CSC321 Lecture 6 Backpropagation

Roger Grosse and Nitish Srivastava

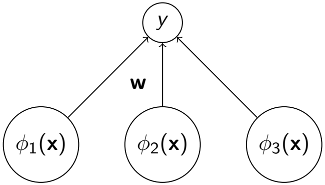
January 22, 2015

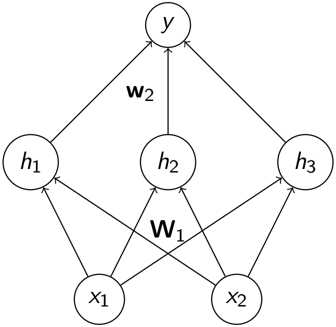
Overview

Recall: In our lecture on linear regression, we saw that we could make the model more powerful using basis functions (features).

We want to learn features instead of hand-designing them.

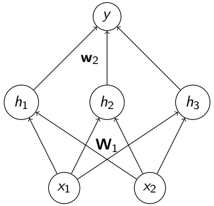
We will see how backpropagation can be used to learn them.

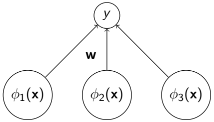




Overview

We can represent these features with a neural net:



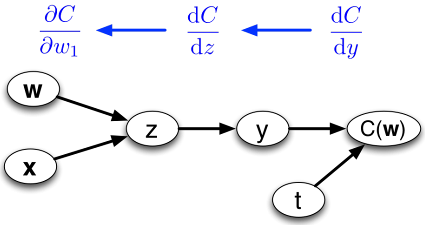


Note the following correspondences:

input units	$\Longleftrightarrow$	inputs $\mathbf{x}$
first layer weights and nonlinearities	$\Longleftrightarrow$	mapping ϕ
hidden unit activations	$\Longleftrightarrow$	feature representation $\phi(\mathbf{x})$
second layer weights	$\Longleftrightarrow$	weights $\mathbf{w}$
output unit	$\Longleftrightarrow$	prediction y

Overview

Now we want to learn the bottom layer weights as well, i.e. learn the features.
Recall from last lecture's example that we work backwards through the computation graph to compute derivatives.

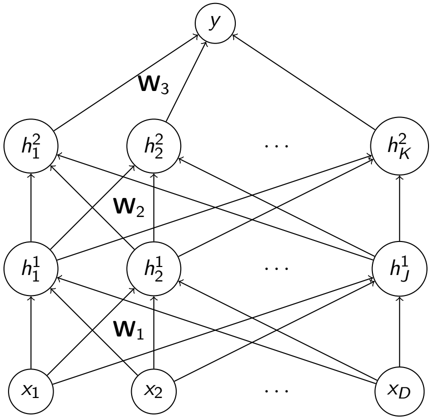


$\frac{\partial C}{\partial w_1}$
$\frac{\mathrm{d}C}{\mathrm{d}z}$
$\frac{\mathrm{d}C}{\mathrm{d}y}$
w
x
z
y
C(**w**)
t

Now we're just doing the same thing, but with a bigger computation graph!

An Example of a Neural Net

Network to classify handwritten digits.



An Example of a Neural Net

Input representation of a digit : 784 dimensional vector.

```
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 221.0 115.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  51.0 254.0 140.0   0.0   0.0   0.0   0.0  22.0 230.0 134.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 168.0 254.0 115.0   0.0   0.0   0.0   0.0  24.0 253.0 104.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  51.0 234.0 254.0  81.0   0.0   0.0   0.0   0.0  91.0 253.0 104.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  13.0 221.0 254.0 168.0   0.0   0.0   0.0   0.0   0.0 141.0 254.0 127.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 114.0 253.0 253.0  76.0   0.0   0.0   0.0   0.0   0.0 207.0 253.0  93.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  15.0 232.0 253.0 102.0   0.0   0.0   0.0   0.0   0.0   0.0 207.0 253.0  17.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 229.0 253.0 202.0  19.0   0.0   0.0   0.0   0.0   0.0  34.0 240.0 253.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   7.0 170.0 254.0 254.0  46.0   0.0   0.0   0.0   0.0   0.0   0.0  47.0 254.0 254.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0  24.0 253.0 254.0 253.0 234.0 163.0  47.0  47.0  26.0   0.0   0.0 130.0 253.0 253.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0  21.0 246.0 254.0 253.0 253.0 253.0 254.0 253.0 232.0 174.0 208.0 232.0 253.0 177.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0  84.0 161.0 211.0 219.0 219.0 254.0 253.0 253.0 253.0 254.0 253.0 244.0  69.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  93.0 142.0 142.0  93.0 170.0 254.0 230.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 183.0 253.0 213.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 170.0 253.0 137.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 220.0 253.0 137.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  60.0 255.0 254.0  80.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  93.0 254.0 253.0  46.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  51.0 254.0 215.0   9.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 106.0 106.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
```

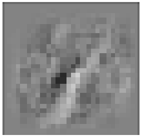
An Example of a Neural Net

Each first-layer hidden unit computes $\sigma(\mathbf{w}_i^T \mathbf{x})$

Here is one of the weight vectors (also called a feature).

