

Deep Learning

AlejandroMllo

This document serves as a very brief survey of the topics covered in each chapter of the book Deep Learning [1].

Disclaimer: This document is completely extracted from [1], the author does not attribute any ownership over the material.

1 Introduction

- Deep Learning (DL) learns complicated concepts by building them out of simpler ones. DL is the study of models composed of either learned functions or learned concepts.
- Data Representation is key when finding patterns.
- Many AI tasks can be solved by designing the right set of features to extract.
- **Representation Learning:** discovers the mapping from representation to output and the representation itself.
 - Makes use of an *Autoencoder* which is a combination of:
 - * Encoder: Converts the input data into a different representation.
 - * Decoder: Converts the new representation back into the original format.
- Factors: sources of influence in the model.
- **Depth**, in a DL model, enables the computer to learn a multistep computer program (not all the information in a layer's activations necessarily encodes factors that explain the input). There are two ways to measure it:
 - Based on the number of sequential instructions, or
 - Based on the depth of the graph describing how concepts are related to each other (used by deep probabilistic models).
- DL is not an attempt to simulate the brain. It borrows ideas from different fields (one of which is neuroscience).

2 Linear Algebra

- Scalar: a single number. Denoted by lowercase variable names and italics.
- Vectors: ordered array of numbers. Denoted by lowercase and bold typeface. $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$
- Matrices: ordered 2-D $m \times n$ array of numbers. Denoted by uppercase and bold typeface.

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & \dots & A_{1,n} \\ \vdots & \dots & \vdots \\ A_{1,m} & \dots & A_{m,n} \end{bmatrix}$$

- Tensors: array of numbers arranged on a regular grid with a variable number of axes.
- Transpose: mirror image of a matrix/vector. $(A^\top)_{i,j} = A_{j,i}$.
- It is possible to add $m \times n$ matrices together. $\mathbf{C} = \mathbf{A} + \mathbf{B}$, where $C_{i,j} = A_{i,j} + B_{i,j}$.
- To add a scalar to a matrix or multiply a matrix by a scalar, perform that operation on each element of the matrix.

- Matrix Multiplication: $\mathbf{C} = \mathbf{AB}$ is defined as:

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$

where \mathbf{A} is of shape $m \times n$, \mathbf{B} is of shape $n \times p$ and \mathbf{C} is of shape $m \times p$.
Matrix product has some useful properties not covered in detail in the book.

- System of linear equations: $\mathbf{Ax} = \mathbf{b}$, where \mathbf{x} is a vector of unknown variables to be solved.
- Identity Matrix (\mathbf{I}_n): $n \times n$ matrix whose entries along the main diagonal are 1, while all the other entries are zero.
- Matrix Inverse (\mathbf{A}^{-1}): $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$. For a matrix to have an inverse, it must be $n \times n$ and all its columns be linearly independent, it means that no column is a linear combination of another column.
- Norm (L^p): function that measures the size of vectors.

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

where $p \in \mathbb{R}$, $p \geq 1$.

- The squared L^2 norm is commonly used and can be calculated as $\mathbf{x}^\top \mathbf{x}$.
- L^∞ norm (max norm): absolute value of the element with the largest magnitude in the vector.
- Frobenius Norm: used to measure the size of a matrix (analogous to the L^2 norm of a vector).

