

教材の利用条件

- 本教材（PDF版）は「クリエイティブ・コモンズ・ライセンス 表示-非営利-継承 4.0 国際 (CC BY-NC-SA 4.0)」のもと提供しています。
 - ライセンスの内容を知りたい方は以下よりご確認ください。
<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.ja>
 - クレジットの表示：資料のいずれかのページに、提供元として下記を明記してください。
 - 東京工業大学 データサイエンス・AI全学教育機構
- 【教員限定】本教材をご利用の方へアンケートのお願い
 - ご回答はこちら：<https://forms.gle/8s7QcaWKfNZm8uBWA>
- 【教員限定】本教材（PPT版）や課題等について
 - 日本国内の教育機関に在籍する教員に限定して提供しています。
 - お申込みはこちら：<https://forms.gle/8PAhW8taARjAsXay5>



応用基礎データサイエンス・AI 第二

第7回 機械学習・深層学習・AI応用

2024.07.24

データサイエンス・AI全学教育機構

講義の概要

1. 強化学習

- ゲーム対戦エージェント
- 身体運動の学習

2. 深層生成モデル

- 変分自己符号化器 (VAE)
- 敵対的生成ネットワーク (GAN)
- 拡散モデル

3. 注意機構とトランスフォーマ

- Word2Vec
- Seq2Seq
- 注意機構
- トランスフォーマ

強化学習

ゲーム対戦エージェント
身体運動の学習

[参考] 機械学習の種類

「応用基礎データサイエンス・AI第二」
第3回講義資料より

- 学習方法の観点からの分類
 - 教師あり学習 (supervised learning)
 - 正解・属性などの情報がラベル付け (アノテーション) されたデータからの学習
 - 回帰, 分類
 - 教師なし学習 (unsupervised learning)
 - ラベル付けされていないデータからの学習
 - クラスタリング, 次元削減 (可視化)
 - 半教師あり学習 (semi-supervised learning)
 - 教師あり学習と教師なし学習の併用
 - 自然言語処理, 画像処理, 音声処理, etc.
 - 強化学習 (reinforcement learning)
 - 行動 (action) と環境からの報酬に基づく学習
 - 知的エージェントの行動制御, 自動運転, ゲームの最適手探索
 - 生成AIのファインチューニング

強化学習

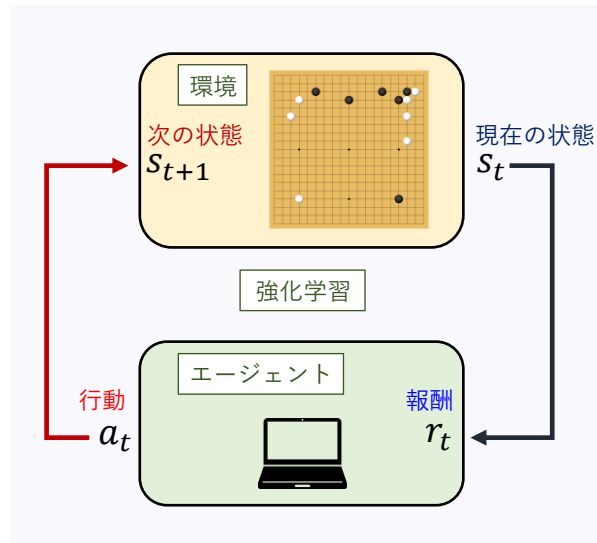
- 与えられた環境で、エージェントの最適な行動選択の方策を学習

- 環境 (environment) — エージェント (agent)
 - ・ 例) 環境：囲碁・将棋 — エージェント：対戦棋士
- 状態 (state)
 - ・ 現在の環境の状態： s_t
- エージェントが取り得る行動 (action)
 - ・ 現在の状態で選択する行動： a_t
- (即時) 報酬 (reward)
 - ・ 現在の状態に遷移した際に得た報酬： r_t
 - ・ 行動 a_t による遷移後の状態と得る報酬： s_{t+1}, r_{t+1}

- 行動の選択基準

- 期待収益 (= 将来の累積報酬の期待値) の最大化
 - ・ 遷移確率： $p_{t,t+1} = P_r(s_{t+1}|s_t, a_t)$
 - ・ 方策 (policy)： $\pi(a_t, s_t) = P_r(a_t|s_t)$
 - ・ 期待収益 R を最大化する方策 π の決定

$$\arg \max_{\pi} R = \arg \max_{\pi} E \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} p_{\tau, \tau+1} \pi(a_{\tau}, s_{\tau}) r_{\tau+1} \right]$$



$\gamma \in [0,1]$: 割引率

ゲーム (チェス・囲碁・将棋) 対戦エージェント

- プロトッププレイヤーに勝つ実力を獲得

- チェス

- Deep Blue (IBM) が人間のチェス世界王者に勝利 (1997)

- 囲碁

- AlphaGo Fan (Google DeepMind) がヨーロッパ王者プロ棋士に勝利 (2015)
 - AlphaGo Master (Google DeepMind) が世界最強と言われるプロ棋士に勝利 (2017)

- 将棋

- Ponanza (山本一成) が電王戦で名人位プロ棋士に勝利 (2017)

- アプローチ

- 強化学習

- エージェント同士で膨大な数の対戦を繰返して学習

- モンテカルロ木探索

- 数十手先までの候補の中から最善手を選択

- 深層学習

- 取り得る手の確率分布や方策を計算するNN

モンテカルロ木探索 (Monte Carlo tree search, MCTS)

- 取り得る手とその先の変化を表す手数分の深さの木をランダムサンプリングに基づいて構築
- ある手を選んだとき、その先ランダムに手を選択して対戦シミュレーションを何回か繰り返し、勝利する回数を調べる
- シミュレーションの結果、勝率が高かった手を最善手とする

身体運動の学習

- ロボットの歩行運動の獲得

- ロボットの運動学習

- 強化学習の利用

- 異なるロボット形状・用途に合う教師データを用意しにくい
オンライン学習が一般的

- 学習におけるシミュレータの利用

- 物理的な運動を含む実環境での学習は時間がかかる

- ⇒ 仮想環境でのシミュレーションによる学習の高速化
シミュレーションでは実環境をすべてカバーできない

- DayDreamer* (2022)

- 世界モデル (world model)

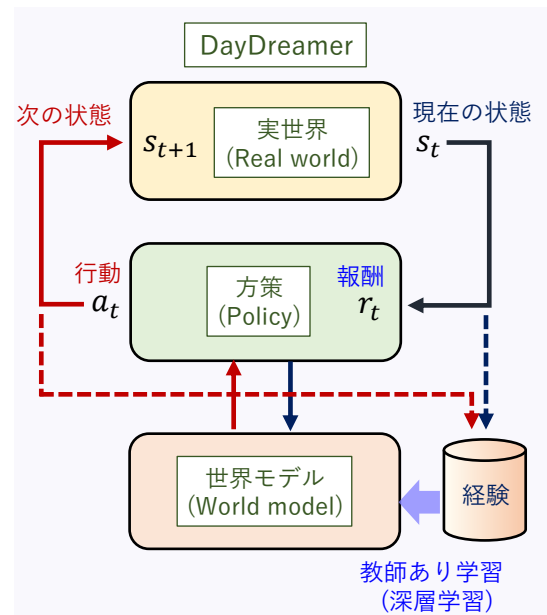
- 実環境 (実世界) での行動/状態遷移/報酬を経験として蓄積
蓄積された経験に基づいて実世界の分布を教師あり学習

