Dive into Deep Learning for NLP

0. Introduction

Haibin Lin and Leonard

gluon-nlp.mxnet.io

aws

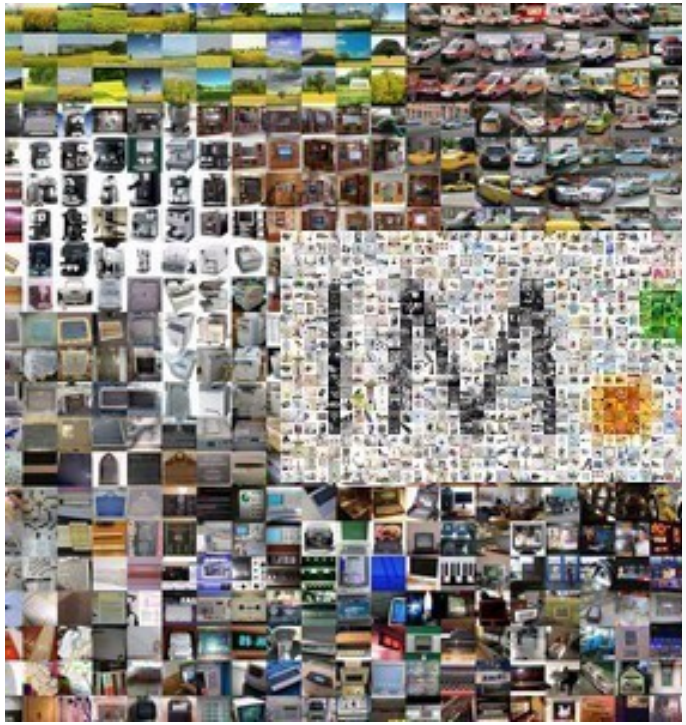
8:30-9:00	Continental Breakfast
9:00-9:45	Introduction and Setup
9:45-10:30	Neural Networks 101
10:30-10:45	Break
10:45-11:15	Machine Learning Basics
11:15-11:45	Context-free Representations for Language
11:45-12:15	Convolutional Neural Networks
12:15-13:15	Lunch Break
13:15-14:00	Recurrent Neural Networks
14:00-14:45	Attention Mechanism and Transformer
14:45-15:00	Coffee Break
15:00-16:15	Contextual Representations for Language
16:15-17:00	Language Generation


