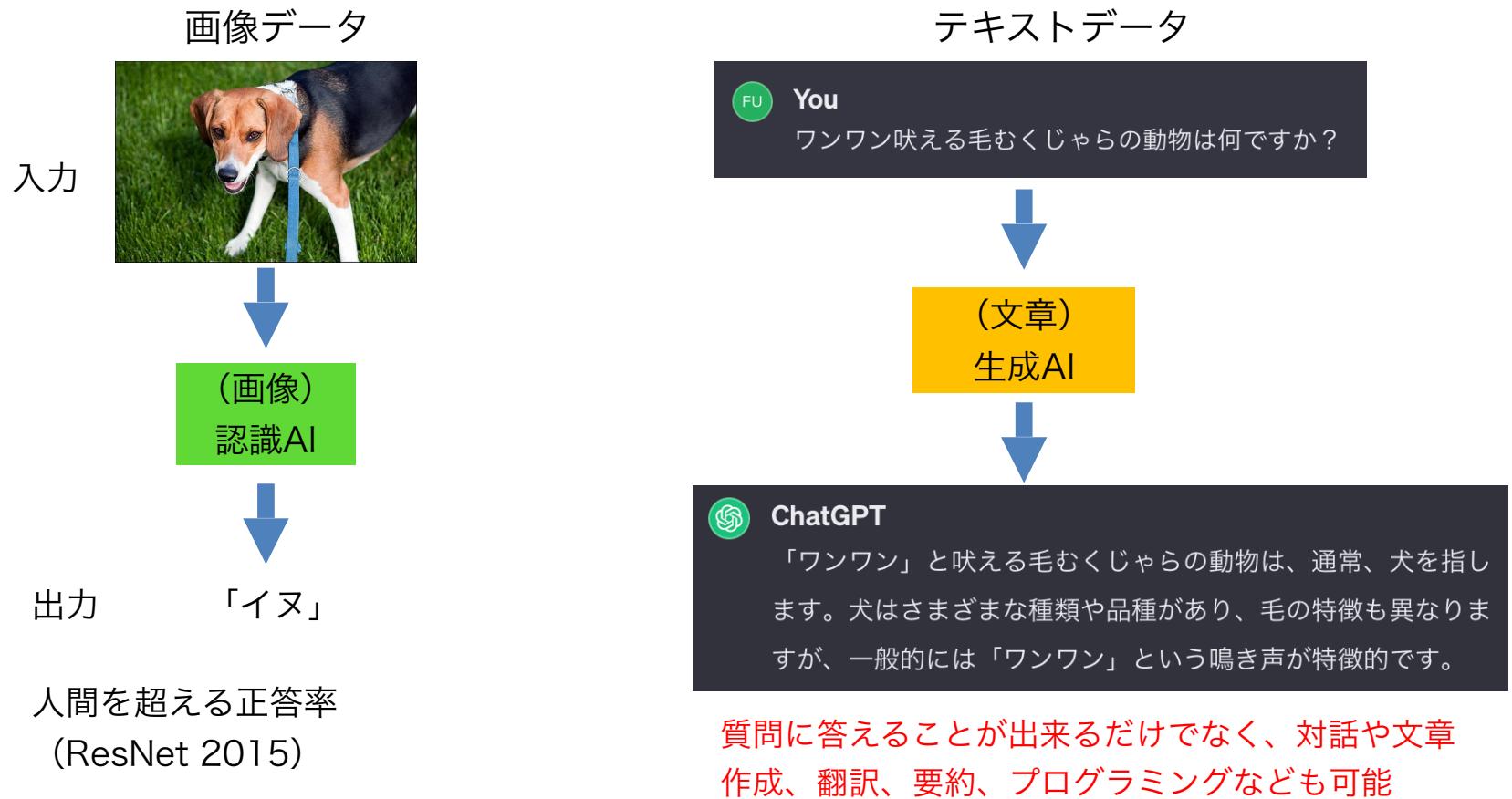


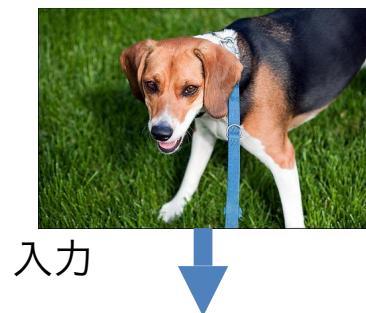
生成AIとは

従来の、正解の値を出力するだけの「認識AI」と違い、
文章や画像などを出力可能なAI



生成AIのタスクと出力

認識AIのタスク: 多クラス分類問題
「与えられた画像から正しいラベルを予測」



(画像)
認識AI

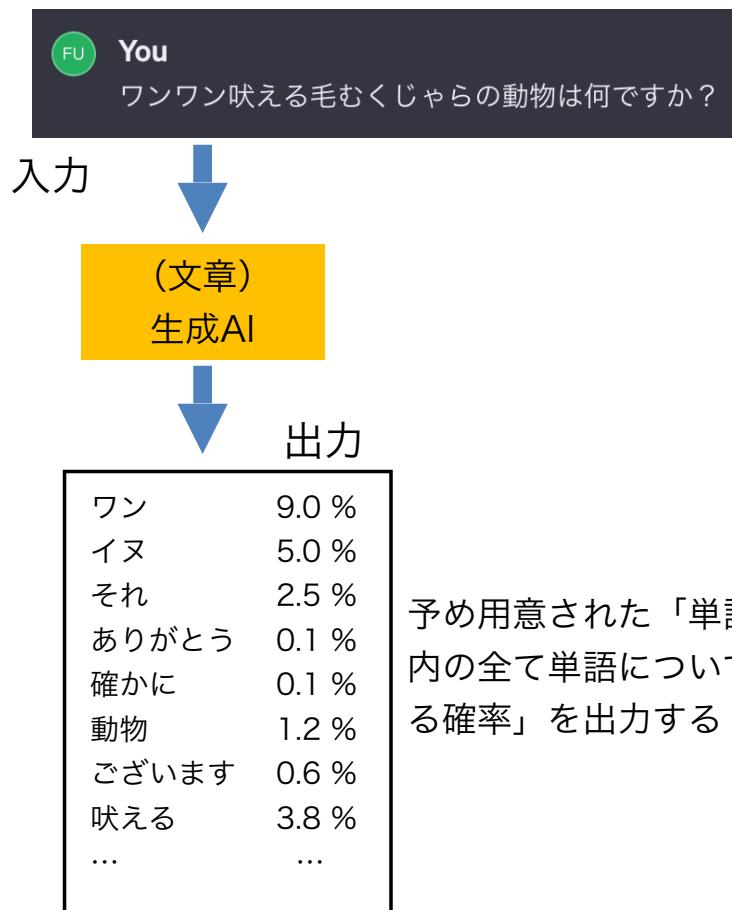
出力

ネコ	23 %
イヌ	52 %
鳥	5 %
自動車	1 %
飛行機	2 %
ポスト	1 %
...	...

