MULTIPLICATIVE INTERACTIONS AND WHERE TO FIND THEM

ICLR 2020

Presenter: Jaeyoung Cheong

1. Introduction

• 저자는 **Multiplicative Interaction**(이하 MI)가 ▲ Language Modeling (Conditional Statement) 등과 같은 알고리즘을 만드는 데 적합하고, ▲ 더 일반적으로 이것은 네트워크 내에서 맥락적 정보(Contextual Information)을 통합하는 효과적인 방법으로서 적합하다고 가정한다.

• 이 페이퍼는 MI 자체에 대해 탐구한 뒤, 실험에서 제안하는 MI가 통합되었을 때 Reinforcement Learning, Sequnce Modeling에서 유의미한 성능 향상이 있음을 보인다.

• 저자의 <MI가 맥락 정보를 통합하는 효과적인 방법>이라는 가정과 위 실험 결과는 **일관적**이다: MI를 적절한 방식으로 사용하는 것은 more date-efficient learning, better generalization, and stronger performance 를 이끌어내는 function class [1]에 대해 **more inductive bias** [2]를 제공할 수 있다.

1. Introduction

[CONTRIBUTIONS]

- to re-explore multiplicative interactions and their design principles
- to aid the community's understanding of other models through them
- to show their efficacy at representing certain solutions
- to empirically apply them to large scale sequence modeling and ~~(reinforcement learning)~~ problems, where we demonstrate *state-of-the-art* results.

2. Multiplicative Interactions

Q. How to combine two different streams of information ?

[Notation and Background]

- Two input variables: $x \in R^n, z \in R^m$
- Goal: to model an $f_{target}(x, z) \in R^k$ that entails some interaction between two variables.
    - x, z might be arbitary hidden activations, different input modalities (비전, 텍스트 등)

[Conventional method: Concatenation]

- We typically use $f(x, z) = W[x; z] + b$.
    - ,where [x;z] represents the concatenation of x and z.
    - ,where $W \in R^{(m+n)*k}$ and $b \in R^k$ are learnable parameters

[Proposed Method: Multiplicative Interactions]

- Authors propose $f(x, z) = z^T W x + z^T U + V x + b$
    - ,where $W \in R^{m*n*k}$ that is a 3D weight tensor
    - , where U, V are 2D weight matrices
    - , where b is a 1D vector

Authors: "We posit that this specific form, while more costly, is **more flexible, providing the right inductive bias** to learn certain families of functions that are of interest in practice"

2. Multiplicative Interactions

저자는 앞에서 살펴본 MI가 다양한 variantions로 기존 연구들에서 사용되어 왔다고 말한다.

