An Exercise in Backpropagation

Matthew Jin

Spring 2025

What, when, and why backpropagation?

What: Backpropagation is one method of calculating gradients of the output of a chain of operations with respect to each component in the chain.

When: Used in anything and everything neural networks related.

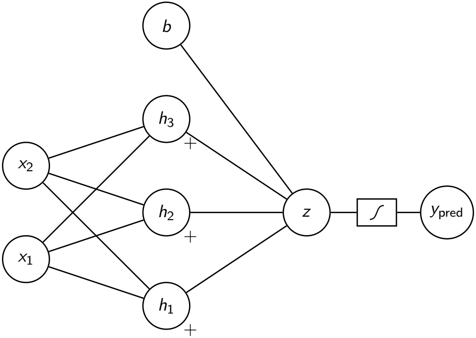
Why: Because some very smart people have built systems that allow computers to automatically compute gradients via backpropagation.

Definitions

We'll be exploring a simple neural network for the binary (0 or 1) classification problem.

- Data: $(\mathbf{X}, \mathbf{y})$ are the dataset and corresponding labels
  - $\mathbf{X} \in \mathbb{R}^{N \times 2}, \mathbf{y} \in \{0, 1\}^N$
- Model: $f_\theta(\mathbf{x}) : \mathbb{R}^2 \to [0, 1]$ (probability $\mathbf{x}$ is in class 1)
  - θ represents all the parameters we want to optimize!
- Activation functions: ReLU (x_+), Sigmoid ($\sigma(x)$)
  - $x_+ = \max(0, x)$
  - $\sigma(x) = \frac{1}{1+e^{-x}}$
- Binary Cross Entropy Loss:
  - $\ell_i(p_i, y_i) = -(y_i \cdot \log p_i + (1 - y_i) \cdot \log(1 - p_i))$
  - $\ell(\mathbf{p}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(p_i, y_i)$

The Network



$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\max(0, \mathbf{x}\mathbf{w}_1)\mathbf{w}_2 + b}_{\mathbf{z}})$$

We want to optimize the network's parameters θ which consist of $\mathbf{w}_1 \in \mathbb{R}^{2\times 3}$, $\mathbf{w}_2 \in \mathbb{R}^{3\times 1}$, and $b \in \mathbb{R}^1$.

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\max(0, \mathbf{x}\mathbf{w}_1)\mathbf{w}_2 + b}_{\mathbf{z}})$$

We want to calculate the gradient of the binary cross entropy loss applied to the output of the network

$$\ell(\underbrace{f_\theta(\mathbf{X})}_{y_{\text{pred}}}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(f_\theta(\mathbf{x}_i), y_i)$$

$$= \frac{1}{N}\sum_{i=1}^{N} -\left(y_i \cdot \log f_\theta(\mathbf{x}_i) + (1 - y_i) \cdot \log(1 - f_\theta(\mathbf{x}_i))\right)$$

with respect to $\mathbf{w}_1$, $\mathbf{w}_2$, and b.

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\max(0, \mathbf{x}\mathbf{w}_1)\mathbf{w}_2 + b}_{\mathbf{z}})$$

From the chain rule, we have

$$\frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{w}_2} = \frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{y}_{\text{pred}}} \frac{\partial\, \mathbf{y}_{\text{pred}}}{\partial\, \mathbf{z}} \frac{\partial\, \mathbf{z}}{\partial\, \mathbf{w}_2}$$

$$\frac{\partial\, \ell(\,\cdot\,)}{\partial\, b} = \frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{y}_{\text{pred}}} \frac{\partial\, \mathbf{y}_{\text{pred}}}{\partial\, \mathbf{z}} \frac{\partial\, \mathbf{z}}{\partial\, b}$$

$$\frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{w}_1} = \frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{y}_{\text{pred}}} \frac{\partial\, \mathbf{y}_{\text{pred}}}{\partial\, \mathbf{z}} \frac{\partial\, \mathbf{z}}{\partial\, \mathbf{h}} \frac{\partial\, \mathbf{h}}{\partial\, \mathbf{x}\mathbf{w}_1} \frac{\partial\, \mathbf{x}\mathbf{w}_1}{\partial\, \mathbf{w}_1}$$

where $\mathbf{h} = \max(0, \mathbf{x}\mathbf{w}_1)$.