予め用意された「ラベルのリスト」内の全てのラベルについて「そのラベルが正解ラベルである確率」を出力する

テキスト生成AIのタスク:

「与えられた文章から**次に来る単語**を予測」
(これを繰り返すことで文章を生成)



予め用意された「単語のリスト（語彙）」内の全て単語について「次に来る単語である確率」を出力する

認識AIと生成AIの違い

文章認識AI（文書分類の場合）：

入力: 文章

出力: カテゴリ（ジャンル）の確率分布

出力範囲小さい（スポーツ、エンタメ、スパムの判断）

埋込: する

アーキテクチャ: RNN、LSTM

文章生成AI:

入力: 文章

出力: 次にくる単語の確率分布

出力範囲とても大きい → 自由な出力に見える

埋込: する

アーキテクチャ: Transformer (2017)

画像認識AI（分類問題の場合）：

入力: 画像

出力: カテゴリの確率分布

出力範囲小さい（イヌ、ネコ）

埋込: しない

アーキテクチャ: CNN

画像生成AI:

入力: 確率分布（乱数、カテゴリ情報）

出力: 画像

埋込: する

アーキテクチャ: VAE、GAN、Diffusion (2015)

「認識」と「生成」で定性的な違いはない。
(そもそも、認識しなければ生成できないので、
このような分類には意味がない。)

「認識」と「生成」で全く異なる。
というより、「画像生成AI」が特殊。
生成モデルによる真の意味での「生成」。

文章生成AI

自然言語処理 (NLP)

人間が日常的に使っている言語をそのままコンピュータに入力して様々な処理をさせること。

- ・ 単語や文章が表す概念についてコンピュータに深く理解させる必要がある。
→ ディープラーニング、埋め込み表現
- ・ 言語は可変長の時系列データなので、時系列データ解析手法を用いる必要がある。
→ RNN、LSTM 等
→ 最近では Transformer が使われる。

時系列データ:
文章、音声、動画、株価変動

大規模言語モデル (LLM)

NLPの分野で発展した大規模な深層ニューラルネットワークモデル。

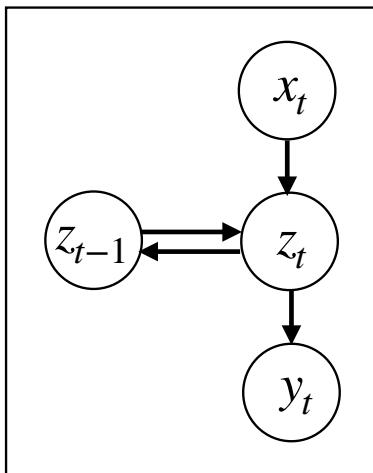
GPT、BERT、RoBERTa

RAG (Retrieval-Augmented Generation)

LLM + 検索機能

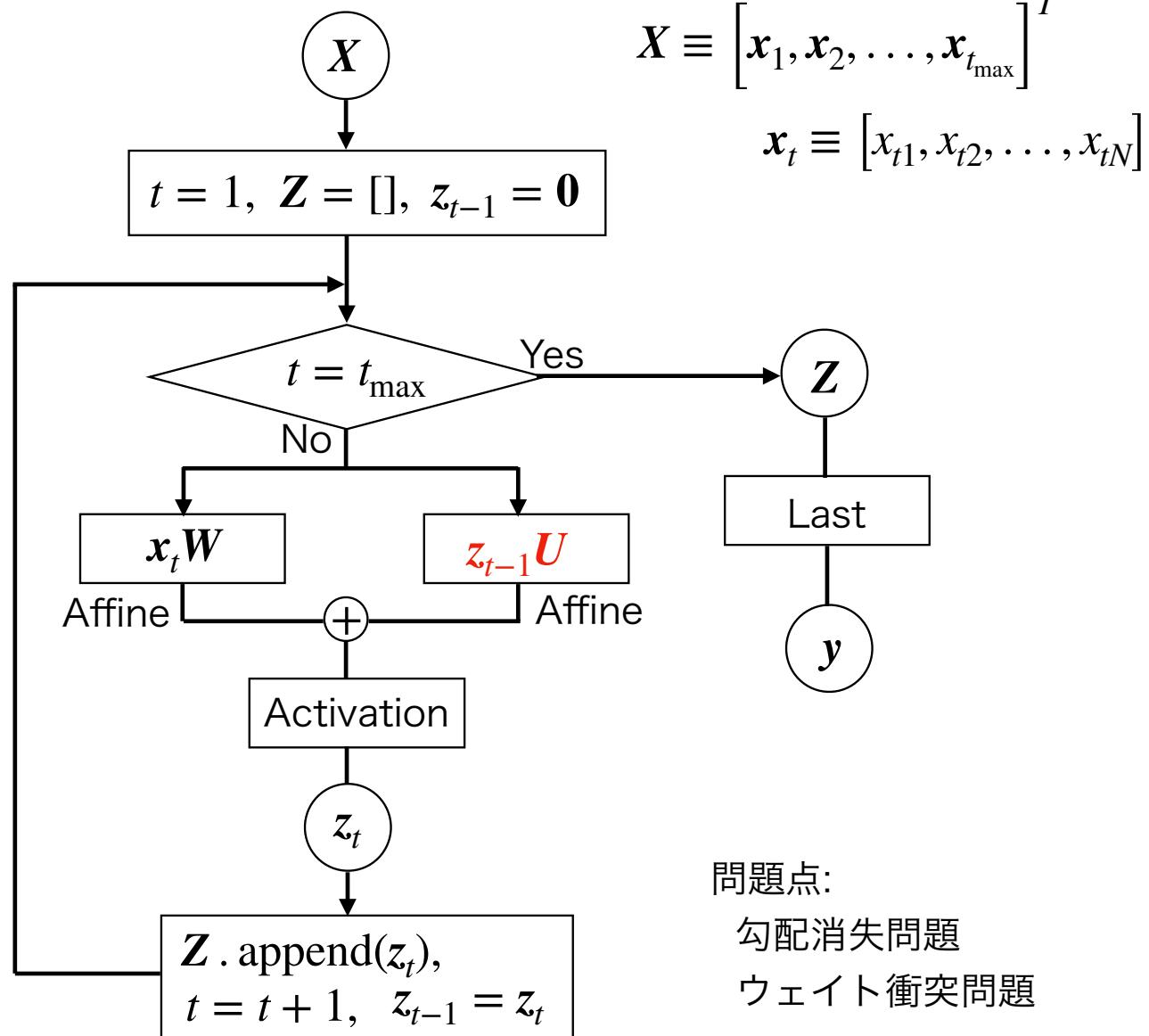
RNN

循環（回帰型）ニューラルネットワーク
(Recurrent Neural Network)



一つ前の系列情報を入力に組み込む

可変長データを扱える。
値の順序を考慮可能。



問題点:
勾配消失問題
ウェイト衝突問題

LSTM (2010 ~ 2017)

長・短期記憶 (Long Short Term Memory, 2007)

