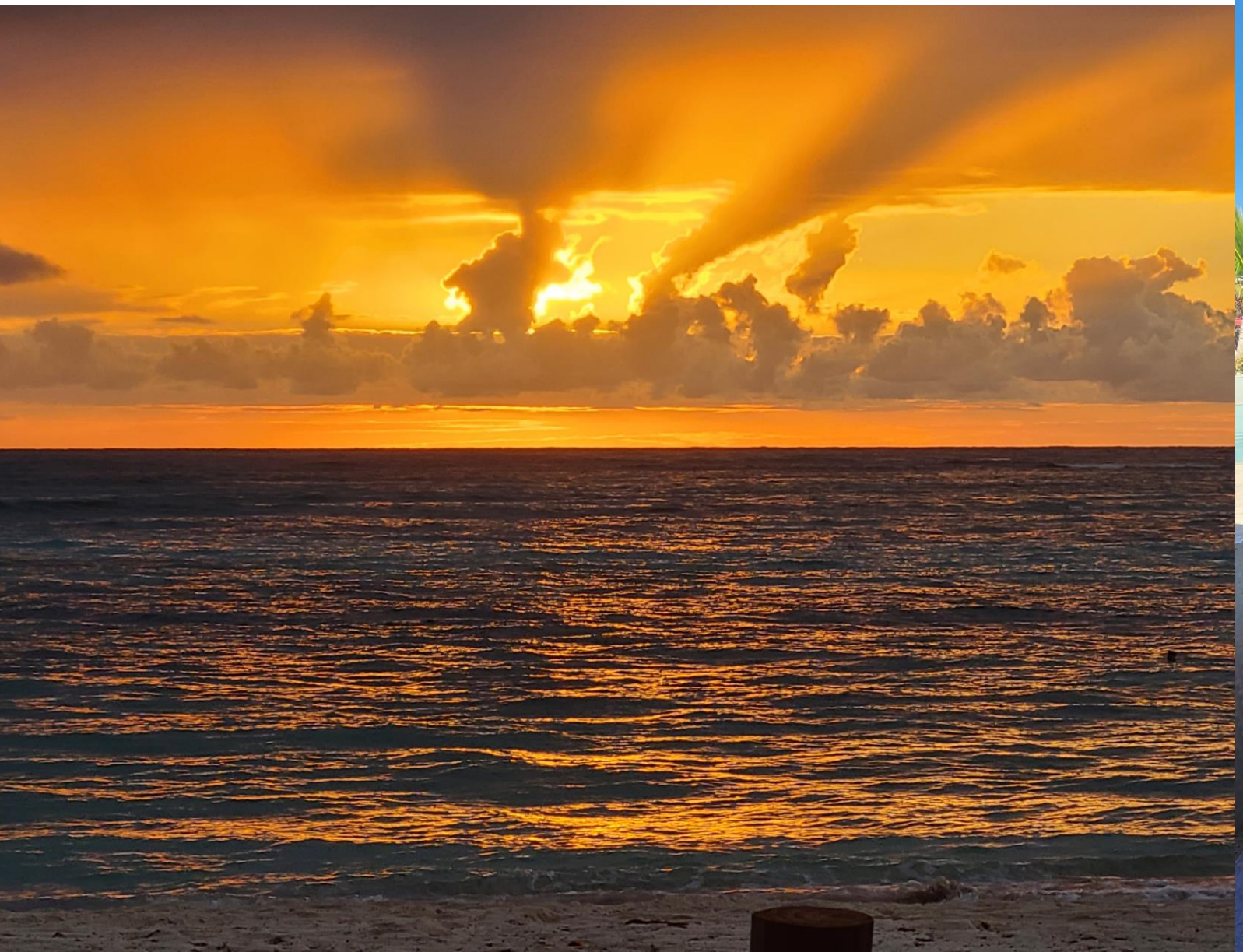
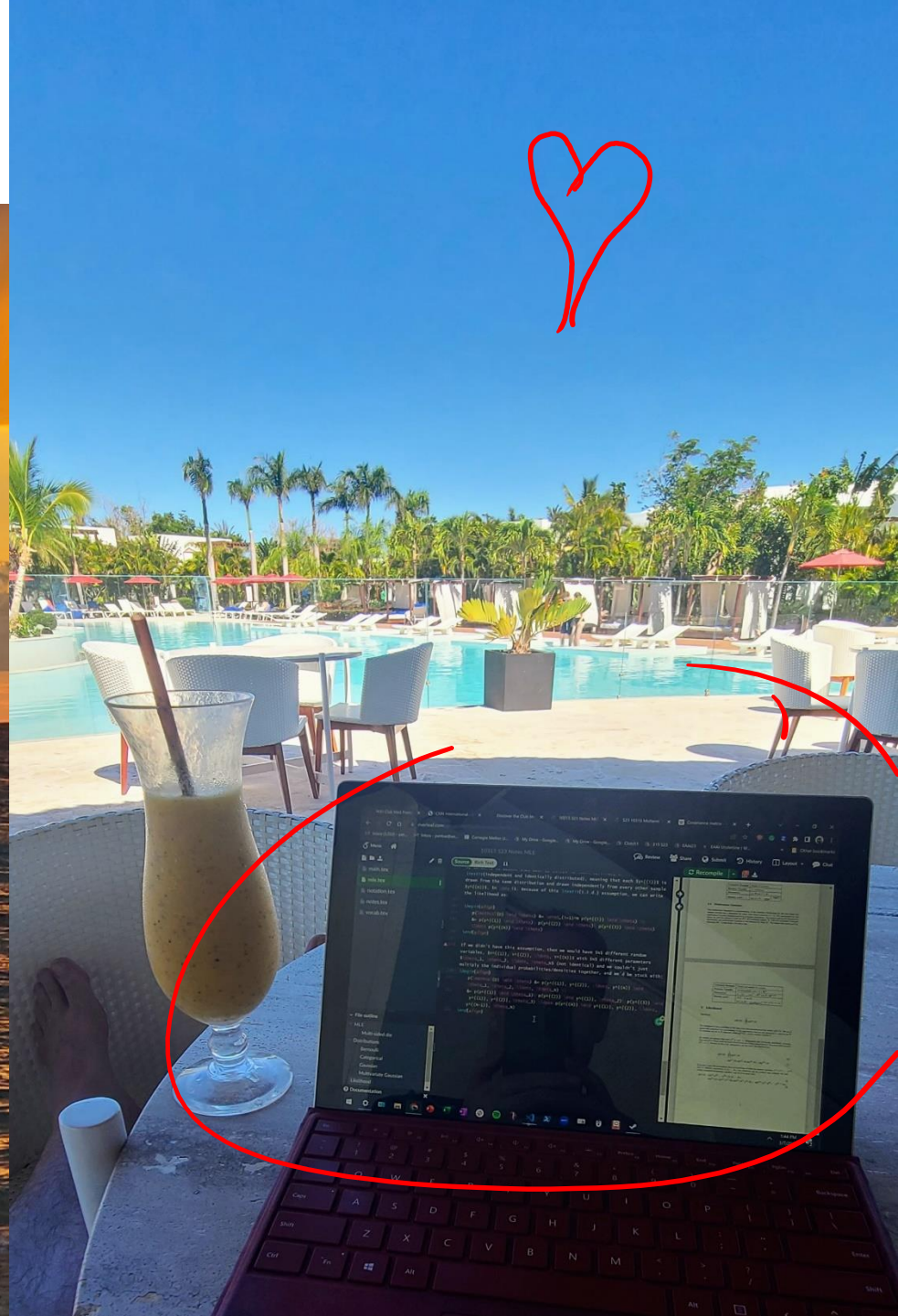
Welcome back!





Course Updates

Feedback – Thanks!

HW Schedule

HW6 / HW7 out Fri am

MLE "Pre"-reading for Wednesday

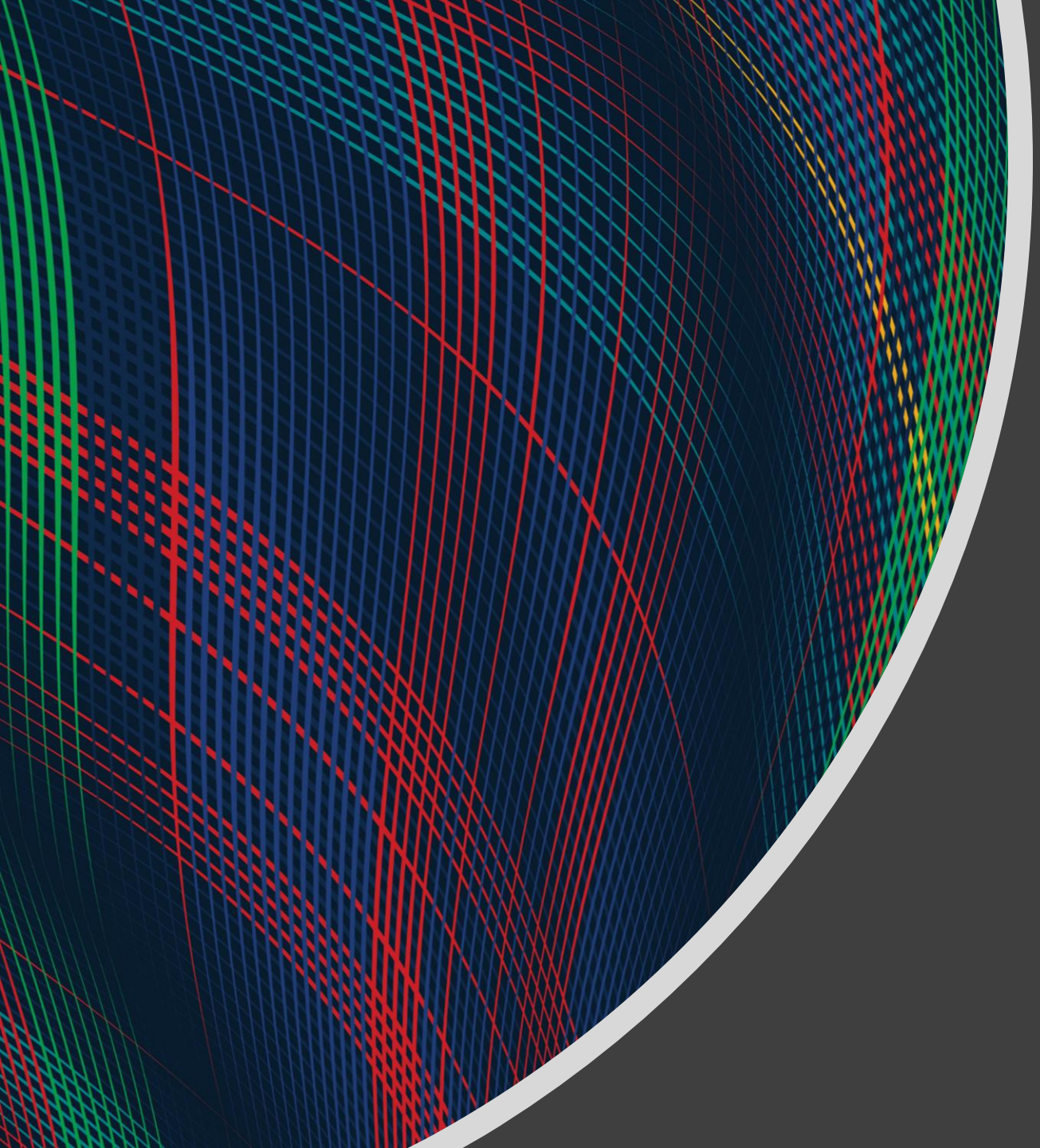
Help

Where are we?

