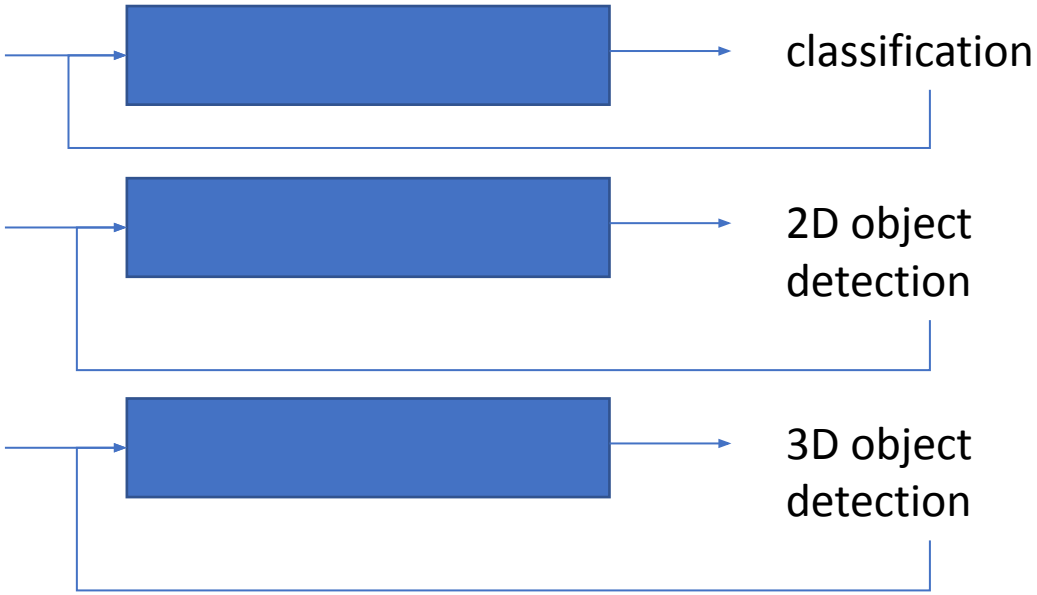
Research For AI Next

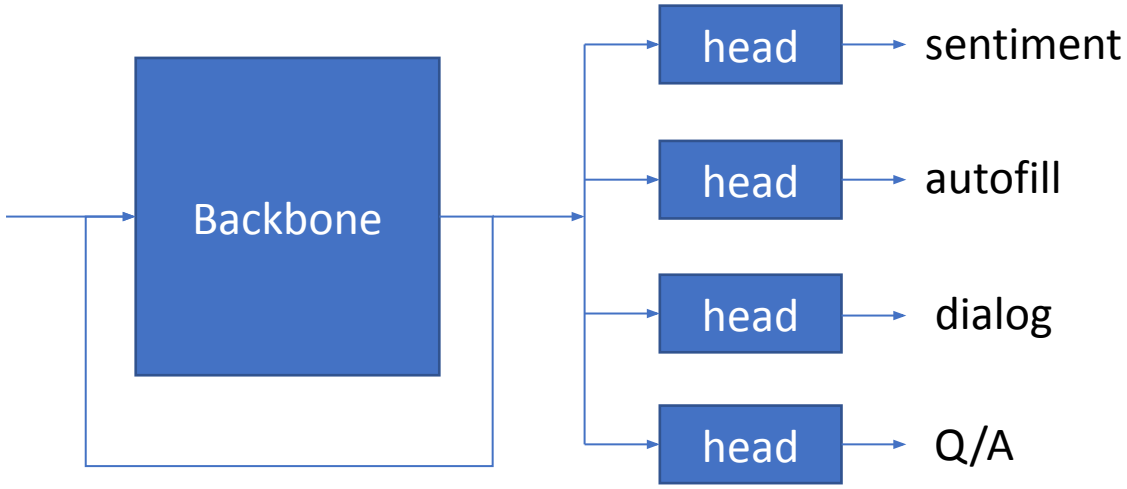
Fei Sun

March 2022

Current DNN Training Methodology

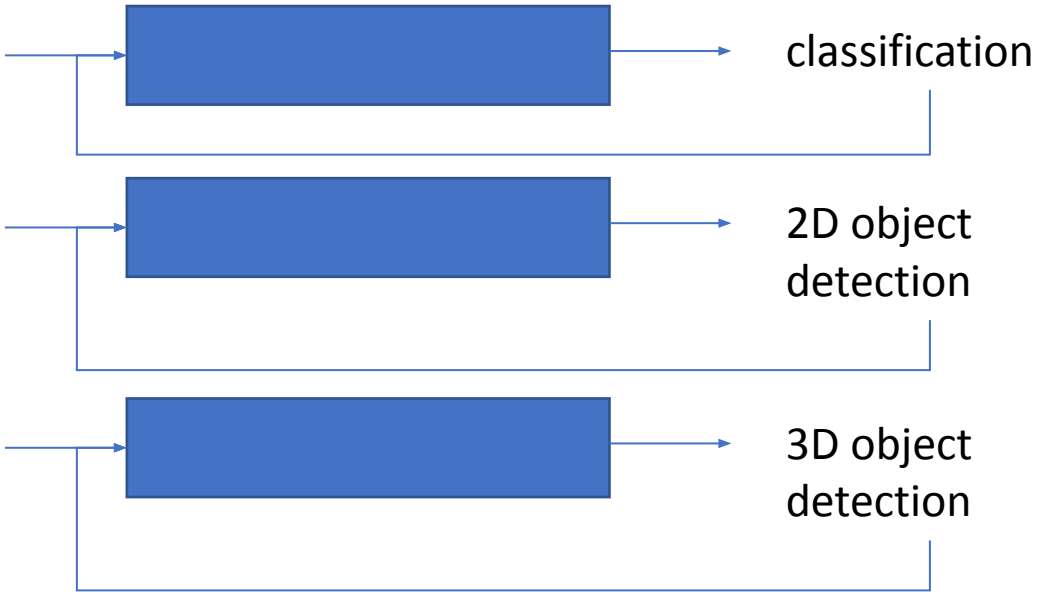
- End-to-end training large models on large dataset
  - For single objective only
- Multiple objective is done via transfer learning
  - Train a large backbone, add multiple heads to it
    - The backbone model is still very large