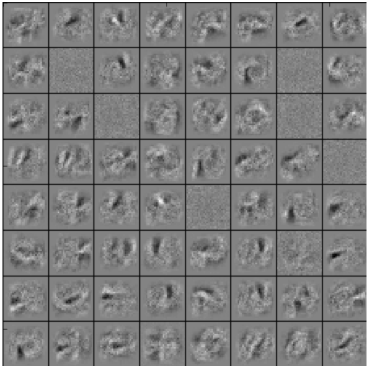
It's reshaped into an image, with gray = 0, white = +, black = -.

To compute $\mathbf{w}_i^T \mathbf{x}$, multiply the corresponding pixels, and sum the result.



An Example of a Neural Net

There are 256 first-level features total. Here are some of them.



An example of a Neural Net

Note: linear and nonlinear hidden units typically serve different purposes. We'll talk about this in more detail next week.

- Nonlinear hidden units allow the network to compute more complex functions (like in the digit classification example).

An example of a Neural Net

Note: linear and nonlinear hidden units typically serve different purposes. We'll talk about this in more detail next week.

- Nonlinear hidden units allow the network to compute more complex functions (like in the digit classification example).
- Linear hidden units can't increase a network's representational power, since the composition of linear functions is linear.

An example of a Neural Net

Note: linear and nonlinear hidden units typically serve different purposes. We'll talk about this in more detail next week.

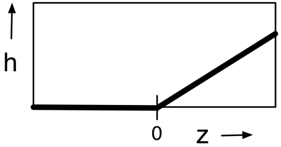
- Nonlinear hidden units allow the network to compute more complex functions (like in the digit classification example).
- Linear hidden units can't increase a network's representational power, since the composition of linear functions is linear.
    - Instead, they're typically used for dimensionality reduction, i.e. reducing the number of trainable parameters. We'll see an example next week when we discuss neural language models.

Question 1: Backprop Example

Now let's work through a backprop example with numbers.

The hidden layer consists of rectified linear units. As a refresher,

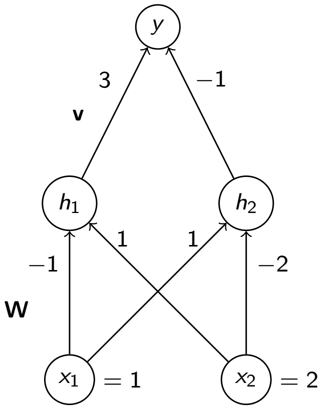
$$h_i = \max(z_i, 0)$$
$$\frac{\mathrm{d}h_i}{\mathrm{d}z_i} = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i \leq 0. \end{cases}$$



Question 1: Backprop Example

Data : $x = (1, 2)$, $t = 1$.
Parameters : $\mathbf{W}$: 2×2 matrix, $\mathbf{v}$: 2-dimensional vector.
Each hidden unit has the rectified linear non-linearity. $h = \max(0, z)$
Output unit is linear. Loss function: $C = \frac{1}{2}(y - t)^2$.
Compute the following -

- The input to each hidden unit. $z_i = \sum_{j=1}^{2} x_j W_{ij}$.
- Output from each hidden unit. $h_i = \max(0, z_i)$.
- Input to the output unit. $y = \sum_{i=1}^{2} v_i h_i$.
- Loss. $C = \frac{1}{2}(y - t)^2$.
- Derivative of loss w.r.t y : $\frac{\partial C}{\partial y} = y - t$.
- Derivative of loss w.r.t h_i : $\frac{\partial C}{\partial h_i} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial h_i}$.
- Derivative of loss w.r.t z_i : $\frac{\partial C}{\partial z_i} = \frac{\partial C}{\partial h_i}\frac{\partial h_i}{\partial z_i}$.
- Derivative of loss w.r.t x_j : $\frac{\partial C}{\partial x_j} = \sum_{i=1}^{2}\frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial x_j}$.
- Derivative of loss w.r.t W_{ij} : $\frac{\partial C}{\partial W_{ij}} = \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial W_{ij}}$.
- Derivative of loss w.r.t v_i : $\frac{\partial C}{\partial v_i} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial v_i}$.



