6. Attention-based networks

6.1. Attention mechanisms

Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Introduction

- Attention is a core human brain functioning mechanism and a complex cognitive function.
- Attention is the ability to concentrate on information parts and not as a whole.
- For example, humans tend to focus on specific parts or aspects of a scene and to identify similarities with other scenes.
- Humans process information by selecting high-value features from huge information sources with limited resources.
- Attention networks are inspired by this behavior.
- From 2020 or so, the dominant models for all the natural language processing tasks. It is based on a form of attention mechanism.

Attention pooling

- An attention mechanism is a system to access a dataset $\mathcal{D}$ consisting of a set of tuples key-value $\mathbf{k}_i, \mathbf{v}_i$ through a query $\mathbf{q}$.

$$\text{Attention}\left(\mathbf{q}, \mathbb{D}\right) = \sum_{i=1}^{N} \alpha\left(\mathbf{q}, \mathbf{k}_i\right) \mathbf{v}_i \tag{1}$$

  where $\alpha\left(\mathbf{q}, \mathbf{k}_i\right) \in \mathbb{R}$ are called *attention weights* and they measure the relevance of a key to the query.

- The weights are normalized so they produce a convex combination, i.e,

$$\sum_{i=1}^{N} \alpha\left(\mathbf{q}, \mathbf{k}_i\right) = 1 \tag{2}$$

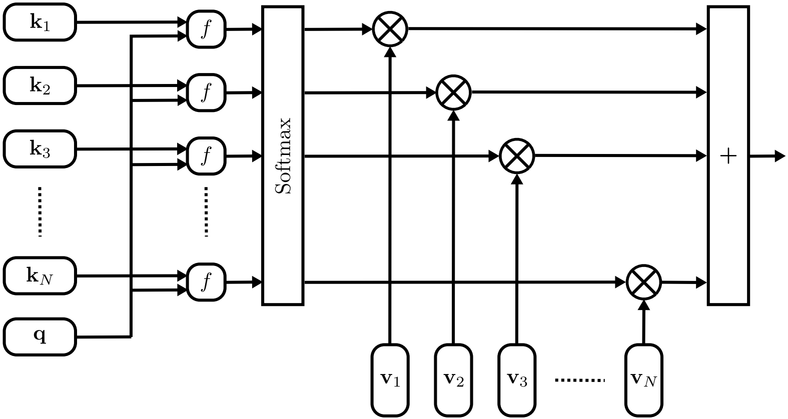
Attention pooling

- A particular form of the weights useful in deep learning is

$$\alpha\left(\mathbf{q}, \mathbf{k}_i\right) = \frac{\exp\left(f\left(\mathbf{q}, \mathbf{k}_i\right)\right)}{\sum_{j=1}^{N} \exp\left(f\left(\mathbf{q}, \mathbf{k}_j\right)\right)} \tag{3}$$

  where $f(\cdot)$ is any function useful to measure some similarity between the query and the key.
- This function is itself a softmax activation and it has properties of probability mass function.

Attention pooling



Attention pooling as a convex combination of values.

- A regression machine can be constructed as an attention-pooling mechanism

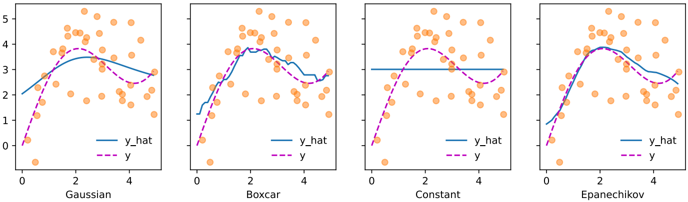
$$g(x) = \sum_{i=1}^{N} y_i \frac{f(x, x_i)}{\sum_{j=1}^{N} f(x, x_j)} \tag{4}$$

- Regression $g(x)$ as x as the query, and the keys are training samples x_i. The values are regressors y_i.

