December 11, 2020 Poster Session

GE Energy Efficient Machine Learning at the Edge

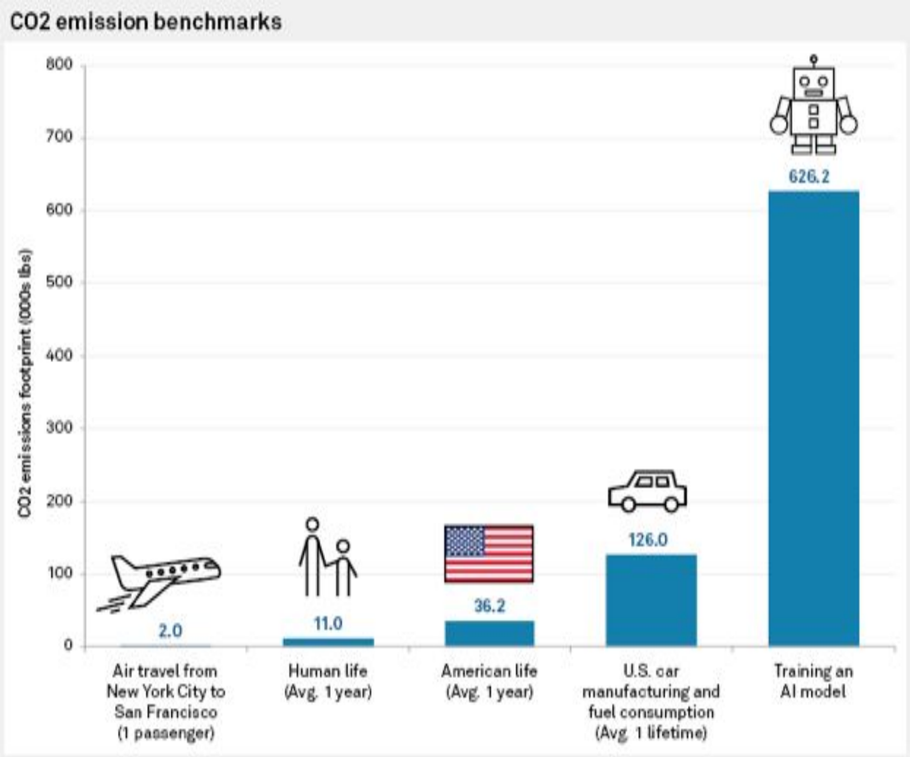
Ji In Choi, Madeleine Georges, Julia Shin,
Olivia Wang, Tiffany Zhu
Mentor: Tapan Shah

Overview

1. Introduction
2. Base Quantization Methods
3. Stochastic Quantization Methods
4. Next Steps

Introduction

- How can we make AI carbon efficient?
- Low Precision ML
    - Can we learn a model from training data with 1-8 bit precision (as compared to 32-64 bits)?



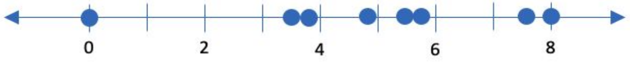
Data compiled Oct. 9, 2019.
An "American life" has a larger carbon footprint than a "Human life" because the U.S. is widely regarded as one of the top carbon dioxide emitters in the world.
Source: College of Information and Computer Sciences at University of Massachusetts Amherst

Quantization

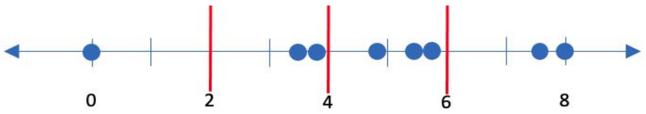
What is Quantization?

Convert data from *q*-bits to *p*-bits where $q > p$

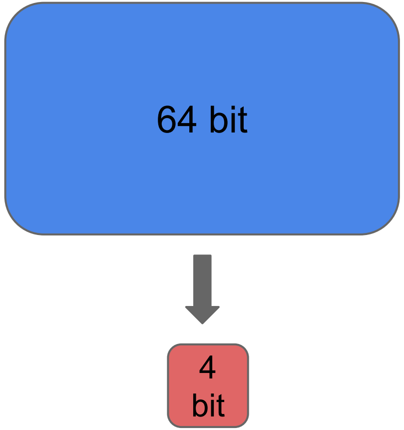
[0, 3.5, 3.8, 4.8, 5.5, 5.6, 7.5, 8]



[0, 2, 2, 4, 4, 4, 6, 6]



Why Quantize?



