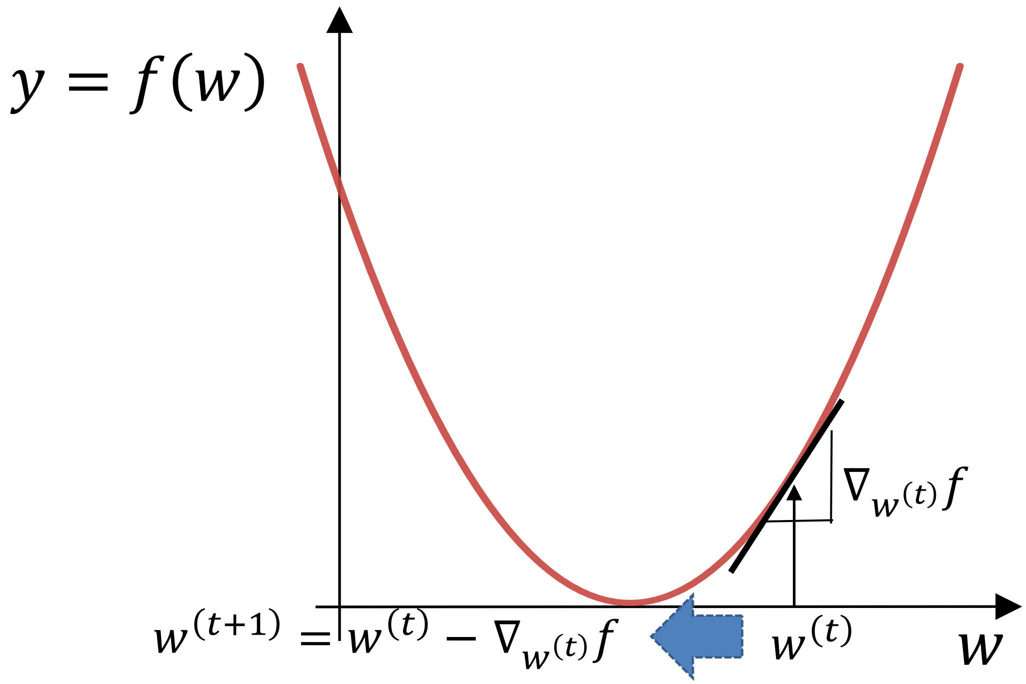
# Backpropagation

Younghoon Kim
(nongaussian@hanyang.ac.kr)

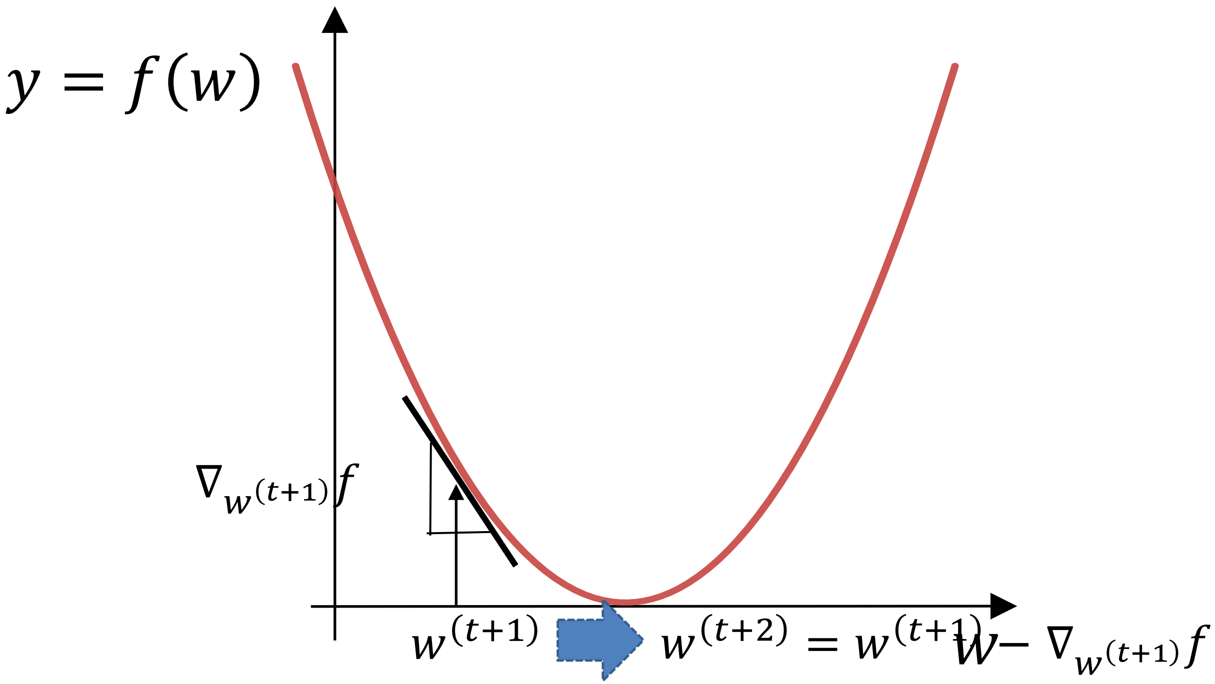
# Contents

- Gradient descent methods
- Computation graph
- Backpropagation
- Automatic gradient computation

# Gradient Descent



# Gradient Descent

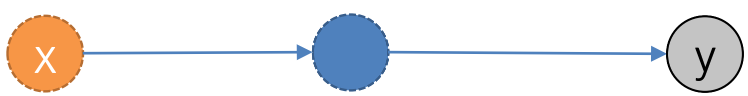


# Computation Graph

- Sigmoid function $y = \sigma(x) = \frac{1}{1+e^{-x}}$


$\alpha(x) = -x$ $\quad$ $\beta(x) = e^{x}$ $\quad$ $\gamma(x) = 1 + x$ $\quad$ $\delta(x) = \frac{1}{x}$

$$y = \sigma(x) = \delta\left(\gamma\left(\beta(\alpha(x))\right)\right) = (\delta \circ \gamma \circ \beta \circ \alpha)(x)$$




