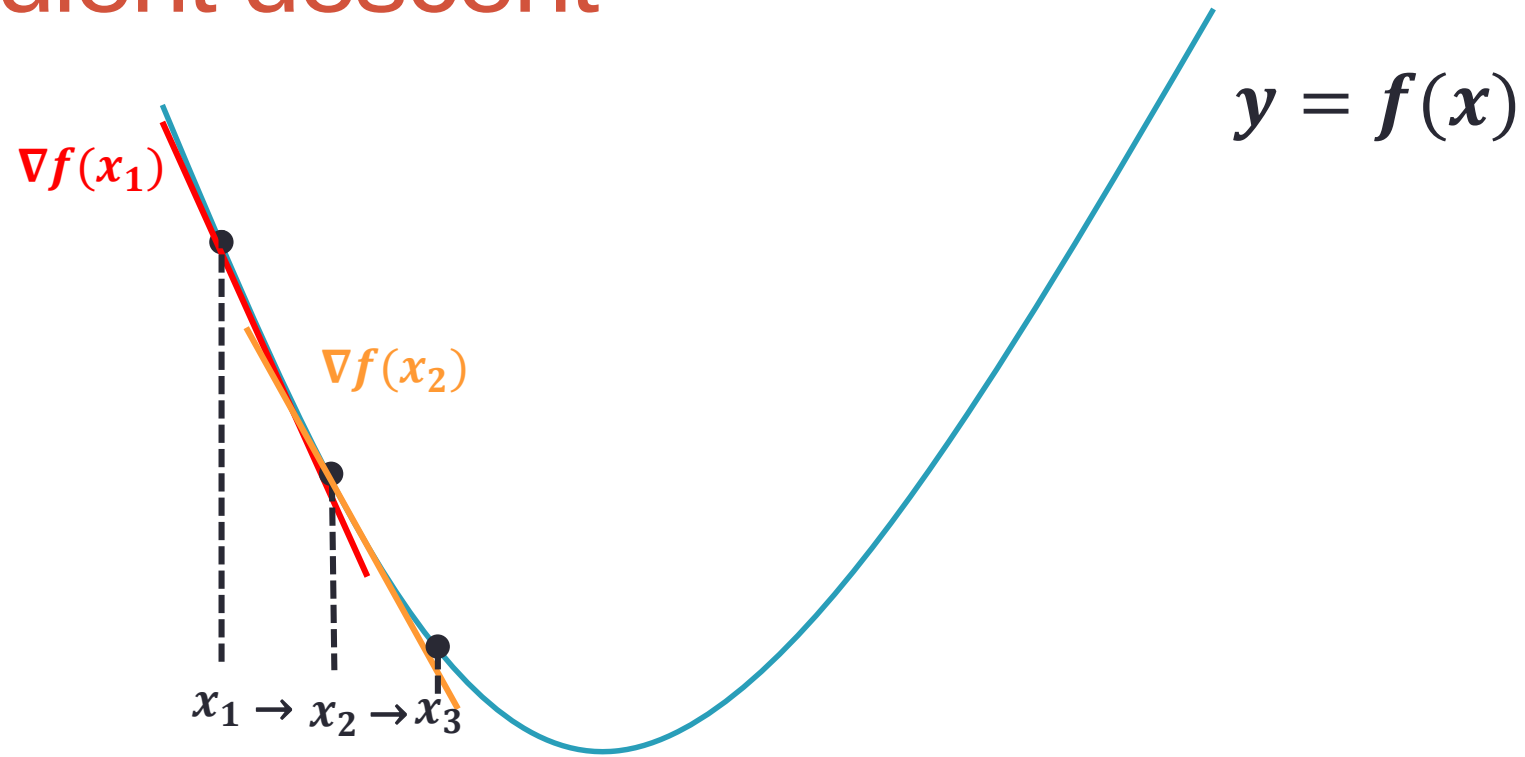
$$\hat{x} = \arg\min f(x)$$

- **If $f(x)$ is differentiable,**

$$\nabla f(x) = 0$$

- **But, in many cases, solving the above equation is a still difficult problem.**

# Gradient descent




$$x_{n+1} = x_n - \alpha \nabla f(x_n)$$

# Chain Rule

- **Chain rule with a single variable**

$$y = f(g(x)) \Rightarrow y' = f'(g(x))g'(x)$$

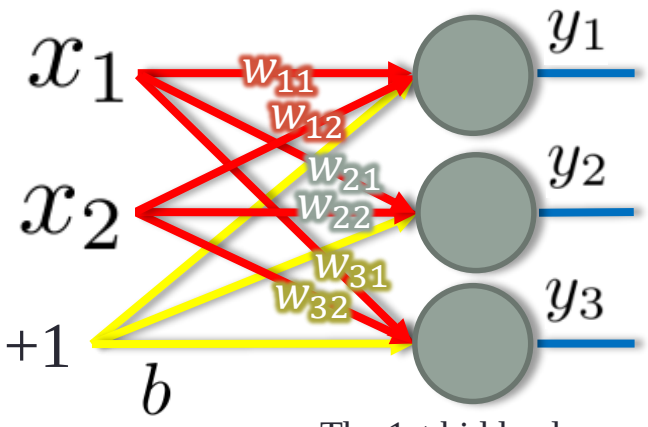
- **Chain Rule (multiple variables)**

$$w = f(x, y, z) \Rightarrow \quad \frac{dw}{dt} = \frac{\partial f}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial f}{\partial z} \cdot \frac{dz}{dt}$$

$$\Delta w \simeq \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial z} \Delta z$$

# Feed-forward neural network

# Feed forward network example: 1st layer




The 1st hidden layer

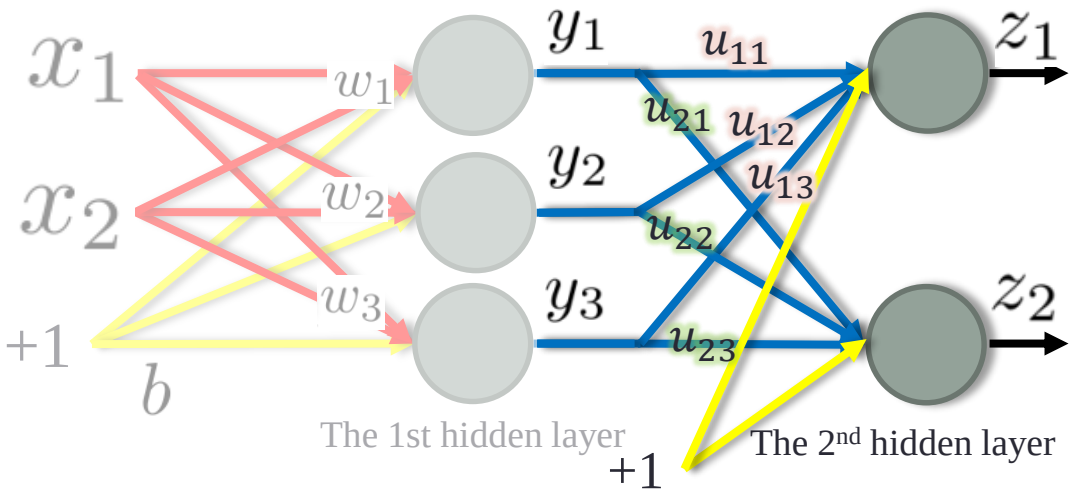
$$y_1 = \varphi(w_{11}x_1 + w_{12}x_2 + b_1)$$

$$y_2 = \varphi(w_{21}x_1 + w_{22}x_2 + b_2)$$

$$y_3 = \varphi(w_{31}x_1 + w_{32}x_2 + b_3)$$

Non-linear function

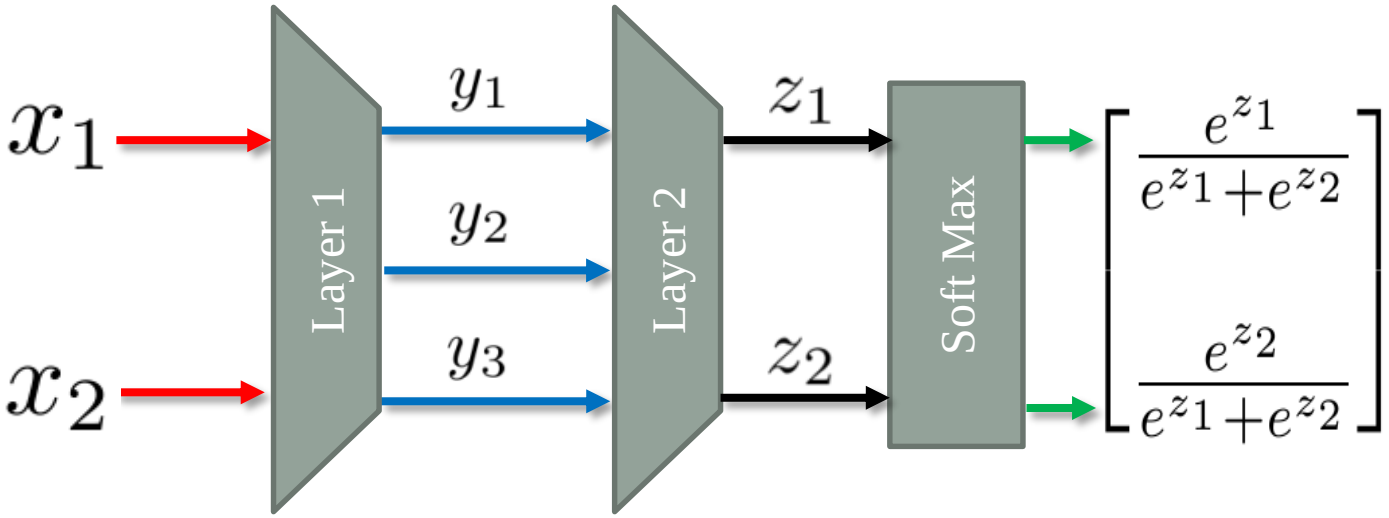
# Feed forward network example: 2nd layer




$$z_1 = \varphi(u_{11}y_1 + u_{12}y_2 + u_{13}y_3 + c_1)$$

$$z_2 = \varphi(u_{21}y_1 + u_{22}y_2 + u_{23}y_3 + c_2)$$

# Forward propagation




The 1st hidden layer  The 2nd hidden layer  Output

$$y_1 = \varphi(w_{11}x_1 + w_{12}x_2 + b_1)$$

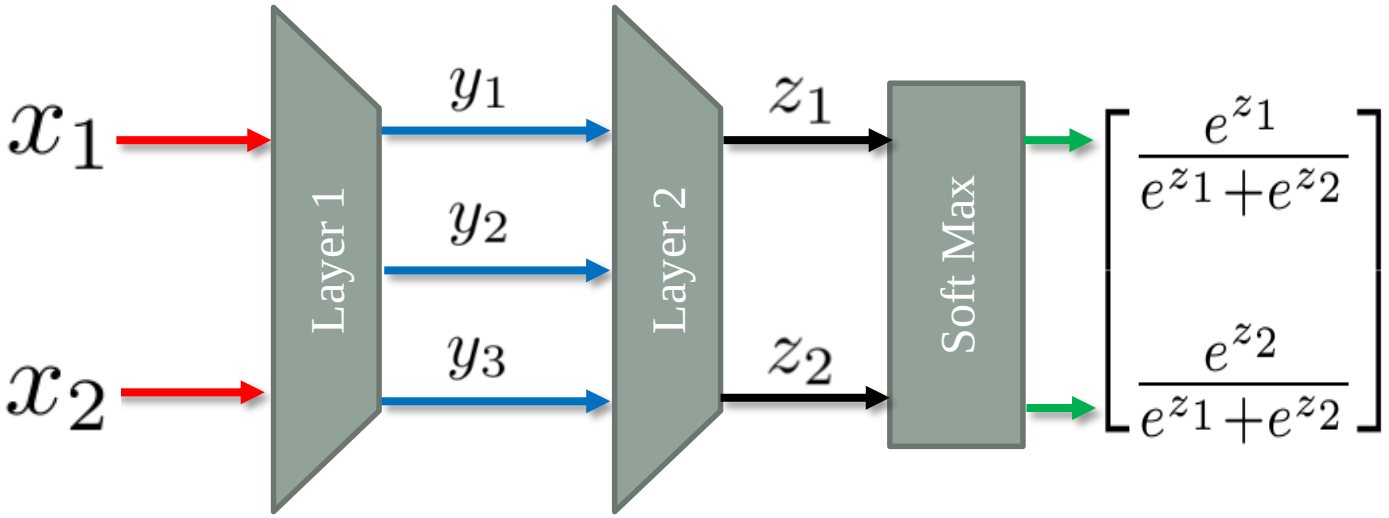
$$y_2 = \varphi(w_{21}x_1 + w_{22}x_2 + b_2)$$

$$y_3 = \varphi(w_{31}x_1 + w_{32}x_2 + b_3)$$

$$z_1 = \varphi(u_{11}y_1 + u_{12}y_2 + u_{13}y_3 + c_1)$$

$$z_2 = \varphi(u_{21}y_1 + u_{22}y_2 + u_{23}y_3 + c_2)$$

# Forward propagation matrix repr.



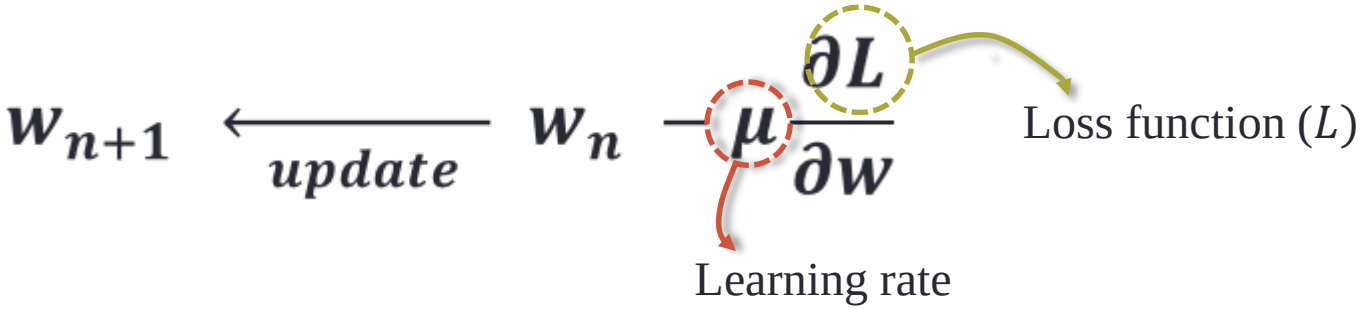

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \varphi\left( \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \varphi\left( \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right)$$

