ANML

An exciting exploration by Abhishek Poswal

Table of Contents

Catastrophic Forgetting

Learning to solve new tasks rapidly degrades previously acquired capabilities.

Understanding the problem solutions

01 Use ***replay methods*** such as saving prior experiences and mixing them with newly encountered data to approximate interleaved.

- Expensive in both storage and computation.

02 Use ***selective plasticity*** to limit the extent to which parameters can be modified in response to new data.

- Manually designed heuristics and scales poorly on larger tasks.

03 Use ***elastic weight consolidation (EWC)*** to alter the plasticity of a given parameter in proportion to manually-selected criteria of Fisher information, which approximates how important each parameter was for solving prior tasks.

- Interesting but falls into manual approaches to reduce CF.
- Computationally heavy depending on the tasks. *(working on evidence)

Solution

Directly incentivize the creation of maximally sparse or disjoint representations, with goal of ***minimizing interference*** between activations.

Common misunderstanding from the above statement is optimizing for sparsity which results into dead neurons after training a model. Therefore, the takeaway should be optimizing for learning without forgetting by using ***context-dependent gating.***

Conflict

OML (Online aware Meta-Learning) employs a MAML-style algorithm to produce a representation (set of neural network layers) that, when frozen and used by additional, downstream neural network layers, minimizes catastrophic forgetting in those downstream layers.

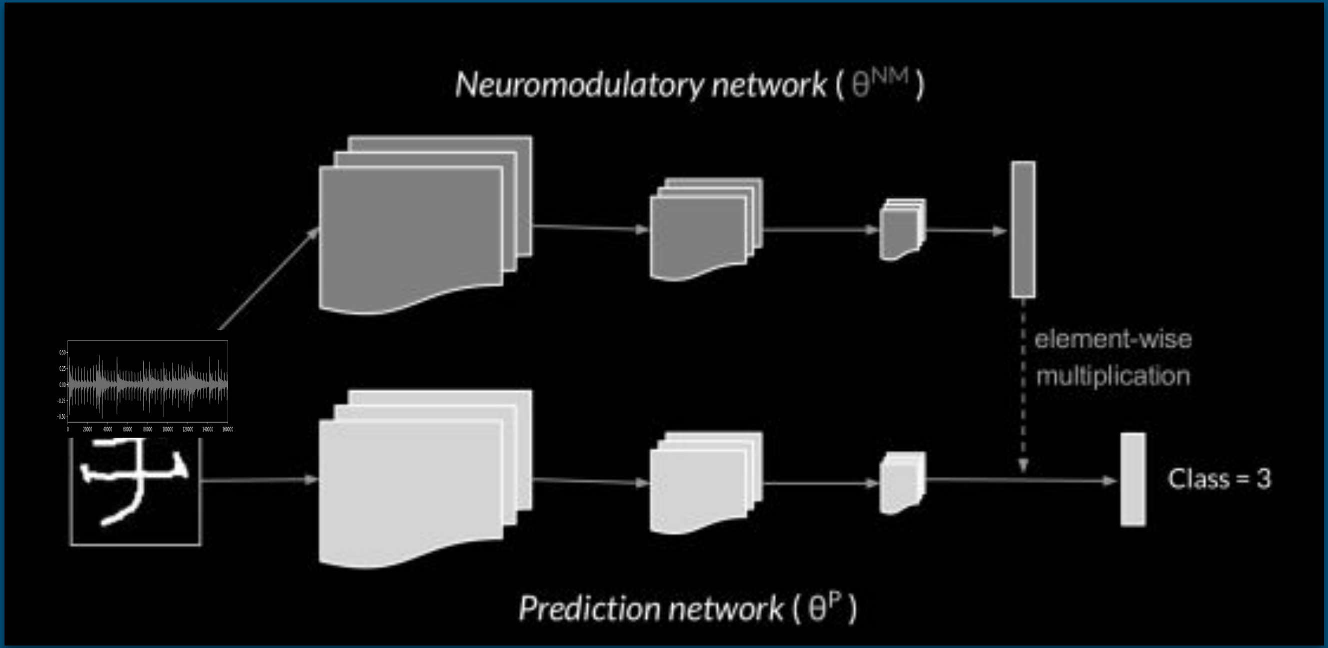
A Neuromodulatory Meta-Learning (ANML)

ANML metalearns a context-dependent gating function (neuromodulatory network) that enables continual learning in another neural network (prediction network), rather than meta-learning representations as in OML.

