

dass die Speicherung der Eingabeinformationen erzwungenermaßen positionsorientiert erfolgt. Erinnerungswürdige Merkmale (wie z.B. die diagonale Bewegung eines Balls), die an unterschiedlichen räumlichen Positionen auftauchen können, werden über Filter extrahiert. Hierbei ist auf die in Kapitel 2.1 genannten Vorteile hinzuweisen. Auf diese Weise erhält der Cell-State eine Struktur, die der einer Feature-Map gleicht. Der Formel 2.3 ist zu entnehmen, dass die Ausgabe ebenfalls eine solche Struktur aufweist.

2.4 Autoencoder

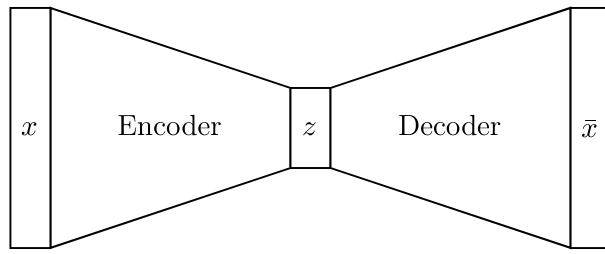


Abbildung 2.7: Struktur eines Autoencoders

Ein Autoencoder ist ein neuronales Netz, welches Eingangsdaten x mithilfe einer Funktion $encode(x)$ in eine Repräsentation z_x überführt und mithilfe einer Funktion $\bar{x} = decode(z_x)$ rekonstruiert. Die Funktionsparameter werden durch ein Training mit einer x mit \bar{x} vergleichenden Verlustfunktion gelernt. Üblicherweise werden Autoencoder zur Dimensionsreduktion und Merkmalsextraktion verwendet. Durch eine schmale Dimension der Schicht z , die Encoder und Decoder verbindet, kann das Netz dazu gezwungen werden, lediglich die für die Rekonstruktion relevantesten Aspekte der Eingabedaten zu erkennen und diese Informationen in z abzubilden. Ein Autoencoder mit solch einem Flaschenhals wird unvollständig genannt.

2.5 VAE

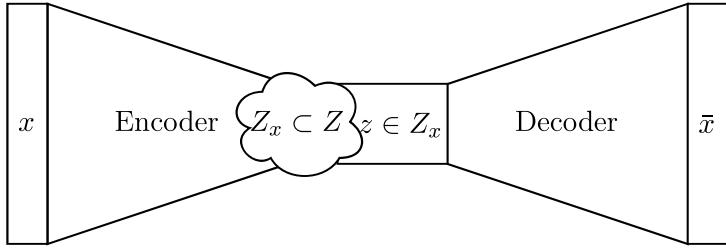


Abbildung 2.8: Struktur eines Variational Autoencoders

Ein Variational Autoencoder (VAE) [Kingma und Welling, 2014] ist ein generatives Modell. Eine konkrete Umsetzung dessen entspricht einem Autoencoder, der schmale Repräsentationen H_i lernt, deren Merkmalsausprägungen $h_1..h_n$ die Parameter einer mehrdimensionalen Wahrscheinlichkeitsverteilung P_{H_i} darstellen. Der Decoder nimmt eine Stichprobe z von P_{H_i} entgegen, um die Eingangsdaten zu rekonstruieren. Abbildung 2.8 zeigt auf, dass der Encoder bei gegebener Eingabe x statt eines einzigen Werts für die latente Repräsentation z eine Menge von möglichen Werten vorhersagt (Z_x). Die Menge wird durch P_{H_i} beschrieben.

Die Verlustfunktion ist bei Wahl der Normalverteilung für P gemäß [Kingma und Welling, 2014] gegeben durch:

$$\begin{aligned} L(x, \bar{x}, (\mu_1, \dots, \mu_k), (\sigma_1^2, \dots, \sigma_k^2)) \\ = L_{\text{Rekonstruktion}}(x, \bar{x}) + D_{KL}(\mathcal{N}((\mu_1, \dots, \mu_k), \text{diag}(\sigma_1^2, \dots, \sigma_k^2)) \parallel \mathcal{N}(0, I)), \quad (2.4) \end{aligned}$$

wobei

- x : die Eingabe ist.
- \bar{x} : die Ausgabe ist.
- (μ_1, \dots, μ_k) : Parameter (h_1, \dots, h_k) für den Erwartungswert sind.
- $(\sigma_1^2, \dots, \sigma_k^2)$: Parameter (h_{k+1}, \dots, h_n) für die Varianz sind.
- $\mathcal{N}(0, I)$: die Wahrscheinlichkeitsverteilung $P(Z)$ ist.