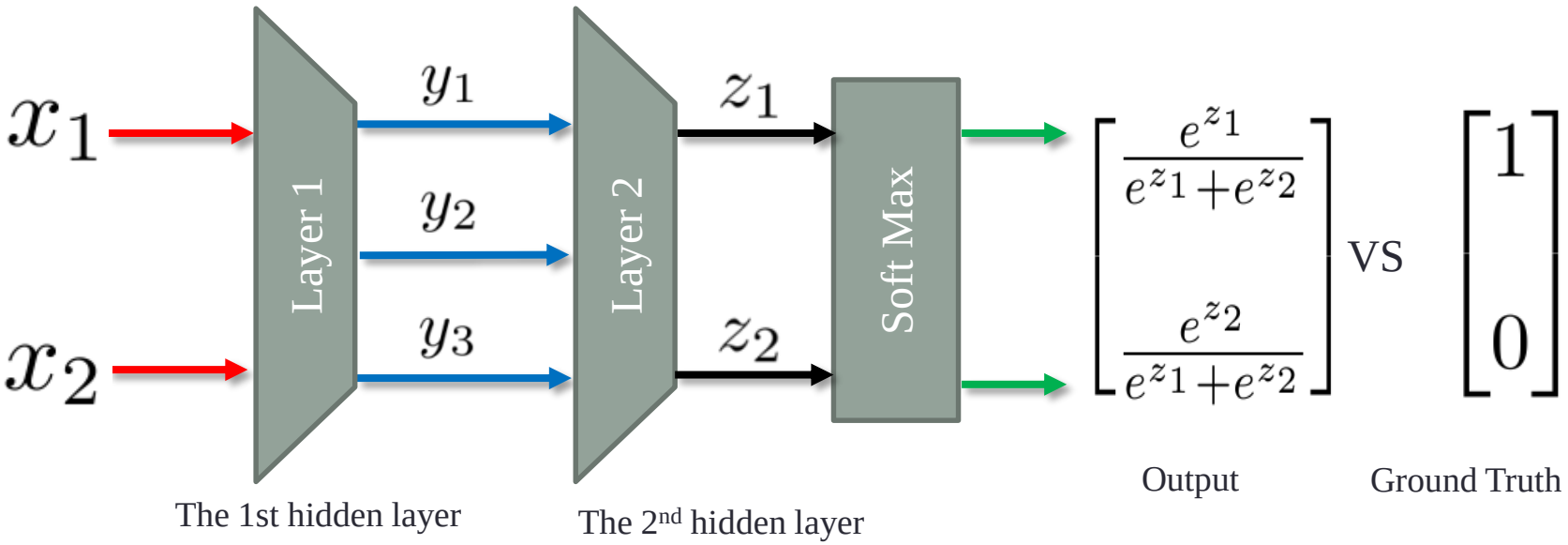
# Back-propagation algorithm

# Weight update method

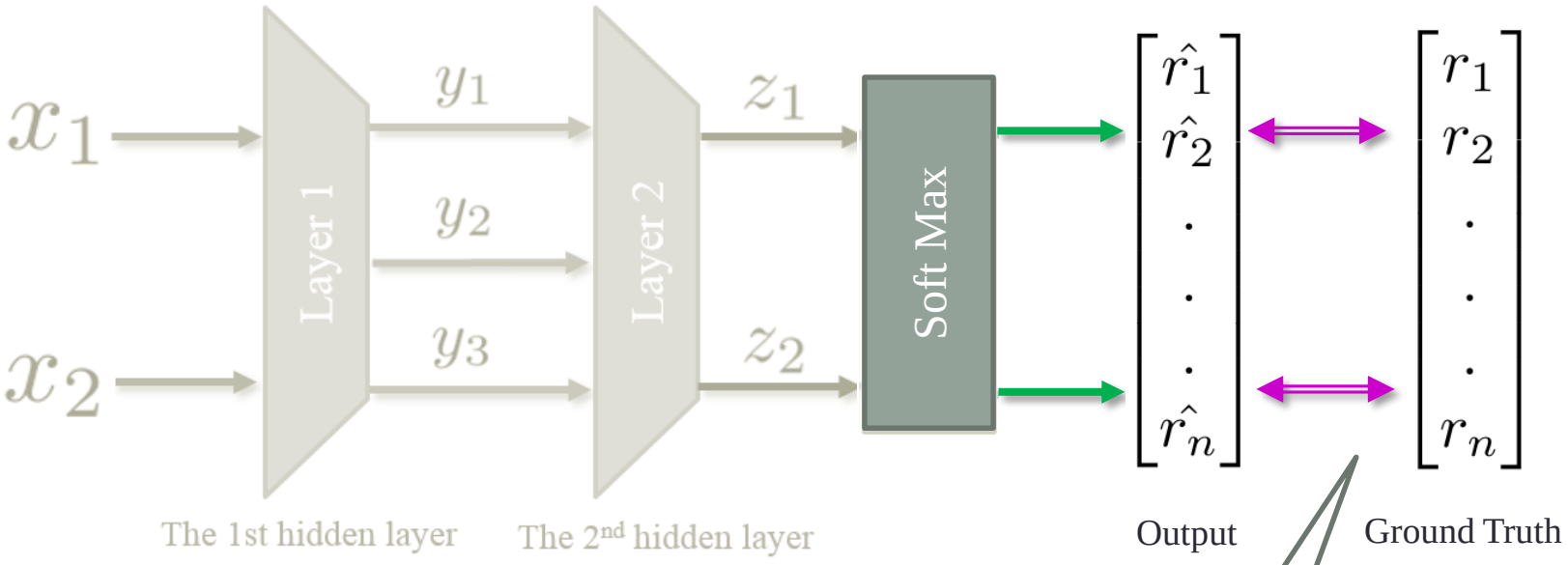
- **Parameter update : Gradient Descent**



# Dataflow diagram



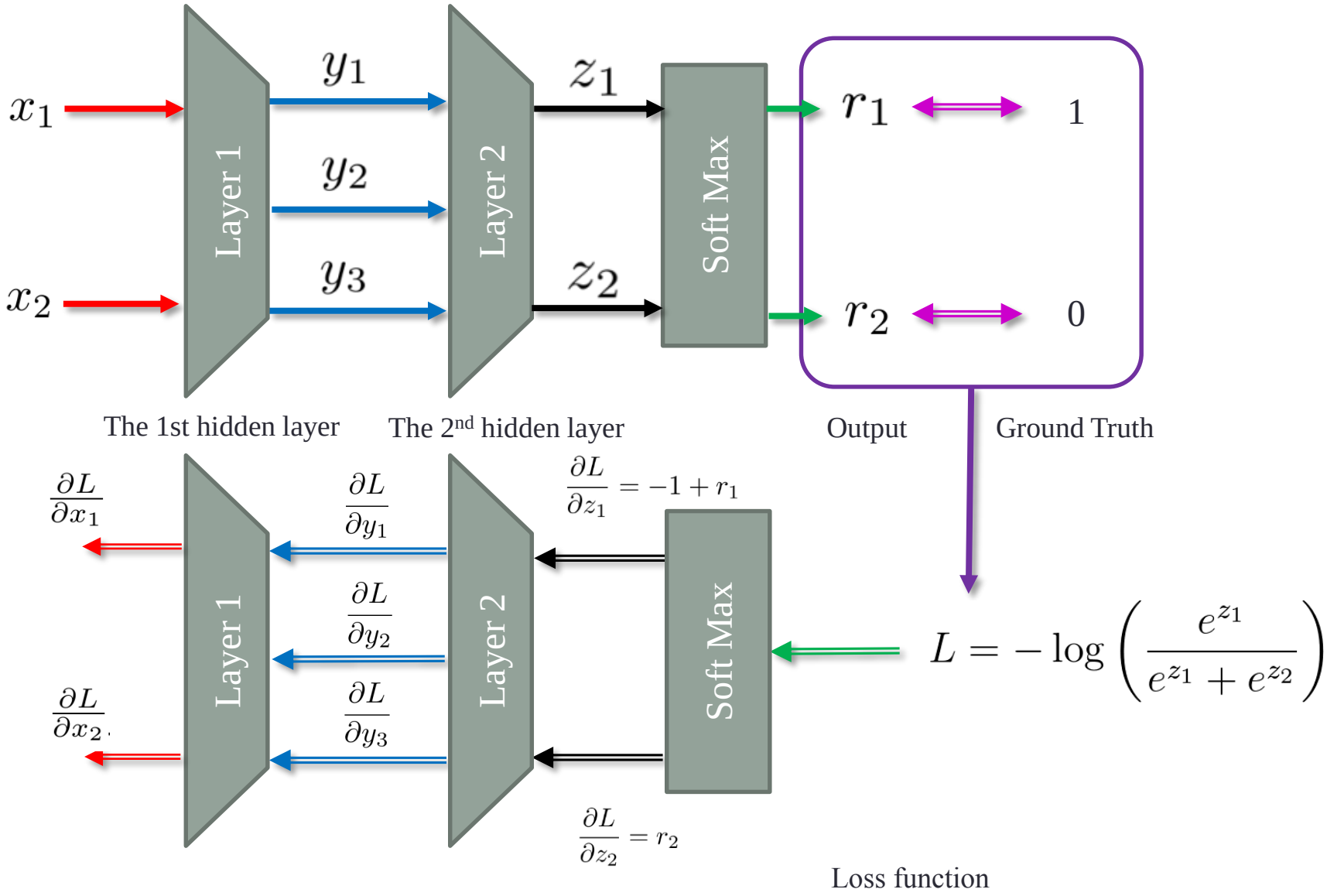
# Back-propagation step; Loss function



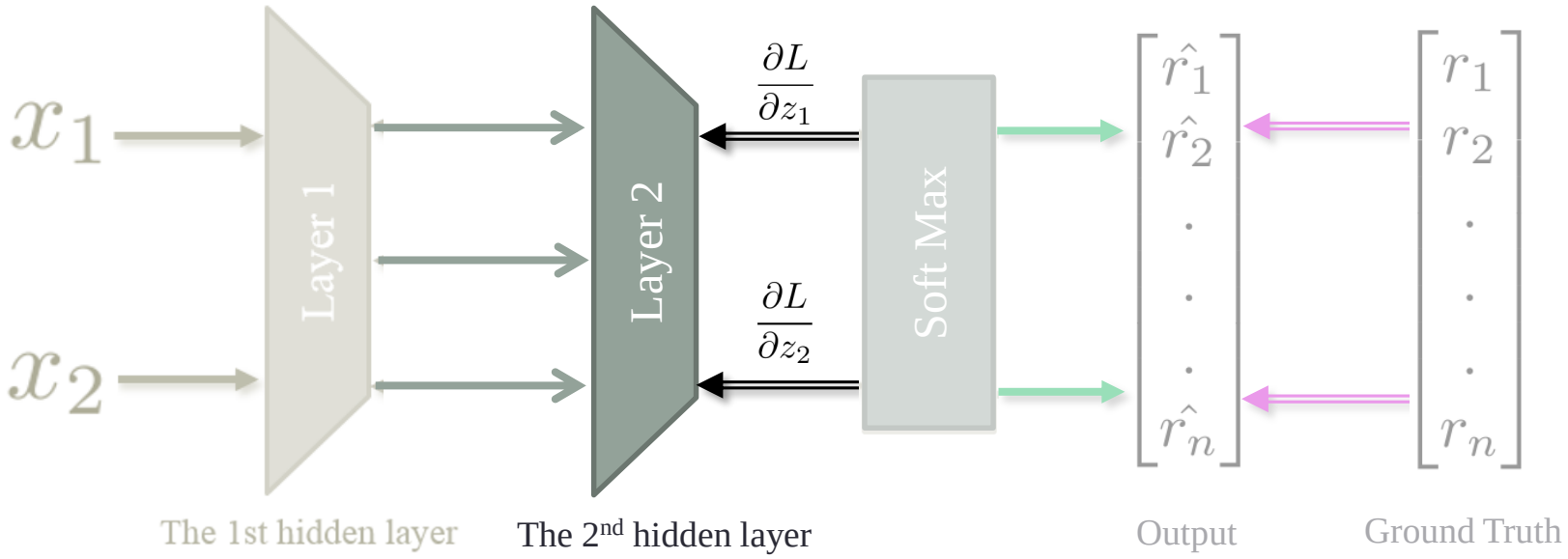

Computing Loss function($L$):
ex) Cross entropy

$$L = -\sum_{i=1}^{n} r_i \log \hat{r}_i$$

# Overview


$x_1$
$x_2$
Layer 1
$y_1$
$y_2$
$y_3$
Layer 2
$z_1$
$z_2$
Soft Max
$r_1$
1
$r_2$
0
The 1st hidden layer
The 2nd hidden layer
Output
Ground Truth
$\frac{\partial L}{\partial x_1}$
$\frac{\partial L}{\partial x_2}$
$\frac{\partial L}{\partial y_1}$
$\frac{\partial L}{\partial y_2}$
$\frac{\partial L}{\partial y_3}$
$\frac{\partial L}{\partial z_1} = -1 + r_1$
$\frac{\partial L}{\partial z_2} = r_2$
$L = -\log\left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}}\right)$
Loss function

# Back-propagation; 2nd layer



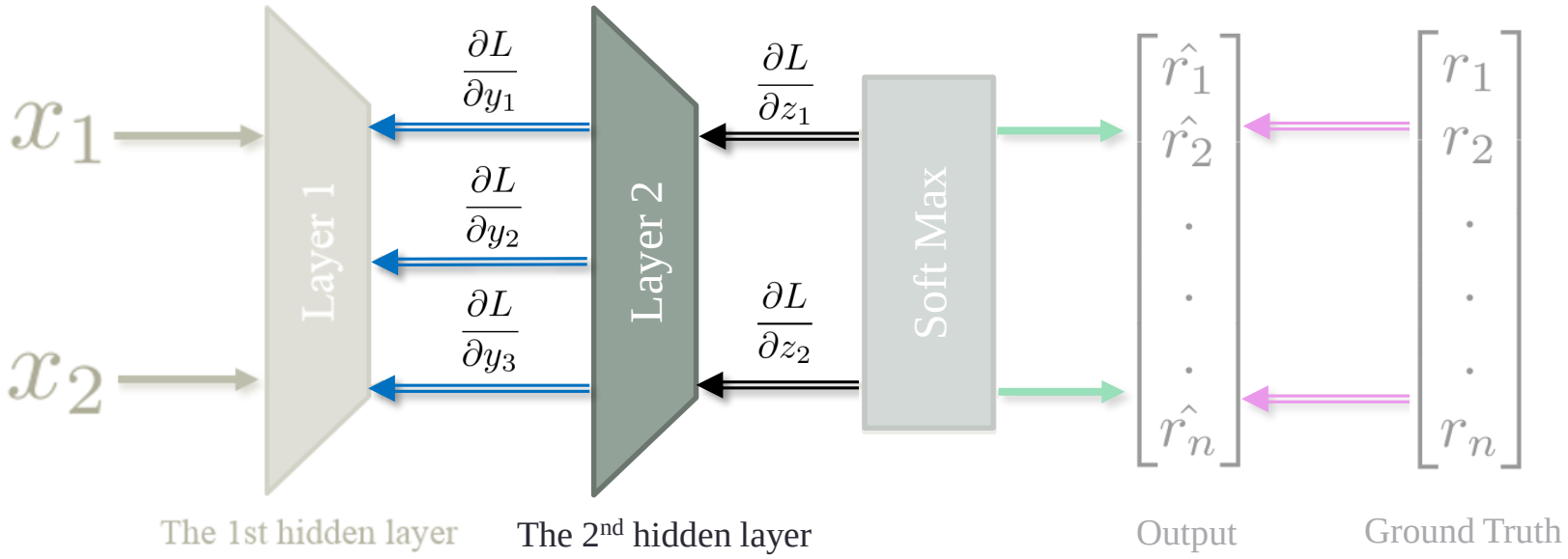

- the Layer 2 has to do
  - Weight update

$$u_{ij}^{new} = u_{ij}^{old} - \mu \frac{\partial L}{\partial u_{ij}}$$

  - Error propagation

$$\frac{\partial L}{\partial y_i} = \sum_{j=1}^{N(z)} \frac{\partial L}{\partial z_j} \frac{\partial z_j}{\partial y_i}$$

