One-shot Meta-learning with Matching Networks

Adrian Gonzalez-Martin[1], Matthew Lee[1], Adrian Daniel Szwarc[1], Talip Uçar[1]
Computer Science Department, University College London, Gower Street, London, WC1E 6BT
talip.ucar.16@ucl.ac.uk

1. Introduction

One-shot learning is learning from one example. It is an important property of natural intelligence. Humans can often learn new concepts using one, or few examples. With advances in areas of machine learning concerning memory, attention and various optimization techniques, there has been a renewed interest in one-shot learning approaches in recent years. But, it has been mostly investigated in the context of vision tasks, rather than language ones. Thus, our work focuses on one-shot learning in a language task, particularly predicting missing words in sentences. We aim to establish a benchmark by using the WikiText-2 [1] dataset, and comparing different one-shot approaches. We use Matching Networks [2] to find our baseline, and then improve on it, using new embeddings and distance metrics such as cosine, Euclidean, and Poincaré.

2. Meta-learning Framework

The success of recent one-shot learning approaches has not been purely down to advances in neural network architectures but also clever meta-learning frameworks.

Meta-learning

Train 1

Label,	Sentence
seafood,	i am on a <blank_token> diet, i see food, and i eat it
relaxed,	i'm not lazy, i'm just very <blank_token>
music,	no need to be a hipster to play <blank_token> in vynils
drink,	don't <blank_token> while driving, you might spill the beer
free,	you're <blank_token> , then you're taxed to death

Test 1

Label,	Sentence
music,	nowadays <blank_token> doesn't sound as before

Train 2

Label,	Sentence
luck,	i get enough exercise pushing my <blank_token>
life,	if you can't see the bright side of <blank_token>, polish the dull side
quantity,	<blank_token> is what you count, quality is what you count on
road,	the <blank_token> to success is always under construction
longer,	if i'm not back in five minutes, just wait <blank_token>

Test 2

Label,	Sentence
life,	when <blank_token> gives you melons, you might be dyslexic

...

Meta-testing

Train 1

Label,	Sentence
today,	most of the choices you make <blank_token> will not benefit you immediately
brain,	the human <blank_token> did not evolve for life in a delayed-return environment
years,	a lot can happen in 100 <blank_token>
lion,	a <blank_token> appears across the plain
chronic,	wild animals rarely experience <blank_token> stress

Test 1

Label,	Sentence
brain,	most of the space in the <blank_token> is taken up by axons

...

3. Preprocessing

"The fox is sick of jumping over dogs."

-> Lowercase, replace numbers with"N", remove punctuation
"the fox is sick of jumping over dogs"

-> Choose missing words and split into labels and sentences
"sick, the fox is <blank_token> of jumping over dogs"

-> Tokenization
["sick", "the", "fox", "is ", "<blank_token>", "jumping", "over ", "dogs"]

-> (vocab*)
{"the" -> 0,
"fox"-> 1,
"is" -> 2,
...}

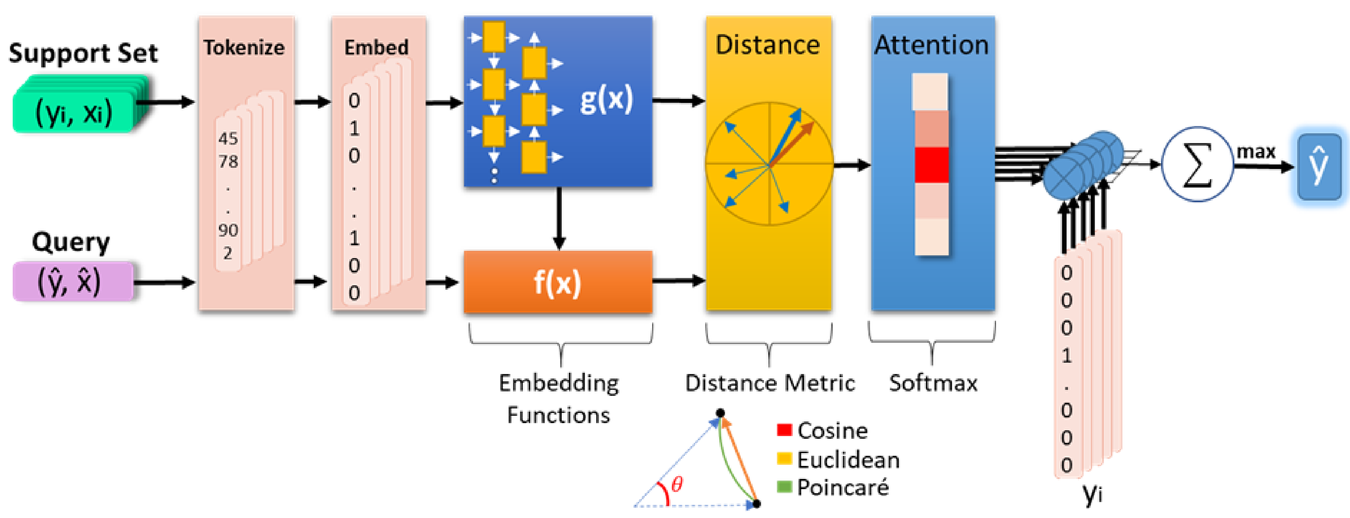
-> (numericalise)
[0, 1, 2, ...]

