Back-Propagation

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University
World’s Smallest Perceptron!

\[ y = wx \]

(a.k.a. line equation, linear regression)

back to the...

function of **ONE** parameter!
Training the world’s smallest perceptron

This is just gradient descent, that means…

this should be the gradient of the loss function

Now where does this come from?
\[ \frac{dL}{dw} \] ...is the rate at which \textbf{this} will change...

\[ L = \frac{1}{2} (y - \hat{y})^2 \]

the loss function

... per unit change of \textbf{this}

\[ y = wx \]

the weight parameter

Let’s compute the derivative...
Compute the derivative

\[
\frac{d\mathcal{L}}{dw} = \frac{d}{dw} \left\{ \frac{1}{2} (y - \hat{y})^2 \right\} \\
= -(y - \hat{y}) \frac{dwx}{dw} \\
= -(y - \hat{y})x = \nabla w \quad \text{just shorthand}
\]

That means the weight update for gradient descent is:

\[
w = w - \nabla w \quad \text{move in direction of negative gradient}
\]

\[
w = w + (y - \hat{y})x
\]
Gradient Descent (world’s smallest perceptron)

For each sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass \( \hat{y} = wx_i \)
   b. Compute Loss \( \mathcal{L}_i = \frac{1}{2}(y_i - \hat{y})^2 \)

2. Update
   a. Back Propagation \( \frac{d\mathcal{L}_i}{dw} = -(y_i - \hat{y})x_i = \nabla w \)
   b. Gradient update \( w = w - \nabla w \)
Training the world’s smallest perceptron
world’s (second) smallest perceptron!

function of two parameters!
Gradient Descent

For each sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass
   b. Compute Loss

2. Update
   a. Back Propagation
   b. Gradient update

we just need to compute partial derivatives for this network
Back-Propagation

\[ \frac{\partial L}{\partial w_1} = \frac{\partial}{\partial w_1} \left\{ \frac{1}{2} (y - \hat{y})^2 \right\} \]

\[ = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_1} \]

\[ = -(y - \hat{y}) \frac{\partial \sum_i w_i x_i}{\partial w_1} \]

\[ = -(y - \hat{y}) \frac{\partial w_1 x_1}{\partial w_1} \]

\[ = -(y - \hat{y}) x_1 = \nabla w_1 \]

\[ \frac{\partial L}{\partial w_2} = \frac{\partial}{\partial w_2} \left\{ \frac{1}{2} (y - \hat{y})^2 \right\} \]

\[ = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_2} \]

\[ = -(y - \hat{y}) \frac{\partial \sum_i w_i x_i}{\partial w_2} \]

\[ = -(y - \hat{y}) \frac{\partial w_2 x_2}{\partial w_2} \]

\[ = -(y - \hat{y}) x_2 = \nabla w_2 \]

Why do we have partial derivatives now?
Back-Propagation

\[
\frac{\partial L}{\partial w_1} = \frac{\partial}{\partial w_1} \left\{ \frac{1}{2} (y - \hat{y})^2 \right\} \\
= -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_1} \\
= -(y - \hat{y}) \frac{\partial \sum_i w_i x_i}{\partial w_1} \\
= -(y - \hat{y}) \frac{\partial w_1 x_1}{\partial w_1} \\
= -(y - \hat{y}) x_1 = \nabla w_1
\]

\[
\frac{\partial L}{\partial w_2} = \frac{\partial}{\partial w_2} \left\{ \frac{1}{2} (y - \hat{y})^2 \right\} \\
= -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_2} \\
= -(y - \hat{y}) \frac{\partial \sum_i w_i x_i}{\partial w_2} \\
= -(y - \hat{y}) \frac{\partial w_2 x_2}{\partial w_2} \\
= -(y - \hat{y}) x_2 = \nabla w_2
\]

Gradient Update

\[
w_1 = w_1 - \eta \nabla w_1 \\
= w_1 + \eta (y - \hat{y}) x_1
\]

\[
w_2 = w_2 - \eta \nabla w_2 \\
= w_2 + \eta (y - \hat{y}) x_2
\]
Gradient Descent