# Back-propagation; 2nd layer




- the Layer 2 has to do
  - Weight update

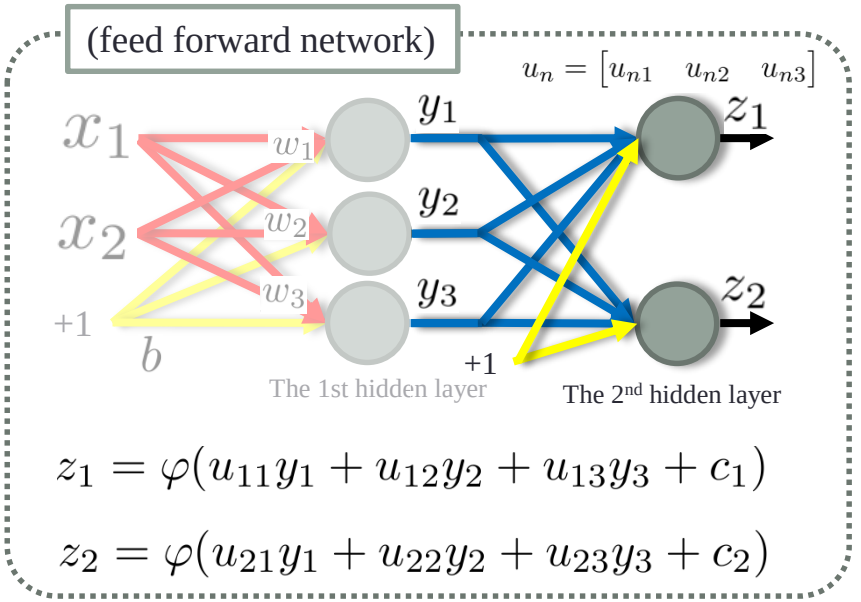
$$u_{ij}^{new} = u_{ij}^{old} - \mu \frac{\partial L}{\partial u_{ij}}$$

  - Error propagation

$$\frac{\partial L}{\partial y_i} = \sum_{j=1}^{N(z)} \frac{\partial L}{\partial z_j} \frac{\partial z_j}{\partial y_i}$$

# Error propagation

$$\frac{\partial L}{\partial y_1} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial y_1} + \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial y_1}$$




$$\frac{\partial L}{\partial y_1} = \frac{\partial L}{\partial z_1}\varphi'(\boldsymbol{u_{11}y_1} + u_{12}y_2 + u_{13}y_3 + c_1)u_{11} + \frac{\partial L}{\partial z_2}\varphi'(\boldsymbol{u_{21}y_1} + u_{22}y_2 + u_{23}y_3 + c_2)u_{21}$$

$$\frac{\partial L}{\partial y_2} = \frac{\partial L}{\partial z_1}\varphi'(u_{11}y_1 + \boldsymbol{u_{12}y_2} + u_{13}y_3 + c_1)u_{12} + \frac{\partial L}{\partial z_2}\varphi'(u_{21}y_1 + \boldsymbol{u_{22}y_2} + u_{23}y_3 + c_2)u_{22}$$

$$\frac{\partial L}{\partial y_1} = \frac{\partial L}{\partial z_1}\varphi'(u_{11}y_1 + u_{12}y_2 + \boldsymbol{u_{13}y_3} + c_1)u_{13} + \frac{\partial L}{\partial z_2}\varphi'(u_{21}y_1 + u_{22}y_2 + \boldsymbol{u_{23}y_3} + c_2)u_{23}$$

$\frac{\partial L}{\partial z_1}$ and $\frac{\partial L}{\partial z_2}$ are from its upper layer.

# Weight updates

$$u_{ij}^{new} = u_{ij}^{old} - \mu \frac{\partial L}{\partial u_{ij}}$$

$$\frac{\partial L}{\partial u_{11}} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial u_{11}} = \frac{\partial L}{\partial z_1}\varphi'(\boldsymbol{u_{11}y_1} + u_{12}y_2 + u_{13}y_3 + c_1)y_1$$

$$\frac{\partial L}{\partial u_{12}} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial u_{12}} = \frac{\partial L}{\partial z_1}\varphi'(u_{11}y_1 + \boldsymbol{u_{12}y_2} + u_{13}y_3 + c_1)y_2$$

$$\frac{\partial L}{\partial u_{13}} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial u_{13}} = \frac{\partial L}{\partial z_1}\varphi'(u_{11}y_1 + u_{12}y_2 + \boldsymbol{u_{13}y_3} + c_1)y_3$$

$$\frac{\partial L}{\partial c_1} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial c_1} = \frac{\partial L}{\partial z_1}\varphi'(u_{11}y_1 + u_{12}y_2 + u_{13}y_3 + \boldsymbol{c_1})$$

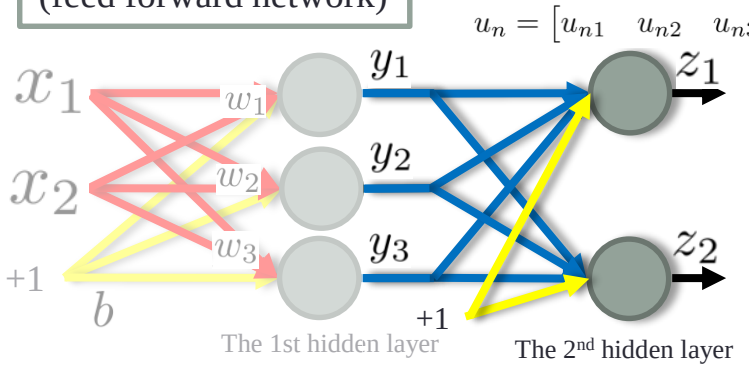
$$\frac{\partial L}{\partial u_{21}} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial u_{21}} = \frac{\partial L}{\partial z_2}\varphi'(\boldsymbol{u_{21}y_1} + u_{22}y_2 + u_{23}y_3 + c_2)y_1$$

$$\frac{\partial L}{\partial u_{22}} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial u_{22}} = \frac{\partial L}{\partial z_2}\varphi'(u_{21}y_1 + \boldsymbol{u_{22}y_2} + u_{23}y_3 + c_2)y_2$$

$$\frac{\partial L}{\partial u_{23}} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial u_{23}} = \frac{\partial L}{\partial z_2}\varphi'(u_{21}y_1 + u_{22}y_2 + \boldsymbol{u_{23}y_3} + c_2)y_3$$

$$\frac{\partial L}{\partial c_2} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial c_2} = \frac{\partial L}{\partial z_2}\varphi'(u_{21}y_1 + u_{22}y_2 + u_{23}y_3 + \boldsymbol{c_2})$$

(feed forward network)

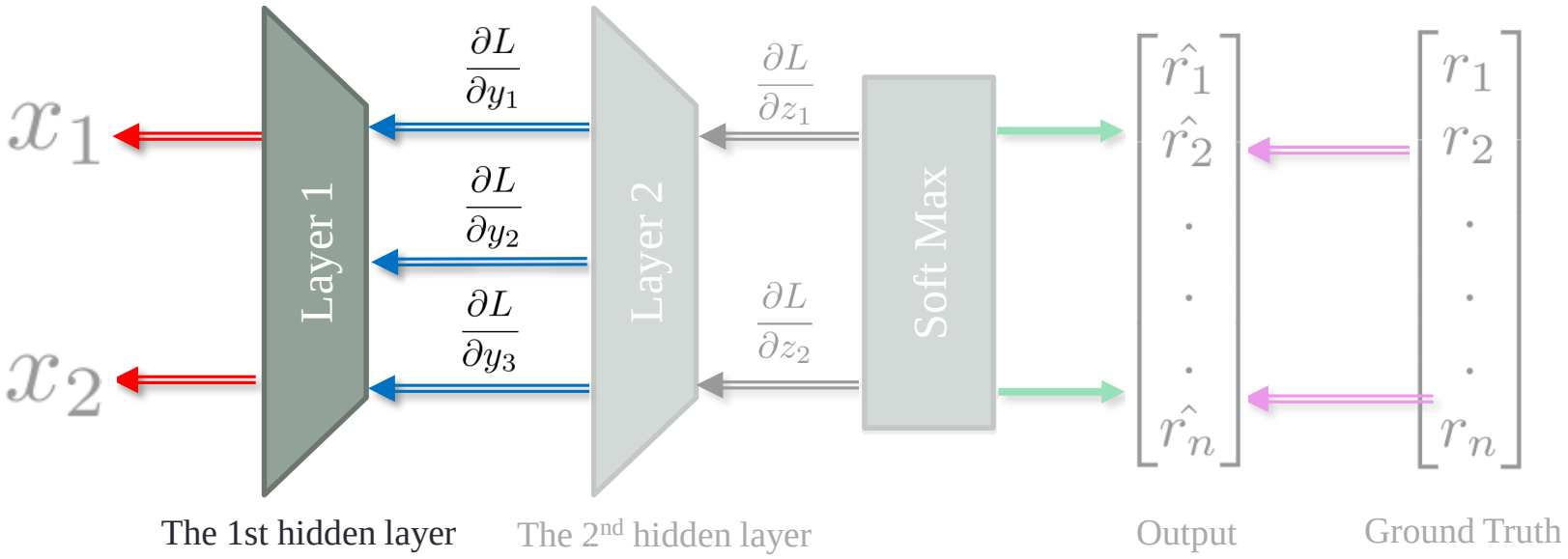



$$z_1 = \varphi(u_{11}y_1 + u_{12}y_2 + u_{13}y_3 + c_1)$$

$$z_2 = \varphi(u_{21}y_1 + u_{22}y_2 + u_{23}y_3 + c_2)$$

$\frac{\partial L}{\partial z_1}$ and $\frac{\partial L}{\partial z_2}$ are from its upper layer.

# Back propagation; 1st layer




- the Layer 1 has to do
  - Weight update

$$w_{ij}^{new} = w_{ij}^{old} - \mu \frac{\partial L}{\partial w_{ij}}$$

  - Error propagation, Input update ???

# Weight updates

$$w_{ij}^{new} = w_{ij}^{old} - \mu \frac{\partial L}{\partial w_{ij}}$$

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial w_{11}} = \frac{\partial L}{\partial y_1}\varphi'(\boldsymbol{w_{11}x_1} + w_{12}x_2 + b_1)x_1$$

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial w_{12}} = \frac{\partial L}{\partial y_1}\varphi'(w_{11}x_1 + \boldsymbol{w_{12}x_2} + b_1)x_2$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial b_1} = \frac{\partial L}{\partial y_1}\varphi'(w_{11}x_1 + w_{12}x_2 + \boldsymbol{b_1})$$

$$\frac{\partial L}{\partial w_{21}} = \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial w_{21}} = \frac{\partial L}{\partial y_2}\varphi'(\boldsymbol{w_{21}x_1} + w_{22}x_2 + b_2)x_1$$

$$\frac{\partial L}{\partial w_{22}} = \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial w_{22}} = \frac{\partial L}{\partial y_2}\varphi'(w_{21}x_1 + \boldsymbol{w_{22}x_2} + b_2)x_2$$

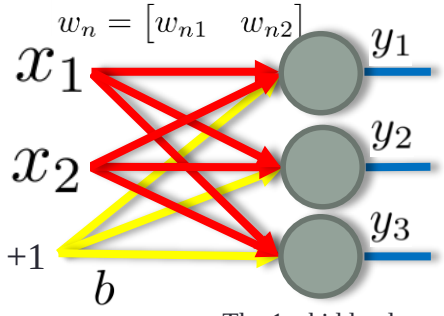
$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial b_2} = \frac{\partial L}{\partial y_2}\varphi'(w_{21}x_1 + w_{22}x_2 + \boldsymbol{b_2})$$

$$\frac{\partial L}{\partial w_{31}} = \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial w_{31}} = \frac{\partial L}{\partial y_3}\varphi'(\boldsymbol{w_{31}x_1} + w_{32}x_2 + b_3)x_1$$

$$\frac{\partial L}{\partial w_{32}} = \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial w_{32}} = \frac{\partial L}{\partial y_3}\varphi'(w_{31}x_1 + \boldsymbol{w_{32}x_2} + b_3)x_2$$

$$\frac{\partial L}{\partial b_3} = \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial b_3} = \frac{\partial L}{\partial y_3}\varphi'(w_{31}x_1 + w_{32}x_2 + \boldsymbol{b_3})$$

(feed forward network)




The 1st hidden layer

$$y_1 = \varphi(w_{11}x_1 + w_{12}x_2 + b_1)$$
$$y_2 = \varphi(w_{21}x_1 + w_{22}x_2 + b_2)$$
$$y_3 = \varphi(w_{31}x_1 + w_{32}x_2 + b_3)$$

$\frac{\partial L}{\partial y_1}$, $\frac{\partial L}{\partial y_2}$ and $\frac{\partial L}{\partial y_3}$ are from its upper layer

# Block-based perspective

# Basic Math

$$\mathbf{Y} = \boldsymbol{X}^{\top}\boldsymbol{A}\boldsymbol{X}$$

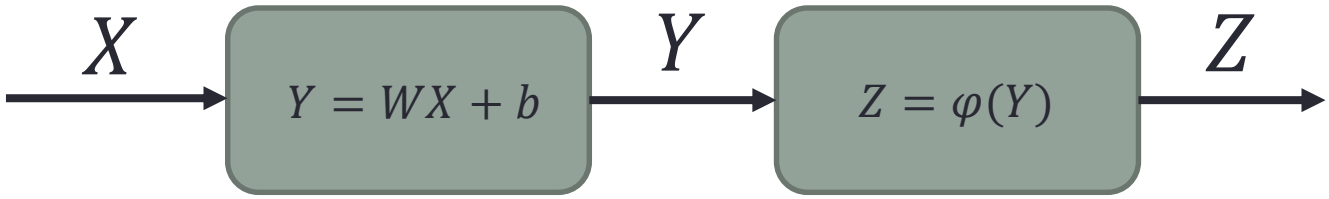
$$\begin{aligned} Y+\Delta Y &\simeq (X+\Delta X)^{\top}A(X+\Delta X) \\ &= X^{\top}AX + \Delta X^{\top}AX + X^{\top}\Delta X + \Delta X^{\top}A\Delta X \\ &\approx X^{\top}AX + \underbrace{X^{\top}(A+A^{\top})}_{=\frac{\partial Y}{\partial X}}\Delta X \end{aligned}$$

