Neural Networks: IV. Optimization (Part 3)

Jan Bauer

jan.bauer@dhbw-mannheim.de

16.05.19

Optimization

- Find the W that minimizes the loss, i.e. $W^* = \arg\min_W L$
  - Problem: Difficult. We only focus on the basics
- School: Gradient $\overset{!}{=} 0$ & check Hessian $\mathcal{H}$
  - **Example**: $0 = \arg\min x^2$
  - Problem: Machine power intense

Recap

Find minima and maxima (recipe)

1. $\nabla_f(x) \stackrel{!}{=} 0$
2. $\mathcal{H}_f(x) \begin{cases} > 0 & \text{minimum} \\ < 0 & \text{maximum} \\ \text{else} & \text{saddlepoint} \end{cases}$

Recap: Positive/negative definite

- A $m \times m$ matrix M is positive definite if the leading principal minors are positive. We write $M > 0$.
- A $m \times m$ matrix M is negative definite if the leading principal minors are negative. We write $M < 0$.
- Not relevant for this course, but for completion:
  - positive semi-definite if "≥ 0". We write $M \geq 0$.
  - negative semi-definite if "≤ 0". We write $M \leq 0$.

Recap: Principle Minors (Example)

Let $M = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{pmatrix}$. The principal minors are then given by

- $M_{1,1} = \det(m_{1,1}) = m_{1,1}$
- $M_{2,2} = \det \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = m_{1,1} \cdot m_{2,2} - m_{1,2} \cdot m_{2,1}$
- $M_{3,3} = \det(M)$

Recap: Gradient & Hessian

Let $f : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto f(x)$. Then...

- $\nabla_f(x) \equiv \dfrac{\partial f}{\partial x} \equiv \sum_i \dfrac{\partial f}{\partial x_i} \mathrm{e}_i = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} & \cdots & \dfrac{\partial f}{\partial x_n} \end{pmatrix}'$
- $\mathcal{H}_f(x) \equiv \begin{pmatrix} \dfrac{\partial f}{\partial^2 x_1} & \dfrac{\partial f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial f}{\partial x_1 \partial x_n} \\ \dfrac{\partial f}{\partial x_2 \partial x_1} & \dfrac{\partial f}{\partial^2 x_2} & \cdots & \dfrac{\partial f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f}{\partial x_n \partial x_1} & \dfrac{\partial f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial f}{\partial^2 x_n} \end{pmatrix}$

Example

Find the extreme values of

- $h_1 : \mathbb{R} \to \mathbb{R}, x \mapsto x^2$
  $\rightsquigarrow 0$
- $h_2 : \mathbb{R}^2 \to \mathbb{R}, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto x^2 + y^2$
  $\rightsquigarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Find the extreme values of

- $f : \mathbb{R}^2 \to \mathbb{R},\ \begin{pmatrix} x \\ y \end{pmatrix} \mapsto 2xy - 5x^2 + 4x - 2y - 4$
- $g : \mathbb{R}^2 \to \mathbb{R},\ \begin{pmatrix} x \\ y \end{pmatrix} \mapsto e^{xy+x^2+y^2}$ (without checking Hessian)

f and g plot

Gradient Descent

Remember: Machine Power intense

- MNIST: Each image is represented by a 784×1 vector
- $\mathcal{H}$ is 784×784
- intense, despite symmetry of $\mathcal{H}$ and using mini-batch

We will "follow the slope" instead

- Recap: $\nabla_f(x)$ points towards the direction of greatest increase
- Example: $f(x) = x^2$

Gradient Descent: Algorithm

- $L(W) = \frac{1}{D} \sum_d L_d(y_{[d]}, X, W)$
- $\nabla_L(W) = \frac{1}{D} \sum_d \nabla_{L_d}(W)(y_{[d]}, X, W)$
- $W_{t+1} = W_t - \eta \cdot \nabla_W L(W_t) \equiv W_t + \Delta W_t$
  - η $(= \eta_t)$ is the learning rate
  - W_0 is the initial value

Gradient Descent: Coding

Find the extreme values of

- $f(x, y) = x^2 + y^2$
- Starting values: $x_0 = y_0 = 3$
- $\eta = 0.01$

Note: You might consider a stopping rule

Code

Plot & Gradient Descent

Gradient Descent: Coding

Find the extreme values of

- $f(x, y) = x^2 + y^2$
- Starting values: $x_0 = y_0 = 3$
- $\eta = 0.01$

Note: You might consider a stopping rule

Gradient Descent

Find the extreme values of

- Learning rate: Perhaps later
- Initial values: Roughly
- Gradient: Numerically vs. analytically
  - Numerically: Slow, approximate & easy
  - Analytically: Fast, exact & difficult

Backpropagation "Algorithm"

repeat:

1. Feed the NN (forward propagation)
2. Derive the gradient by backprop ("Backward propagation of errors")
3. Optimize

Example 1 & 2

Note: For the sigmoid function $\sigma(x)$, it holds that

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x))$$