Scribe: Andrew Hanson

Optimization is Critical for Successful NNs

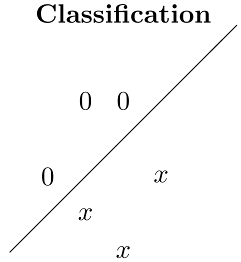
Being able to take derivatives and find gradients is necessary for optimization.
One of the most important algorithms for training NNs is **backpropagation** (BP).

BP = stocastic gradient descent (SGD) + chain rule

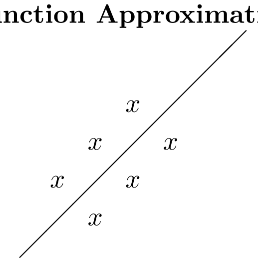
Perceptron Algorithm

update rule: $w = w + x \quad \text{or} \quad w = w - x$

Classification



Function Approximation



$y = wx - b$

Loss Function

The loss function should give the error of the perceptron. Here we are trying to fit a line to the data.

$$\text{example loss function: } L(w,b) = \sum_{i=0}^{N} (y_i - wx_i + b)^2 \quad \text{(want to minimize)}$$

Minimizing the loss gives us a good fit to the training data.

Gradients for Optimization

Setting the derivatives or the gradient to zero helps us find the minimum of the loss.

$$\frac{\partial L}{\partial w} = \sum_{i=0}^{N} 2(y_i - wx_i - b)(-x_i) = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=0}^{N} 2(y_i - wx_i - b)(-1) = 0$$

$$\nabla L(w,b) = \begin{bmatrix} \frac{\partial L}{\partial w} \\ \frac{\partial L}{\partial b} \end{bmatrix} = 0$$

Finding a Good Loss Function is Important

Perceptron Example

Here y is the label or target output. (supervised learning)

$$(x, y) : y \in \{-1, +1\}$$

$$L(w) = \frac{1}{2}(w^T x - y)^2 \qquad \frac{\partial L}{\partial w} = (w^T x - y)(x)$$

Gradient Descent

$$w_{\text{new}} = w_{\text{old}} - \eta\frac{\partial L}{\partial w} = w - \eta(w^T x - y)x \quad (\text{where } \eta = \text{learning rate})$$

Part of this expression is similar to the error used for the perceptron learning rule assignment.

$$\Delta w = -\eta(\text{error})x$$

The idea is that by choosing a good loss function, you can find a good set of weights w by iteratively working to minimize that loss function.

$$\min(L(w))$$

The loss function needs to capture the error or difference between the current w and a better w.

Matrix Derivatives

Scalars:

$f : \mathbb{R} \to \mathbb{R} \quad \frac{df}{dx}$ (familiar)

Vector:

$$f : \mathbb{R}^d \to \mathbb{R}, \quad \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{bmatrix}$$

$d(f) : \mathbb{R}^d \to \mathbb{R}^d$ The derivative is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$.

Example 1:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \in \mathbb{R}^3, \quad f : \mathbb{R}^3 \to \mathbb{R}$$

$$f(W) = 3w_1 w_2 + w_3$$

$$d(f) : \mathbb{R}^3 \to \mathbb{R}^3$$

$$\frac{\partial f}{\partial w_1} = 3w_2, \quad \frac{\partial f}{\partial w_2} = 3w_1, \quad \frac{\partial f}{\partial w_3} = 1$$

$$\nabla f(W) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \frac{\partial f}{\partial w_3} \end{bmatrix} = \begin{bmatrix} 3w_2 \\ 3w_1 \\ 1 \end{bmatrix}$$

Example 2:

$$w \in \mathbb{R}^d, \quad f(w) = w^T x, \quad f : \mathbb{R}^d \to \mathbb{R}$$

(If we proceed like a scalar derivative.)

$$\frac{df}{dw} = x \text{ (scaler)}$$

Or we can decompose the dot product:

$$f(w) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

$$\frac{\partial f}{\partial w_1} = x_1, \quad \frac{\partial f}{\partial w_2} = x_2, \quad \ldots$$

$$\frac{\partial f}{\partial w} = \nabla f = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = x \text{ (vector)}$$

Example 3:

$$f(w) = \|w\|^2, \quad w \in \mathbb{R}^d, \quad f : \mathbb{R}^d \to \mathbb{R}$$

$$= w^T w$$

$$= \begin{bmatrix} w_1, w_2, \ldots, w_d \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$= w_1^2 + w_2^2 + \cdots + w_d^2$$

Partial Derivative:

$$\frac{\partial f}{\partial w_i} = 2w_i$$

Gradient:

$$\nabla f = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_d \end{bmatrix} = 2w$$

Taylor Series

scaler case:

$$f(x+\Delta x) \approx f(x) + \Delta x f'(x) + \tfrac{1}{2}(\Delta x)^2 f''(x)$$