For each sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass \( \hat{y} = f_{\text{MLP}}(x_i; \theta) \)
   b. Compute Loss \( \mathcal{L}_i = \frac{1}{2}(y_i - \hat{y}) \)

2. Update
   a. Back Propagation
   b. Gradient update

\[ \nabla w_{1i} = -(y_i - \hat{y})x_{1i} \]
\[ \nabla w_{2i} = -(y_i - \hat{y})x_{2i} \]
\[ w_{1i} = w_{1i} + \eta(y - \hat{y})x_{1i} \]
\[ w_{2i} = w_{2i} + \eta(y - \hat{y})x_{2i} \]

(two BP lines now)

(adjustable step size)
We haven’t seen a lot of ‘propagation’ yet because our perceptrons only had one layer…
multi-layer perceptron

function of **FOUR** parameters and **FOUR** layers!
The diagram illustrates a neural network with four layers:

1. **Input Layer 1**:
   - Input: $x$
   - Weight: $w_1$
   - Bias: $b_1$

2. **Hidden Layer 2**:
   - Activation: $a_1 = f_1(w_1x + b_1)$

3. **Hidden Layer 3**:
   - Weight: $w_2$
   - Activation: $a_2 = f_2(a_1w_2)$

4. **Output Layer 4**:
   - Weight: $w_3$
   - Activation: $a_3 = f_3(a_2w_3)$

The output layer $a_3$ is connected to the final output $y$. The network processes the input $x$ through these layers, applying weights and activation functions at each step.
\[ a_1 = w_1 \cdot x + b_1 \]
\[ a_1 = w_1 \cdot x + b_1 \]
\[
\begin{align*}
    a_1 &= w_1 \cdot x + b_1 \\
    a_2 &= w_2 \cdot f_1(w_1 \cdot x + b_1)
\end{align*}
\]
\[ a_1 = w_1 \cdot x + b_1 \]
\[ a_2 = w_2 \cdot f_1(w_1 \cdot x + b_1) \]
\[ a_1 = w_1 \cdot x + b_1 \]

\[ a_2 = w_2 \cdot f_1(w_1 \cdot x + b_1) \]

\[ a_3 = w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1)) \]

\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]
\[ a_1 = w_1 \cdot x + b_1 \]
\[ a_2 = w_2 \cdot f_1(w_1 \cdot x + b_1) \]
\[ a_3 = w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1)) \]
\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]
\[ a_1 = w_1 \cdot x + b_1 \]
\[ a_2 = w_2 \cdot f_1(w_1 \cdot x + b_1) \]
\[ a_3 = w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1)) \]
\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]
Entire network can be written out as one long equation

\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]

We need to train the network:

*What is known? What is unknown?*
Entire network can be written out as a long equation

\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]

We need to train the network:

*What is known? What is unknown?*
Entire network can be written out as a long equation

\[ y = f_3(w_3 \cdot f_2(w_2 \cdot f_1(w_1 \cdot x + b_1))) \]

activation function sometimes has parameters

We need to train the network:

*What is known? What is unknown?*
Learning an MLP

Given a set of samples and a MLP

$$\{x_i, y_i\}$$

$$y = f_{\text{MLP}}(x; \theta)$$

Estimate the parameters of the MLP

$$\theta = \{f, w, b\}$$
Stochastic Gradient Descent

For each random sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass
   b. Compute Loss

2. Update
   a. Back Propagation
   b. Gradient update

\[ \hat{y} = f_{\text{MLP}}(x_i; \theta) \]

\[ \frac{\partial L}{\partial \theta} \]

\[ \theta \leftarrow \theta - \eta \nabla \theta \]

vector of parameter partial derivatives

vector of parameter update equations
So we need to compute the partial derivatives

$$\frac{\partial L}{\partial \theta} = \begin{bmatrix} \frac{\partial L}{\partial w_3} & \frac{\partial L}{\partial w_2} & \frac{\partial L}{\partial w_1} & \frac{\partial L}{\partial b} \end{bmatrix}$$
Remember,

Partial derivative \( \frac{\partial L}{\partial w_1} \) describes...