Plan

Today

- Applications
  - ML design
  - ML model cards
  - Toolkits for neural networks
  - Neural networks for imaging
  - Neural networks for language



10-315 Introduction to ML

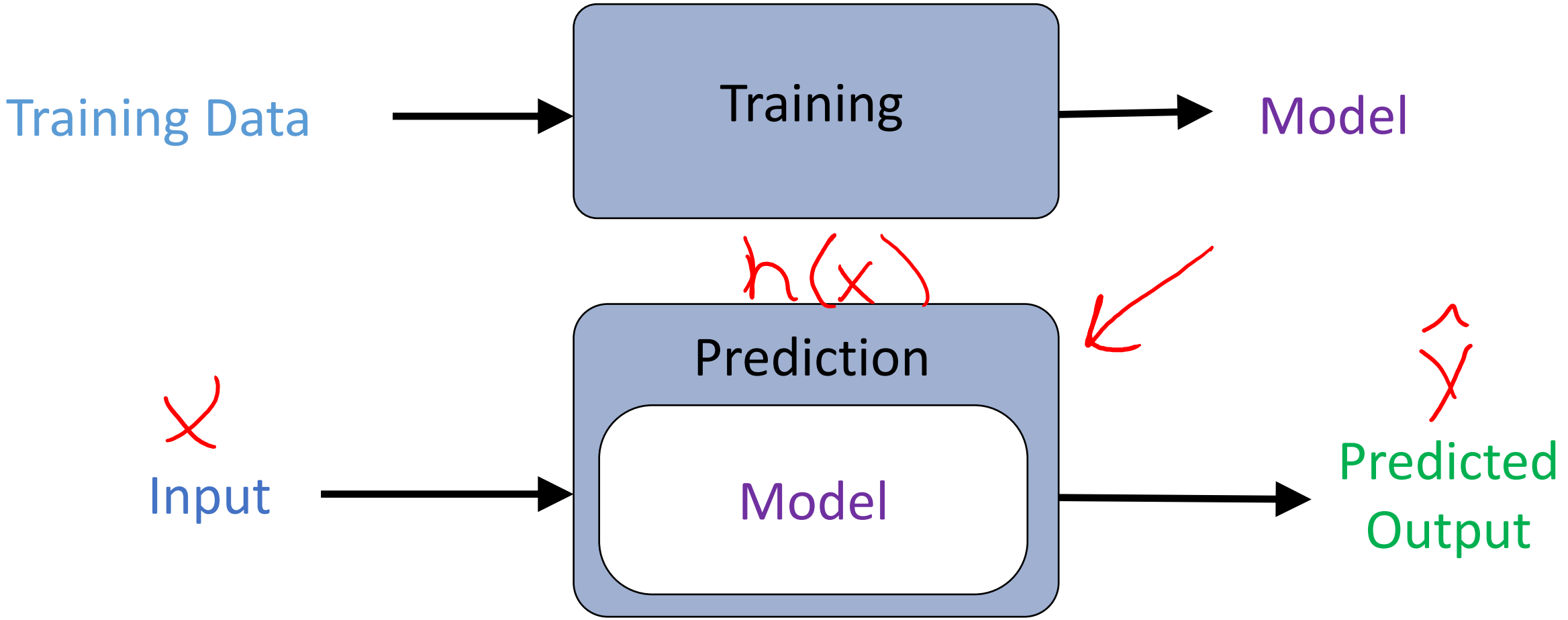
ML Applications & Neural Networks for Imaging and Language

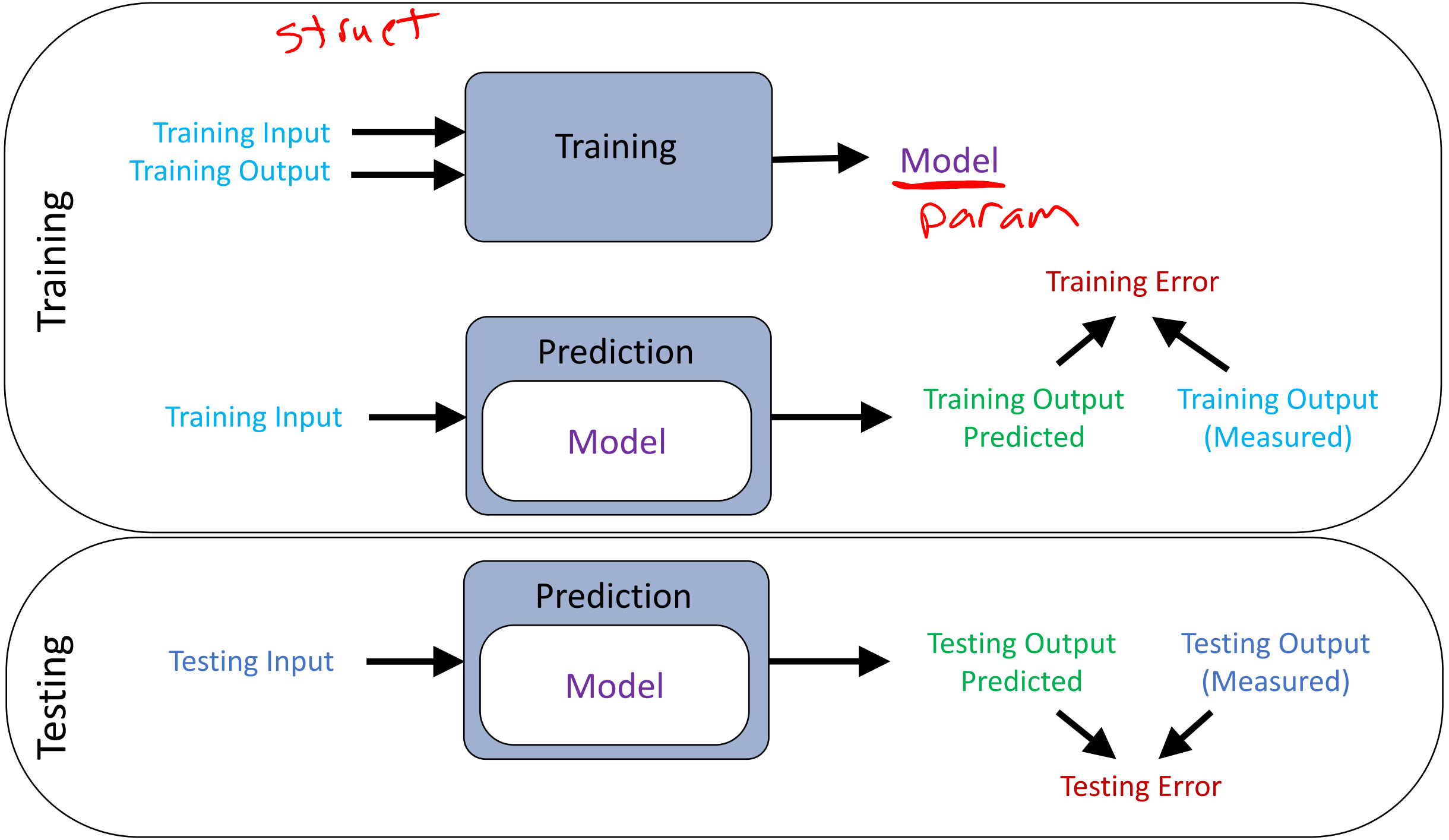
Instructor: Pat Virtue

Model Design

Reminder: Machine Learning Tasks

Using (training) data to learn a model that we'll later use for prediction




struct
Training
Training Input
Training Output
Training
Model
param
Training Error
Prediction
Model
Training Input
Training Output
Predicted
Training Output
(Measured)
Testing
Testing Input
Prediction
Model
Testing Output
Predicted
Testing Output
(Measured)
Testing Error

AI in the News

News:
https://gadgets.ndtv.com/social-networking/news/flemish-scrollers-ai-ml-bot-tool-software-belgium-politicians-distracted-phone-detect-troll-twitter-2480493
Source: Twitter @Flemish Scroller

Gadgets 360
An NDTV venture

HOME GUIDE NEWS REVIEWS FEATURES

AI-Based Bot Detects Politicians Distracted by Phone, Posts Photo on Twitter and Asks Them to Focus

The software searches for phones and then look for distracted politicians during livestreams of parliamentary sessions.

