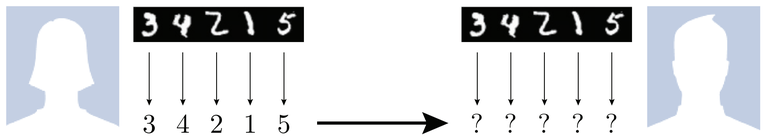
The Description Length of Deep Learning Models

Léonard Blier (École Normale Superieure, Paris), Yann Ollivier (FAIR)

1. Learning models: A compression problem

The compression viewpoint on machine learning [5, 2] holds that a good model of data is a model that is good at losslessly compressing the data, including the cost of describing the model itself. This automatically penalizes more complex models.
Given their huge number of parameters, are deep neural networks able to compress data, including the cost of describing the weights?

Alice wants to transmit some information to Bob efficiently. She has a dataset $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$ where $x_1, ..., x_n$ are some inputs and $y_1, ..., y_n$ some labels. Bob also has the data $x_1, ..., x_n$, but he does not have the labels. Alice can send the labels directly, or she can send a model that allows Bob to recompute the labels from the inputs.



2. Coding data with a probabilistic model

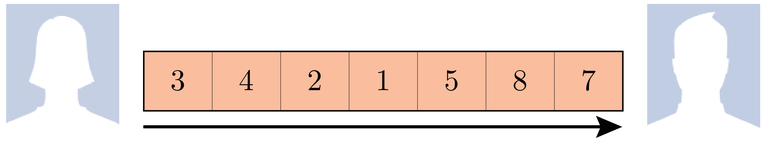
Shannon–Huffman code [1]: Suppose that Alice and Bob have agreed in advance on a model p, and both know the inputs $x_{1:n}$. Then there exists a code to transmit the labels $y_{1:n}$ losslessly with codelength

$$L_p(y_{1:n}|x_{1:n}) = -\sum_{i=1}^{n} \log_2 p(y_i|x_i)$$

- If the model p predicts y_i well, for example $p(y_i|x_i) = 0.9$, the codelength of y_i with p is only 0.15 bits. If the model p is wrong, for example $p(y_i|x_i) = 0.01$, the codelength of y_i with p is 6.64 bits.
- This coincides with the *cross-entropy loss*.

3. Baseline for compression: Uniform encoding

Alice can send the labels without using the inputs, with the uniform encoding: $L^{\text{unif}}(y_{1:n}|x_{1:n}) = n \log_2 K$

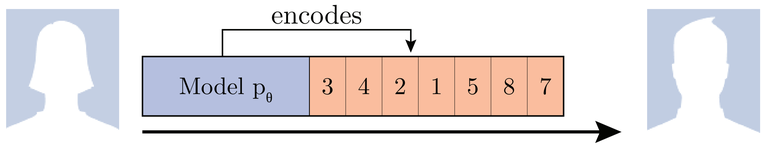


On a classification task with 10 classes, each label costs 3.32 bits to encode with the uniform code. This baseline for compression is 199 kbits for MNIST and 166 kbits for CIFAR10.

4. Send a model, then send the data

Alice can first send a model p_{θ^*}, and then use it to encode the data:

$$L^{\text{2-part}}_{\theta^*}(y_{1:n}|x_{1:n}) := L_{\text{param}}(\theta^*) - \sum_{i=1}^{n} \log_2 p_{\theta^*}(y_i|x_i)$$



- Alice can encode θ^* with the float32 binary encoding. In deep learning, this is much worse than the uniform encoding.
- *Network compression* [3] tools allow for more efficiency in sending the model, but this is still much worse than the uniform encoding.

5. Variational code

If Alice and Bob agreed on a Bayesian prior over θ, Alice can first send a probability distribution β over θ, which costs $\text{KL}(\beta\|\text{prior})$, and then send the data.

$$L^{\text{var}}_{\beta}(y_{1:n}|x_{1:n}) = \text{KL}(\beta\|\alpha) - \mathbb{E}_{\theta\sim\beta}\left[\sum_{i=1}^{n} \log_2 p_\theta(y_i|x_i)\right]$$

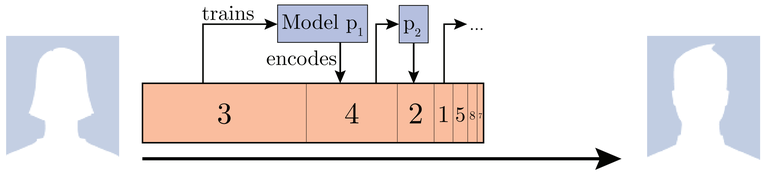
- If β is a multivariate Gaussian, L^{var}_{β} can be optimized by SGD.
- Variational inference is known as a regularization method for deep learning models [4].

6. Coding the data online

Alice can send the data online: Each time Bob receives a data, they both re-train the model using all data already sent. This updated model is then used to encode the next data.

$$L^{\text{preq}}(y_{1:n}|x_{1:n}) = \sum_{i=1}^{n} -\log_2 p_{i-1}(y_i|x_i)$$

where p_{i-1} is the model trained with the data $(x_{1:i-1}, y_{1:i-1})$.



7. Experimental results

Compression bounds obtained with deep learning models, for several codes. We used many architectures (Shallow networks, MLP, LeNet, VGGs), and selected the best ones for compression.

Code	MNIST			CIFAR10		
	Codelength (kbits)	Comp. Ratio	Test Acc	Codelength (kbits)	Comp. Ratio	Test Acc
Uniform	199	1.	10%	166	1.	10%
float32 2-part	> 8.6Mb	$> 45.$	98.4%	> 428Mb	$> 2500.$	**92.9%**
Network compr.	> 400	$> 2.$	98.4%	> 14Mb	$> 83.$	**93.3%**
Variational	22.2	0.11	98.2%	89.0	0.54	66,5%
Online	**4.10**	**0.02**	**99.5%**	**45.3**	**0.27**	**93.3%**

Deep Learning models with the online code do compress MNIST and CIFAR10 well. Moreover, in our experiments the models which were the best for compression were also the best for accuracy.

8. Conclusion

- Variational methods yields surprisingly inefficient codelengths, despite explicitly minimizing this criterion. This might explain why variational inference as a regularization method often does not reach optimal test performance.
- Despite their many parameters, deep learning models do compress the data well, even when accounting for the cost of describing the model.
- This is consistent with Solomonoff's theory of induction [5] and Chaitin's hypothesis that "*comprehension is compression*".

9. References

[1] D. J. C. Mackay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.
[2] P. D. Grünwald. The Minimum Description Length principle. MIT press, 2007.
[3] S. Han, H. Mao, and W. J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv:1510.00149, 2015a.
[4] G. E. Hinton and D. Van Camp. Keeping Neural Networks Simple by Minimizing the Description Length of the Weights. COLT, 1993.
[5] R. Solomonoff. A formal theory of inductive inference. Information and control, 1964.