IMPERIAL

Model Compression

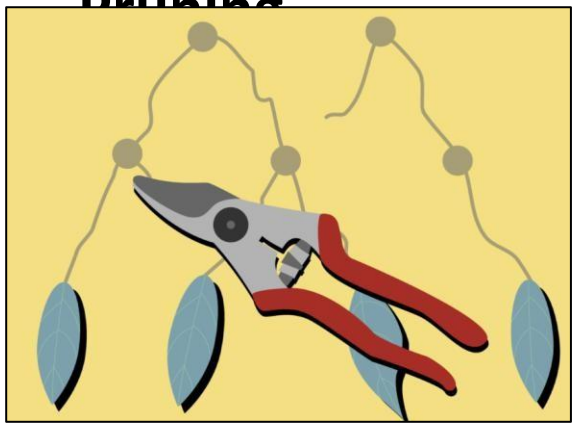
1/12/2025

Shamsuddeen Muhammad
Google DeepMind Academic Fellow,
Imperial College London
https://shmuhammadd.github.io/

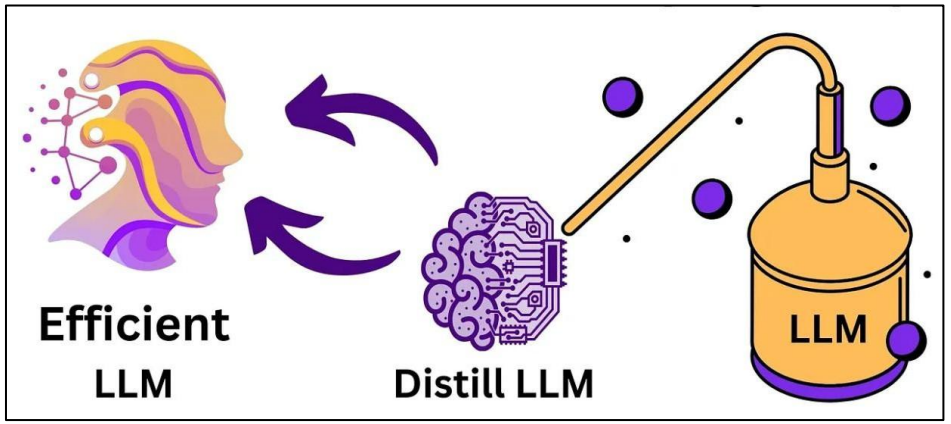
Idris Abdulmumin
Postdoctoral Research Fellow,
DSFSI, University of Pretoria
https://abumafrim.github.io/

Model Compression

Model Pruning



Knowledge Distillation



Teacher - Student

Model Pruning

- Pruning means removing parts of a neural network that are not very useful, like removing weak or unnecessary connections between neurons.
- Think of a huge model as a big, overgrown tree. Pruning is like cutting away branches that do not help the tree grow.
- After pruning:
  - The model becomes smaller
  - Runs faster
  - Uses less memory
  - Still performs almost as well, if pruning is done carefully

Model Pruning

- Imagine a neural network as a huge city full of roads. Some roads are very busy. they carry a lot of traffic. Others are nearly empty, almost no cars use them.
- Pruning is like closing those empty, useless roads. The city still functions the same, but more efficiently.
- Neural networks also have **‘busy’** connections and ‘**unused**’ ones.

- Pruning removes the weak connections so the model runs faster, uses less memory, and is easier to deploy, especially important for African NLP environments where compute is limited.”

Model Pruning

- **Basic Idea**: Among billions of parameters, some are bound to be less important than others
- Pruning involves removing weakly important parameters from pre-trained models

- **Lottery ticket hypothesis** (Frankle et al. 2016) suggests that a subnetwork exists for every neural networks, which when trained in isolation, reach test accuracy comparable to the original model.

- Identifying the winning ticket (subnetwork) is crucial, and can be derived by pruning a pre-trained network.

Why Pruning?

- Pruning interacts with compute vs performance tradeoff
- Pruning highlights which weights/layers encode critical knowledge, helping in model interpretation
- Moderate pruning sometimes improves generalization
- **Biological Analogy**: Synaptic pruning in the brain inspires efficient architectures.

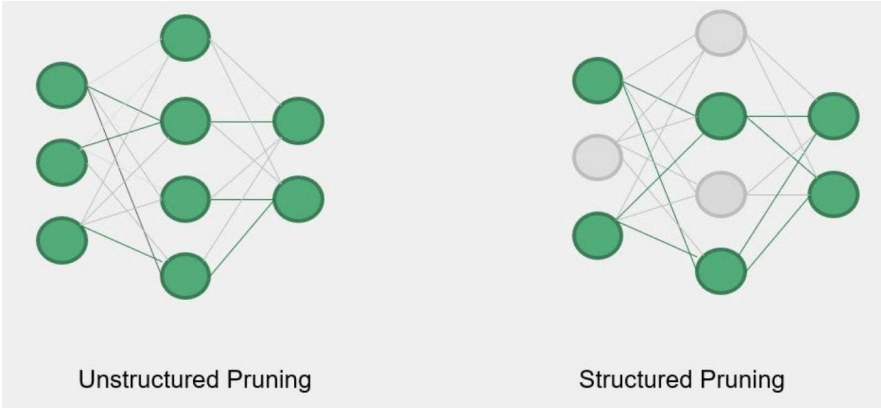


Image source:
https://towardsdatascience.com/iterative-pruning-methods-for-artificial-neural-networks-in-julia-c605f547a485/