By Edited by Gadgets 360 Newsdesk | Updated: 6 July 2021 16:34 IST

Share on Facebook | Tweet | Snapchat | Share | Reddit | Email | Comment

ML App Design

Distracted Politician Detection

Design the app by connecting smaller input/output ML Tasks

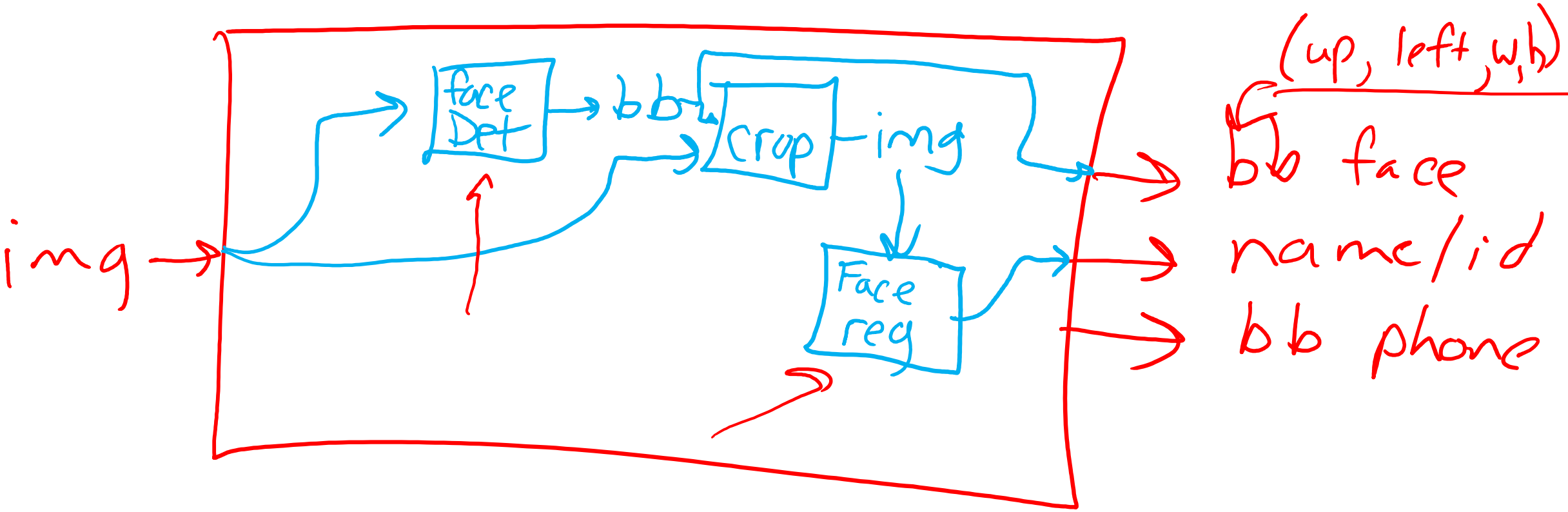


Image credit: Twitter @Flemish Scroller

ML App Design

Distracted Politician Detection

What is the performance measure?

Image credit: Twitter @Flemish Scroller

ML App Design

Distracted Politician Detection

What data do we need?

Image credit: Twitter @Flemish Scroller

ML App Design

Distracted Politician Detection

What can go wrong

Image credit: Twitter @Flemish Scroller

ML Model Cards

Mitchell, Margaret, et al.

"Model cards for model reporting.“

Proceedings of the conference on fairness, accountability, and transparency. 2019.

ML Model Cards

Model Card

- **Model Details**. Basic information about the model.
  - Person or organization developing model
  - Model date
  - Model version
  - Model type
  - Information about training algorithms, parameters, fairness constraints or other applied approaches, and features
  - Paper or other resource for more information
  - Citation details
  - License
  - Where to send questions or comments about the model
- **Intended Use**. Use cases that were envisioned during development.
  - Primary intended uses
  - Primary intended users
  - Out-of-scope use cases
- **Factors**. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.
  - Relevant factors
  - Evaluation factors
- **Metrics**. Metrics should be chosen to reflect potential real-world impacts of the model.
  - Model performance measures
  - Decision thresholds
  - Variation approaches
- **Evaluation Data**. Details on the dataset(s) used for the quantitative analyses in the card.
  - Datasets
  - Motivation
  - Preprocessing
- **Training Data**. May not be possible to provide in practice. When possible, this section should mirror Evaluation Data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution over various factors in the training datasets.
- **Quantitative Analyses**
  - Unitary results
  - Intersectional results
- **Ethical Considerations**
- **Caveats and Recommendations**

Exercise: ML Model Hunt

Search the web to find the model card for a real-world model

Model Card

- **Model Details**. Basic information about the model.
  - Person or organization developing model
  - Model date
  - Model version
  - Model type
  - Information about training algorithms, parameters, fairness constraints or other applied approaches, and features
  - Paper or other resource for more information
  - Citation details
  - License
  - Where to send questions or comments about the model
- **Intended Use**. Use cases that were envisioned during development.
  - Primary intended uses
  - Primary intended users
  - Out-of-scope use cases
- **Factors**. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.
  - Relevant factors
  - Evaluation factors

Teachable Machine

Transfer learning / fine-tuning

Neural Network Toolkits

Pytorch

Network Network Toolkits

Pytorch in this course

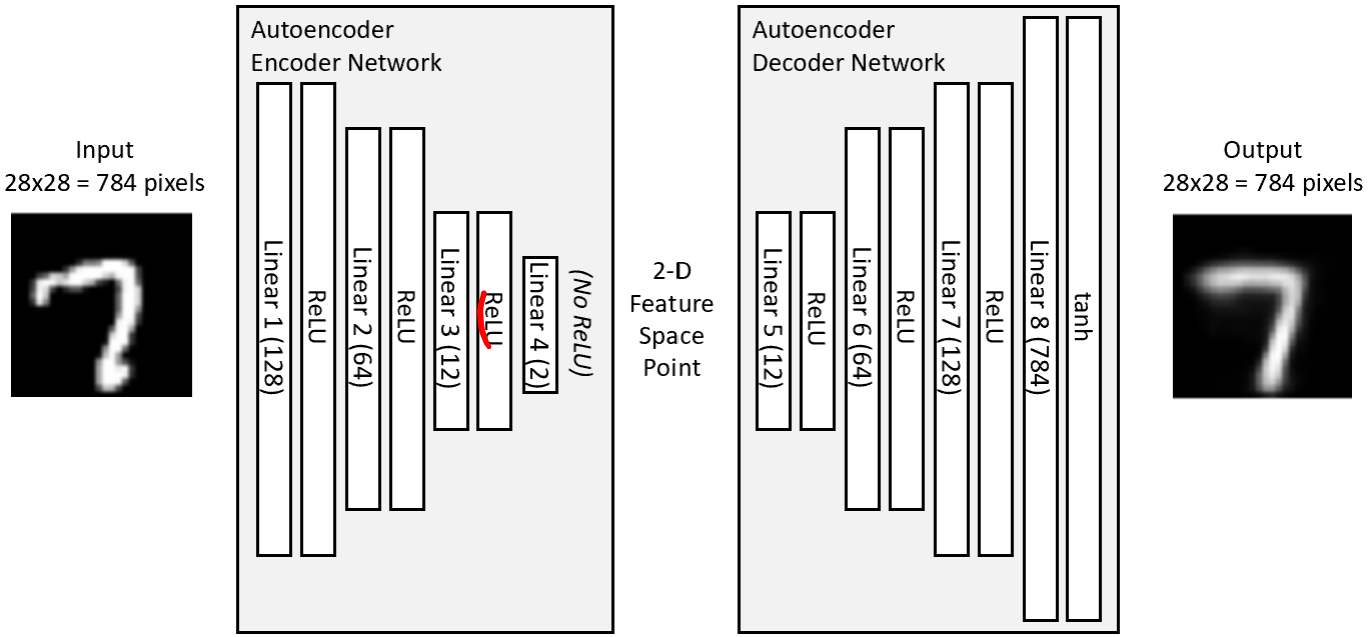
- Already used behind the scenes in HW1
- HW7 transfer learning and exploration
- Mini-project if you want

Pytorch for HW1 Networks

Autoencoder

bottleneck

128 → 64 → 12 → 2 → 12 → 64 → 128 → 784



Source: https://github.com/L1aoXingyu/pytorch-beginner/blob/master/08-AutoEncoder/simple_autoencoder.py

```python
class autoencoder(nn.Module):
    def __init__(self):
        bottleneck = 2
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128),
            nn.ReLU(True),
            nn.Linear(128, 64),
            nn.ReLU(True),
            nn.Linear(64, 12),
            nn.ReLU(True),
            # nn.Linear(12, 3))
            nn.Linear(12, bottleneck))
        self.decoder = nn.Sequential(
            # nn.Linear(3, 12),
            nn.Linear(bottleneck, 12),
            nn.ReLU(True),
            nn.Linear(12, 64),
            nn.ReLU(True),
            nn.Linear(64, 128),
            nn.ReLU(True),
            nn.Linear(128, 28 * 28),
            nn.Tanh())

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Pytorch for HW1 Networks

Autoencoder

```
for epoch in range(num_epochs):
    for data in dataloader:
        img, _ = data
        img = img.view(img.size(0), -1)
        if torch.cuda.is_available():
            img = Variable(img).cuda()
        else:
            img = Variable(img)
        # ===================forward=====================
        output = model(img)
        loss = criterion(output, img)
        # ===================backward====================
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