- 行動の選択基準 (方策) の最適化

- 世界モデルに基づいて報酬の最大化

- 実世界への適用と世界モデルの学習のサイクル

- 得られた方策に基づいた経験の蓄積と世界モデルの学習を反復



* Philip Wu, et al., "DayDreamer: World models for physical robot learning," arXiv:2206.14176v1 [cs.RO], 2022 (<https://doi.org/10.48550/arXiv.2206.14176>)

わずか1時間で四脚ロボットが歩行動作を学習したデモ
<https://www.youtube.com/watch?v=xAXvfVTgqr0>

深層生成モデル

変分自己符号化器 (VAE)

敵対的生成ネットワーク (GAN)

拡散モデル

生成モデルとは

- 観測データがある分布に従うとしたときの確率分布のモデル化

- 生成モデルによりできること

- サンプルング

- ・ 尤もらしい新たな人工データの生成
例) 生成画像, 生成テキスト, 合成音声

- 密度推定

- ・ 対象とするデータがどれだけ尤もらしいかの評価
例) 翻訳結果の日本語らしさの評価

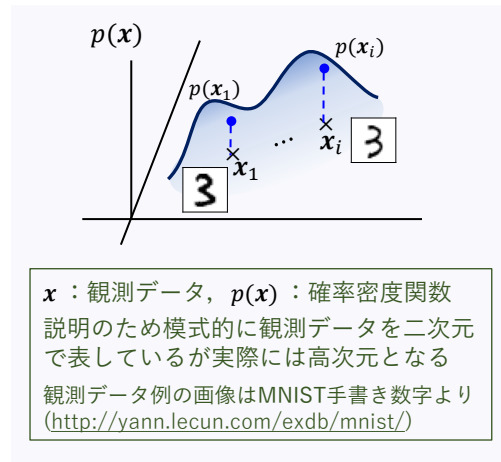
- 補間

- ・ データの復元・モーフィング(変形)
例) 欠損部分の修復, 不要部分の消去, ノイズ除去

- 深層生成モデル

- データが生成される確率分布を陽または陰に表現するNN

- ・ 変分自己符号化器 (VAE)
- ・ 敵対的生成ネットワーク (GAN)
- ・ 拡散モデル
- ・ WaveNet



- ・ 一般に正解 (真の分布) は未知であり, 入力データ (事例) のみで学習するため教師なし学習となる
- ・ 深層生成モデルは画像生成だけでなく, 音声合成, テキスト処理, ロボット制御などにも応用されている
- ・ WaveNet については第6回講義資料を参照

自己符号化器

- 入力と同じ値を出力するように学習されたFFNN

- 入力: $x \in \mathbb{R}^L$
- 符号器 (encoder)
 - ・ 入力を中間的な表現 (符号 (code)) に変換: $c \in \mathbb{R}^M$
通常 $M < L$ に選ぶ
- 復号器 (decoder)
 - ・ 中間表現 (符号) から入力を復元: $\hat{x} \in \mathbb{R}^L \approx x$
- 学習
 - ・ 入出力間の二乗誤差最小化

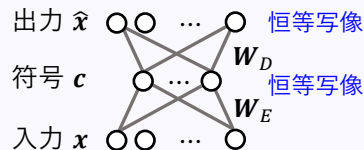
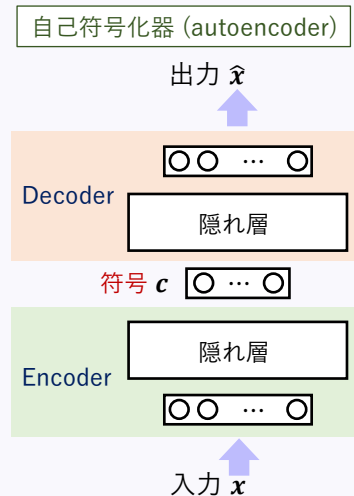
- 自己符号化器と主成分分析

- 単純構成の自己符号化器
 - ・ 符号器・復号器ともに一層構造
 - ・ 活性化関数は恒等写像
 - ⇒ 入出力間は線形変換の関係

$$\hat{x} = W_D^T c = W_D^T (W_E^T x) = (W_D^T W_E^T) x$$

$M < L$ の場合は主成分分析による次元圧縮と同等

・ 符号器・復号器とも
隠れ層の構成は任意
・ 確率的な構造を持た
ない自己符号化器は
生成モデルではない



変分自己符号化器 (VAE)

- 自己符号化器の構造を持つ深層生成モデル

- 潜在変数による分布関数の表現

- 観測データ x
- 観測データ x の確率密度関数 $p(x)$: 未知
- 潜在変数 (latent variable) z : 観測できない変数

$$p(x) = \int p_{\theta}(x, z) dz = \int p_{\theta}(x|z)p_{\theta}(z) dz$$

- 変分自己符号化器 (variational autoencoder, VAE)

- 仮定 (観測データが連続値の場合)

- 潜在変数が正規分布に従うとする

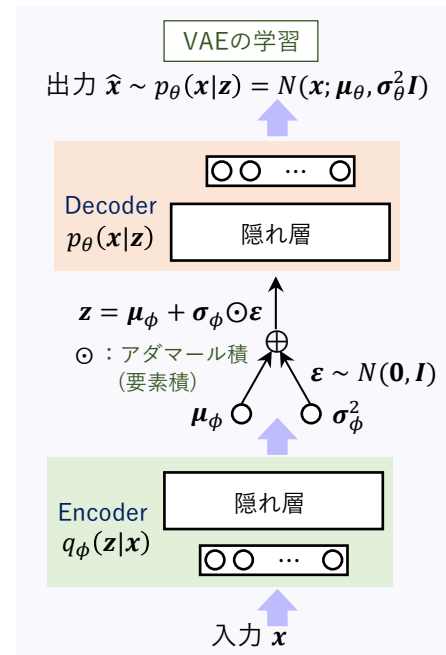
$$p_{\theta}(z) \sim N(z; \mathbf{0}, I), \quad p_{\theta}(x|z) \sim N(x; \mu_{\theta}, \sigma_{\theta}^2 I), \quad q_{\phi}(z|x) = N(z; \mu_{\phi}, \sigma_{\phi}^2 I)$$

