Introduction to Deep Learning

1. Logistics, Software, Linear Algebra

STAT 157, Spring 2019, UC Berkeley

Mu Li and Alex Smola

courses.d2l.ai/berkeley-stat-157

aws


Logistics

Goals

- **Introduction** to Deep Learning
  (basic MLP, optimization, convolutions, sequences)
- **Theory**
  - Capacity control (weight decay, dropout, batch norm)
  - Optimization, models, overfitting, objective functions
- **Practice**
  - Write code in Python / MxNet Gluon / Jupyter
  - Solve realistic problems
- **Project**
  - Ability to start original research in Deep Learning **in a team**

Getting there

- **Course**
  - Tuesday and Thursday 3:30-5:00pm in LeConte 3
  - Slides and notebooks online on website
  - YouTube channel with lectures at goo.gl/zcBh2U
  - Github repository at **d2l-ai/berkeley-stat-157**
- **Dive into Deep Learning**
  - Jupyter Notebooks
  - Github repository at **d2l-ai/d2l-en**

Contacts

- **Lecturers**
  - Mu Li and Alex Smola
    Office hours Thursdays 1:00-2:30pm in Evans Hall 337
  - Email berkeley-stat-157@googlegroups.com
- **TA**
  - Ryan Theisen
    Office hours Wednesdays, likely 2:00-4:00pm in Evans Hall
  - Email theisen@berkeley.edu
- Course discussion site discuss.mxnet.io/c/courses

Homework

- **10 assignments**
  - Due 1 week after posted
    1/29, 2/5, 2/12, 2/19, 2/26, 3/12, 4/2, 4/9, 4/16, 4/30
  - Solutions and feedback returned 1 week after submitted.
  - Solutions based on students' submissions (we will acknowledge the writer)
- **Best 9 out of 10** homeworks count
- You can be **late by 2 days once** for any reason
- 50/50 mix of programming and theory

Homework

- **Submit homework via GitHub**
  - Submit the homework by 4pm on the Tuesday it's due
  - TA performs pull request after deadline
  - Submit as PDF (theory) and Jupyter notebooks (code)
  - TA commits annotated feedback via Git - add **rythei**
- **Logistics**
  - Github account & repository (email to course)
  - Permission for TA to read/write the repository

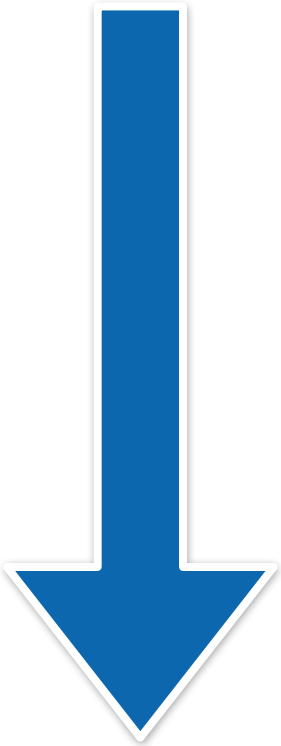
Exam

- **March 19, 2019**
- Midterm exam only
  - Open book but not open computer/phone/tablet
  - Problems will be similar to homework
- **Why**
  - At least one test that has to be done in person (you *could* fake the rest)
  - No final exam due to projects

Project

- **Original work in machine learning**
  - Existing tools applied to novel problem
  - Novel tools
  - Ask lecturers & TA for advice if you're stuck
- Research 'with training wheels' **simulates academic process**
  - Research in a team (**3-5 students**)
  - Deliverables with schedule / deadlines
  - End result is a paper/report/presentation (NIPS template)

Project



- 2/5 **Register team** (names, working title)
- 3/5 **Project proposal** (1-2 page, 5 min talk)
- 4/22-25 (or earlier) **Talk to TA** to discuss
- 5/7-9 **Final presentation** & **report**
  (6-20 pages report, 6-20 slides talk)

- Talk to TA and lecturers (it's our job to help)
- Start **early** (last minute projects fail often)
- No, you cannot do it alone. **This is teamwork.**

Have fun. Learn stuff.

