Multiplicative Interactions and Where to Find Them[1]

March 26, 2020

[1] "Multiplicative Interactions and Where to Find Them" by Jayakumar et al.

Overview

Overview

- Authors explore scenarios where we fuse two variables x, z “multiplicatively”, instead of addition (concatenation).

Overview

- Authors explore scenarios where we fuse two variables x, z "multiplicatively", instead of addition (concatenation).
- They unify the notions of gating, attention, metric learning, etc by a single notion of *multiplicative interaction* (MI).

Overview

- Authors explore scenarios where we fuse two variables x, z "multiplicatively", instead of addition (concatenation).
- They unify the notions of gating, attention, metric learning, etc by a single notion of *multiplicative interaction* (MI).
- They show, that such multiplicative interactions are more *expressive*

Overview

- Authors explore scenarios where we fuse two variables x, z "multiplicatively", instead of addition (concatenation).
- They unify the notions of gating, attention, metric learning, etc by a single notion of *multiplicative interaction* (MI).
- They show, that such multiplicative interactions are more *expressive*
  - i.e. can represent broader set of functions with smaller amount of parameters.

Overview

- Authors explore scenarios where we fuse two variables x, z "multiplicatively", instead of addition (concatenation).
- They unify the notions of gating, attention, metric learning, etc by a single notion of *multiplicative interaction* (MI).
- They show, that such multiplicative interactions are more *expressive*
  - i.e. can represent broader set of functions with smaller amount of parameters.
- They achieve SotA performance on several tasks simply by replacing concatenation with MI.

General definition of Multiplicative Interaction

Authors define multiplicative interaction as a function between two variables x and z:

$$\mathcal{M}(x, z) = \mathbf{z}^T \mathbb{W} \mathbf{x} + \mathbf{z}^T \mathbf{U} + \mathbf{V}\mathbf{x} + \mathbf{b},$$

where:

- x, z are vectors
- $\mathbb{W}$ is a 3d-tensor
- U, V are 2d-matrices and b is a bias vector.

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

- $\mathbf{z}^T \mathbb{W} \mathbf{x}$ is almost the same as normal product $\mathbf{z}^T A \mathbf{x}$

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

- $\mathbf{z}^T \mathbb{W} \mathbf{x}$ is almost the same as normal product $\mathbf{z}^T A \mathbf{x}$
- We should just abstract away from what lies in a cell of the matrix

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

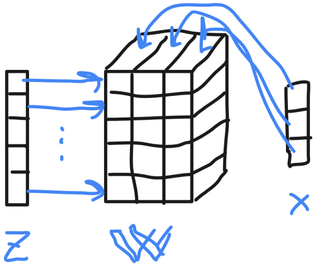
- $\mathbf{z}^T \mathbb{W} \mathbf{x}$ is almost the same as normal product $\mathbf{z}^T A \mathbf{x}$
- We should just abstract away from what lies in a cell of the matrix
- Imagine that each cell of matrix W is an apple

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

- $\mathbf{z}^T \mathbb{W} \mathbf{x}$ is almost the same as normal product $\mathbf{z}^T A \mathbf{x}$
- We should just abstract away from what lies in a cell of the matrix
- Imagine that each cell of matrix W is an apple
- Then $z^\top A x$ produces an apple

Side note: how to compute $\mathbf{z}^T \mathbb{W} \mathbf{x}$

- $\mathbf{z}^T \mathbb{W} \mathbf{x}$ is almost the same as normal product $\mathbf{z}^T A \mathbf{x}$
- We should just abstract away from what lies in a cell of the matrix
- Imagine that each cell of matrix W is an apple
- Then $z^\top A x$ produces an apple
- Then in case of $\mathbb{W}$ each apple is a 1d-vector



MIs are more expressive

MIs are more expressive

- It is known, that MLP are *universal approximators*, they can approximate any function with arbitrary small error ε if they are wide enough.

MIs are more expressive

- It is known, that MLP are *universal approximators*, they can approximate any function with arbitrary small error ε if they are wide enough.
- But they are not able to *represent* any function, i.e. to model it with $\varepsilon = 0$.