長期依存関係を学習可能

勾配消失問題、重み衝突問題を解決

$$i_t = \text{sigmoid} (x_t W_{\text{in}} + z_{t-1} U_{\text{in}})$$

$$o_t = \text{sigmoid} (x_t W_{\text{out}} + z_{t-1} U_{\text{out}})$$

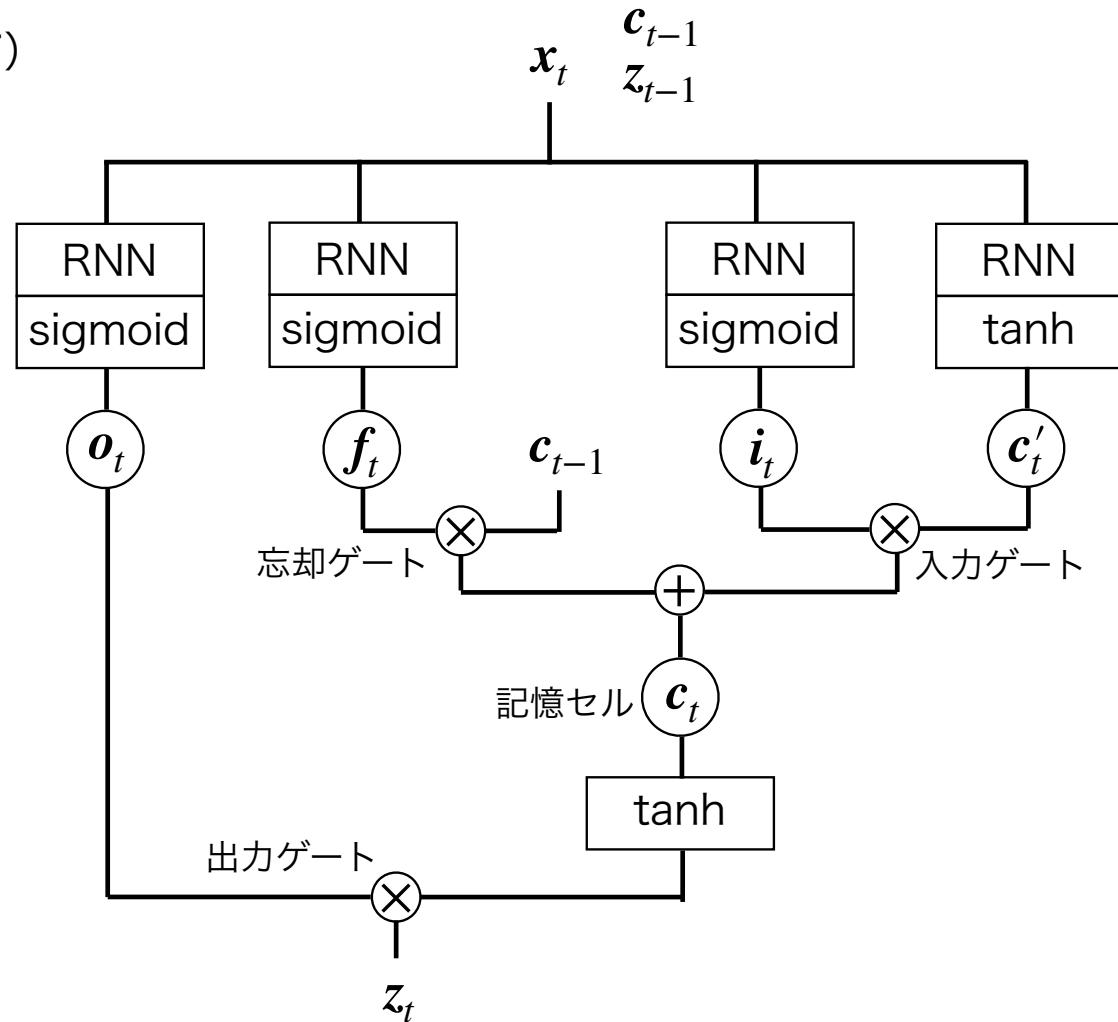
$$f_t = \text{sigmoid} (x_t W_{\text{forget}} + z_{t-1} U_{\text{forget}})$$

$$c'_t = \tanh (x_t W_{\text{cell}} + z_{t-1} U_{\text{cell}})$$

$$c_t = f_t \times c_{t-1} + i_t \times c'_t$$

$$h_t = o_t \times \tanh (c_t)$$

\times : アダマール積



Transformer (2017)

デコーダ構成:

入力文章から次の単語を予測するネットワーク

GPT (OpenAI, 2018)

エンコーダ構成:

入力文章に対する埋め込みベクトルを出力するネットワーク

BERT (Google, 2018)、RoBERTa (Facebook, 2019)

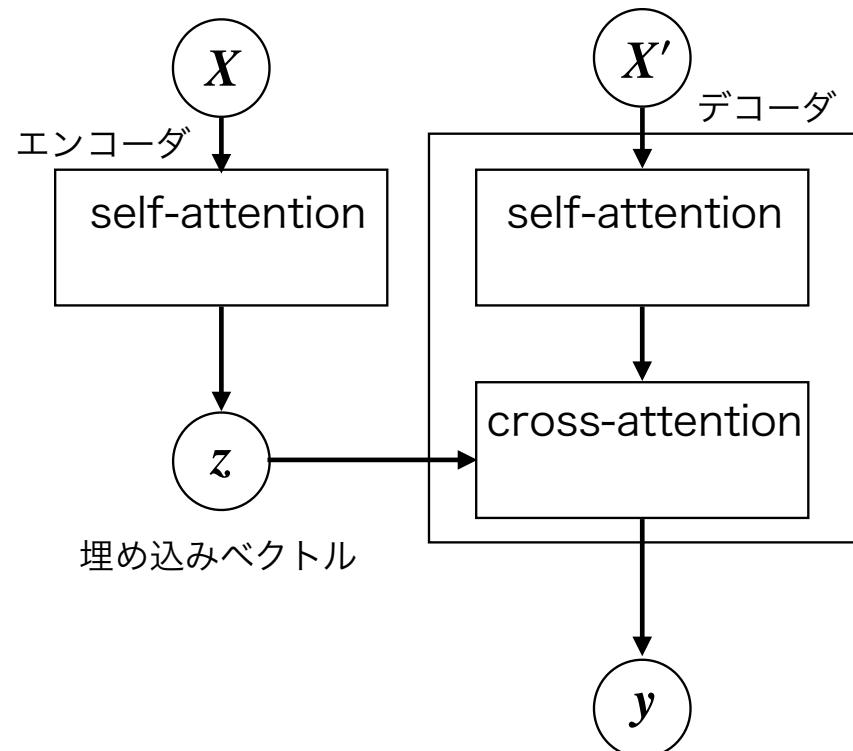
エンコーダ・デコーダ構成:

2種類の異なる入力を扱える（元々は翻訳器）

T5 (Google, 2020)

RNNやLSTMのように単語列をシーケンス処理するのではなく、一度に全ての単語の埋め込み処理を行う。

Transformer (エンコーダ・デコーダ構成)
の概形



Attention (self-attention)

“Attention is all you need” (Ashish Vaswani, 2017)

x_i 入力埋込ベクトル

$q_i = x_i W_q$ クエリ (照会ベクトル) 注文の仕方を学習

$k_i = x_i W_k$ キー (辞書の見出し語) メニューの表示方法を学習

$v_i = x_i W_v$ バリュー (見出し語の意味) 商品 자체を学習

出力埋込ベクトル (文脈も考慮した埋込ベクトル)

$$z_i = \text{attention} \{ q_i, \text{dictionary}(K, V) \}$$

$$= \sum_j^M f(q_i, k_j) v_j^T$$

重み (0 ~ 1)