$y = \sigma(x)$

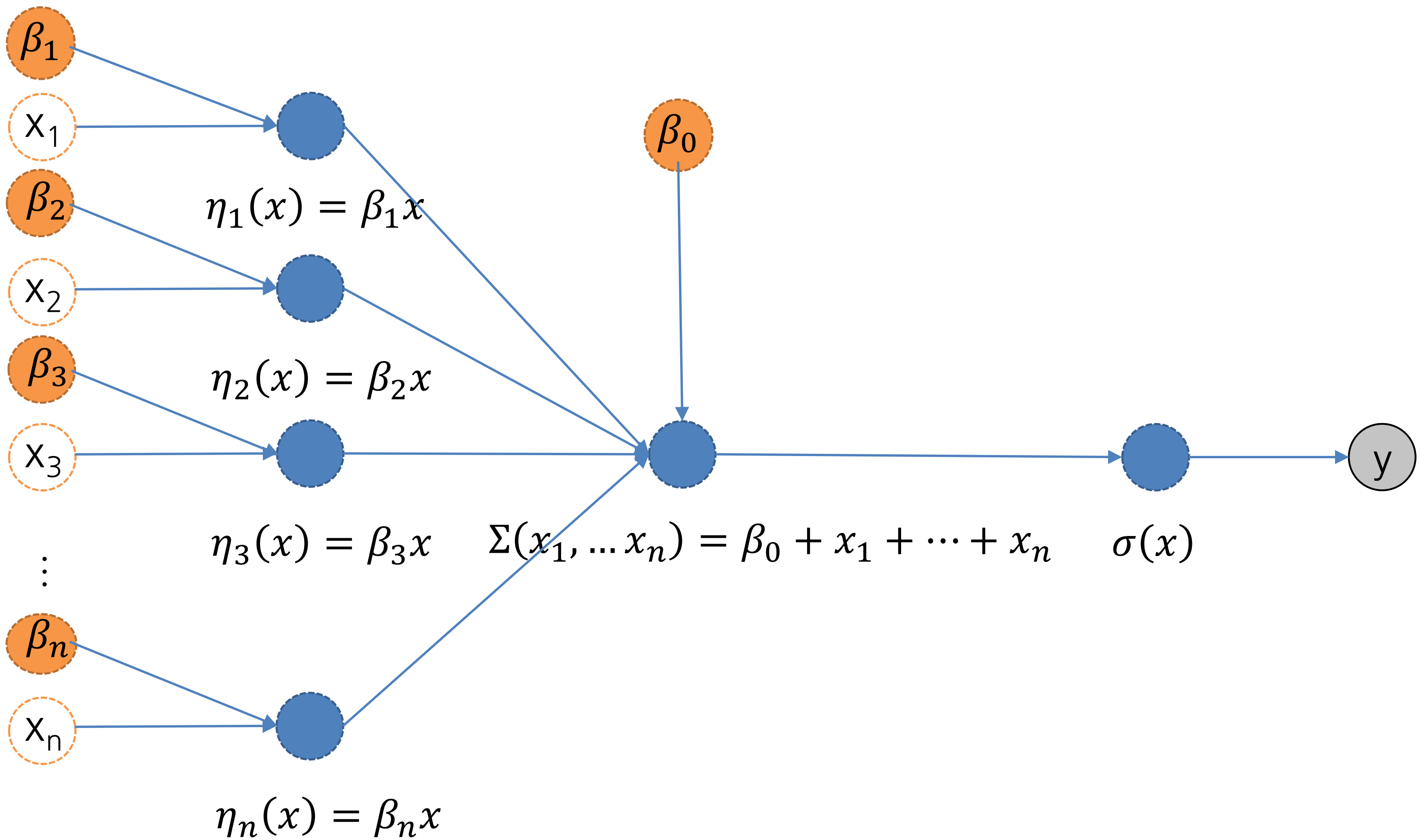
# Computation Graph

- Sigmoid function $y = \sigma(x) = \frac{1}{1+e^{-x}}$

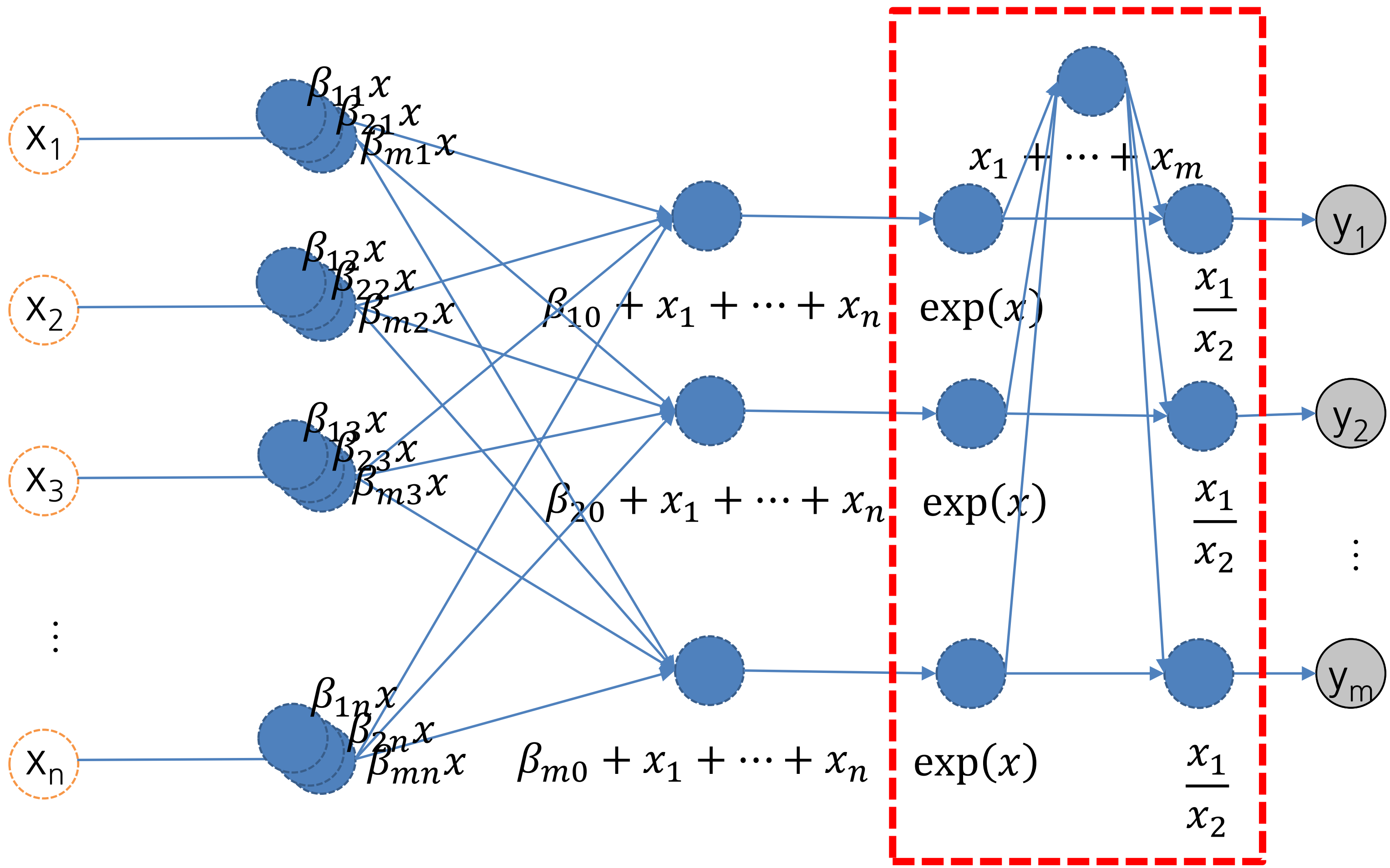

$\alpha(x) = -x$ $\quad \beta(x) = e^{x}$ $\quad \gamma(x) = 1 + x$ $\quad \delta(x) = \frac{1}{x}$

$$\frac{\partial y}{\partial x} = ?$$

# Logistic Regression

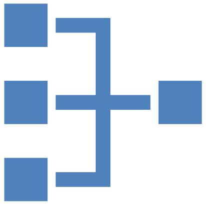


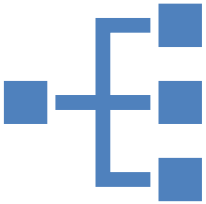
# Multivariate Logistic Regression



# BACKPROPAGATION

# Backpropagation





## Forward propagation

- 주어진 입력으로 계산 그래프(=네트워크)의 출력을 계산합니다.

## Backpropagation

- 전체 오류(Loss)에 기여하는 정도에 비례하여 네트워크의 모델 파라미터를 조정합니다.
- 백워드 패스로 꼬리(출력)에서 머리(입력) 방향로 가중치를 업데이트합니다.

