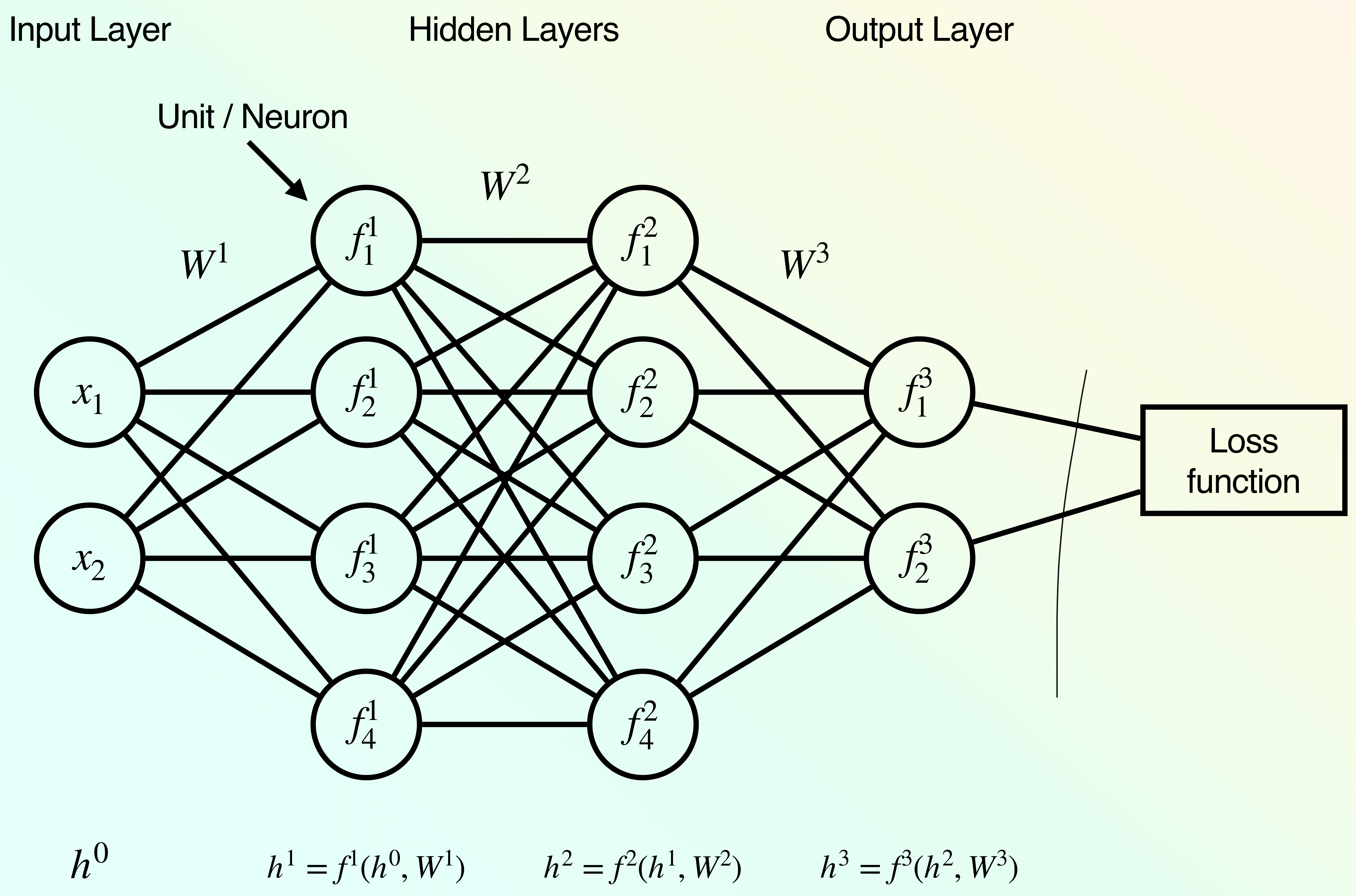
GRADIENTS OF NEURAL NETWORKS


Input Layer
Hidden Layers
Output Layer
Unit / Neuron
W^1
W^2
W^3
x_1
x_2
f_1^1
f_2^1
f_3^1
f_4^1
f_1^2
f_2^2
f_3^2
f_4^2
f_1^3
f_2^3
Loss function
h^0
$h^1 = f^1(h^0, W^1)$
$h^2 = f^2(h^1, W^2)$
$h^3 = f^3(h^2, W^3)$