- 学習

- 入力から潜在変数の値を推定
- 潜在変数から出力を生成
- $p(x)$ の最尤推定 $\Rightarrow D_{\text{KL}}(q_{\phi}(z|x) \parallel p_{\theta}(z|x))$ の最小化

$$\min_{\theta, \phi} \left\{ -\mathbb{E}_{q_{\phi}} [\log p_{\theta}(x|z)] + D_{\text{KL}}(q_{\phi}(z|x) \parallel p_{\theta}(z)) \right\}$$

KL情報量



$p(x)$ に従うサンプル x の生成:

$N(\mathbf{0}, I)$ に従う z のサンプルを復号器に入力したときの出力
生成例: <https://github.com/ChengBinJin/VAE-Tensorflow>

* D. P. Kingma, M. Welling, "Auto-encoding variational Bayes," arXiv:1312.6114v11 [stat.ML], 2013 (<https://doi.org/10.48550/arXiv.1312.6114>)

[参考] カルバック・ライブラー (KL) 情報量

- 二つの確率分布の差異を測る尺度

- 定義

- 離散確率分布

- 二つの確率分布 $P = \{p_k\}, Q = \{q_k\}$
 - P の Q に対するKL情報量 $D_{KL}(P \parallel Q)$

$$D_{KL}(P \parallel Q) = \sum_k p_k \log \frac{p_k}{q_k} = E_P \left[\log \frac{p_k}{q_k} \right]$$

- 連続確率分布

- 二つの確率分布 $P: p(x), Q: q(x)$

$$D_{KL}(P \parallel Q) = \int p(x) \log \frac{p(x)}{q(x)} dx = E_P \left[\log \frac{p(x)}{q(x)} \right]$$

- 性質

- (1) $D_{KL}(P \parallel Q) \geq 0$ (Gibbsの不等式)
- (2) $D_{KL}(P \parallel P) = 0$ (不等式で等式が成り立つのは $P \equiv Q$ のときのみ)
- (3) $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$ (非対称)

カルバック・ライブラー情報量
(Kullback–Leibler divergence)

- KL情報量, KLダイバージェンスとも呼ばれる
- KL 距離という用語も見かけるが, 数学的な距離の定義である対称性と三角不等式を満足しないことから, D_{KL} は距離とは呼べないことに注意する

$E_P[\cdot]$ は確率分布 P に関する期待値を表す

[参考] VAE：最尤推定の導出

- KL情報量の最小化の解釈

- 対象とするKL情報量

$$\begin{aligned} D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x}) - \log p_{\theta}(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x}) - \log p_{\theta}(\mathbf{x}|\mathbf{z}) - \log p_{\theta}(\mathbf{z}) + \log p_{\theta}(\mathbf{x})] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log p_{\theta}(\mathbf{x}|\mathbf{z})] + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x}) - \log p_{\theta}(\mathbf{z})] + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x})] \\ &= -\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] + D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z})) + \log p_{\theta}(\mathbf{x}) \end{aligned}$$

- 最適化の目的関数

$$-\log p_{\theta}(\mathbf{x}) + D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x})) = -\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] + D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}))$$

- 右辺はVAEの最適化における目的関数

- 左辺第1項： $-\log p_{\theta}(\mathbf{x})$

⇒ \mathbf{x} が与えられたときの負の対数尤度

- 左辺第2項： $D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x}))$

⇒ $q_{\phi}(\mathbf{z}|\mathbf{x})$ と $p_{\theta}(\mathbf{z}|\mathbf{x})$ のKL情報量，下限は 0

$q_{\phi}(\mathbf{z}|\mathbf{x}) \approx p_{\theta}(\mathbf{z}|\mathbf{x})$ が成り立てば目的関数の最小化は最尤推定に一致する

右辺の符号を反転した値は ELBO
(evidence lower bound) と呼ばれる

敵対的生成ネットワーク (GAN)

- 特定の分布を仮定しない深層生成モデル

- 観測データの分布

- 一般には高次元かつ複雑

- ⇒ 基本的な分布では表せず推定は困難

- 観測データ x に対する $p(x)$ の値が直接求められなくても

- 観測データの分布 $p(x)$ に従うサンプルが生成できれば有用

- 敵対的生成ネットワーク* (generative adversarial network, GAN)

- 生成器 (generator)

- ノイズ z から真の実 (real) データに似せた偽 (fake) データ $G(z)$ を生成

- 識別器 (discriminator)

- 入力が真のデータである確率 $D(x)$ を出力

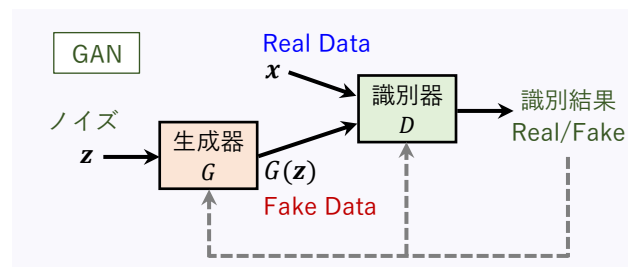
- ⇒ 理想的な識別器 : $D(x) = 1, 1 - D(G(z)) = 1$

- 完璧な生成器 : $1 - D(G(z)) = 0$

- ネットワークの学習

- 生成器は識別器を騙すように最適化
 - 識別器は真贋を識別できるように最適化

$$\min_G \max_D \{E_x[\log D(x)] + E_z[\log(1 - D(G(z)))]\}$$



* Ian J. Goodfellow, et al., "Generative adversarial nets," arXiv:1406.2661v1 [stat.ML], 2014 (<https://doi.org/10.48550/arXiv.1406.2661>)

GANの画像生成への応用

- 自然画像の生成へのアプローチ

- GANの問題点

- 学習の困難さ

- 生成器と識別器の学習のバランスが難しい
識別器が生成器に比べて性能が上がり過ぎる場合
⇒ 評価関数の第2項の勾配が消失し生成器の学習が進まない

- モード崩壊

- 同じような画像ばかりが生成される
一旦偽画像が生成できれば他の画像を生成する学習が進まない

- CycleGAN* (2017)

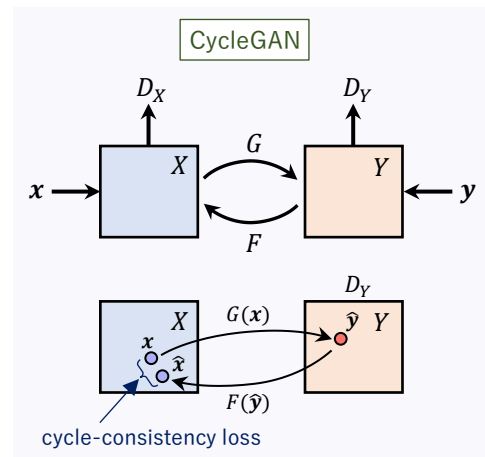
- 画像間の変換

- 異なる二つの画像データの組: $\{x, y\}, x \in X, y \in Y$
 - 二つの生成ネットワーク $X: \{G(x), D_X\}, Y: \{F(y), D_Y\}$

- ネットワークの学習

- 循環的な合成変換の繰返し

$$\begin{aligned}x &\rightarrow G(x) \rightarrow F(G(x)) \approx x \\y &\rightarrow F(y) \rightarrow G(F(y)) \approx y\end{aligned}$$



* Jun-Yan Zhu, et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," arXiv:1703.10593v7 [cs.CV], 2017 (<https://doi.org/10.48550/arXiv.1703.10593>)
画像生成例は上記論文を参照のこと

拡散モデル

- 最近の画像生成サービスで利用されている深層生成モデル

- テキスト情報からの画像生成 (text-to-image) サービス

例) Stable Diffusion, DALL-E 2, Midjourney, Imagen, etc.