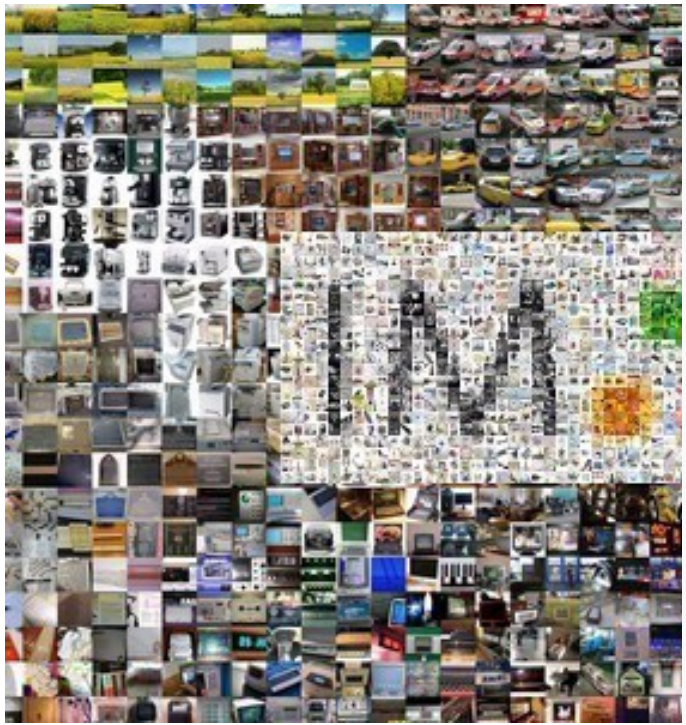

Deep Learning

Classify Images

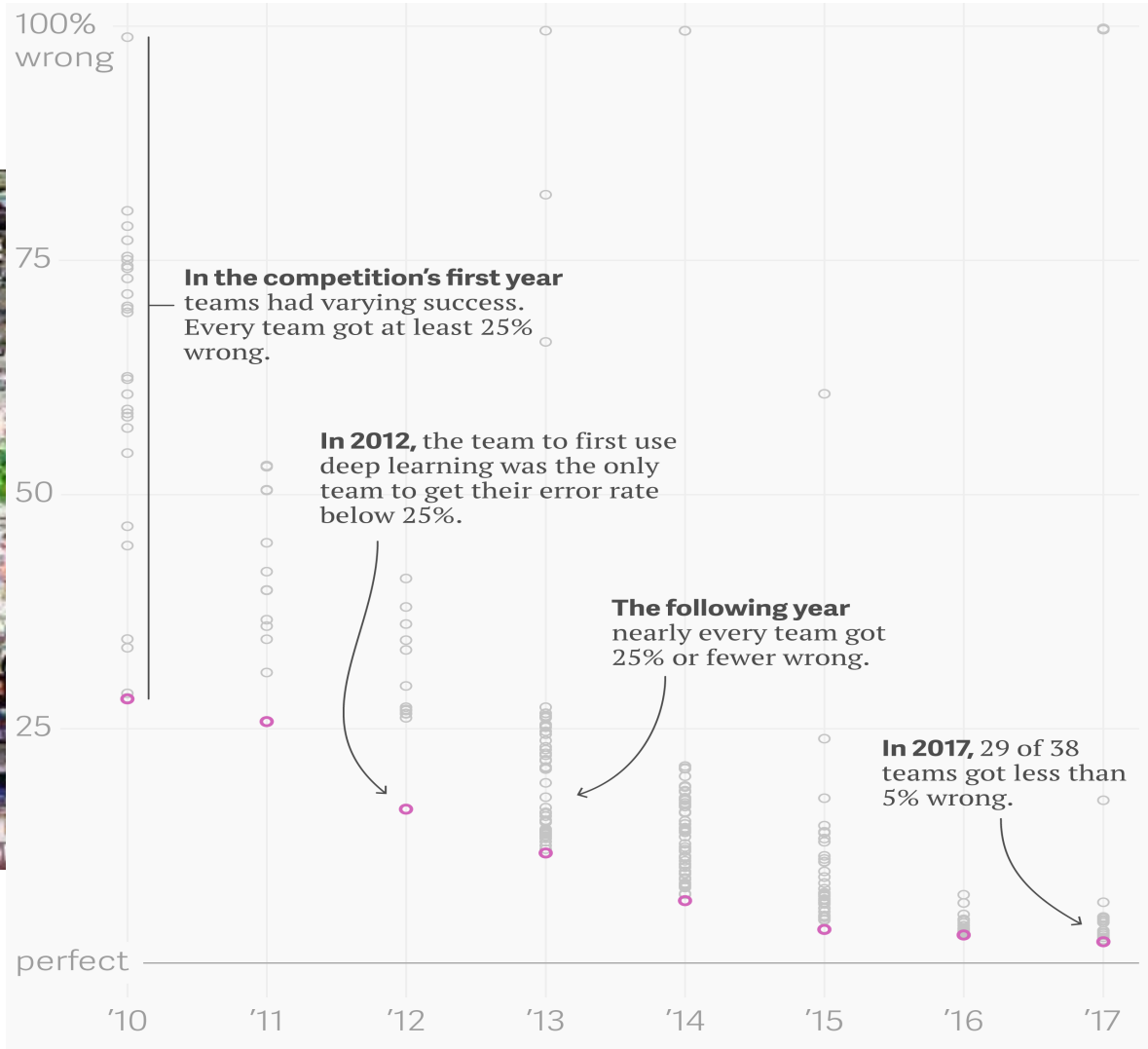
http://www.image-net.org/

aws

Classify Images



Yanofsky, Quartz
https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/



Detect and Segment Objects

courses.d2l.ai/berkeley-stat-157

https://github.com/matterport/Mask_RCNN

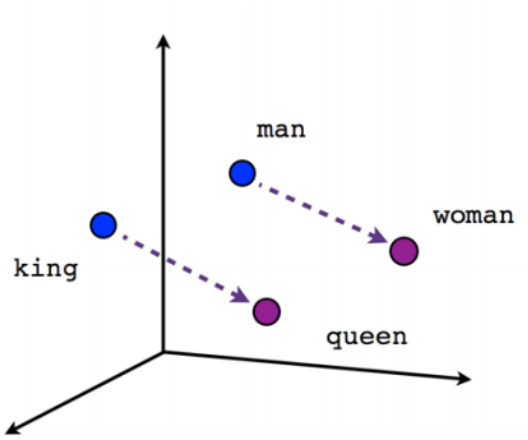