# Training in PyTorch

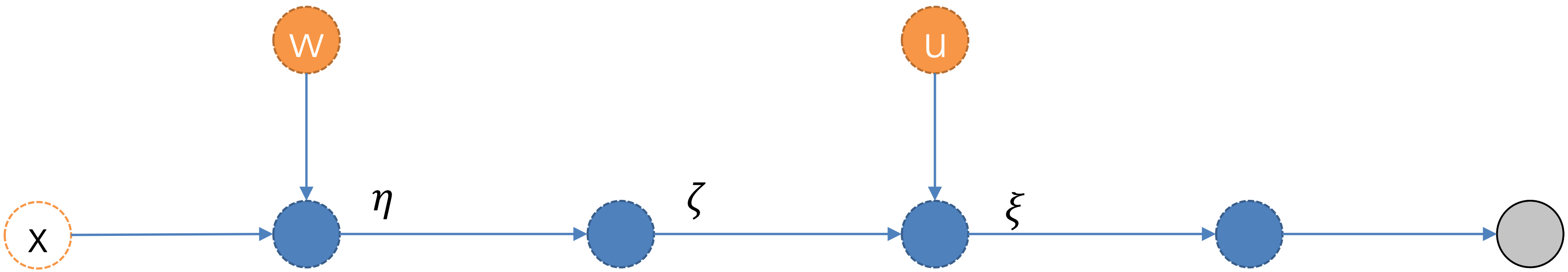
```
for epoch in range(EPOCHS):
    # Training loop
    for X, y in train_dataloader:

        model.train()

        y_pred = model(X)

        loss = loss_fn(y_pred, y)
        train_loss += loss.item()

        acc = accuracy(y_pred, y)
        train_acc += acc

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Forward propagation

Backpropagation

# Forward Propagation

- Let $\sigma(w, u; x) = \log(1 + ue^{-wx})$

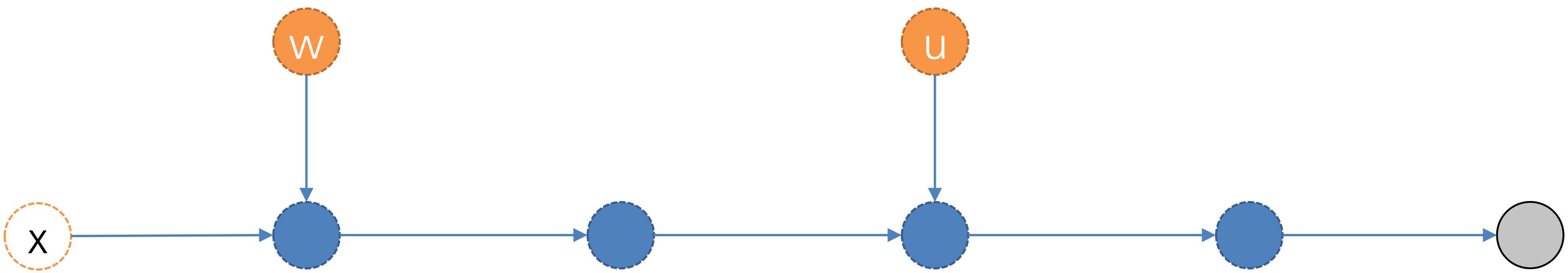



$\alpha(w; x) = -wx$ $\quad\beta(\eta) = e^{\eta}$ $\quad\gamma(u, \zeta) = 1 + u\zeta$ $\quad\delta(\xi) = \log \xi$

$$\sigma(w, u; x) = \delta\left(\gamma\left(\beta\big(\alpha(x)\big)\right)\right) = (\delta \circ \gamma \circ \beta \circ \alpha)(x)$$

