CS230

Neurobiologically Inspired Encoding and Transfer Learning

George Sivulka
Department of Applied Physics
Department of Mathematics
Stanford University
gsivulka@stanford.edu

1 Introduction

1.1 Background & Related Work

Deep convolutional neural networks (CNNs) have proven to be the most successful models of the nervous system's sensory processing paradigms to date. Significantly, The Baccus Lab at Stanford has achieved state of the art success modeling the retinal processing with a three layer CNN, mimicking the structure and number of cell layers in the retina [1] (Figure 1.1).

While these models have lent significant insights into both the brain's neural computations and the circuit mechanisms that pertain to relevant natural stimuli, there exists additional potential for these biologically motivated systems to inform and augment traditional machine learning tasks. Indeed, as the retina can be thought of as a preprocessor serving all downstream vision tasks, evolutionary consensus would hold that it approximates a near-optimal solution for many tasks. There are currently no transfer learning paradigms or classification models that employ biologically pre-trained networks. The aforementioned CNN model of retinal spiking at different times, referred to as Deep Retina for the remainder of this paper, poses an excellent opportunity to analyze the efficacy of these biologically motivated systems.

This research builds upon previous models of retinal spiking, particularly investigating:

1. A "fully convolutional" Deep Retina model, modifying current retinal prediction models to better serve encoding tasks
2. The efficacy of these learned visual encodings at improving two LSTM-based video classification tasks

1.2 Objective/Model Description

First, to better investigate encoding problems, the Baccus Lab's Deep Retina (DR) model was changed to be "fully convolutional". As retinal data only contains spiking from a few cells with localized receptive fields, this allows the model to learn a filter for each cell. This filter can be employed during encoding tasks to tile that cell's response across input locations.

To do this, Deep Retina's final dense layer, trained to output the spiking of the few cells recorded in the data, is replaced with another convolutional layer (Figure 1.2). The output of this convolutional layer is element wise multiplied by a selection matrix that is incentivized to be one hot

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

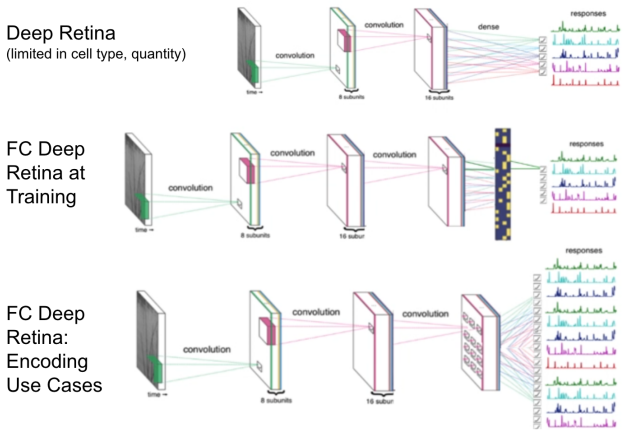

Figure 1: Schematic of the Baccus Lab's "Deep Retina" CNN, trained on data of diverse visual stimuli coupled with electrophysiological recordings from salamander retina.

via a 'semantic' loss regularizer term by Xu et. al. [2]:

$$\mathcal{L}(\text{one} - \text{hot}, \forall\, P_c\ filters) \;=\; -\beta \sum_{i=1}^{c} \log \sum_{j=1}^{n} p_j \prod_{k=1, k\neq j}^{n} (1 - p_k)$$

Thus a model trained on the same sparse cell data as before can learn a convolutional filter for each cell, allowing it to tile the filter of each cell during encoding tasks. As such, once trained (Figure 1.3), the final selection matrix is removed and the learned filters can act on and encode all locations in the latent space.

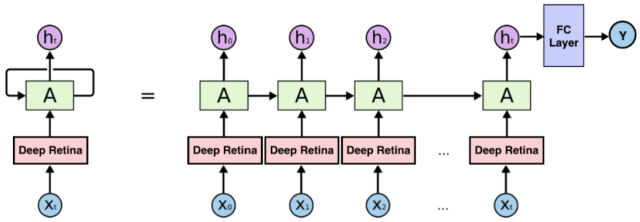

Figure 2: Schematic of the LSTM architecture with input (X) frame bins first encoded by Deep Retina. Video classification (y) is performed on a FC-layer's output of the final internal units (h) of the LSTM.