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\overbrace{\max(0, \mathbf{x}\mathbf{w}_1)}^{\mathbf{h}} \mathbf{w}_2 + b}_{\mathbf{z}})$$

Note that $\ell(f_\theta(\mathbf{X}), \mathbf{y})$ is a scalar value while $\mathbf{z} \in \mathbb{R}^N$, so how is the gradient calculated?

$$\frac{\partial\, \ell(\,\cdot\,)}{\partial\, \mathbf{y}_{\text{pred}}} = \begin{bmatrix} \frac{\partial\, \ell(\cdot)}{\partial\, y_{\text{pred}\,1}} & \cdots & \frac{\partial\, \ell(\cdot)}{\partial\, y_{\text{pred}\,n}} \end{bmatrix}^\top$$

What about the derivative of $\mathbf{y}_{\text{pred}}$ with respect to $\mathbf{z}$? Both are vectors of size N. Since $\mathbf{y}_{\text{pred}} = \sigma(\mathbf{z})$ is an element-wise operation,

$$\frac{\partial\, \mathbf{y}_{\text{pred}}}{\partial\, \mathbf{z}} = \sigma(\mathbf{z}) \cdot (1 - \sigma(\mathbf{z})) = \mathbf{y}_{\text{pred}} \cdot (1 - \mathbf{y}_{\text{pred}})$$

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\overbrace{\max(0, \mathbf{x}\mathbf{w}_1)}^{\mathbf{h}} \mathbf{w}_2 + b}_{\mathbf{z}})$$

Next, we want to calculate $\frac{\partial \mathbf{z}}{\partial \mathbf{h}}$. Here $\mathbf{z} \in \mathbb{R}^N$ and $\mathbf{h} \in \mathbb{R}^{N \times 3}$. We have $\mathbf{z} = \mathbf{h}\mathbf{w}_2 + b$ which looks something like

$$\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ \vdots & \vdots & \vdots \\ h_{N1} & h_{N2} & h_{N3} \end{bmatrix} \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ w_{2,3} \end{bmatrix}$$

Since $z_i = h_{i1}w_{2,1} + h_{i2}w_{2,2} + h_{i3}w_{2,3}$, the gradient of $\mathbf{z}$ w.r.t. each row of $\mathbf{h}$ is $\mathbf{w}_2^\top$.

On the other hand, the gradient of $\mathbf{z}$ w.r.t. w_{2j} depends on $h_{1j}, \ldots, h_{Nj}$.

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\overbrace{\max(0, \mathbf{x}\mathbf{w}_1)}^{\mathbf{h}} \mathbf{w}_2 + b}_{\mathbf{z}})$$

What about the gradient for the bias b?
b is automatically broadcasted in the equation but can be written as $\mathbf{z} = \mathbf{h}\mathbf{w}_2 + \mathbf{1}^\top b$ where $\mathbf{1} \in \mathbb{R}^N$. Thus, the gradient is

$$\frac{\partial \, \mathbf{z}}{\partial \, b} = \mathbf{1}^\top$$

Gradient Calculations

$$f_\theta(\mathbf{x}) = \sigma(\underbrace{\overbrace{\max(0, \mathbf{x}\mathbf{w}_1)}^{\mathbf{h}} \mathbf{w}_2 + b}_{\mathbf{z}})$$

To calculate the gradient of $\mathbf{w}_1$, we need $\frac{\partial \mathbf{h}}{\partial \mathbf{x}\mathbf{w}_1}$ and $\frac{\partial \mathbf{x}\mathbf{w}_1}{\partial \mathbf{w}_1}$.

For $\frac{\partial \mathbf{h}}{\partial \mathbf{x}\mathbf{w}_1}$, we have

$$\frac{\partial \max(0, \mathbf{x}\mathbf{w}_1)_{ij}}{\partial \mathbf{x}\mathbf{w}_1} = \begin{cases} 1 & \text{if } (\mathbf{x}\mathbf{w}_1)_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

For $\frac{\partial \mathbf{x}\mathbf{w}_1}{\partial \mathbf{w}_1}$, we have

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ \vdots & \vdots & \vdots \\ h_{N1} & h_{N2} & h_{N3} \end{bmatrix} = \left(\begin{bmatrix} x_{11} & x_{12} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{bmatrix} \begin{bmatrix} w_{1,11} & w_{1,12} & w_{1,13} \\ w_{1,21} & w_{1,22} & w_{1,23} \end{bmatrix} \right)_+$$

Similar to $\mathbf{w}_2$, the gradient for each column of $\mathbf{w}_1$ depends on $\mathbf{x}$.