画像生成例は各サービスのサイトを参照

- 拡散モデル (diffusion model)* (2015)

- 拡散過程 (forward diffusion process)

- 元の複雑な分布から扱いやすい単純な分布への変換

元画像 $\mathbf{x}^{(0)} \sim q(\mathbf{x}^{(0)})$ に徐々にノイズを加えていき、最終的に単純なガウス分布に従うノイズ画像 $\mathbf{x}^{(T)} \sim N(\mathbf{0}, I)$ に変換

$$q(\mathbf{x}^{(T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^T q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})$$

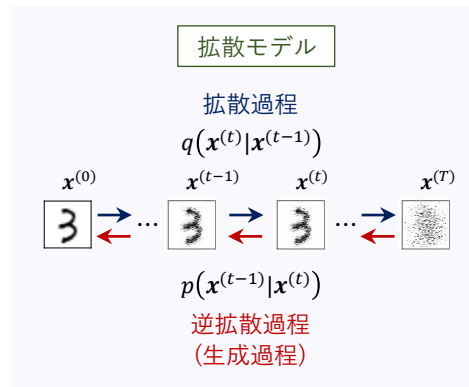
- 逆拡散過程 (生成過程) (reverse generative diffusion process)

- 単純な分布に従うノイズから元の複雑な分布への変換

⇒ 拡散過程を逆に辿り、徐々にノイズを取り除いていく
最終的にノイズ画像から元画像の分布に近い画像を生成

$$p(\mathbf{x}^{(0)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^T p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$$

静止画に加えて動画生成 (text-to-video) サービスも提供され始めた(Sora, 2024)



* Jascha Sohl-Dickstein, et al., "Deep unsupervised learning using nonequilibrium thermodynamics," arXiv:1503.03585v8 [cs.LG], 2015 (<https://doi.org/10.48550/arXiv.1503.03585>)

注意機構とトランスフォーマ

Word2Vec

Seq2Seq

注意機構

トランスフォーマ

自然言語処理における単語の分散表現

- 単語が持つ意味表現の数値化

- 分散表現

- 単語が持つ意味をベクトルで数値表現

- 語彙空間から低い次元のベクトル空間への写像 (埋め込み)

⇒ 自然言語の語彙空間をone hot ベクトルで表したときの次元数 = 語彙数 $N_V \approx 10^6$

分散表現ベクトルの次元数設定 $N_D = 50 \sim 1000$

- ニューラルネットワークにより獲得

- 分散表現を用いる利点

- 異なる単語の意味の近さを数値化できる
- 単語同士の加減算が意味を持つ

⇒ “king” - “man” + “woman” ⇒ ?

“biggest” - “big” + “small” ⇒ ?

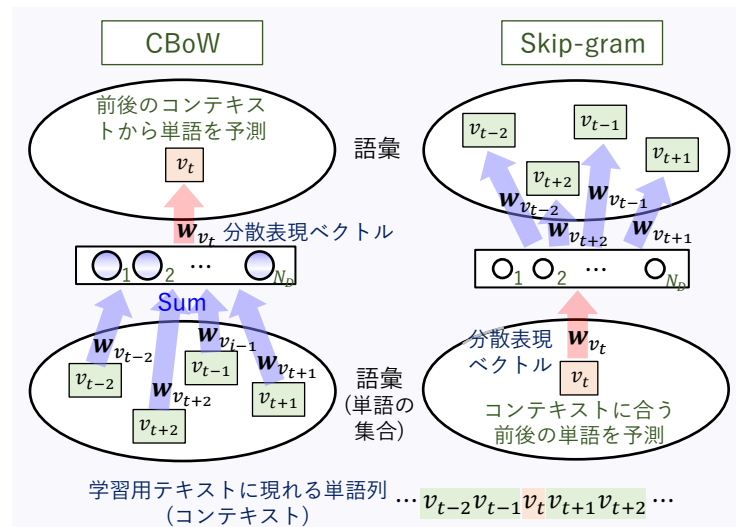
- Word2Vec* (2013)

- 分散表現の学習手法

- CBow (continuous bag-of words) model
- Skip-gram model

単語埋め込み (word embedding) と呼ばれる

単語長によらず一定次元のベクトルで表現



* Tomas Mikolov, et al., "Efficient estimation of word representations in vector space," arXiv:1301.3781v3 [cs.CL], 2013 (<https://doi.org/10.48550/arXiv.1301.3781>)

自然言語処理における単語の分散表現

- 単語が持つ意味表現の数値化

- 分散表現

- 単語が持つ意味をベクトルで数値表現

- ・ 語彙空間から低い次元のベクトル空間への写像 (埋め込み)

⇒ 自然言語の語彙空間をone hot ベクトルで表したときの次元数 = 語彙数 $N_V \approx 10^6$

分散表現ベクトルの次元数設定 $N_D = 50 \sim 1000$

- ・ ニューラルネットワークにより獲得

- 分散表現を用いる利点

- ・ 異なる単語の意味の近さを数値化できる
- ・ 単語同士の加減算が意味を持つ

⇒ “king” - “man” + “woman” ⇒ “queen”

“biggest” - “big” + “small” ⇒ “smallest”

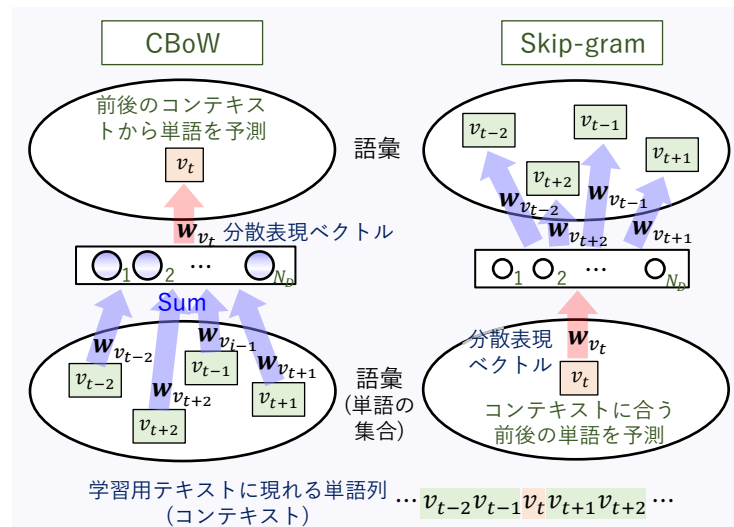
- Word2Vec* (2013)

