Deep Learning

Lecture 2: Mathematical principles and backpropagation

Chris G. Willcocks

Durham University

Lecture Overview

1 Foundational statistics

- probability density function
- joint probability density function
- marginal and conditional probability
- expected values

2 Foundational calculus

- derivative of a function
- rules of differentiation
- partial derivative of a function
- rules of partial differentiation
- the Jacobian matrix

3 Mathematics of neural networks

- neural network functions
- computational graphs
- reverse mode of differentiation

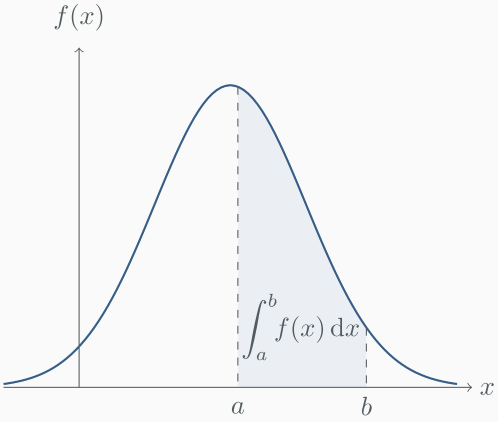
Definition: Probability density function

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called a probability density function if

$$\forall x \in \mathbb{R} : f(x) \geq 0,$$

and it's integral exists, where

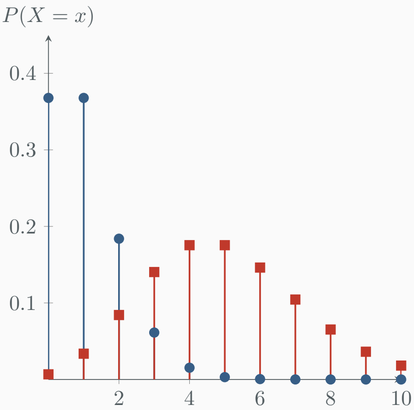
$$\int_{\mathbb{R}^n} f(x)\,\mathrm{d}x = 1.$$



Definition: Probability mass function

This is the discrete case of a probability density function, which has the same conditions, but where the integral is replaced with a sum

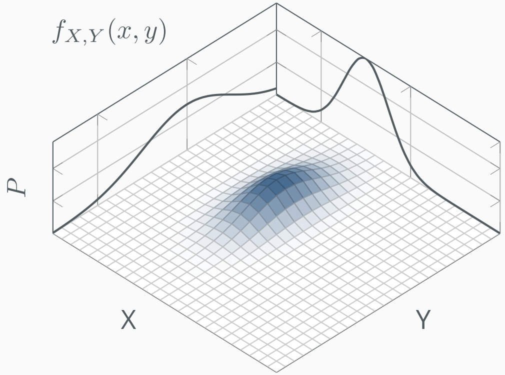
$$\sum_{i=1} P(X = x_i) = 1.$$



Definition: Joint density function

The joint density function $f_{X,Y}(x,y)$ for a pair of random variables is an extension of a PDF (non-negative function that integrates to 1) where

$$P(\underbrace{(X,Y)}_{\text{can be more than a pair}} \in \mathcal{A}) = \iint_{\mathcal{A}} f_{X,Y}(x,y)\,\mathrm{d}x\,\mathrm{d}y$$



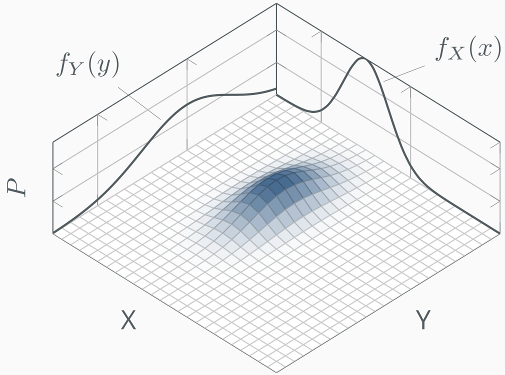
Definition: Marginal density function

The marginal density for the random variable X is where we integrate out the other dimensions