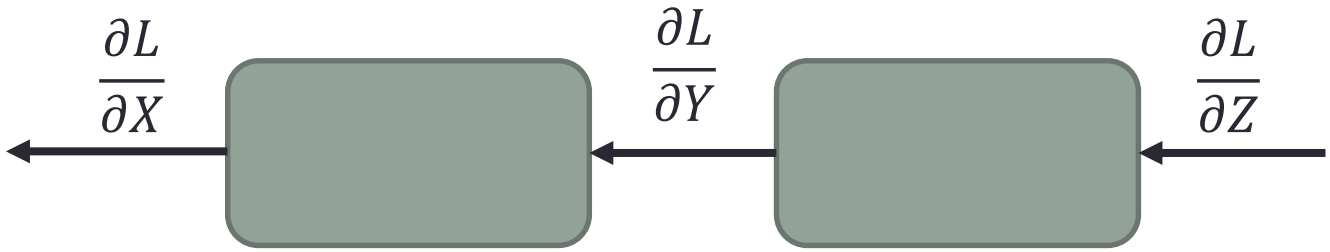
$$\frac{\partial Y}{\partial A} \Rightarrow \quad Y+\Delta Y \approx X^{\top}(A+\Delta A)X = X^{\top}AX + X^{\top}\Delta AX$$

$$\begin{aligned} \Delta Y &= X^{\top}\Delta AX \\ &= tr(X^{\top}\Delta AX) \\ &= tr(XX^{\top}\Delta A) \end{aligned}$$

$$\therefore \frac{\partial Y}{\partial A} = XX^{\top}$$

# Block-based representation



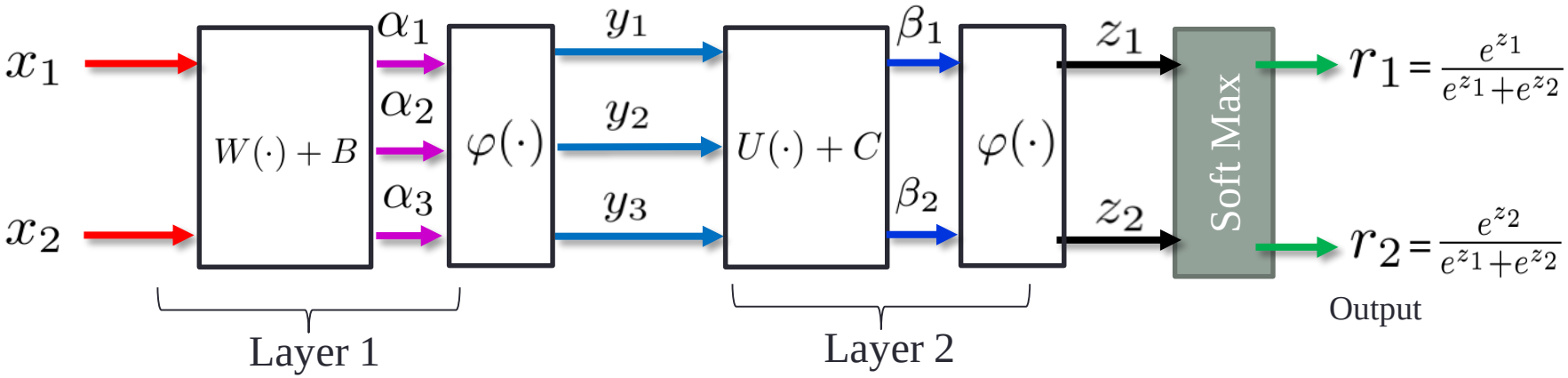




$$\left(\frac{\partial L}{\partial X}\right)^{\top} = W^{\top}\left(\frac{\partial L}{\partial Y}\right)^{\top} \qquad \left(\frac{\partial L}{\partial Y}\right)^{\top} = \operatorname{diag}\left(\varphi'(Z)\right)\left(\frac{\partial L}{\partial Z}\right)^{\top}$$

$$\left(\frac{\partial L}{\partial W}\right)^{\top} = \left(\frac{\partial L}{\partial Y}\right)^{\top} X^{\top}$$

$$\left(\frac{\partial L}{\partial b}\right)^{\top} = \left(\frac{\partial L}{\partial Y}\right)^{\top}$$

# Forward propagation (block-based representation)




Layer 1

$$\begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} = \begin{bmatrix}w_{11} & w_{12}\\ w_{21} & w_{22}\\ w_{31} & w_{32}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b_1\\ b_2\\ b_3\end{bmatrix}$$

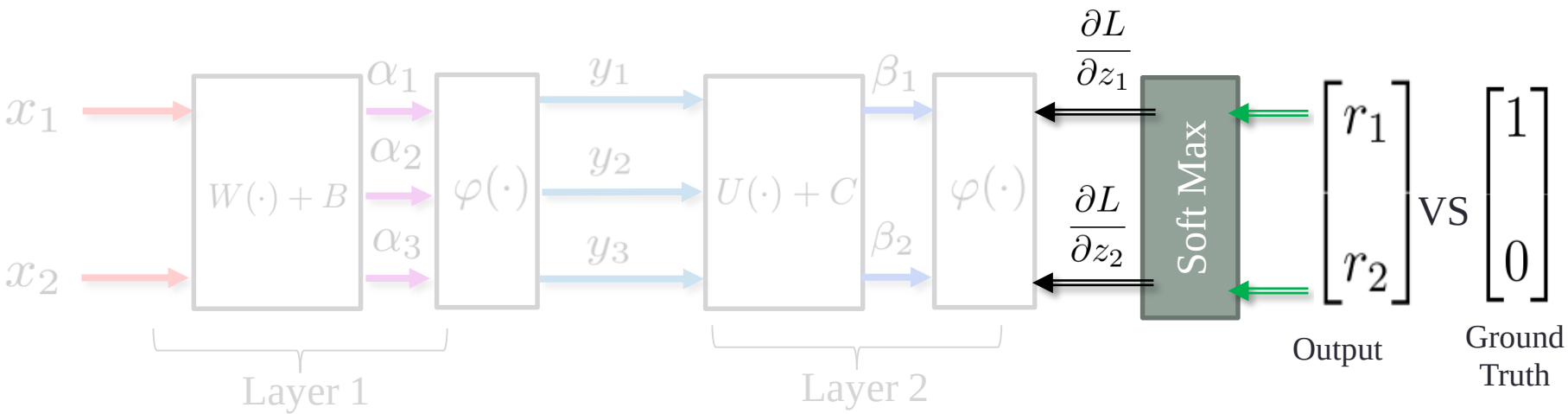
$$\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix} = \begin{bmatrix}\varphi(\alpha_1)\\ \varphi(\alpha_2)\\ \varphi(\alpha_3)\end{bmatrix}$$

Layer 2

$$\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix} = \begin{bmatrix}u_{11} & u_{12} & u_{13}\\ u_{21} & u_{22} & u_{23}\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix} + \begin{bmatrix}c_1\\ c_2\\ c_3\end{bmatrix}$$

$$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}\varphi(\beta_1)\\ \varphi(\beta_2)\end{bmatrix}$$

# Backward propagation; 2nd layer

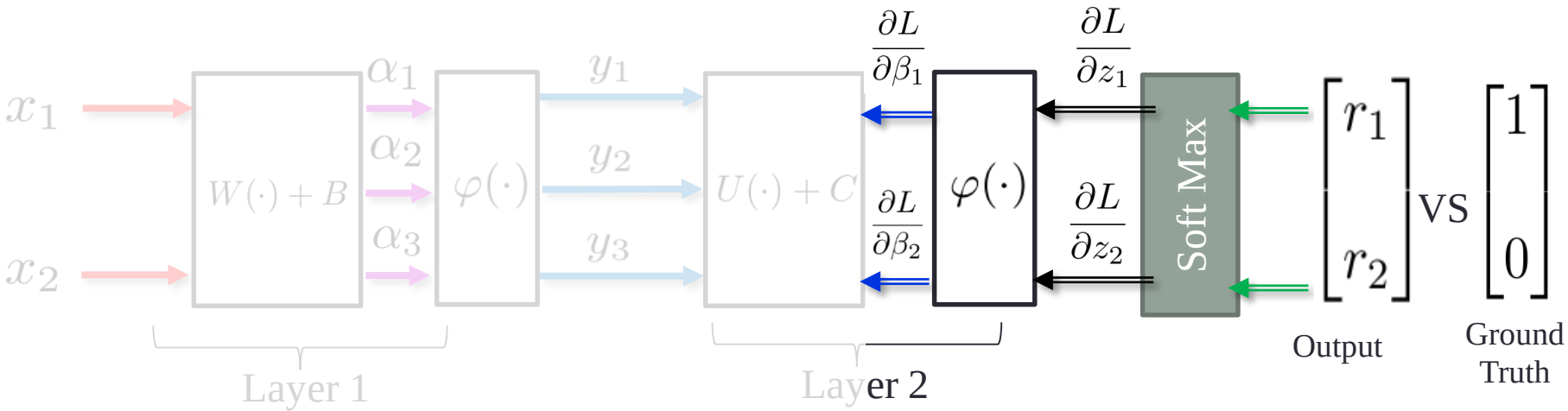



- Error propagation

$$\frac{\partial L}{\partial z_1} = -1 + r_1$$

$$\frac{\partial L}{\partial z_2} = r_2$$

# Backward propagation; 2nd layer

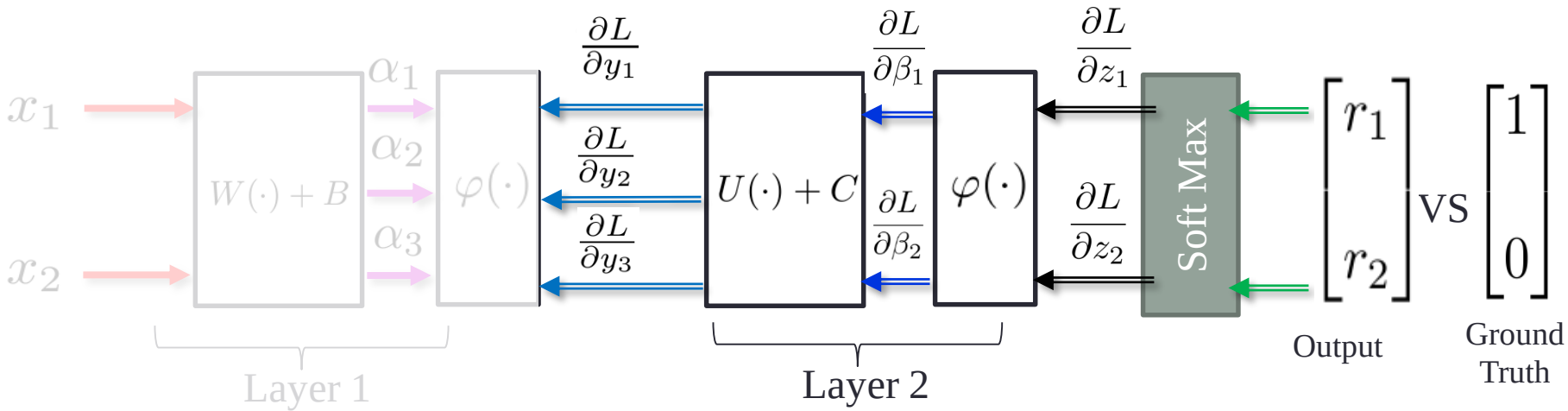



- Error propagation

$$\frac{\partial L}{\partial \beta_1} = \varphi'(\beta_1)\frac{\partial L}{\partial z_1}$$

$$\frac{\partial L}{\partial \beta_2} = \varphi'(\beta_2)\frac{\partial L}{\partial z_2}$$

# Backward propagation; 2nd layer




- Error propagation

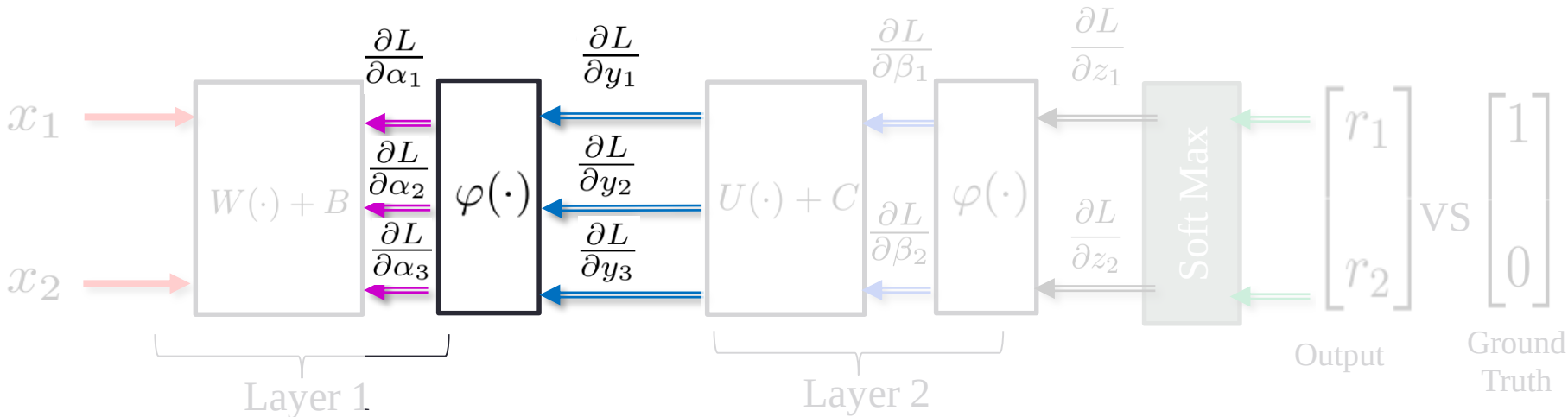
$$\begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \frac{\partial L}{\partial y_3} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{21} \\ u_{12} & u_{22} \\ u_{13} & u_{23} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial \beta_1} \\ \frac{\partial L}{\partial \beta_2} \end{bmatrix}$$

- Weight update

$$\begin{bmatrix} \frac{\partial L}{\partial u_{11}} & \frac{\partial L}{\partial u_{12}} & \frac{\partial L}{\partial u_{13}} \\ \frac{\partial L}{\partial u_{21}} & \frac{\partial L}{\partial u_{22}} & \frac{\partial L}{\partial u_{23}} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial \beta_1} \\ \frac{\partial L}{\partial \beta_2} \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial L}{\partial c_1} \\ \frac{\partial L}{\partial c_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial \beta_1} \\ \frac{\partial L}{\partial \beta_2} \end{bmatrix}$$

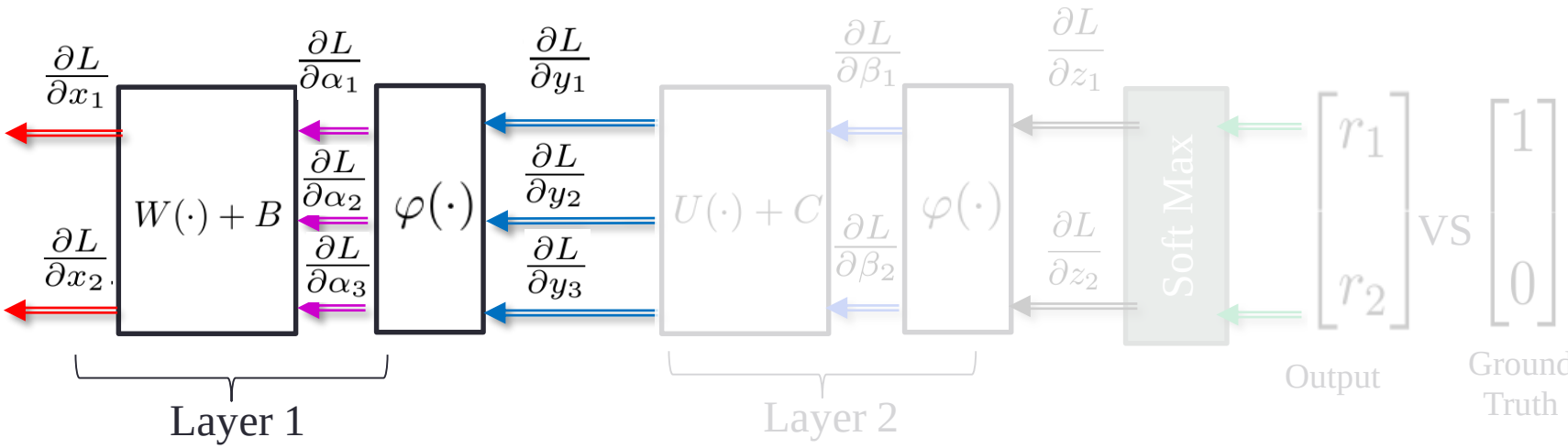
# Backward propagation; 1st layer




- Error propagation

$$\begin{bmatrix} \frac{\partial L}{\partial \alpha_1} \\ \frac{\partial L}{\partial \alpha_2} \\ \frac{\partial L}{\partial \alpha_3} \end{bmatrix} = \begin{bmatrix} \varphi'(\alpha_1) & 0 & 0 \\ 0 & \varphi'(\alpha_2) & 0 \\ 0 & 0 & \varphi'(\alpha_3) \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \frac{\partial L}{\partial y_3} \end{bmatrix}$$

