Learning Algorithms for Active Learning

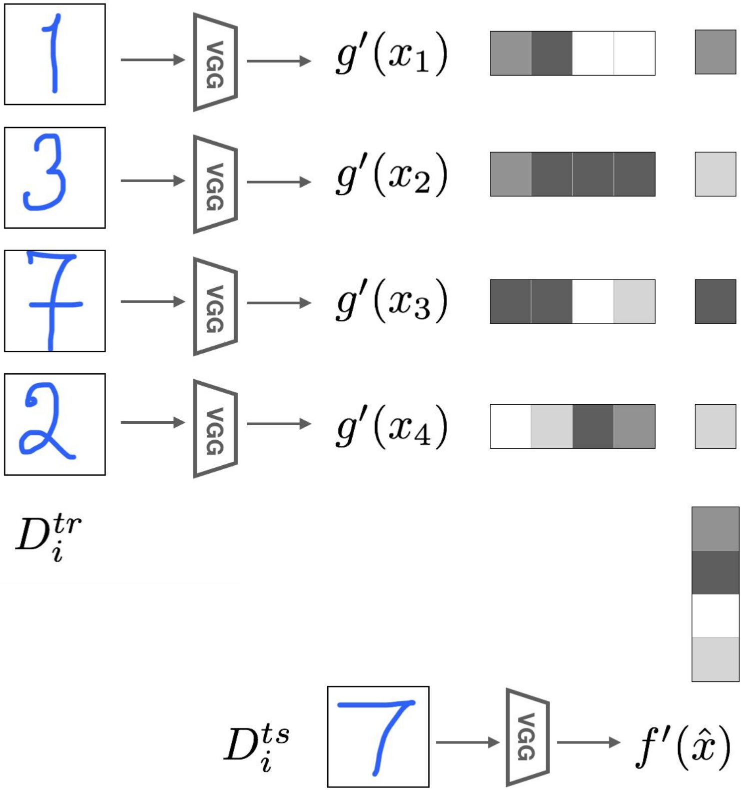
Learning Algorithms for Active Learning

Philip Bachman [*1] **Alessandro Sordoni** [*1] **Adam Trischler** [1]

Plan

- Background
  - Matching Networks
  - Active Learning
- Model
- Applications: Omniglot and MovieLens
- Critique and discussion

Background: Matching Networks (Vinyals et al. 2016)



embedding of probe item

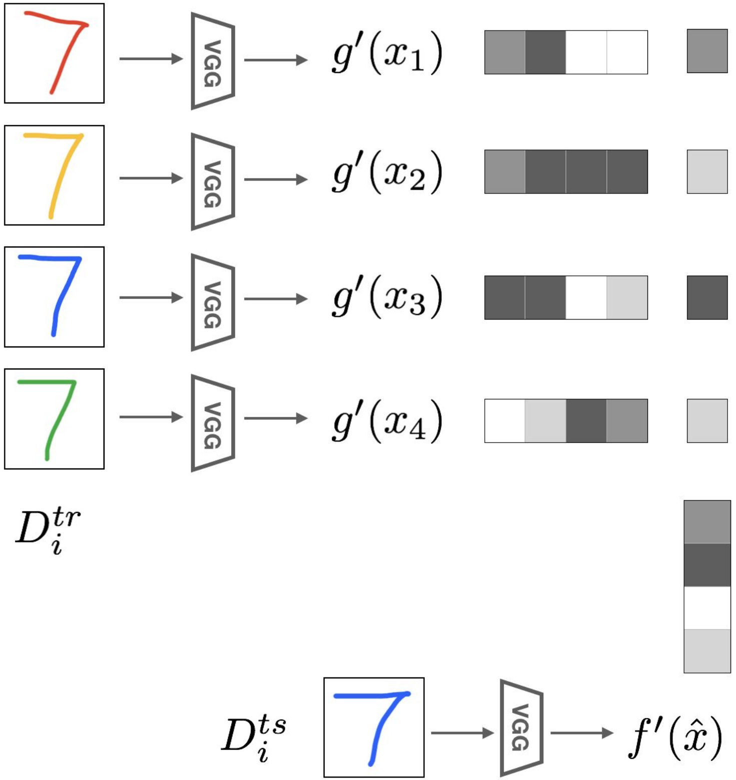
embedding of example

$$\hat{y} = \sum_{i=1}^{k} a(f'(\hat{x}), g'(x_i))y_i$$

cosine distance (e.g.)