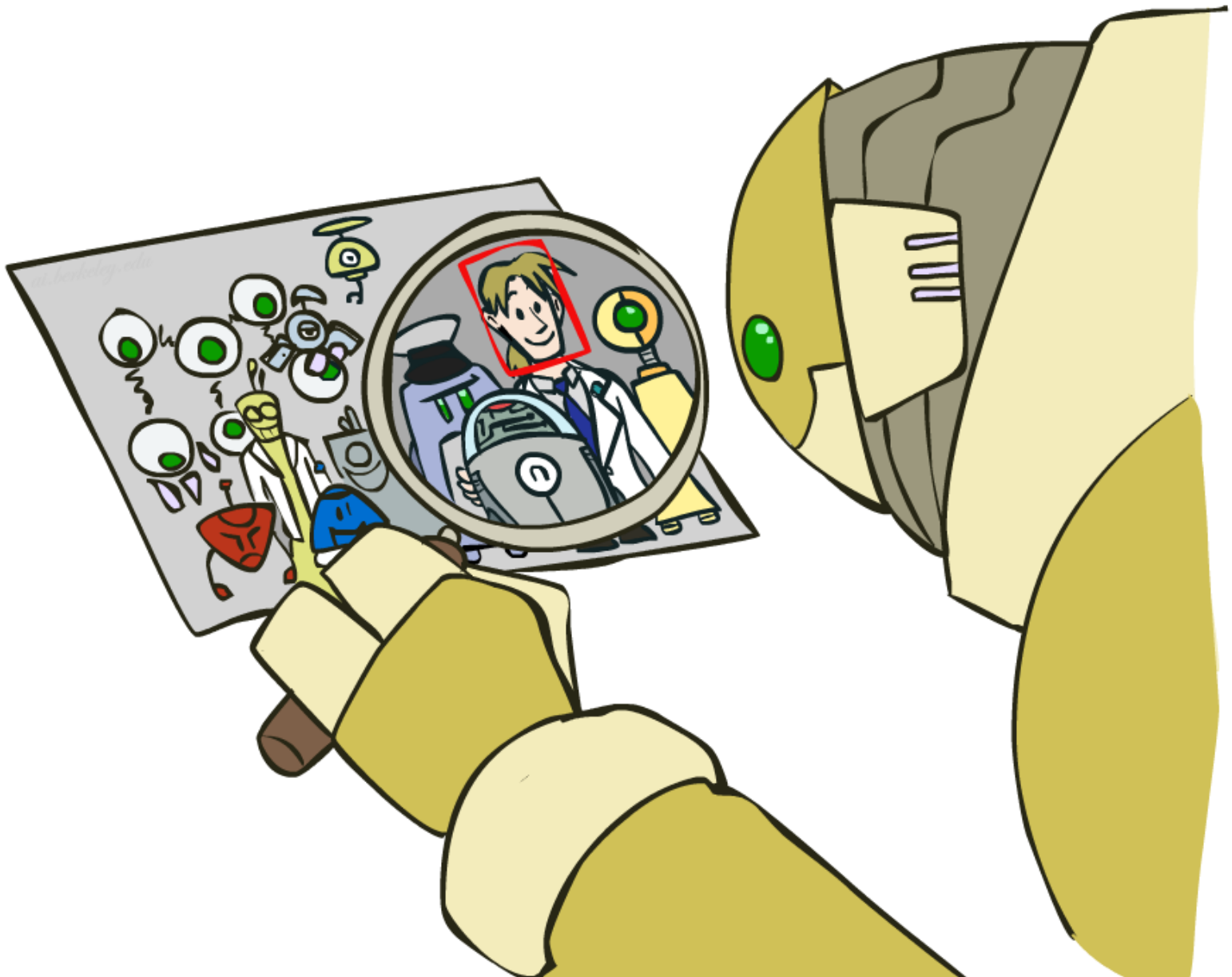
Source: https://github.com/L1aoXingyu/pytorch-beginner/blob/master/08-AutoEncoder/simple_autoencoder.py

```
class autoencoder(nn.Module):
    def __init__(self):
        bottleneck = 2
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128),
            nn.ReLU(True),
            nn.Linear(128, 64),
            nn.ReLU(True),
            nn.Linear(64, 12),
            nn.ReLU(True),
            # nn.Linear(12, 3))
            nn.Linear(12, bottleneck))
        self.decoder = nn.Sequential(
            # nn.Linear(3, 12),
            nn.Linear(bottleneck, 12),
            nn.ReLU(True),
            nn.Linear(12, 64),
            nn.ReLU(True),
            nn.Linear(64, 128),
            nn.ReLU(True),
            nn.Linear(128, 28 * 28),
            nn.Tanh())

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Convolutional Neural Networks