Reduce storage by **20** times

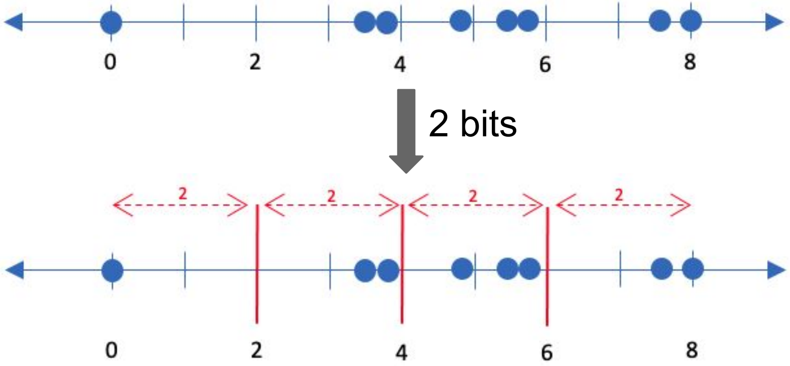
Base Quantization Methods

- Simple Quantizer
- Simple Quantizer by Column
- Quantile Quantizer
- Quantile Quantizer by Column

Base Quantization Methods

Simple Quantizer

- Use *minimum* and *maximum* values in the entire dataset
- Create 2^p bins with *uniform* range (i.e. equal bin widths)
- E.g.



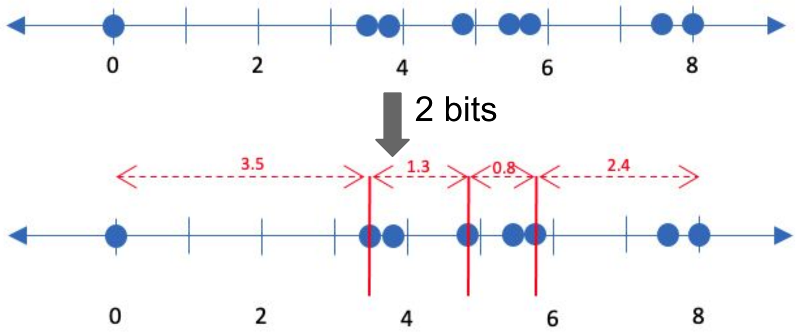
Simple Quantizer By Column

- Same as Simple Quantizer but for each column

Base Quantization Methods

Quantile Quantizer

- Use different quantiles over the entire dataset
- Create 2^p bins with *unequal* range (i.e. unequal bin widths)
- May help account for distribution of dataset
- E.g.