# Backpropagation

- Let $\sigma(w, u; x) = \log(1 + ue^{-wx})$




$\alpha(w; x) = -wx \qquad \beta(\eta) = e^{\eta} \qquad \gamma(u, \zeta) = 1 + u\zeta \qquad \delta(\xi) = \log \xi$

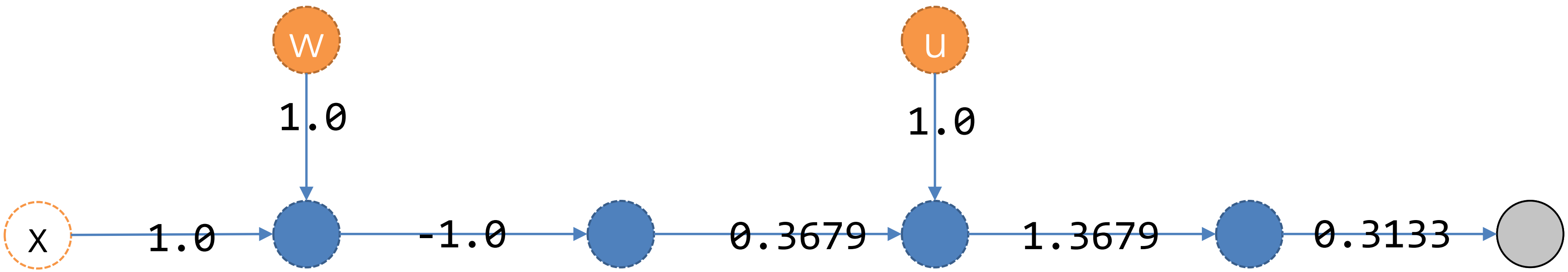
$$\sigma(w, u; x) = \delta\left(\gamma\left(\beta\big(\alpha(x)\big)\right)\right) = (\delta \circ \gamma \circ \beta \circ \alpha)(x)$$

$$\frac{\partial \sigma(w, u; x)}{\partial w} = \frac{\partial \delta(\xi)}{\partial \xi} \frac{\partial \gamma(u, \zeta)}{\partial \zeta} \frac{\partial \beta(\eta)}{\partial \eta} \frac{\partial \alpha(w; x)}{\partial w}$$

chain rule

# A Step of Iteration: Forward Propagation

- Let $\sigma(w, u; x) = \log(1 + ue^{-wx})$

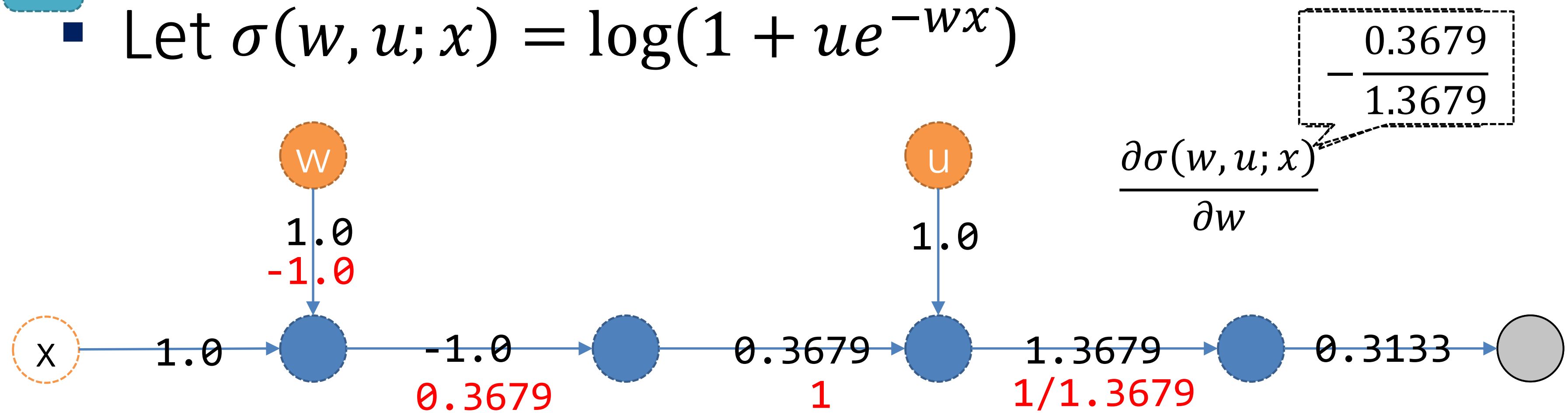



$\alpha(w; x) = -wx$ $\quad \beta(\eta) = e^{\eta}$ $\quad \gamma(u, \zeta) = 1 + u\zeta$ $\quad \delta(\xi) = \log \xi$