1. Hypernetwork [3]

- Main network의 weights를 another network에서 generation

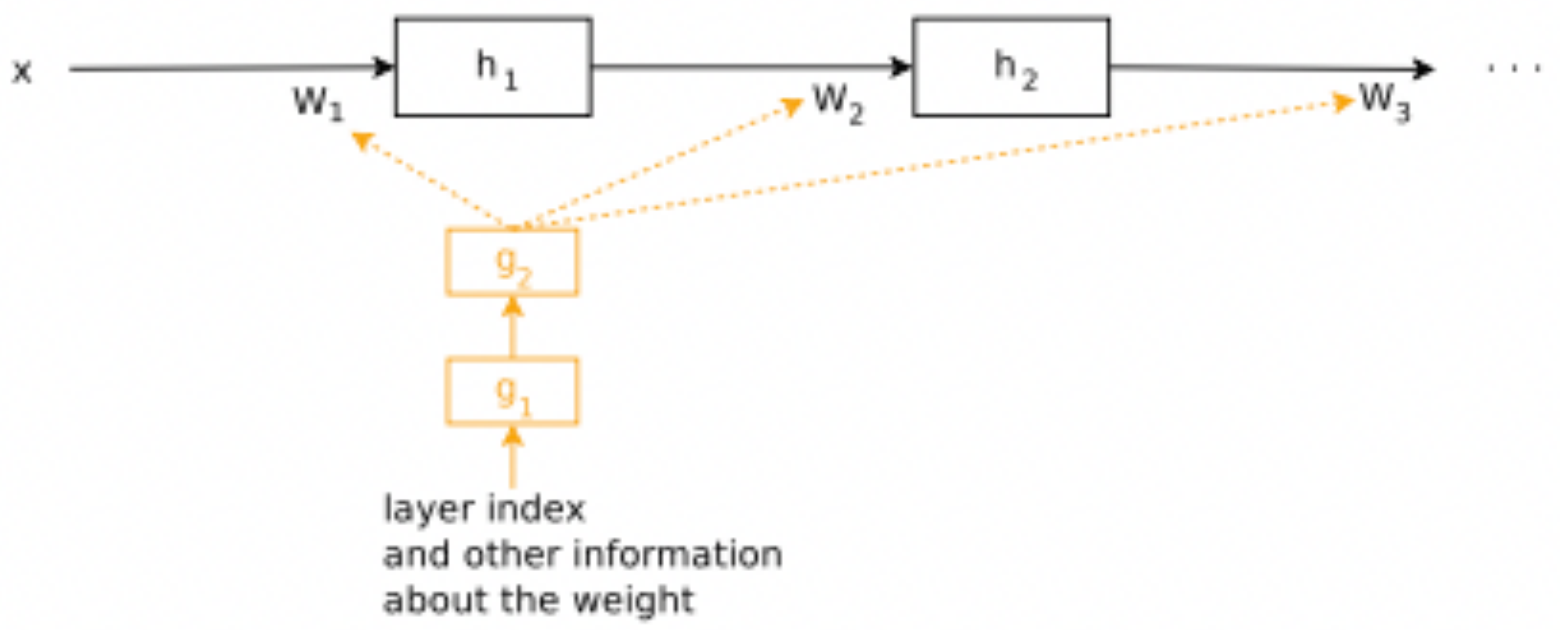


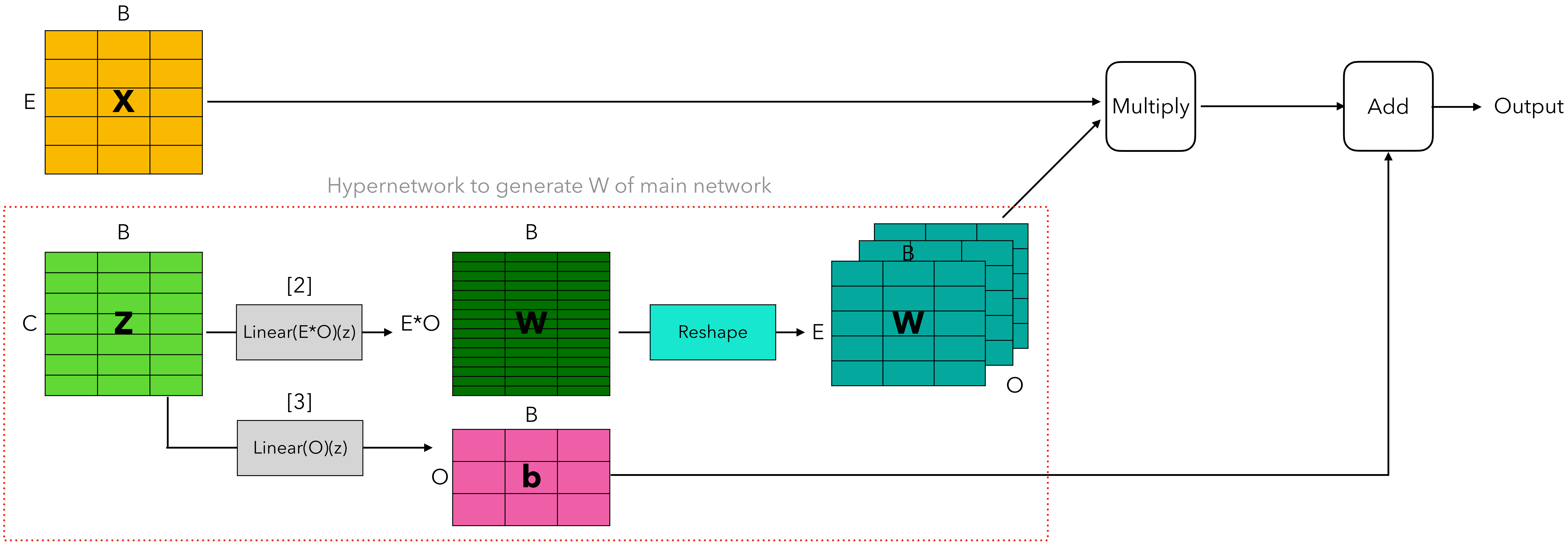
Figure 1: A hypernetwork generates the weights for a feedforward network. Black connections and parameters are associated the main network whereas orange connections and parameters are associated with the hypernetwork.