Der linke Term in der Verlustfunktion berechnet den Rekonstruktionsfehlerwert und sorgt dafür, dass der Encoder repräsentative Merkmale extrahiert. Der rechte Term stellt eine Regularisierung dar, welche durch eine hohe KL-Divergenz bestraft, wenn sich P_{H_i} zu sehr von einer deklarierten „Code-Verteilung“ ([Ian Goodfellow, 2016], Kap. 20.10.3) entfernt. Eine Stichprobenentnahme von der Code-Verteilung soll eine mögliche Wertezuweisung der latenten Dimension z generieren, die von dem Decoder in eine den

Daten im Datensatz ähnelnde Ausgabe dekodiert werden kann. Als Code-Verteilung wird in [Kingma und Welling, 2014] $\mathcal{N}(0, I)$ mit I als Identitätsmatrix verwendet.

Die Verlustfunktion ergibt sich aus einer im Folgenden auf Basis von [Doersch, 2021] erläuterten Problemstellung, die aus Sicht der Bayesschen Statistik gestellt wird. Das Ziel ist es bei generativen Modellen wie VAE, die unbekannte Verteilung P_{data} , die die Daten $X = x_1, \dots, x_N$ (Trainings- und Testdaten) generiert hat, mit einer Verteilung P_{model} zu modellieren, sodass Daten $\bar{x} \sim P_{model}$ generiert werden können, die den Daten von X ähneln. (vgl. [Doersch, 2021], Kap. 1)

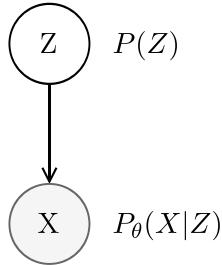


Abbildung 2.9: Graphisches Modell ohne Variational-Inference

Weil die Komplexität der Verteilung $P(X)$ in der Praxis zu groß ist, als dass eine Annäherung möglich ist, werden latente Variablen Z , welche eine Abstraktion der beobachtbaren Variablen X darstellen, mit einer voraussichtlich weniger komplexen Verteilung $P(Z)$ eingeführt. Dadurch liegt ein gerichtetes graphisches Modell mit einem Knoten der Zufallsvariablen Z , der auf den Knoten der Zufallsvariablen X zeigt, vor. Die Transformation von Z auf X erfolgt über eine Funktion $P(X|Z)$ mit den Parametern θ . Die Rand-Wahrscheinlichkeitsverteilung $P(X)$ kann nun durch Anwendung der Produktregel der bedingten Wahrscheinlichkeiten auf die multivariate Wahrscheinlichkeitsverteilung $P(X, Z)$ folgendermaßen formuliert werden:

$$P(X) = \int P(X|z; \theta)P(z) dz$$

Ziel ist die Maximierung der Log-Likelihood von $P(X)$ über alle X . Wenn die summierte Log-Likelihood unter Stichprobenentnahme von $P(z)$ maximal ist, dann liegt eine P_{data} optimal abbildende Verteilung P_{model} vor (vgl. [Ian Goodfellow, 2016], Kap. 5.5). Die Berechnung von optimalen Parametern θ nimmt allerdings aufgrund der großen Anzahl an als Stichproben zu entnehmenden Werten für Z bzw. aufgrund des hochdimensionalen Integrals eine exponentielle Zeitkomplexität an (vgl. [Doersch, 2021], Kap. 2 und [Blei

u. a., 2017], Kap. 2.1). Diese Problematik kann dem Fluch der Dimensionalität zugeschrieben werden (siehe auch [Ian Goodfellow, 2016], Kap. 5.11.1). Daher wird der Raum von Z auf solche Werte eingeschränkt, die mit hoher Wahrscheinlichkeit X erzeugt haben. Dafür wird eine Funktion $P(z|X)$ benötigt, die eine Verteilung über die Werte von Z liefert, welche mit hoher Wahrscheinlichkeit X produziert haben.