$$\frac{\partial \alpha(w; x)}{\partial w} = -1.0 \qquad \frac{\partial \beta(\eta)}{\partial \eta} = 0.3679 \qquad \frac{\partial \gamma(u, \zeta)}{\partial \zeta} = 1.0 \qquad \frac{\partial \delta(\xi)}{\partial \xi} = \frac{1}{1.3679}$$

# A Step of Iteration: Backpropagation

- Let $\sigma(w, u; x) = \log(1 + ue^{-wx})$




$\alpha(w; x) = -wx$ $\quad\beta(\eta) = e^{\eta}$ $\quad\gamma(u, \zeta) = 1 + u\zeta$ $\quad\delta(\xi) = \log \xi$

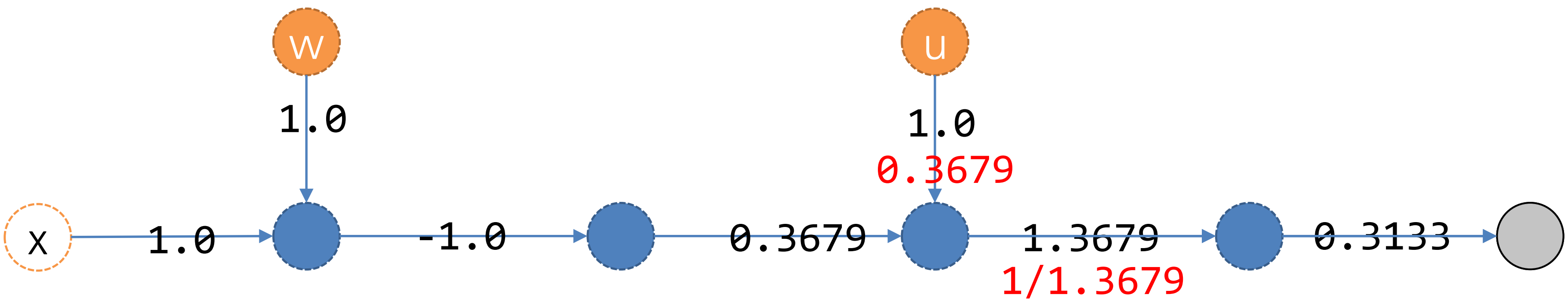
$\frac{\partial \alpha(w;x)}{\partial w} = -1.0$ $\quad\frac{\partial \beta(\eta)}{\partial \eta} = 0.3679$ $\quad\frac{\partial \gamma(u,\zeta)}{\partial \zeta} = 1.0$ $\quad\frac{\partial \delta(\xi)}{\partial \xi} = \frac{1}{1.3679}$