- This architecture is not $f(x;\theta)$ but $f(x;g(z;\phi))$. In the case where f, g are affine. Such a Network is exactly equivalent to the MI.
  - The MI that $f(x,z) = z^TWx + z^TU + Vx + b$ described above can be decomposed: $W' = z^TW + V, b' = z^TU + b$.
    - Then, the paraphrased MI that $f(x,z) = W'x + b'$
      - ,where W' is the generated weight matrix and b' is the generated bias from some hypernetwork.

2. Multiplicative Interactions

B: Batch size
E: Input size
C: Context size
O: Output size

[1] $f(x,z) = z^T W x + z^T U + V x + b$
[2] $W' = z^T W + V$
[3] $b' = z^T U + b$
[4] $f(x,z) = W'x + b'$



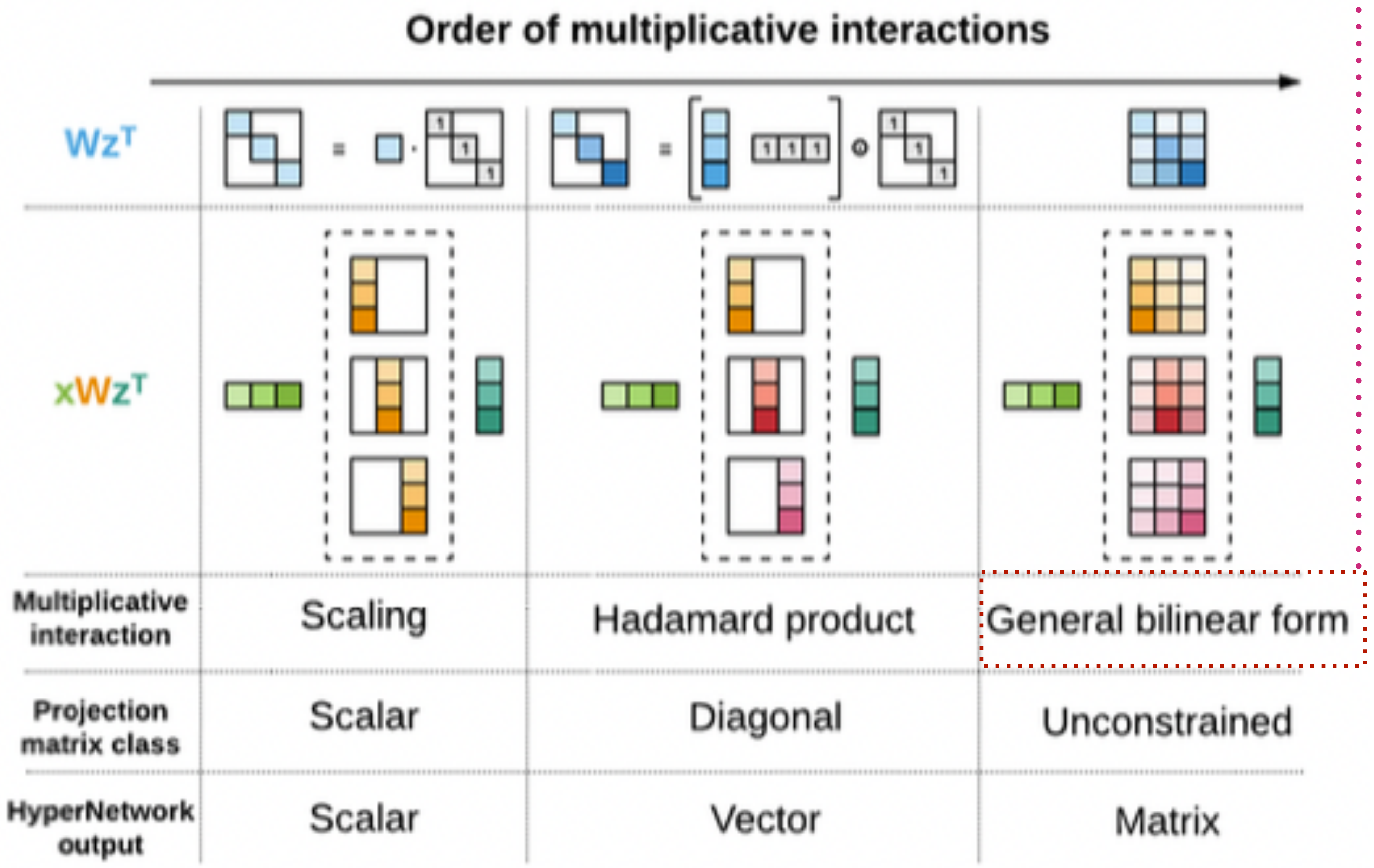
2. Multiplicative Interactions

2. Diagonal Forms and Gating Mechanisms

- Consider the diagonal approximation to the projected $W'(=z^T W + V)$
  - Authors: Multiplying with $W' = diag(a_1, ..., a_n)$ in $z^T W'$ can be implemented efficiently as $f = a \odot x$
    - This form resembles commonly used gating methods: (ex. Gated Linear Unit: $f(x, z) = x \odot \sigma(z)$)
    - It can be viewed as a hypernetworkd as well, where $z^t W$ represents the function generating parameters

$$Q(\mathbf{x}) = 5x_1^2 + 3x_2^2 + 2x_3^2 - x_1x_2 + 8x_2x_3$$

$$Q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} = [\, x_1 \quad x_2 \quad x_3 \,] \begin{bmatrix} 5 & -1/2 & 0 \\ -1/2 & 3 & 4 \\ 0 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

