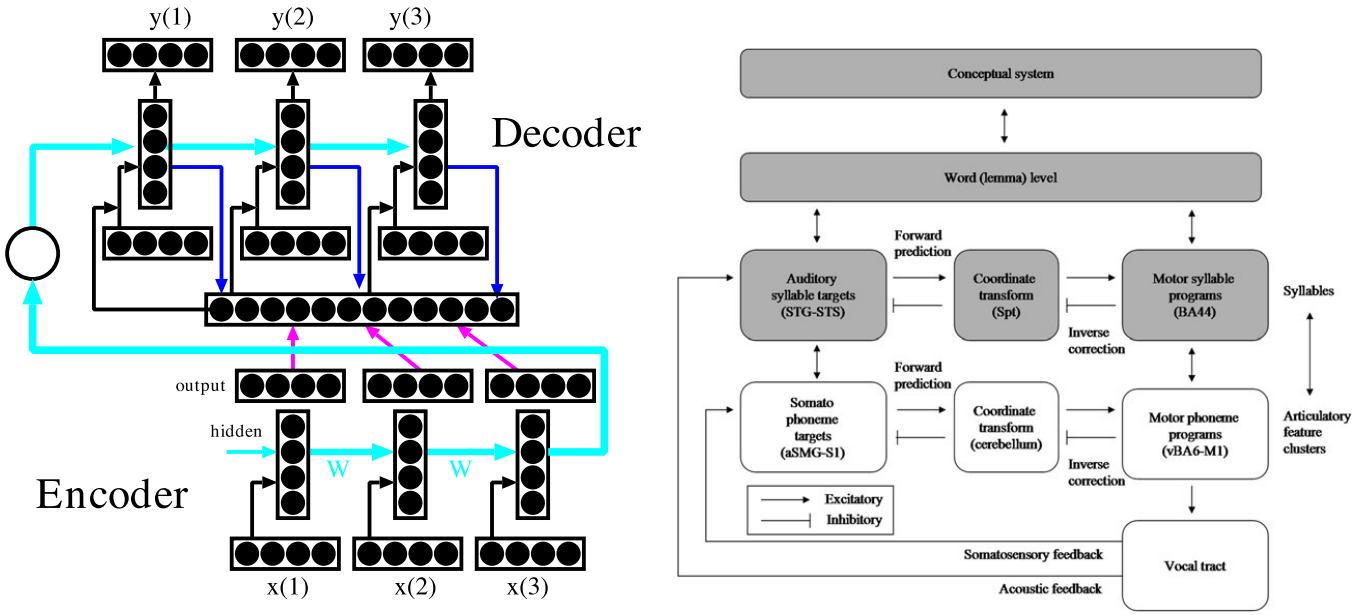
注意機構の補足説明 cnps2023 演題 2 大門他

Shin Asakawa

March 18, 2023

I. 符号化器-復号化器モデル (Seq2Seq モデル)

- 深層学習における注意とは，重要度を表す重みベクトルとみなしうる (勝者占有回路，winner-take-all circuit)。
- 画像中画素や文中の単語など，ある要素を予測・推論するために，他の要素とどれだけ強く相関しているかを注意ベクトルによって表現，
- 注意ベクトルで重み付けした値の和を対象への注目度の近似値とする



左: 提案モデルの大まかな大脳皮質への対応図。左脳皮質の大まかな位置関係を模した図。符号化器部分は，周シルビウス裂後部に対応し，前頭系 (44 野，図中最左のマル印) を介して運動系へ情報が伝達される。

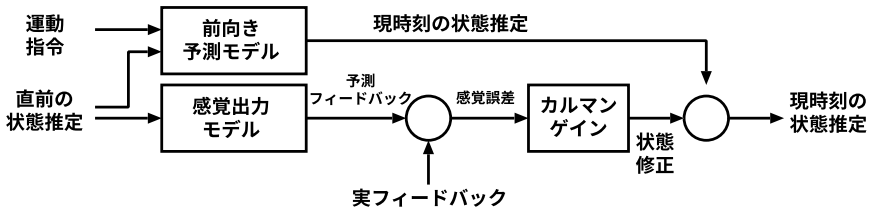
右: HSFC (Hickok,2000) の模式図。左右両図の大まかな対応関係に注目されたい。