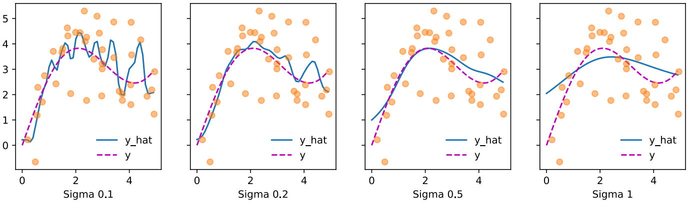
- Function $y = 2sin(x) + x + \varepsilon$ is to be approximated with the N-W regressor, where ε_i is a Gaussian noise of zero mean and unit variance.
- The training data consists of 100 samples distributed uniformly in the interval $0 \sim 4$.
- The performance is compared wrt the following kernels:

$$
\begin{aligned}
f(\mathbf{q}, \mathbf{k}) &= \exp\left(-\frac{1}{2}\|\mathbf{q} - \mathbf{k}\|^2\right) && \text{Gaussian} \\
f(\mathbf{q}, \mathbf{k}) &= 1 \text{ if } \|\mathbf{q} - \mathbf{k}\| \leq 1 && \text{Boxcar} \\
f(\mathbf{q}, \mathbf{k}) &= \max\left(0, 1 - \|\mathbf{q} - \mathbf{k}\|\right) && \text{Epanechikov}
\end{aligned}
\tag{5}
$$

Comparisons of different kernels.



Results of the Gaussian kernel with different width parameters.



Zhang, Lipton, Li, Smola. Dive Into Deep Learning, 2023

Attention functions

Dot product

- Euclidian distance

$$f(\mathbf{q}, \mathbf{k}) = -\|\mathbf{q} - \mathbf{k}\|^2 = \|\mathbf{q}\|^2 - \|\mathbf{k}\| + 2\mathbf{q}^\top \mathbf{k} \tag{6}$$

If we assume that the norm of the keys are approximately constant, and taking into account that $\mathbf{q}$ is the same for all keys

$$f(\mathbf{q}, \mathbf{k}) = 2\mathbf{q}^\top \mathbf{k} + \text{constant} \tag{7}$$

- We can normalize the dot product with respect to the dimension D of the vectors and apply a softmax:

$$\alpha(\mathbf{q}, \mathbf{k}_i) = \frac{\exp\left(D^{-\frac{1}{2}} \mathbf{q}^\top \mathbf{k}_i\right)}{\sum_{j=1}^{N} \exp\left(D^{-\frac{1}{2}} \mathbf{q}^\top \mathbf{k}_j\right)} \tag{8}$$

Attention functions
Dot product

- The attention mechanism input to the sum block of slide 5 can be written as
$$\mathbf{z} = \mathbf{V}^\top \mathrm{softmax}\left(D^{\frac{1}{2}} \mathbf{q}^\top \mathbf{K}\right) \tag{9}$$
where $\mathbf{K}$ contains all the keys and $\mathbf{V}$ all the values.
- In practice, the query and the key do not have the same dimension, therefore, transformation matrices are used:
$$\mathbf{z} = \mathbf{V}^\top \mathrm{softmax}\left(D^{\frac{1}{2}} \mathbf{q}^\top \mathbf{M}\mathbf{K}\right) \in \mathbb{R}^N \tag{10}$$
where $\mathbf{M} \in \mathbb{R}^{D_q \times D_k}$ transforms from the space of queries to the one of keys.

Attention functions
Dot product

- If $\mathbf{Q}$ contains a set of M queries, we can construct a set of responses as

$$\mathbf{Z} = \mathbf{V}^\top \text{softmax}\left(D^{\frac{1}{2}} \mathbf{Q}^\top \mathbf{M}\mathbf{K}\right) \in \mathbb{R}^{N \times M} \tag{11}$$

- This is necessary for training purposes by using mini batches.