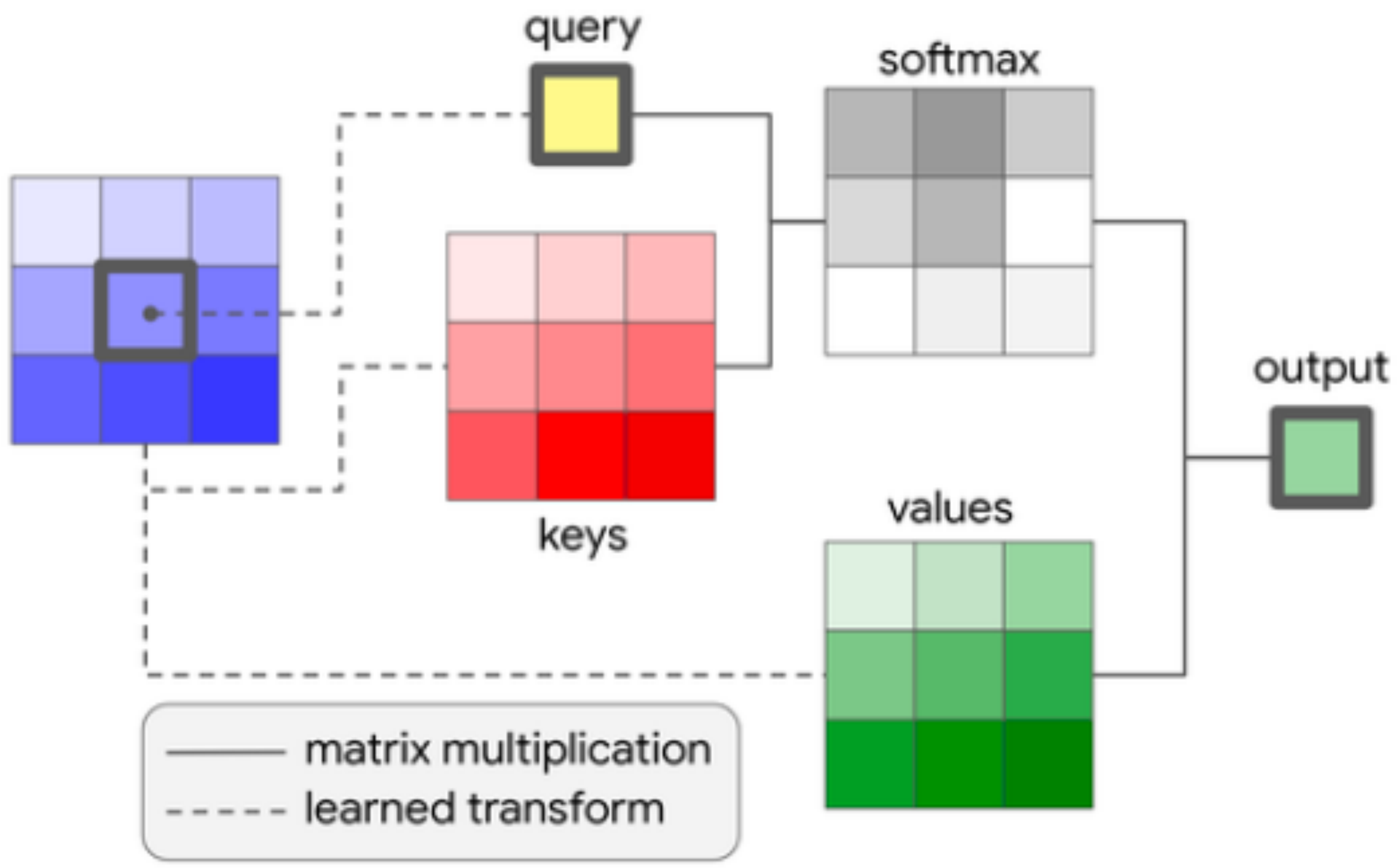


2. Multiplicative Interactions

3. Attention and Multiplicative Interactions

- The **attention systems in sequence modeling** similarly use multiplicative interactions to effectively scale different parts of the input.

- **Attention systems** can suppress or amplify certain inputs and allow long-range dependencies by combining inputs across time-steps. We use **these insights** to posit that while more expensive, Considering a higher order interaction might prove more beneficial to such sysstems.



3. Experimental Setup

- **Aim:** MI can boost performance across a wide range of problems and domains
  - We conjecture that this is because they effectively allow better integration of different kinds of information