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}y\,,$$

and similarly

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}x\,.$$



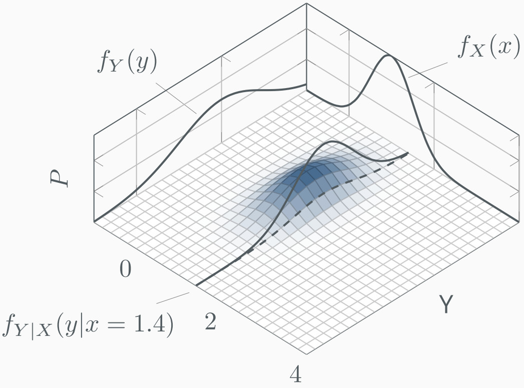
Definition: Conditional density function

The conditional density for pairs of random variables is

$$f_{X|Y}(x|Y=y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

which implies that the joint density is the product of the conditional density and the marginal density for the conditioning variable

$$\begin{aligned} f_{X,Y}(x,y) &= f_{X|Y}(x|y)f_Y(y) \\ &= f_{Y|X}(y|x)f_X(x) \end{aligned}$$



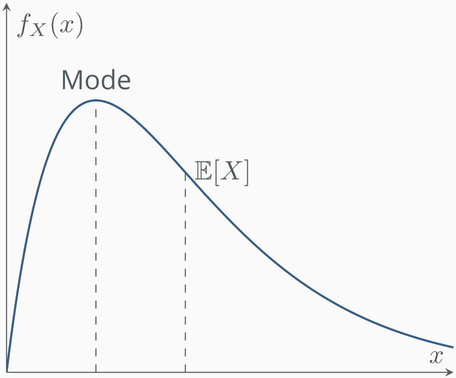
Definition: Expected value

The expected value or mean value for a continuous random variable is defined as

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x)\,\mathrm{d}x$$

also for a measurable function of X

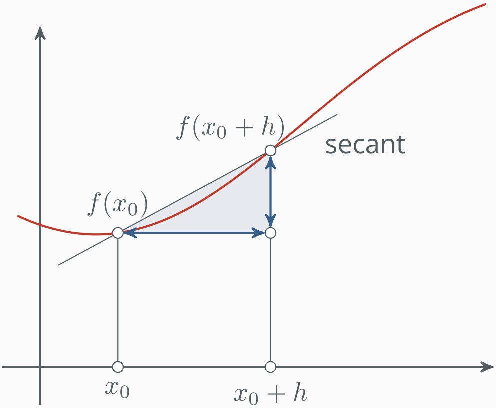
$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\,\mathrm{d}x$$



Definition: Derivative

For $h > 0$ the derivative of a function $f : \mathbb{R} \to \mathbb{R}$ at x is defined as the limit

$$f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

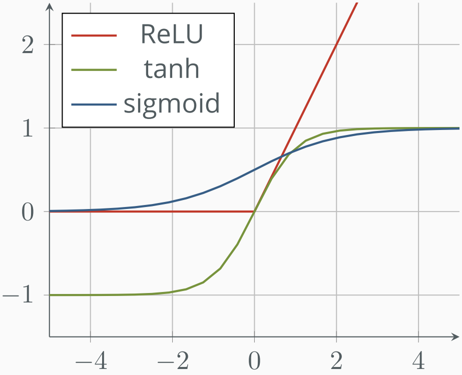


Example: Useful derivatives

These are some useful derivatives of common activation functions

1. $\mathrm{ReLU}'(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$
2. $\tanh'(x) = 1 - \tanh^2(x)$
3. $\mathrm{sigmoid}'(x) = \mathrm{sigmoid}(x) \cdot (1 - \mathrm{sigmoid}(x))$
4. $\sin'(x) = \cos(x)$

Try these derivatives and test some more on `https://www.desmos.com/calculator`



Foundational calculus rules of differentiation

Rules of differentiation

The **sum rule** is defined

$$(f(x) + g(x))' = f'(x) + g'(x)$$

The **product rule** is defined

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

The **quotient rule** is defined

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2}$$

The **chain rule** is defined

$$(g(f(x)))' = (g \circ f)'(x) = g'(f(x))f'(x)$$

Rules of differentiation

The **power rule** is defined

$$(x^n)' = nx^{n-1}$$

Example: What is the derivative of $h(x) = \sin(x^2)$?