- 分散表現の学習手法

- ・ CBoW (continuous bag-of words) model
- ・ Skip-gram model

単語埋め込み (word embedding) と呼ばれる

単語長によらず一定次元のベクトルで表現



* Tomas Mikolov, et al., "Efficient estimation of word representations in vector space," arXiv:1301.3781v3 [cs.CL], 2013 (<https://doi.org/10.48550/arXiv.1301.3781>)

[参考] 単語の分散表現：Word2Vec

- Word2Vec における学習の定式化

- Skip-gram モデル

- 語彙

- 語彙数 N_V : $V = \{v_1, v_2, \dots, v_{N_V}\}$

- 分散表現ベクトル

- 次元数 N_D : $\mathbf{w}_{v_i} = [w_{i1} \dots w_{ij} \dots w_{iN_D}]^T \in \mathbb{R}^{N_D}$
- 重み行列

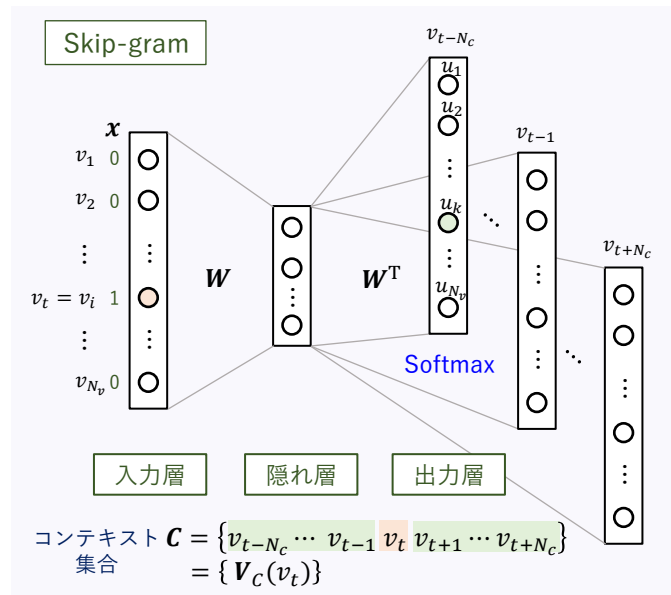
$$\mathbf{W} = [\mathbf{w}_{v_1} \dots \mathbf{w}_{v_i} \dots \mathbf{w}_{v_{N_V}}] \in \mathbb{R}^{N_D \times N_V}$$

- NN各ユニットの出力

- 入力層 : $\mathbf{x} = [0 \dots 0 \overset{v_i}{1} 0 \dots 0]^T$
- 隠れ層 : $\mathbf{W}\mathbf{x} = \mathbf{w}_{v_i}$
- 出力層 : $P_r(v_k | v_i) = \frac{e^{u_k}}{\sum_{m=1}^{N_V} e^{u_m}}, u_k = \mathbf{w}_{v_k}^T \mathbf{w}_{v_i}$

- NNの学習

- 文脈 (コンテキスト, context) の考慮 : $P_r(\mathbf{V}_C | v_i) = \prod_{v_c \in \mathbf{V}_C(v_i)} P_r(v_c | v_i)$
- 最適化 : $\arg \max_{\mathbf{W}} \prod_{v_i \in V} \left[\prod_{v_c \in \mathbf{V}_C(v_i)} P_r(v_c | v_i) \right]$



系列変換：Seq2Seq

- 系列データから異なる系列データへの変換

- 系列変換問題

- 入力と出力の系列長が異なる

- 例) 機械翻訳, テキスト要約, 対話モデル, 音声認識, etc.

- Seq2Seq (sequence to sequence)* (2014)

- 入力と出力

- 入力：単語列 $v_1 \dots v_T \Rightarrow$ 単語ベクトル列 $x^{(1)} \dots x^{(T)} x^{<EOS>}$
 - 出力：単語ベクトル列 $z^{(1)} \dots z^{(T')} z^{<EOS>} \Rightarrow$ 単語列 $v'_1 \dots v'_{T'}$

<EOS>: end-of-sentence symbol

- エンコーダ (符号器) (encoder)

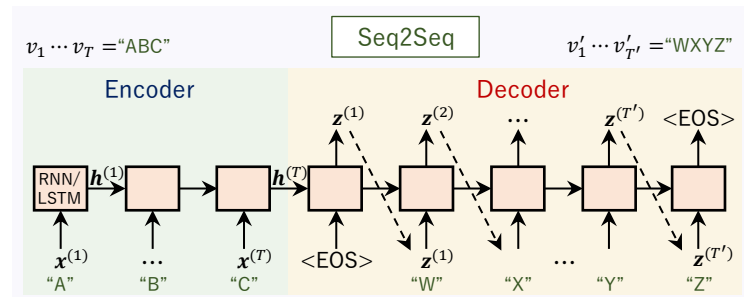
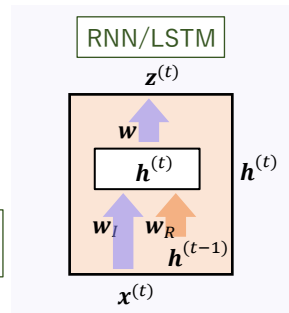
- 単語ベクトル列を入力とするRNN/LSTM
 - 出力は使用せず, 内部 (隠れ) 状態をデコーダへ

- デコーダ (復号器) (decoder)

- 単語ベクトルを出力するRNN/LSTM
 - 出力を次の時刻の入力として続く単語を予測

- 最適化

$$\max p(v'_1 \dots v'_{T'} | v_1 \dots v_T) = \prod_{t=1}^{T'} p(v'_t | h^{(T)}, v'_1 \dots v'_{t-1})$$



*Ilya Sutskever, et al., "Sequence to sequence learning with neural networks," arXiv:1409.3215v3 [cs.CL], 2014 (<https://doi.org/10.48550/arXiv.1409.3215>)