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

- Symmetric Matrix: $\mathbf{A} = \mathbf{A}^\top$.
- Unit Vector: $\|\mathbf{x}\|_2 = 1$ (unit norm).
- Vectors \mathbf{x} and \mathbf{y} are orthogonal if $\mathbf{x}^\top \mathbf{y} = 0$. If both this vectors have unit norm, then they are called orthonormal.
- Orthogonal Matrix: square matrix whose rows are mutually orthonormal and whose columns are mutually orthonormal.
- An eigenvector of a square matrix \mathbf{A} is a nonzero vector \mathbf{v} such that multiplication by \mathbf{A} alters only the scale of \mathbf{v} : $\mathbf{Av} = \lambda \mathbf{v}$. The scalar λ is known as the eigenvalue corresponding to this eigenvector. The eigendecomposition of \mathbf{A} is given by: $\mathbf{A} = \mathbf{V} \text{diag}(\lambda) \mathbf{V}^{-1}$, where \mathbf{V} is the matrix whose columns are each eigenvector of \mathbf{A} and $\text{diag}(\lambda)$ is the diagonal matrix of eigenvalues such that the eigenvalue at $\lambda_{i,i}$ is the one associated with the eigenvector (column) i of \mathbf{V} . Eigendecomposition is not defined for every matrix.
- Singular Value Decomposition (SVD): $\mathbf{A} = \mathbf{UDV}^\top$, where \mathbf{A} is a $m \times n$ matrix. \mathbf{U} ($m \times m$) is composed by the eigenvectors of \mathbf{AA}^\top ; \mathbf{V} ($n \times n$) is composed by the eigenvectors of $\mathbf{A}^\top \mathbf{A}$; and \mathbf{D} ($m \times n$) is a diagonal matrix composed by the eigenvalues of \mathbf{AA}^\top or $\mathbf{A}^\top \mathbf{A}$.
- Moore-Penrose pseudoinverse: The pseudoinverse of \mathbf{A} is defined as a matrix $\mathbf{A}^+ = \mathbf{VD}^+ \mathbf{U}^\top$, where \mathbf{U} , \mathbf{D} and \mathbf{V} are the SVD of \mathbf{A} , and the pseudoinverse \mathbf{D}^+ of a diagonal matrix \mathbf{D} is obtained by taking the reciprocal of its nonzero elements then making the transpose of the resulting elements.
- The Trace operator gives the sum of the diagonal entries of a matrix: $\text{Tr}(\mathbf{A}) = \sum_i A_{i,i}$. It has some useful identities to manipulate expressions.
- Determinant ($\det(\mathbf{A})$): function that maps matrices to real scalars. It is equal to the product of all the eigenvalues of a matrix.
- Principal Components Analysis (PCA): ML algorithm that can be derived using only knowledge of basic Linear Algebra. The reader can find a description of the implementation at section 2.12.

3 Probability and Information Theory

- Possible sources of uncertainty:
 1. Inherent stochasticity in the system being modeled.
 2. Incomplete observability.
 3. Incomplete modeling.

- Probability can be seen as the extension of logic to deal with uncertainty.
- Random Variable: variable x (discrete or continuous) that can take on different values randomly.
- Probability Distribution (PD): description of how likely a random variable or a set of random variables is to take on each of its possible states.
- Probability Mass Function (PMF): PD over discrete variables. A PMF, P , maps from a state of a random variable to the probability of that random variable taking on that state. A PMF acting on many variables at the same time is known as a joint probability distribution: $P(x = x, y = y)$.
- Probability Density Function (PDF): describes PD of continuous random variables. A PDF, $p(x)$, gives the probability of landing inside an infinitesimal region, not at a specific state.
- Marginal PD: it is the PD of a subset of a set of variables of which we know the PD.
- Conditional Probability: the probability of some event $x = x$, given that some other event $y = y$ has happened.

$$P(y = y \mid x = x) = \frac{P(y = y, x = x)}{P(x = x)}, \text{ where } P(x = x) > 0.$$

- Chain Rule of Conditional Probabilities: any joint PD over many random variables may be decomposed into conditional distributions over only one variable:

$$P(x^{(1)}, \dots, x^{(n)}) = P(x^{(1)}) \prod_{i=2}^n P(x^{(i)} \mid x^{(1)}, \dots, x^{(i-1)}).$$

- Two random variables x and y are independent ($x \perp y$) if:

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, p(x = x, y = y) = p(x = x)p(y = y)$$

- Two random variables x and y are conditionally independent given a random variable z ($x \perp y \mid z$) if:

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, z \in \mathbf{z}, p(x = x, y = y \mid z = z) = p(x = x \mid z = z)p(y = y \mid z = z)$$

- The expectation ($\mathbb{E}_{x \sim P}[f(x)]$), or expected value, of some function $f(x)$ w.r.t. a PD $P(x)$ is the average, or mean value, that f takes on when x is drawn from P .

$$\mathbb{E}_{x \sim P}[f(x)] = \sum_x P(x)f(x), \text{ for discrete variables.}$$

$$\mathbb{E}_{x \sim p}[f(x)] = \int p(x)f(x)dx, \text{ for continuous variables.}$$

- The variance gives a measure of how much the values of a function of a random variable x vary as we sample different values of x from its PD. The square root of the variance is the standard deviation.

$$\text{Var}(f(X)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$

- The covariance gives some sense of how much two values are linearly related to each other, as well as the scale of these variables. Two variables have zero covariance if they are not linearly dependent.