Diese Funktion kann mittels *Variational Inference* annähernd berechnet werden. Hierbei wird davon ausgegangen, dass eine günstige Deklarierung der Verteilung $P(Z)$ zur Modellierung ausreicht (siehe A-priori-Wahrscheinlichkeitsverteilung). Die wahre A-posteriori-Wahrscheinlichkeitsverteilung $p(z|x)$ wird daraufhin mittels Optimierung eines Stellvertreters $q(z|x)$ geschätzt, wobei $q(z|x)$ und die A-priori-Wahrscheinlichkeitsverteilung $p(z)$ einer definierten parametrisierten Wahrscheinlichkeitsdichtefunktion (z.B. Gauß-Funktion) entsprechen. (vgl. [Blei u. a., 2017], Kap. 2) Von dieser Funktion können dann Stichproben entnommen werden.

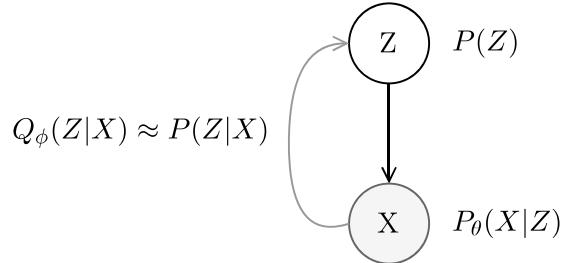


Abbildung 2.10: Graphisches Modell mit Variational-Inference

Das Ziel ist also nun, eine Verteilung $Q(z|X)$ bzw. $Q(z)$ zu finden. Dafür müssen die Parameter (sowohl θ als auch ϕ) des Modells so trainiert werden, dass sich $Q(z|x)$ $P(z|x)$ annähert:

$$\min D_{KL}(q(z|x) \parallel p(z|x))$$

Durch Umformung ergibt sich für die KL-Divergenz:

$$\begin{aligned}
 D_{KL}(q(z|x) || p(z|x)) &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x) - \log p(z|x)] && \text{Definition } D_{KL}(P || Q) \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x) - \log(p(z, x)/p(x))] && \text{Satz von Bayes} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x) - (\log p(z, x) - \log p(x))] && \text{Log-Quotientregel} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x) - \log p(z, x) + \log p(x)] && \text{Distributivgesetz} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x) - \log p(z, x)] + \log p(x) && \text{A.1} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log q(z|x)/p(z, x)] + \log p(x) && \text{Log-Quotientregel} \\
 &= -\mathbb{E}_{z \sim q(z|x)}[\log p(z, x)/q(z|x)] + \log p(x) && \text{A.2}
 \end{aligned}$$

Der Erwartungswert ($\mathbb{E}[X]$) stellt den sogenannten ELBO (*Evidence Lower Bound*) Term dar. Im Folgenden wurde nach $\log p(x)$ aufgelöst.

$$\begin{aligned}
 \iff \log p(x) &= \mathbb{E}_{z \sim q(z|x)}[\log p(z, x)/q(z|x)] + D_{KL}(q(z|x) || p(z|x)) \\
 &\geq \mathbb{E}_{z \sim q(z|x)}[\log p(z, x)/q(z|x)] && D_{KL} \geq 0
 \end{aligned}$$

Hierbei gilt, dass $\log p(x)$ eine Konstante ist, die sich während der Optimierung des Stellvertreters $q(z|x)$ niemals ändert. Die gezeigte Formel impliziert also, dass, wenn der ELBO maximiert wird, die KL-Divergenz minimiert wird. Somit kann die Maximierung vom ELBO als neues Ziel herhalten. Dies ist von Vorteil, da für die Berechnung des ELBO Terms sämtliche notwendige Komponenten zur Verfügung stehen.