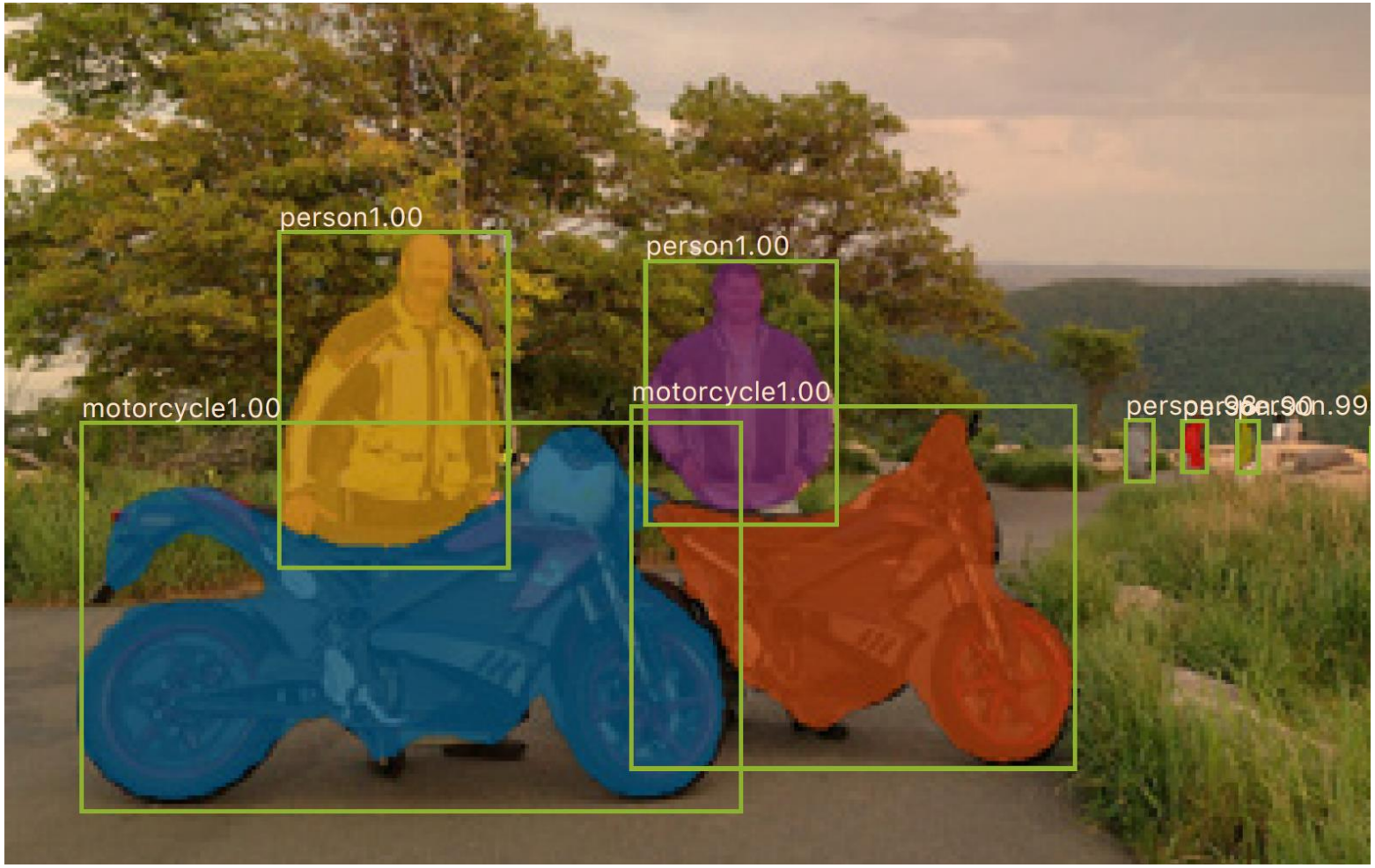
Computer Vision: How far along are we?



Computer Vision: How far along are we?

Terminator 2, 1991 https://www.youtube.com/watch?v=9MeaaCwBW28

Computer Vision: How far along are we?



0.2 seconds per image (2017)

Mask R-CNN

He, Kaiming, et al. "Mask R-CNN." *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017.

Computer Vision: How far along are we?

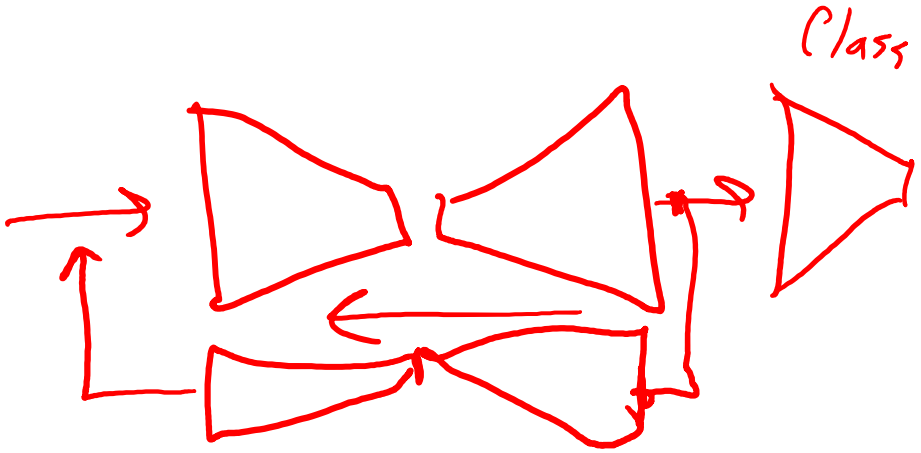
“My CPU is a neural net processor, a learning computer”

Terminator 2, 1991

Computer Vision: Autonomous Driving

Tesla, Inc: https://vimeo.com/192179726

Computer Vision: Domain Transfer



CycleGAN

Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV 2017.

Text to Image

GigaGAN

Changing texture with prompting. At coarse layers, we use the prompt "A teddy bear on tabletop" to fix the layout. Then at fine layers, we use "A teddy bear with the texture of [fleece, crochet, denim, fur] on tabletop". (Youtube link)