Deep Learning

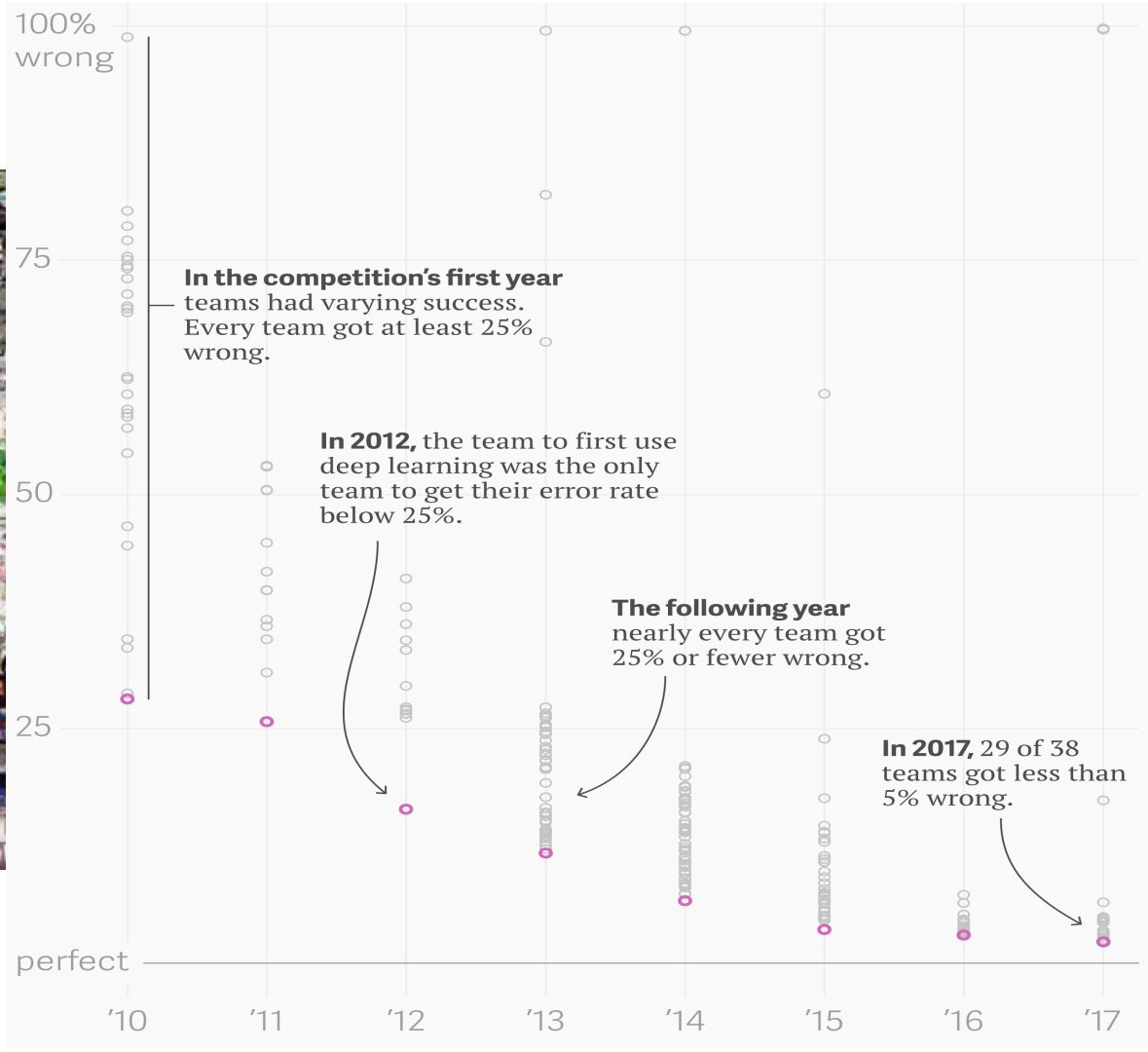
Classify Images

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

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

gluon-nlp.mxnet.io

https://github.com/matterport/Mask_RCNN

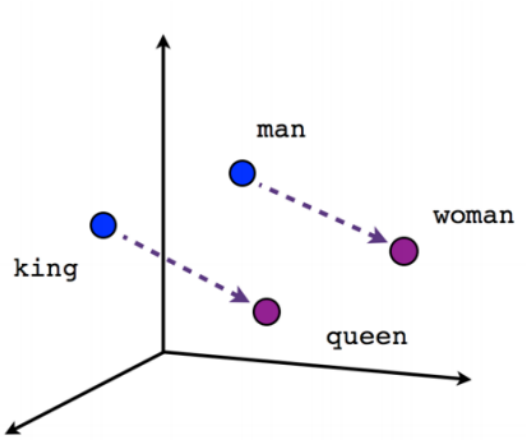