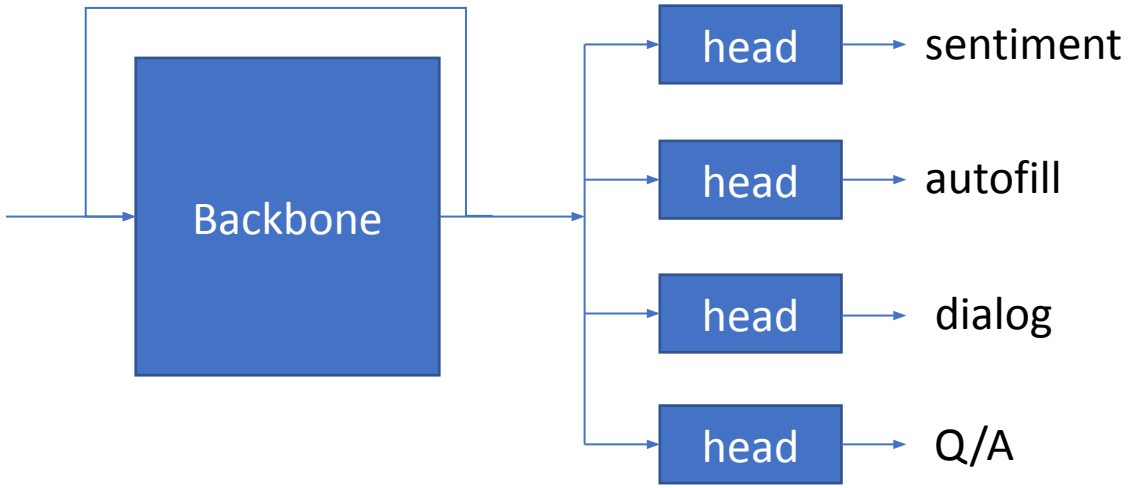




Pros and Cons of DNN Training

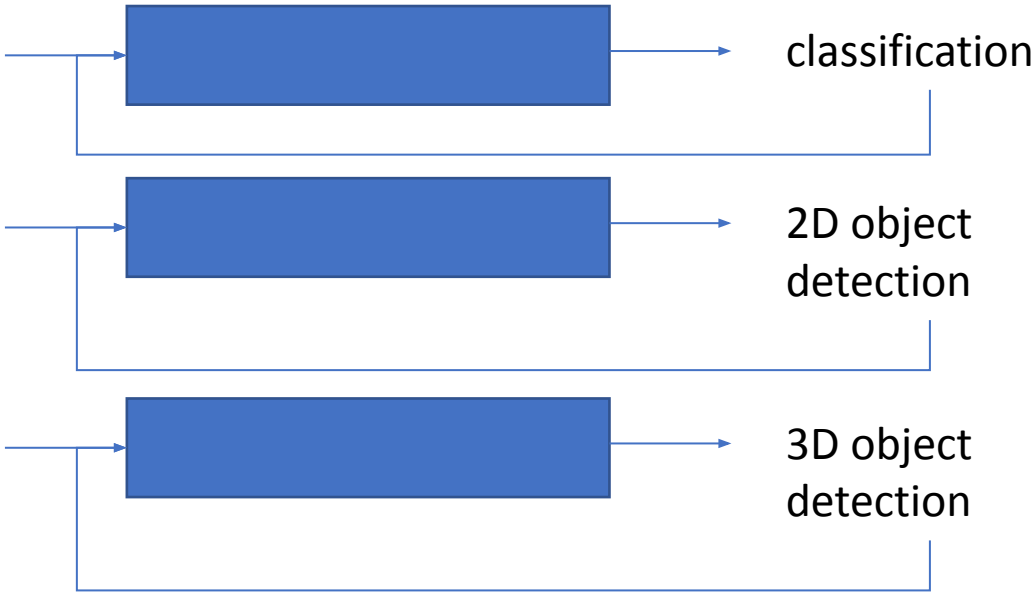
- Pros:
  - Target one application, accuracy is highest
  - Training methodology is mature
- Cons:
  - For each application, need to train a new large model
  - Data collection needs to be done for different applications differently
  - Model size is bounded due to training hardware limitations