Style transfer

https://github.com/zhanghang1989/MXNet-Gluon-Style-Transfer/

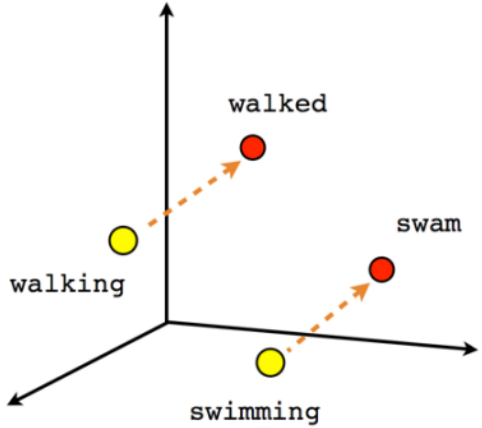
Synthesize Faces

Karras et al, ICLR 2018

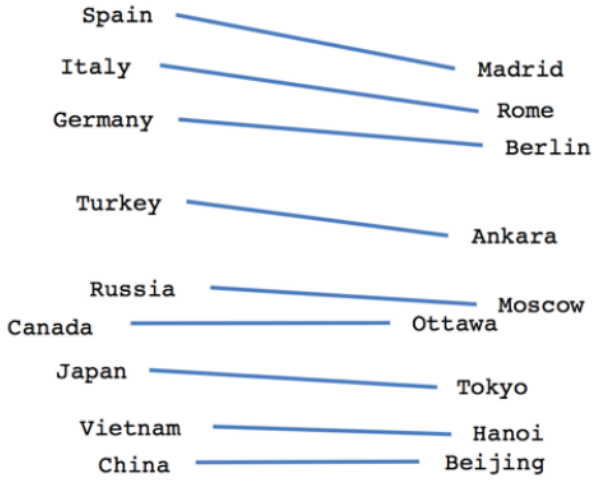
Analogies



Male-Female



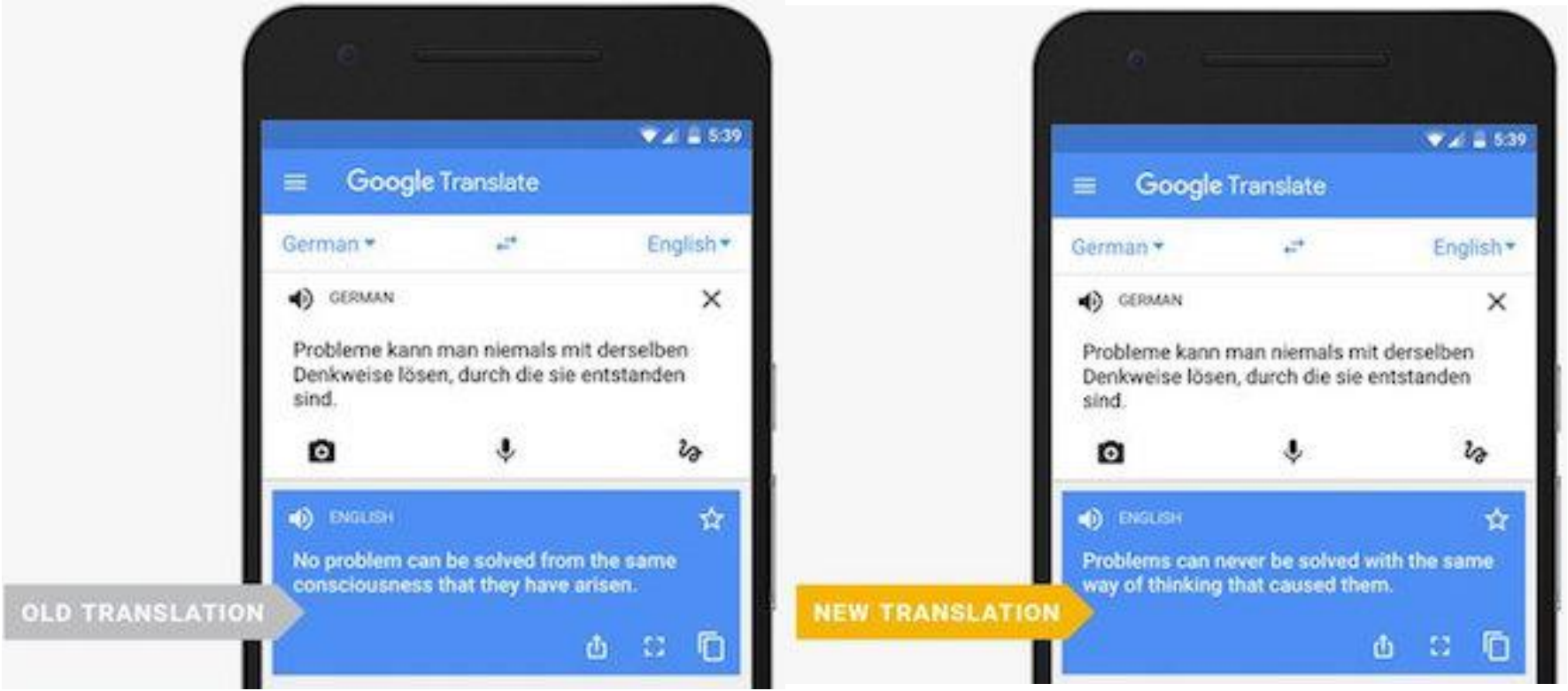
Verb tense



Country-Capital

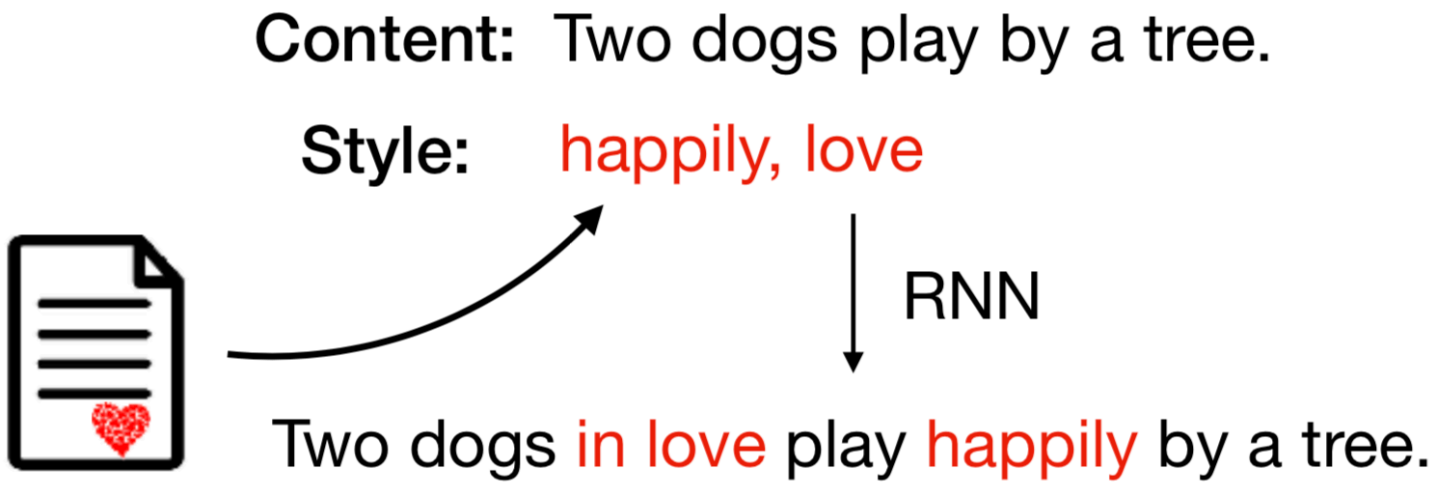
https://www.tensorflow.org/tutorials/word2vec