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} = f'(x)$$

matrix/vector case:

$$x \in \mathbb{R}^d, \quad f : \mathbb{R}^d \to \mathbb{R}, \quad \Delta x \in \mathbb{R}^d$$

$$f(x+\Delta x) \approx f(x) + \Delta x \cdot \nabla f$$

To produce a scalar from two vectors, we can use the **dot product**, which is equivalent to transposing the first vector and then applying matrix multiplication if they are both column vectors:

$$\begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \dots & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = \mathbb{R}$$

$$f(x+\Delta x) \approx f(x) + \Delta x^T \nabla f + \dots + \tfrac{1}{2} \Delta x^T \Delta x f''(x)$$

Hessian Matrix and Directional Derivatives

Gradient and Hessian:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} \quad \text{(2x2)}$$

The diagonal terms are equal, so this is a symmetric matrix. This is called the **Hessian** $H \in \mathbb{R}^{d \times d}$.
—

Second-order Taylor Expansion:

$$f(x+\Delta x) \approx f(x) + (\Delta x)^T \nabla f + \tfrac{1}{2}(\Delta x)^T H (\Delta x)$$

This quadratic form is written as:

$$\text{Quadratic form: } x^T A x$$

—

Directional Derivative:

$$x \in \mathbb{R}^d$$

$$\lim_{\alpha \to 0} \frac{f(x+\alpha d) - f(x)}{\alpha} = f'_d(x) = \langle d, \nabla f(x) \rangle$$

Where: - d is the direction vector. - $\langle d, \nabla f(x) \rangle$ is the dot product of d and the gradient $\nabla f(x)$.
—

Directional Derivative Examples

Example 1:

$f(x) = W^T x$

$$\lim_{\alpha \to 0} \frac{W^T(x + \alpha d) - W^T x}{\alpha} = \lim_{\alpha \to 0} \frac{W^T d\alpha}{\alpha} = W^T d = \langle d, W \rangle$$

$\nabla f(x) = W$

—

Example 2: Quadratic Form

$f(x) = \frac{1}{2} x^T A x, \quad A = A^T, \quad x \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d \times d}$

$$\lim_{\alpha \to 0} \frac{f(x + \alpha d) - f(x)}{\alpha} = \lim_{\alpha \to 0} \frac{\frac{1}{2}(x + \alpha d)^T A (x + \alpha d) - \frac{1}{2} x^T A x}{\alpha}$$

Expanding the quadratic form:

$$= \lim_{\alpha \to 0} \frac{1}{2\alpha} \left(d^T A x + \alpha x^T A d + \alpha^2 d^T A d \right)$$

$$= \lim_{\alpha \to 0} \tfrac{1}{2} \left(\langle d, Ax \rangle + \langle Ax, d \rangle + \alpha \langle d, Ad \rangle \right)$$

$$= \tfrac{1}{2} (\langle d, Ax \rangle + \langle Ax, d \rangle)$$

$$= \tfrac{2}{2} \langle d, Ax \rangle = \langle d, Ax \rangle$$

Gradient of the Quadratic Form:

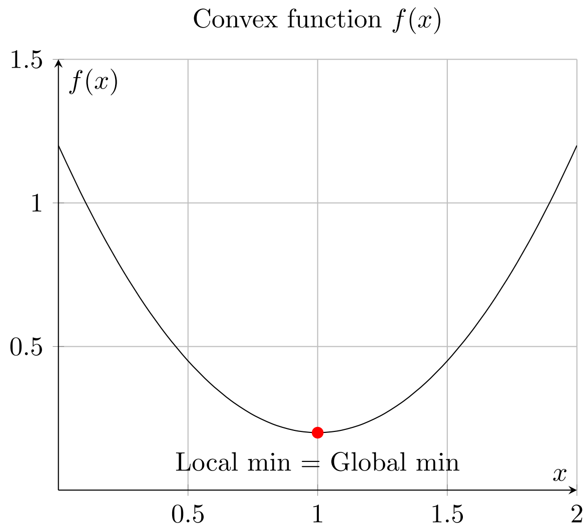
$\nabla f = Ax$

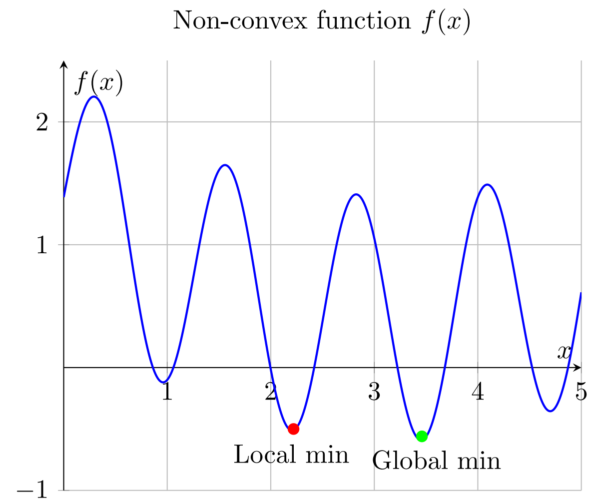
Convexity

Convex Function Illustration:

A convex function has the property that its local minimum is also a global minimum.