# Backward propagation; 1st layer




- Error propagation

$$\begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial \alpha_1} \\ \frac{\partial L}{\partial \alpha_2} \\ \frac{\partial L}{\partial \alpha_3} \end{bmatrix}$$

- Weight update

$$\begin{bmatrix} \frac{\partial L}{\partial w_{11}} & \frac{\partial L}{\partial w_{12}} \\ \frac{\partial L}{\partial w_{21}} & \frac{\partial L}{\partial w_{22}} \\ \frac{\partial L}{\partial w_{31}} & \frac{\partial L}{\partial w_{32}} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial \alpha_1} \\ \frac{\partial L}{\partial \alpha_2} \\ \frac{\partial L}{\partial \alpha_3} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

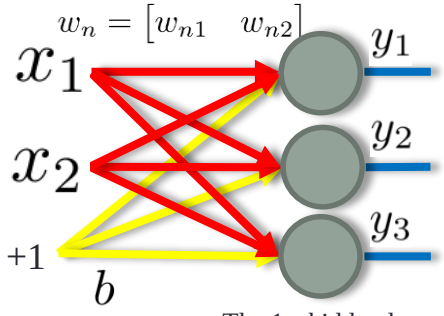
$$\begin{bmatrix} \frac{\partial L}{\partial b_1} \\ \frac{\partial L}{\partial b_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial \beta_1} \\ \frac{\partial L}{\partial \beta_2} \end{bmatrix}$$

# Input Optimization while fixing all weights

# Input update

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_1} + \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial x_1}$$

(feed forward network)



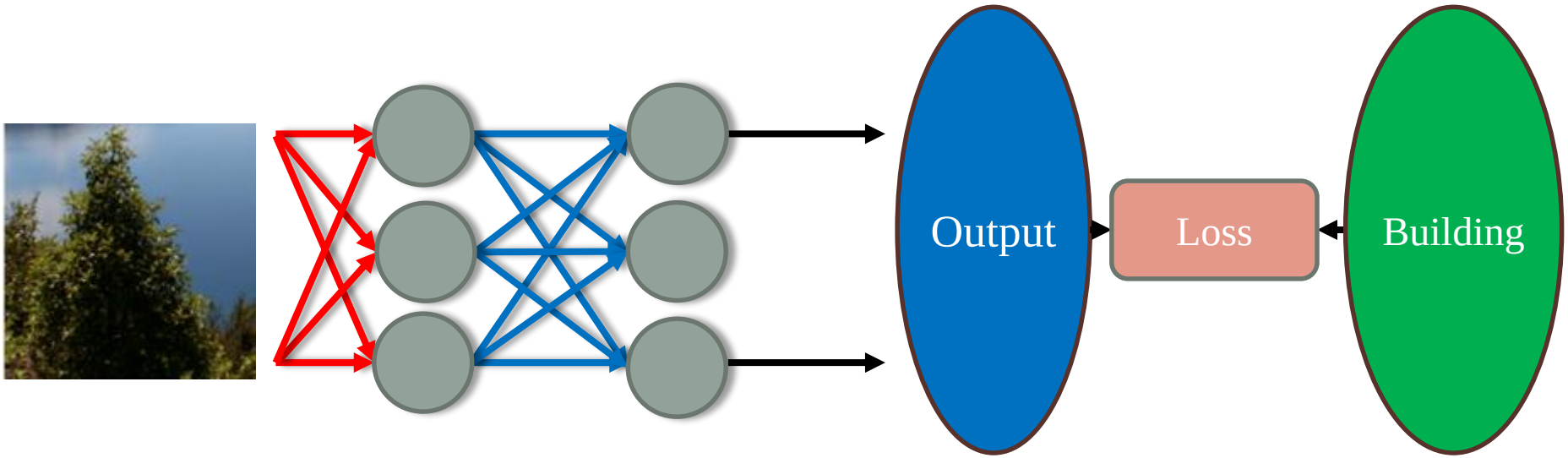

The 1st hidden layer

$$y_1 = \varphi(w_{11}x_1 + w_{12}x_2 + b_1)$$
$$y_2 = \varphi(w_{21}x_1 + w_{22}x_2 + b_2)$$
$$y_3 = \varphi(w_{31}x_1 + w_{32}x_2 + b_3)$$

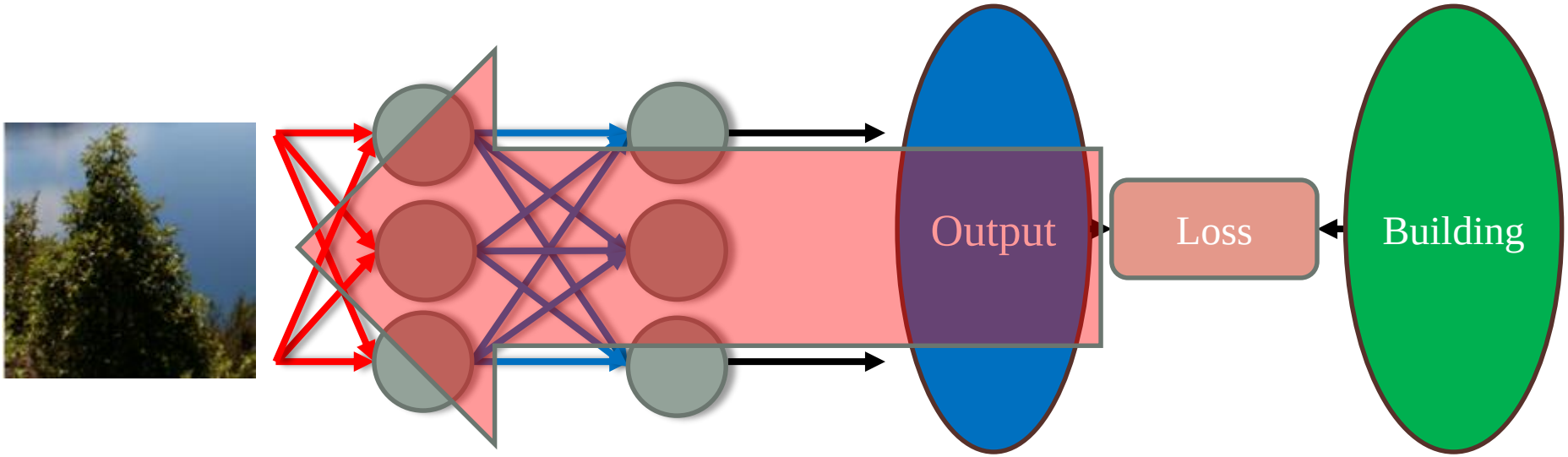
$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_1} + \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial x_1} = \frac{\partial L}{\partial y_1}\varphi'(\boldsymbol{w_{11}x_1} + w_{12}x_2 + b_1)w_{11} + \frac{\partial L}{\partial y_2}\varphi'(\boldsymbol{w_{21}x_1} + w_{22}x_2 + b_1)w_{21} + \frac{\partial L}{\partial y_3}\varphi'(\boldsymbol{w_{31}x_1} + w_{32}x_2 + b_1)w_{31}$$

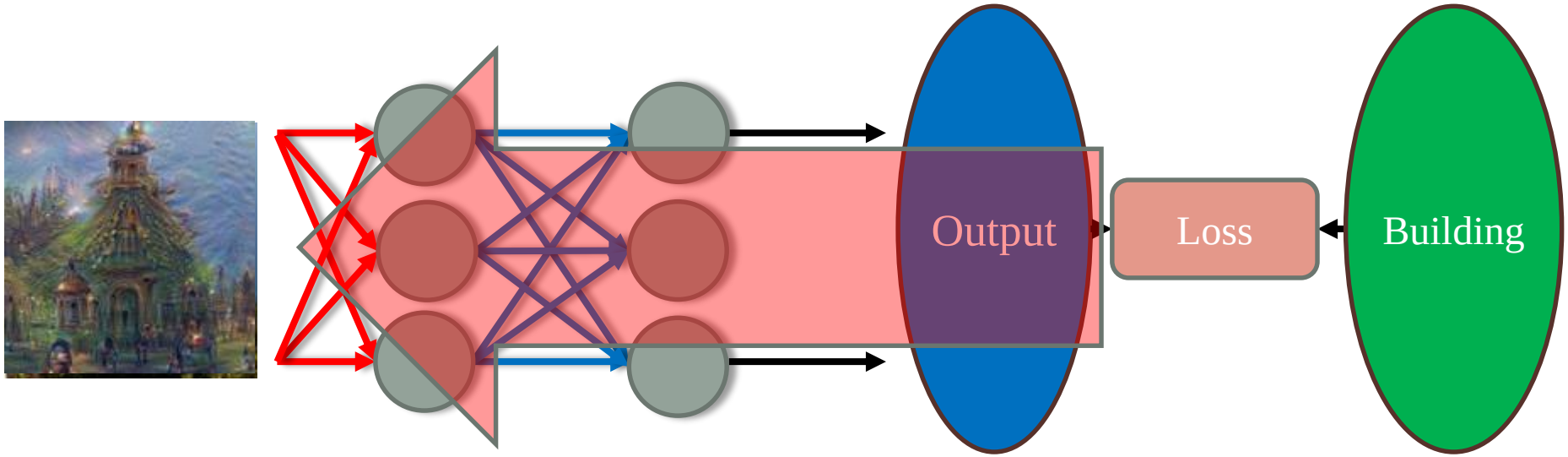
$$\frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_2} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_2} + \frac{\partial L}{\partial y_3}\frac{\partial y_3}{\partial x_2} = \frac{\partial L}{\partial y_1}\varphi'(w_{11}x_1 + \boldsymbol{w_{12}x_2} + b_1)w_{12} + \frac{\partial L}{\partial y_2}\varphi'(w_{21}x_1 + \boldsymbol{w_{22}x_2} + b_1)w_{22} + \frac{\partial L}{\partial y_3}\varphi'(w_{31}x_1 + \boldsymbol{w_{32}x_2} + b_1)w_{32}$$

$\frac{\partial L}{\partial y_1}$, $\frac{\partial L}{\partial y_2}$ and $\frac{\partial L}{\partial 3}$ are from its upper layer.


Output
Loss
Building

$$I^{new} = I^{old} - \mu \frac{\partial L}{\partial I}$$




Output
Loss
Building

# Inceptionism: Going Deeper into Neural Networks


https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

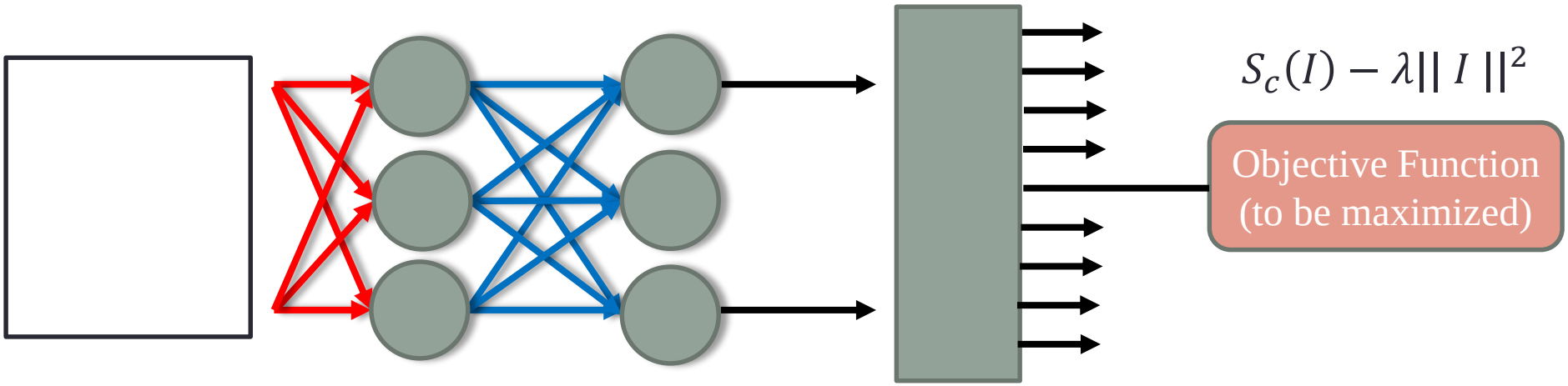
# Inceptionism: Going Deeper into Neural Networks

*Neural net "dreams"— generated purely from random noise, using a network trained on places by MIT Computer Science and AI Laboratory. See our Inceptionism gallery for hi-res versions of the images above and more (Images marked "Places205-GoogLeNet" were made using this network).*