$$\text{Cov}(f(X), g(y)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(g(y) - \mathbb{E}[g(y)])]$$

- The correlation normalizes the contribution of each variable in order to measure only how much the variables are related.
- The covariance matrix of a random vector $\mathbf{x} \in \mathbb{R}^n$ is an $n \times n$ matrix, such that $\text{Cov}(\mathbf{x})_{i,j} = \text{Cov}(x_i, x_j)$. The diagonal elements of the covariance give the variance: $\text{Cov}(x_i, x_i) = \text{Var}(x_i)$.
- The Bernoulli Distribution is a distribution over a single binary random variable. It is controlled by a single parameter $\phi \in [0, 1]$, which gives the probability of the random variable being equal to 1. $P(x = x) = \phi^x(1 - \phi)^{1-x}$, $\mathbb{E}_{\mathbf{x}}[\mathbf{x}] = \phi$, $\text{Var}_{\mathbf{x}}(\mathbf{x}) = \phi(1 - \phi)$.
- The multinoulli, or categorical, distribution is a distribution over a single discrete variable with k different states, where k is finite. It is parametrized by a vector $\mathbf{p} \in [0, 1]^{k-1}$, where p_i gives the probability of the i -th state. The final, k -th state's probability is given by $1 - 1^\top \mathbf{p}$.
- Gaussian, normal, distribution: distribution over the real numbers. The parameters $\mu \in \mathbb{R}$ and $\sigma \in (0, \infty)$ control the normal distribution.

$$\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right).$$

To evaluate the PDF efficiently, parametrize the distribution with $\beta \in (0, \infty)$, to control the precision, or inverse variance:

$$\mathcal{N}(x; \mu, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{1}{2}\beta(x - \mu)^2\right).$$

Multivariate normal distribution: the normal distribution generalized to \mathbb{R}^n . It may be parametrized with a positive definite symmetric matrix Σ :

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \sqrt{\frac{1}{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right).$$

where $\boldsymbol{\mu}$ is a vector with the mean of the distribution, and Σ gives the covariance matrix.

Use a precision matrix β , to evaluate several times for different values of the parameters:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \beta^{-1}) = \sqrt{\frac{\det(\beta)}{(2\pi)^n}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \beta(\mathbf{x} - \boldsymbol{\mu})\right).$$

- Exponential distribution: PD with a sharp point at $x = 0$: $p(x; \lambda) = \lambda \mathbf{1}_{x \geq 0} \exp(-\lambda x)$. It uses $\mathbf{1}_{x \geq 0}$ to assign probability zero to all negative values of x .
- Laplace distribution: PD that places a sharp peak of probability mass at an arbitrary point μ : $\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x - \mu|}{\gamma}\right)$.
- Dirac delta function: makes possible to specify that all the mass in a PD clusters around a point.
- Mixture distribution: PD made up of several component distributions.
- Latent variable: random variable that is not possible to observe directly.
- Logistic sigmoid:

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

- Softplus function: (smoothed version of $x^+ = \max(0, x)$)

$$\zeta(x) = \log(1 + \exp x)$$

- Bayes' Rule:

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{x})P(\mathbf{y}|\mathbf{x})}{P(\mathbf{y})}$$

It is possible to compute $P(\mathbf{y})$ as: $P(\mathbf{y}) = \sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x})P(\mathbf{x})$

- Information theory: quantifies how much information is present in a signal.
 - Basic intuition: learning that an unlikely event has occurred is more informative than learning that a likely event has occurred. Likely events have low (or zero) information content. Less likely events have higher information content. Independent events have additive information content.
 - Self-information of an event $x = x$: $I(x) = -\log P(x)$ [nats]. One nat is the amount of information gained by observing an event of probability $\frac{1}{e}$. (The book uses log as the natural logarithm).
 - Shannon entropy: quantifies the amount of uncertainty in an entire PD. $H(\mathbf{x}) = \mathbb{E}_{x \sim P}[I(x)]$.
 - To measure how different two distributions over the same random variable are, use the Kullback-Leibler (KL) divergence: $D_{KL}(P||Q) = \mathbb{E}_{x \sim P}[\log P(x) - \log Q(x)]$.
 - Cross-entropy: $H(P, Q) = -\mathbb{E}_{x \sim P} \log Q(x)$.
- A Structured probabilistic model, or graphical model, represents the factorization of a PD with a graph (represents the PD as factors). Each node in the graph is a random variable, and an edge connecting two random variables means that the PD is able to represent direct interaction between those two random variables.
 - Directed models represent factorization into conditional PD.

$$p(\mathbf{x}) = \prod_i p(\mathbf{x}_i | \text{PaG}(x_i)),$$

where $\text{PaG}(x_i)$ represents the parents of x_i .

