CS 559 Deep Learning

Backpropagation

Gokberk Cinbis
Where we are...

\[ s = f(x; W) = Wx \]  \hspace{1cm} \text{scores function}

\[ L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \]  \hspace{1cm} \text{SVM loss}

\[ L = \frac{1}{N} \sum_{i=1}^{N} L_i + \sum_k W_k^2 \]  \hspace{1cm} \text{data loss + regularization}

want \[ \nabla_W L \]
Optimization

# Vanilla Gradient Descent

```python
while True:
    weights_grad = evaluate_gradient(loss_fun, data, weights)
    weights += - step_size * weights_grad  # perform parameter update
```

(image credits to Alec Radford)
Gradient Descent

\[
\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}
\]

**Numerical gradient**: slow :(, approximate :(, easy to write :)

**Analytic gradient**: fast :), exact :), error-prone :(  

In practice: Derive analytic gradient, check your implementation with numerical gradient

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Computational Graph

- $f = Wx$
- $L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$
Convolutional Network (AlexNet)

input image

weights

loss

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Neural Turing Machine

input tape

loss
Neural Turing Machine
\[ f(x, y, z) = (x + y)z \]
e.g. \( x = -2, y = 5, z = -4 \)
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, \ y = 5, \ z = -4 \)

\[ q = x + y \quad \frac{\partial q}{\partial x} = 1, \ \frac{\partial q}{\partial y} = 1 \]

\[ f = qz \quad \frac{\partial f}{\partial q} = z, \ \frac{\partial f}{\partial z} = q \]

Want: \( \frac{\partial f}{\partial x}, \ \frac{\partial f}{\partial y}, \ \frac{\partial f}{\partial z} \)
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, \ y = 5, \ z = -4 \)

\[
q = x + y \quad \frac{\partial q}{\partial x} = 1, \ \frac{\partial q}{\partial y} = 1
\]

\[
f = qz \quad \frac{\partial f}{\partial q} = z, \ \frac{\partial f}{\partial z} = q
\]

Want:

\[
\frac{\partial f}{\partial x}, \ \frac{\partial f}{\partial y}, \ \frac{\partial f}{\partial z}
\]
\[ f(x, y, z) = (x + y)z \]
e.g. \( x = -2, y = 5, z = -4 \)

\[
q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1
\]

\[
f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q
\]

Want: \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \)

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, y = 5, z = -4 \)

\[ q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1 \]

\[ f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q \]

Want: \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \)
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, \ y = 5, \ z = -4 \)

\[
q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1
\]

\[
f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q
\]

Want: \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \)
\[ f(x, y, z) = (x + y)z \]
e.g. \( x = -2, \ y = 5, \ z = -4 \)

\[
\begin{align*}
q &= x + y \\
\frac{\partial q}{\partial x} &= 1, \quad \frac{\partial q}{\partial y} = 1
\end{align*}
\]

\[
\begin{align*}
f &= qz \\
\frac{\partial f}{\partial q} &= z, \quad \frac{\partial f}{\partial z} = q
\end{align*}
\]

Want: \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \)
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, y = 5, z = -4 \)

\[ q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1 \]

\[ f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q \]

Want: \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \)

Chain rule:

\[ \frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} \]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson

CS 559 Deep Learning
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, y = 5, z = -4 \)

\[ q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1 \]

\[ f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q \]

Want:

\[ \frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial f}{\partial z} \]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
\[ f(x, y, z) = (x + y)z \]

e.g. \( x = -2, \ y = 5, \ z = -4 \)

\[ q = x + y \quad \frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1 \]

\[ f = qz \quad \frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q \]

Want:
\[ \frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial f}{\partial z} \]

Chain rule:
\[ \frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial x} \]
activations

\[ f(x) \]

\[ f(y) \]

\[ f(z) \]
activations

\[
\begin{align*}
\frac{\partial z}{\partial x} \\
\frac{\partial z}{\partial y}
\end{align*}
\]

"local gradient"
activations

\[ \frac{\partial L}{\partial x} = \frac{\partial z}{\partial x} \frac{\partial L}{\partial z} \frac{\partial z}{\partial x} \]

“local gradient”