Selective Activation: The neuromodulatory network has the flexibility to explicitly turn on and off activations in subsets of the prediction network conditioned on the input.

Selective activation in turn enables selective plasticity because the strength of backward gradients is a function of how active neurons were during the (modulated) forward pass, indirectly controlling which subset of the network will learn for each type of input. By leveraging metalearning to optimize when and where to gate activations to maximize continual learning, our approach explores the potential for solutions beyond hand-designed selective plasticity strategies.

A Neuromodulatory Meta-Learning (ANML)



Dataset (Youtube & Google Audioset)

Dataset

Youtube Audioset (632 classes)

- balanced_train_segments.csv
- eval_segments.csv

Datapoint description:

```
YTID, start_seconds, end_seconds, positive_labels
--PJHxphWEs, 30.000, 40.000, "/m/09x0r,/t/dd00088"
```

Google Audioset (35 classes)

- X_train.npy & y_train.npy
- X_test.npy & y_test.npy

Bash Script to Scrape from Youtube

```bash
#sampling rate can be increased or decreased
SAMPLE_RATE=22050

# fetch_clip(videoID, startTime, endTime)
fetch_clip() {
  echo "Fetching $1 ($2 to $3)..."
  outname="$1_$2"
  if [ -f "$./train1/{outname}.wav" ]; then
    echo "Downloaded Already."
    return
  fi

  youtube-dl https://youtube.com/watch?v=$1 \
    --quiet --extract-audio --audio-format wav \
    --output "$outname.%(ext)s"
  if [ $? -eq 0 ]; then
    # If we don't pipe `yes`, ffmpeg seems to steal a
    # character from stdin. I have no idea why.
    yes | ffmpeg -loglevel quiet -i "./$outname.wav" -ar $SAMPLE_RATE \
      -ss "$2" -to "$3" "./train1/${outname}_out.wav"
    mv "./train1/${outname}_out.wav" "./train1/$outname.wav"
    # gzip "./train1/$outname.wav"
  else
    # Give the user a chance to Ctrl+C.
    sleep 1
  fi
}

grep -E '^[^#]' | while read line
do
  fetch_clip $(echo "$line" | sed -E 's/, / /g')
done
```

Audioset Preprocessing

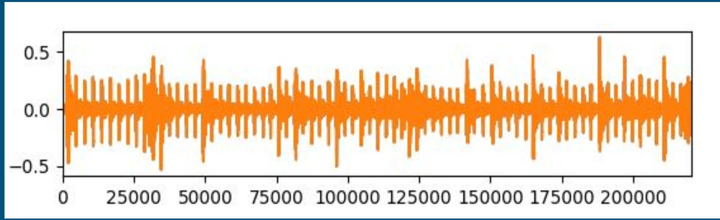
Libraries and Processing

- tensorflow.transforms/scipy
- tensorflow.keras
- soundfile
- PIL
- youtube-dl

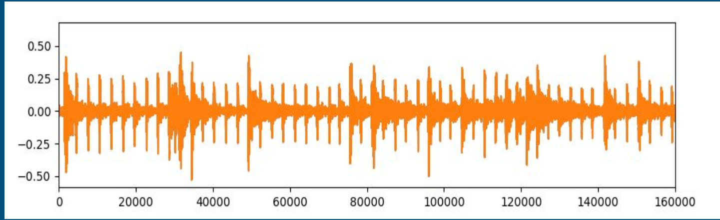
➔ Taking 10 sec samples
➔ Sampling rate at 16kHz
➔ Normalization
➔ Padding
➔ Hopping every 10ms

Waveform Visualization

Waveform before preprocessing:



Waveform after preprocessing:



Settings Used

Architecture

Network Neuromodulation:

```
('conv1_nm', [nm_channels, 3, 3, 3, 1, 0]),
('bn1_nm', [nm_channels]),
('conv2_nm', [nm_channels, nm_channels, 3, 3, 1, 0]),
('bn2_nm', [nm_channels]),
('conv3_nm', [nm_channels, nm_channels, 3, 3, 1, 0]),
('bn3_nm', [nm_channels]),

('nm_to_fc', [size_of_representation, size_of_interpreter]),
```