Changing style with prompting. At coarse layers, we use the prompt "A mansion" to fix the layout. Then at fine layers, we use "A [modern, Victorian] mansion in [sunny day, dramatic sunset]". (Youtube link)

https://mingukkang.github.io/GigaGAN/

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, Taesung Park. "Scaling up GANs for Text-to-Image Synthesis", CVPR 2023.

Text to Image

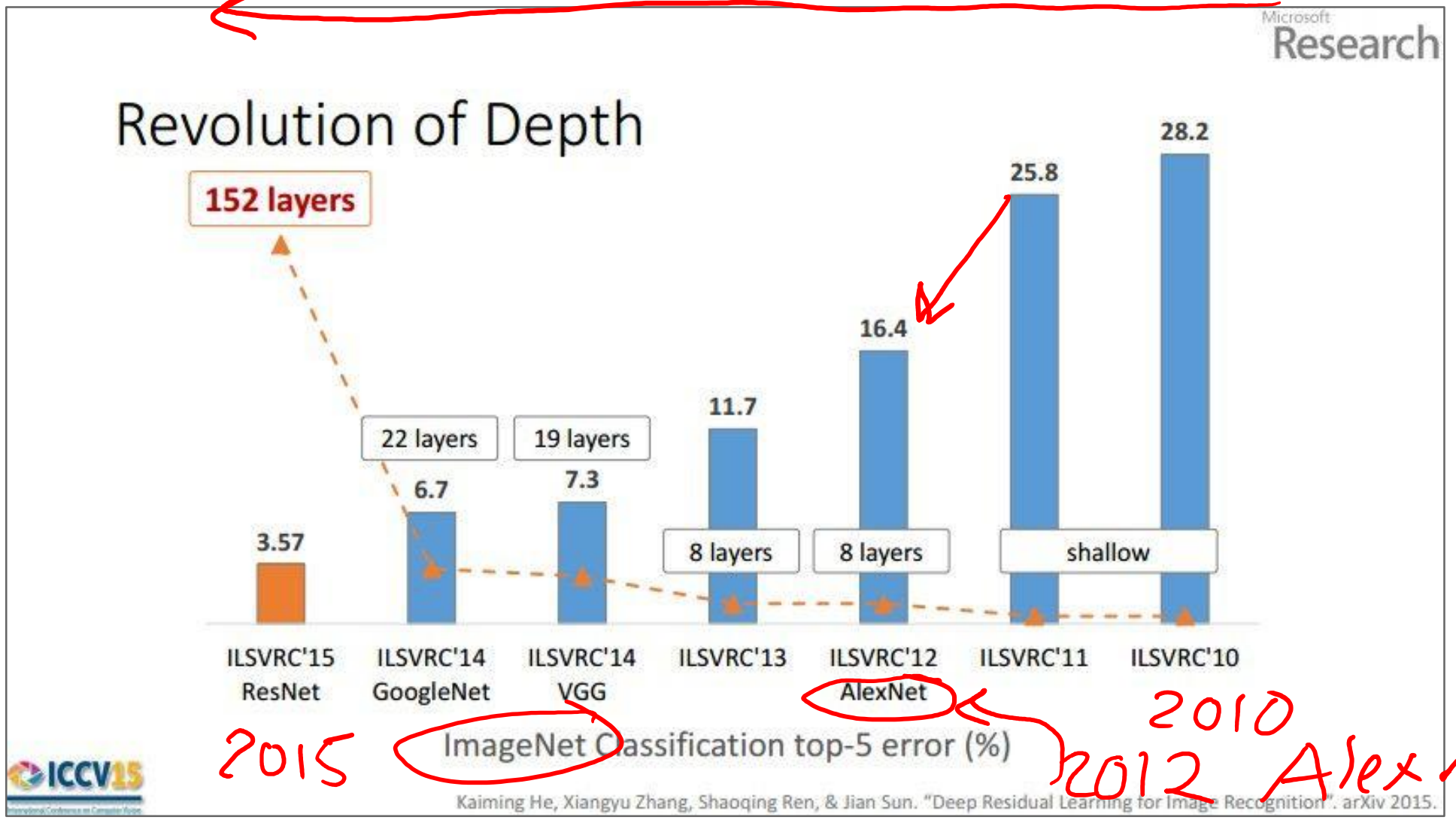
DALL-E

https://openai.com/blog/dall-e/

https://openai.com/dall-e-2/

an armchair in the shape of an avocado. . . .

CNNs for Image Recognition



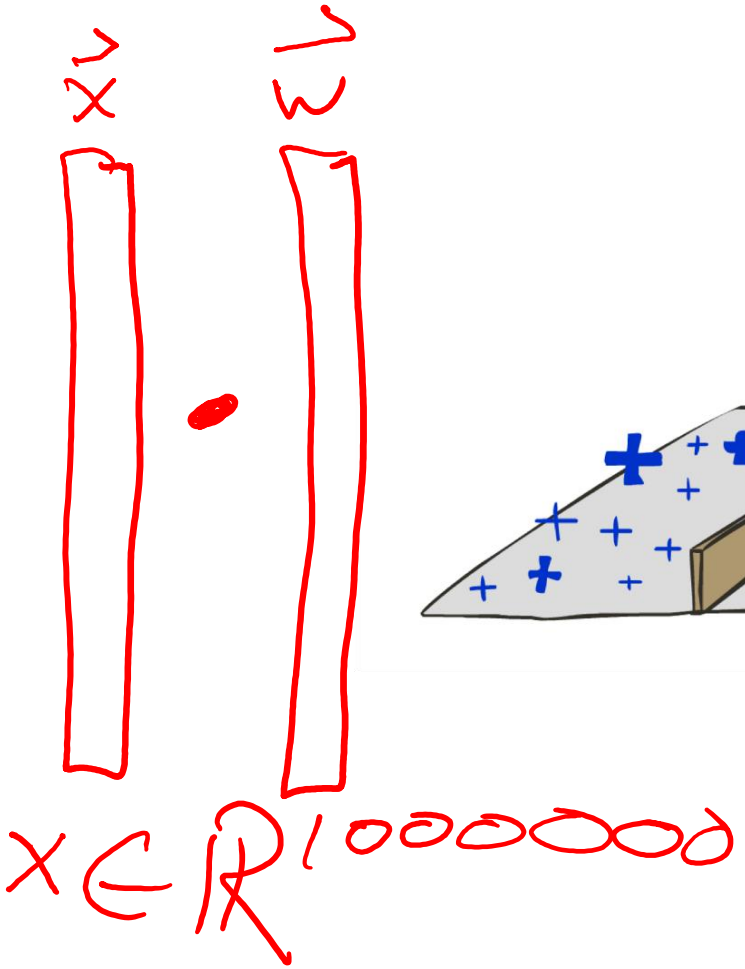
Slide from Kaiming He

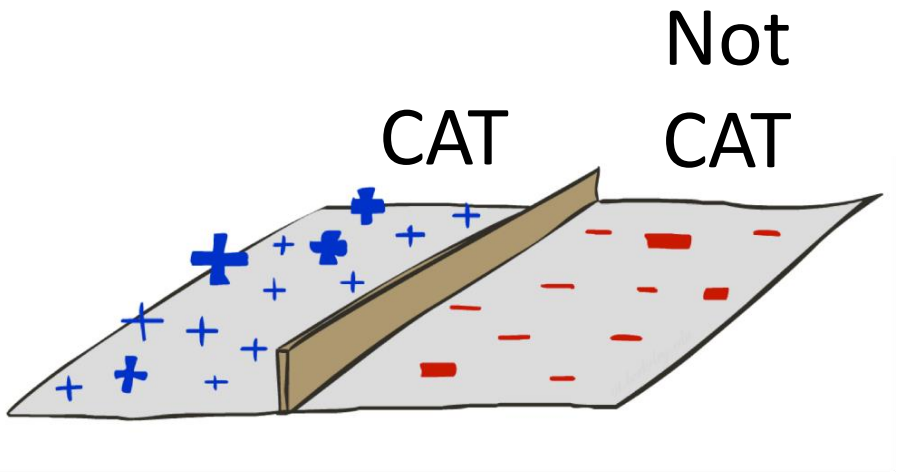
Convolutional Neural Networks

1. Measuring the current state of computer vision
2. Why convolutional neural networks
   - Old school computer vision
   - Image features and classification