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

\[ \frac{\partial z}{\partial y} \]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
activations

\[ \frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial x} \]

\[ \frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial y} \]

“local gradient”

gradients

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
activations

\[
\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial x}
\]

“local gradient”

\[
\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial y}
\]

gradients
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]
Another example:

\[
f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}
\]

\[
f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x
\]

\[
f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a
\]

\[
f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -\frac{1}{x^2}
\]

\[
f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1
\]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}} \]

\[ f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x \]

\[ f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a \]

\[ f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -\frac{1}{x^2} \]

\[ f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1 \]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ (-\frac{1}{1.37^2})(1.00) = -0.53 \]

\[ f(x) = e^x \quad \rightarrow \quad \frac{d f}{d x} = e^x \]

\[ f_a(x) = ax \quad \rightarrow \quad \frac{d f}{d x} = a \]

\[ f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{d f}{d x} = -\frac{1}{x^2} \]

\[ f_c(x) = c + x \quad \rightarrow \quad \frac{d f}{d x} = 1 \]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[
\begin{align*}
  f(x) &= e^x & \rightarrow & & \frac{df}{dx} &= e^x \\
  f_a(x) &= ax & \rightarrow & & \frac{df}{dx} &= a \\
  f_c(x) &= c + x & \rightarrow & & \frac{df}{dx} &= 1
\end{align*}
\]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ (1)(-0.53) = -0.53 \]

\[
\begin{align*}
  f(x) &= e^x \\
  f_a(x) &= a x
\end{align*}
\]

\[
\begin{align*}
  \frac{df}{dx} &= e^x \\
  \frac{df}{dx} &= a
\end{align*}
\]

\[
\begin{align*}
  f_c(x) &= c + x \\
  f(x) &= \frac{1}{x}
\end{align*}
\]

\[
\begin{align*}
  \frac{df}{dx} &= -1/x^2 \\
  \frac{df}{dx} &= 1
\end{align*}
\]
Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -\frac{1}{x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$
Another example: 

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ (e^{-1})(-0.53) = -0.20 \]

\[ f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x \]

\[ f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a \]

\[ f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2 \]

\[ f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1 \]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[
\begin{align*}
  f(x) &= e^x & \quad \frac{df}{dx} &= e^x \\
  f_a(x) &= ax & \quad \frac{df}{dx} &= a \\
  f_c(x) &= c + x & \quad \frac{df}{dx} &= 1 \\
  f(x) &= \frac{1}{x} & \quad \frac{df}{dx} &= -\frac{1}{x^2}
\end{align*}
\]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ (-1) \times (-0.20) = 0.20 \]

\[ f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x \]

\[ f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a \]

\[ f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -\frac{1}{x^2} \]

\[ f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1 \]
Another example:  \[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x \]

\[ f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a \]

\[ f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1 \]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ [\text{local gradient}] \times [\text{its gradient}] \]

\[ [1] \times [0.2] = 0.2 \]

\[ [1] \times [0.2] = 0.2 \] (both inputs!)

\[ f(x) = e^x \quad \Rightarrow \quad \frac{df}{dx} = e^x \]

\[ f_a(x) = ax \quad \Rightarrow \quad \frac{df}{dx} = a \]

\[ f(x) = \frac{1}{x} \quad \Rightarrow \quad \frac{df}{dx} = -\frac{1}{x^2} \]

\[ f_c(x) = c + x \quad \Rightarrow \quad \frac{df}{dx} = 1 \]
Another example:

\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[
\begin{align*}
  f(x) &= e^x \\
  f_a(x) &= ax
\end{align*}
\]

\[
\begin{align*}
  \frac{df}{dx} &= e^x \\
  \frac{df}{dx} &= a
\end{align*}
\]

\[
\begin{align*}
  f(x) &= \frac{1}{x} \\
  f_c(x) &= c + x
\end{align*}
\]

\[
\begin{align*}
  \frac{df}{dx} &= -\frac{1}{x^2} \\
  \frac{df}{dx} &= 1
\end{align*}
\]
Another example:

\[
f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}
\]

[local gradient] x [its gradient]

x0: [2] x [0.2] = 0.4
w0: [-1] x [0.2] = -0.2

\[
f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x
\]

\[
f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a
\]

\[
f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2
\]

\[
f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1
\]