Machine Translation



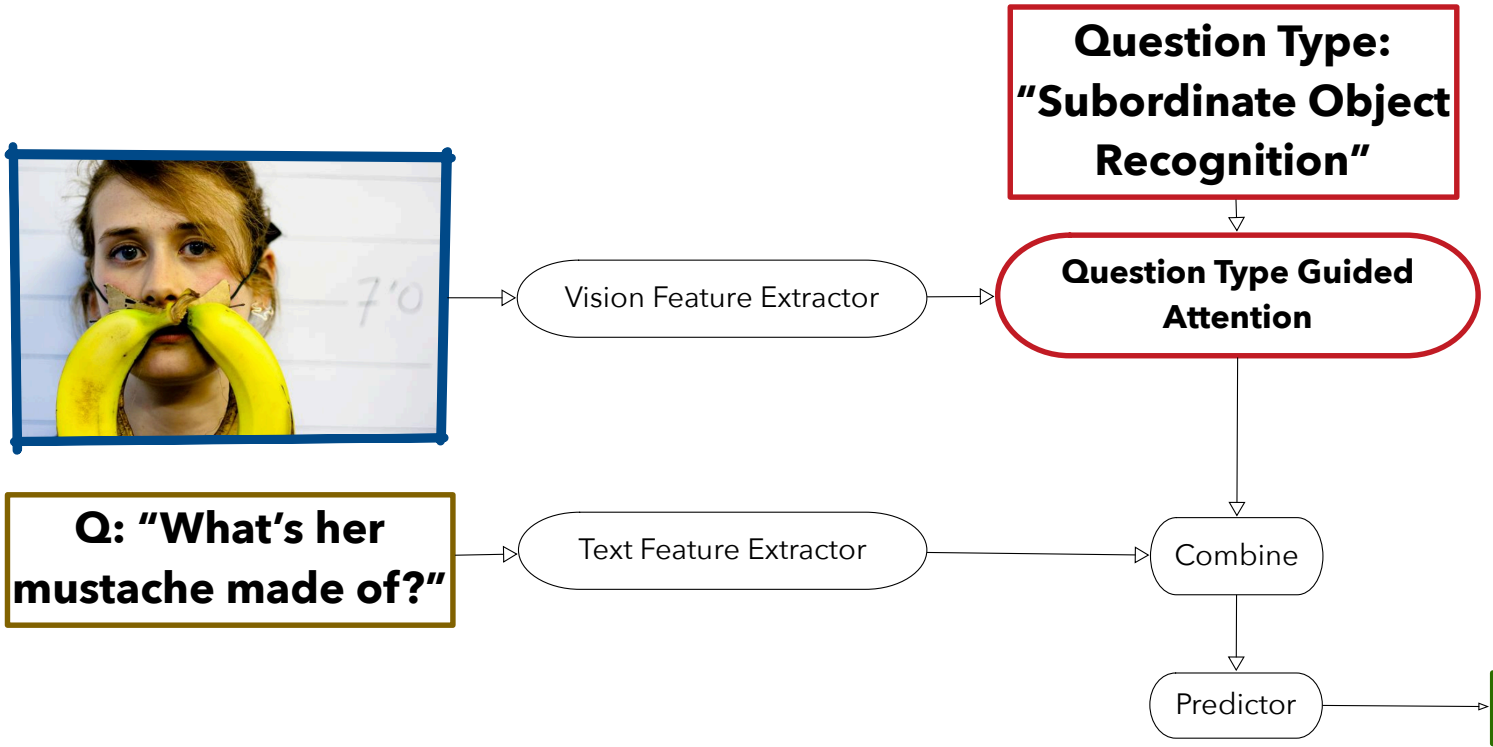
https://www.pcmag.com/news/349610/google-expands-neural-networks-for-language-translation

Text synthesis



Li et al, NACCL, 2018

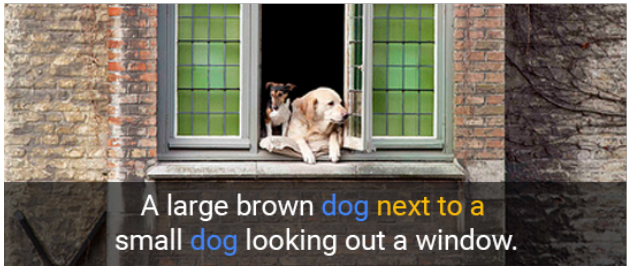
Question answering



Shi et al, 2018, Arxiv

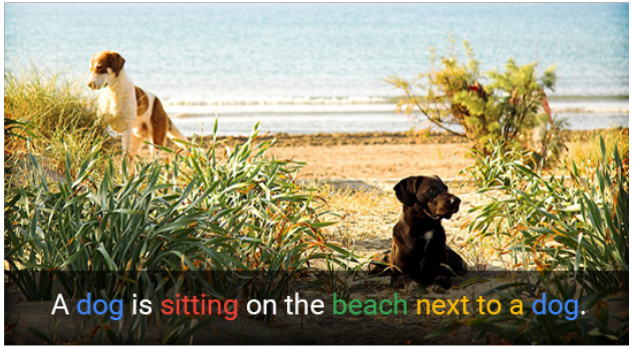
Image captioning

Human captions from the training set



courses.d2l.ai/berkeley-stat-157

Automatically captioned



Shallue et al, 2016
https://ai.googleblog.com/2016/09/show-and-tell-image-captioning-open.html

aws


Software

Tools

d2l.ai/chapter_crashcourse/install.html

- **Python**
  - Everyone is using it in machine learning & data science
  - Conda package manager (for simplicity)
- **Jupyter**
  - So much easier to keep track of your experiments
  - Obviously you should put longer code into modules
- **Reveal (for notebook slides)**
  `conda install -c conda-forge rise`
- **Apache MXNet Gluon**
  - Scalability & ease of use
  - Imperative interface