3. Convolution “nuts and bolts”

Image Classification

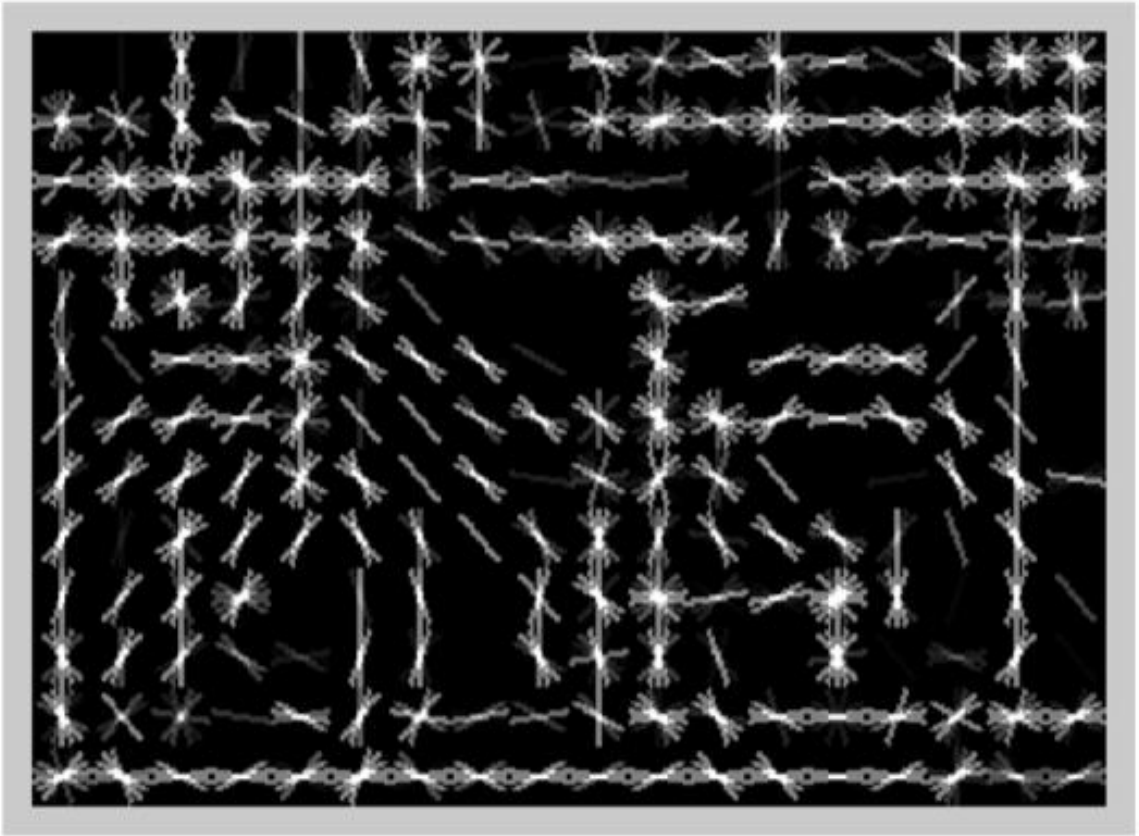
What's the problem with just directly classifying raw pixels in high dimensional space?





CAT

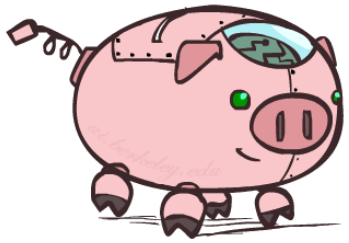
Image Classification

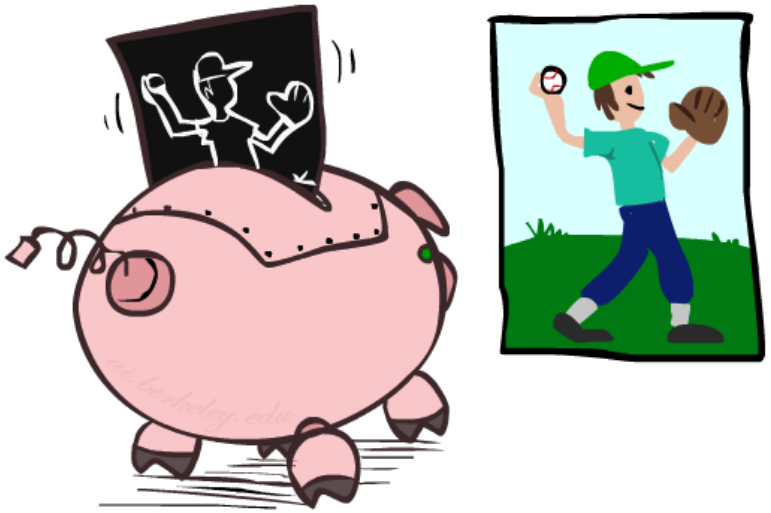


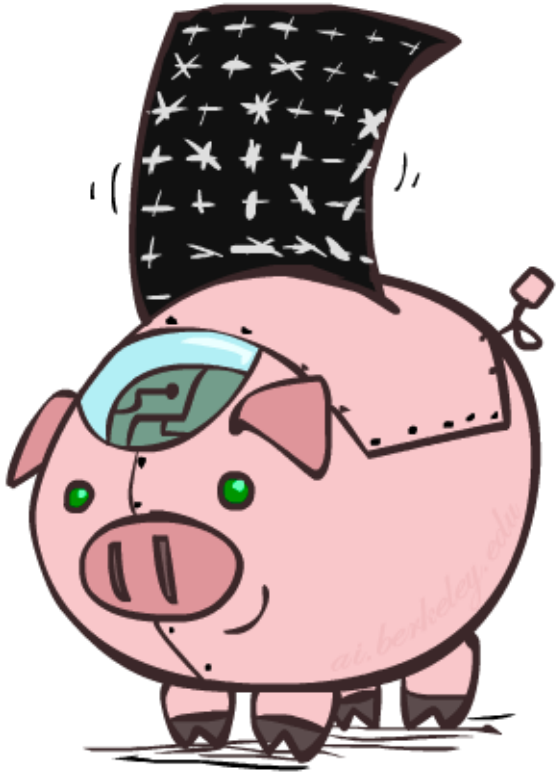
[Dalal and Triggs, 2005]

HoG Filter

HoG: Histogram of oriented gradients



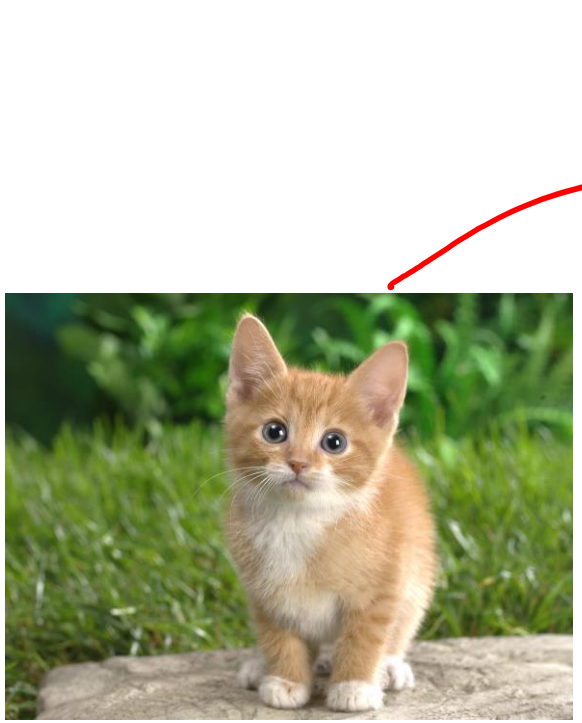


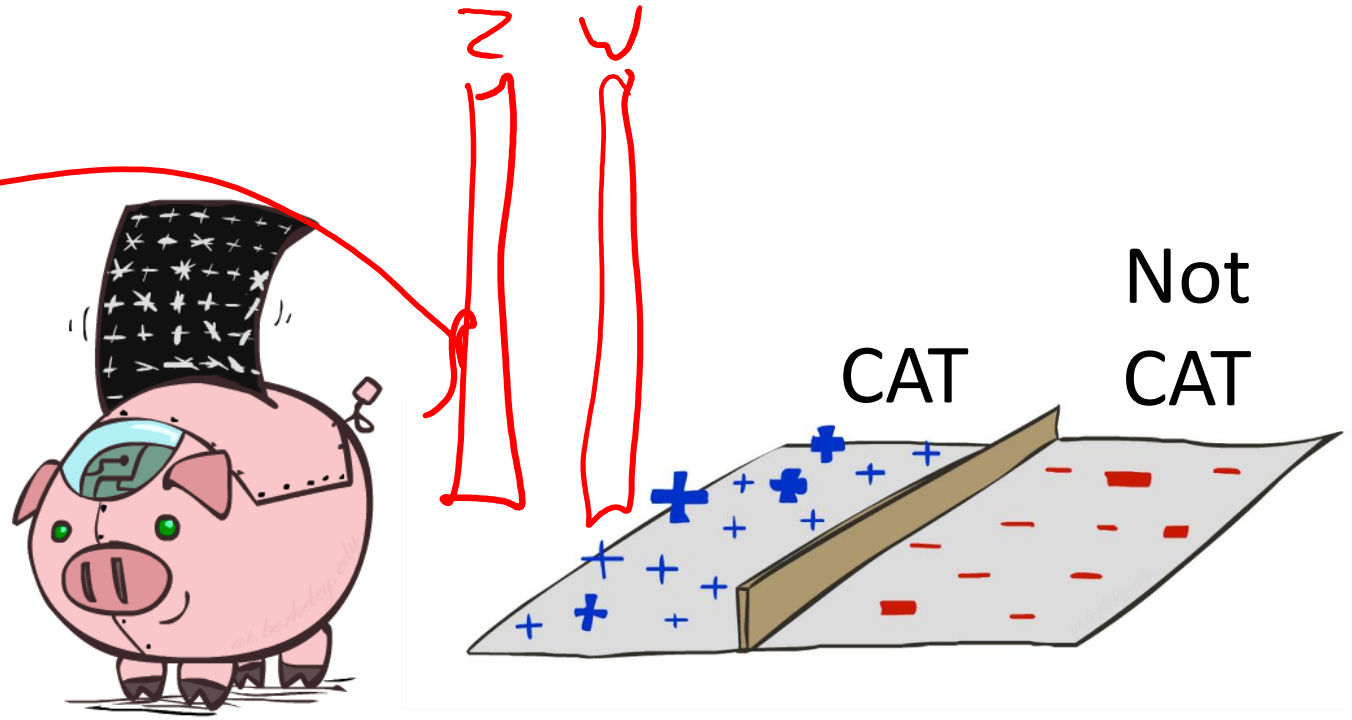


[Dalal and Triggs, 2005]

Image Classification

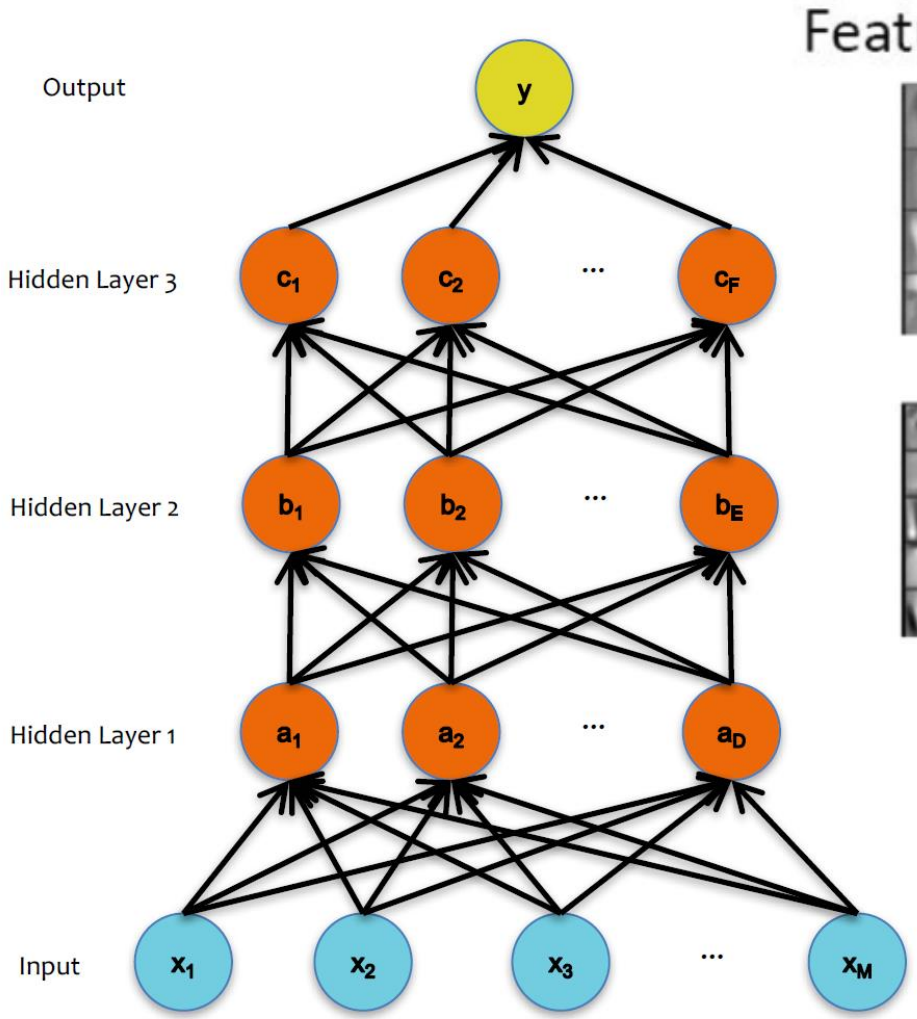
HOG features passed to a linear classifier (logistic regression / SVM)

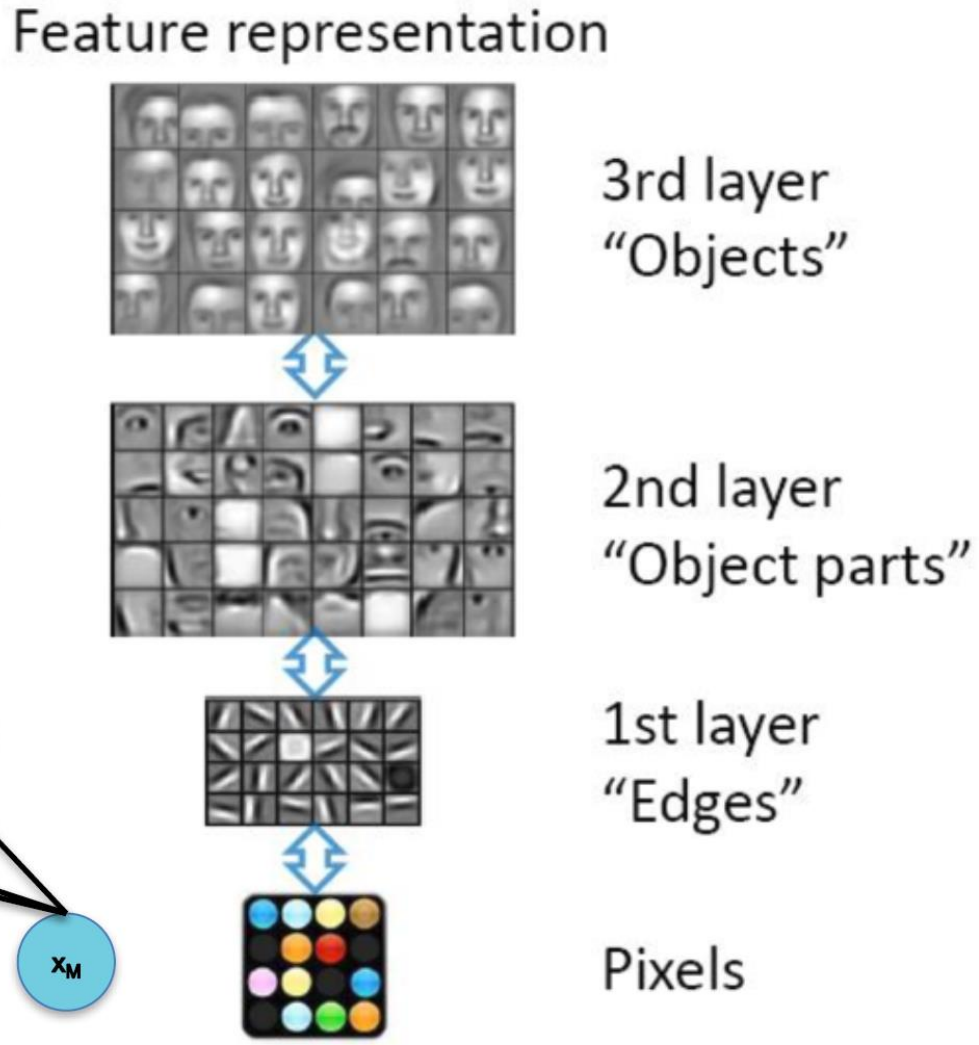




CAT

Classification: Learning Features



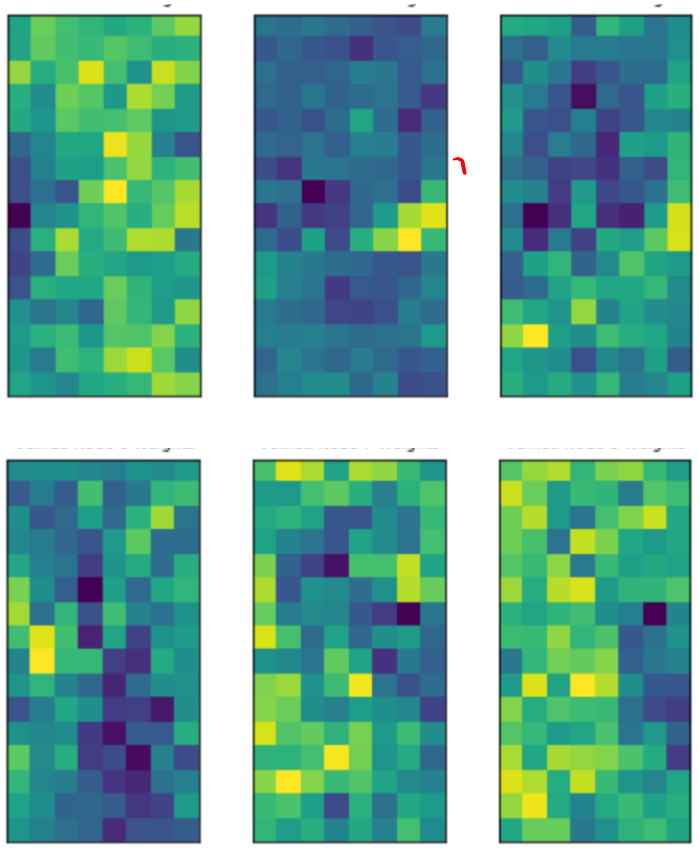


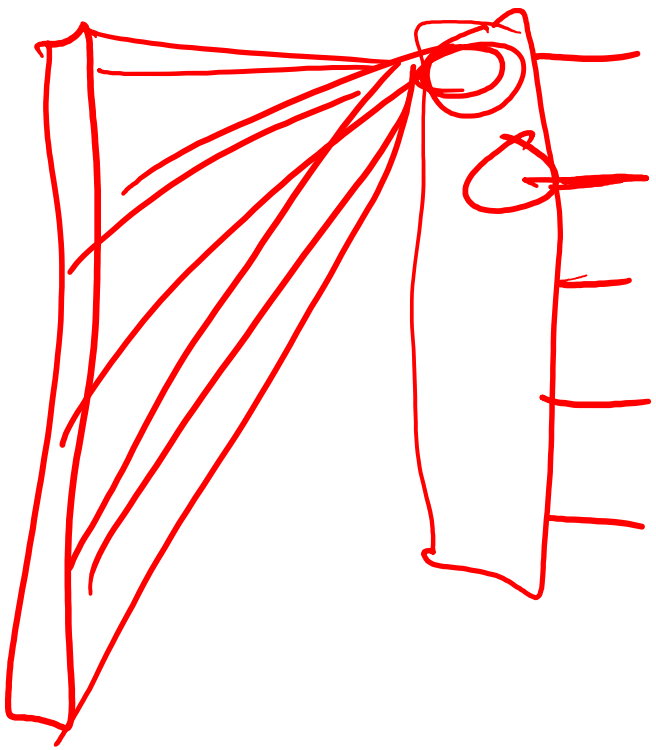
Example from Honglak Lee (NIPS 2010)

Classification: Learning Features