Next, to analyze the efficacy of these retinal computations in a video classification setting, this research constructed a hybrid "DeepRetina-LSTM". This LSTM takes inputs of binned video frames passed through Deep Retina over the course of a video, and outputs a one-hot video classification label (Figure 2) .

2 Dataset and Features

2.1 Retinal Spiking Dataset

The aforementioned Deep Retina was trained on 40 binned 10ms frames of a natural movie (Figure 3) labeled with the experimentally measured neural spiking data across cells. These neural spikes were measured with a micro-electrode array, recording 60 channels of extracellular electrophysiological signals from a retina exposed to natural movies. Standard spike sorting algorithms were used to read

spike times across 8 cells from the raw data. The 8 cell spike time were then binned in accordance with the window to return 8 firing rates for each labeled video frame example.

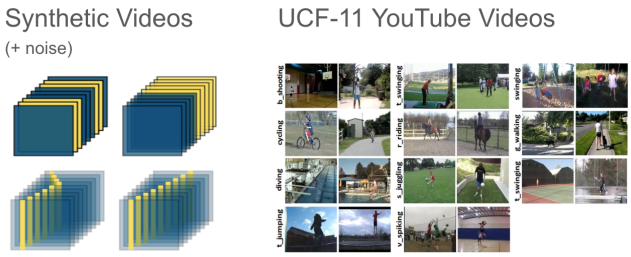

Figure 3: A natural movie next to four of the seven synthetically generated stimuli (Left); Screenshots from the UCF-11 Dataset (Right)

2.2 Synthetic Video Classification Dataset

The first dataset for video classification included a variety of simple artificial stimuli known to evoke retinal response–allowing for a proof of concept first pass at this transfer learning system (Figure 3).

This data was manually constructed with custom num.py methods and independently generated noise (within the datas.py file) and actively loaded into testing and training environments as batches of size [B, T, D, H, W]. Here B = 100 videos in each batch, T = the start times of each 400ms long binned video frames separated by 40ms each, D = 400ms for each frame, and H = W = 50, the height and width of each frame). Accompanying these input video batches, the loader returned label batches of size [B, K] with K = the number of video classes for one-hot labels.

2.3 Real Video Classification Dataset

For the harder problem of video classification of real world actions–the UCF11 (YouTube Action) Dataset was used [3]. Preprocessing involved cropping the video frames' height and width to 240 by 240 pixels, upsampling each frame to match the 40ms frame rate during Deep Retina's training, binning each step into ten 400ms data points, and ensuring at least 100 frames for each video by cutting or padding videos with extra blank frames. Batches were of size 10 due to computational resource limitations during training.

3 Methods

In order to test the trained Deep Retina across its efficacy as both a transfer learning and fixed encoding paradigm, four different models of the aforementioned CNN + LSTM architecture were run for synthetic classification.

As a control, the deep retina architecture was first replaced by a Xavier initialized CNN of the same hyperparameters with untrainable weights. Next, the same Xavier initialized CNN was run with trainable weights. Both cases were then replicated with fully convolutional Deep Retina parameters, set to be trainable and frozen.

For the UCF-11 dataset, due to computational limitations, only two models were run: a DR with frozen weights and a Xavier initialized DR architecture with trainable weights.