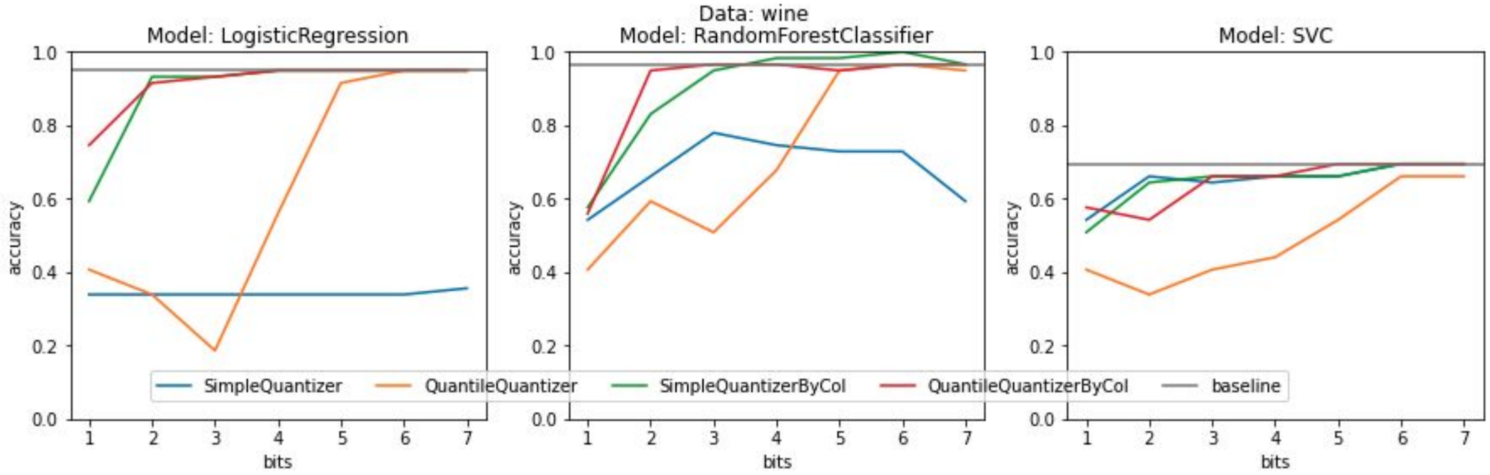


Quantile Quantizer by Column

- Same as Quantile Quantizer but for each column

Base Quantization Methods - Results

- Ran the 4 base quantizers on datasets from OpenML
  - QuantileQuantizerByCol may be the best quantizer



Base Quantization Methods - Results

- Using paired t-test

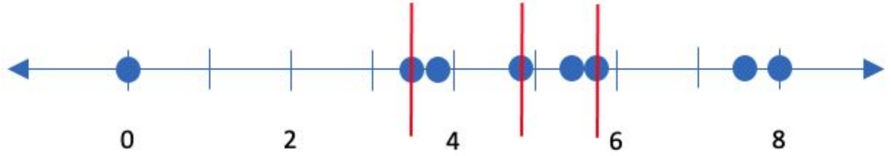
$$H_0 : a_q - a_0 = 0$$
$$H_1 : a_q - a_0 \neq 0$$

- Final results:
  1. By **column** quantizers work better
  2. **Quantile** quantizers work better than the Simple quantizers
  3. **QuantileQuantizerByColumn** is best of the base quantizers

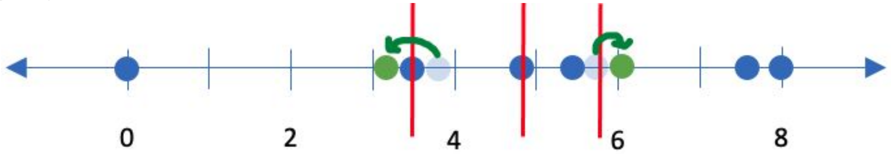
Bits	QuantileQuantizerByColumn vs Simple Quantizer	
	t-Statistic	p-value
1	5.9405	2.1239e-07
2	6.1571	9.5375e-08
3	5.5912	7.6428e-07
4	4.2122	9.6499e-05
5	4.2644	8.1072e-05
6	2.5539	0.0135
7	3.1717	0.0024
8	2.4289	0.0184

Stochastic Quantization Methods

- The process of using a stochastic method is:
  - 1. Given dataset X, create bins using a deterministic base quantizer
    - We use QuantileQuantizerByColumn in our experiments



  - 2. Apply a stochastic process to X to put each data point into the bins generated from step 1.



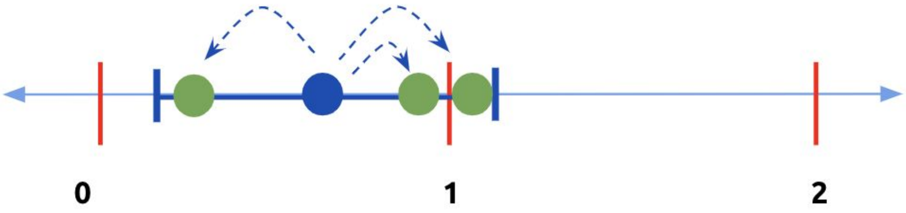
- Goal: lower the quantization error by introducing randomness
- Two methods considered: Dithering and Stochastic Rounding

Stochastic Quantization Methods

Dithering

- Random noise is added to each input value
- Assuming data is scaled then for each data point X_i:

$$X_i \leftarrow X_i + noise$$
$$noise \sim Uniform(-\,bin\_width/2,\ bin\_width/2)$$