-> (basic embedding)
[
[1, 0, 0, ...],
[0, 1, 0, ...],
[0, 0, 1, ...]
]

* can have seperate vocabs for labels and sentences

4. Matching Network Architecture

Embeddings + Similarity Measure + Attention



5. Embedding, Attention and Distance Metrics

Embedding: Full Context Embeddings

- $g(x)$ employs a bidirectional LSTM, with the support set S treated as a sequence. The ith stage of the LSTM has as input $g'(x_i)$ - the initial embedding of the previous step. The LSTM is also a function of the last hidden state $\overrightarrow{h}_{i-1}$ and last output vector $\overrightarrow{c}_{i-1}$ as usual.

$$\overrightarrow{h}_i, \overrightarrow{c}_i = \texttt{LSTM}(g'(x_i), \overrightarrow{h}_{i-1}, \overrightarrow{c}_{i-1})$$
$$\overleftarrow{h}_i, \overleftarrow{c}_i = \texttt{LSTM}(g'(x_i), \overleftarrow{h}_{i-1}, \overleftarrow{c}_{i-1})$$

- $f(\hat{x})$ is a one way LSTM with an attention mechanism that unfolds over K steps:

$$f(\hat{x}) = \texttt{attLSTM}(f'(\hat{x}), g(S), K)$$

where like g', f' is our initial embedding but for the query; $g(S)$ represents $g(x_i)$ already applied to all x_i and K is the number of steps as noted.

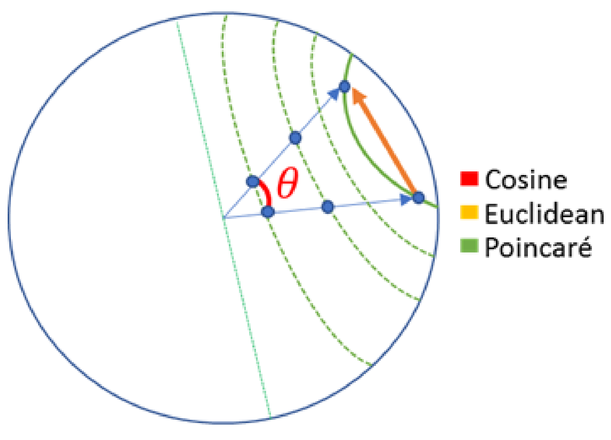
Attention:

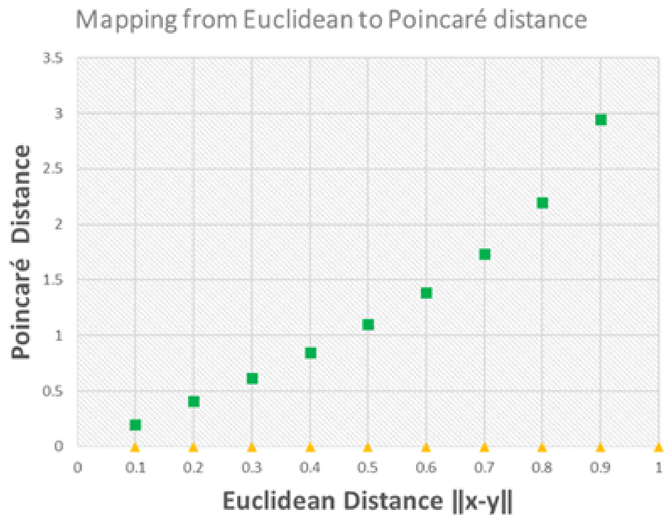
- The attention kernel applies the cosine distance, c(.), between the embedded vectors of two data points before the softmax function normalizes the resulting weights:

$$a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}), g(x_i))}}{\sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}}$$

Distance Metrics:

- Investigating three distance metrics: **Cosine, Euclidean**, and **Poincaré**.





6. References

[1] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016.

[2] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.