Additional Resources

- CS231N page on backpropagation
- Justin Johnson's notes on matrix gradient calculations
- Erik Learned-Miller's notes on vector/matrix/tensor derivatives
- The Matrix Cookbook

- Gregory Gundersen's notes on the reparameterization trick

Appendix

Batched Linear Gradients

Consider the equation $Y = XW$ where $Y \in \mathbb{R}^{N \times D_y}$, $X \in \mathbb{R}^{N \times D_x}$, and $W \in \mathbb{R}^{D_x \times D_y}$. On the forward pass, the matrix multiplication can be rewritten as

$$Y_{ij} = \sum_{k=1}^{D_x} X_{ik} W_{kj}$$

Given this formula, how do we calculate the gradients $\frac{\partial Y}{\partial W}$ and $\frac{\partial Y}{\partial X}$? What if we add a loss function $\ell(Y) : \mathbb{R}^{N \times D_y} \to \mathbb{R}$? How do we calculate $\frac{\partial \ell(Y)}{\partial W}$ and $\frac{\partial \ell(Y)}{\partial X}$?

$$Y_{ij} = \sum_{k=1}^{D_x} X_{ik} W_{kj}$$

First, let us focus on the gradient with respect to one element in the weight matrix $\frac{\partial Y}{\partial W_{kj}}$ which should have the same shape as $Y \in \mathbb{R}^{N \times D_y}$.

$$\frac{\partial Y}{\partial W_{kj}} = \begin{bmatrix} 0 & \cdots & \frac{\partial Y_{1j}}{\partial W_{kj}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial Y_{Nj}}{\partial W_{kj}} & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \cdots & X_{1k} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & X_{Nk} & \cdots & 0 \end{bmatrix}$$

Appendix

Batched Linear Gradients

$$Y_{ij} = \sum_{k=1}^{D_x} X_{ik} W_{kj}$$

The loss gradient $\frac{\partial\,\ell(Y)}{\partial\,Y}$ also has shape $\mathbb{R}^{N \times D_y}$ since $\ell(Y)$ is a scalar and can be written as

$$\frac{\partial\,\ell(Y)}{\partial\,Y} = \begin{bmatrix} \frac{\partial\,\ell(Y)}{\partial\,Y_{11}} & \cdots & \frac{\partial\,\ell(Y)}{\partial\,Y_{1D_y}} \\ \vdots & \ddots & \vdots \\ \frac{\partial\,\ell(Y)}{\partial\,Y_{N1}} & \cdots & \frac{\partial\,\ell(Y)}{\partial\,Y_{ND_y}} \end{bmatrix}$$

For a specific W_{kj}, we have

$$\frac{\partial\,\ell(Y)}{\partial\,W_{kj}} = \sum_{ij} \frac{\partial\,\ell(Y)}{\partial\,Y_{ij}} \cdot \frac{\partial\,Y_{ij}}{\partial\,W_{kj}} = \sum_{i=1}^{N} X_{ik} \cdot \frac{\partial\,\ell(Y)}{\partial\,Y_{ij}} = \left(X^\top \frac{\partial\,\ell(Y)}{\partial\,Y} \right)_{kj}$$

Thus, $\frac{\partial\,\ell(Y)}{\partial\,W} = X^\top \frac{\partial\,\ell(Y)}{\partial\,Y}$. Intuitively, this makes sense since the gradients of each weight should be aggregated across the batch of inputs.

Appendix

Batched Linear Gradients

$$Y_{ij} = \sum_{k=1}^{D_x} X_{ik} W_{kj}$$

Similarly, for $\frac{\partial Y}{\partial X_{ik}} \in \mathbb{R}^{N \times D_y}$,

$$\frac{\partial Y}{\partial X_{ik}} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \frac{\partial Y_{i1}}{\partial X_{ik}} & \cdots & \frac{\partial Y_{iD_y}}{\partial X_{ik}} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ W_{k1} & \cdots & W_{kD_y} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$$

which gives us

$$\frac{\partial \ell(Y)}{\partial X_{ik}} = \sum_{ij} \frac{\partial \ell(Y)}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial X_{ik}} = \sum_{j=1}^{D_y} W_{kj} \cdot \frac{\partial \ell(Y)}{\partial Y_{ij}} = \left(\frac{\partial \ell(Y)}{\partial Y} W^\top \right)_{ik}$$

Thus, $\frac{\partial \ell(Y)}{\partial X} = \frac{\partial \ell(Y)}{\partial Y} W^\top$. This indicates that the gradient of the k-th component of a single input X_i depends on the corresponding row W_k in the weight matrix.