THE STRUCTURE OF NEURAL NETWORKS

ABSTRACT PROCESS

We can write the neural network outputs as a sequential process.

$$
\begin{aligned}
h^0 &= x \\
h^1 &= f^1(h^0, W^1) \\
h^2 &= f^2(h^1, W^2) \\
&\vdots \\
h^i &= f^i(h^{i-1}, W^i) \\
&\vdots \\
h^k &= f^k(h^{k-1}, W^k)
\end{aligned}
$$

To be concise, we can write the network output as $h^k = f(x, \theta)$, $\theta = \{W^i\}_{i=1}^k$

LOSS FUNCTION

EXAMPLE

$$l(\theta) = \frac{1}{2}\mathbf{E}\left[\left(f(X,\theta) - Y\right)^2\right]$$

→ will apply it backward for each h with respect to $\underset{\sim}{w}$ for each layer

We have a batch of data $D = (x, y)$ of m samples

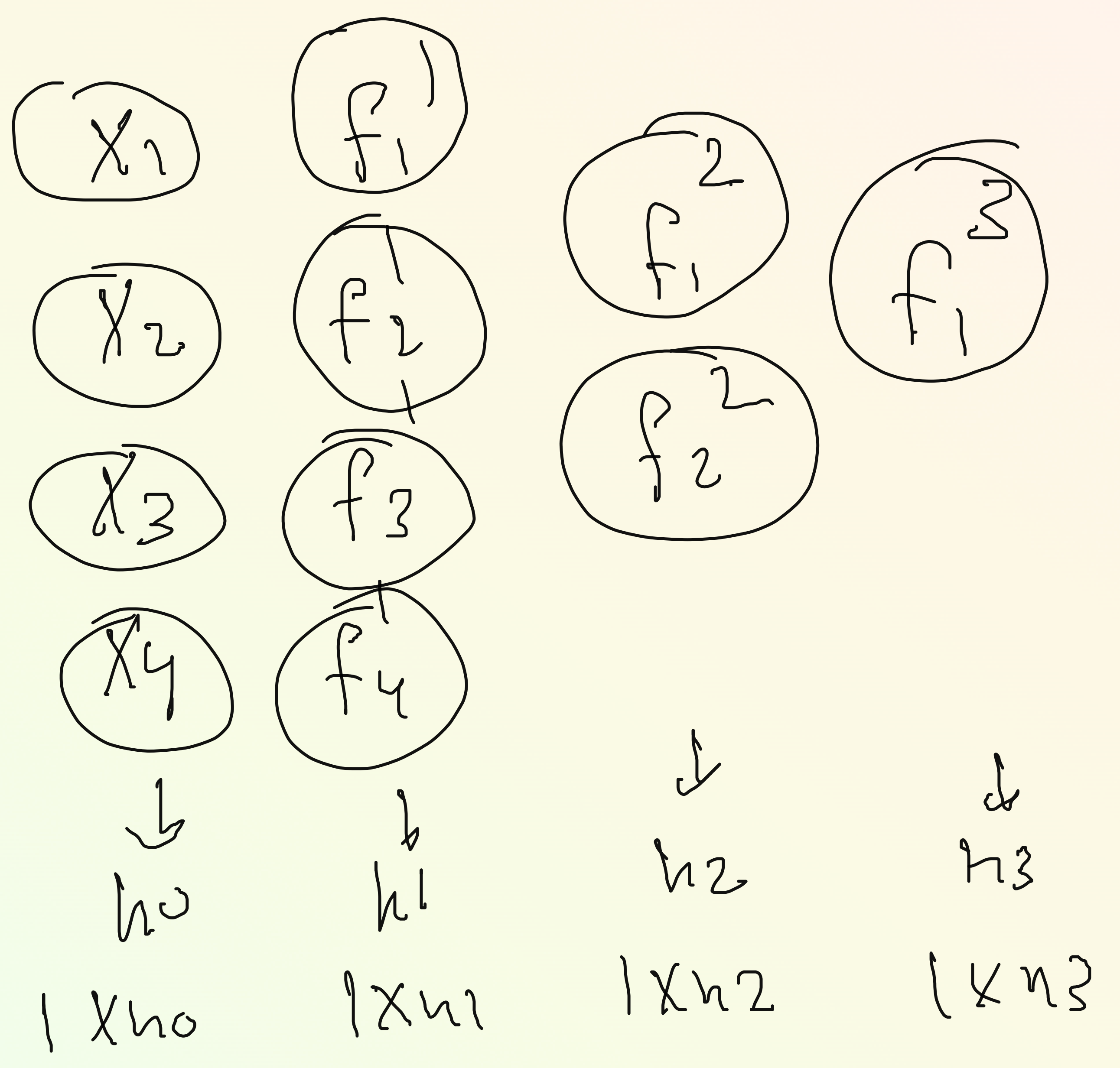
$x \in \mathbb{R}^{m \times n_0}$ and $y \in \mathbb{R}^{m \times 1}$, x_i and y_i are the features and target for the i^{th} data point.