Question 1: Backprop Example

Forward pass:

$$\begin{aligned} z_1 &= W_{11}x_1 + W_{12}x_2 \\ &= -1 \cdot 1 + 1 \cdot 2 = 1 \\ z_1 &= W_{21}x_1 + W_{12}x_2 \\ &= 1 \cdot 1 + -2 \cdot 2 = -3 \\ h_1 &= \max(z_1, 0) = 1 \\ h_2 &= \max(z_2, 0) = 0 \\ y &= v_1 h_1 + v_2 h_2 \\ &= 3 \cdot 1 + -1 \cdot 0 = 3 \end{aligned}$$

Loss and loss derivative:

$$\begin{aligned} C &= \frac{1}{2}(y - t)^2 = 2 \\ \frac{\partial C}{\partial y} &= y - t = 2 \end{aligned}$$

Activation derivatives:

$$\begin{aligned} \frac{\partial C}{\partial h_1} &= \frac{\partial C}{\partial y}\frac{\partial y}{\partial h_1} \\ &= \frac{\partial C}{\partial y} v_1 \\ &= 2 \cdot 3 = 6 \\ \frac{\partial C}{\partial h_2} &= \frac{\partial C}{\partial y}\frac{\partial y}{\partial h_2} \\ &= \frac{\partial C}{\partial y} v_2 \\ &= 2 \cdot -1 = -2 \\ \frac{\partial C}{\partial z_1} &= \frac{\partial C}{\partial h_1}\frac{\partial h_1}{\partial z_1} \\ &= 6 \cdot 1 = 6 \\ \frac{\partial C}{\partial z_2} &= \frac{\partial C}{\partial h_2}\frac{\partial h_2}{\partial z_2} \\ &= -2 \cdot 0 = 0 \end{aligned}$$

$$\begin{aligned} \frac{\partial C}{\partial x_1} &= \frac{\partial C}{\partial z_1}\frac{\partial z_1}{\partial x_1} + \frac{\partial C}{\partial z_2}\frac{\partial z_2}{\partial x_1} \\ &= \frac{\partial C}{\partial z_1} W_{11} + \frac{\partial C}{\partial z_2} W_{21} \\ &= 6 \cdot -1 + 0 \cdot 1 = -6 \\ \frac{\partial C}{\partial x_2} &= \frac{\partial C}{\partial z_1}\frac{\partial z_1}{\partial x_2} + \frac{\partial C}{\partial z_2}\frac{\partial z_2}{\partial x_2} \\ &= \frac{\partial C}{\partial z_1} W_{12} + \frac{\partial C}{\partial z_2} W_{22} \\ &= 6 \cdot 1 + 0 \cdot -2 = 6 \end{aligned}$$

Question 1: Backprop Example

Weight derivatives:

$$\begin{aligned}
\frac{\partial C}{\partial v_1} &= \frac{\partial C}{\partial y}\frac{\partial y}{\partial v_1} \\
&= \frac{\partial C}{\partial y} h_1 \\
&= 2 \cdot 1 = 2 \\
\frac{\partial C}{\partial v_2} &= \frac{\partial C}{\partial y}\frac{\partial y}{\partial v_2} \\
&= \frac{\partial C}{\partial y} h_2 \\
&= 2 \cdot 0 = 0 \\
\frac{\partial C}{\partial W_{11}} &= \frac{\partial C}{\partial z_1}\frac{\partial z_1}{\partial W_{11}} \\
&= \frac{\partial C}{\partial z_1} x_1 \\
&= 6 \cdot 1 = 6
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C}{\partial W_{12}} &= \frac{\partial C}{\partial z_1}\frac{\partial z_1}{\partial W_{12}} \\
&= \frac{\partial C}{\partial z_1} x_2 \\
&= 6 \cdot 2 = 12 \\
\frac{\partial C}{\partial W_{21}} &= \frac{\partial C}{\partial z_2}\frac{\partial z_2}{\partial W_{21}} \\
&= \frac{\partial C}{\partial z_2} x_1 \\
&= 0 \cdot 1 = 0 \\
\frac{\partial C}{\partial W_{22}} &= \frac{\partial C}{\partial z_2}\frac{\partial z_2}{\partial W_{22}} \\
&= \frac{\partial C}{\partial z_2} x_2 \\
&= 0 \cdot 2 = 0
\end{aligned}$$

Question 1: Backprop Example

You should go back through this example and think about which quantities were required to compute which other quantities. Some observations:

- You first do the feed-forward pass, then compute $\partial C/\partial y$, then backpropagate the derivatives.
- The update to the weight connecting unit A to unit B requires the activation of A and the backpropagated derivative for the input to B.
- We never actually used the value of the loss C for anything. But it's really cheap to compute during training, since the cost of computing activations is shared between the loss and gradient computations.
- We never used the input derivatives $\partial C/\partial x_j$ either. We'll see a pretty neat use for these when we talk about convolutional nets.