Trade-offs with Model Pruning

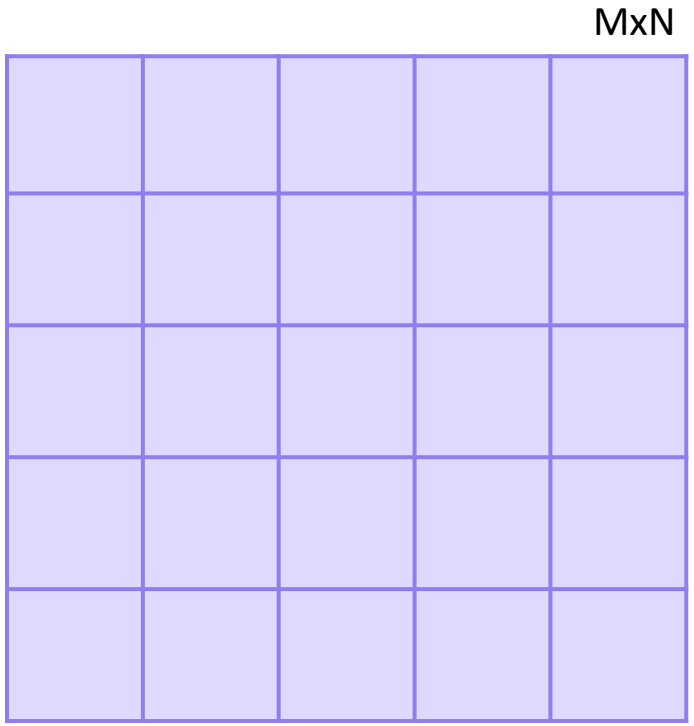
- **Accuracy vs. Efficiency**: Higher pruning saves compute but risks accuracy drops
- **Dependence on Hardware Type:** Several pruning strategies, particularly the unstructured ones depend heavily on hardware configurations for scaling up
- **Generalization**: Pruned models may generalize differently across domains.
- **Fine-tuning Needs**: Some pruning requires extra fine-tuning to recover performance.
- **Universality**: No single pruning strategy works best for all models or tasks.

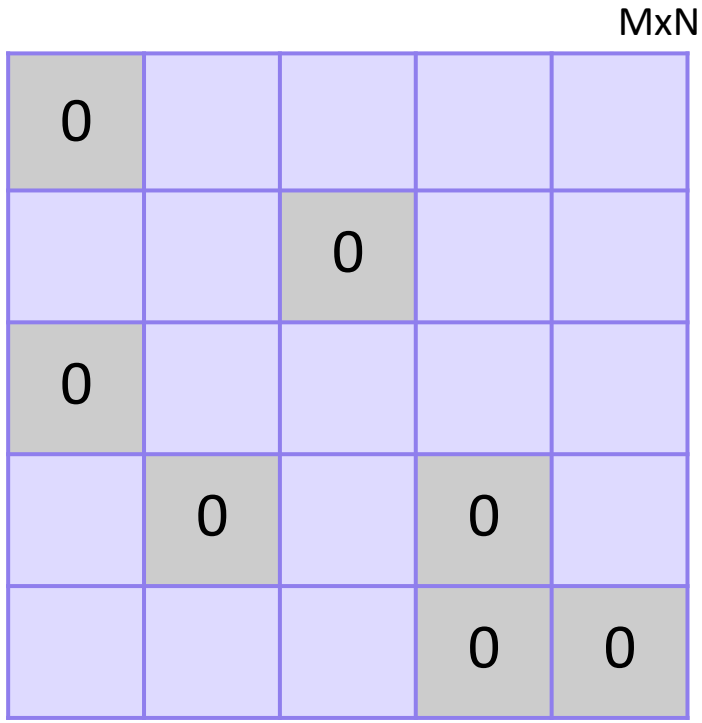
Pruning Techniques

- Two primary techniques to prune models:
  - Unstructured Pruning
  - Structured Pruning

Unstructured Pruning

- Involves zeroing out individual model weights
- No “pattern” in which weights are pruned





Knowledge Distillation (KD)