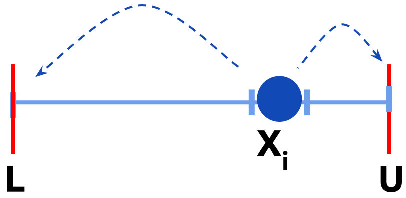


Stochastic Quantization Methods

Stochastic Rounding

- The input value is rounded to one of bordering quantization levels with probability dependent on proximity
- Given a point X_i and its neighboring upper bin U and lower bin L:

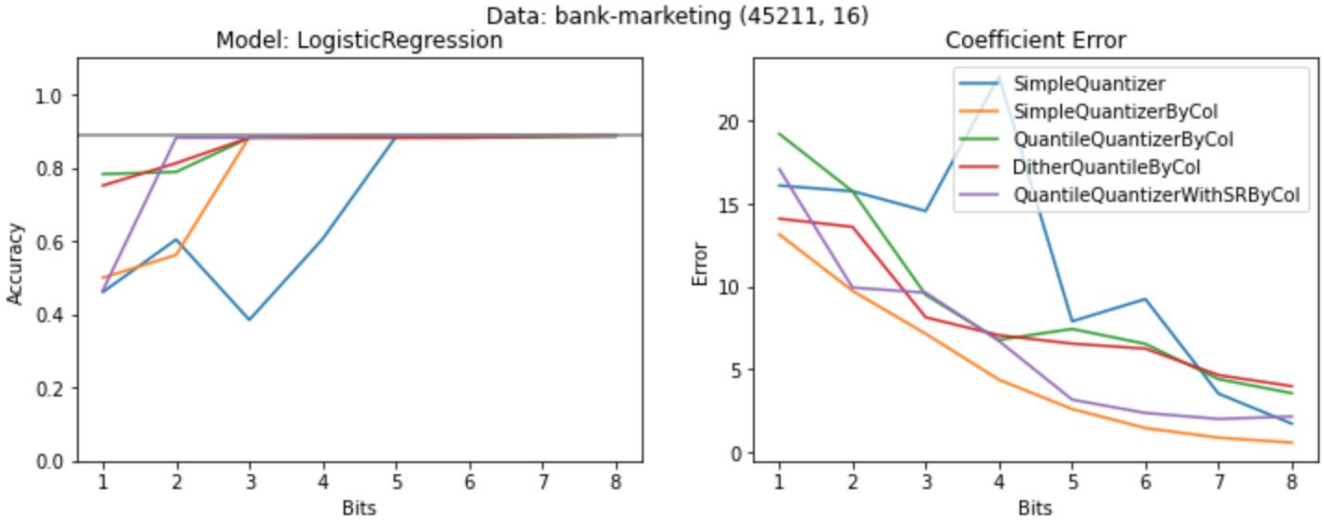
$$round(X_i) = \begin{cases} U & with\ probability\ \frac{(X_i - L)}{(U - L)} \\ L & with\ probability\ \frac{(U - X_i)}{(U - L)} \end{cases}$$



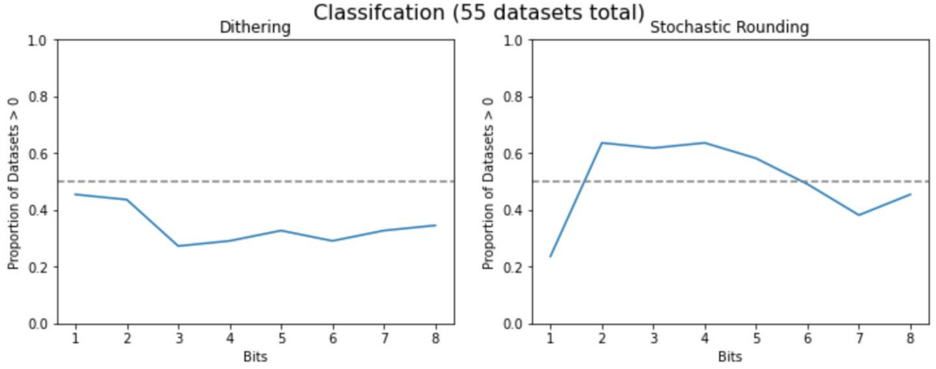
Stochastic Quantization Method - Results