So, how do you compute it?
The Chain Rule
According to the chain rule...

\[
\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]

Intuitively, the effect of weight on loss function:

\[
\frac{\partial L}{\partial w_3}
\]

rest of the network \[\cdots\]

\[
f_2 \xrightarrow{w_3} a_3 \xrightarrow{f_3} \hat{y} \rightarrow L(y, \hat{y})
\]

depends on \[
\frac{\partial f_3}{\partial a_3}
\]

depends on \[
\frac{\partial a_3}{\partial w_3}
\]

depends on \[
\frac{\partial L}{\partial f_3}
\]
\[
\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]

Chain Rule!
rest of the network \[ f_2 \xrightarrow{w_3} a_3 \xrightarrow{f_3} \hat{y} \]

\[
\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]

\[= -\eta (y - \hat{y}) \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}\]

Just the partial derivative of L2 loss
rest of the network

\[
\begin{align*}
\frac{\partial L}{\partial w_3} & = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3} \\
& = -\eta(y - \hat{y}) \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\end{align*}
\]

Let’s use a Sigmoid function

\[
\frac{ds(x)}{dx} = s(x)(1 - s(x))
\]
Let's use a Sigmoid function

\[
\frac{ds(x)}{dx} = s(x)(1 - s(x))
\]
\[
\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]
\[
= -\eta(y - \hat{y}) \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]
\[
= -\eta(y - \hat{y}) f_3(1 - f_3) \frac{\partial a_3}{\partial w_3}
\]
\[
= -\eta(y - \hat{y}) f_3(1 - f_3) f_2
\]
\[
\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial w_2}
\]

already computed. re-use (propagate)!
The Chain rule says...

\[
\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1}
\]
The Chain rule says...

\[
\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1}
\]

already computed.
re-use (propagate)!
\[
\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3}
\]

\[
\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial w_2}
\]

\[
\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1}
\]

\[
\frac{\partial L}{\partial b} = \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial b}
\]
depends on

\[
\begin{align*}
\frac{\partial L}{\partial w_3} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3} \\
\frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial w_2} \\
\frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1} \\
\frac{\partial L}{\partial b} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial f_1} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial b}
\end{align*}
\]
\[
\begin{align*}
\frac{\partial L}{\partial w_3} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3} \\
\frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial f_2} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial w_2} \\
\frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial f_2}{\partial a_2} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1} \\
\frac{\partial L}{\partial b} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial f_2}{\partial a_2} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial b}
\end{align*}
\]
Stochastic Gradient Descent

For each example sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass
   \[ \hat{y} = f_{\text{MLP}}(x_i; \theta) \]
   b. Compute Loss \( L_i \)

2. Update
   a. Back Propagation
   \[
   \begin{align*}
   \frac{\partial L}{\partial w_3} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial a_3}{\partial w_3} \\
   \frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial f_2}{\partial a_2} \frac{\partial a_2}{\partial w_2} \\
   \frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial f_2}{\partial a_2} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial w_1} \\
   \frac{\partial L}{\partial b} &= \frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial a_3} \frac{\partial f_2}{\partial a_2} \frac{\partial f_1}{\partial a_1} \frac{\partial a_1}{\partial b}
   \end{align*}
   \]
   b. Gradient update
   \[
   \begin{align*}
   w_3 &= w_3 - \eta \nabla w_3 \\
   w_2 &= w_2 - \eta \nabla w_2 \\
   w_1 &= w_1 - \eta \nabla w_1 \\
   b &= b - \eta \nabla b
   \end{align*}
   \]
Stochastic Gradient Descent

For each example sample \( \{x_i, y_i\} \)

1. Predict
   a. Forward pass \( \hat{y} = f_{MLP}(x_i; \theta) \)
   b. Compute Loss \( \mathcal{L}_i \)

2. Update
   a. Back Propagation \( \frac{\partial \mathcal{L}}{\partial \theta} \)
   b. Gradient update \( \theta \leftarrow \theta + \eta \frac{\partial \mathcal{L}}{\partial \theta} \)

vector of parameter partial derivatives

vector of parameter update equations