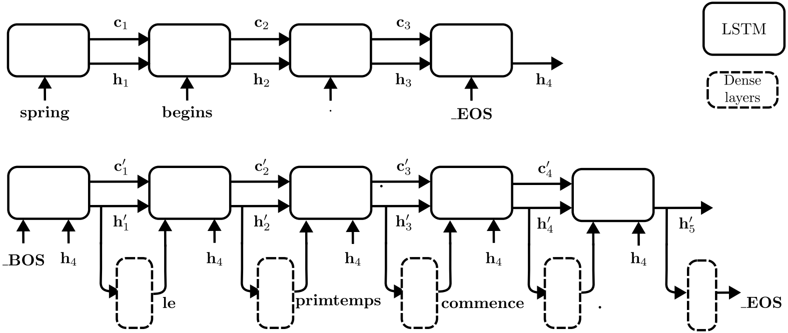
Additive attention

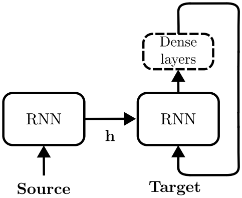
- The additive attention function also assumes that the query and the key have different lengths. The scoring, in this case, is

$$f\left(\mathbf{q}, \mathbf{k}\right) = \mathbf{w}_f \tanh\left(\mathbf{W}_q^\top \mathbf{q} + \mathbf{W}_k^\top \mathbf{k}\right) \tag{12}$$

- Matrices $\mathbf{W}_q$ and $\mathbf{W}_k$, and vector $\mathbf{w}_f$ are trainable parameters. Therefore, this is equivalent to a dense or fully connected network with one hidden layer with tanh activation and one linear scalar output.

- Recall the RNN sequence to sequence machine translation.

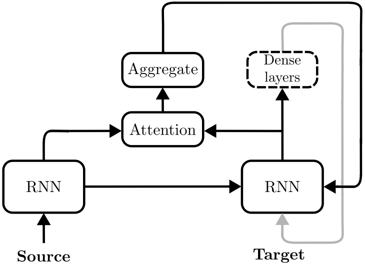




- The main limitation of this method is the length of states **h**. There may be not enough space to code long sequences.

- This limitation can be overcome with the use of attention mechanisms: when a token is predicted, a model attends only to parts of the input sequence that are relevant to the prediction.
- These parts are then used to modify the state before producing the next prediction.
- This gave rise to the idea of transformers.

- The encoder RNN passes states $\mathbf{h}_t$ to the decoder. The decoder updates its states at every step with an attention pooling.



- If $\mathbf{h}'_{t'}$ are the states of the decoder

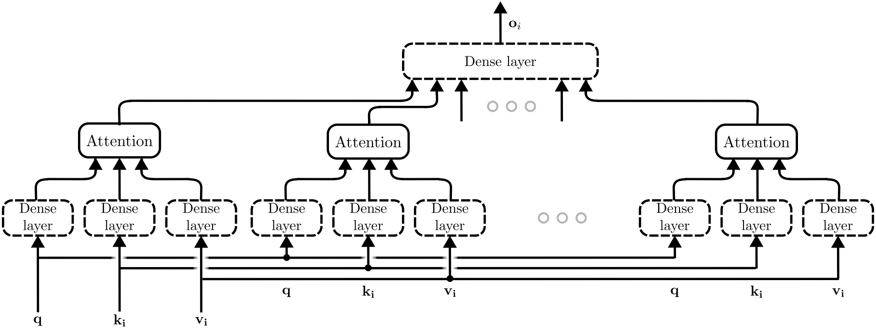
$$\mathbf{c}_{t'} = \sum_{t=1}^{T} \alpha\left(\mathbf{h}'_{t'-1}, \mathbf{h}_t\right)\mathbf{h}_t \qquad (13)$$

- Then, $\mathbf{c}'_t$ is passed to the dense block to generate the next state $\mathbf{h}'_{t'}$ and a new token.

- The additive attention scoring in Eq. (12) is used.

Multi head attention

- The multi-head attention combines different behaviors of the same attention mechanism into a feature space.
- The responses are linearly combined at the output.



$$\mathbf{h}_{i,j} = f\left(\mathbf{W}_j^{(q)\top}\mathbf{q}_i, \mathbf{W}_j^{(k)\top}\mathbf{k}_i, \mathbf{W}_j^{(v)\top}\mathbf{v}_i\right), \quad \mathbf{o}_i = \sum_j \mathbf{w}_j^{(o)\top}\mathbf{h}_{i,j} \quad (14)$$