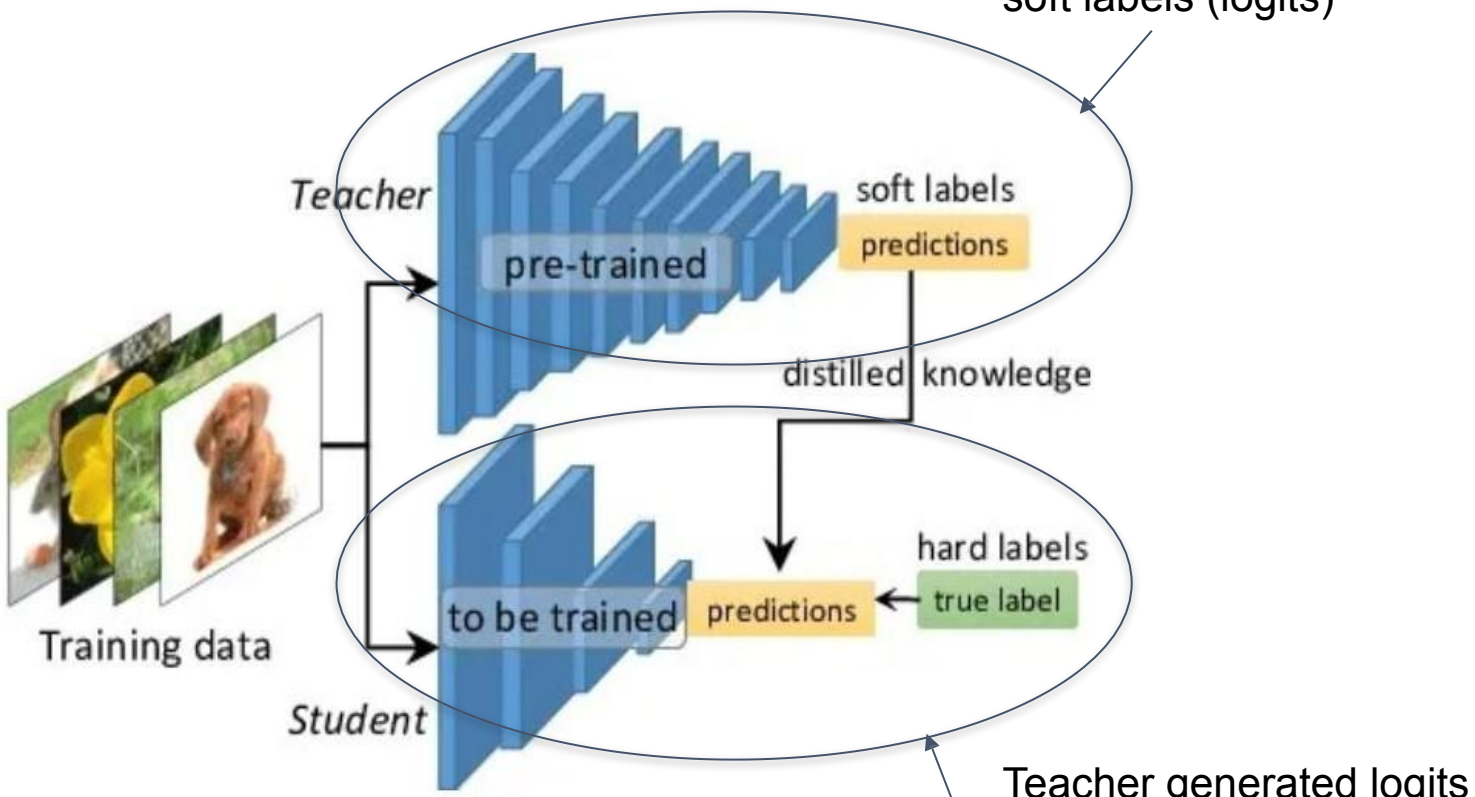
- Knowledge distillation is a model-compression technique in which a large, high-capacity model ("**teacher**") transfers its learned behaviour to a smaller, more efficient model ("**student**").

- Instead of training the student only on gold labels, the student learns to approximate the *output distribution* or *intermediate representations* of the teacher.

- This yields a compact model with competitive performance but lower computational cost.

Knowledge Distillation (KD): Types

Hinton et al., 2015



Core Idea: Knowledge Distillation (KD)

A teacher model produces soft targets, probability distributions over classes or tokens, that contain richer information than hard labels. These soft targets encode:

- Relative class similarities
- Uncertainty patterns
- Implicit linguistic or contextual knowledge

The student model is trained to match these distributions, effectively inheriting the teacher's knowledge.

Knowledge Distillation (KD)

Distillation provides:

- Smaller models for deployment (mobile devices, low-resource settings).
- Faster inference while preserving accuracy.
- Energy-efficient NLP, crucial for African contexts with limited compute.

Well-Known Distilled NLP Models

- **DistilBERT** – 40% smaller, 60% faster, retains ~97% of BERT's performance.
- **TinyBERT** – Distills both hidden states and attention.
- **MiniLM** – Distills attention maps and value-relation matrices.

Categories of KD

- **White-box KD**: Full access to the teacher's internal components (logits, hidden states, attention maps)
- **Meta KD**: Teacher helps guide student training strategies (e.g., data selection, curriculum)
- **Black-box KD**: Only the final output of the teacher is available, e.g., via API

IMPERIAL

Q and A