Laptop / Desktop / Generic Cloud with Linux

- **Conda**

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sh Miniconda3-latest-Linux-x86_64.sh
mkdir d2l-en
cd d2l-en
curl https://www.d2l.ai/d2l-en.zip -o d2l-en.zip
unzip d2l-en.zip
rm d2l-en.zip
```

- **Install environment**

```
conda env create -f environment.yml
cd d2l-en
source activate gluon
jupyter notebook
```

- **GPU**
  - Install NVIDIA drivers / CUDA / CUDNN / TensorRT
  - Update `environment.yml` to replace `mxnet` with `mxnet-cu92`

Laptop / Desktop with MacOS

- **Conda**

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOS-x86_64.sh
sh Miniconda3-latest-MacOS-x86_64.sh
mkdir d2l-en
cd d2l-en
curl https://www.d2l.ai/d2l-en.zip -o d2l-en.zip
unzip d2l-en.zip
rm d2l-en.zip
```

- **Install environment**

```
conda env create -f environment.yml
cd d2l-en
source activate gluon
jupyter notebook
```

- **GPU**
  - Install NVIDIA drivers / CUDA / CUDNN / TensorRT
  - Update `environment.yml` to replace `mxnet` with `mxnet-cu92`

Laptop / Desktop with Windows

- **Conda**

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe
```

Click on `Miniconda3-latest-Windows-x86_64.exe`

```
mkdir d2l-en
cd d2l-en
curl https://www.d2l.ai/d2l-en.zip -o d2l-en.zip
unzip d2l-en.zip
rm d2l-en.zip
```

- **Install environment**

```
conda env create -f environment.yml
cd d2l-en
activate gluon
jupyter notebook
```

- **GPU**
  - Install NVIDIA drivers / CUDA / CUDNN / TensorRT
  - Update `environment.yml` to replace `mxnet` with `mxnet-cu92`

AWS (via Deep Learning AMI)

- Install Base AMI (Ubuntu with NVIDIA drivers)
  - aws.amazon.com/machine-learning/amis/
  - aws.amazon.com/marketplace/pp/B077GFM7L7
- Follow steps for basic MXNet / Jupyter install as before
  - For P2 / P3 / G2 / G3 instances NVIDIA drivers default
  - Update the Conda environment
- SSH with port forwarding
  - d2l.ai/chapter_appendix/jupyter.html
  - d2l.ai/chapter_appendix/aws.html (setting up AWS account)

Colab

- Go to colab.research.google.com
- Activate the GPU supported runtime (this is a K80 GPU)
- Install MXNet