Style transfer

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

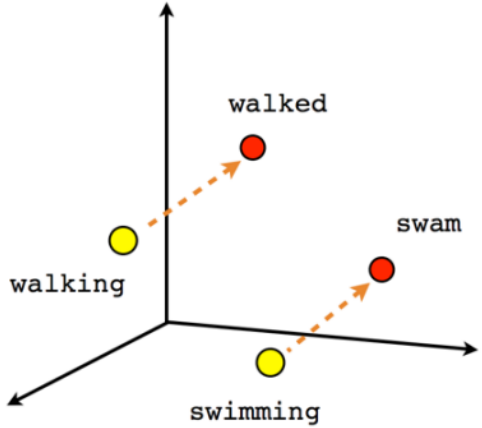
Synthesize Images

Brock et al, 2019

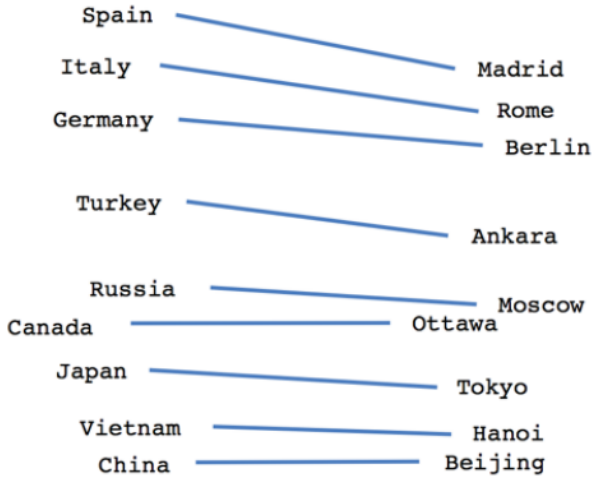
Analogies



Male-Female



Verb tense

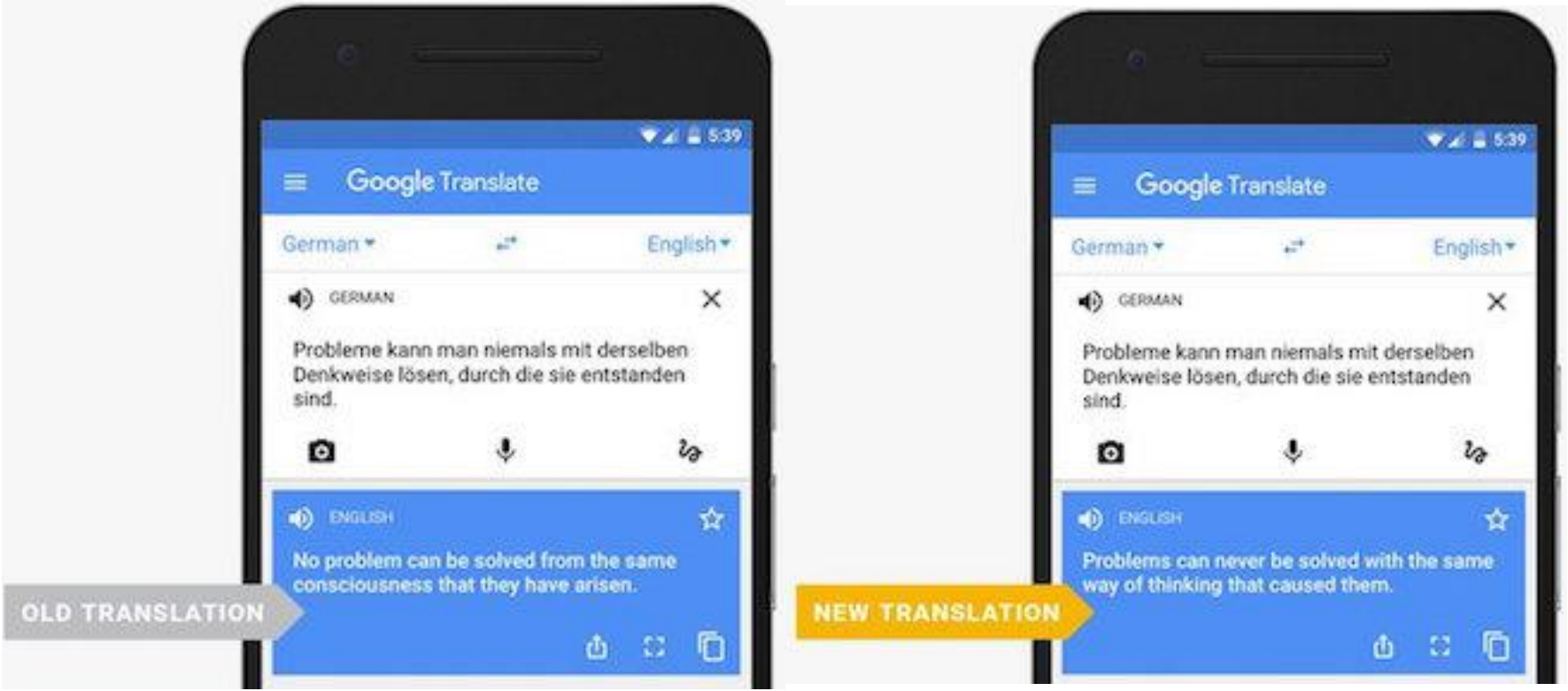


