1 Introduction

Staring with : Sparsity in Deep Learning .

- 13:00: Sparsity can reduce the memory foot print as well as **shorten training time**
- This looks like its going to be a good one.
- You should reviews:
  1. Translational Equivaraiance
  2. REcurrence
  3. Structured weight sharing
  4. POoling
  5. Locality
- Seems like [Friston 2008] could be a good read - the more neurons a biological brain has, the sparser it gets[Herculano-Houzel et al 2021] (Woof that read).
- Human brain starts, sparse, hasn an early phase of densification followed by massive pruning, and then remains at a relatively stable sparsity level. Yet even fully grown brains change up to 40% of their synapses every day. [Hawkins 2017]
- MOdels can get so dense that 95% of its parameters can be predicted from the remaining 5% [Denil et al. 2014]. This coud possible because over-parameterized models are easier to train(with SGD).
  - So basically over-parameterizaion is initially wanted because it leads to strong "convexity-like property">
  - 13:25 : Of course, this can *still* lead to overfitting.
    - **Thus, sparsification *can* be seen as a form of regularization**
- **13:28: The framework of *Minimum Description Length* Provides a Baysesian interpretation and a clear interpretation as data compression**
- // Weird comment: Can sparsification can somehow be a way to interpret habits?
- **Sparsified Models can provide good interpretability(this could go hand in hand with training stearing using VOG)**
- **Adversarial attacks** is basically attempting to fool models with deceptive data. Its aim is to caus emalfunction in a machine learning model(especially with inferring). Yeah so using very specifically crafted attacks you can deceive a machine learning algorithm.

  - Sparsity can help agains adversarial attacks!!

- What: State of the art NLP networks can take up to 350GiB of parameters.
- 14:09: It seems to be a common thing to re-add connections during training to maintain a constant model complexity after sparsification.

2 Overview of Model Compression Techniques

The main paper introduces 6 main model compression techniques:

1. **Down-SIzing Models** Creating a smaller dense network to solve the same task.(Pretty straightforward)
2. **Operator Factorization** Decomproses operators. e.g. decomposing matrix multiplication of dense layers into small operations(Like what Ed is doing). Methods for decomposition can go as follow;
    1. Matrices : using singular value decomposition[Sainath et al. 2013]
    2. Tensors : tensor train decomposition[Kanjilal et al. 1993]
3. **Value Quantization** finds low-precision encoding for parameters in the networks. Im guessing this doesnt solve computation as much as pruning
4. **Value Compression**
5. **Parameter Sharing** can lead to model compression by exploting redundancy in parameter space.
6. Good ol' **Sparsification**: Models can still operate in high-dimensional feature spaces but reduce the representational complexity by only using a subset of the dimensions at a time.