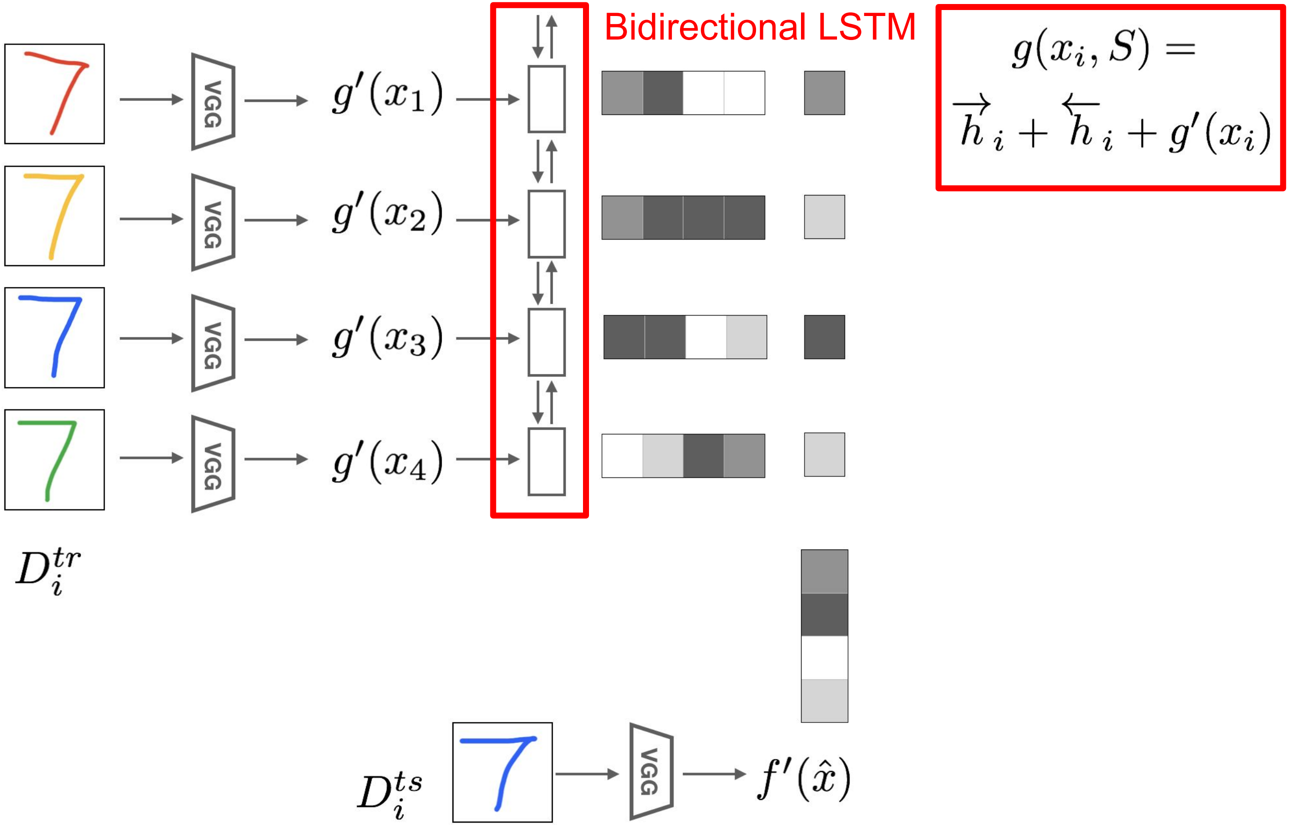
label of example

Background: Matching Networks



Motivates context-sensitive encoder

Background: Matching Networks



Background: Matching Networks

Desidirata for $\hat{x}$ encoding:

- Depend on embeddings of examples, g(S)
- Be able to selectively ignore some examples (e.g. outliers)
- Build invariance to the order of the examples

$$\longrightarrow \quad \mathrm{attLSTM}(f'(\hat{x}), g(S), K)$$

$$\hat{h}_k, c_k = \mathrm{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \tag{3}$$