Some other things you might have noticed:

- A lot of the derivatives turn out to be 0. This "sparsity" property results from using linear rectification as the nonlinearity.
- Some of the derivatives are much larger than any of the weights, or activations, e.g. $\partial C/\partial W_{12} = 12$. This is a result of numbers getting multiplied together as we propagate the derivatives. This is a preview of the "exploding gradient" phenomenon we'll discuss later.

Question 2: vectorized computations

On the left are the computations performed by the network. Write them in terms of matrix and vector operations. Let $\sigma(\mathbf{v})$ denote the logistic sigmoid function applied elementwise to a vector $\mathbf{v}$. Let $\mathbf{W}$ be a matrix where the (i, j) entry is the weight from visible unit j to hidden unit i.

$$z_i = \sum_j w_{ij} x_j \qquad\qquad \mathbf{z} =$$

$$h_i = \sigma(w_i) \qquad\qquad \mathbf{h} =$$

$$y = \sum_i v_i h_i \qquad\qquad \mathbf{y} =$$

Question 2: vectorized computations

On the left are the computations performed by a network. Write them in terms of matrix and vector operations. Let $\sigma(\mathbf{v})$ denote the logistic sigmoid function applied elementwise to a vector $\mathbf{v}$. Let $\mathbf{W}$ be a matrix where the (i, j) entry is the weight from visible unit j to hidden unit i.

$$z_i = \sum_j w_{ij} x_j \qquad \mathbf{z} = \mathbf{W}\mathbf{x}$$

$$h_i = \sigma(z_i) \qquad \mathbf{h} = \sigma(\mathbf{z})$$

$$y = \sum_i v_i h_i \qquad y = \mathbf{v}^T \mathbf{h}$$

Question 2: vectorized computations

Now, on the left we have the gradients computed during backpropagation. Write them in terms of matrix and vector operations. Recall that the (i, j) entry of $\mathbf{W}$ is the weight connecting visible unit j to hidden unit i. Let $\sigma'(\mathbf{v})$ denote the derivative of the sigmoid function applied elementwise to $\mathbf{v}$. Denote elementwise multiplication by $\circ$.

$$
\begin{aligned}
\frac{\partial C}{\partial v_i} &= h_i \frac{\partial C}{\partial y} \\
\frac{\partial C}{\partial h_i} &= v_i \frac{\partial C}{\partial y} \\
\frac{\partial C}{\partial z_i} &= \sigma'(z_i) \frac{\partial C}{\partial h_i} \\
\frac{\partial C}{\partial w_{ij}} &= x_j \frac{\partial C}{\partial z_i}
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\mathbf{v}} C &= \\
\nabla_{\mathbf{h}} C &= \\
\nabla_{\mathbf{z}} C &= \\
\nabla_{\mathbf{W}} C &=
\end{aligned}
$$

Question 2: vectorized computations

Now, on the left we have the gradients computed during backpropagation. Write them in terms of matrix and vector operations. Recall that the (i, j) entry of $\mathbf{W}$ is the weight connecting visible unit j to hidden unit i. Let $\sigma'(\mathbf{v})$ denote the derivative of the sigmoid function applied elementwise to $\mathbf{v}$. Denote elementwise multiplication by $\circ$.

$$\frac{\partial C}{\partial v_i} = h_i \frac{\partial C}{\partial y}$$
$$\frac{\partial C}{\partial h_i} = v_i \frac{\partial C}{\partial y}$$
$$\frac{\partial C}{\partial z_i} = \sigma'(z_i) \frac{\partial C}{\partial h_i}$$
$$\frac{\partial C}{\partial w_{ij}} = x_j \frac{\partial C}{\partial z_i}$$

$$\nabla_{\mathbf{v}} C = \frac{\partial C}{\partial y} \mathbf{h}$$
$$\nabla_{\mathbf{h}} C = \frac{\partial C}{\partial y} \mathbf{v}$$
$$\nabla_{\mathbf{z}} C = \sigma'(\mathbf{z}) \circ \nabla_{\mathbf{h}} C$$
$$\nabla_{\mathbf{W}} C = (\nabla_{\mathbf{z}} C) \mathbf{x}^T$$

Question 2: vectorized computations

Note: the vectorized operations we just derived are better than working with scalars, but they still don't give an efficient implementation.

- It considers only one training case at a time, so if we implemented it in Python, we would need to do a for loop over training cases!
- In Assignment 1, we'll use matrices to represent the activations of all the units in a layer for an entire batch of training cases. You'll derive the update rules in matrix form. You should use the same strategies we used in solving this problem.