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.

Paper says sparsification can be one of the most powerful of the techniques above.

Model sparsification *for inference* has the most amount of research done, with sparsification *for training* comes at second with "Epheremral" sparsification in a close third

- *Inference* is just the normal training *forward pass* but for the purpose of prediction/estimation/whatchumacalit.
- 16:25 : Nice Description of Jacobian Matrix: It encodes the rate of change of a given vector-valued function's outputs with respect to its inputs.
- **Hessian Matrix**: For a twice-differentiable Loss L, it is the matrix of second order derivaties of the loss function with respect to teh weights. Tends to be expressed as : $\Delta_w^2 L$.

  - **Intuitively its roles is to express the local geometry("Curvature") of the loss around a given point** w

- So I finally, yes finally, found that the probailisitc formulation to DL comes from [Martens and Grosse 2015]. This is where you assume that the input vectors x re drawn from a distribution $Q_{y|x}$

3 Language Processing(side project)

17:43 : Im reading : "All-But-The-Top: Simple and Effective Post-Processing for Word Representations".

- Popular Real-valued word representations are *word2vec* and *GloVe*
- They aim to capture linguistic regularities.
- Words should not really be modeled as discrete atomic units because that doesnt quite catpure the relation between the words

Well im new to this so im not getting much of what this paper is trying to say. So Ill jump online and see if I can read some of the basics.

Word Embedding A representation of a word that can be processed/understood by a machine.

Meaning of *Meaning* Idea that is represented by a word, phrase, etc.

Wordnet Taxonomy that has **hypernysms** (is-a) relation ship. They creator defines it as a large lexical database of English. Nouns, verbs, adjectives and adverbs are groups into sets of cognitive synonyms. e.g. synonyms for word 'good' will be grouped into (full,good), (estimable,good,honorable, respectable). i.e. group is kind of its own concept.

Wordnet is not really capale of expressing nuances of words

- All NLPs nowadays use words as atomic symbols. i.e. represented binarylly (?).e.g. [0 0 0 0 0 1 0 0]. This discrete representation