$$h_k = \hat{h}_k + f'(\hat{x}) \tag{4}$$

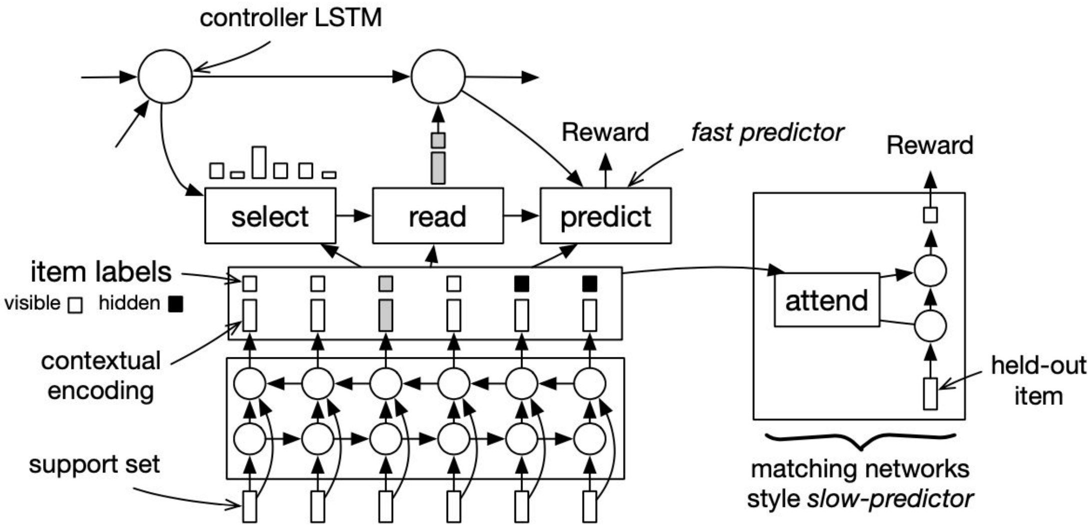
$$r_{k-1} = \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i))g(x_i) \tag{5}$$

$$a(h_{k-1}, g(x_i)) = \mathrm{softmax}(h_{k-1}^T g(x_i)) \tag{6}$$

Background: Active Learning

- Most real-world settings: many unlabeled examples, few labeled ones
- *Active Learning*: Model requests labels; tries to maximize both task performance and data efficiency
  - E.g. task involving medical imaging: radiologist can label scans by hand, but it's costly
- Instead of using heuristics to select items for which to request labels, Bachman et al. use meta learning to learn an active learning strategy for a given task

Proposed Model: "Active MN"



Algorithm 1 End-to-end active learning loop (for Eq. 3)

```
# encode items in S with context-sensitive encoder
# and encode items in E with context-free encoder
```
$S = \{(x, y)\}$, $S_0^u = \{(x, \cdot)\}$, $S_0^k = \emptyset$, $E = \{(\hat{x}, \hat{y})\}$

for $t = 1 \ldots T$ **do**

 # select next instance

$i \leftarrow \texttt{SELECT}(S_{t-1}^u, S_{t-1}^k, h_{t-1})$

 # read labeled instance and update controller

$(x_i, y_i) \leftarrow \texttt{READ}(S, i)$

$h_t \leftarrow \texttt{UPDATE}(h_{t-1}, x_i, y_i)$

 # update known / unknown set

$S_t^k \leftarrow S_{t-1}^k \cup \{(x_i, y_i)\}$

$S_t^u \leftarrow S_{t-1}^u \setminus \{(x_i, \cdot)\}$

 # perform fast prediction (save loss for training)

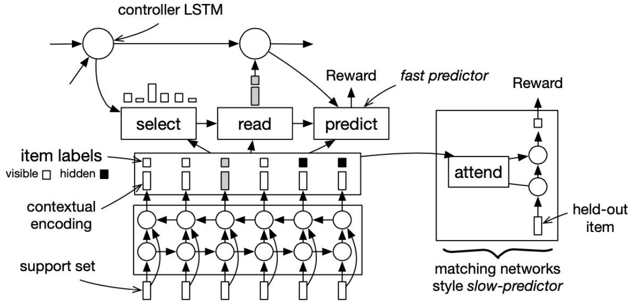
$L_t^S \leftarrow \texttt{FAST-PRED}(S, S_t^u, S_t^k, h_t)$

end for

 # perform slow prediction (save loss for training)

$L_T^E \leftarrow \texttt{SLOW-PRED}(E, S_T^u, S_T^k, h_T)$

Individual Modules



Context Free and Sensitive Encodings

- Gain context by using a bi-directional LSTM over independent encodings

Selection

- At each step *t,* places a distribution P_t^u over all unlabeled items in S_t^u
- P_t^u computed using a gated, linear combination of features that measure controller-item and item-item similarity

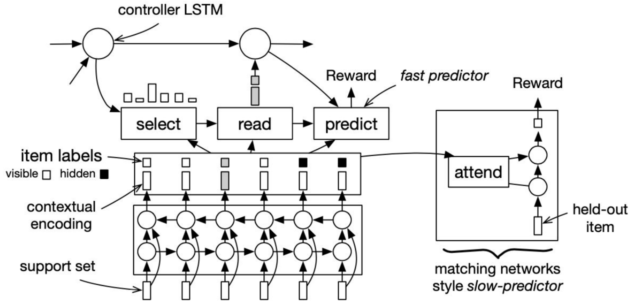
Reading

- Concatenates embedding and label for item selected, then applies linear transformation