# DEEP INSIDE CONVOLUTION NETWORKS

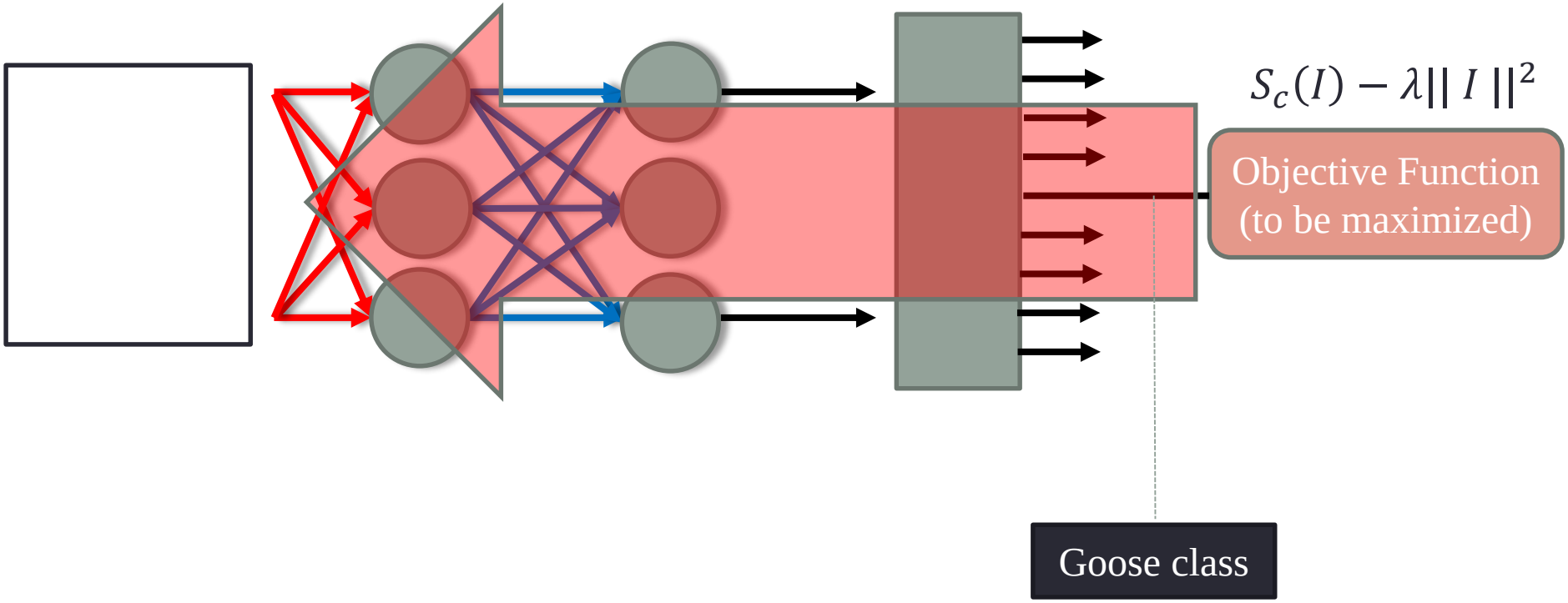
Inputs maximizing class score

# Inputs maximizing class score

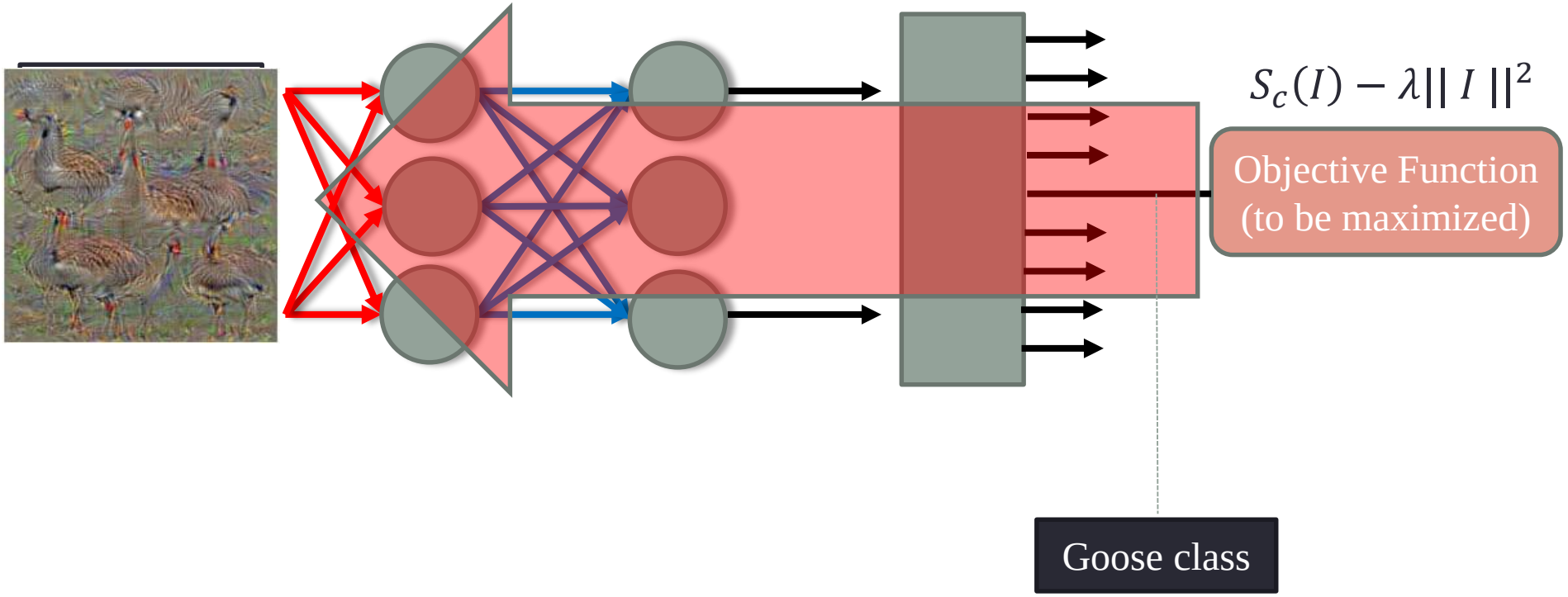



K. Simonyan, A. Vedaldi, A. Zisserman, "**Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps",** *ICLR Workshop 2014*

# Inputs maximizing class score



# Maximizing class score



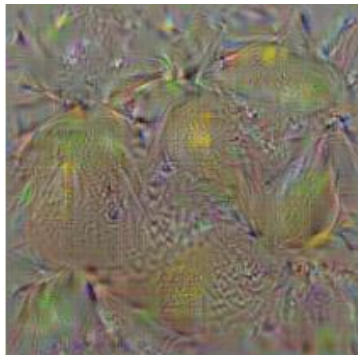
# Inputs maximizing class score

dumbbell

cup

dalmatian

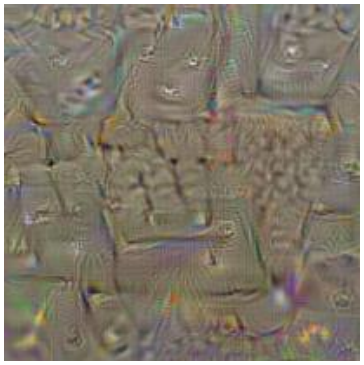
bell pepper



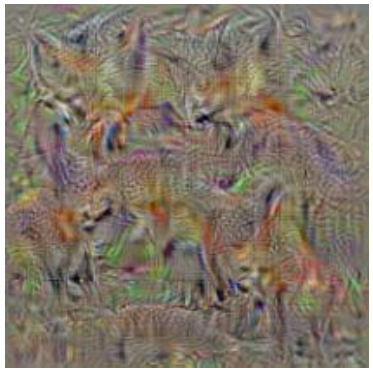
lemon

husky

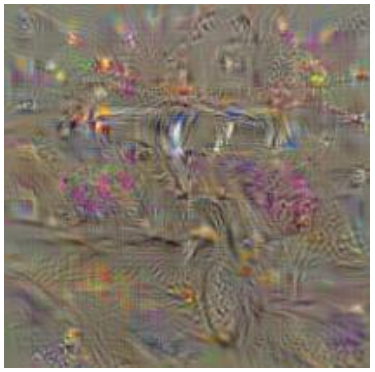
# Inputs maximizing class score



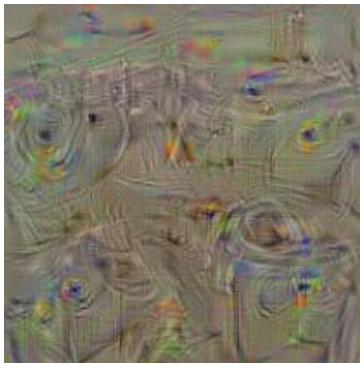
computer keyboard



kit fox



limousine



Washing machine

goose

ostrich

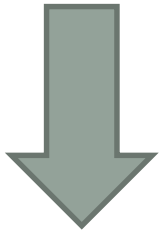
# DEEP INSIDE CONVOLUTION NETWORKS

Saliency visualization

# Saliency visualization

- Linear score model for class c:

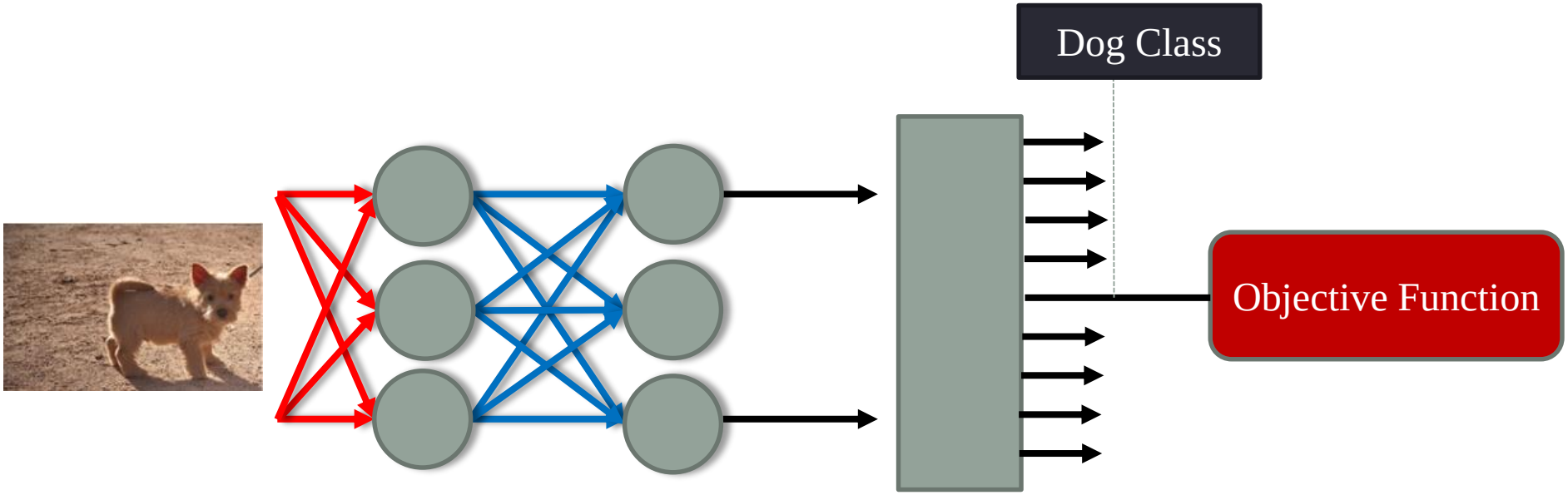
$$S_c(I) \approx wI + b$$



$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

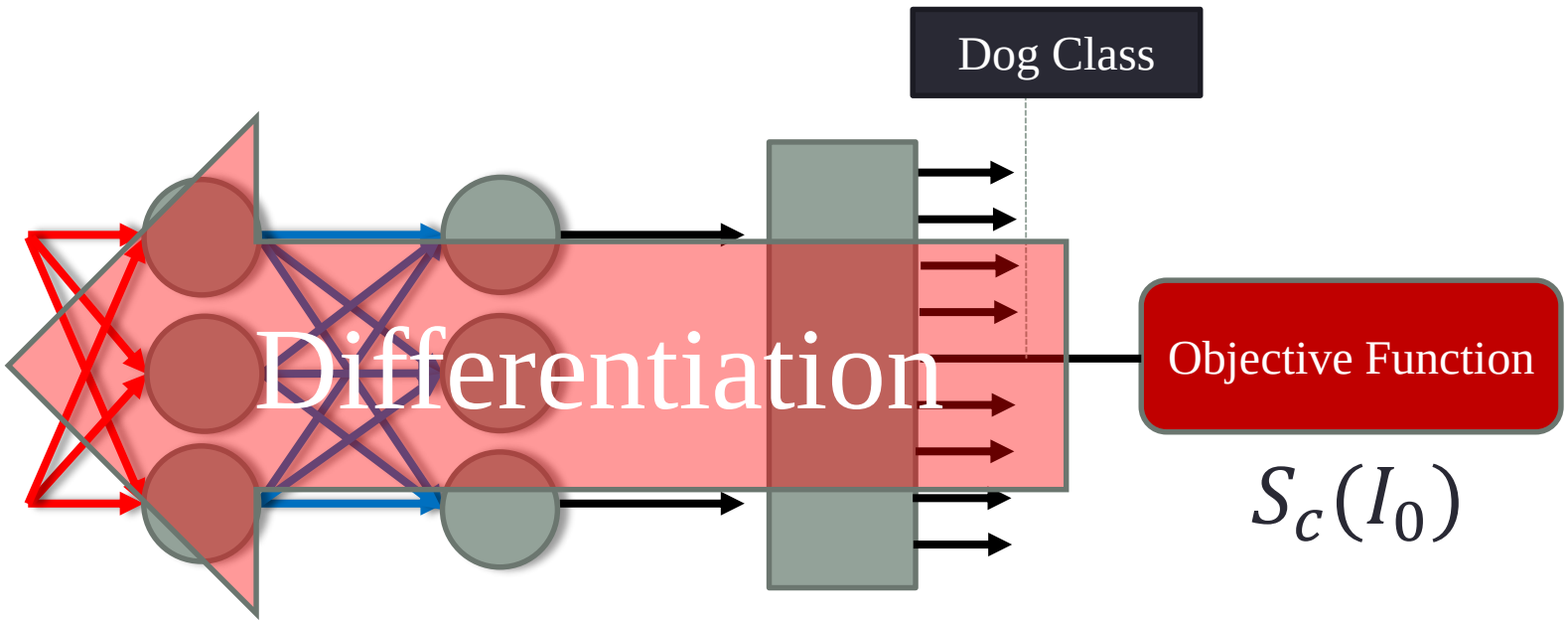
$w$ : importance of corresponding pixels of $I$ for class $c$

# Saliency visualization



# Saliency visualization

$$w = \frac{\partial S_c}{\partial I}\Big|_{I_0}$$

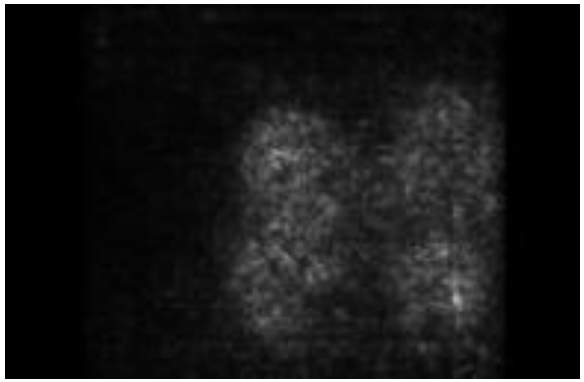



$I_0 =$ 

# Saliency visualization

$$I_0 =$$

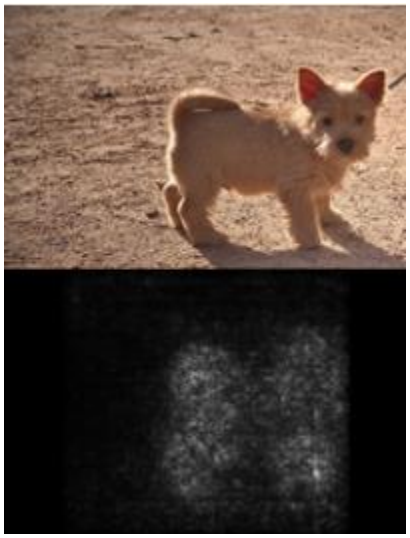
$$w = \left. \frac{\partial S_c}{\partial I} \right|_{I_0}$$



