# UNDERSTANDING INTERMEDIATE LAYERS USING LINEAR CLASSIFIER PROBES

**Guillaume Alain & Yoshua Bengio**
Department of Computer Science and Operations Research
Université de Montréal
Montreal, QC. H3C 3J7
guillaume.alain.umontreal@gmail.com

## ABSTRACT

Neural network models have a reputation for being black boxes. We propose a new method to better understand the roles and dynamics of the intermediate layers.

Our method uses linear classifiers, referred to as "probes", where a probe can only use the hidden units of a given intermediate layer as discriminating features. Moreover, these probes cannot affect the training phase of a model, and they are generally added after training.

We demonstrate how this can be used to develop a better intuition about models and to diagnose potential problems.

## 1 INFORMATION THEORY

It was a great discovery when Claude Shannon repurposed the notion of *entropy* to represent information contents in a formal way. It laid the foundations for the discipline of information theory. Naturally, we would like to ask some questions about the information contents of the many layers of convolutional neural networks.

- What happens when we add more layers?
- Where does information flow in a neural network with multiple branches?
- Does having multiple auxiliary losses help? (e.g. Inception model)

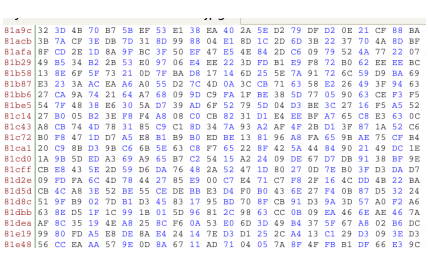
Intuitively, for a training sample $x_i$ with its associated label $y_i$, a deep model is getting closer to the correct answer in the higher layers. It starts with the difficult job of classifying $x_i$, which becomes easier as the higher layers distill $x_i$ into a representation that is easier to classify. One might be tempted to say that this means that the higher layers have more *information* about the ground truth, but this would be incorrect.

Here there is a mismatch between two different concepts of information. The notion of entropy *fails* to capture the essence of those questions. This is illustrated in a formal way by the *Data Processing Inequality*. It states that, for a set of three random variables satisfying the dependency $X \rightarrow Y \rightarrow Z$, then we have that $I(X;Z) \leq I(X;Y)$ where $I(X;Y)$ is the mutual information. Figure 1 illustrates this concept.

## 2 PROBES

The purpose of using many deterministic layers that they perform useful transformations to the data with the goal of *ultimately fitting a linear classifier at the very end*. They are a tool to transform data into a form to be fed to a boring linear classifier.

With this in mind, it is natural to ask if that transformation is sudden or progressive, and whether the intermediate layers already have a representation that is immediately useful to a linear classifier. We refer the reader to Figure 2 for a diagram of probes being inserted in the usual deep neural network. The conceptual framework that we propose is one where the intuitive notion of *information* is equivalent with *immediate suitability for a linear classifier* (instead of being related to entropy).



(a) hex dump of picture of a lion

(b) same lion in human-readable format

Figure 1: The hex dump represented on the left has more information contents than the image on the right. Only one of them can be processed by the human brain in time to save their lives. Computational convenience matters. Not just entropy.

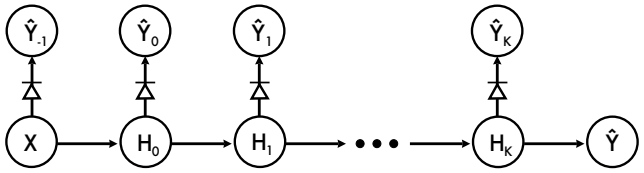
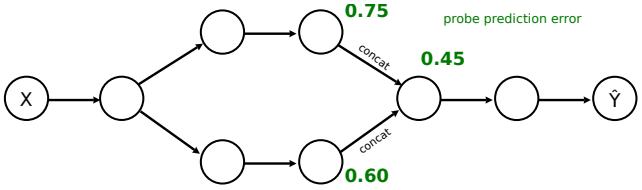



Figure 2: Probes being added to every layer of a model. Note that **the model parameters are not affected by the probes**. We add a little diode symbol through the arrows to indicate that the gradients will not backpropagate through those connections (implemented with `tf.stop_gradient` in tensorflow).

The authors of Donahue et al. present the idea of transferring features from the last two layers of a model to different target domain and fitting a new linear classifier on them. There are some common themes in our work, but the purpose and methodology are different.

### 2.1 PROBES ON BIFURCATING TOY MODEL

Here we show a hypothetical example in which a model contains a bifurcation with two paths that later recombine (through concatenation, but addition would work also). We are interested in knowing whether those two branches are useful, or whether one is potentially redundant or useless.