Controller

- Input: r_t from reading module, and applies LSTM update: $h_t = \mathbf{LSTM}(h_{t-1}, r_t)$

Prediction Rewards



Prediction Reward: $R(E, S_t, h_t) \equiv \sum_{(\hat{x},\hat{y}) \in E} \log p(\hat{y}|\hat{x}, h_t, S_t)$

Objective: $\underset{\theta}{\text{maximize}} \underset{(S,E)\sim\mathcal{D}}{\mathbb{E}} \left[\underset{\pi(S,T)}{\mathbb{E}} \left[\sum_{t=1}^{T} R(E, S_t, h_t) \right] \right] \longrightarrow \underset{(S,E)\sim\mathcal{D}}{\mathbb{E}} \left[\underset{\pi(S,T)}{\mathbb{E}} \left[\sum_{t=1}^{T} \tilde{R}(S_t^u, S_t, h_t) + R(E, S_T, h_T) \right] \right]$

Fast Prediction

- Attention-based prediction for each unlabeled item using cosine sim. to labeled items
  - Sharpened by a non-negative matching score between x_i^u and the control state
- Similarities between context-sensitive embeddings don't change with *t* -> can be precomputed

Slow Prediction

- Modified Matching Network prediction
  - Takes into account distinction between labeled and unlabeled items
  - Conditions on active learning control state

Full Algorithm

Algorithm 1 End-to-end active learning loop (for Eq. 3)

 # encode items in S with context-sensitive encoder
 # and encode items in E with context-free encoder
$S = \{(x, y)\}, S_0^u = \{(x, \cdot)\}, S_0^k = \emptyset, E = \{(\hat{x}, \hat{y})\}$
for $t = 1 \ldots T$ **do**
  # select next instance
  $i \leftarrow \texttt{SELECT}(S_{t-1}^u, S_{t-1}^k, h_{t-1})$
  # read labeled instance and update controller
  $(x_i, y_i) \leftarrow \texttt{READ}(S, i)$
  $h_t \leftarrow \texttt{UPDATE}(h_{t-1}, x_i, y_i)$
  # update known / unknown set
  $S_t^k \leftarrow S_{t-1}^k \cup \{(x_i, y_i)\}$
  $S_t^u \leftarrow S_{t-1}^u \setminus \{(x_i, \cdot)\}$
  # perform fast prediction (save loss for training)
  $L_t^S \leftarrow \texttt{FAST-PRED}(S, S_t^u, S_t^k, h_t)$
end for
 # perform slow prediction (save loss for training)
$L_T^E \leftarrow \texttt{SLOW-PRED}(E, S_T^u, S_T^k, h_T)$

Tasks

Goal: maximize some combination of task performance and data efficiency

Test model on:

- Omniglot
  - 1623 characters from 50 different alphabets
- MovieLens (bootstrapping a recommender system)
  - 20M ratings on 27K movies by 138K users

Experimental Evaluation: Omniglot Baseline Models

1. **Matching Net (random)**
    a. **Choose samples randomly**
2. **Matching Net (balanced)**
    a. **Ensure class balance**
3. **Minimum-Maximum Cosine Similarity**
    a. **Choose items that are different**

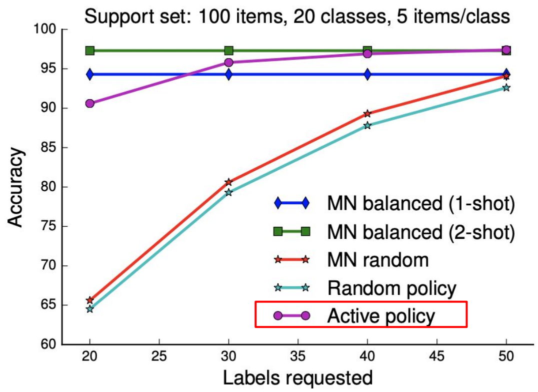
Experimental Evaluation: Omniglot Performance

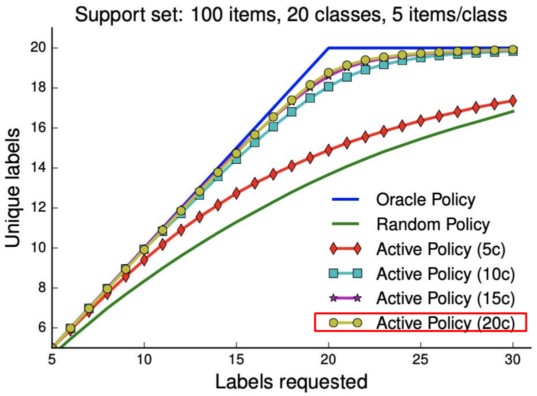
Table 1. Results for our active learner and baselines for the N-way, K-shot classification settings.