Convex vs. Non-Convex Functions





Local Minimum Criterion:

A local minimum x satisfies:

$$f(x) \leq f(x + \Delta x)$$

Taylor Series and Local Minima Criterion

First-order Approximation:

$$f(x + \Delta x) \approx f(x) + (\Delta x)^T \nabla f(x)$$

Local Minima Condition:

$$f(x) \leq f(x) + (\Delta x)^T \nabla f(x)$$

$$0 \leq (\Delta x)^T \nabla f(x)$$

$$\boxed{(\Delta x)^T \nabla f(x) \geq 0}$$

Choosing a Direction:

For $x \in \mathbb{R}^d$, choose $\Delta x = -\nabla f(x)$:

$$-(\Delta x)^T \nabla f \geq 0$$

$$\|\nabla f\|^2 \leq 0$$

Since the norm cannot be negative:

$$\boxed{\nabla f(x) = 0}$$

(Local minimum criterion)

Second-Order Condition and Gradient Descent

Second-Order Taylor Approximation:

$$f(x + \Delta x) \approx f(x) + \frac{1}{2}(\Delta x)^T H(\Delta x) \geq f(x)$$

$$\frac{1}{2}(\Delta x)^T H(\Delta x) \geq 0$$

Since we know H is symmetric, for this condition to hold for all Δx, H must be **positive semi-definite** (PSD).

$$\boxed{H \geq 0}$$

Hessian Matrix and Positive Semi-Definiteness:

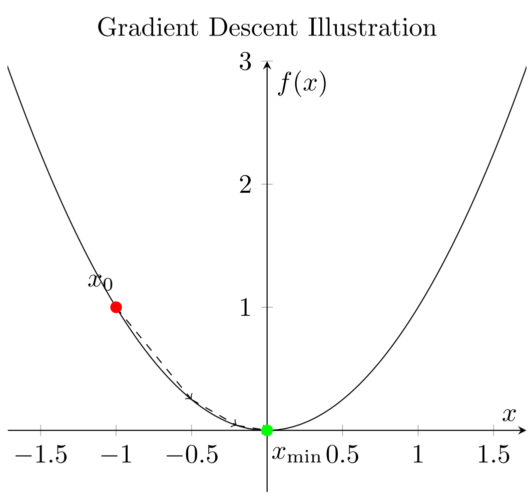
$$\frac{d^2 f}{dx^2} = H \geq 0$$

The eigen values of H are all greater than or equal to zero. So, for the eigen matrix of H...

$$\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}, \quad \text{all } \lambda \geq 0$$

Gradient Descent:

Graphical Representation:



Update Rule

$$x_{\text{new}} = x_{\text{old}} - \eta \nabla f(x_{\text{old}})$$

$$\boxed{w_{\text{new}} = w_{\text{old}} - \eta \nabla f(w_{\text{old}})}$$

Where: - η is the **learning rate**. - ∇f is the **gradient**, which points in the direction of the steepest ascent. - **Gradient descent** moves in the opposite direction to minimize the function.

Gradient Descent and Directional Derivative

From the Update Rule:

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

$$\Delta x = -\eta \nabla f(x_t)$$

Continuous Form Approximation:

$$\frac{\Delta x}{\Delta t} \approx \frac{dx}{dt} = -\eta \nabla f(x_t)$$

(Exponential decay behavior)

Directional Derivative:

$$\lim_{\alpha \to 0} \frac{f(x + \alpha d) - f(x)}{\alpha} = \langle d, \nabla f(x) \rangle$$

(We want: $f(x + \alpha d) \ll f(x)$)
(set: $d = -\nabla f(x)$)

Inner Product of the Gradient with Itself:

$$\langle -\nabla f(x), \nabla f(x) \rangle = -\|\nabla f(x)\|^2$$

Newton's Method (Second Order Optimization)

Gradient Descent Step:

$$x_{t+1} = x_t + \alpha d$$

$$= x_t + \alpha \nabla f(x_t) \quad \text{(learning rate)}$$

Newton's Method (Using Hessian)

To apply Newton's method, we need both the gradient and the Hessian matrix.

$$f(x + \Delta x) = f(x) + (\Delta x)^T \nabla f + \tfrac{1}{2} (\Delta x)^T H (\Delta x)$$

Finding the Minimum:

Taking the gradient on both sides:

$$\nabla f(x + \Delta x) = \nabla f(x) + H \Delta x$$

At the minimum, the gradient must be zero:

$$\nabla f(x + \Delta x) = 0$$

$$0 = \nabla f(x) + H \Delta x = \nabla f(x) + H(x_{k+1} - x_k)$$

Newton's Update Rule:

$$H(x_{k+1} - x_k) = -\nabla f(x_k)$$

$$x_{k+1} - x_k = -H^{-1}(\nabla f(x_k))$$

$$\boxed{x_{k+1} = x_k - H^{-1}(\nabla f(x_k))}$$

Newton's Method: Pros and Cons

Newton's Method is computationally expensive, but it converges faster (quadratic convergence).