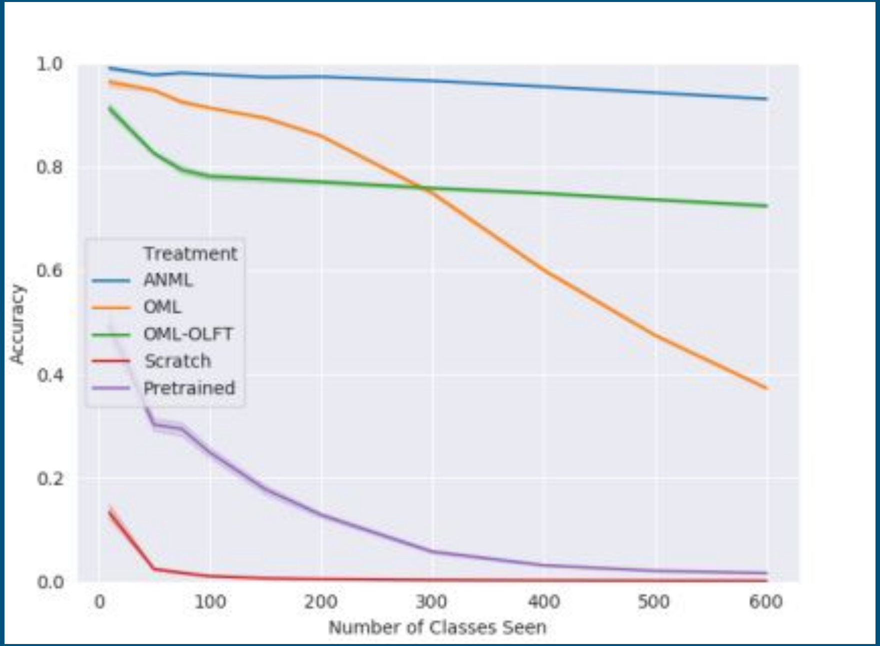
Prediction Network:

```
('conv1', [channels, 3, 3, 3, 1, 0]),
('bn1', [channels]),
('conv2', [channels, channels, 3, 3, 1, 0]),
('bn2', [channels]),
('conv3', [channels, channels, 3, 3, 1, 0]),
('bn3', [channels]),
('fc', [1000, size_of_representation]),
```

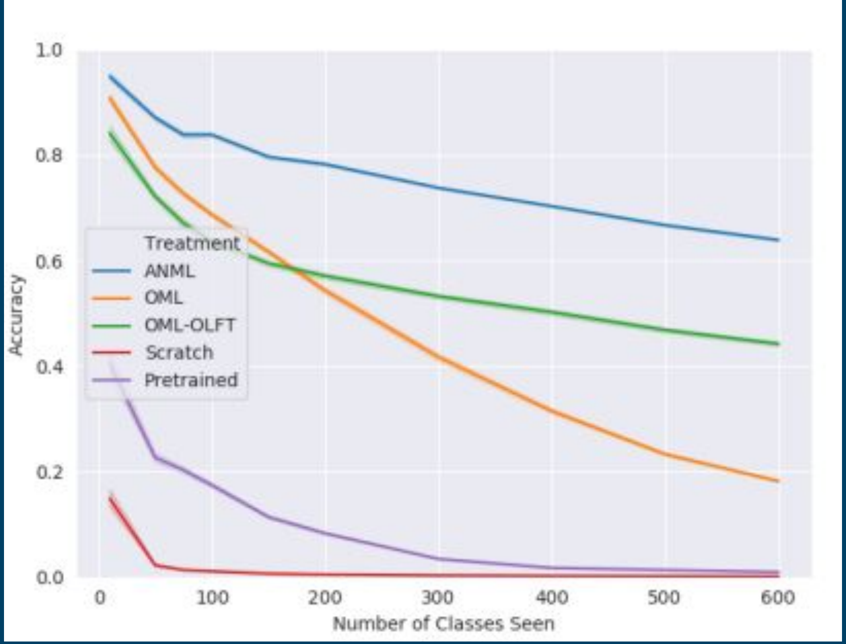
Parameters

```
class Params:
  sample_rate: float = 2050.0
  stft_window_seconds: float = 0.025
  stft_hop_seconds: float = 0.010
  mel_bands: int = 64
  mel_min_hz: float = 125.0
  mel_max_hz: float = 7500.0
  log_offset: float = 0.001
  patch_window_seconds: float = 0.96
  patch_hop_seconds: float = 0.48

  @property
  def patch_frames(self):
    return int(round(self.patch_window_seconds / self.stft_hop_seconds))

  @property
  def patch_bands(self):
    return self.mel_bands

  num_classes: int = 35
  conv_padding: str = 'same'
  batchnorm_center: bool = True
  batchnorm_scale: bool = False
  batchnorm_epsilon: float = 1e-4
  classifier_activation: str = 'sigmoid'

  tflite_compatible: bool = False
```