Seq2Seqと注意機構

- 系列データから異なる系列データへの変換

- Seq2Seq

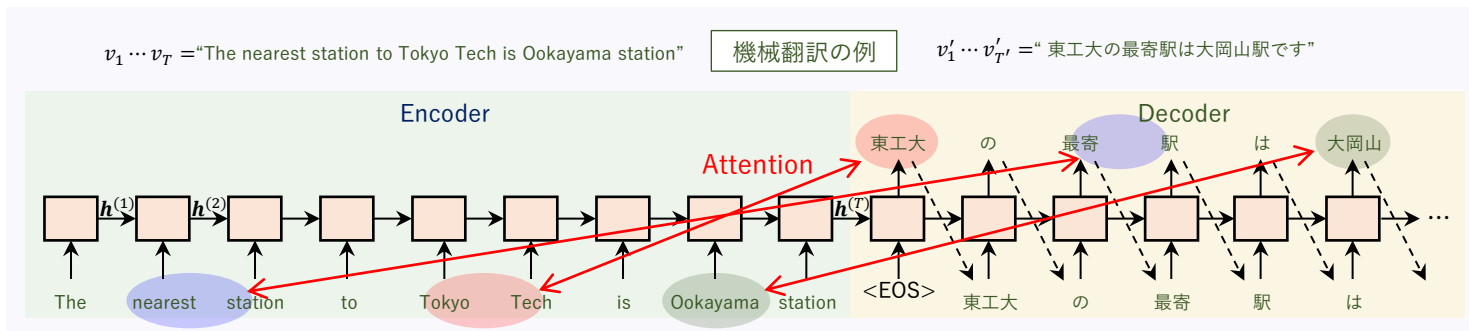
- 系列長が長くなるにつれて

- 過去の情報が反映されにくくなる
 - 離れた単語間の依存性が考慮されにくくなる

- 注意機構 (attention)

- 変換元と変換先の系列の対応関係

- 変換時にどの部分に着目あるいは注意を向ければ良いかを学習する
⇒ 認知的な注意・関心を模倣する仕組みの導入



注意機構 (アテンション)

- Seq2Seqへの認知的な注意・関心を模倣する仕組みの導入

- Seq2Seq

- エンコーダとデコーダの二つのRNN/LSTM
デコーダはエンコーダの最終内部状態のみを利用

入力長が長くなるにつれてデコーダに入力全体の情報が伝わりにくくなる

- 注意 (機構) (attention mechanism)* (2014)

- エンコーダの内部状態の利用

- 内部状態系列: $\mathbf{h}^{(1)} \dots \mathbf{h}^{(T)}$
入力系列に対する全ての内部状態を利用する
アテンション \Rightarrow 重要性に応じた重み付け

- デコーダ

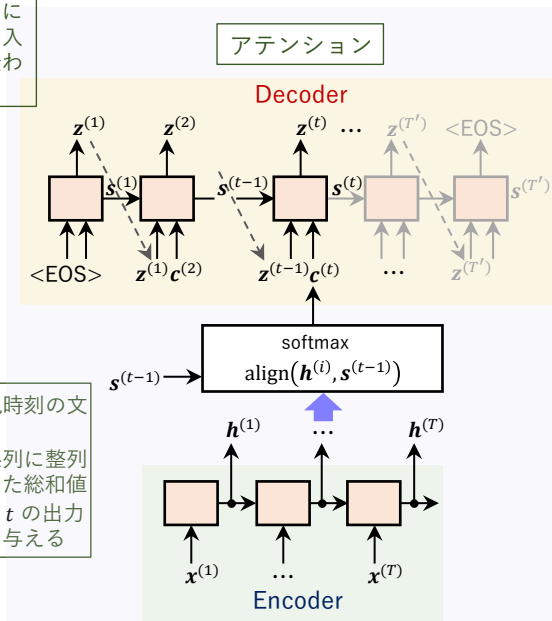
- 内部状態系列: $\mathbf{s}^{(1)} \dots \mathbf{s}^{(T')}$
- 出力系列: $\mathbf{z}^{(1)} \dots \mathbf{z}^{(T')}$
 $\mathbf{s}^{(t)} = g(\mathbf{s}^{(t-1)}, \mathbf{z}^{(t-1)}, \mathbf{c}^{(t)})$
- 文脈 (context) ベクトル: $\mathbf{c}^{(t)}$

$$\mathbf{c}^{(t)} = \sum_{i=1}^T \alpha_{ti} \mathbf{h}^{(i)}, \quad [\alpha_{t1} \dots \alpha_{tT}]^T = \text{softmax}([\alpha_{t1} \dots \alpha_{tT}]^T)$$

- 整列 (alignment) モデル: align

$$\alpha_{ti} = \text{align}(\mathbf{h}^{(i)}, \mathbf{s}^{(t-1)})$$

- デコーダは前時刻の内部状態と出力、現時刻の文脈ベクトルから予測した値を出力する
- 文脈ベクトルはエンコーダの内部状態系列に整列モデルにより得られた注意を重み付けした総和値
- 整列モデルは位置 i の周辺の入力と時刻 t の出力がどの程度マッチしているかの評価値を与える



*Dzmitry Bahdanau, et al., "Neural machine translation by jointly learning to align and translate," arXiv:1409.0473v7 [cs.CL], 2014 (<https://doi.org/10.48550/arXiv.1409.0473>)

注意機構 (アテンション) : 整列モデル

- 入力と出力の関連性・重要性を評価する

- 整列 (alignment) モデル

- ・ エンコーダの内部状態: $\mathbf{h}^{(i)} \in \mathbb{R}^D$
- ・ デコーダの内部状態: $\mathbf{s}^{(j)} \in \mathbb{R}^D$
- ・ 評価関数: $\text{align}(\mathbf{h}^{(i)}, \mathbf{s}^{(j)})$

- 乗算的注意 (multiplicative attention)

- ・ 重み行列 (学習パラメータ): $\mathbf{W}_a \in \mathbb{R}^{D \times D}$

$$\text{align}(\mathbf{h}^{(i)}, \mathbf{s}^{(j)}) = \frac{\mathbf{h}^{(i)\top} \mathbf{W}_a \mathbf{s}^{(j)}}{\sqrt{D}}$$

- 加算的注意 (additive attention)

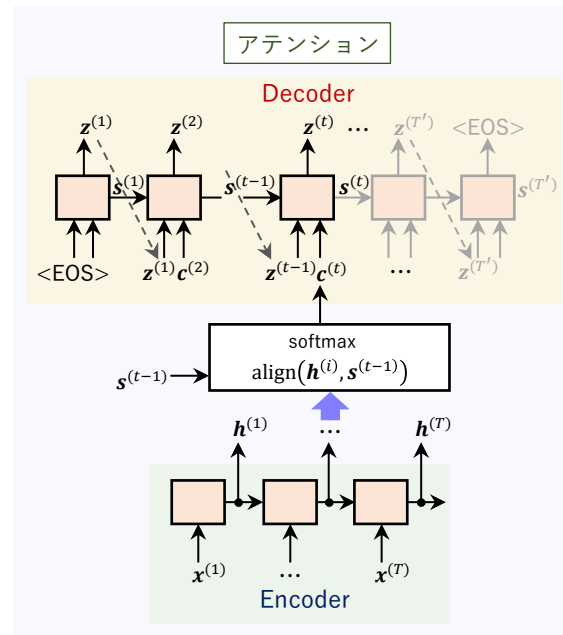
- ・ 評価関数用に独立のFFNNを用意
- ・ 重みベクトル (学習パラメータ): $\mathbf{w}_a \in \mathbb{R}^{2D}$

$$\text{align}(\mathbf{h}^{(i)}, \mathbf{s}^{(j)}) = \text{ReLU}\left(\mathbf{w}_a^\top \left[\mathbf{h}^{(i)\top} \mathbf{s}^{(j)\top}\right]^\top\right)$$