saliency map

yacht

dog



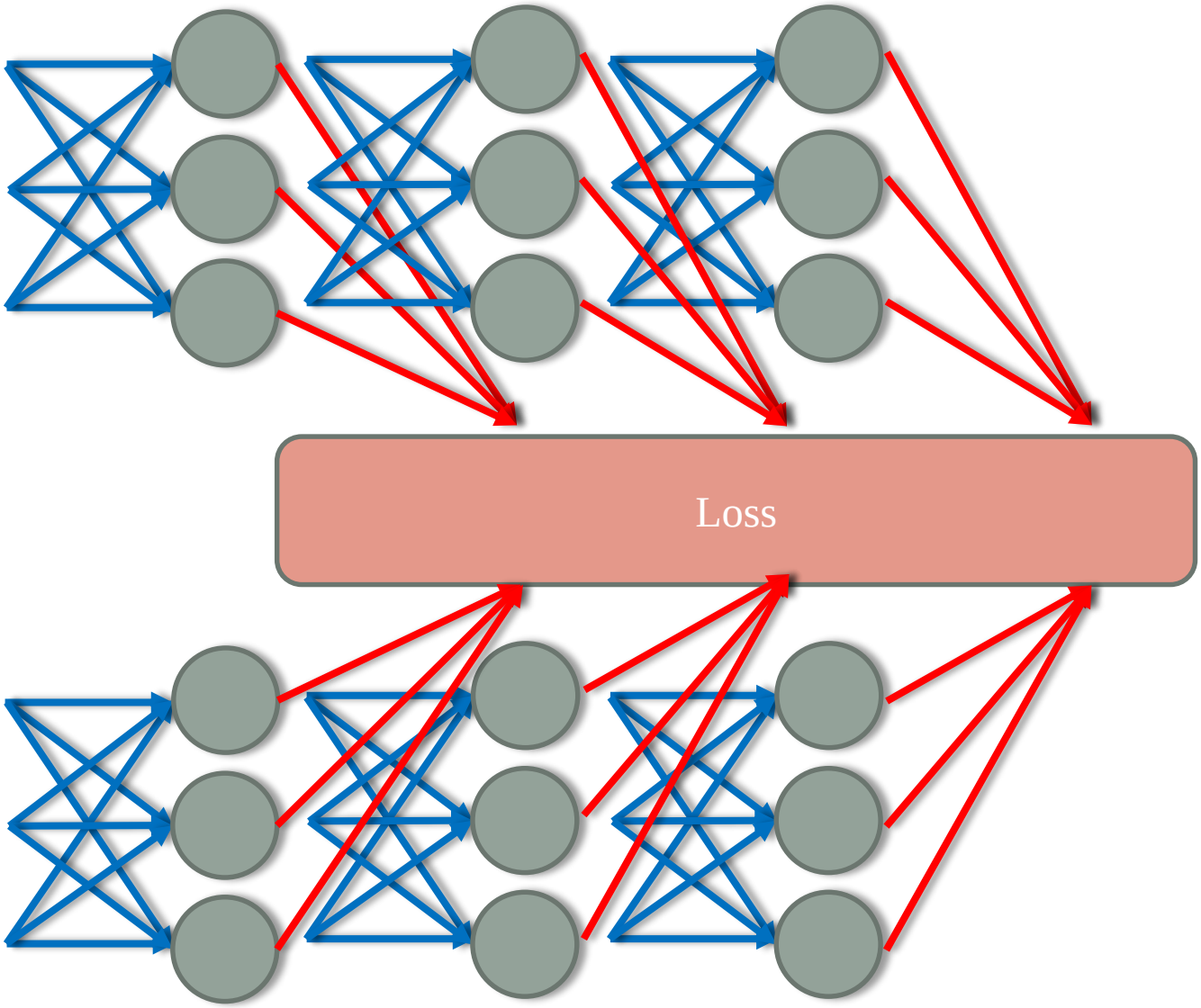
monkey

washing machine

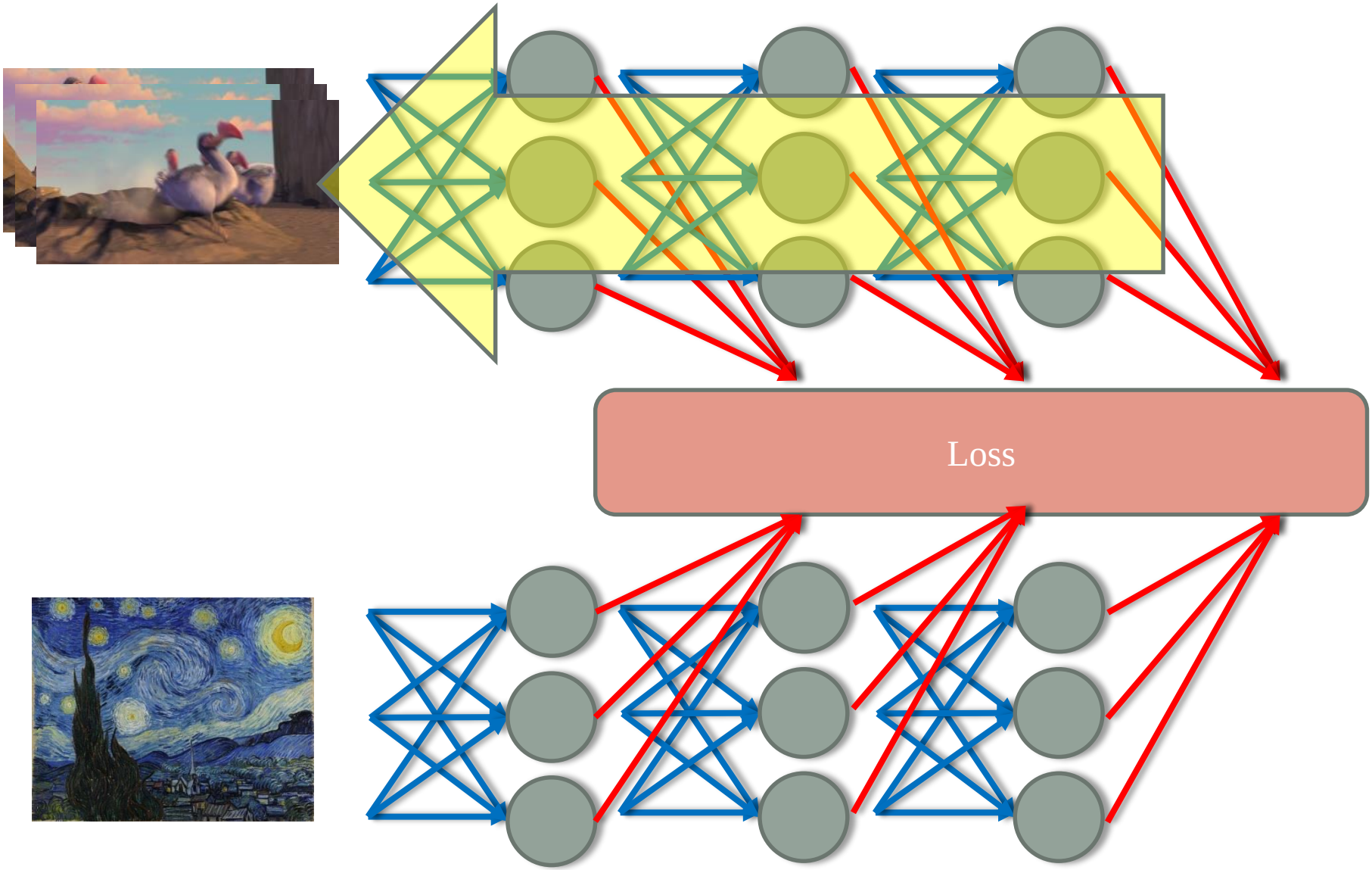
cow

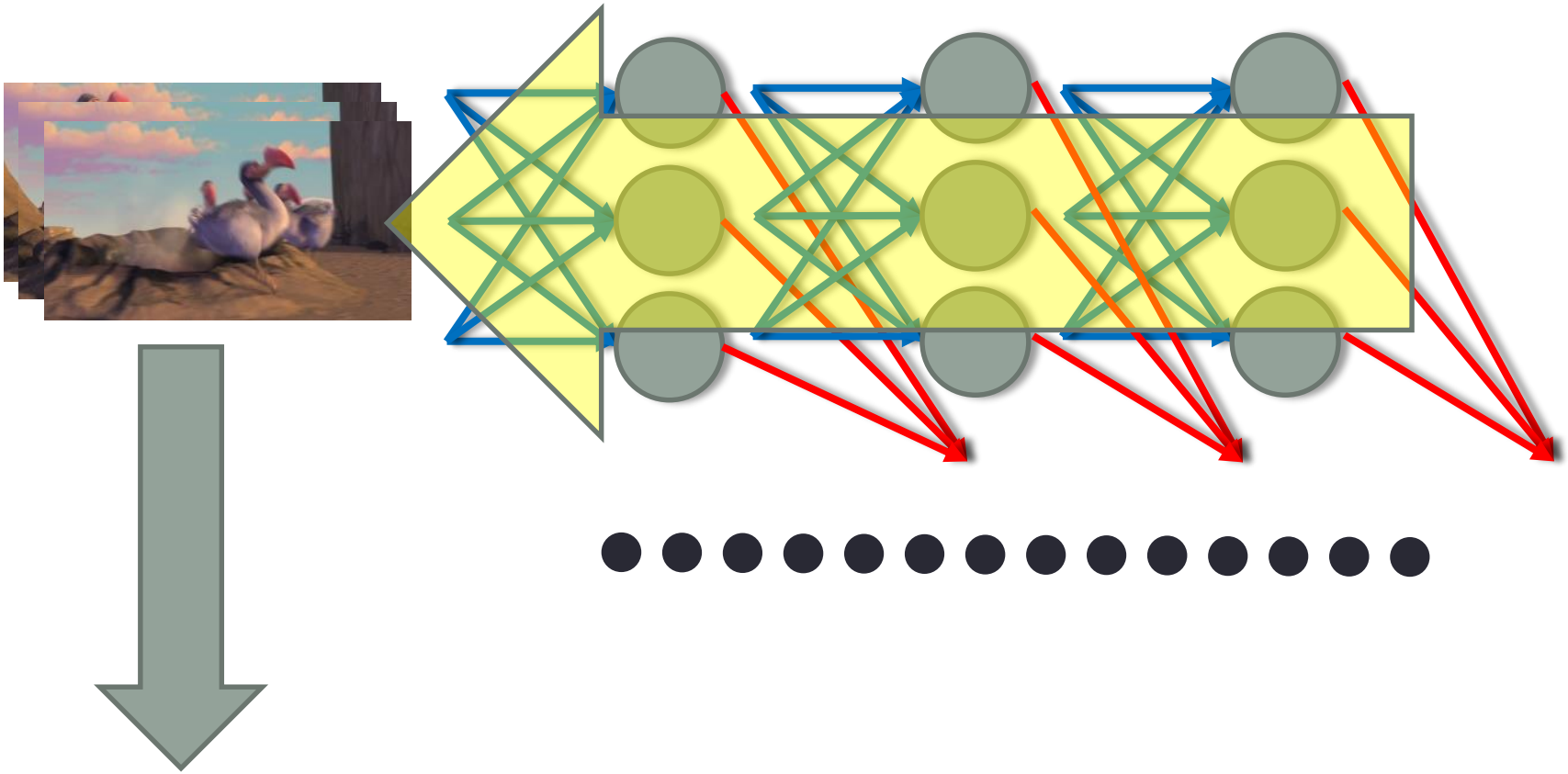
building

# A NEURAL ALGORITHM OF ARTISTIC STYLE

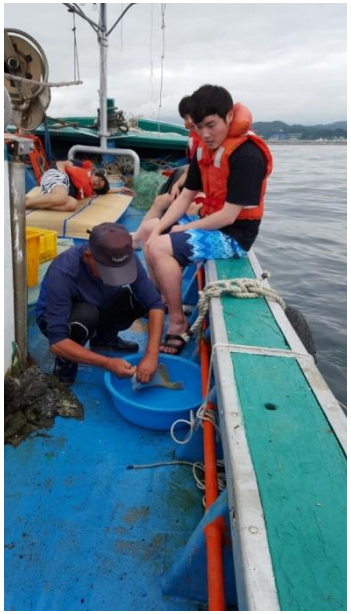


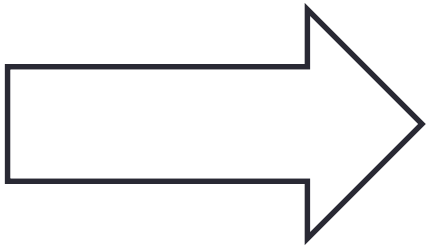

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. **"A neural algorithm of artistic style."** *arXiv preprint arXiv:1508.06576* (2015).


Loss

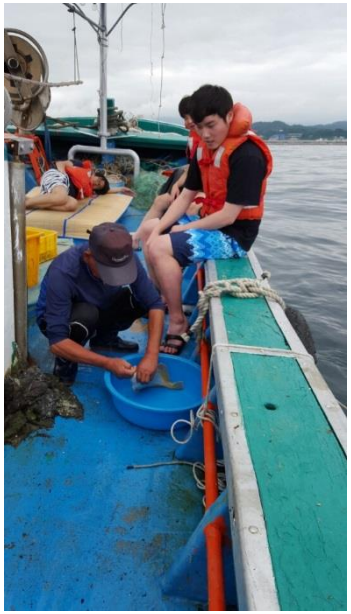


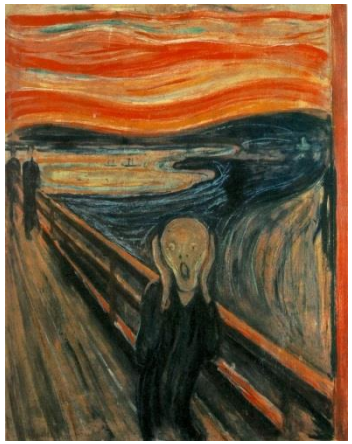
# Artistic style

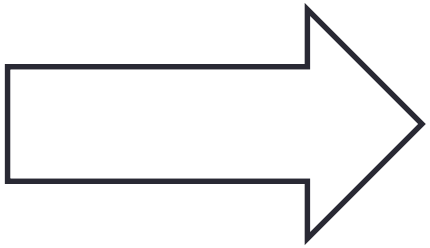




# Artistic style







# Backups

# Vector-by-scalar

**Vector-by-scalar** [ edit ]

The derivative of a vector $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$, by a scalar $x$ is written (in numerator layout notation) as

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_m}{\partial x} \end{bmatrix}.$$

In vector calculus the derivative of a vector **y** with respect to a scalar $x$ is known as the **tangent vector** of the vector **y**, $\frac{\partial \mathbf{y}}{\partial x}$. Notice here that $\mathbf{y}:\mathbf{R}^1 \rightarrow \mathbf{R}^m$.

# Scalar-by-vector

The derivative of a scalar $y$ by a vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, is written (in numerator layout notation) as

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix}.$$

In vector calculus, the gradient of a scalar field $y$ in the space $\mathbf{R}^n$ (whose independent coordinates are the components of $\mathbf{x}$) is the derivative of a scalar by a vector. In physics, the electric field is the vector gradient of the electric potential.

The directional derivative of a scalar function $f(\mathbf{x})$ of the space vector $\mathbf{x}$ in the direction of the unit vector $\mathbf{u}$ is defined using the gradient as follows.

$$\nabla_{\mathbf{u}} f(\mathbf{x}) = \nabla f(\mathbf{x}) \cdot \mathbf{u}$$

Using the notation just defined for the derivative of a scalar with respect to a vector we can re-write the directional derivative as $\nabla_{\mathbf{u}} f = \frac{\partial f}{\partial \mathbf{x}} \mathbf{u}$. This type of notation will be nice when proving product rules and chain rules that come out looking similar to what we are familiar with for the scalar derivative.

# Vector-by-vector

### Vector-by-vector [ edit ]

Each of the previous two cases can be considered as an application of the derivative of a vector with respect to a vector, using a vector of size one appropriately. Similarly we will find that the derivatives involving matrices will reduce to derivatives involving vectors in a corresponding way.

The derivative of a vector function (a vector whose components are functions) $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$, with respect to an input vector, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, is written (in numerator layout notation) as

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}.$$

In vector calculus, the derivative of a vector function **y** with respect to a vector **x** whose components represent a space is known as the **pushforward (or differential)**, or the **Jacobian matrix**.

# Scalar-by-matrix

**Scalar-by-matrix** [ edit ]

The derivative of a scalar $y$ function of a $p \times q$ matrix $\mathbf{X}$ of independent variables, with respect to the matrix $\mathbf{X}$, is given (in numerator layout notation) by

$$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{21}} & \cdots & \frac{\partial y}{\partial x_{p1}} \\ \frac{\partial y}{\partial x_{12}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{p2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1q}} & \frac{\partial y}{\partial x_{2q}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{bmatrix}.$$

Important examples of scalar functions of matrices include the trace of a matrix and the determinant.