$$\begin{aligned}
g(x) &= \sin(x) \\
g'(x) &= \cos(x) \\
f(x) &= x^2 \\
f'(x) &= 2x && \triangleright \text{ power rule} \\
h'(x) &= g'(f(x))f'(x) && \triangleright \text{ chain rule} \\
&= \cos(x^2)2x
\end{aligned}$$

Definition: Partial derivatives

For a function $f : \mathbb{R}^n \to \mathbb{R}$ of n variables $x_1, \dots, x_n$ the partial derivatives are defined

$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(\mathbf{x})}{h}$$

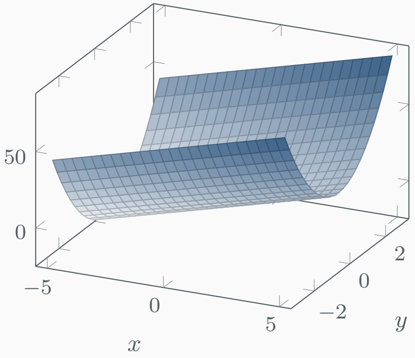
$$\vdots$$

$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, \dots, x_{n-1}, x_n + h) - f(\mathbf{x})}{h}$$

which get collected into a row vector known simply as the gradient of f with respect to $\mathbf{x}$

$$\nabla_{\mathbf{x}} f = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \cdots \frac{\partial f(\mathbf{x})}{\partial x_n} \right] \in \mathbb{R}^{1 \times n}$$

Example function: $f(x, y) = 4x + 7y^2$



Rules of partial differentiation

These rules of differentiation still apply, replacing derivatives with partial derivatives

The **sum rule** is defined

$$\frac{\partial}{\partial \mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})) = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial g}{\partial \mathbf{x}}$$

The **product rule** is defined

$$\frac{\partial f}{\partial \mathbf{x}}(f(\mathbf{x})g(\mathbf{x})) = \frac{\partial f}{\partial \mathbf{x}} g(\mathbf{x}) + f(\mathbf{x})\frac{\partial g}{\partial \mathbf{x}}$$

The **chain rule** is defined

$$\frac{\partial}{\partial \mathbf{x}}(g \circ f)(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}(g(f(\mathbf{x}))) = \frac{\partial g}{\partial f}\frac{\partial f}{\partial \mathbf{x}}$$

Example:

Calculate the partial derivative of $z^4 - \sin(y^2 + x)$ w.r.t. y

By use of the chain rule

$$\frac{\partial}{\partial y}(z^4 - \sin(y^2 + x)) = -\cos(y^2 + x)2y$$

Also we can calculate for x and z

$$\frac{\partial}{\partial x}(z^4 - \sin(y^2 + x)) = -\cos(y^2 + x)$$
$$\frac{\partial}{\partial z}(z^4 - \sin(y^2 + x)) = 4z^3$$

Try your own and test your answers on `https://www.wolframalpha.com`

Definition: the Jacobian matrix

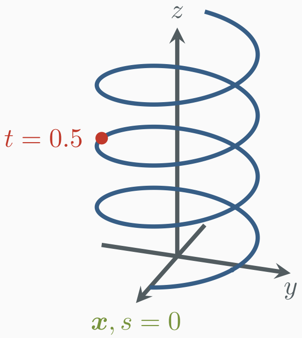
The collection of all first-order partial derivatives of a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$

$$\boldsymbol{J}_f = \nabla_{\mathbf{x}} f = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix},$$

$$\boldsymbol{J}_f(i,j) = \frac{\partial f_i}{\partial x_j}$$

Example function $f : \mathbb{R}^2 \to \mathbb{R}^3$

$f(t,s) = \langle \sin(t) + s, \cos(t), \frac{6t}{\pi} \rangle$



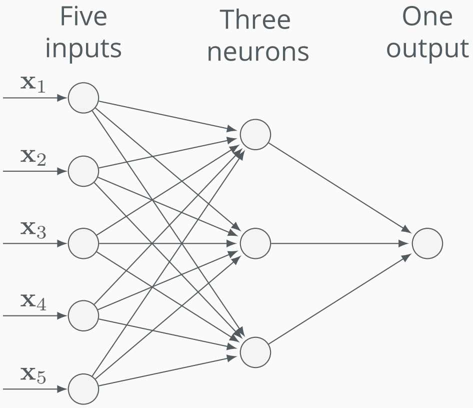
Definition: multilinear map & vector sum