Country-Capital

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

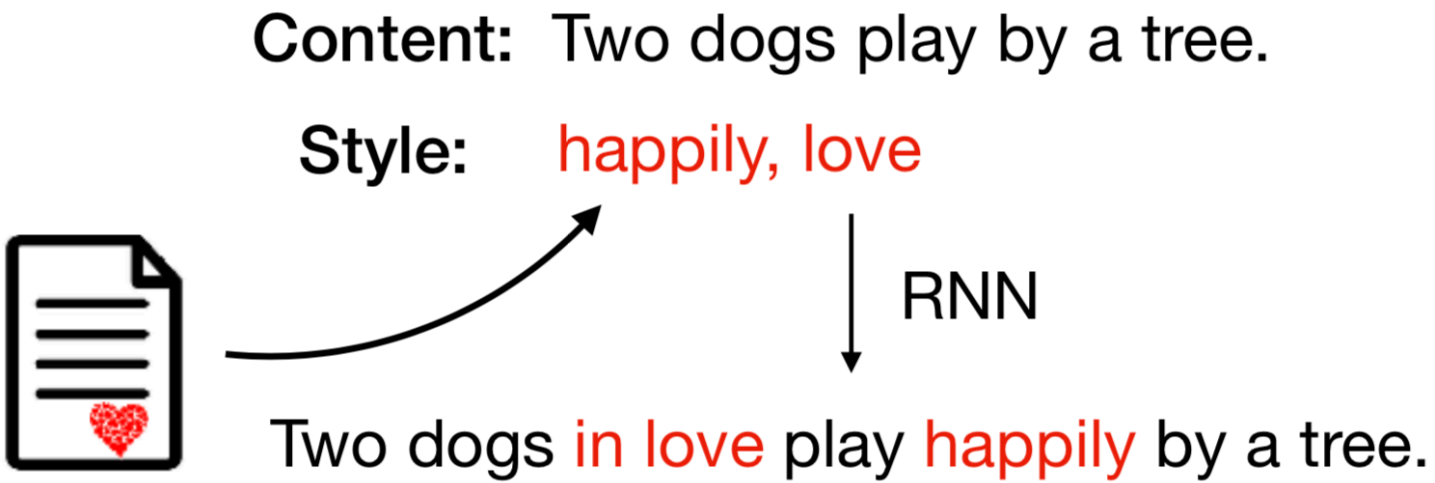
aws

Machine Translation



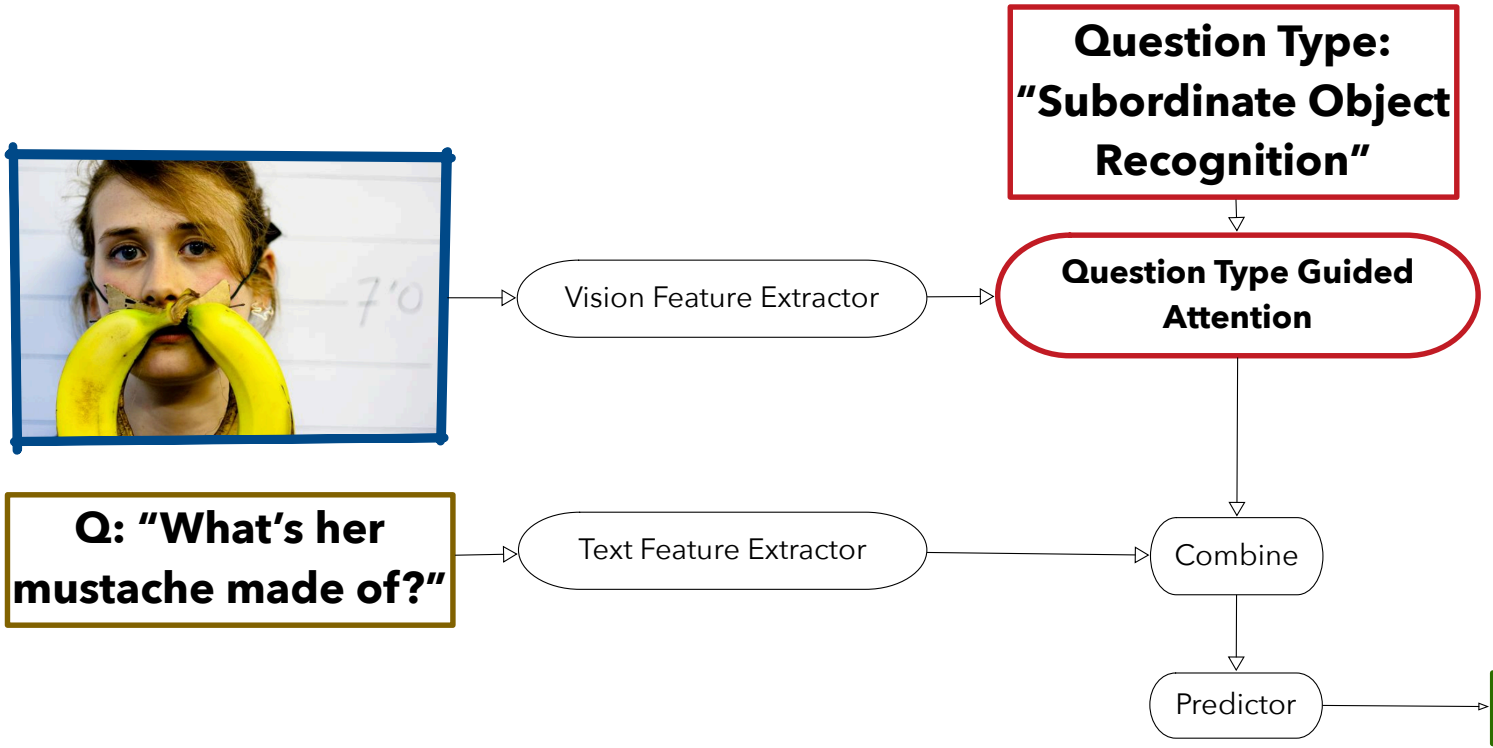
https://www.pcmag.com/news/349610/google-expands-neural-networks-for-language-translation