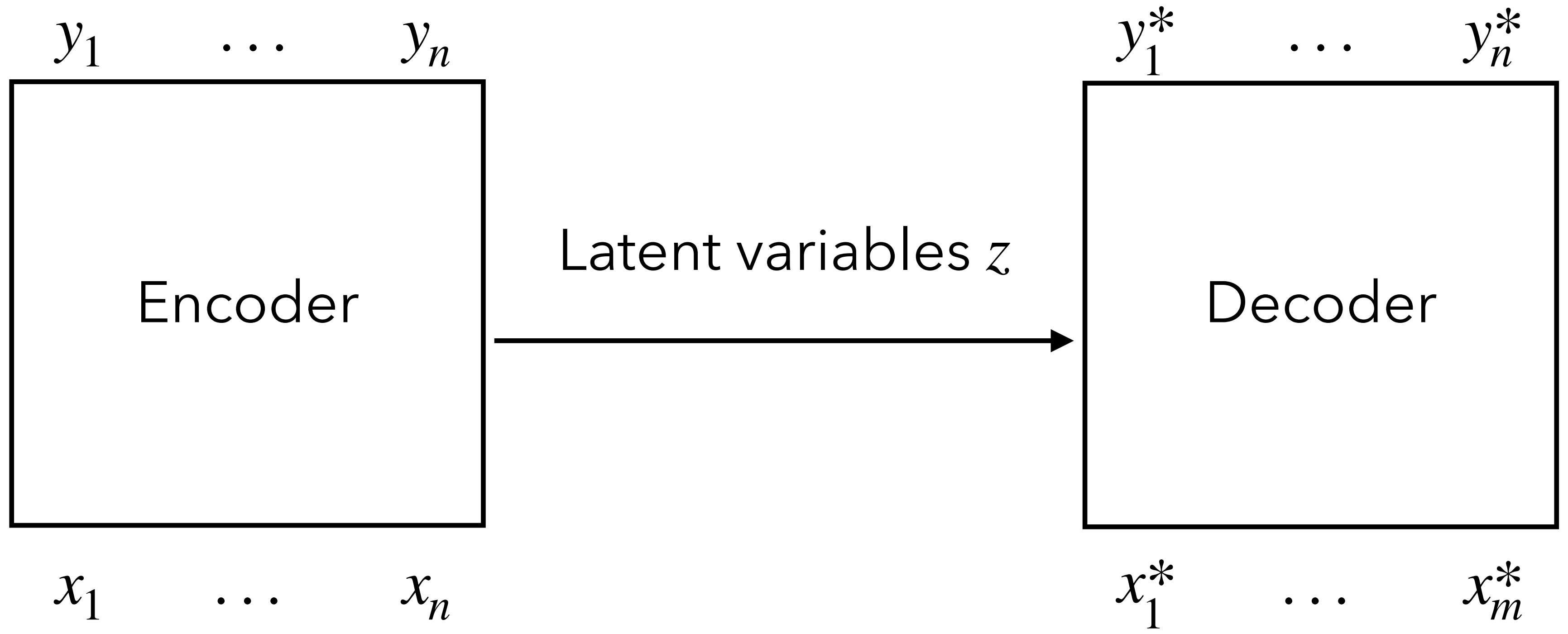
- **Experiments**
  - (a) Latent variables in decoder models
  - ~~(b) contextual information in multitask RL~~
  - (c) recurrent state in sequence models

- **Remark**
  - In all experimental cases, authors implement MI using ▲ a series of standard linear layers ▲ with a reshape operation in between to form the intermediate matrix
  - The quantity $f_1(z) = z^T W + B$ represent the 2D output of projecting the **contextual information.**
    - 2D-contextual projection = generated weights using hypernet

3. Experimental Setup

- **[1] Latent variable models with multiplicative decoders**
  - We investigate how **contextual latent variables** can be **better integrated** into neural **decoders**
  - We consider neural processes for **few-shot regression**
    - **[Here, we consider only the concept of Language Model and seq2seq, not neural processes]**
    - It work by predicting a function value y_* at new observations x_* having observed previous values (x_i, y_i)

