Computer Arithmetic in Deep Learning

Bryan Catanzaro
@ctnzr
What do we want AI to do?

- Drive us to work
- Serve drinks?
- Help us communicate
- Help us find things
- Keep us organized
- Guide us to content

Scientists See Promise in Deep-Learning Programs

@ctnzr
OCR-based Translation App
Baidu IDL

你好

你好

你好

hello
AskADoctor can assess 520 different diseases, representing ~90 percent of the most common medical problems.
A yellow bus driving down a road with green trees and green grass in the background.

Living room with white couch and blue carpeting. Room in apartment gets some afternoon sun.
Sample questions and answers
Natural User Interfaces

• Goal: Make interacting with computers as natural as interacting with humans

• AI problems:
  – Speech recognition
  – Emotional recognition
  – Semantic understanding
  – Dialog systems
  – Speech synthesis
Demo

- Deep Speech public API
Computer vision: Find coffee mug
Computer vision: Find coffee mug
Why is computer vision hard?
Artificial Neural Networks

Neurons in the brain      Deep Learning: Neural network

Andrew Ng
Computer vision: Find coffee mug

Andrew Ng
Supervised learning (learning from tagged data)

X → Y

Input: Image
Output tag: Yes/No (Is it a coffee mug?)

Data:
- Image: Yes
- Image: No

Learning X → Y mappings is hugely useful
Machine learning in practice

- Progress bound by latency of hypothesis testing

Think really hard... Hack up in Matlab

Run on workstation
Deep Neural Net

• A very simple universal approximator

\[ y_j = f \left( \sum_i w_{ij} x_i \right) \]

One layer

\[ f(x) = \begin{cases} 
0, & x < 0 \\
x, & x \geq 0 
\end{cases} \]

nonlinearity
Why Deep Learning?

1. Scale Matters
   - Bigger models usually win

2. Data Matters
   - More data means less cleverness necessary

3. Productivity Matters
   - Teams with better tools can try out more ideas
Training Deep Neural Networks

\[ y_j = f \left( \sum_i w_{ij} x_i \right) \]

- Computation dominated by dot products
- Multiple inputs, multiple outputs, batch means GEMM
  - Compute bound
- Convolutional layers even more compute bound

@ctnzr
Computational Characteristics

• High arithmetic intensity
  – Arithmetic operations / byte of data
  – $O(\text{Exaflops}) / O(\text{Terabytes}) : 10^6$
  – Math limited
  – **Arithmetic matters**

• Medium size datasets
  – Generally fit on 1 node

Training 1 model: ~20 Exaflops
Speech Recognition: Traditional ASR

- Getting higher performance is hard
- Improve each stage by engineering

![Graph showing the relationship between data + model size and accuracy. The graph indicates that as data + model size increases, accuracy improves, but at a diminishing rate. An arrow labeled 'Expert engineering.' points upwards, suggesting that expert engineering can help achieve higher accuracy.]

@ctnzr
Speech recognition: Traditional ASR

• Huge investment in features for speech!
  – Decades of work to get very small improvements
Speech Recognition 2: Deep Learning!

• Since 2011, deep learning for features

```
"The quick brown fox jumps over the lazy dog."
```
Speech Recognition 2: Deep Learning!

- With more data, DL acoustic models perform better than traditional models.
Speech Recognition 3: “Deep Speech”

- End-to-end learning

“The quick brown fox jumps over the lazy dog.”
Speech Recognition 3: “Deep Speech”

- We believe end-to-end DL works better when we have big models and lots of data.

![Graph showing the comparison of Traditional ASR, DL V1 for Speech, and Deep Speech in terms of accuracy vs. data + model size.](chart.png)
End-to-end speech with DL

• Deep neural network predicts characters directly from audio
Recurrent Network

- RNNs model temporal dependence
- Various flavors used in many applications
  - LSTM, GRU, Bidirectional, ...
  - Especially sequential data (time series, text, etc.)
- Sequential dependence complicates parallelism
- Feedback complicates arithmetic
Connectionist Temporal Classification (a cost function for end-to-end learning)

\[ P(\_TH\_E\_C\_AAA\_TT\_\_\_\_\_) \]

\[ P(\_T\_H\_EE\_C\_AA\_T\_\_\_\_\_) \]

- We compute this in log space
- Probabilities are tiny
Training sets

- Train on 45k hours (~5 years) of data
  - Still growing

- Languages
  - English
  - Mandarin

- End-to-end deep learning is key to assembling large datasets

![Graph showing WER vs. Hours of Audio]