Text synthesis



Li et al, NACCL, 2018

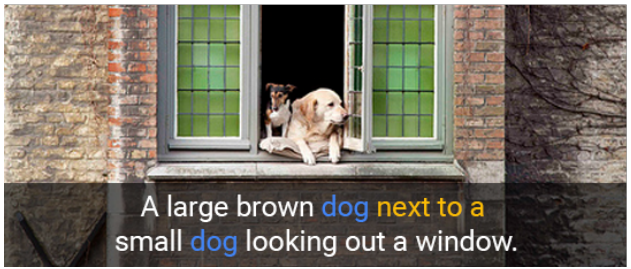
Question answering



Shi et al, 2018, Arxiv

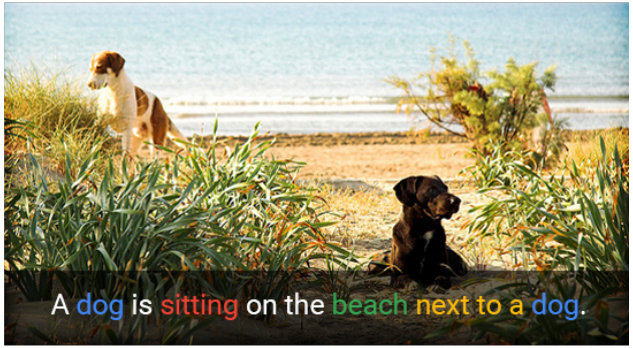
Image captioning

Human captions from the training set



gluon-nlp.mxnet.io

Automatically captioned



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

aws


Software

Programming for deep learning used to be…

```
void computeLogregSoftmaxGrad(NVMatrix& labels, NVMatrix& probs, NVMatrix& target, bool add, float coeff) {
    int numCases = probs.getLeadingDim();
    int numOut = probs.getFollowingDim();
    assert(labels.getNumElements() == numCases);
    assert(probs.isContiguous());
    assert(target.isContiguous());
    assert(labels.isContiguous());
    assert(probs.isTrans());

    dim3 threads(LOGREG_GRAD_THREADS_X, LOGREG_GRAD_THREADS_Y);
    dim3 blocks(DIVUP(numCases, LOGREG_GRAD_THREADS_X), DIVUP(numOut, LOGREG_GRAD_THREADS_Y));
    if (!add) {
        target.resize(probs);
        kLogregSoftmaxGrad<false><<<blocks, threads>>>(probs.getDevData(), labels.getDevData(), target.getDevData(),
                                                       numCases, numOut, coeff);
    } else {
        kLogregSoftmaxGrad<true><<<blocks, threads>>>(probs.getDevData(), labels.getDevData(), target.getDevData(),
                                                      numCases, numOut, coeff);
    }

    getLastCudaError("computeLogregSoftmaxGrad: Kernel execution failed");
}
```

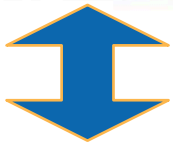
Krizhevsky et al, code from cuda-convnet (AlexNet)

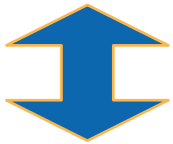
Programming for deep learning today

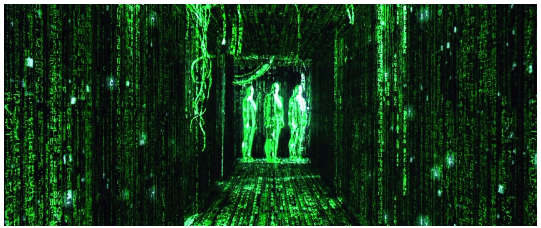
```
loss_function = gluon.loss.SoftmaxCrossEntropyLoss()
loss = loss_function(output)
loss.backward()
```

Programming for deep learning today


```
output = model(data)
```

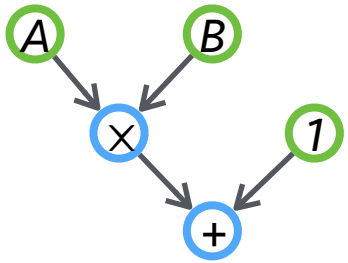






[crazy GPU stuff]