For this hypothetical situation, we indicate the probe prediction errors on the graphical model. The upper path has a prediction error of $0.75$, the lower path has $0.60$, and their combination has $0.45$. Although the upper path has "less information" than the lower path, we can see here that it is not redundant information, because when we concatenate the features of the two branches we get a prediction error of $0.45 < 0.60$.

If the concatenated layer had a prediction error of $0.60$ instead of $0.45$, then we could declare that the above branch did nothing useful. It may have nonzero weights, but it's still useless.

Naturally, this kind of conclusion might be entirely wrong. It might be the case that the branch above contains very meaningful features, and they simply happen to be useless to a linear classifier applied right there. The idea of using linear classification probes to understand the roles of different branches is suggested as a heuristic instead of a hard rule.

## 3 PROBES ON INCEPTION V3

We have performed an experiment using the Inception v3 model on the ImageNet dataset (Szegedy et al., 2015; Russakovsky et al., 2015). We show using colors in Figure 3 how the predictive error of each layer can be measured using probes. This can be computed at many different times of training, but here we report only after minibatch 308230, which corresponds to about 2 weeks of training.

One of the particularities of this model is that it has an auxiliary head, so it would be natural to look at the error reported by the probes on that auxiliary head during training. Does it come down faster than on the main branch of the model ?

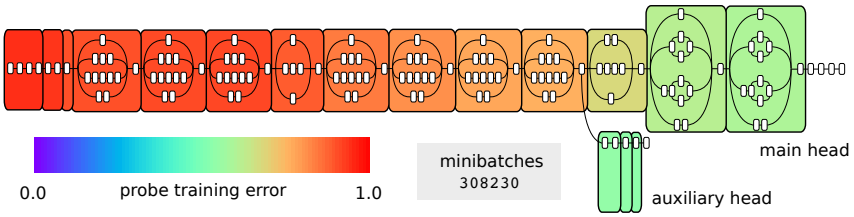



Figure 3: The auxiliary head, shown under the model, was observed to have a prediction error that was slightly better than the main head. This is not necessarily a condition that will hold at the end of training, but merely an observation.

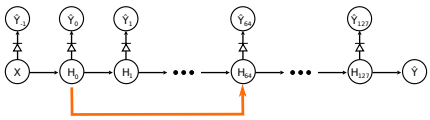
## 4 PATHOLOGICAL BEHAVIOR ON SKIP CONNECTIONS

In our paper we investigate two ways to modify a deep model in order to facilitate training. One of the techniques that we try is to add a very long skip connection that bypasses the first half of the model completely (figure 4a). We picked the model to be pathologically deep, so much so that it would not be trainable without the help of the extra skip connection.

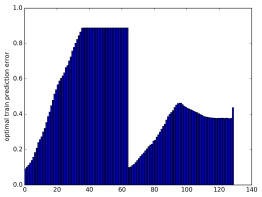
However, using probes we show that this solution is not working as intended, because half of the model is actually dead. The skip connection left a dead segment and skipped over it.

The lesson that we want to show the reader is not that skip connections are bad. Far from it. Our goal here is to show that linear classification probes are a tool to understand what is happening internally in such situations. Sometimes the successful minimization of a loss fails to capture important details.

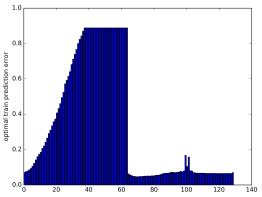
Additionally, by plotting the probes errors for each layer at multiple training moments (4b, 4c), we can also observe the dynamics of training. We have observed a kind of domino effect, whereby the predictive power of the features of any layer $k$ tend to increase before that of layer $k+1$. We provide videos on `https://youtu.be/x8j4ZHCR2FI` where this effect can be seen as a kind of ripple spreading through consecutive layers.




(a) Model with 128 layers. A skip connection goes from the beginning straight to the middle of the graph.



(b) probes after 500 minibatches



(c) probes after 2000 minibatches

Figure 4: Pathological skip connection being diagnosed.

### ACKNOWLEDGMENTS

Yoshua Bengio is a senior CIFAR Fellow. The authors would like to acknowledge the support of the following agencies for research funding and computing support: NSERC, FQRNT, Calcul Québec, Compute Canada, the Canada Research Chairs and CIFAR. Thanks to Nicolas Ballas for fruitful discussions, to Reyhane Askari and Mohammad Pezeshki for proofreading and comments, and to all the ICLR reviewers for their comments.

## REFERENCES

Y Bengio, Paolo Frasconi, and P Simard. The problem of learning long-term dependencies in recurrent networks. In *Neural Networks, 1993., IEEE International Conference on*, pp. 1183–1188. IEEE, 1993.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition.

Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, pp. 91, 1991.

Kevin Jarrett, Koray Kavukcuoglu, Yann Lecun, et al. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153. IEEE, 2009.

David MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.