$$\frac{\partial \sigma(w,u;x)}{\partial w} = \frac{\partial \delta(\xi)}{\partial \xi}\frac{\partial \gamma(u,\zeta)}{\partial \zeta}\frac{\partial \beta(\eta)}{\partial \eta}\frac{\partial \alpha(w;x)}{\partial w} = \frac{1}{1.3679} \cdot 1.0 \cdot 0.3679 \cdot (-1.0) = -0.2690$$

$$w^{(t+1)} = w^{(t)} + lr\frac{\partial \sigma(w,u;x)}{\partial w} = 1.0 + (0.001) \cdot (-0.2690) = 0.9997$$

# A Step of Iteration: Backpropagation

- Let $\sigma(w,u;x) = \log(1 + ue^{-wx})$




$\alpha(w;x) = -wx$ $\quad$ $\beta(\eta) = e^{\eta}$ $\quad$ $\gamma(u,\zeta) = 1 + u\zeta$ $\quad$ $\delta(\xi) = \log \xi$

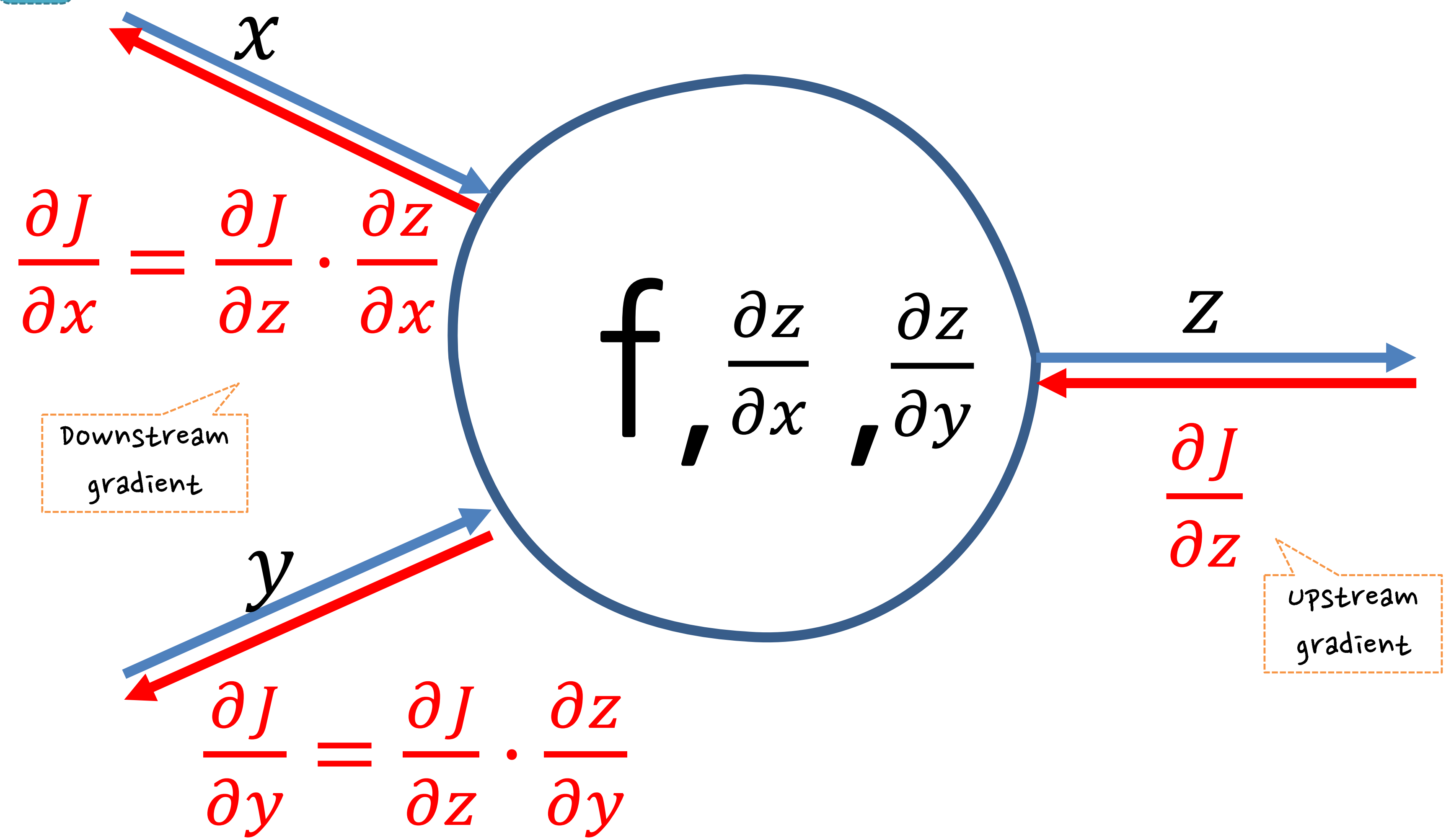
$$\frac{\partial\alpha(w;x)}{\partial w} = -1.0 \qquad \frac{\partial\beta(\eta)}{\partial\eta} = 0.3679 \qquad \frac{\partial\gamma(u,\zeta)}{\partial\zeta} = 1.0 \qquad \frac{\partial\delta(\xi)}{\partial\xi} = \frac{1}{1.3679}$$