$$\begin{aligned}
 \text{ELBO} &= \mathbb{E}_{z \sim q(z|x)}[\log p(z, x)/q(z|x)] \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log p(x|z)p(z)/q(z|x)] \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log p(x|z) + \log p(z)/q(z|x)] && \text{Log-Produktregel} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log p(x|z)] + \mathbb{E}_{z \sim q(z|x)}[\log p(z)/q(z|x)] && \text{Definition } \mathbb{E}[X] \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log p(x|z)] - \mathbb{E}_{z \sim q(z|x)}[\log q(z|x)/p(z)] && \text{A.2} \\
 &= \mathbb{E}_{z \sim q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x) || p(z)) && \text{Definition } D_{KL}(P || Q)
 \end{aligned}$$

Um den Wert zu maximieren, sollte also $\log p(x|z)$ möglichst groß und die KL-Divergenz zwischen $q(z|x)$ und $p(z)$ möglichst klein sein. Daher ist das Ziel der Maximierung der Log-Likelihood und der Minimierung der KL-Divergenz zwischen der A-posteriori-

Wahrscheinlichkeitsverteilung $q(z|x)$ und der A-priori-Wahrscheinlichkeitsverteilung $p(z)$ gegeben. Dieses Ziel kann als Verlustfunktion (siehe Gleichung 2.4) verwendet werden, um optimale Parameter θ und ϕ zu lernen.

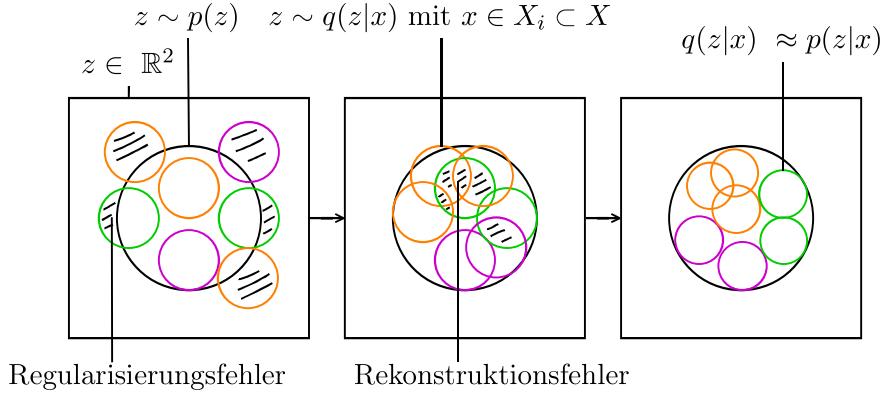


Abbildung 2.11: Erzwungene Ordnung im latenten Raum des VAEs

In Abbildung 2.11 wurden die Daten X zur Visualisierung in 3 Mengen aufgeteilt (X_i), wobei die Elemente derselben Menge ähnliche und die Elemente unterschiedlicher Mengen unähnliche Merkmale aufweisen. Die Abbildung zeigt auf, wie während des Trainings vom VAE durch die Verlustfunktion eine Ordnung im latenten Raum geschaffen wird.

2.6 Clustering

Clustering ist eine Methode der unüberwachten Gruppierung von Objekten, bei der versucht wird, Gruppen zu finden, die jeweils möglichst Objekte mit ähnlichen Eigenschaften beinhalten und im Vergleich ihrer Objekte mit Objekten anderer Gruppen möglichst unterschiedliche Eigenschaften aufweisen. (vgl. [Guojun Gan, 2007], Kap. 1.1)

Clustering Probleme können in Hard-Clustering und Soft-Clustering unterteilt werden. Beim Hard-Clustering wird ein Datenpunkt lediglich einem Cluster zugeordnet. Beim Soft-Clustering wird dem Datenpunkt für jedes Cluster eine Zugehörigkeitswahrscheinlichkeit berechnet, sodass der Datenpunkt mehreren Clustern zugewiesen werden kann. (vgl. [Guojun Gan, 2007], Kap. 1.2.4) Die Datenpunkte können nach einem Soft-Clustering jeweils einem einzigen Cluster „hart“ zugeordnet werden, indem das Cluster mit der höchsten Zugehörigkeitswahrscheinlichkeit ausgewählt wird (vgl. [Guojun Gan, 2007], Kap. 8).