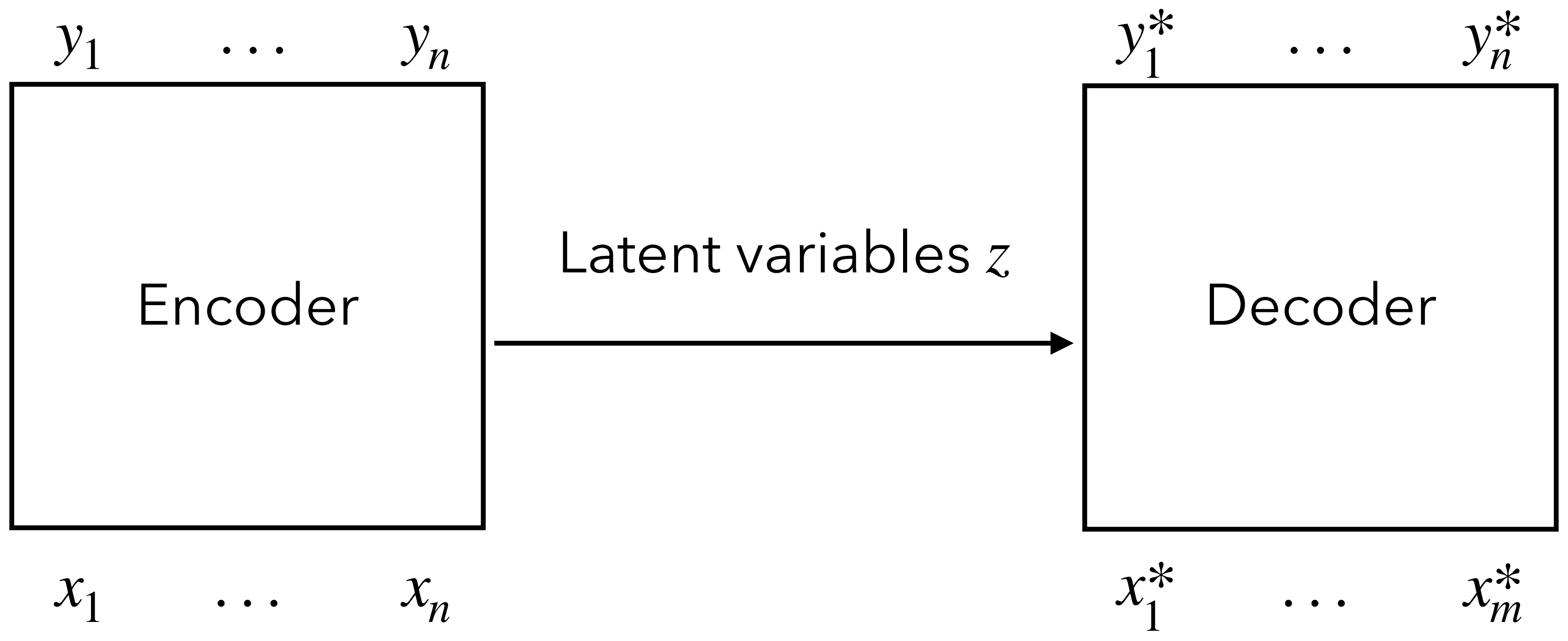
Text datset $D = (x_i, y_i)$



New data point x_i^* is mapped to y_i^* throuh a decoder network

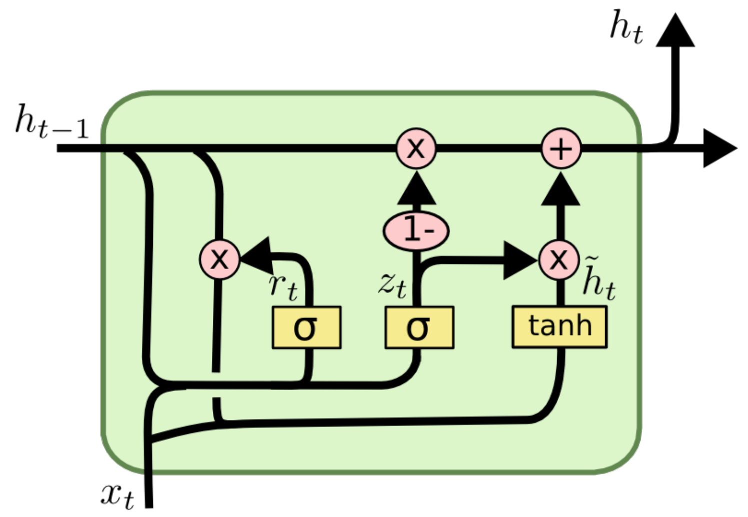
3. Experimental Setup

- **Aim**: To increase the experessivity of the decoder by improving the conditioning on z