Declarative (Symbolic) Programs




```
A = Variable('A')
B = Variable('B')
C = B * A
D = C + 1
f = compile(D)
d = f(A=np.ones(10),
      B=np.ones(10)*2)
```

C can share memory with *D*

- **Advantages**:
  - Opportunities for optimization
  - Easy to serialize models
  - More portable across languages
- **Disadvantages**:
  - Hard to debug
  - Unsuitable for dynamic graphs
  - Can't use native code

Imperative Programs


```
import numpy as np
a = np.ones(10)
b = np.ones(10) * 2
c = b * a
print c
d = c + 1
```

Easy to debug, easy to code

- **Advantages**
  - Straightforward and flexible
  - Use language natively (loops, control flow, debugging)
- **Disadvantages**
  - Hard to optimize (Fixed with JIT compiler!)

MXNet Gluon

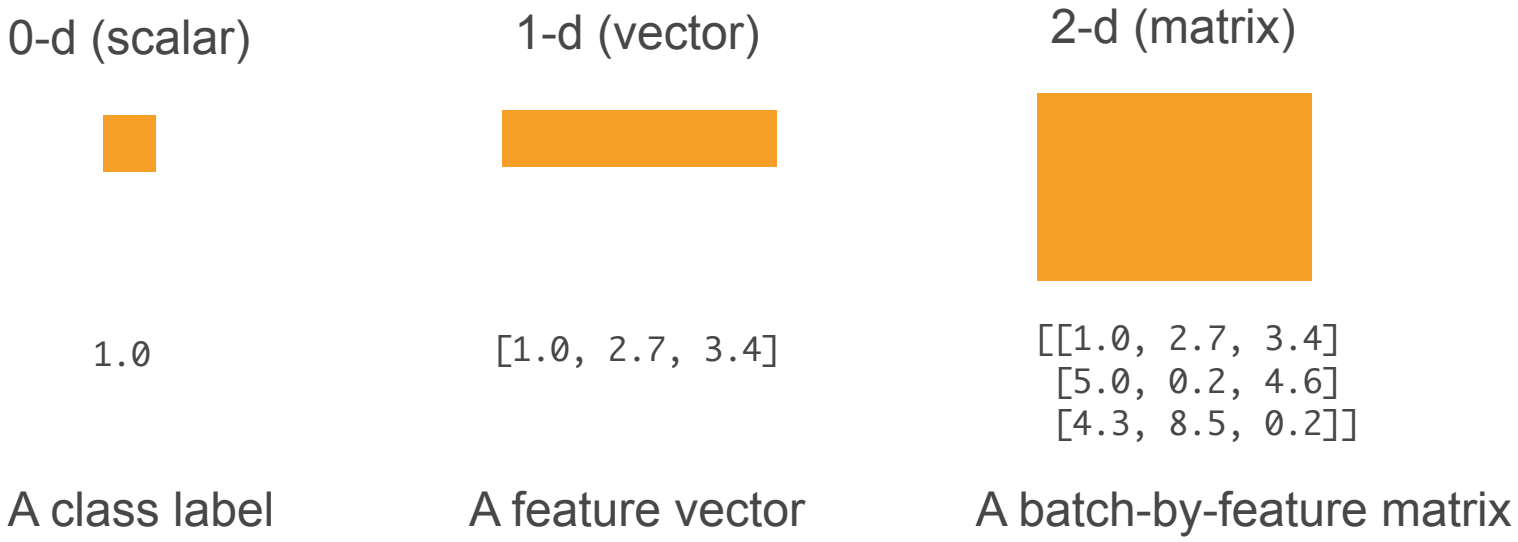
Q: Why GLUON?

A:  + 


NDArray

N-dimensional Array Examples

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



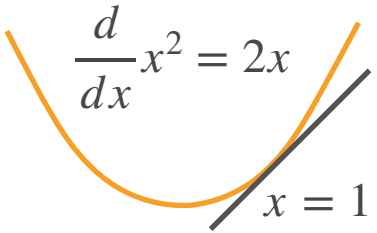
Review Scalar Derivative

y	a	x^n	$\exp(x)$	$\log(x)$	$\sin(x)$
$\frac{dy}{dx}$	0	nx^{n-1}	$\exp(x)$	$\frac{1}{x}$	$\cos(x)$

(a is not a function of x)

y	$u+v$	uv	$y=f(u),\ u=g(x)$
$\frac{dy}{dx}$	$\frac{du}{dx}+\frac{dv}{dx}$	$\frac{du}{dx}v+\frac{dv}{dx}u$	$\frac{dy}{du}\frac{du}{dx}$