@ctnzr
Performance for RNN training

- 55% of GPU FMA peak using a single GPU
- ~48% of peak using 8 GPUs in one node
- This scalability key to large models & large datasets
Computer Arithmetic for training

- Standard practice: FP32
- But big efficiency gains from smaller arithmetic
  - e.g. NVIDIA GP100 has 21 Tflops 16-bit FP, but 10.5 Tflops 32-bit FP
- Expect continued push to lower precision
- Some people report success in very low precision training
  - Down to 1 bit!
  - Quite dependent on problem/dataset
Training: Stochastic Gradient Descent

\[ w' = w - \frac{\gamma}{n} \sum_i \nabla_w Q(x_i, w) \]

• Simple algorithm
  – Add momentum to power through local minima
  – Compute gradient by backpropagation

• Operates on minibatches
  – This makes it a GEMM problem instead of GEMV

• Choose minibatches stochastically
  – Important to avoid memorizing training order

• Difficult to parallelize
  – Prefers lots of small steps
  – Increasing minibatch size not always helpful
Training: Learning rate

\[ w' = w - \frac{\gamma}{n} \sum_{i} \nabla_w Q(x_i, w) \]

- \( \gamma \) is very small (1e-4)
- We learn by making many very small updates to the parameters
- Terms in this equation often very lopsided

Computer Arithmetic Problem
Cartoon optimization problem

\[ Q = -(w - 3)^2 + 3 \]

\[ \frac{\partial Q}{\partial w} = -2(w - 3) \]

\[ \gamma = .01 \]
Cartoon Optimization Problem

\[ \frac{\partial Q}{\partial w} \]

\[ \gamma \frac{\partial Q}{\partial w} \]
Rounding is not our friend

\[ \frac{w}{\gamma \frac{\partial Q}{\partial w}} \]

Resolution of FP16

[Erich Elsen]
Solution 1
Stochastic Rounding [S. Gupta et al., 2015]

• Round up or down with probability related to the distance to the neighboring grid points

\[
x = 100, \ y = 0.1, \ \epsilon = 1
\]

\[
x + y = \begin{cases} 
100 & \text{w.p.} \ 0.99 \\
101 & \text{w.p.} \ 0.01 
\end{cases}
\]

• Efficient to implement
  – Just need a bunch of random numbers
  – And an FMA instruction with round-to-nearest-even
Stochastic Rounding

- After adding .01, 100 times to 100
  - With r2ne we will still have 100
  - With stochastic rounding we will expect to have 101

- Allows us to make optimization progress even when the updates are small
Solution 2
High precision accumulation

- Keep two copies of the weights
  - One in high precision (fp32)
  - One in low precision (fp16)
- Accumulate updates to the high precision copy
- Round the high precision copy to low precision and perform computations
High precision accumulation

• After adding .01, 100 times to 100
  – We will have exactly 101 in the high precision weights, which will round to 101 in the low precision weights

• Allows for accurate accumulation while maintaining the benefits of fp16 computation

• Requires more weight storage, but weights are usually a small part of the memory footprint
Deep Speech Training Results

- FP16 storage
- FP32 math

[Graph showing CTC cost vs Epoch number with different lines for baseline_18353_train, baseline_18353_dev, float16_highprecaccum_18913_train, and float16_highprecaccum_18913_dev]
Deployment

• Once a model is trained, we need to deploy it
• Technically a different problem
  – No more SGD
  – Just forward-propagation
• Arithmetic can be even smaller for deployment
  – We currently use FP16
  – 8-bit fixed point can work with small accuracy loss
    • Need to choose scale factors for each layer
  – Higher precision accumulation very helpful
    • Although all of this is ad hoc
Magnitude distributions

- "Peaked" power law distributions
Determinism

- Determinism very important
- So much randomness, hard to tell if you have a bug
- Networks train despite bugs, although accuracy impaired
- Reproducibility is important
  - For the usual scientific reasons
  - Progress not possible without reproducibility

- We use synchronous SGD
Conclusion

• Deep Learning is solving many hard problems

• Many interesting computer arithmetic issues in Deep Learning

• The DL community could use your help understanding them!
  – Pick the right format
  – Mix formats
  – Better arithmetic hardware
Thanks

- Andrew Ng, Adam Coates, Awni Hannun, Patrick LeGresley, Erich Elsen, Greg Diamos, Chris Fougner, Mohammed Shoeybi
  ... and all of SVAIL

Bryan Catanzaro
@ctnzs