与えられたタスクを上手くこなせる
ような埋込ベクトル z を出力できる
ように W_q 、 W_k 、 W_v を学習する。

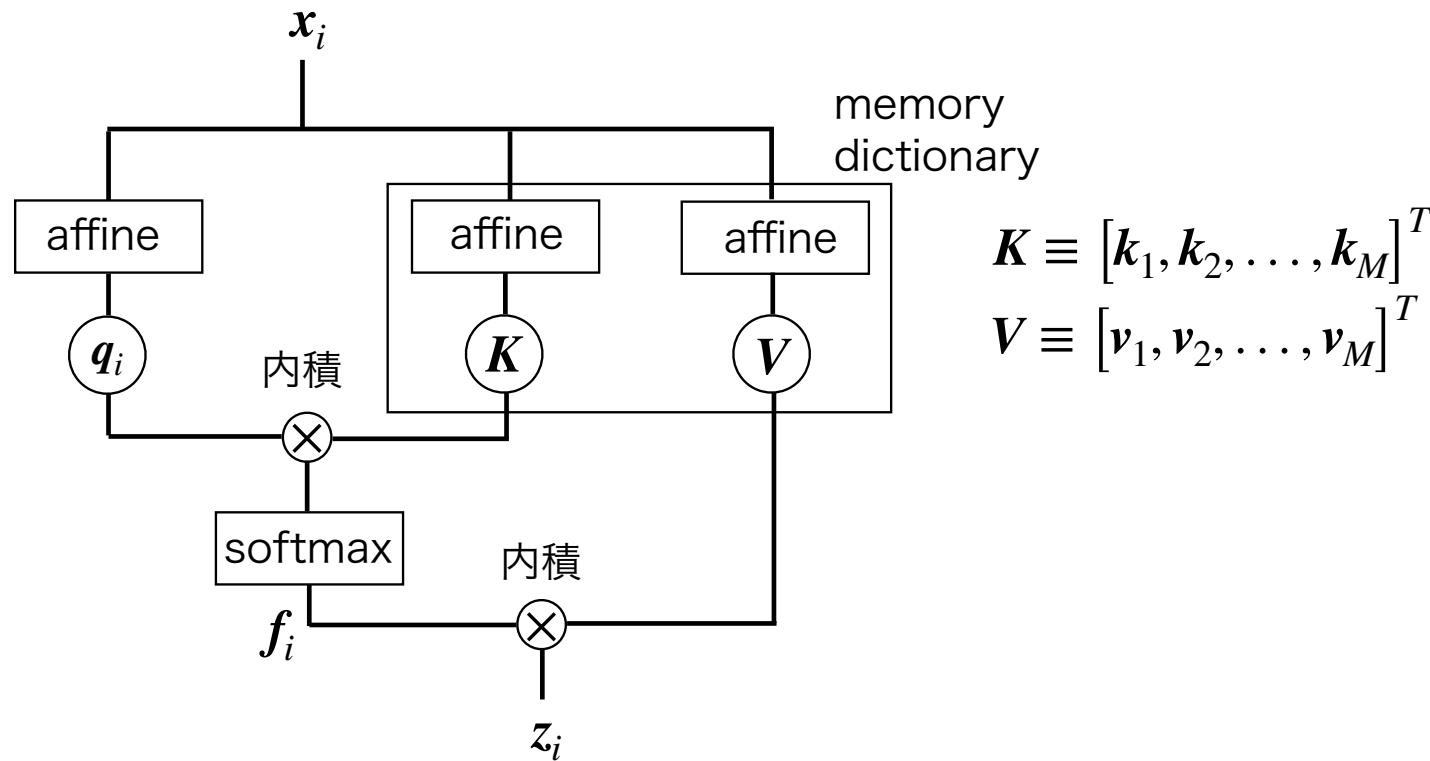
語順の情報はない (単語間の相関しかない) ので、
何らかの方法で入れてやる必要がある。

$$f(q_i, k_j) = \text{softmax}(s_{ij}) = \frac{e^{s_{ij}}}{\sum_{j'}^M e^{s_{ij'}}} \quad s_{ij} = \frac{1}{\sqrt{N}} q_i k_j^T$$

内積(類似度)を規格化したも

Attention (self-attention)

$$X \equiv [x_1, x_2, \dots, x_M]^T \quad \text{入力文章の埋込ベクトル}$$



大規模言語モデル (LLM)

トークン

テキスト生成AI：次に来る「単語」を予測 ← 「単語」とは？

(トークン)

1. 「単語」を定義し、文章を分割

- 単語（日本国語大辞典 50万）
多すぎる
新語の追加が難しい
 - 「銀行」、「へ」、「行き」、「お金」、「を」、「引き出す」、「。」
- 一文字（常用漢字約 2千 + ひらがな + カタカナ + 英数 + その他）
同じ文字が異なる使われ方をするので学習が困難
 - 「銀」、「行」、「へ」、「行」、「き」、「お」、「金」、「を」、「引」、「き」、「出」、「す」、「。」
- サブワード（数万）
丁度良いサイズ
「概念」とある程度1対1対応
(することを期待)
 - 「銀行」、「へ」、「行き」、「お金」、「を」、「**引き**」、「**出す**」、「。」
 - 連続で出現する頻度などに基づいて決める
(トークナイゼーション)

以下、「単語（トークン）」 = 「サブワード」とする

自己教師あり学習 (GPTの場合)

2. 教師データ、学習方法

「吾輩は猫である。名前はまだない。どこで生れたか頓と見当がつかぬ。…」

文章 次の単語

教師データ (w_1, w_2, \dots, w_K, t)

(吾輩, は)

(吾輩は, 猫)

(吾輩は猫, である)

(吾輩は猫である, 。)

(吾輩は猫である。, 名前)

(吾輩は猫である。名前, は)

(吾輩は猫である。名前は, まだ)

(吾輩は猫である。名前はまだ, ない)

一つの文章から教師データがたくさん取れる

(自己教師あり学習)

アノテーション作業不要

...

損失関数やウェイトの更新方法は画像の場合と同じ

単語の数値化

3. 単語の数値化

語彙 (数万)

index w

- 1. ネコ
- 2. リンゴ
- 3. 銀行
- 4. へ
- 5. ポスト
- 6. 行く
- 7. 金
- 8. 引く
- 9. 自動車
- 10. こたつ
- 11. 鳥
- ...

このままではDLに入力できない

w_2 : リンゴ

One-hot encoding

$$v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

数値にはなったが、
次元が無駄に大きすぎる

異なる単語が同じベクトルに
ならなければ何でも良いはず

埋め込み行列

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \dots]$$
$$\mathbf{u}_i = \mathbf{U}v_i$$

$$\mathbf{u}_2 = \begin{bmatrix} 0.356 \\ 0.246 \\ -0.224 \\ -0.105 \\ 0.542 \end{bmatrix}$$

(実際には数百次元)

成分を 0, 1 ではなく実数にすることで次元
を大幅に削減可能 (低次元のベクトル空間に
多くの単語を詰め込める)

単語間の概念上の距離を反映
(数値は学習で決まる)

LLMのアーキテクチャ (GPTの場合)