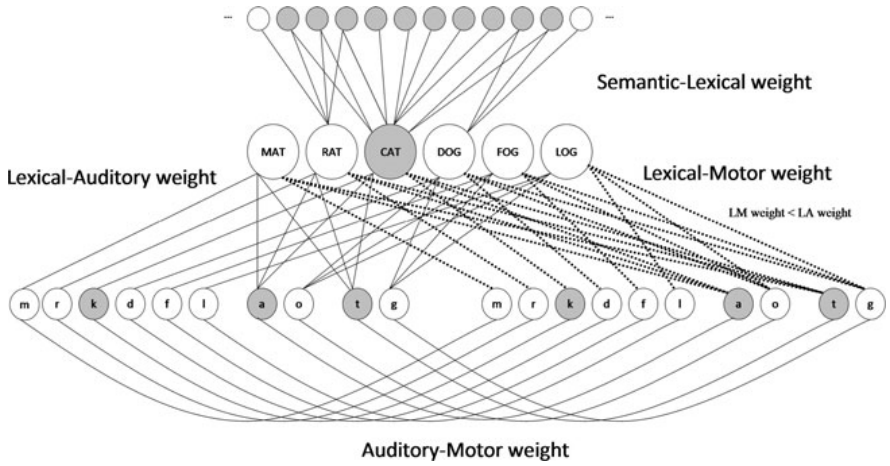
I. 符号化器-復号化器モデル (Seq2Seq モデル) continued.

- **seq2seq** モデルは言語モデリングの分野で生まれた (Sutskever, et al. 2014)。
- 入力系列 (ソース) を新しい配列 (ターゲット) に変換することを目的としており，両系列は可変長の長さを持つことができる。
- 例としては，テキストまたは音声の複数言語間の機械翻訳，質疑応答の対話生成，あるいは文法木への構文解析などがある。

seq2seq モデルは通常，符号化器と復号化器のアーキテクチャを持ち，以下のように構成される。

- **符号化器 encoder** は，入力系列を処理，情報を **固定長** の文脈ベクトル (文埋め込みまたは「思考」ベクトルとも呼ばれる) に圧縮する。 この表現は，**全体** のソース系列の意味の良い要約であることが期待される。
- **復号化器 decoder** は，変換された出力を発するために，文脈ベクトルで初期化される。





2. 翻訳モデルにおける注意

- 符号化器の最後の隠れ状態から単一の文脈ベクトルを構築するのではなく，文脈ベクトルとソース入力全体との間にショートカットを作成する。
- これらのショートカット接続の重みは，各出力要素ごとにカスタマイズ可能
- 文脈ベクトルが全入力系列にアクセスできる間は，忘れる心配はない。
- ソースとターゲットの間のアライメントは，文脈ベクトルによって学習され，制御される。 基本的に文脈ベクトルは 3 つの情報を用いる。

1. 符号化器側の中間層状態 (ソース)
2. 復号化器側の中間層状態 (ターゲット)
3. ソースとターゲットのアラインメント(配置情報) すなわちどの位置の情報に着目すべきかを決定する

3. 注意の種類

- **文脈ベースの注意 Content-base attention** (Graves2014): $\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \cos(\mathbf{s}_t, \mathbf{h}_i)$
- **加算的注意 Additive attention** (Bahdanau2015): $\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a [\mathbf{s}_t; \mathbf{h}_i])$
  Loung2015 論文では，“concat”,Vaswani2017 論文では “additive atttention” として言及されている注意のこと
- **位置ベースの注意 Location-Base attention** (Luong2015): $\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \mathbf{s}_t)$
  これにより，ソフトマックス配置(アライメント)がターゲット位置のみに依存するように単純化される。
- **一般化注意 General attention** (Luong2015): $\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{s}_t^\top \mathbf{W}_a \mathbf{h}_i$
  $\mathbf{W}_a$ は，注意層の訓練可能な重み行列
- **内積型注意 Dot-Product attention attention** (Luong2015): $\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{s}_t^\top \mathbf{h}_i$
- **規格化内積型注意 Scaled Dot-Product attention** (Vaswani2017): $\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \frac{\mathbf{s}_t^\top \mathbf{h}_i}{\sqrt{n}}$
  注：尺度因子を除いて内積型注意に酷似する；ここで、n はソース中間層状態の次元。 BERT および GPT で用いられている。

$[\cdot;\cdot]$ は 2 つのベクトルの連接を表す。

4. 定義

長さ n のソース系列 $\mathbf{x}$ と，長さ m のターゲット系列 $\mathbf{y}$ を考える。 太字はベクトルであることを示す。

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]$$
$$\mathbf{y} = [y_1, y_2, \ldots, y_m]$$

復号化器ネットワークは，位置 t の出力語に対して隠れ状態 $\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_t), t \in [1, m]$ であり，文脈ベクトル $\mathbf{c}_t$ は入力系列の隠れ状態の和で配置(アライメント) の得点で重み付けをする。