Model	5-way			10-way		
	1-shot	2-shot	3-shot	1-shot	2-shot	3-shot
Matching Net (random)	$69.8\%_{\pm 0.10}$	$93.1\%_{\pm 0.07}$	$98.5\%_{\pm 0.04}$	$67.3\%_{\pm 0.10}$	$91.2\%_{\pm 0.06}$	$97.6\%_{\pm 0.06}$
Matching Net (balanced)	$97.9\%_{\pm 0.07}$	$98.9\%_{\pm 0.07}$	$99.2\%_{\pm 0.06}$	$96.5\%_{\pm 0.04}$	$98.3\%_{\pm 0.03}$	$98.7\%_{\pm 0.05}$
Active MN	$97.4\%_{\pm 0.11}$	$99.0\%_{\pm 0.08}$	$99.3\%_{\pm 0.03}$	$94.3\%_{\pm 0.24}$	$98.0\%_{\pm 0.07}$	$98.5\%_{\pm 0.06}$
Min-Max-Cos	$97.4\%_{\pm 0.11}$	$99.3\%_{\pm 0.02}$	$99.4\%_{\pm 0.04}$	$93.5\%_{\pm 0.11}$	$98.4\%_{\pm 0.02}$	$98.8\%_{\pm 0.03}$

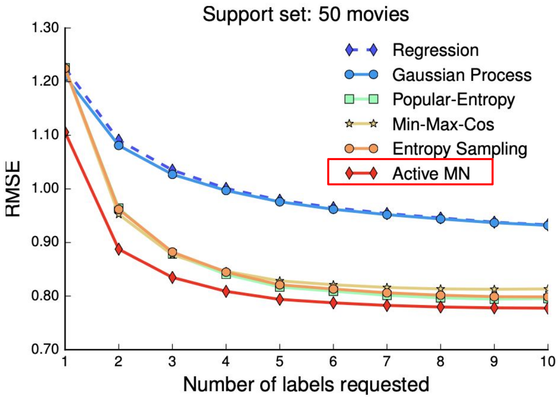
Experimental Evaluation: Data Efficiency

Omniglot Performance





MovieLens Performance



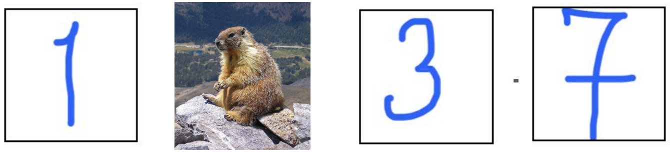
Conclusion

Introduced model that learns active learning algorithms end-to-end.

- Approaches optimistic performance estimate on Omniglot
- Outperforms baselines on MovieLens

Critique/Discussion Points

examples



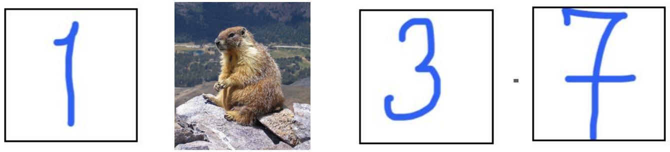
probe

- Controller doesn’t condition its label requests on the probe item

Image source: https://en.wikipedia.org/wiki/File:Marmot-edit1.jpg,

Critique/Discussion Points

examples



probe

- Controller doesn’t condition its label requests on the probe item
- In Matching Networks, the embeddings of the examples don’t depend on the probe item

Image source: https://en.wikipedia.org/wiki/File:Marmot-edit1.jpg,

Critique/Discussion Points

- Active learning is useful in settings where data is expensive to label, but meta-learned active learning requires lots of labeled data for training, even if this labeled data is spread across tasks. Can you think of domains where this is / is not a realistic scenario?

Critique/Discussion Points

- Active learning is useful in settings where data is expensive to label, but meta-learned active learning requires lots of labeled data for training, even if this labeled data is spread across tasks. Can you think of domains where this is / is not a realistic scenario?
- In their ablation studies, they observed that taking out the context-sensitive encoder had no significant effect. Are there are applications where you think this encoder could be essential?
- In this work, they didn't experiment with NLP tasks. Are there any NLP tasks you think this approach could help with?