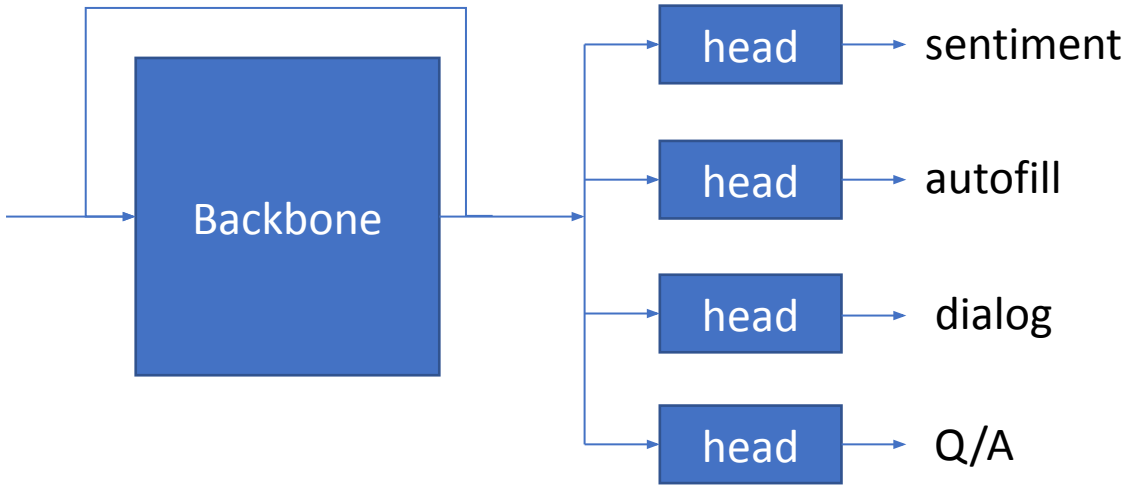




Training Cost is Extremely High

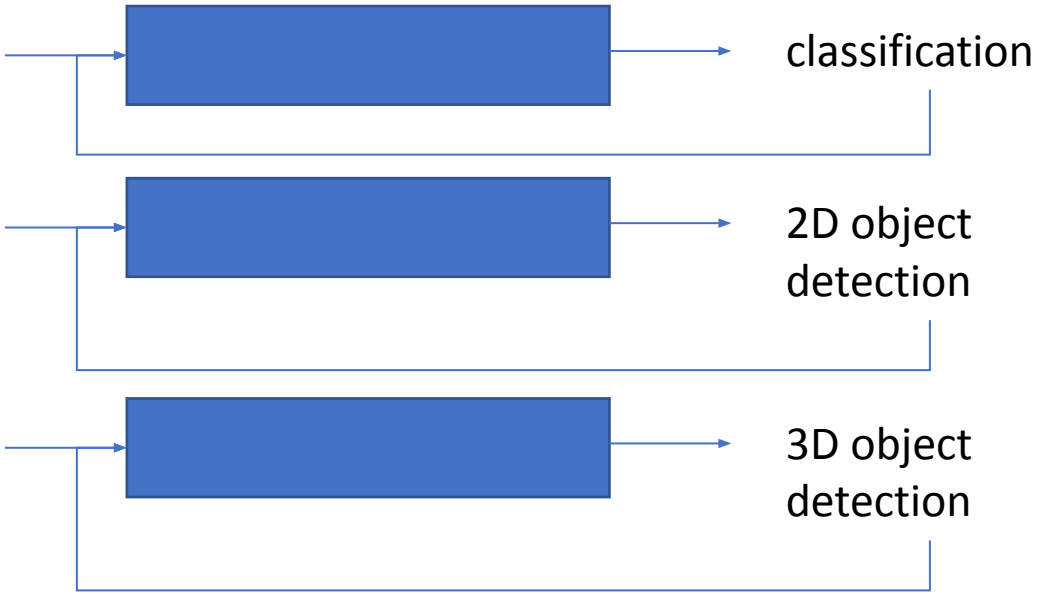
- Model is exceedingly large (training one model is expensive)
- Train one model for each application (need to train many models)
- Transfer learning using backbone cannot solve the problem
  - Backbone model is large
  - We cannot only train a few backbone models and be done.
    - “I think there is a world market for maybe five computers”

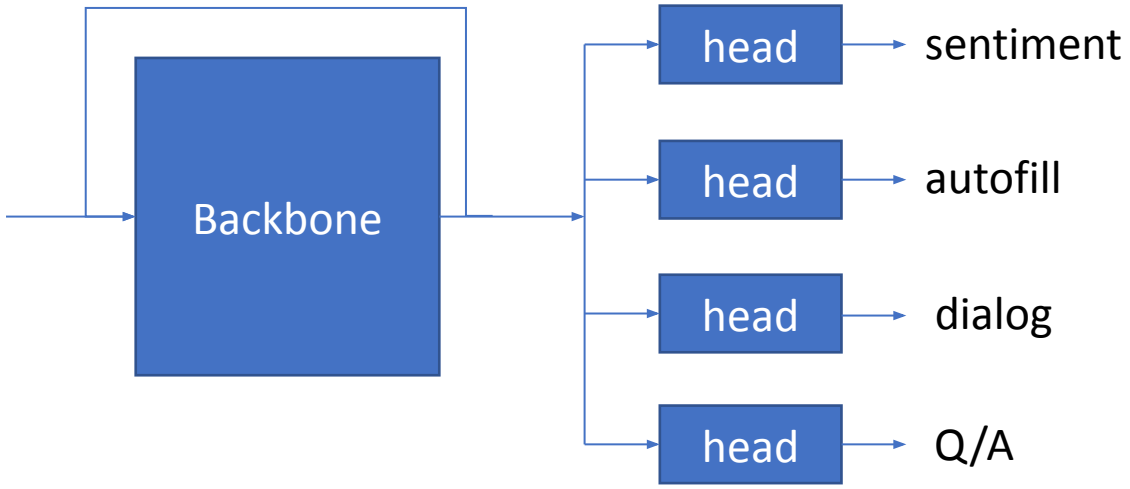




Training Data Collection is Extremely High

- Large model require lots of training data
- Different applications require different training data
- Unsupervised learning still require large amount of data
  - Reduce cost of data labeling