$$l_D(\theta) = \frac{1}{m}\frac{1}{2}\sum_{i=1}^{m}\left(f(x_i,\theta) - y_i\right)^2$$

homework 3

LOSS FUNCTION

EXAMPLE



$$l(\theta) = \frac{1}{2}\mathbf{E}\left[\left(f(X,\theta) - Y\right)^2\right]$$

We have a batch of data $D = (x, y)$ of m samples

$x \in \mathbb{R}^{m \times n_0}$ and $y \in \mathbb{R}^{m \times 1}$, x_i and y_i are the features and target for the i^{th} data point.

$$l_D(\theta) = \frac{1}{m}\frac{1}{2}\|f_i(X,\theta) - y\|_2^2$$

$$l(\theta) = \frac{1}{2}\frac{1}{m}\left(h^3_{i,1} - y_1\right)^2$$

$$\theta = \theta - \eta \nabla l(\theta)$$

$$\nabla l(\theta) = \begin{bmatrix} \frac{\partial l(\theta)}{\partial w^3} \\ \frac{\partial l(\theta)}{\partial w^2} \\ \frac{\partial l(\theta)}{\partial w^1} \end{bmatrix}$$

GRADIENT DESCENT

$$\frac{\partial l(\theta)}{\partial h^3_{1,1}} = (h_{1,1} - y_i) = \delta_1$$

$$h^3_{1,1} = f^3_1(h^2, w^3) = h^2 w^{3\,T}_{(n_3 \times n_2)}{}^T$$

$$1 \times n_2 \quad n_2 \times n_3 \quad 2 \times 1$$

$$\begin{bmatrix} h^2_{1,1} & h^2_{1,2} \end{bmatrix} \begin{bmatrix} w^3_{1,1} \\ w^3_{1,2} \end{bmatrix}$$

$$\frac{\partial l(\theta)}{\partial w^3_{1,1}} = \frac{\partial l(\theta)}{\partial h^3_{1,1}} \frac{\partial h^3_{1,1}}{\partial w^3_{1,1}}$$

SET UP

$$\nabla l(\theta) = ?$$

We need to know compute the loss function's gradient with respect to all the parameters in the neural network before running gradient descent.

BACKPROP

FORWARD PASS

Compute the outputs of each layer and the loss $l_D(\theta)$

$$
\begin{aligned}
h^0 &= x \\
h^1 &= f^1(h^0, W^1) \\
h^2 &= f^2(h^1, W^2) \\
&\vdots \\
h^i &= f^i(h^{i-1}, W^i) \\
&\vdots \\
h^k &= f^k(h^{k-1}, W^k)
\end{aligned}
$$

BACKPROP

BACKWARD PASS

Using the results of the forward pass, apply the chain rule to compute the derivatives for each layer

Compute $\frac{\partial l_D(\theta)}{\partial f(X, \theta)}$

Then compute $\frac{\partial l_D(\theta)}{\partial W^k}$ and $\frac{\partial l_D(\theta)}{\partial h^{k-1}}$

Then compute $\frac{\partial l_D(\theta)}{\partial W^{k-1}}$ and $\frac{\partial l_D(\theta)}{\partial h^{k-2}}$

Repeat till W^1

$$
\begin{aligned}
h^0 &= x \\
h^1 &= f^1(h^0, W^1) \\
h^2 &= f^2(h^1, W^2) \\
&\vdots \\
h^i &= f^i(h^{i-1}, W^i) \\
&\vdots \\
h^k &= f^k(h^{k-1}, W^k)
\end{aligned}
$$

$$\frac{\partial l_D(\theta)}{\partial f(X, \theta)}$$

BACKPROP

BACKWARD PASS

What are the partial derivatives for $\frac{\partial l_D(\theta)}{\partial W^i}$ and $\frac{\partial l_D(\theta)}{\partial h^i}$?

NEXT CLASS

Next Class — Training Neural Networks