Matching Networks for One Shot Learning

- Learn from a small sample space
    - Metric Learning
    - External Learning

> Some non-parametric model is able to learn new samples, like KNN, which would keep the sample instead of discarding it.

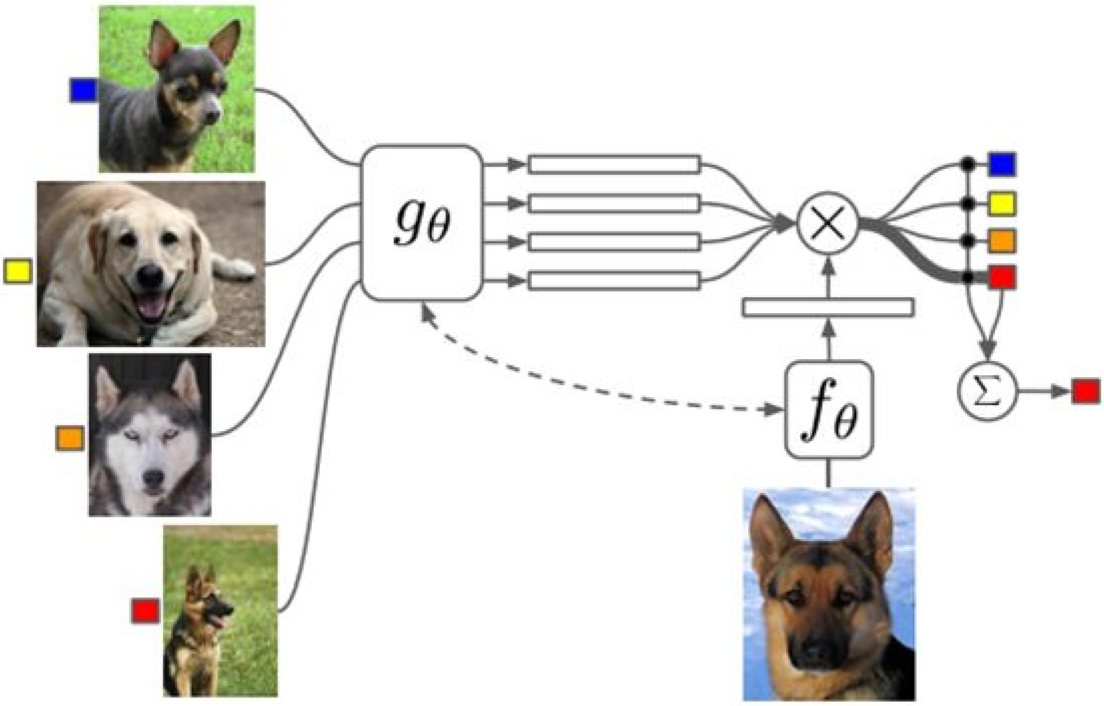

Figure 1: Matching Networks architecture

- Add external memories onto the network
- In seq2seq, P(B |A), A, B are sets.
- 4 left imgs make the support set. Img in right bottom is a test example. They combined as a task.

- `Prediction = f(support_set, tets_example)`, $P(\hat{y}|\hat{x}, S)$, $S = (x_i, y_i)_{i=1}^{k}$, k is the number of support examples.
    - In this case, the model is $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$, where $\hat{y}$ is the linear combination of smaples from support set's labels. The weights are the relationship between the test example and support set.

Attention Kernel

- $a(\hat{x}, x_i)$ could be treat as attention kernel, and thus the prediction is the image's label from the support set whose gets the most attentions.
- The most common attention kernel is the `softmax` distance in `cosine` distance. (f, g are two embedding functions which could be realized by NN)

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

Full Context Embeddings

- embedding vector $emb\_x_i = g(x_i) \leftarrow g(x_i, S)$. Support set is randomly chose each time and the embedding function should also consider about decreasing the difference between randomly chosen in support set and x_i.
- Take the relationship between `word` and `context` in machine translation. S could be viewed as context of x_i, so the LSTM is used in this paper.
- The embedding function of text function f:

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

$f'(\hat{x})$ is the input of CNN embedding layer. $g(S)$ is the output of embedding function of support set, K is the time steps of LSTM, which is equal to the number of imgs in support set.