```
!pip install mxnet-cu92
!pip install d2l
```


Linear Algebra

Scalars

- **Simple operations**

$$c = a + b$$

$$c = a \cdot b$$

$$c = \sin a$$

- **Length**

$$|a| = \begin{cases} a & \text{if} a > 0 \\ -a & \text{otherwise} \end{cases}$$

$$|a + b| \leq |a| + |b|$$

$$|a \cdot b| = |a| \cdot |b|$$

Vectors

- **Simple operations**

$$c = a + b \quad \text{where } c_i = a_i + b_i$$
$$c = \alpha \cdot b \quad \text{where } c_i = \alpha b_i$$
$$c = \sin a \quad \text{where } c_i = \sin a_i$$

Definition of a vector space

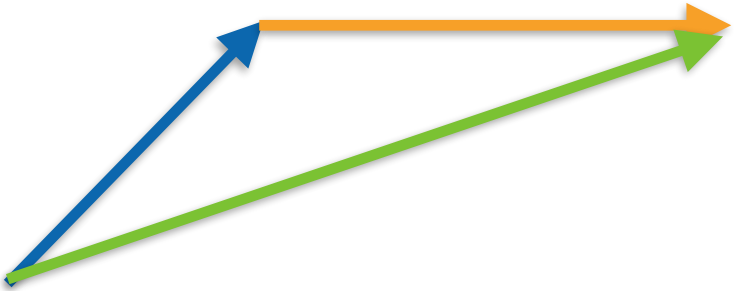
- **Length**

$$\|a\|_2 = \left[\sum_{i=1}^{m} a_i^2\right]^{\frac{1}{2}}$$

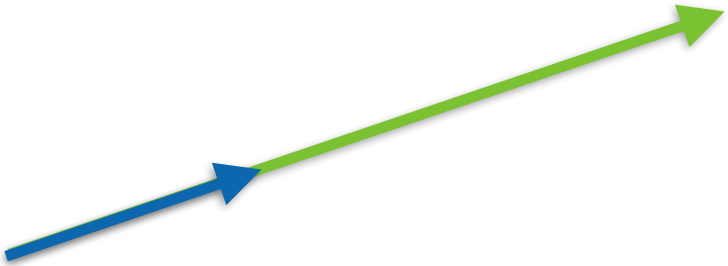
Definition of a norm

$$\|a\| \geq 0 \text{ for all } a$$
$$\|a + b\| \leq \|a\| + \|b\|$$
$$\|a \cdot b\| = |a| \cdot \|b\|$$

Vectors



$$c = a + b$$



$$c = \alpha \cdot b$$

Mathematician's **'parallel for all do'**

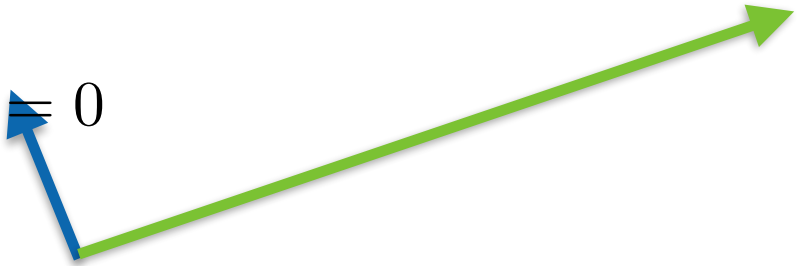
Vectors

- **Dot product**

$$a^\top b = \sum_i a_i b_i$$

- **Orthogonality**

$$a^\top b = \sum_i a_i b_i = 0$$



(e.g. if we have two vectors that are orthogonal with a third, their linear combination is it, too)

Matrices

- **Simple operations**

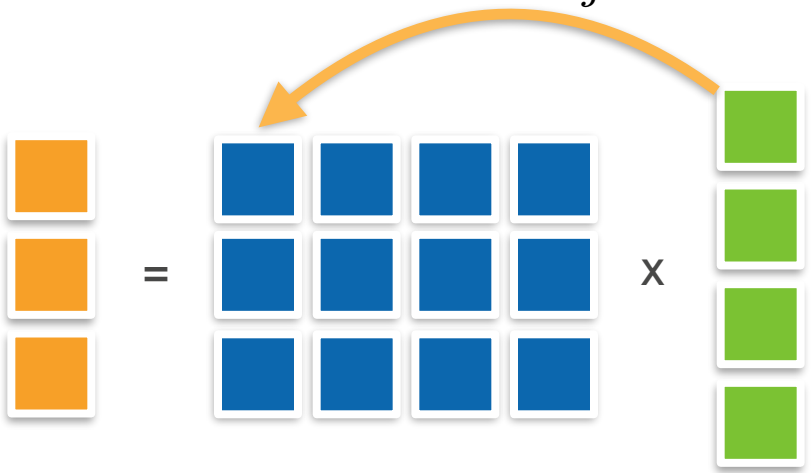
$$C = A + B \qquad \text{where } C_{ij} = A_{ij} + B_{ij}$$
$$C = \alpha \cdot B \qquad \text{where } C_{ij} = \alpha B_{ij}$$
$$C = \sin A \qquad \text{where } C_{ij} = \sin A_{ij}$$

- **Functional Analysis 101**
  vector = function, matrix = linear operator
  most theorems work sort-of in infinite dimensional spaces

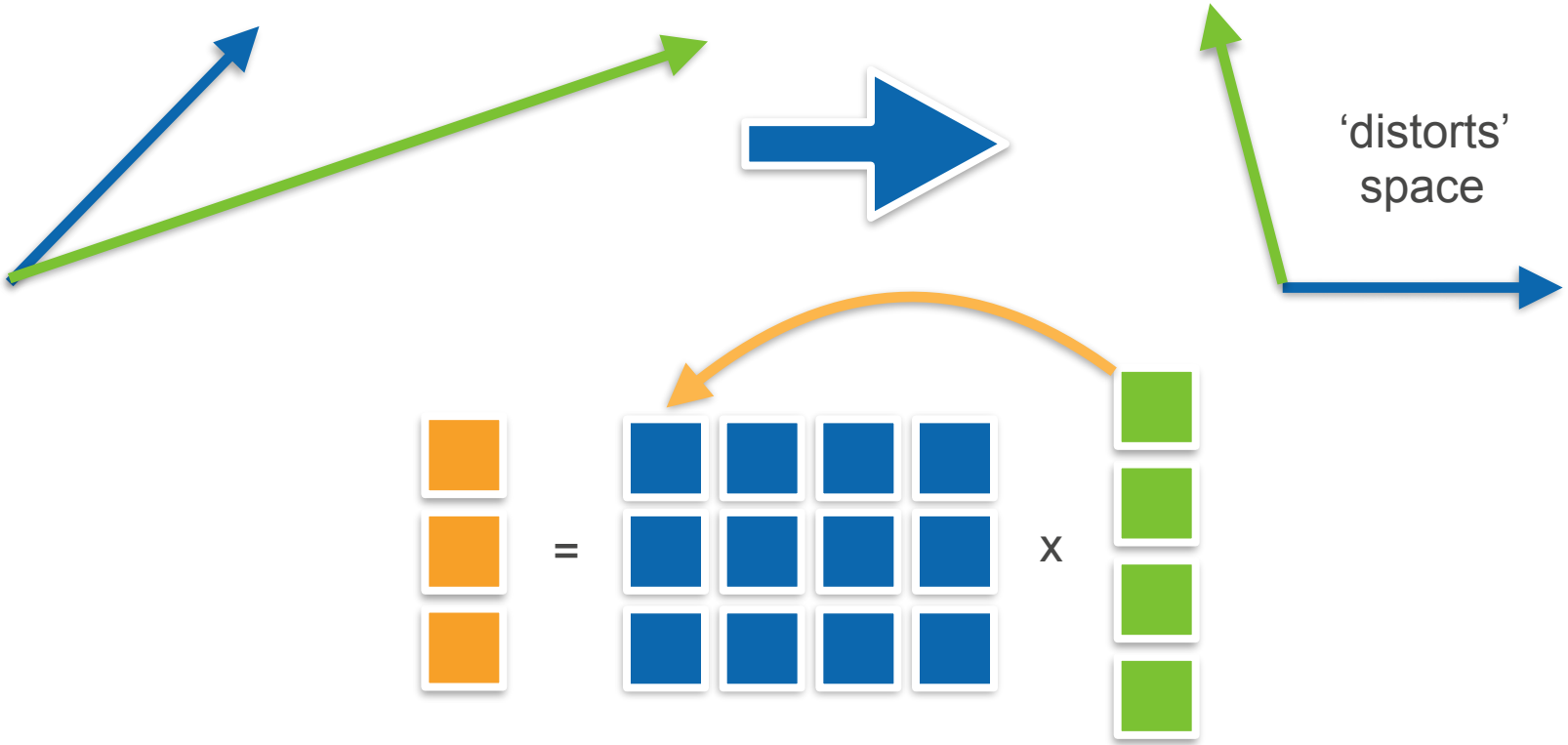
Matrices

- **Multiplications (matrix vector)**