Training & Testing Results (Omniglot)



Training



Testing

Aha!

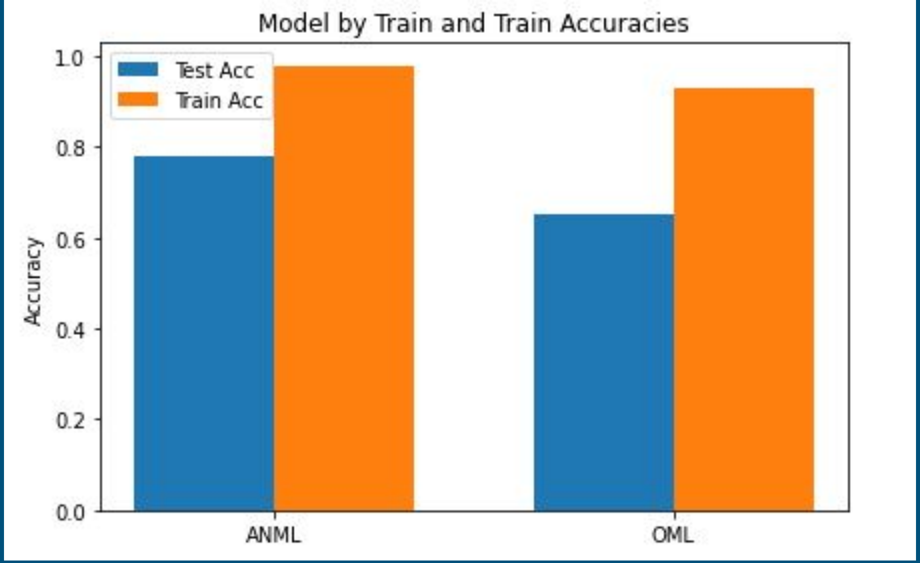
More discoveries

Google Audioset Results

What did we learn after testing?

- ANML is a state of the art machine learning algorithm, that when combined with Meta-Learning produces the most accurate results as opposed to its challengers such as OML.

- With about 79% accuracy on the meta-testing test set, ANML beat OML by 14% on 35 classes, although the accuracy for only differed by 6% with ANML at 98.4%.



Experiment trends (Youtube Dataset)

```
experiment   : INFO     step: 40        training acc 0.23809523809523808
Task 629 has been added to the list
Task 682 has been added to the list
Task 941 has been added to the list
Task 307 has been added to the list
Task 252 has been added to the list
Task 213 has been added to the list
Task 188 has been added to the list
Task 116 has been added to the list
Task 580 has been added to the list
```

14/08/2021

```
experiment   : INFO     step: 1800        training acc 0.6904761904761905
Task 658 has been added to the list
Task 650 has been added to the list
Task 893 has been added to the list
Task 535 has been added to the list
Task 943 has been added to the list
Task 950 has been added to the list
Task 742 has been added to the list
Task 171 has been added to the list
Task 324 has been added to the list
Task 4 has been added to the list
```