前処理

入力文章

$$\{w_1, w_2, w_3, \dots\}$$

↓ One-hot encoding

$$\{v_1, v_2, v_3, \dots\}$$

↓ 埋め込み

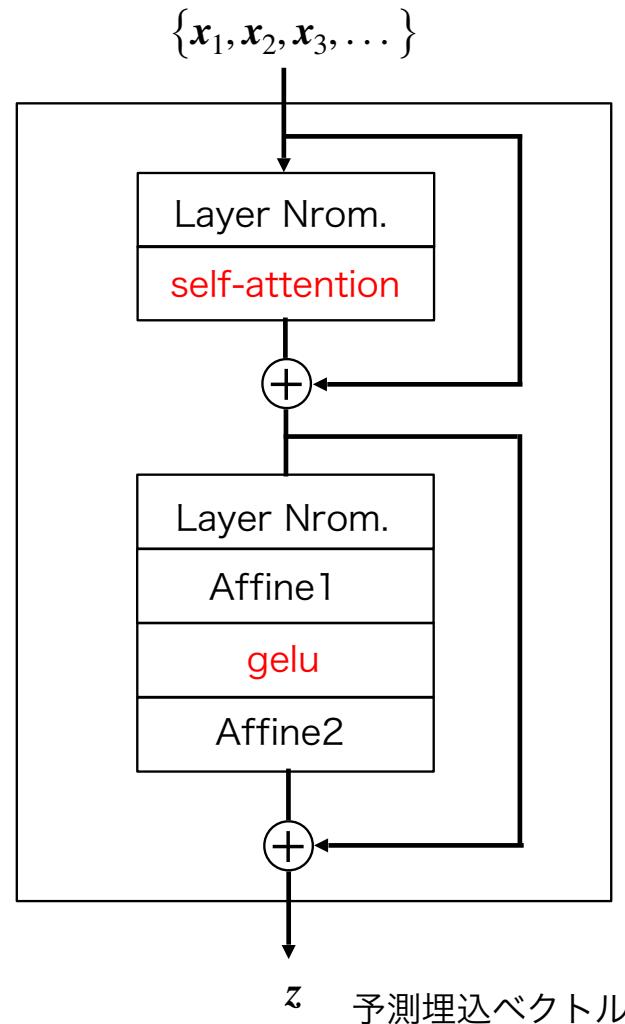
$$\{u_1, u_2, u_3, \dots\}$$

↓ 語順情報の追加

$$x_i = u_i + p_i$$

$$\{x_1, x_2, x_3, \dots\}$$

語順情報を含んだ
埋め込みベクトル



Transformer の
デコーダブロック

本当は過学習を防ぐための
ドロップアウト層もある。

このブロックを繰り返す
(GPT-3 では $N = 100$)

LLMのアーキテクチャ (GPTの場合)

次に来る単語の確率分布

$$\mathbf{y} = \text{softmax}(\mathbf{U}^T \mathbf{z}) \quad (\text{one-hot vectorと同じ次元})$$

$\mathbf{U}^T \mathbf{z}$ は語彙中の全ての埋め込みベクトルとの内積をとることを意味する
→ 各単語と \mathbf{z} との類似度に相当

softmax 関数により類似度が 0-1 の範囲に収まる
→ 確率として解釈ができるようになる

次に来る単語の予測

$$w = \text{argmax}(\mathbf{y}) \quad (\mathbf{y} \text{ の要素の中で最大のもののインデックス})$$

学習は交差エントロピー誤差

$$L = - \sum_{i=1}^M \sum_{k=0}^C t_{ik} \ln y_{ik} \quad \mathbf{W} \leftarrow \mathbf{W} + \eta \mathbf{X}^T (\mathbf{T} - \mathbf{Y})$$

$$\mathbf{y} = \begin{bmatrix} 0.001 \\ 0.237 \\ 0.112 \\ 0.039 \\ 0.058 \\ \vdots \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$



You

思考実験です。論理的に考えてください。「何でも切れるハサミ」で「絶対に切れない紙」を切ろうとするとどうなりますか？



ChatGPT

このような状況は「逆説」や「パラドックス」と呼ばれ、論理的な分析が難しい場合があります。与えられた条件では、「何でも切れるハサミ」で「絶対に切れない紙」を切ろうとすると、矛盾が生じます。

もし「何でも切れるハサミ」が本当に何でも切れる力を持っていると仮定すると、それは「絶対に切れない紙」も切ることができるのはずです。しかし、同時に「絶対に切れない紙」であるため、切ることは不可能であるとされています。

このような場面では、仮定や条件が矛盾しているか、または前提が相容れないかのいずれかがあります。このようなパラドックスはしばしば論理の限界や仮定の不備を指摘するために用いられ、議論の深化や理論の修正につながることがあります。

FU You

思考実験です。論理的に考えてください。右手で頭をさわると右手が痛い、右手でお腹をさわると右手が痛い。右手で足をさわっても右手が痛い場合、身体のどの部位に原因があると思いますか？



ChatGPT

この思考実験において、右手で頭やお腹、足を触ると右手が痛いという経験は通常ではありません。通常、身体の特定の部位を触ったときにその部位が痛むことが期待されます。しかし、このような状況が起こる場合、何らかの不自然な状態や異常が考えられます。

論理的に考えると、右手で異なる部位を触っても右手が痛むという状況は通常の生理的な反応ではなく、むしろ神経系や感覚系における何らかの問題を示唆している可能性があります。例えば、脳の特定の部分や神経経路に障害がある場合、触覚情報が正しく処理されず、誤った痛覚が生じる可能性があります。

ただし、これはあくまで架空のシナリオであり、実際の医学的な状況とは異なります。現実の医学的な問題は非常に複雑であり、具体的な症状や検査結果に基づいて専門家によって診断されるべきです。

文章生成AI

過小評価:

- 「検索が楽になる（余計な広告なしで検索可能）」
- 「翻訳の精度が上がる」
- 「より自然な対話が可能なチャットボット」

プログラムコードの作成等、生成AIにしかできないことがある
生成AIは、単語の使われ方から、単語が持つ「概念」をちゃんと抽出でき
ている

過大評価:

- 「人間と同じように文章を理解し、回答可能な汎用AI」
- 「人間の代わりにどんなタスクも任せられるようになる」
- 「新たな理論を発見したり、クリエイティブなことができる」

生成AIは「反射的、直感的」に、最も妥当と思われる回答を出力するだけ
推論は今のところあまり上手くできない

今後の展望

- ・ 推論能力の強化
 - 1. プロンプトエンジニアリング
 - ・ マジックワードの追加: “Let's think step by step”
 - ・ 入力テキストの分割（途中式を書かせるようなもの）
 - ・ （関数型プログラミング言語のソースコードを経由させる）
 - 2. フайнチューニング (GPT → ChatGPT)
 - 3. モデル自体の強化、改良
 - ・ 教師データやパラメータの増加による創発を期待
 - ・ 推論エンジン、知識ベースを組み込む（エキスパートシステム）
 - ・ その他（ネットワーク構造の見直し）
 - ・ ハルシネーション（幻覚）、「でっちあげ」回答をしないようにする
 - ・ 省エネ化（学習効率を上げる）
 - ・ マルチモーダル
 - ・ 説明性、解釈性の向上（XAI）