$$c = Ab \text{ where } c_i = \sum_j A_{ij} b_j$$



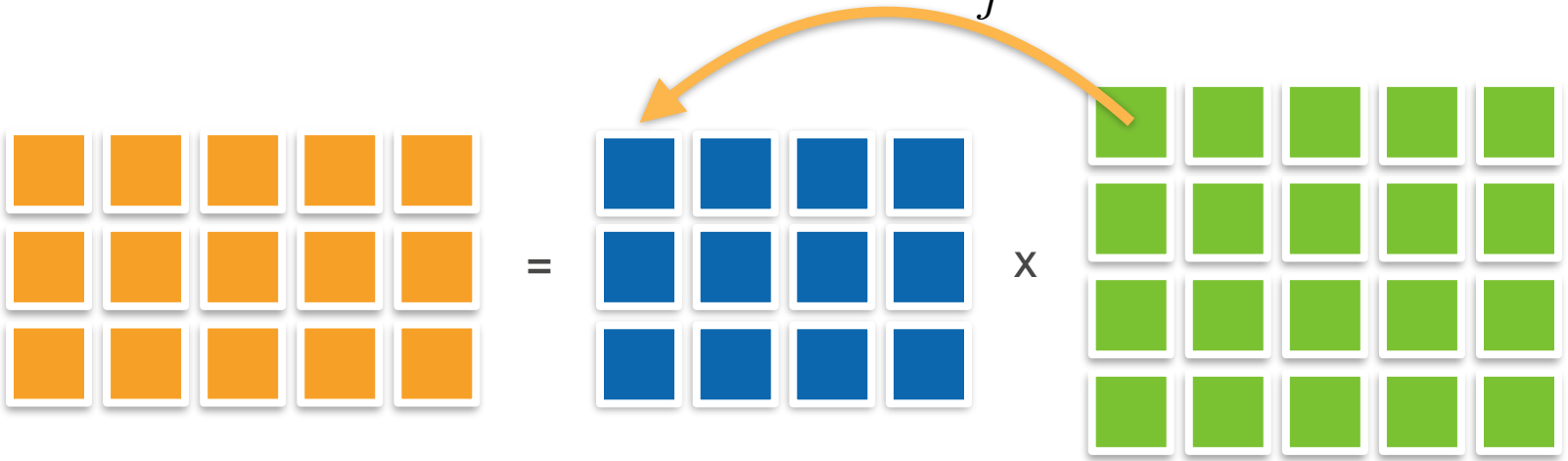
Matrices



Matrices

- **Multiplications (matrix matrix)**

$$C = AB \text{ where } C_{ik} = \sum_j A_{ij} B_{jk}$$



Matrices

- **Norms**

$$c = A \cdot b \text{ hence } \|c\| \leq \|A\| \cdot \|b\|$$

  - Choices depending on how to measure length of b and c
- **Popular norms**
  - Frobenius norm

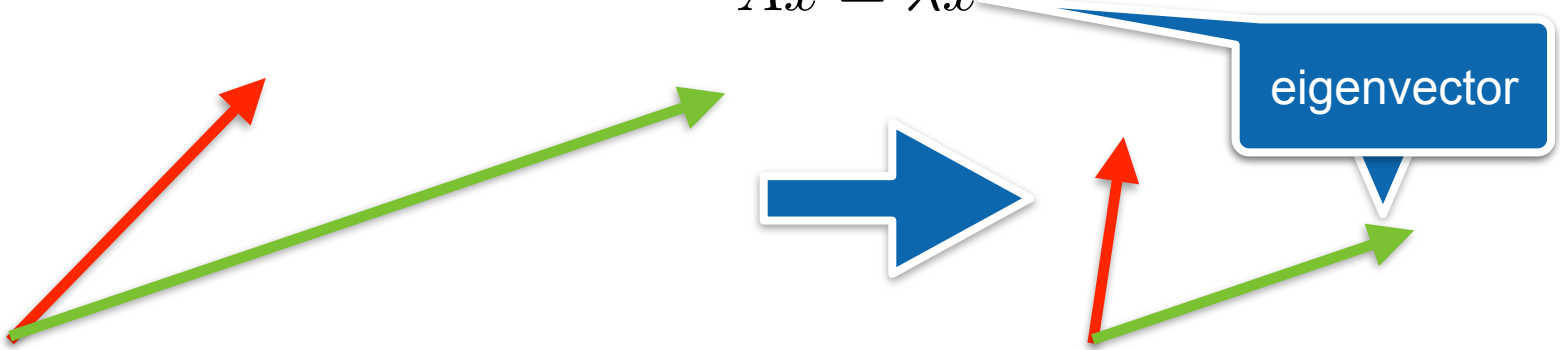
$$\|A\|_{\text{Frob}} = \left[\sum_{ij} A_{ij}^2\right]^{\frac{1}{2}}$$

  - H-infinity norm (or often just matrix norm) as smallest value for which condition holds for Euclidean norms on vectors.

Matrices

- **Eigenvectors and eigenvalue**
  - Vectors that aren't changed by the matrix

$$Ax = \lambda x$$



  - For symmetric matrices we can always find this

Special Matrices

- **Symmetric, antisymmetric** $A_{ij} = A_{ji}$ and $A_{ij} = -A_{ji}$

- **Positive definite**

$$\|x\|^2 = x^\top x \geq 0 \text{ generalizes to } x^\top A x \geq 0$$

(all positive eigenvalues)

Special Matrices

- **Orthogonal Matrices**
  - All rows of the matrix are orthogonal to each other
  - All rows of the matrix have unit length

$$U \text{ with } \sum_j U_{ij} U_{kj} = \delta_{ik}$$

Show that $U^\top U = \mathbf{1}$

  - Rewrite in matrix form

$$UU^\top = \mathbf{1}$$

- **Permutation Matrices**

$$P \text{ where } P_{ij} = 1 \text{ if and only if } j = \pi(i)$$

Show that P is orthogonal


GPUs love matrices and vectors
(they have many processors)


ndarray

N-dimensional Array Examples

- N-dimensional array, short for ndarray, is the main data structure for machine learning and neural networks

0-d (scalar)

