UCL AI SOC

The Future is Calling

UCL AI Society – Tutorials for ML
Live Session 2

Building Blocks of ML

By Jianqiao Mao,

November 19th, 2020

CONTENTS

01

Optimisation

How to maximise and minimise

Loss function

- Way to assess the (supposed) fitness of algorithm with a number

$$L_{\mathbf{w}} = \sum_{i=1}^{D} (y_i - \mathbf{x}_i^T \mathbf{w})^2, \quad L_{\mathbf{w}_1} < L_{\mathbf{w}_2} \implies \mathbf{w}_1 \text{ better fit than } \mathbf{w}_2$$

- Often **differentiable** to allow minimisation by gradient methods

$$\mathbf{w}_{\text{best}} = \text{argmin}_{\mathbf{w}} L_{\mathbf{w}} \implies \frac{\partial}{\partial \mathbf{w}} L_{\mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_{\text{best}}} = 0$$

Derivative

- The derivative of a function is the rate of change of that function

$$f(x_0 + \Delta x) \approx f(x_0) + \Delta x \left. \frac{\partial}{\partial x} f(x) \right|_{x=x_0}$$

- Going against the derivative decreases the function (if possible)

$$f\left(x_0 - \epsilon \left. \frac{\partial}{\partial x} f(x) \right|_{x=x_0} \right) < f(x_0) \quad \text{for a sufficiently small } \epsilon$$

Gradient Descent

- Repeatedly applying the same update

$$x_2 = x_1 - \epsilon \frac{\partial}{\partial x} f(x) \Big|_{x=x_1}, \quad x_3 = x_2 - \epsilon \frac{\partial}{\partial x} f(x) \Big|_{x=x_2}$$

- If we apply at critical point, the update doesn’t change

$$x_{T+1} = x_T - \epsilon \frac{\partial}{\partial x} f(x) \Big|_{x=x_T} = x_T - 0 = x_T$$

Closed Form

- Sometimes we can solve directly

$$L_w = w(1-w) \implies \frac{\partial}{\partial w} L_w = 1 - 2w \implies w_{\text{best}} = \frac{1}{2}$$

- Performance of direct approach vs iterative fitting

$$\mathbf{y} = W\mathbf{x}, \quad W^{-1} \text{ takes } \mathcal{O}(D^3), \quad \text{GD takes } \mathcal{O}(D^2 T)$$

Mini-batch iterative fitting

- Data can be very large and thus we can't fit all into memory

$$D = [D^{(1)}, \ldots, D^{(K)}], \quad L_{\mathbf{w}}^{(k)} = \sum_{x \in D^{(k)}} (y - f_{\mathbf{w}}(x))^2$$

- Stochastic updates based on mini-batches

$$\mathbf{w}_t^{(k+1)} = \mathbf{w}_t^{(k)} - \epsilon \frac{\partial}{\partial \mathbf{w}} L_{\mathbf{w}}^{(k)} \bigg|_{\mathbf{w}=\mathbf{w}_t^{(k)}}, \quad \mathbf{w}_{t+1}^{(0)} = \mathbf{w}_t^{(K)}$$

02

Advanced Linear Algebra

Properties of Matrices

Eigendecomposition of matrices

- Eigenvalues are intrinsic properties of matrices

$$A\mathbf{v} = \underbrace{\lambda}_{\text{eigenvalue}} \underbrace{\mathbf{v}}_{\text{eigenvector}}, \quad A \in R^{N\times N} \implies A \text{ has } N \text{ eigenvalues}$$

- Eigenvectors - directions of change, eigenvalues - magnitude

$$A\mathbf{x} = A(\alpha\mathbf{v}_1 + \beta\mathbf{v}_2) = \alpha A\mathbf{v}_1 + \beta A\mathbf{v}_2 = \underbrace{\alpha\lambda_1\mathbf{v}_1}_{\text{stretches by } \alpha} + \underbrace{\beta\lambda_2\mathbf{v}_2}_{\text{stretches by } \beta}$$

Rank of matrix

- Number of non-zero eigenvalues

$$\mathrm{rank}(A) = 0 \implies A_{i,j} = 0 \text{ for all } i, j$$

- Full rank if rank is equal to number of columns

$$\mathrm{rank}(A) = \mathrm{rank}(A^T)$$

Determinant

- Product of eigenvalues

$$||A\mathbf{x}|| = ||\mathbf{x}|| \prod_i \lambda_i = ||\mathbf{x}|| \det(A)$$

- A matrix is invertible if square and has non-zero determinant

$$A\mathbf{v}_1 = \lambda_1 \mathbf{v}_1 \iff A^{-1}\mathbf{v}_1 = \frac{1}{\lambda_1} \implies \det(A^{-1}) = \frac{1}{\det(A)}$$

03

Kernels

Dealing with non-linearity

Feature maps

- A feature map is a transformation of the input dataset

$$\phi(\mathbf{x}) = [x_1 x_2, x_3^7, e^{x_4}]$$

- Feature maps could be infinite

$$\phi(\mathbf{x}) = [1^{||\mathbf{x}||}, 2^{||\mathbf{x}||}, \ldots]$$

Kernel trick

- Inner (dot) product of feature maps defines a kernel

$$\phi(\mathbf{x}_1)^T\phi(\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2)$$

- If we know the kernel we may not need the feature map

$$\mathbf{w}_{\text{best}} = \sum_i \alpha_i \phi(\mathbf{x}_i) \implies \mathbf{w}_{\text{best}}^T \phi(\mathbf{x}_{\text{test}}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_{\text{test}})$$

Properties of kernels

- Not every function with 2 arguments is a kernel

$$k(\mathbf{x}, \mathbf{y}) = -\mathbf{x}^T\mathbf{y} \text{ is not a kernel}$$

- Kernels need to be positive (semi-)definite

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) \implies K\mathbf{v}_i = \underbrace{\lambda_i}_{\text{non-negative}} \mathbf{v}_i$$

Properties of kernels

- There are commonly used kernels

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + r)^c \text{ or } \exp(\gamma||\mathbf{x} - \mathbf{y}||^2)$$

- Sums and products of kernels is a kernel

$$k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y})k_2(\mathbf{x}, \mathbf{y}) + k_3(\mathbf{x}, \mathbf{y})$$

Thank you!