画像生成AI

画像生成AI

2005 オートエンコーダ

2013 変分オートエンコーダ

2014 GAN

2014 U-Net

2015 NIC

2015 Attention

2015 Diffusion

2016 スタイル変換

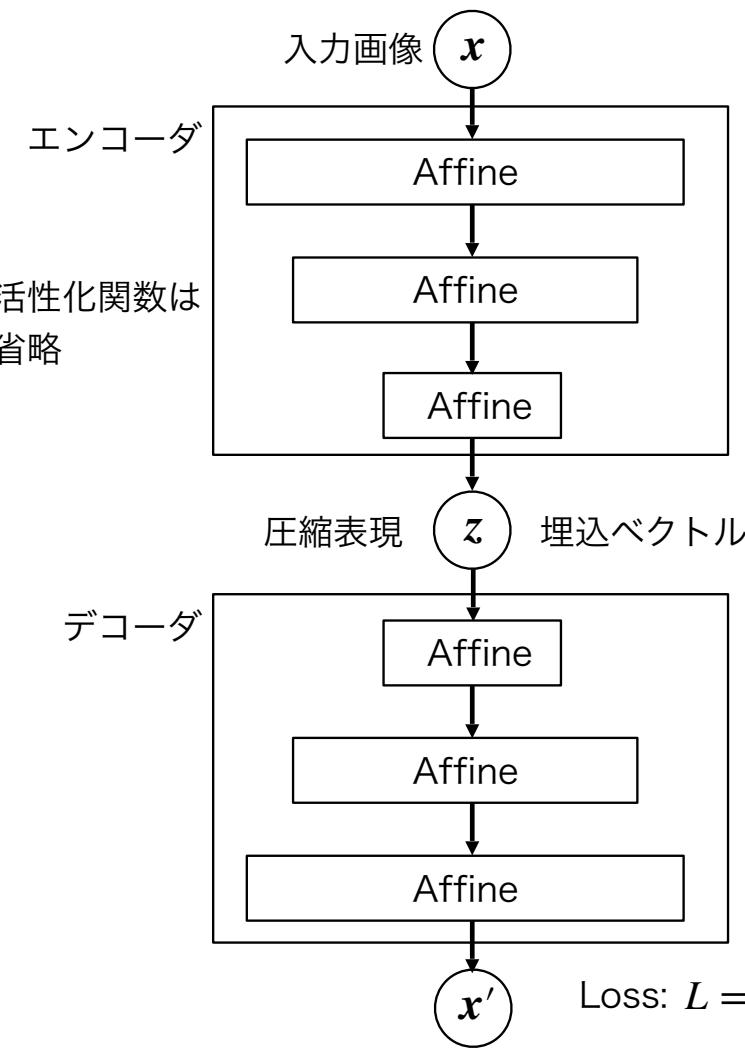
2017 Transformer

物体検出

セグメンテーション

YOLO

AE (オートエンコーダ)



オートエンコーダ（自己符号化器）

画像を一旦圧縮し、それを復元して元の画像に戻すだけのネットワーク。正解ラベルは自分自身なので（わざわざアノテーションする必要がないという意味で）教師なし学習である。

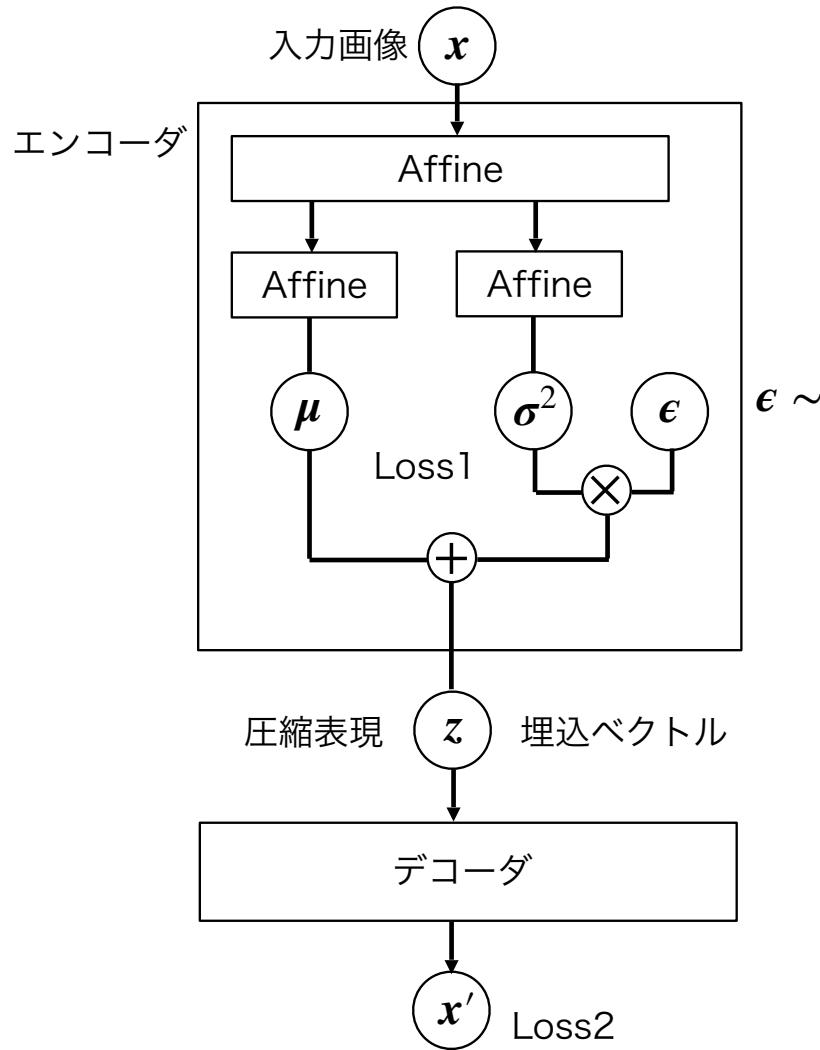
最も重要な点は、 z のサイズが元画像 x のサイズよりも小さい、ということである。これは、エンコーダによって入力画像が持っている情報が削られることを意味する。それでも何とかデコーダで画像を復元出来るようにするために、無駄な（なくても困らない）情報を見つけてそれらを優先的に削る事がエンコーダに求められることになる。

十分学習されたエンコーダを使うことで入力画像から（無駄の少ない）良い圧縮表現（埋込ベクトル）を得ることが可能となる。

AEは異常検知やノイズ除去などにも使われる

$$\text{Loss: } L = - \sum_{i=1}^N [x_i \ln x'_i + (1 - x_i) \ln(1 - x'_i)]$$

VAE (変分オートエンコーダ)



埋込ベクトル z が $z \sim N(z | \mu(x), \sigma^2(x))$ に従うように、AEのエンコーダに制限を課す。これにより画像は埋込ベクトル空間では正規分布する。

エンコーダで作られる埋込ベクトルは、多少のノイズが乗ってもデコーダで再構築可能なものとなる。
また、デコーダはノイズの乗った圧縮表現から入力画像を再構築することを学ぶ。

このようにして十分学習されたデコーダに、 z の代わりに乱数 $N(0,1)$ を入力すると、（それっぽい）画像を新たに生成することが出来るようになる。

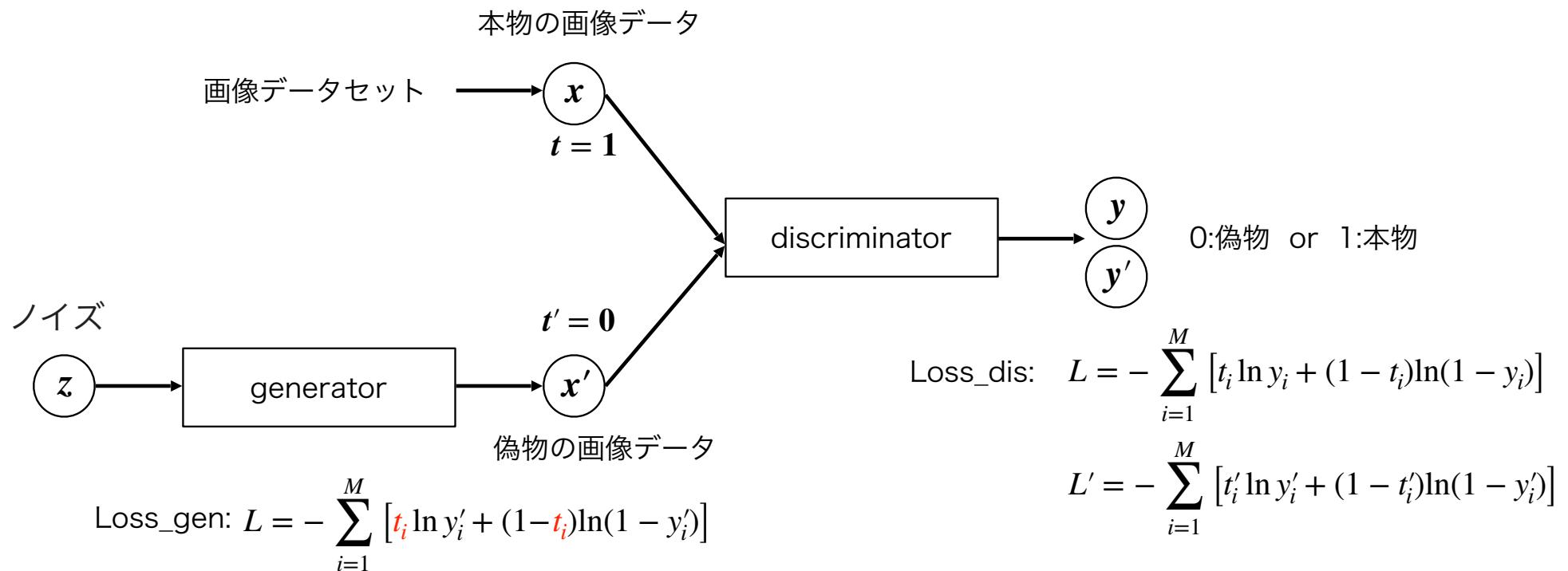
$$\text{Loss1: } L = -\frac{1}{2} \sum_{j=1}^K \left(1 + \ln \sigma_j^2 - \mu_j^2 - \sigma_j^2 \right) \quad \begin{array}{l} \text{潜在口ス} \\ \text{を保証するため} \end{array}$$