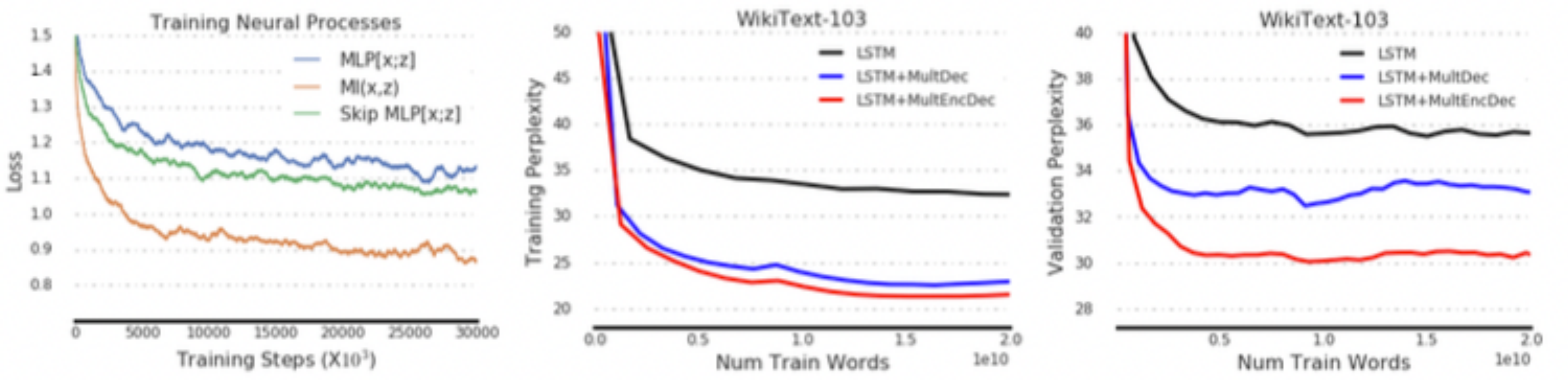


$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

- **Standard Approach [Concatenation]**
  - MLP([x;z]) in a variant LSTM (for simplicity in our seminar)
- **Additional Standard Approach [Multiplicative Interactions]**
  - Skip-MLP([x;z]) in a variant LSTM (for simplicity in our seminar)
- **Proposed Approach [Multiplicative Interactions]**
  - MI([x;z]) in LSTM (for simplicity in our seminar)



$$
\begin{aligned}
z_t &= \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \\
r_t &= \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \\
\tilde{h}_t &= \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$



3. Experimental Setup

- **[2] Word-level language modelling with recurrent models**
    - At each time-step, the network outputs a prediction about the next-word in the sequence.

    - **A standard architecture**
        - [1] To project one-hot word vectors x_t to **input embeddings** $z_t^i = Wx_t$
        - [2] Extend this to the **output embedding** of LSTM:$z_{t+1}^o = W_2 h_t x_t + b$
        - [3] Finally, the output $y_{t+1} = softmax(z_{t+1}^o W^T + b_2)$ where W is the embedding weights

    - **A Proposed architecture**
        - [1] Output Embedding:
            - $c = relu(W_3 x_t + b)$
            - $z_{t+1}^o = MI(c^T, h_t)$
        - [2] Finally, the output may be same.

Table 1: Word-level perplexity on WikiText-103

Model	Valid	Test	No. Params
LSTM Rae et al. (2018)	34.1	34.3	88M
Gated CNN Dauphin et al. (2017)	-	37.2	-
RMC Santoro et al. (2018)	30.8	31.6	-
Trellis Networks Bai et al. (2019)	-	30.35	180M
TransformerXL Dai et al. (2018)	**17.7**	**18.3**	**257M**
LSTM (ours)	34.7	36.7	88M
LSTM + MultDec	31.7	33.7	105M
LSTM + MultEncDec	**28.9**	**30.3**	**110M**

4. Conclusion and Future work

- We **hope** that this work leads to a **broader understanding and consideration** of such methods by practitioners, and in some cases replacing the standard practice of concatenation when using conditioning, contextual inputs, or additional sources of information.

- While attention models use some of these multiplicative interactions, we **hope that applying some of the lessons from this work** (such as higher order interactions) will allow even greater integration of information in attention systems.

References

[1] https://d2l.ai/d2l-en.pdf p.306

[2] https://en.wikipedia.org/wiki/Inductive_bias

[3] Ha, David, Andrew Dai, and Quoc V. Le. "Hypernetworks." *arXiv preprint arXiv:1609.09106* (2016).