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ \sigma(x) = \frac{1}{1 + e^{-x}} \]

\[ \frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left( \frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left( \frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x))\sigma(x) \]
\[ f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}} \]

\[ \sigma(x) = \frac{1}{1 + e^{-x}} \]

\[ d\sigma(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \left( \frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left( \frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x) \]

\[ (0.73) \times (1 - 0.73) = 0.2 \]
Patterns in backward flow

**add** gate: gradient distributor

**max** gate: gradient router

**mul** gate: gradient... “switcher”? 

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Gradients add at branches

See multivariable chain rule.
Implementation: forward/backward API

Graph (or Net) object. (Rough psuedo code)

```python
class ComputationalGraph(object):
    #...
    def forward(inputs):
        # 1. [pass inputs to input gates...]
        # 2. forward the computational graph:
        for gate in self.graph.nodes_topologically_sorted():
            gate.forward()
        return loss # the final gate in the graph outputs the loss

    def backward():
        for gate in reversed(self.graph.nodes_topologically_sorted()):
            gate.backward() # little piece of backprop (chain rule applied)
        return inputs_gradients
```

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Implementation: forward/backward API

\[ \text{(x,y,z are scalars)} \]
Implementation: forward/backward API

```python
class MultiplyGate(object):
    def forward(x, y):
        z = x * y
        self.x = x  # must keep these around!
        self.y = y
        return z

    def backward(dz):
        dx = self.y * dz  # [dz/dx * dL/dz]
        dy = self.x * dz  # [dz/dy * dL/dz]
        return [dx, dy]
```

(x, y, z are scalars)
Example: Torch Layers
Example: Torch Layers
Example: Torch MulConstant

\[ f(X) = aX \]

- **initialization**
- **forward**
- **backward**
Example: Caffe Layers
Caffe Sigmoid Layer (CPU)

\[ \sigma(x) = \frac{1}{1 + e^{-x}} \]

\[(1 - \sigma(x)) \sigma(x) \]

*top_diff (chain rule)
Gradients for vectorized code (\(x, y, z\) are now vectors)

This is now the Jacobian matrix (derivative of each element of \(z\) w.r.t. each element of \(x\))
Vectorized operations

\[ f(x) = \max(0, x) \] (elementwise)

4096-d input vector \rightarrow f(x) = \max(0, x) \rightarrow 4096-d output vector
Vectorized operations

$$f(x) = \max(0, x) \quad (\text{elementwise})$$

$$\frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} \frac{\partial L}{\partial f}$$

Q: what is the size of the Jacobian matrix?

4096-d input vector

4096-d output vector

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Vectorized operations

\[ \frac{\partial L}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial L}{\partial f} \end{bmatrix} \]

Jacobian matrix

Q: what is the size of the Jacobian matrix? [4096 x 4096!]

Q2: what does it look like?

4096-d input vector

\[ f(x) = \max(0,x) \text{ (elementwise)} \]

4096-d output vector

Slide by Fei-Fei Li, Andrej Karpathy & Justin Johnson
Vectorized operations

4096-d input vector

\( f(x) = \max(0,x) \) (elementwise)

4096-d output vector

Q: what is the size of the Jacobian matrix? [4096 x 4096!]

Q2: what does it look like? [A diagonal matrix of 1s and 0s]
Vectorized operations

in practice we process an entire minibatch (e.g. 100) of examples at one time:

$100 \times 4096$-d input vectors

$f(x) = \max(0, x)$ (elementwise)

$100 \times 4096$-d output vectors

i.e. Jacobian would technically be a $[409600 \times 409600]$ matrix.
Summary so far

- neural nets will be very large: no hope of writing down gradient formula by hand for all parameters
- **backpropagation** = recursive application of the chain rule along a computational graph to compute the gradients of all inputs/parameters/intermediates
- implementations maintain a graph structure, where the nodes implement the **forward() / backward()** API.
- **forward**: compute result of an operation and save any intermediates needed for gradient computation in memory
- **backward**: apply the chain rule to compute the gradient of the loss function with respect to the inputs.