4 Results & Discussion

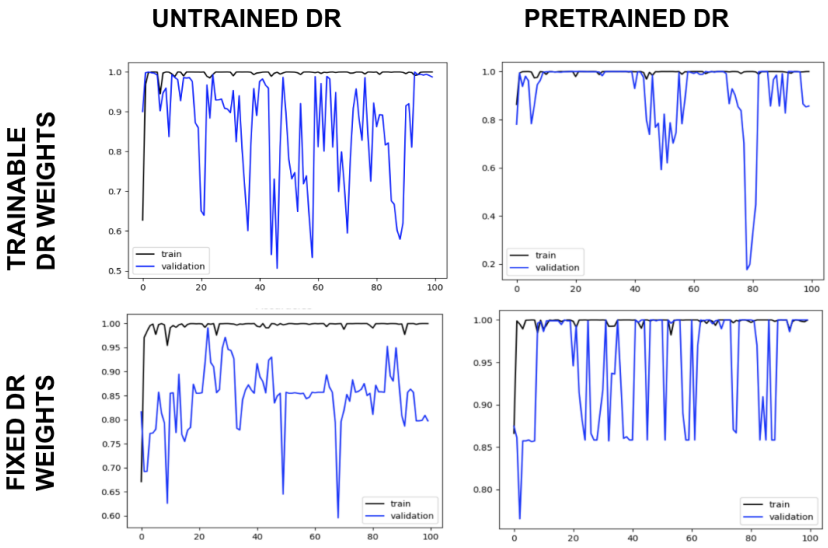

Figure 4: Accuracy curve matrix for synthetic video classification

As the synthetic video classes were generated for the purpose of highly diverse and suggestive retinal responses, this problem was too simple to yield meaningful results from a final accuracy standpoint.

However, comparing accuracy curves across training epoch yields insights confirming the expected benefits of the Deep Retina architecture for this task. Indeed, a pre-trained DR with trainable weights performs best, achieving almost 100% accuracy immediately, and exhibiting the least stochasticity of all models. A pre-trained DR with frozen weights also quickly achieves almost perfect accuracy while exhibiting slightly more stochasticity. Both Xavier initialized DR architectures, with trainable and frozen weights exhibit large amounts of stochasticity and poor classification performance, with the untrainable DR exhibiting the worst validation accuracy of almost 85%.

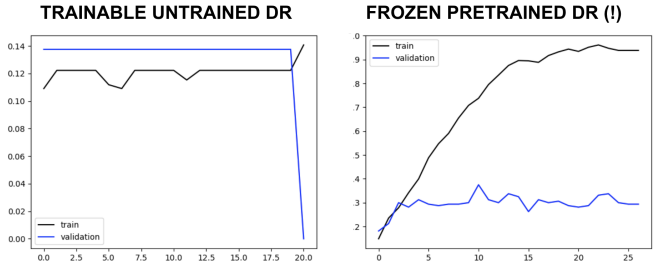

Figure 5: Accuracy curves for UCF-11 video classification

While the UCF-11 classification is more difficult, the frozen DR model still outperforms the trainable randomly initialized model. In fact, while the frozen DR model yields higher accuracy (92% during training and 29% during validation) and learns relatively quickly, the Xavier initialized model fails to converge at all. This issue may potentially be due to poor hyperparameters and is being further investigated.

Significantly, an issue throughout the present analysis is the fact that the BatchNorm (BN) function in Pytorch that this work employed created a discrepancy between training and validation classification tasks. This is due to the fact that it accumulates BN statistics over time during training, but leaves the BN statistics fixed during validation. Both the synthetic and UCF-11 analyses exhibit

potential overfitting to these BatchNorm statistics, leading to poor and highly stochastic validation accuracies for all models. Additional work freezing BN from a preliminary epoch, in order to match statistics during training and validation, is currently being executed.

5 Conclusion/Future Work

While further research validating and improving these results is necessary, the current report presents a sufficient and suggestive–yet preliminary–foray into the potential benefits of biologically-inspired neural networks. As both an encoding and transfer learning paradigm, computations learned from neural spiking in the retina allow models to train faster and more accurately than Xavier initialized versions of the same architectures.

Future research efforts will focus on rectifying validation and training discrepancies, further analyzing UCF-11 performance, and applying these encodings to RL and Meta-RL tasks.

6 Acknowledgements

This research would not be possible without the assistance of Satchel Grant, Josh Melander, and Professor Stephen Baccus of The Baccus Lab at Stanford.

Additionally, this research is indebted to the teaching staff of Stanford's CS 230: "Deep Learning" to which it was submitted as coursework.

References

[1] McIntosh, L., Maheswaranathan, N., Nayebi, A., Ganguli, S., Baccus, S. (2016). Deep learning models of the retinal response to natural scenes. In Advances in neural information processing systems (pp. 1369-1377).

Xu, J., Zhang, Z., Friedman, T., Liang, Y., Broeck, G. V. D. (2017). A semantic loss function for deep learning with symbolic knowledge. arXiv preprint arXiv:1711.11157.

J. Liu, J. Luo and M. Shah, (2009). Recognizing realistic actions from videos "in the wild", CVPR, Miami, FL.