$$\text{Loss2: } L = - \sum_{i=1}^N \left[x_i \ln x'_i + (1 - x_i) \ln (1 - x'_i) \right] \quad \text{再構築口ス (AEと同じ)}$$

GAN (Generative Adversarial Network)

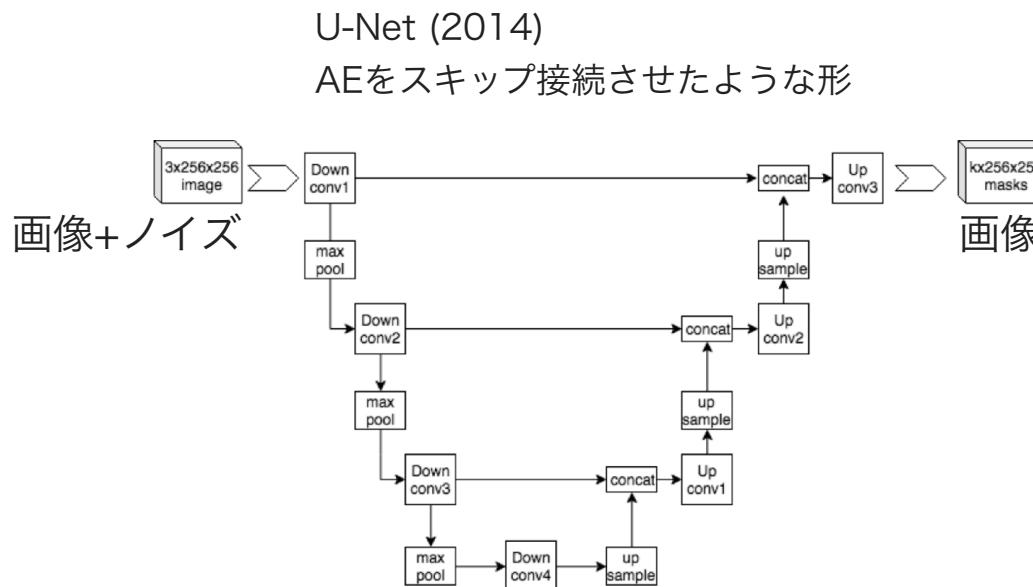
敵対的生成ネットワーク:

生成器（ジェネレータ）と識別器（ディスクリミネータ）の2つのネットワークを競合させながら学習

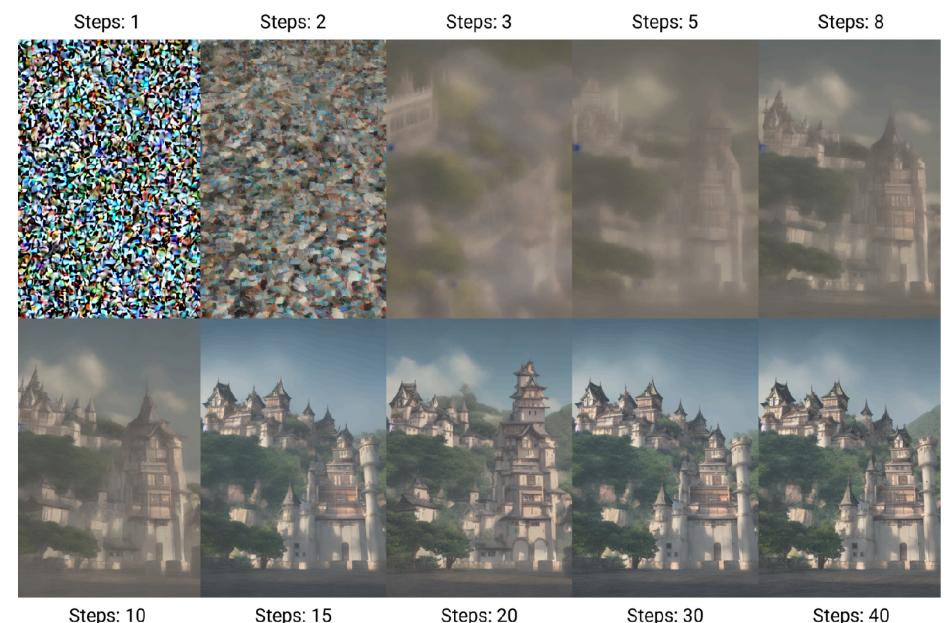


Diffusion model

Denoising Diffusion Probabilistic Models では、ノイズ除去ネットワークを学習。VAEを多段組にしたものに相当。



<https://ja.wikipedia.org/wiki/U-Net>



https://en.wikipedia.org/wiki/Diffusion_model

このように訓練したU-Netに完全なランダムノイズを入力することでVAEと同様に画像を生成することが可能。計算は遅いがVAEよりクオリティの高い画像が得られるらしい。

Diffusion model

Google

Imagen (2022-05) use T5 language model

Imagen 2 (2023-12)

Veo (2024-05) video generation

OpenAI

DALL · E2 (2022-04)

DALL · E2 (2023-10)

Sora (2024-02) video generation

Stability AI (イギリス)

Stable Diffusion (2022-08)

Stable Diffusion 3 (2024-02)

Midjourney (イギリス)

Midjourney (2022)

Diffusion probabilistic model (DPM)

Variational diffusion model

Score-based model

Denoising

Denoising DPM

Denoising diffusion implicit model

Improved denoising DPM

マルチモーダル

認識AIと生成AIの違い

文章認識AI（文書分類の場合）：

入力: 文章

出力: カテゴリ（ジャンル）の確率分布

出力範囲小さい（スポーツ、エンタメ、スパムの判断）

埋込: する

アーキテクチャ: RNN、LSTM

文章生成AI:

入力: 文章

出力: 次にくる単語の確率分布

出力範囲とても大きい → 自由な出力に見える

埋込: する

アーキテクチャ: Transformer (2017)

画像認識AI（分類問題の場合）：

入力: 画像

出力: カテゴリの確率分布

出力範囲小さい（イヌ、ネコ）

埋込: しない

アーキテクチャ: CNN

画像生成AI:

入力: 確率分布（乱数、カテゴリ情報）

出力: 画像

埋込: する

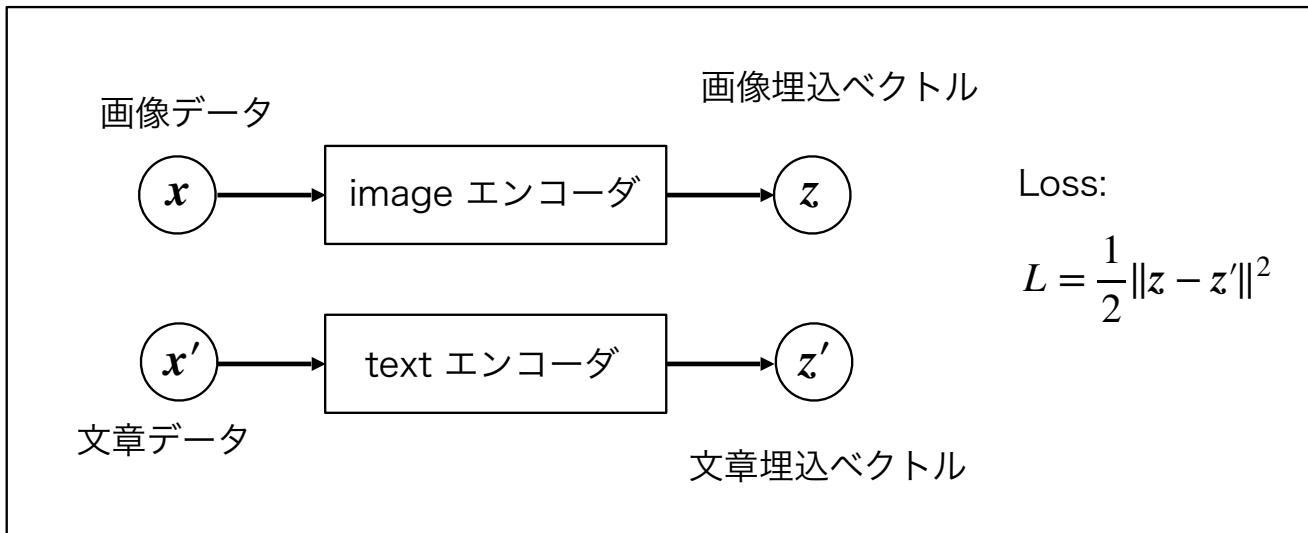
アーキテクチャ: VAE、GAN、Diffusion (2015)

「認識」と「生成」で定性的な違いはない。
(そもそも、認識しなければ生成できないので、
このような分類には意味がない。)

「認識」と「生成」で全く異なる。
というより、「画像生成AI」が特殊。
生成モデルによる真の意味での「生成」。

CLIP (OpenAI, 2021)

画像生成AIと文章生成AIの橋渡しのための共通埋込ベクトルの学習。
インターネット上から膨大な数の「画像とキャプションのセット」を取ってきて、
以下のような学習を行う。



この共通埋込ベクトルから画像を復元するデコーダを作れば
文章から画像を生成することができるようになる。
(画像からのキャプション生成は NIC という手法が前からある。)

XAI

解釈可能性の意義

- 医療、金融、政治、法律などの業界では、AIのブラックボックス性は特に問題視される

AIの研究開発の原則の策定

OECDプライバシーガイドライン、同・セキュリティガイドライン等を参考に、関係ステークホルダーの参画を得つつ、研究開発に関する原則を国際的に参照される枠組みとして策定することに向け、検討に着手することが必要。

研究開発に関する原則の策定に当たっては、少なくとも、次に掲げる事項をその内容に盛り込むべき。

① 透明性の原則

AIネットワークシステムの動作の説明可能性及び検証可能性を確保すること。

② 利用者支援の原則

AIネットワークシステムが利用者を支援するとともに、利用者に選択の機会を適切に提供するよう配慮すること。

③ 制御可能性の原則

人間によるAIネットワークシステムの制御可能性を確保すること。

④ セキュリティ確保の原則

AIネットワークシステムの頑健性及び信頼性を確保すること。

⑤ 安全保護の原則

AIネットワークシステムが利用者及び第三者の生命・身体の安全に危害を及ぼさないように配慮すること。

⑥ プライバシー保護の原則

AIネットワークシステムが利用者及び第三者のプライバシーを侵害しないように配慮すること。

⑦ 倫理の原則

ネットワーク化されるAIの研究開発において、人間の尊厳と個人の自律を尊重すること。

⑧ アカウンタビリティの原則

ネットワーク化されるAIの研究開発者が利用者等関係ステークホルダーへのアカウンタビリティを果たすこと。

総務省

解釈可能性に関する研究

- ・解釈可能モデリング最初から解釈可能性を考慮しながらモデルを作ること
- ・XAI (eXplainable AI) (Google) Responsible AI

定義:

「予測以外の追加情報をモデルから抽出する技術の総称」

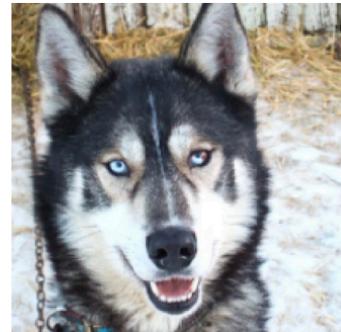
IG (Integrated Gradients)

LIME (Local Interpretable Model-agnostic Explanations)

SHAP (SHapley Additive exPlanations)

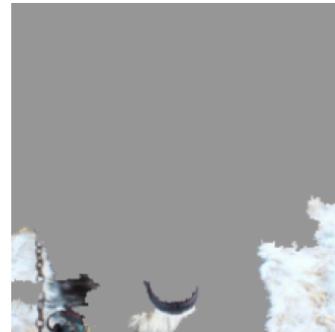
狼とハスキー犬の画像識別問題

ハスキー犬の画像



AIはこれを狼と誤判断

(a) Husky classified as wolf



(b) Explanation

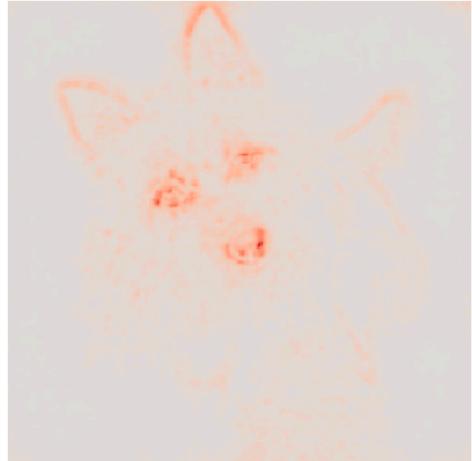
狼と判断した根拠を示す画像
(LIMEで抽出)

arXiv:1602.04938v3 [cs.LG] 9 Aug 2016

予測にどの特徴量が大きく寄与しているのかを可視化可能。
しかし、AIが何故その特徴量を重要視したのかは依然不明。

XAIの注意点

意図的に異なる解釈へ誘導することが可能

DLへの入力 画像データ	Original Image	Manipulated Image	視覚的に殆ど知覚できない 撮動が加えられた画像データ (DLの出力は同じ)
抽出された 「説明マップ」			
			任意の説明マップへ誘導 "this explanation was manipulated"

Ann-Kathrin Dombrowski, arXiv:1906.07983v2 [stat.ML] 25 Sep 2019