As a Result

- Largest models can only be trained by a few profitable companies
  - Google, Meta, OpenAI, DeepMind
- Largest production models can only be used in a few high margin areas
  - Recommendation and related
- Largest models can only be trained in areas with lots of data
  - Transfer learning can only be applied to closely related areas
- Models cannot be any larger due to hardware limits
  - End-to-end training

Question

- Are we at the tipping point?
  - Training methodologies no longer scale
  - DNN suitable applications are exceedingly narrow

- Is it time to revamp the training methodologies?
  - Reduce the model size
  - Reduce the training data size
  - Reduce the end-to-end training effort

Explore Composition in DNN Training?

- Assemble several pre-trained modules to target new applications
  - A method to facilitate reuse
- Pros
  - Module training is simpler
  - May compose models much larger than what monolithic end-to-end training enables
  - Training data requirement is much less
- Cons
  - Model for specific application is much more complicated
  - Model for specific application may achieve worse accuracy than end-to-end training at the same scale
    - May be compensated by larger models

Deep Learning vs Analog and Digital Circuits

	Deep learning	Analog circuit	Digital circuit
Value representation	Continuous	Continuous	Discrete
Complexity	???	Simple	Complex
Error due to variation	???	Propagate to entire circuit	Cut-off at clock edge
Composition	???	No	Yes

Composition Best Suited for…

- Model too complex for end-to-end training
  - But may be decomposed to independent modules
    - Each stage may be independently trained (kind of)
    - Human decomposition may not be best
- Data too little for large model training
  - Decomposed modules are smaller, requiring less training data
- Quick/cheap model training and deployment
  - Less quality requirement
  - Long tail niche application

What Research May Contribute Our Understanding towards Composition?

- Model compression
- Memory
- Neural symbolic computing
- Control flow model

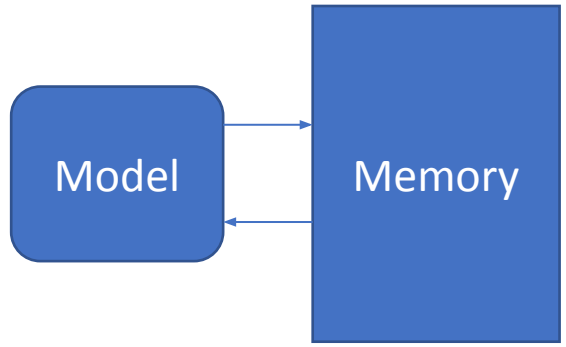
- Application: reinforcement learning

Model Compression

- Composition based model may be larger
  - Worse accuracy is compensated with model size
- Mode compression reduces the deployment cost
- Examples:
  - Quantization (well studied)
  - Weight sparsity (well studied)
  - Activation sparsity (more on this later)
  - Novel model architectures
    - Different from conventional models based on MM and CONV using large batch size
      - E.g. deep equilibrium models

Memory

- Prior knowledge is saved in memory
- Dual purposes:
  - Enhanced RNN
    - Break unrolled timestep limit
      - Train on short sequence and inference on long sequence (with no accuracy loss)
    - More complex logic between time steps
      - More on this later
  - Decouple training knowledge storage and methods extraction
    - A special case of composition
    - Model size may be significantly smaller
      - Model may be reused more



Neural Symbolic Computing

- Examples of decomposition
  - Special case: Neural module network
- Decomposition is achieved during end-to-end training
  - No prior knowledge is applied
- Boundary specification
  - Symbols?
  - No error propagation

Control Flow Model

- MOE is a simple example
  - If…else…
- More complex gating network (more than MLP)
- Generalization of activation sparsity
- With feedback loop
  - RNN
- With memory

Which Areas are You Interested in Collaborating?

- I don’t know the aforementioned areas much, but I’d like to explore more
  - They are not hot topics right now, but they are getting much hotter in recent years
  - Many challenges remain and may not be easy to solve
    - We need to understand these challenges first
- We may find some small topics along the line and explore
- Other topic suggestions welcome

- Collaboration approach: Open domain collaboration
  - For the purpose of jointly publishing papers
  - Neither party submit patents