$$\frac{\partial\gamma(u,\zeta)}{\partial u} = 0.3679$$

$$\frac{\partial\sigma(w,u;x)}{\partial u} = \frac{\partial\delta(\xi)}{\partial\xi}\frac{\partial\gamma(u,\zeta)}{\partial u} = \frac{1}{1.3679}\cdot 0.3679 = 0.2690$$

$$u^{(t+1)} = u^{(t)} + lr\frac{\partial\sigma(w,u;x)}{\partial u} = 1.0 + (0.001)\cdot(0.2690) = 1.0003$$

# Flow of Gradients

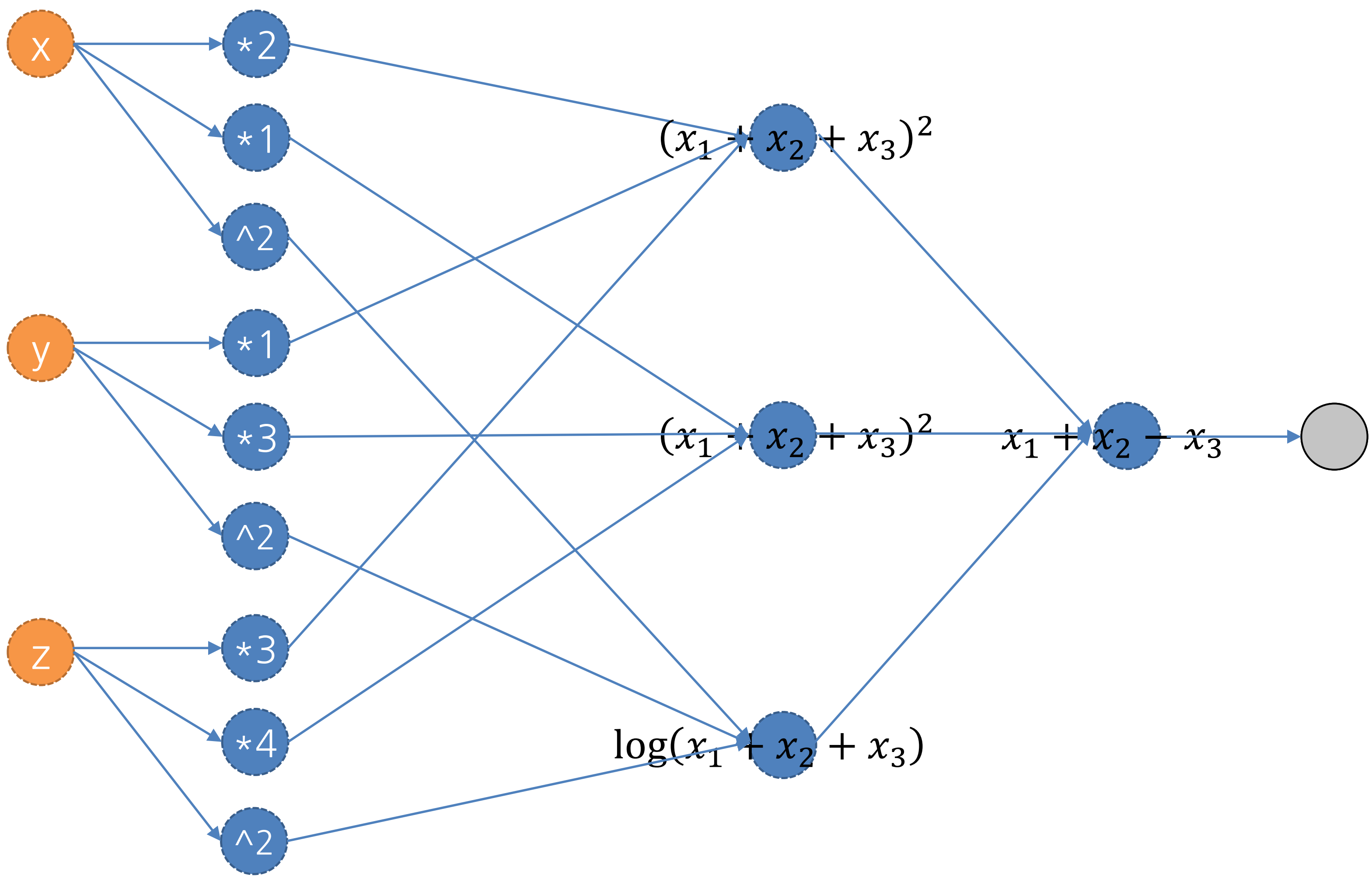


# Exercise: Problem

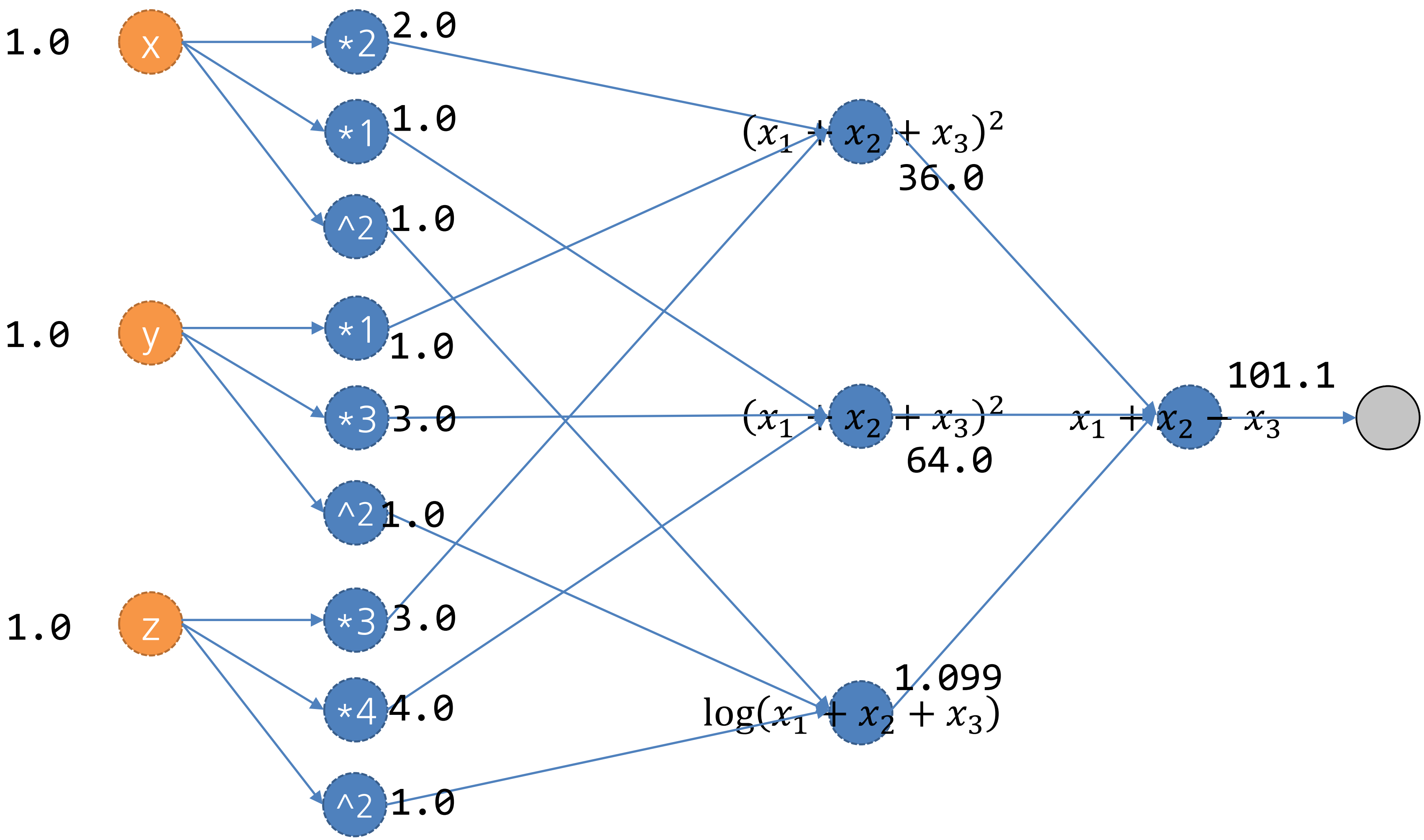
- Let
  - $f(x,y,z) = (2x + y + 3z)^2 + (x + 3y + 4z)^2 - \log(x^2 + y^2 + z^2)$
- Draw the computation graph
- For (x, y, z) = (1, 1, 1)
  - Compute the gradient with backpropagation
    - 1) Perform the forward computation
    - 2) Then, do the backpropagation