In analog with vector calculus this derivative is often written as the following.

$$\nabla_{\mathbf{X}} y(\mathbf{X}) = \frac{\partial y(\mathbf{X})}{\partial \mathbf{X}}$$

Also in analog with vector calculus, the **directional derivative** of a scalar $f(\mathbf{X})$ of a matrix $\mathbf{X}$ in the direction of matrix $\mathbf{Y}$ is given by

$$\nabla_{\mathbf{Y}} f = \operatorname{tr}\left(\frac{\partial f}{\partial \mathbf{X}} \mathbf{Y}\right).$$

# Why $\frac{\partial L}{\partial W} = X\frac{\partial L}{\partial Y}$?

- We want to find $\frac{\partial L}{\partial W}$ satisfying:
  - $\Delta L = \text{tr}\left(\frac{\partial L}{\partial W}\,\Delta W\right)$.
- from
  - $Y = WX$
  - $\Delta L = \frac{\partial L}{\partial Y}\Delta Y$.
- [Intuitive derivation]
  - $\Delta Y = \Delta W X$
  - $\Delta L = \frac{\partial L}{\partial Y}\Delta Y = \frac{\partial L}{\partial Y}\Delta W X = \text{tr}\left(\frac{\partial L}{\partial Y}\Delta W X\right) = \text{tr}\left(X\frac{\partial L}{\partial Y}\Delta W\right)$
    - $\frac{\partial L}{\partial W} = X\frac{\partial L}{\partial Y}$

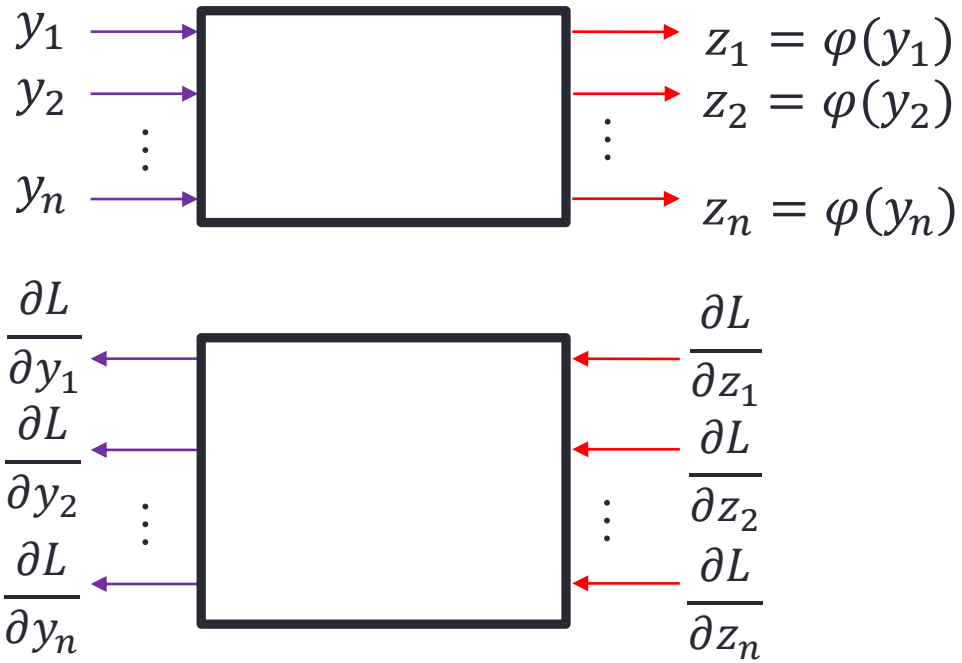

$Y = WX$ $\quad Z = Y + b$ $\quad V = \phi(Z)$


$$\left(\frac{\partial L}{\partial X}\right)^{\top} = W^{\top}\left(\frac{\partial L}{\partial Y}\right)^{\top} \qquad \left(\frac{\partial L}{\partial Y}\right)^{\top} = \left(\frac{\partial L}{\partial Z}\right)^{\top} \qquad \left(\frac{\partial L}{\partial Z}\right)^{\top} = \mathrm{diag}\left(\phi'(Z)\right)\left(\frac{\partial L}{\partial V}\right)^{\top}$$

$$\left(\frac{\partial L}{\partial W}\right)^{\top} = \left(\frac{\partial L}{\partial Y}\right)^{\top} X^{\top} \qquad \left(\frac{\partial L}{\partial b}\right)^{\top} = \left(\frac{\partial L}{\partial Z}\right)^{\top}$$

# Proof

- Element-wise operation




$$\frac{\partial L}{\partial y_i} = \frac{\partial L}{\partial z_i}\varphi'(z_i)$$

$$\left(\frac{\partial L}{\partial Y}\right)^{\mathsf{T}} = \mathrm{diag}\left(\varphi'(Z)\right)\left(\frac{\partial L}{\partial Z}\right)^{\mathsf{T}}$$

# Proof

$$\Delta Y = W\Delta X$$

$$\Delta L \cong \frac{\partial L}{\partial Y}\Delta Y$$

$$\Delta Y \cong W\Delta X$$

$$\Rightarrow \Delta L \cong \boxed{\frac{\partial L}{\partial Y}} W\Delta X \qquad = \frac{\partial L}{\partial X}$$

$$\therefore \left(\frac{\partial L}{\partial X}\right)^{\top} = W^{\top}\left(\frac{\partial L}{\partial Y}\right)^{\top}$$

$$\Delta y = \Delta W X$$

$$\Delta L \approx \mathrm{tr}\left(\frac{\partial L}{\partial W}\Delta W\right)$$

$$\Delta y \cong \Delta W X$$

$$\Delta L \approx \frac{\partial L}{\partial Y}\Delta Y$$

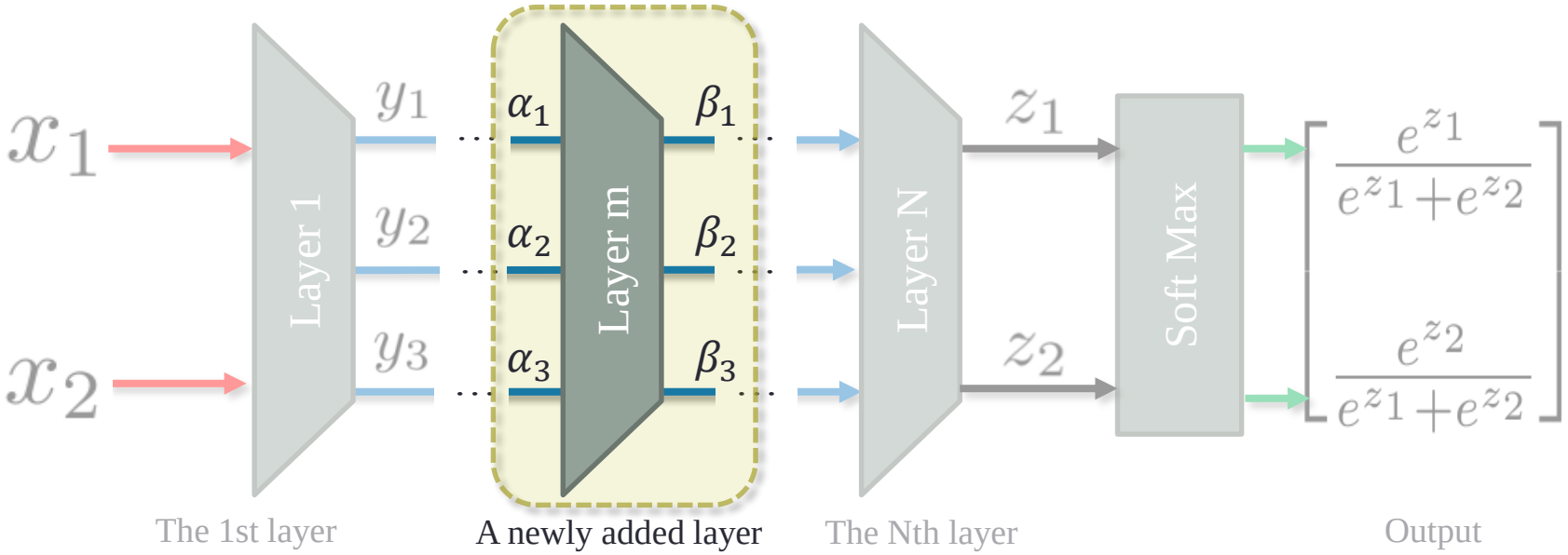
$$\Delta L \cong \frac{\partial L}{\partial Y}\Delta W X = \mathrm{tr}\left(\frac{\partial L}{\partial Y}\Delta W X\right)$$

$$= \mathrm{tr}\left(X\frac{\partial L}{\partial Y}\Delta W\right)$$

$$\therefore \left(\frac{\partial L}{\partial W}\right)^{\top} = \left(\frac{\partial L}{\partial Y}\right)^{\top} X^{\top}$$

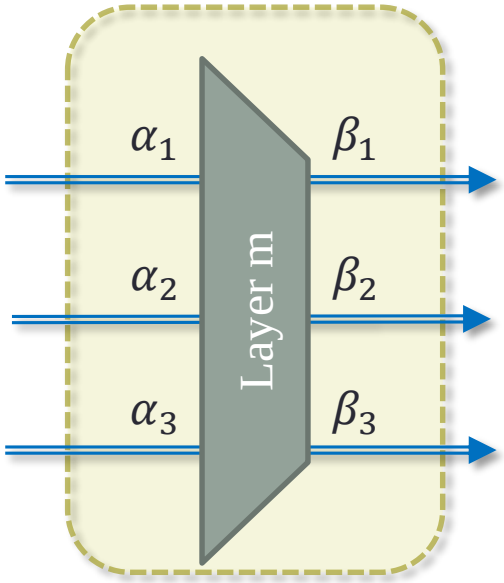
# New Layer Design

# New layer addition

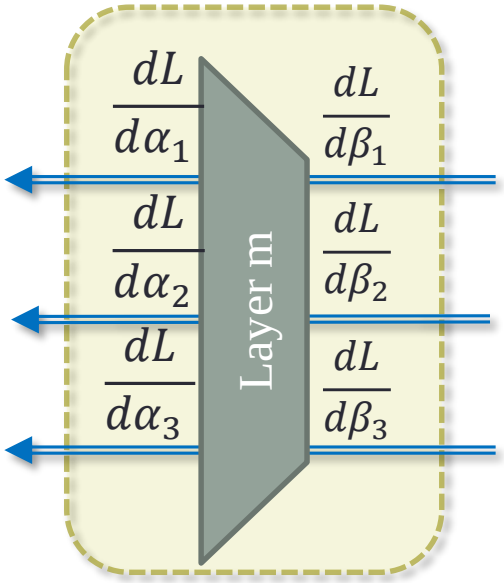


# New layer design

- Forward pass
  - Compute output from input
- Backward pass
  - Compute the derivatives w.r.t. data
  - Update weights ??

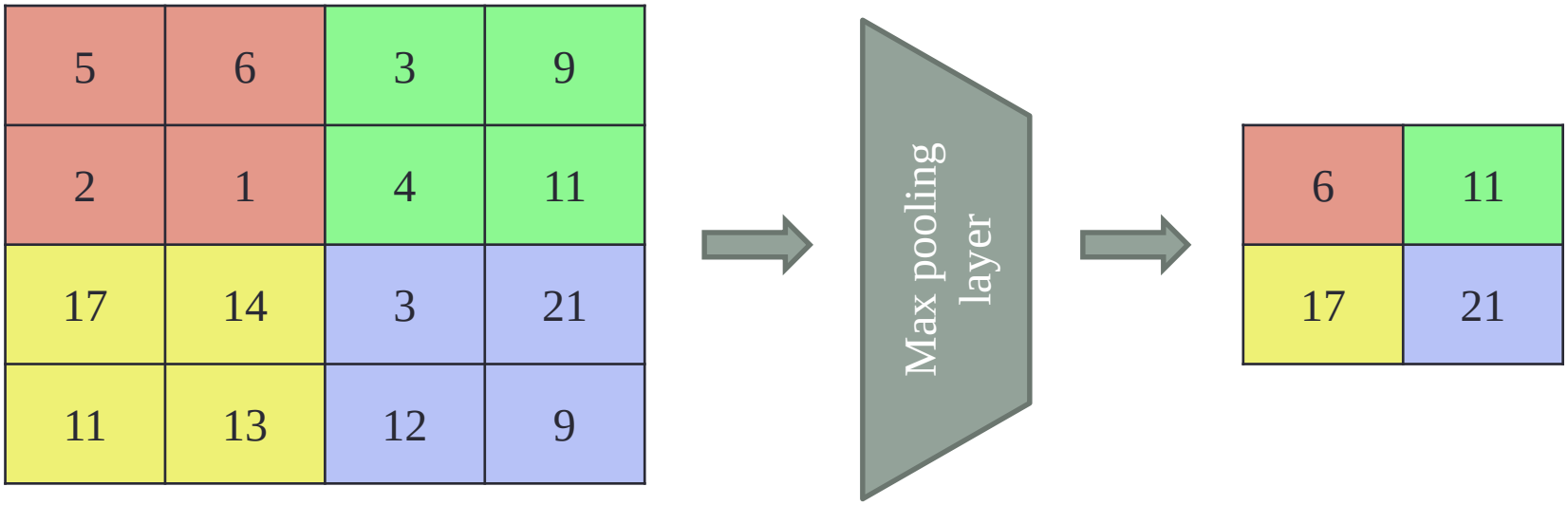



Forward pass




Backward pass

# Example: Max pooling



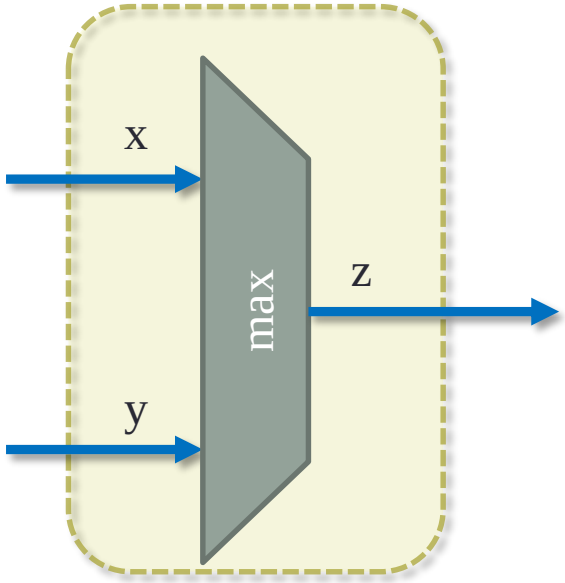
# Derivatives of max

- For forward pass

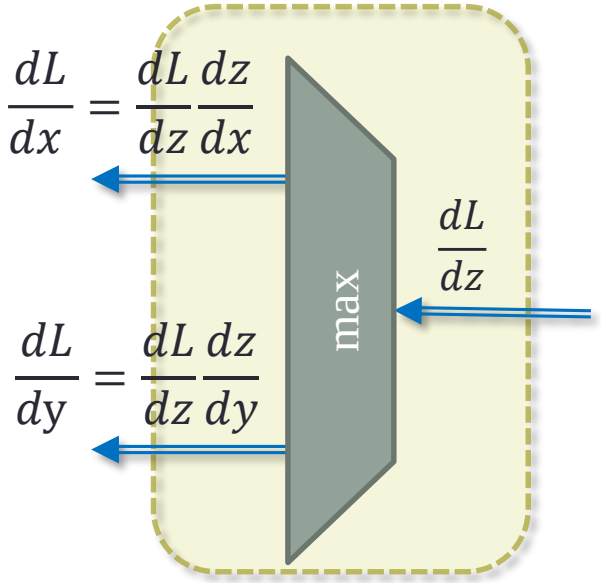
$$z = max(x, y) = \begin{cases} x & \text{if } x \geq y \\ y & \text{if } x < y \end{cases}$$

- For backward pass

$$\frac{dz}{dx} = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if } x < y \end{cases}$$




Forward pass




Backward pass