MIs are more expressive

- It is known, that MLP are *universal approximators*, they can approximate any function with arbitrary small error ε if they are wide enough.
- But they are not able to *represent* any function, i.e. to model it with $\varepsilon = 0$.
- **Definition** Hypothesis space $\mathcal{H}_{f(\theta)}$ is a set of those functions that $f(\theta)$ can represent

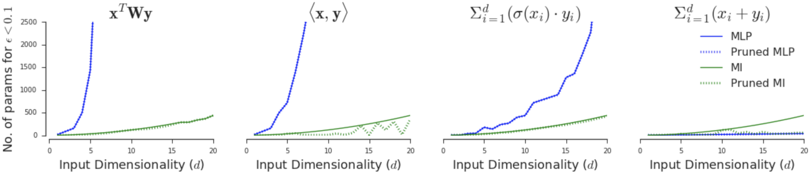
MIs are more expressive

- It is known, that MLP are *universal approximators*, they can approximate any function with arbitrary small error ε if they are wide enough.
- But they are not able to *represent* any function, i.e. to model it with $\varepsilon = 0$.
- **Definition** Hypothesis space $\mathcal{H}_{f(\theta)}$ is a set of those functions that $f(\theta)$ can represent
- **Theorem**: given an MLP and MI, hypotheses class for MI is *strictly* larger than for MLP.

MIs are more expressive

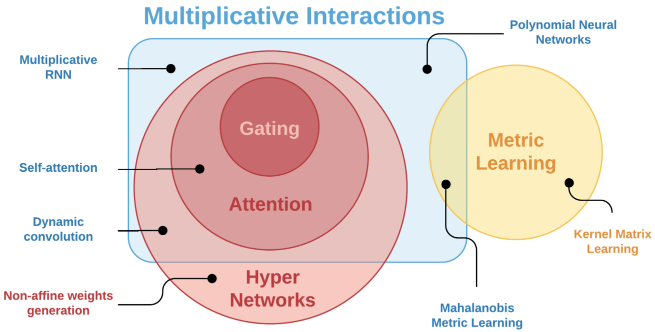
- It is known, that MLP are *universal approximators*, they can approximate any function with arbitrary small error ε if they are wide enough.
- But they are not able to *represent* any function, i.e. to model it with $\varepsilon = 0$.
- **Definition** Hypothesis space $\mathcal{H}_{f(\theta)}$ is a set of those functions that $f(\theta)$ can represent
- **Theorem**: given an MLP and MI, hypotheses class for MI is *strictly* larger than for MLP.
- This means, that MI can represent any function from $\mathcal{H}_{\text{MLP}}$, but there are some functions that MI can represent, but MLP cannot

MIs are more expressive



Examples of MIs

There are a lot of examples of MIs in the modern literature:



Examples of MI

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:
  - x is an input vector, we want to attend over its elements

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:
  - x is an input vector, we want to attend over its elements
  - we generate $m = \mathcal{M}(x, z)$ and compute $m \odot x$

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:
  - x is an input vector, we want to attend over its elements
  - we generate $m = \mathcal{M}(x, z)$ and compute $m \odot x$
- Metric learning can be seen as a MI:

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:
  - x is an input vector, we want to attend over its elements
  - we generate $m = \mathcal{M}(x, z)$ and compute $m \odot x$
- Metric learning can be seen as a MI:
  - To have $\mathcal{M}(x, z) = z^\top x$ we just need to set $\mathbb{W}$ to "identity" (zeros everywhere and ones on the main diagonal)

Examples of MI

- Attention can be seen as $\mathcal{M}(x, z)$:
  - x is an input vector, we want to attend over its elements
  - we generate $m = \mathcal{M}(x, z)$ and compute $m \odot x$
- Metric learning can be seen as a MI:
  - To have $\mathcal{M}(x, z) = z^\top x$ we just need to set $\mathbb{W}$ to "identity" (zeros everywhere and ones on the main diagonal)
  - More general forms of metric learning as $d_C(x, z) = (\mathbf{x} - \mathbf{z})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{z})$ can be also expressed as $\mathcal{M}(x, z)$.
- Hypernetworks, gating, etc

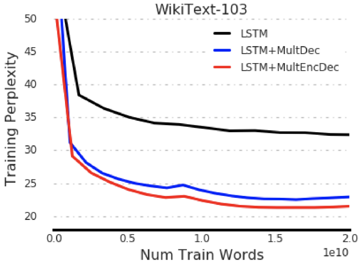
Experiment: integrate MI in LSTM

Experiment: integrate MI in LSTM

In vanilla LSTM all interactions are "concatentation"-based:

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

Authors replaced some of them with multiplicative ones and observed the boost in performance:



Conclusion

- Authors also did some experiments for few-shot learning and multi-task learning
- MI is a powerful tool to integrate different sources of information
- We have countless scenarios to explore them