$$\frac{\delta f(x,y,z)}{\delta x} = 10x + 10y + 20z - \frac{2x}{x^2 + y^2 + z^2}$$

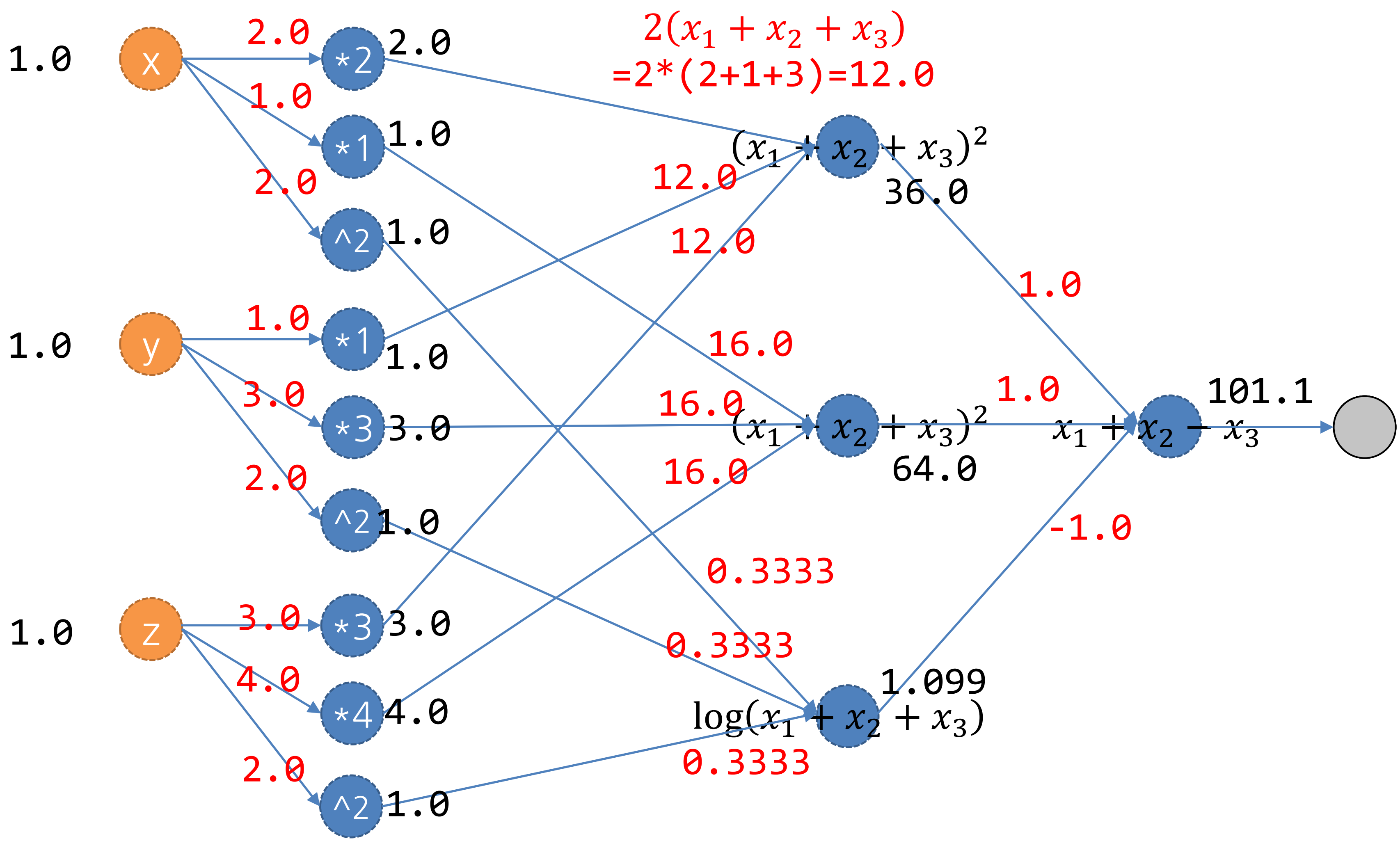
# Exercise: Computation Graph



# Exercise: Forward Computation

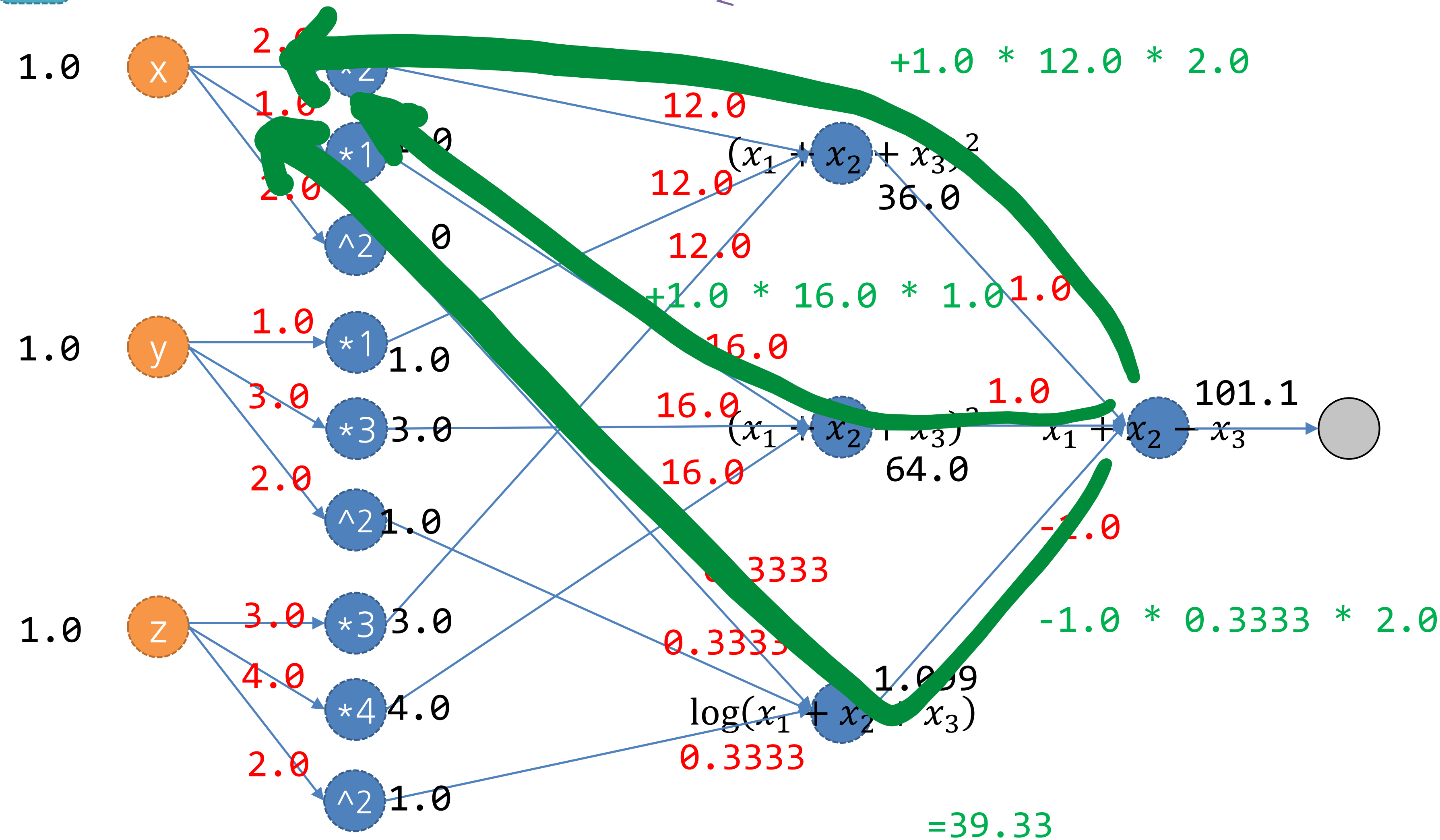


# Exercise: Backpropagation



# Exercise:

$$\frac{\delta f(x,y,z)}{\delta x} = 10x + 10y + 20z - \frac{2x}{x^2 + y^2 + z^2}$$



# Exercise

- 다음 f()를 최소화하기 위해 초기값 1, 1, 1에서 시작하여 SGD를 이용해 100번 반복하여 변수 x, y, z를 갱신하시오.

$$f(x, y, z) = (x + y + z)^2 + (x - 1)^2 + (y - 1)^2 + (z - 1)^2$$

# Optimization with PyTorch

- 다음 f()를 최소화하기 위해 초기값 1, 1, 1에서 시작하여 SGD를 이용해 100번 반복하여 변수 x, y, z를 갱신하시오.

$$f(x, y, z) = (x + y + z)^2 + (x - 1)^2 + (y - 1)^2 + (z - 1)^2$$

# Exercise

- Let
  - $f(x, y) = 2x^2 + y^2 + e^{4xy}$
- Initially, (x, y) = (1, 1)
  - $f(x, y)$ 를 최소화하기 위해 경사하강법을 100번 반복하여 보세요.