Classification Data

- 55 classification datasets from OpenML
- Logistic Regression

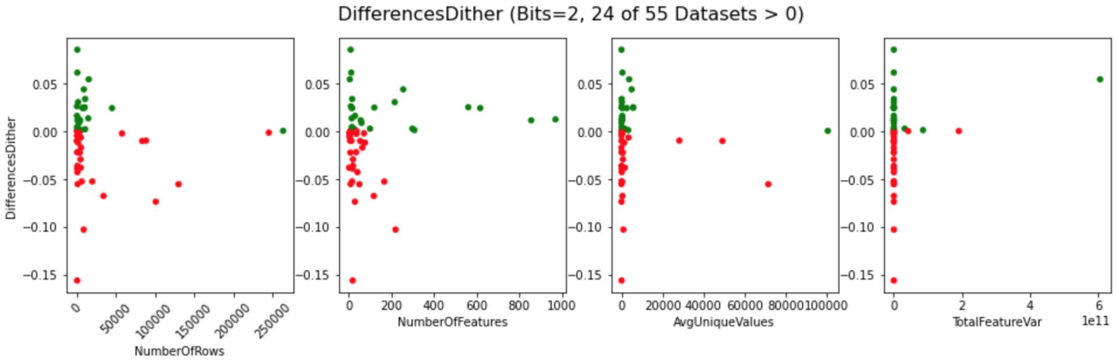


Stochastic Quantization Method - Results



Classification Data

- Dithering is better for 1 bit
- Stochastic Rounding is better overall

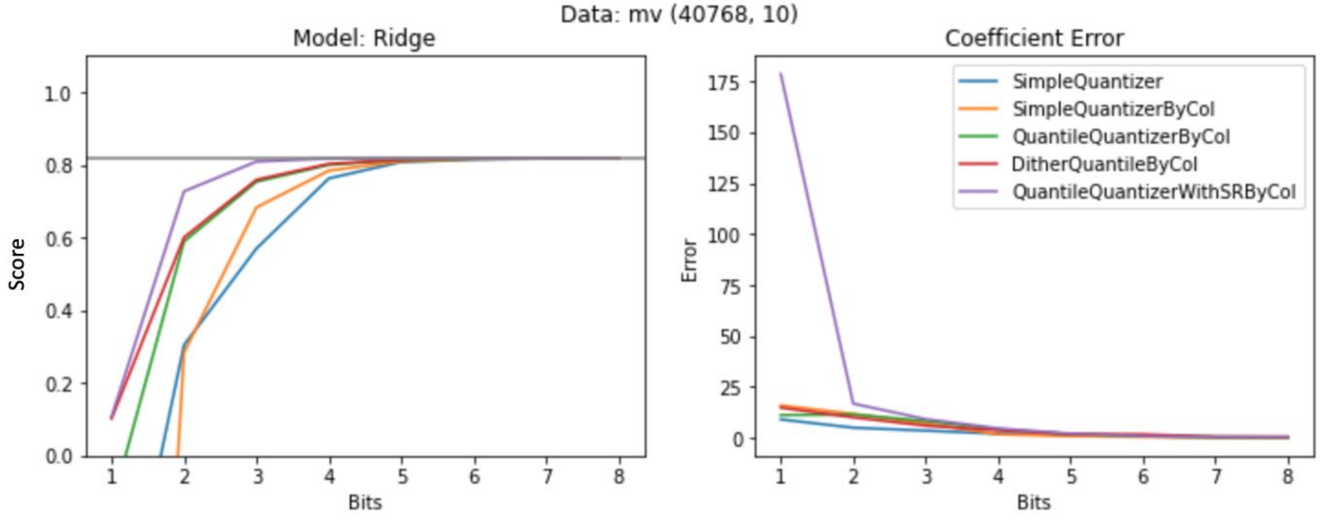


- No clear pattern between improvement in accuracies and dataset attributes

Stochastic Quantization Method - Results

Regression Data

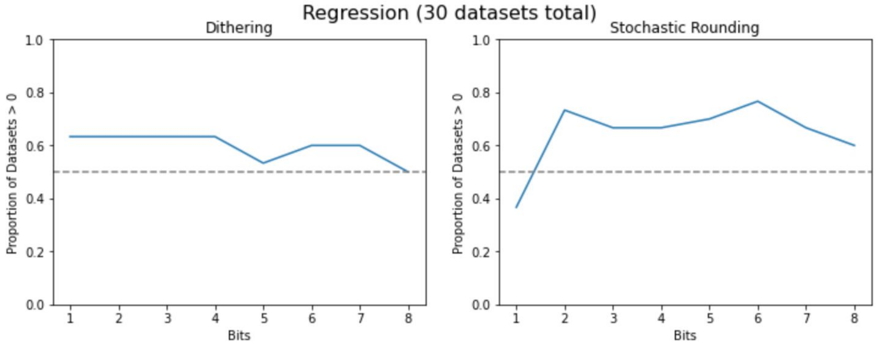
- 30 regression datasets from OpenML
- Ridge Regression



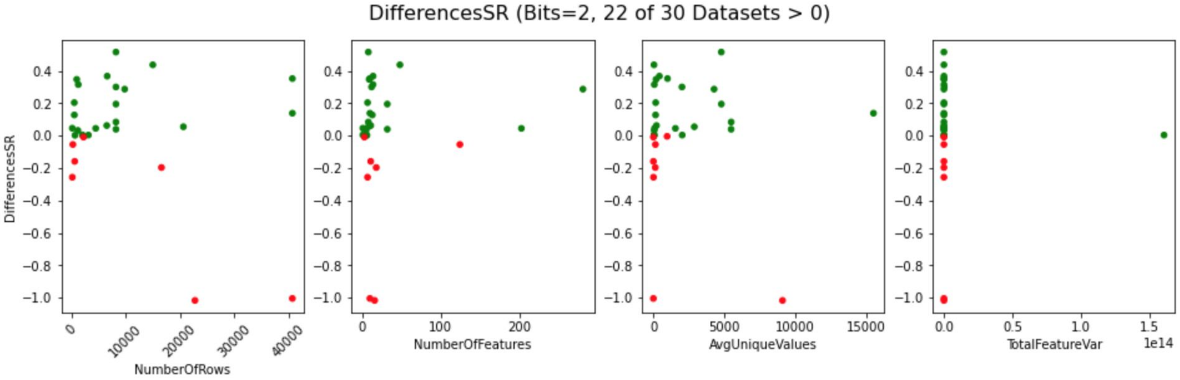
Stochastic Quantization Method - Results

Regression Data

- *Both* dithering and Stochastic Rounding work better on *regression* data



- No clear pattern between improvement in accuracies and dataset attributes



Conclusions

- **Base Quantizers**:
  - Using *quantiles* for each *column* is the best of the base methods
- **Stochastic Quantizers**:
  - Dithering and stochastic rounding can further improve quantization
  - Dithering and stochastic rounding work better on *regression* datasets than classification datasets
  - *Stochastic rounding* works better than dithering

Next Steps

Mini-batch Learning

Leveraging reinforcement learning to iteratively update quantizer