19/08/2021

```
Fetching gU1PDdat7Nw (170.000 to 180.000)...
ERROR: Private video
Sign in if you've been granted access to this video
Fetching gUBxjQ6u1Vk (100.000 to 110.000)...
ERROR: Private video
Sign in if you've been granted access to this video
Fetching gUKF_nucISY (30.000 to 40.000)...
Fetching gUnmimbP_lE (10.000 to 20.000)...
Fetching gUpHlWC4gZ0 (10.000 to 20.000)...
Fetching gUxRhvNbsoE (10.000 to 20.000)...
Fetching gVBPeamgFro (40.000 to 50.000)...
Fetching gVMoI2ukbtc (30.000 to 40.000)...
Fetching gVY3eNmnr5k (60.000 to 70.000)...
Fetching gV_lRnOpMm0 (100.000 to 110.000)...
Fetching gVfqMDr86Ks (30.000 to 40.000)...
ERROR: Video unavailable
Fetching gVgfohZexjk (510.000 to 520.000)...
Fetching gVxVSH8_Z2U (50.000 to 60.000)...
Fetching gVynDIr35Ok (220.000 to 230.000)...
Fetching gW33LYEvoaw (140.000 to 150.000)...
Fetching gW9U5w-IaTo (120.000 to 130.000)...
Fetching gWMrFrICmNw (30.000 to 40.000)...
WARNING: unable to download video info webpage: HTTP Err
ERROR: Sign in to confirm your age
This video may be inappropriate for some users.
Fetching gX-llX-PVjw (160.000 to 170.000)...
Fetching gX8S48EH1VE (430.000 to 440.000)...
Fetching gX9OuOKLLTQ (70.000 to 80.000)...
Fetching gXBH4_kdnh0 (320.000 to 330.000)...
Fetching gXFi6q8lQN4 (30.000 to 40.000)...
Fetching gXGvMW-GN2Y (70.000 to 80.000)...
Fetching gXRtfnb_M0U (28.000 to 38.000)...
Fetching gXSHC3rJExc (20.000 to 30.000)...
Fetching gXpwexewzTE (20.000 to 30.000)...
ERROR: Video unavailable
Fetching gY9rtF2vEOQ (20.000 to 30.000)...
Fetching gYBXpwc6gdg (190.000 to 200.000)...
Fetching gYD1y9sEgRc (10.000 to 20.000)...
Fetching gYVQJv7rIhk (30.000 to 40.000)...
Fetching gYnd_jZGGDk (230.000 to 240.000)...
Fetching gZOsVXInWlQ (160.000 to 170.000)...
Fetching gZbQi1Y9nUI (10.000 to 20.000)...
```

Progress

Accomplishment 1

- Understood Meta-Learning and Neuromodulatory Meta-Learning Models.
- Extracted and Preprocessed the datasets.

Accomplishment 2

- Pipelined the audio dataset into the customized model architecture and hypertuned the parameters.
- Trained/Training the dataset.

Attention areas

Challenge 1

- Processing and training large datasets on comparatively low computational power.
- Taking over a week to extract and download google dataset.

Challenge 2

- Results contrastive of the expectations.

Compression of Data-points using Quantization

Quantization

Here, we propose replay using memory indexing, a novel method that is heavily influenced by biological replay and hippocampal* indexing theory: experience generates an index in the hippocampus that encodes locations in cortical space.

1. A streaming learning model that implements hippocampal indexing theory using tensor quantization to efficiently store hidden representations (e.g., CNN feature maps) for later replay. We should implement this compression using Product Quantization.

2. Similar approach has been used in REMIND CL paper. They have demonstrated the robustness of quantization by pioneering streaming Visual Question Answering (VQA), in which an agent must answer questions about images that cannot be readily done with existing models.

*region of brain primarily associated with memory.

References

1. 2017. PNAS. Overcoming catastrophic forgetting in neural networks. Kirkpatrick et al. **+ EWC**
2. 2018. Trans on PAMI. Learning without Forgetting. Li and Hoiem. + **LwF**
3. 2017. CVPR. **iCaRL**: Incremental Classifier and Representation Learning. Sylvestre-Alvise Rebuffi et al.
4. 2020. ECCV. **REMIND** Your Neural Networks to Prevent Catastrophic Forgetting. Tyler L. Hayes et al. **+ uses PQ to compress extracted features as exemplars**
5. 2019. ArXiv. Latent Replay for Real-Time Continual Learning. Lorenzo Pellegrini et al.
6. 2019. NeurIPS. Meta-Learning Representations for Continual Learning. Khurram Javed and Martha White. **+ 1st MetaCL paper (OML)**
7. 2020. ECAI. Learning to Continually Learn. Shawn Beaulieu et al.

Thank you!

Any Questions?