```
1.0
```

A class label

1-d (vector)

```
[1.0, 2.7, 3.4]
```

A feature vector

2-d (matrix)

```
[[1.0, 2.7, 3.4]
 [5.0, 0.2, 4.6]
 [4.3, 8.5, 0.2]]
```

A example-by-feature matrix

ND Array Examples, cont

3-d

```
[[[0.1, 2.7, 3.4]
  [5.0, 0.2, 4.6]
  [4.3, 8.5, 0.2]]
 [[3.2, 5.7, 3.4]
  [5.4, 6.2, 3.2]
  [4.1, 3.5, 6.2]]]
```

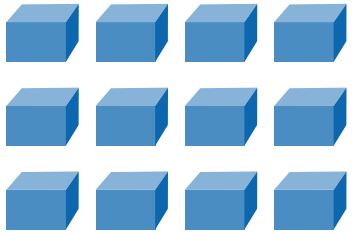
A RGB image (width x height x channels)

4-d

```
[[[[. . .
    . . .
    . . .]]]
```

A batch of RGB images (batch-size x width x height x channels)

5-d



```
[[[[. . .
    . . .
    . . .]]]
```

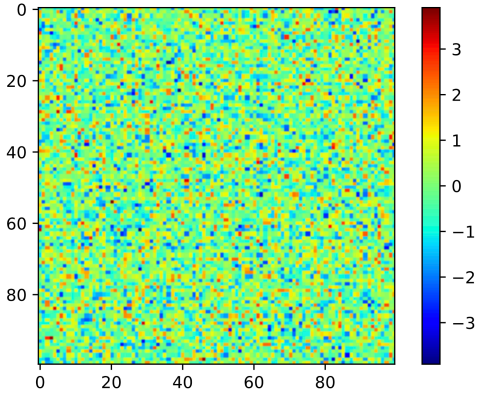
A batch of videos (batch-size x time x width x height x channels)

Create Arrays

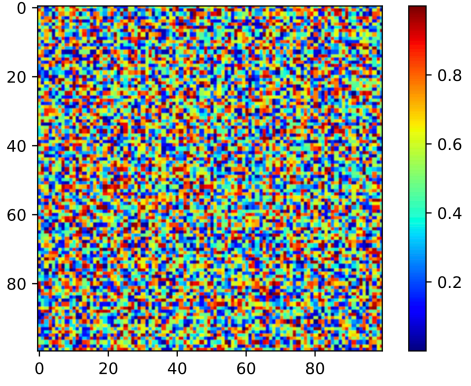
- Create arrays with
  - A shape, e.g. 3-by-4 matrix
  - Data type for each element, e.g. float
  - Element values, e.g all 0s, or random values

100-by-100 matrix with elements generated from

A normal distribution

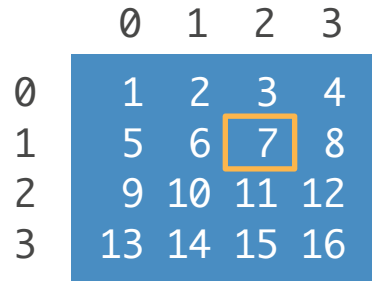


A uniform distribution

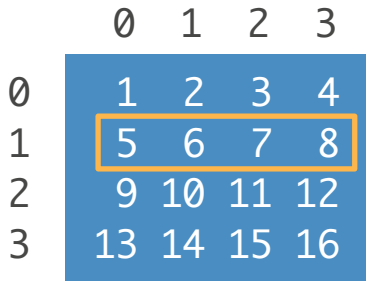


Access Elements

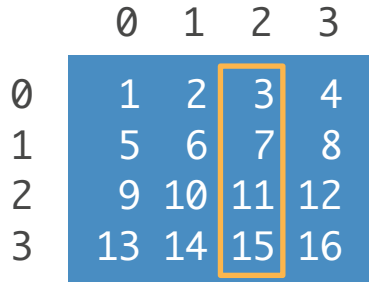
An element: [1, 2]

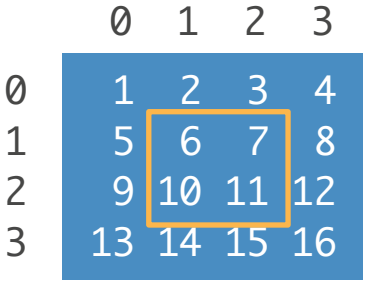


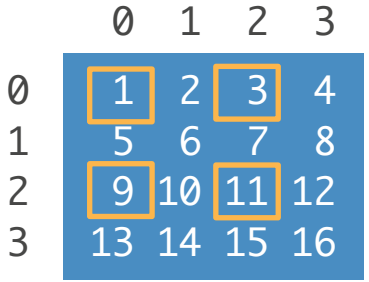
A row: [1, :]



A column: [1, :]







Pointers

- d2l.ai/chapter_crashcourse/ndarray.html
- beta.mxnet.io/guide/crash-course/1-ndarray.html
- Gilbert Strang's Linear Algebra course github.com/juanklopper/MIT_OCW_Linear_Algebra_18_06
- Berkeley HPC course (GPU sections) sites.google.com/lbl.gov/cs267-spr2018/

Summary

- **Logistics**
  Homework, Project, Exam, Hours, User group
- **Deep Learning**
  Quick overview
- **Software**
  Installation, cloud
- **Linear Algebra**
  Basic notation
- **NDArray**