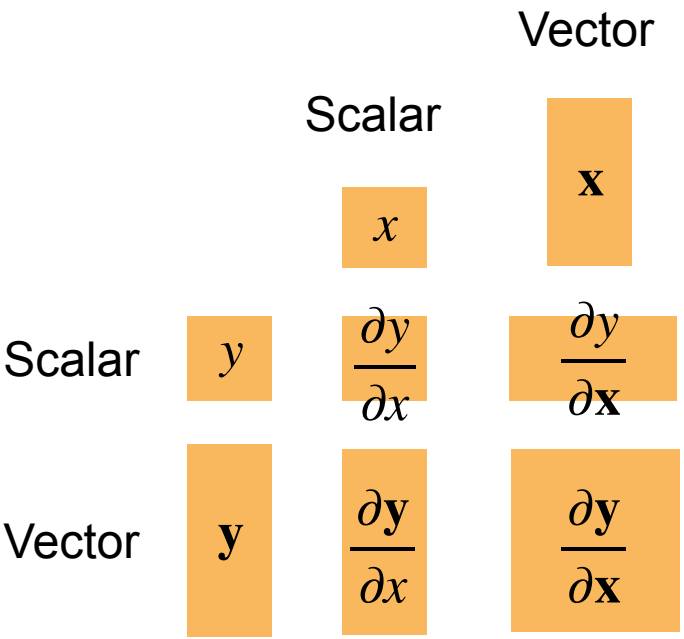
Derivative is the slope of the tangent line



The slope of the tangent line is 2

Gradients

- Generalize derivatives into vectors



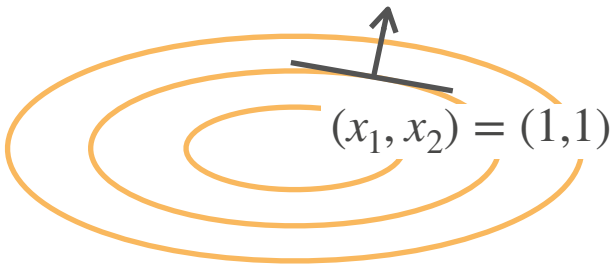
$\partial y / \partial \mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \frac{\partial y}{\partial \mathbf{x}} = \left[\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_n} \right]$$

	x	$\mathbf{x}$
y	$\frac{\partial y}{\partial x}$	$\frac{\partial y}{\partial \mathbf{x}}$
$\mathbf{y}$	$\frac{\partial \mathbf{y}}{\partial x}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

$$\frac{\partial}{\partial \mathbf{x}} x_1^2 + 2x_2^2 = [2x_1, 4x_2]$$

Direction (2, 4), perpendicular to the contour lines



Generalize to Vectors

- Chain rule for scalars:

$$y = f(u),\ u = g(x) \qquad \frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$$

- Generalize to vectors straightforwardly

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial \mathbf{x}} \qquad \frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$$

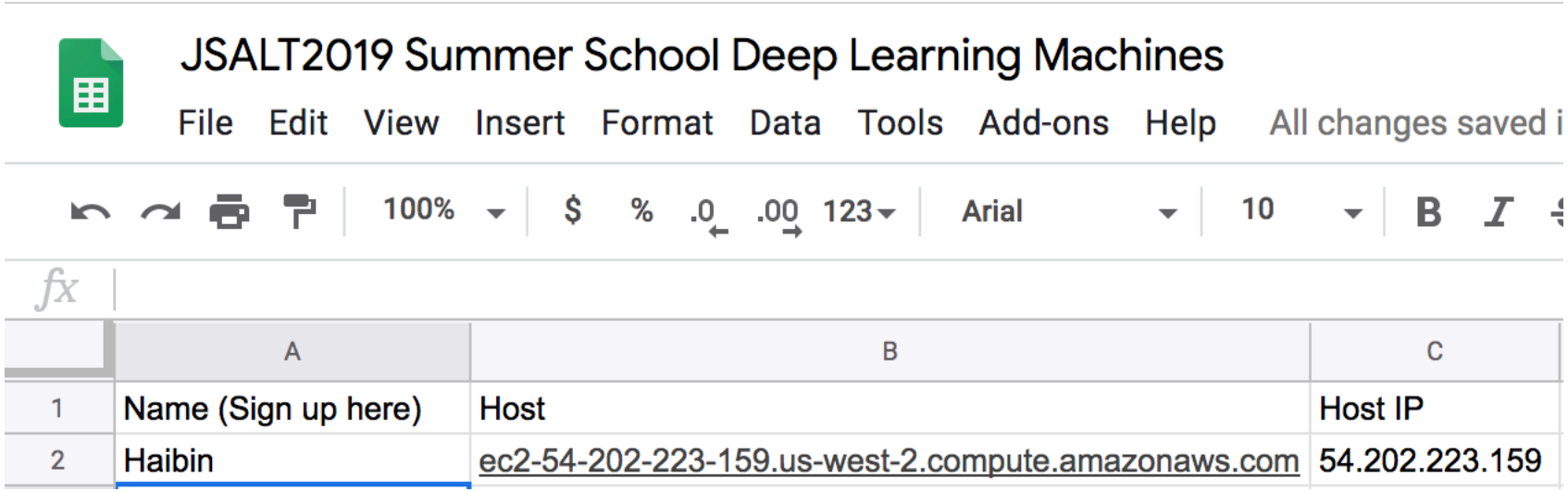
$$(1,n)\quad (1,)\ (1,n) \qquad (1,n)\quad (1,k)\ \ (k,n) \qquad (m,n)\ \ (m,k)\ (k,n)$$