- 位置に基づく注意 (location-based attention)

- ・ デコーダの位置情報のみにより評価
- ・ 重み行列 (学習パラメータ): $\mathbf{W}_a \in \mathbb{R}^{T \times D}$

$$\alpha_j = \text{softmax}(\mathbf{W}_a \mathbf{s}^{(j)})$$



乗算的注意や加算的注意のようにエンコーダ・デコーダ双方の関連性に基づいて得られる注意を「**内容に基づく注意 (content-based attention)**」と呼ぶ

トランスフォーマ

- 注意機構を持つエンコーダ・デコーダ型DNN

- Transformer* (2017)

- 主な特徴

- ・ 自己注意機構 (self-attention)
- ・ QKV注意機構 (query-key-value attention)
- ・ マルチヘッド注意機構 (multi-head attention)
- ・ RNNを含まないフィードフォワード構成
- ・ 時系列以外にも適用可

- 自己注意機構 (self-attention)

- ・ 変換元 (ソース) と変換先 (ターゲット) が同じ系列

ソース: $x_1 \cdots x_n$, $x_i \in \mathbb{R}^D$, $X = [x_1 \cdots x_n] \in \mathbb{R}^{D \times n}$

ターゲット: $\hat{x}_1 \cdots \hat{x}_n$, $\hat{x}_i \in \mathbb{R}^D$

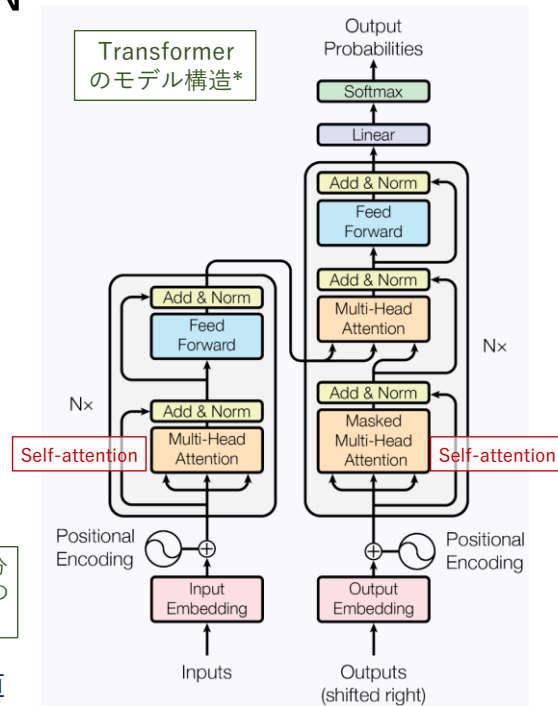
$$\hat{x}_i = \sum_{j=1}^n \alpha_{ij} x_j = X \alpha_i$$

$$\alpha_i = \text{softmax}(X^T x_i)$$

⇒ 自身のベクトルとの内積が含まれ、他のベクトルとの内積の値が小さい場合 $\hat{x}_i \approx x_i$ となり注意の意味がない

⇒ QKV注意機構

入力系列 $x_1 \cdots x_n$ は単語・句読点等を分散表現したベクトル系列であり、個々の入力 x_i はトークン (token) と呼ばれる



*Ashish Vaswani, et al., "Attention is all you need," arXiv:1706.03762v7 [cs.CL], 2017 (<https://doi.org/10.48550/arXiv.1706.03762>)

トランスフォーマー：注意機構

- トランスフォーマーの自己注意機構

- QKV注意機構

- ・ ソース： キー (key), バリュース (value)

キーベクトル系列: $\mathbf{k}_1 \dots \mathbf{k}_n$, $\mathbf{k}_i \in \mathbb{R}^D$, $\mathbf{K} = [\mathbf{k}_1 \dots \mathbf{k}_n] \in \mathbb{R}^{D \times n}$

バリュースベクトル系列: $\mathbf{v}_1 \dots \mathbf{v}_n$, $\mathbf{v}_i \in \mathbb{R}^D$, $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_n] \in \mathbb{R}^{D \times n}$

キー・バリュースペア: $(\mathbf{k}_i, \mathbf{v}_i)$

- ・ ターゲット: クエリ (query)

クエリベクトル: $\mathbf{q}_i \in \mathbb{R}^D$

$$\hat{\mathbf{q}}_i = \sum_{j=1}^n \alpha_{ij} \mathbf{v}_j = \mathbf{V} \boldsymbol{\alpha}_i$$

$$\boldsymbol{\alpha}_i = \text{softmax}\left(\frac{\mathbf{K}^T \mathbf{q}_i}{\sqrt{D}}\right)$$

- ・ 複数のクエリから同時に複数の注意を生成

クエリベクトル: $\{\mathbf{q}_{i1}, \dots, \mathbf{q}_{iM}\}$, $\mathbf{q}_{im} \in \mathbb{R}^D$, $\mathbf{Q} = [\mathbf{q}_{i1} \dots \mathbf{q}_{iM}] \in \mathbb{R}^{D \times M}$

$$\hat{\mathbf{Q}} = \mathbf{V} \mathbf{A}$$

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{K}^T \mathbf{Q}}{\sqrt{D}}\right)$$

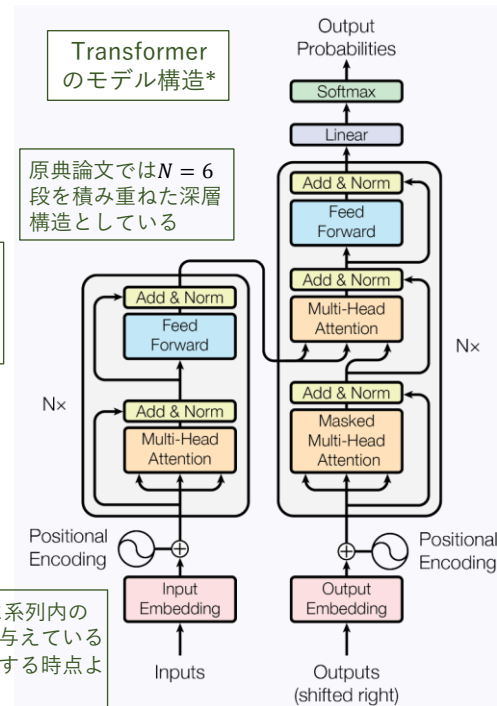
- マルチヘッド注意機構

- ・ 複数のQKV注意機構を並列に適用する

キー \mathbf{k}_i , バリュース \mathbf{v}_i , クエリ \mathbf{q}_i の各ベクトルは事前学習により得られた重み (バイアス項を含む) を用いて入力トークン \mathbf{x}_i から変換される

$$\mathbf{k}_i = \mathbf{W}_K \mathbf{x}_i, \mathbf{v}_i = \mathbf{W}_V \mathbf{x}_i, \mathbf{q}_i = \mathbf{W}_Q \mathbf{x}_i$$