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \mathbf{h}_i \qquad \text{出力のための文脈ベクトル } y_t$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) \qquad 2 \text{ つの単語 } y_t \text{ 及び } x_i \text{ を配置}$$

$$= \frac{\exp(\text{score}(\mathbf{s}_{t-1}, \mathbf{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\mathbf{s}_{t-1}, \mathbf{h}_{i'}))} \qquad \text{アラインメント得点のソフトマックス}$$

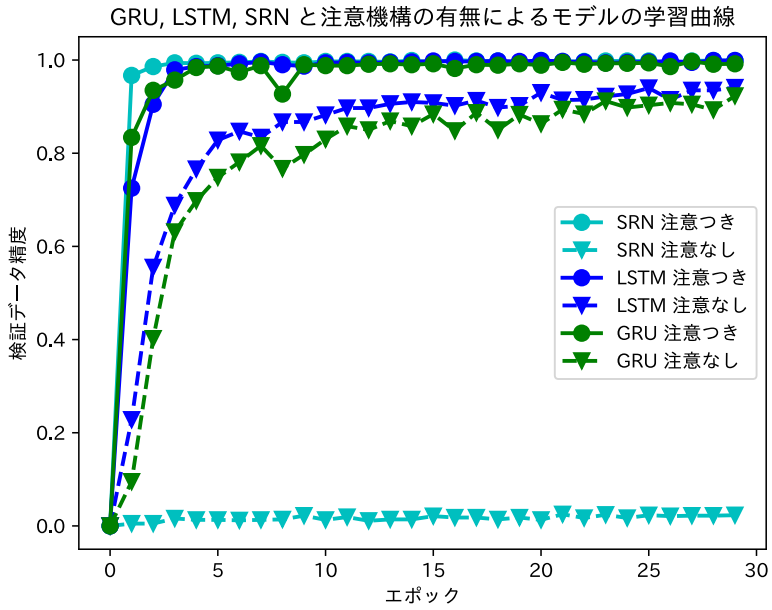
5. アラインメント関数

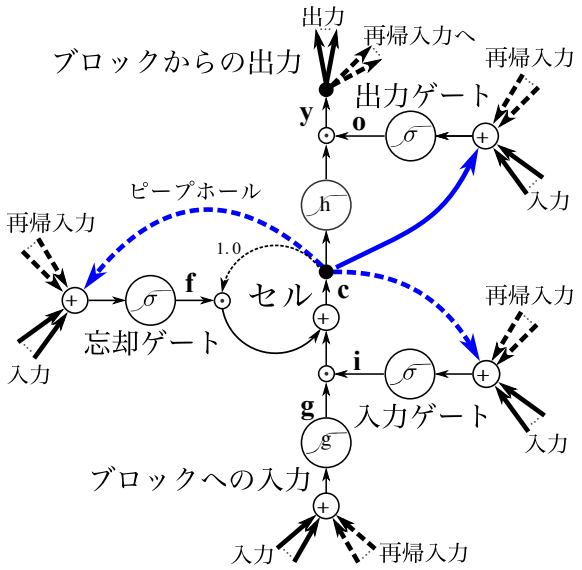
アライメントモデルは，位置 i の入力と位置t の出力の対 (y_t, x_i) に，それらがどれだけ一致しているかに基づいて得点 $\alpha_{t,i}$ を付与する。 $\alpha_{t,i}$ の集合は，各出力に対して各ソース中間層状態をどの程度考慮すべきかを定義する重みである。 Bahdanau 論文では，アライメント得点 α は単一の中間層を持つフィードフォワードネットワークによって計量化され，このネットワークはモデルの他の部分と同時に学習される。 得点関数は，非線形活性化関数として $\tanh$ が使用されていることを考えると，以下のような形となる:

$$\text{score}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \left[\mathbf{s}_t; \mathbf{h}_i\right])$$

ここで，$\mathbf{v}_a$ と $\mathbf{W}_a$ は共にアライメントモデルで学習される重み行列である。

6. Building block モデルごとの性能





- SRN without 注意以外のモデルでは，学習が可能であることが分かる。
- このことから，注意成分と RNN 成分とで，いずれが speech errors に関連しているのかを見定めたい，というリサーチクエスチョンが提起できる。本発表では，このことに焦点を当てた。
  - ***モデル0***: *全てのパラメータを微調整*
  - ***モデル1***: *注意機構を固定して，GRU 側を微調整*
  - ***モデル2***: *GRU 側を固定して，注意を微調整*