Week 15 Report

Thomas Deatherage
NFHM/BioCosmos
6 December 2024

Time Slot

1) What progress did you make in the last week?

- End of semester presentations
- Met with Roman to discuss presentations, semester wrap-up, next steps
- Kickoff meeting with HAAG-admin

2) What are you planning on working on next?

- Continue with making the unicom code we have distributed again

3) Is anything blocking you from getting work done?

- Nope.

Abstracts & Summaries

Deep learning as a tool for ecology and evolution

Abstract: Deep learning is driving recent advances behind many everyday technologies, including speech and image recognition, natural language processing and autonomous driving. It is also gaining popularity in biology, where it has been used for automated species identification, environmental monitoring, ecological modelling, behavioural studies, DNA sequencing and population genetics and phylogenetics, among other applications. Deep learning relies on artificial neural networks for predictive modelling and excels at recognizing complex patterns.

In this review we synthesize 818 studies using deep learning in the context of ecology and evolution to give a discipline-wide perspective necessary to promote a rethinking of inference approaches in the field. We provide an introduction to machine learning and contrast it with mechanistic inference, followed by a gentle primer on deep learning. We review the applications of deep learning in ecology and evolution and discuss its limitations and efforts to overcome them. We also provide a practical primer for biologists interested in including deep learning in their toolkit and identify its possible future applications.

We find that deep learning is being rapidly adopted in ecology and evolution, with 589 studies (64%) published since the beginning of 2019. Most use convolutional neural networks (496 studies) and supervised learning for image identification but also for tasks using molecular data, sounds, environmental data or video as input. More sophisticated uses of deep learning in biology are also beginning to appear.
Operating within the machine learning paradigm, deep learning can be viewed as an alternative to mechanistic modelling. It has desirable properties of good performance and scaling with increasing complexity, while posing unique challenges such as sensitivity to bias in input data. We expect that rapid adoption of deep learning in ecology and evolution will continue, especially in automation of biodiversity monitoring and discovery and inference from genetic data. Increased use of unsupervised learning for discovery and visualization of clusters and gaps, simplification of multi-step analysis pipelines, and integration of machine learning into graduate and postgraduate training are all likely in the near future.

Summary (Claude AI): Let me provide a comprehensive summary of this review paper on deep learning applications in ecology and evolution.

Key Points:

1. Overview and Growth:
- Deep learning is rapidly being adopted in ecology and evolution, with 589 studies (64%) published since 2019
- The majority of studies use convolutional neural networks (496 studies) for image identification tasks

2. Main Applications:
- Species identification and classification from images, video, and audio
- Environmental monitoring and biodiversity assessment
- Analysis of genetic data and population genetics
- Behavioral studies and movement tracking
- Ecological modeling and species distribution prediction

3. Key Advantages:
- Can handle complex patterns and large datasets
- Excels at feature extraction directly from raw data
- Can combine multiple data types (images, genetic sequences, environmental data etc.)
- Scales well with increasing complexity
- Can solve problems where mechanistic models don't exist

4. Limitations and Challenges:
- Requires large amounts of training data
- Sensitive to bias in input data
- Can be computationally intensive
- Results may lack interpretability ("black box" problem)
- May not outperform simpler methods for small datasets

5. Future Directions:
- Increased automation in biodiversity monitoring
- More unsupervised learning applications
- Better integration with mechanistic modeling
- Improved handling of small datasets
- Integration into graduate/postgraduate training

6. Technical Requirements:
- Most applications use Python or R with libraries like TensorFlow and PyTorch
- GPU computing resources often needed for efficient processing
- Various architectures available (CNNs, RNNs, VAEs, GANs etc.)

This paper provides a thorough overview of how deep learning is transforming ecological and evolutionary research, while acknowledging both its potential and limitations. The authors suggest deep learning will become an integral part of the biologist's toolkit, complementing rather than replacing traditional approaches.

Relevance: Better understanding the current state of ML techniques in ecological research.

Scripts and Code Blocks

Two objectives:

1) Distributed clustering as described in https://github.com/deepglint/unicom/issues/20
2) Distributed UNICOM training.

To point two, this is a part of the UNICOM code (from their repo) that I'm trying to square with the code we've written up to this point. The UNICOM code makes heavy use of torch's distributed utilities:

```
def __init__(
    self,
    dataset,
    num_replicas=None,
    rank=None,
    shuffle=True,
    seed=0,
):
    super().__init__(dataset, num_replicas=num_replicas, rank=rank, shuffle=shuffle)

    # In distributed sampling, different ranks should sample
    # non-overlapped data in the dataset. Therefore, this function
    # is used to make sure that each rank shuffles the data indices
    # in the same order based on the same seed. Then different ranks
    # could use different indices to select non-overlapped data from the
    # same data list.
    self.seed = sync_random_seed(seed)

def __iter__(self):
    # deterministically shuffle based on epoch
    if self.shuffle:
        g = torch.Generator()
        # When :attr:`shuffle=True`, this ensures all replicas
        # use a different random ordering for each epoch.
        # Otherwise, the next iteration of this sampler will
        # yield the same ordering.
        g.manual_seed(self.epoch + self.seed)
        indices = torch.randperm(len(self.dataset), generator=g).tolist()
    else:
        indices = torch.arange(len(self.dataset)).tolist()
    # add extra samples to make it evenly divisible
    # in case that indices is shorter than half of total_size
    indices = (indices * math.ceil(self.total_size / len(indices)))[
        : self.total_size
    ]
    assert len(indices) == self.total_size
    # subsample
    indices = indices[self.rank : self.total_size : self.num_replicas]
    assert len(indices) == self.num_samples
    return iter(indices)
```

Flow Charts/Diagrams

N/A

Documentation

No documentation to add really just yet

Results Visualization + Proof of Work

Next Week's Proposal

- Wrap up meeting with colab
- Continue distributed code
- Test with VLM4Bio on HiPerGator