- Undirected models represent factorizations into a set of functions. Any set of nodes connected to each other is called a clique. Each clique $C(i)$ is associated with a factor $\phi^{(i)}(C^{(i)})$.

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^{(i)}(C^{(i)}),$$

where Z is a normalizing constant defined to be the sum or integral over all states of the product of the ϕ functions.

4 Numerical Computation

- Algorithms that solve mathematical problems by methods that update estimates of the solution via an iterative process.
- Underflow: numbers near zero that are rounded to zero.
- Overflow: numbers with large magnitude are approximated as ∞ or $-\infty$.
- Conditioning: how rapidly a function changes w.r.t. small changes in its inputs.
- Optimization: minimize or maximize an objective function $f(\mathbf{x})$ by altering \mathbf{x} . Sometimes, the value that minimizes or maximizes a function, is denoted with a superscript *: $\mathbf{x}^* = \arg \min f(\mathbf{x})$.
- Gradient Descent: reduce $f(x)$ by moving x in small steps with the opposite sign of the derivative. $f(x - \epsilon \text{sign}(f'(x)))$.
- Critical point: point with zero slope.
- In the context of Deep Learning it is tried to find a value of f that is very low but not necessarily minimal in any formal sense.
- The partial derivative $\frac{\partial}{\partial x_i} f(\mathbf{x})$ measures how f changes as only the variable x_i increases at point \mathbf{x} . The gradient generalizes the notion of derivative to the case where the derivative is w.r.t. a vector: the gradient of f is the vector containing all the partial derivatives, denoted $\nabla_{\mathbf{x}} f(\mathbf{x})$.
- The directional derivative in direction \mathbf{u} (a unit vector) is the slope of the function f in direction \mathbf{u} . The minimization occurs when the gradient points directly uphill, and the negative gradient points directly downhill.
- It is possible to decrease f by moving in the direction of the negative gradient. This is the method of steepest descent, or gradient descent: $\mathbf{x}' = \mathbf{x} - \epsilon \nabla_{\mathbf{x}} f(\mathbf{x})$, where ϵ is the learning rate, a positive scalar determining the size of the step. This method converges when every element of the gradient is zero (or very close to zero). To jump directly to the critical point, solve: $\nabla_{\mathbf{x}} f(\mathbf{x}) = 0$. Hill climbing is the generalization of this method for discrete spaces.
- If there is the function $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, then the Jacobian matrix $\mathbf{J} \in \mathbb{R}^{n \times m}$ of \mathbf{f} is defined such that $J_{i,j} = \frac{\partial}{\partial x_j} f(\mathbf{x})_i$.
- The Hessian matrix $\mathbf{H}(f)(\mathbf{x})$ is defined such that:

$$\mathbf{H}(f)(\mathbf{x})_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}).$$

Equivalently, the Hessian is the Jacobian of the gradient. Also, it is symmetric (if the function is continuous at such points). The second derivative in a specific direction represented by a unit vector \mathbf{d} is given by $\mathbf{d}^T \mathbf{H} \mathbf{d}$; when \mathbf{d} is an eigenvector of \mathbf{H} , the second derivative in that direction is the corresponding eigenvalue.

To the extent that the function to be minimized can be approximated well by a quadratic function, the eigenvalues of the Hessian thus determine the scale of the learning rate.

Using the eigendecomposition of the Hessian matrix, it is possible to generalize the second derivative test to multiple dimensions.

- Newton's Method: uses a second-order Taylor series expansion to approximate $f(\mathbf{x})$ near some point $\mathbf{x}^{(0)}$.
- In Deep Learning, sometimes the functions used are restricted to those that are either Lipschitz continuous or have Lipschitz continuous derivatives. A Lipschitz continuous function is a function f whose rate of change is bounded by a Lipschitz constant $\mathcal{L} : \forall \mathbf{x}, \forall \mathbf{y}, |f(\mathbf{x}) - f(\mathbf{y})| \leq \mathcal{L} \|\mathbf{x} - \mathbf{y}\|_2$.
- Constrained Optimization: used to find the maximal or minimal value of $f(\mathbf{x})$ for values of \mathbf{x} in some set S .

References

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.