Environment Setup

- **Setup Instructions**
  - https://tinyurl.com/JSALT2019
- **Tutorial Website**
  - https://jsalt19.mxnet.io
- **Related Resources**
  - GluonNLP http://gluon-nlp.mxnet.io/
  - GluonCV http://gluon-cv.mxnet.io/
  - Deep Graph Library http://dgl.ai/
  - Dive into deep learning book http://d2l.ai/
  - MXNet forum https://discuss.mxnet.io/

Grab a Machine

- Fill in your name in https://tinyurl.com/jsalt-gpu



JSALT2019 Summer School Deep Learning Machines

File Edit View Insert Format Data Tools Add-ons Help All changes saved i

100% $ % .0 .00 123 Arial 10 B I

fx

	A	B	C
1	Name (Sign up here)	Host	Host IP
2	Haibin	ec2-54-202-223-159.us-west-2.compute.amazonaws.com	54.202.223.159

Connect to the Instance

- download the key from [secret link here]
- ssh to the machine

  chmod 400 jsalt19.pem;
  ssh -i jsalt19.pem ubuntu@your_ip -L 8888:localhost:8888

```
➜  Downloads chmod 400 jsalt19.pem; ssh -i jsalt19.pem ubuntu@54.202.223.159 -L 8888:localhost:8888
=============================================================================
      __|  __|_  )
      _|  (     /   JSALT 2019 Deep Learning (Ubuntu)
     ___|\___|___|
=============================================================================

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1084-aws x86_64v)

Apache MXNet Website: https://beta.mxnet.io
D2L.ai: http://d2l.ai
GluonNLP: https://gluon-nlp.mxnet.io
GluonCV: https://gluon-cv.mxnet.io

Enjoy deep learning!
```

Prepare Environment

- Create a tmux session
  tmux new -s gluonnlp
- Create a conda environment
  conda env update --prune mxnet/jsalt19; conda activate jsalt19

```
      Successfully uninstalled gluonnlp-0.7.0.dev0
Successfully installed gluonnlp-0.7.0.dev0 mxnet-cu92mkl-1.5.0b20190613

#
# To activate this environment, use
#
#     $ conda activate jsalt19
#
# To deactivate an active environment, use
#
#     $ conda deactivate

ubuntu@ip-172-31-4-238:~$ conda activate jsalt19
(jsalt19) ubuntu@ip-172-31-4-238:~$
```

Clone and Launch Notebooks

git clone https://github.com/eric-haibin-lin/JSALT19-GluonNLP;

jupyter notebook

```
(jsalt19) ubuntu@ip-172-31-4-238:~$ jupyter notebook
[I 02:12:29.882 NotebookApp] Using EnvironmentKernelSpecManager...
[I 02:12:29.882 NotebookApp] Started periodic updates of the kernel list (every 3 minutes).
[I 02:12:30.923 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.4.1
[I 02:12:30.923 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 02:12:30.923 NotebookApp] The Jupyter Notebook is running at:
[I 02:12:30.923 NotebookApp] http://localhost:8888/?token=a0f8f026225a211eed7ee86f5d2f6dd926c82aee4dbda8e
[I 02:12:30.923 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 02:12:30.927 NotebookApp] No web browser found: could not locate runnable browser.
[C 02:12:30.928 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///run/user/1000/jupyter/nbserver-8650-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=a0f8f026225a211eed7ee86f5d2f6dd926c82aee4dbda8e1
[I 02:12:30.928 NotebookApp] Starting initial scan of virtual environments...
```