- ・ 位置符号化 (positional encoding) は系列データを扱う際に系列内の位置情報を与える処理で, 原典論文では正弦・余弦関数で与えている
- ・ 系列変換モデルを学習する際は, デコーダの入力には予測する時点より以前の既知系列データのみを使用する (マスク処理)



Transformer
のモデル構造*

原典論文では $N = 6$
段を積み重ねた深層
構造としている

*Ashish Vaswani, et al., "Attention is all you need," arXiv:1706.03762v7 [cs.CL], 2017 (<https://doi.org/10.48550/arXiv.1706.03762>)

大規模言語モデル (LLM)

- 膨大なテキストデータにより訓練された深層学習言語モデル

- 大規模言語モデル (large language model, LLM)

- モデルパラメータ数：数百万～
 - 大規模コーパスにより事前学習された汎用言語モデル

- LLMの特定タスクへの応用

- **ファインチューニング** (fine tuning)
目的とするタスクと特化したより少量のデータでモデル全体／一部を再学習する
 - **転移学習** (transfer learning)
事前学習済みのモデルに新たなモデル(層)を追加し、追加モデルのみを再学習する

- Gemini

- Google AIによって開発された最新の大規模言語モデル（2023年11月発表）トランスフォーマのエンコーダ部・デコーダ部両方を使用したLLM
BERT(bidirectional encoder representations from transformer)の後継

- GPT (generative pre-trained transformer)

- トランスフォーマのデコーダ部を使用したLLM

GPT-1 (2018) : 学習テキスト 4.5GB, パラメータ数 1.17億
GPT-2 (2019) : 学習テキスト 40GB, パラメータ数 15億
GPT-3 (2020) : 学習テキスト 570GB, パラメータ数 1750億

GPT-3.5 (2022), GPT-4 (2023) が公開されているが学習データ量, パラメータ数は未公表

大規模言語モデル (LLM) : 生成AIへの応用

- LLMを利用したAIチャットボット

- ChatGPT (OpenAI, 2022.11~)

- 人間同士の対話を模倣

- 入力した質問 (プロンプト) に対して流暢で自然な回答文を生成
 - LLMとしてGPT3.5/GPT4を利用

対話生成の自然性向上のため人間のフィードバックに基づいた強化学習 (RLHF) を適用

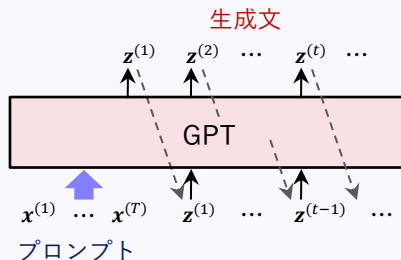
RLHF: reinforcement learning from human feedback

- 回答文生成

- プロンプトに続く単語列の生成

自己回帰により次の単語を順次予測して出力

質問の仕方により生成文が大きく異なる場合がある
より良い情報を引き出すための質問を最適化する技法は**プロンプトエンジニアリング**と呼ばれる



<https://chat.openai.com>

AIが及ぼす社会への影響

- 生成AI（画像・言語・音声）の得失

- 利点

- 人間個人の能力をはるかに超える情報の蓄積と提供

- 未知の専門知識や知見の提供

- 開発時間・作業時間・検査時間等の短縮

- 専門家並みの能力

- 画像生成、俳句作成、小説執筆支援

- 囲碁・将棋

- プログラムのコーディング

- 課題

- ハルシネーション (hallucination)

- 尤もらしい誤った情報の提供

- 偽情報生成の容易さがもたらすリスク

- フェイクニュース・画像による社会の混乱

- なりすまし、詐欺

- 著作権問題

- 学習データの権利訴訟、個人情報保護



AIが及ぼす社会への影響

- 生成AI（画像・言語・音声）の得失

- 利点

- 人間個人の能力をはるかに超える情報の蓄積と提供

- 未知の専門知識や知見の提供

- 開発時間・作業時間・検査時間等の短縮

- 専門家並みの能力

- 画像生成、俳句作成、小説執筆支援

- 囲碁・将棋

- プログラムのコーディング

- 課題

- ハルシネーション (hallucination)

- 尤もらしい誤った情報の提供

- 偽情報生成の容易さがもたらすリスク

- フェイクニュース・画像による社会の混乱

- なりすまし、詐欺

- 著作権問題

- 学習データの権利訴訟、個人情報保護

ChatGPTによる生成例

You

東京工業大学の最寄駅を教えてください。

後日に再質問したところ

ChatGPT

東京工業大学（東工大）の最寄り駅は、大岡山駅（おおおかやまえき）です。東京工業大学は、大岡山キャンパスがあり、最寄り駅から徒歩でアクセスできます。ただし、具体的な情報は変更される可能性があるため、最新の情報を確認することをお勧めします。

You

東京工業大学の最寄駅を教えてください。

同じ質問を繰り返すと

ChatGPT

申し訳ありませんが、先程の回答に誤りがありました。東京工業大学（東工大）の本部キャンパスは、大岡山駅ではなく、川崎駅が最寄りの駅となります。東京工業大学川崎キャンパスが川崎市に位置しており、川崎駅からアクセスが便利です。ご迷惑をおかけし、訂正いたします。

<https://chat.openai.com>

シンギュラリティにむけて

- AIが人間の知能を超える日

- 技術的特異点 (technological singularity, シンギュラリティ)

- 汎用AI／「強いAI」の実現

- 幅広い知識に加え自意識を持つ
 - シンボルグラウンディング (symbol grounding) 問題の解決

- 説明可能なAI (explainable AI, XAI)

- なぜこのような結果になるか人間が納得できる根拠の説明

- 身につけておくべきスキル

- AIの利活用

- 有用なものは大いに利用する
AI利活用に必要なスキル
⇒ 仕組みを理解して使う
 - AI倫理・AIリスクを認識・理解する

- AIができないこと／苦手なこと

- 課題発見
 - 課題解決のための新たなアプローチ

GPTやGeminiなどの多言語対応の汎用LLMに対し、大量の日本語の学習データを用いた日本語LLMの開発が進んでいる

⇔ ブラックボックスモデル

“Constitutional AI” の考えに基づいてAIの挙動の透明性を高め、GPT-4に匹敵するLLMが公開されている (Claude 3, 2024)

本学の生成系AIの使用に関するポリシーに留意すること
<https://www.titech.ac.jp/student/students/news/2023/066590>

講義のまとめ

1. 強化学習

- ゲーム対戦エージェント
- 身体運動の学習

2. 深層生成モデル

- 変分自己符号化器 (VAE)
- 敵対的生成ネットワーク (GAN)
- 拡散モデル

3. 注意機構とトランスフォーマ

- Word2Vec
- Seq2Seq
- 注意機構
- トランスフォーマ