A multilinear map is a function $f : \mathbb{R}^n \to \mathbb{R}^m$

$$f(\mathbf{x}) = \boldsymbol{W}\mathbf{x} + \mathbf{b}$$
$$\frac{\partial}{\partial \mathbf{x}}(\boldsymbol{W}\mathbf{x} + \mathbf{b}) = \boldsymbol{W}$$

A vector summation $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{x}) = \sum_{i=1} x_i$$
$$\boldsymbol{J}_f = \left[\frac{\partial x_1}{\partial x_1}, \frac{\partial x_2}{\partial x_2}, \cdots, \frac{\partial x_n}{\partial x_n}\right] = [1, 1, ..., 1]$$



Example: computational graphs

Consider a neural network with one linear layer

$$f(\mathbf{x}) = \boldsymbol{W}\mathbf{x} + \mathbf{b},$$

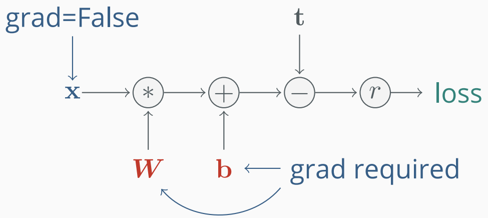
and r as the squared L_2 (Euclidean) norm

$$r(\mathbf{x}) = ||\mathbf{x}||_2^2 = \sum_{i=1} x_i^2,$$

where the network loss function $f : \mathbb{R}^n \to \mathbb{R}$ is the cost from ground truth labels $\mathbf{t}$

$$\text{loss} = ||f(\mathbf{x}) - \mathbf{t}||_2^2 = ||(\boldsymbol{W}\mathbf{x} + \mathbf{b}) - \mathbf{t}||_2^2.$$

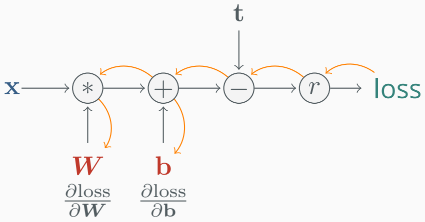
This is implemented as a computational graph



Backpropagation: reverse accumulation

Backpropagation is a reverse accumulation method suited for $f : \mathbb{R}^n \to \mathbb{R}^m$ where $m \ll n$ (usually $m = 1$). The algorithm is:

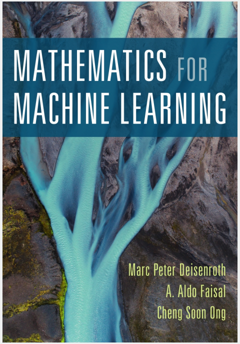
1. set **requires_grad=True** for any parameters we want to optimise ($\boldsymbol{W}$ and $\mathbf{b}$)
2. calculate the loss by a forward pass (feed the network $\mathbf{x}$ and see what the error is)
   - when doing this, save intermediate values from earlier layers
3. from the loss, traverse the graph in reverse to accumulate the derivatives of the loss at the leaf nodes



Further study recommended books

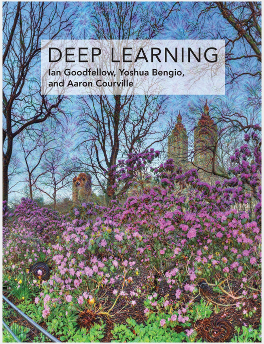
Deisenroth et al., 2020

More examples



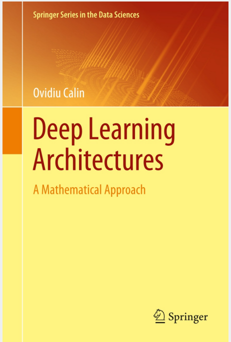
Goodfellow et al., 2016

Undergrad level



Calin, 2020

PhD level



References I

[1] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. Mathematics for machine learning. Available online, Cambridge University Press. 2020.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Available online, MIT press. 2016.

[3] Ovidiu Calin. Deep learning architectures: a mathematical approach. Springer, 2020.