Precision Reallocation

Using linear programming to reallocate available bits to each feature

Deep Learning Models

Evaluate Quantization Methods on more complex, non-linear models

References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In ACL, 2019.
[2] Gupta, Suyog, Agrawal, Ankur, Gopalakrishnan, Kailash, and Narayanan, Pritish. Deep learning with limited numerical precision. arXiv preprint arXiv:1502.02551, 2015.
[3] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. International Conference on Machine Learning. page 4035–4043, 2017.
[4] Hao, Karen. Training a single AI model can emit as much carbon as five cars in their lifetimes. MIT Technology Review, 2019.
[5] Onkar Dabeer, and Upamanyu Madhow. Channel estimation with low-precision analog-to-digital conversion. 2010 IEEE International Conference on Communications, Cape Town, 2010, pp. 1-6, doi: 10.1109/ICC.2010.5501995.
[6] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. arXiv:1907.10597v3
[7] Ye, S., Zhang, T., Zhang, K., Li, J., Xie, J., Liang, Y., Liu, S., Lin, X., and Wang, Y. A unified framework of dnn weight pruning and weight clustering/quantization using admm. arXiv preprint arXiv:1811.01907, 2018.

Thanks!

Questions?

jic2124
mg4128
js5569
yw3324
tz2196
@columbia.edu