Input features

Weights from fully-connected layer

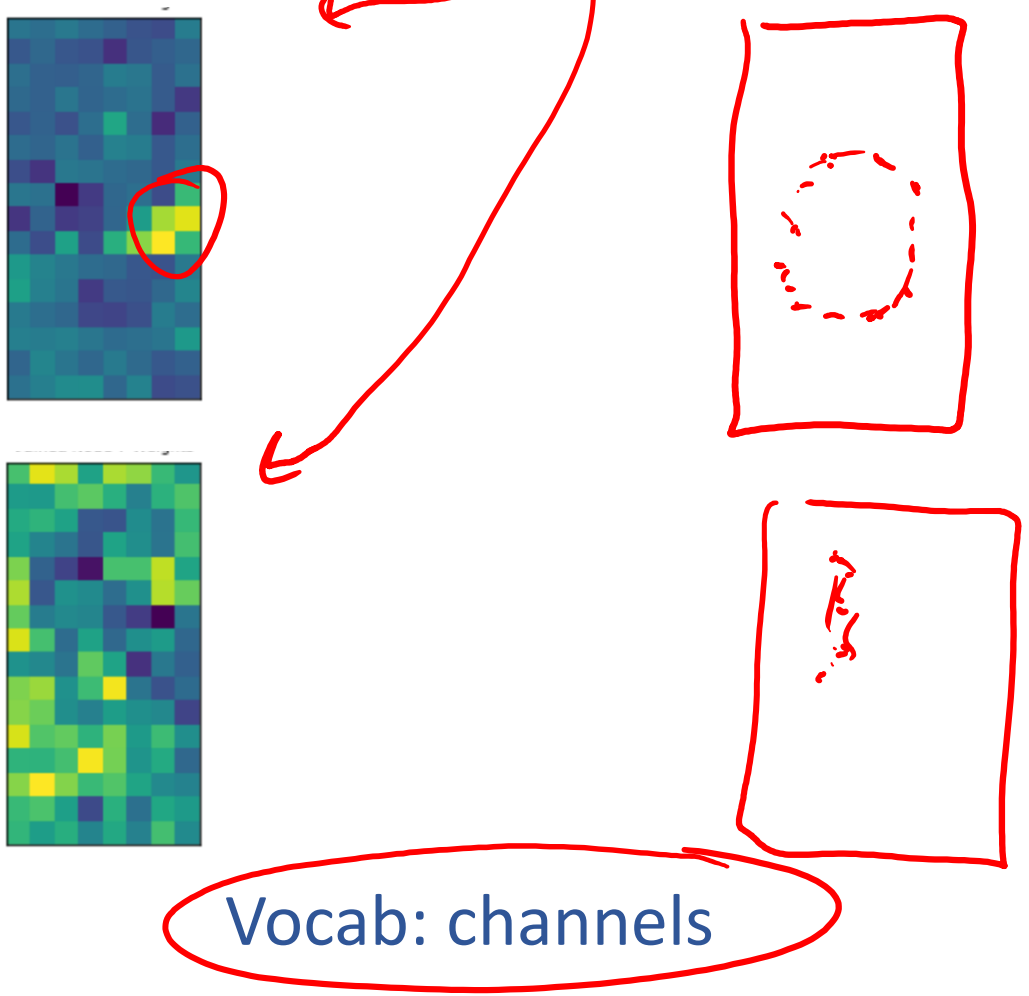




Classification: Learning Features

Input features

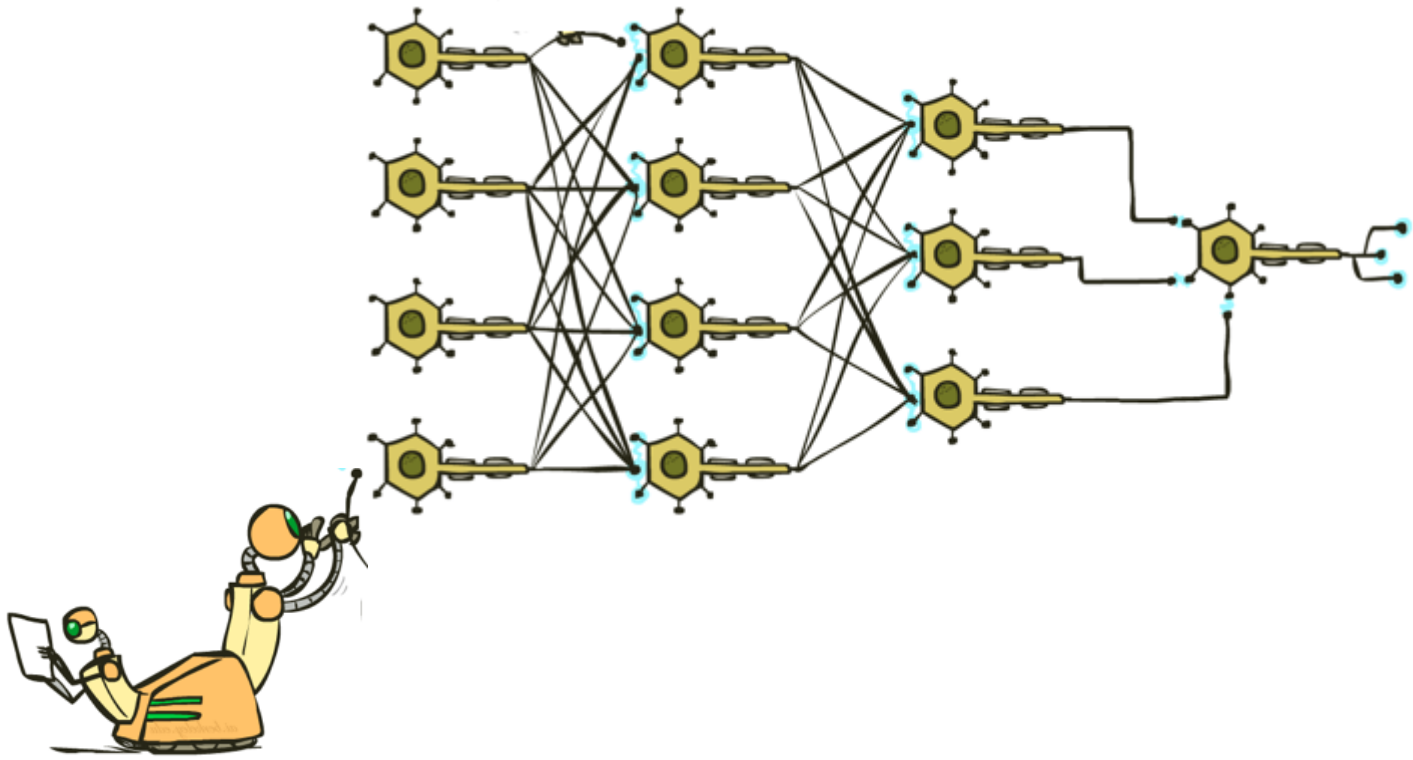
Weights from
fully-connected layer



Vocab: channels

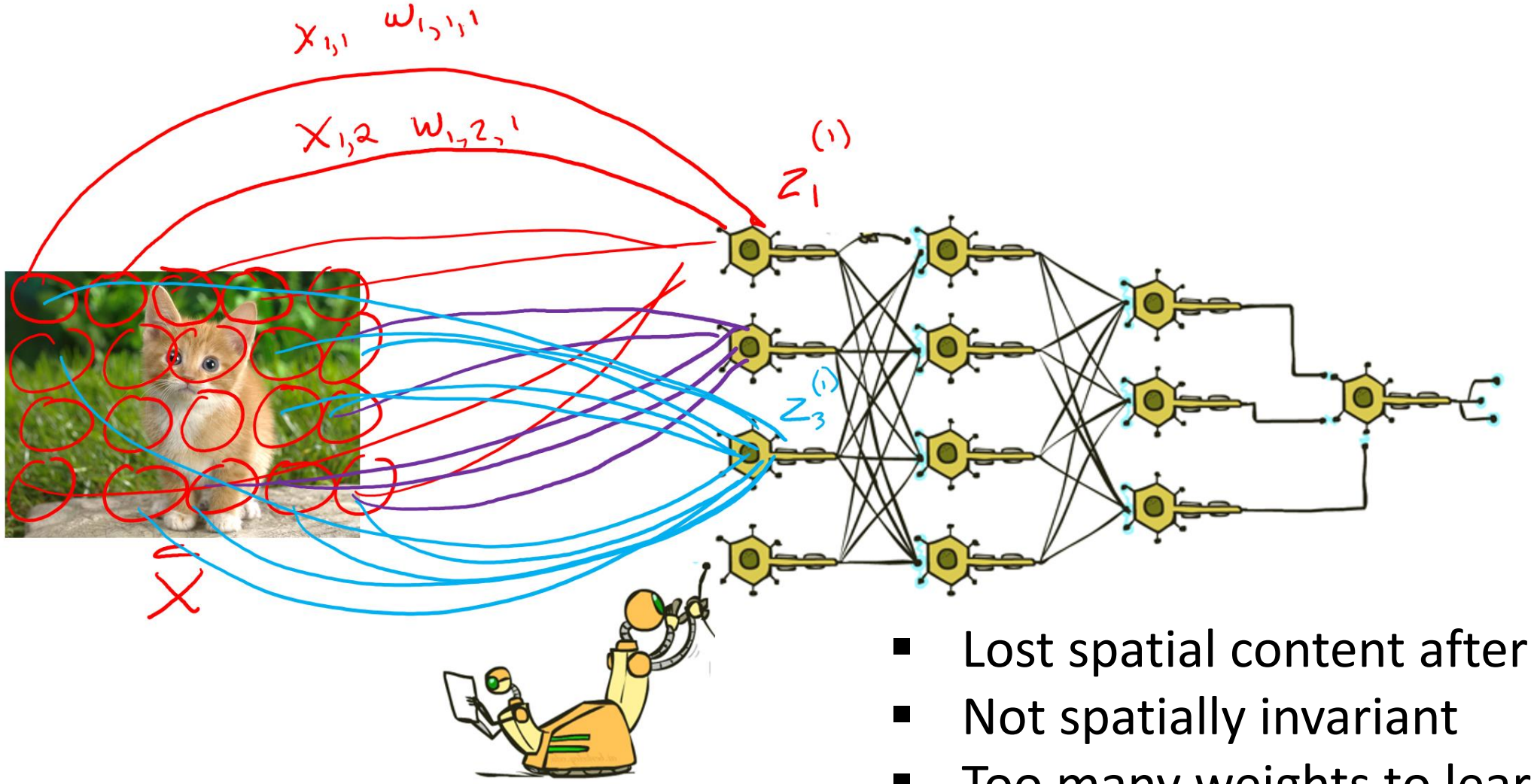
Classification: Deep Learning

Fully connected neural network?



CAT

Classification: Deep Learning



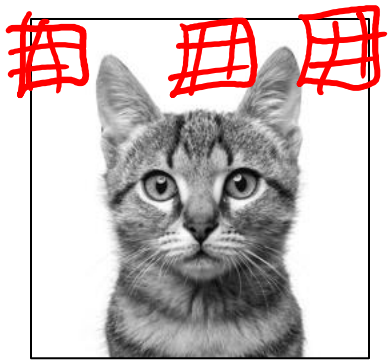
- Lost spatial content after first layer
- Not spatially invariant
- Too many weights to learn

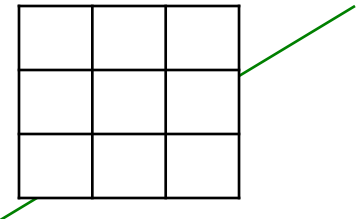
Convolutional Neural Networks

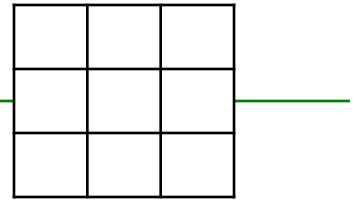
filter
kernel
weights

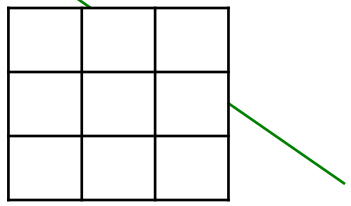
3 channels

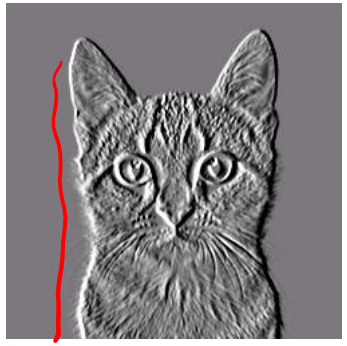
Convolution

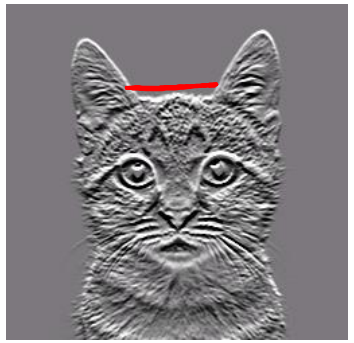


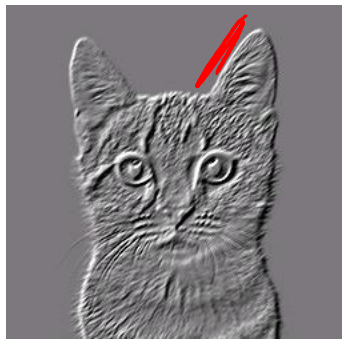


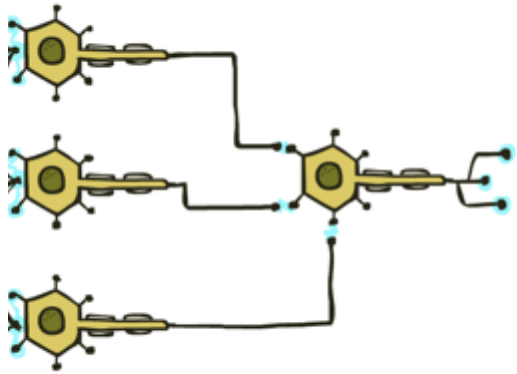






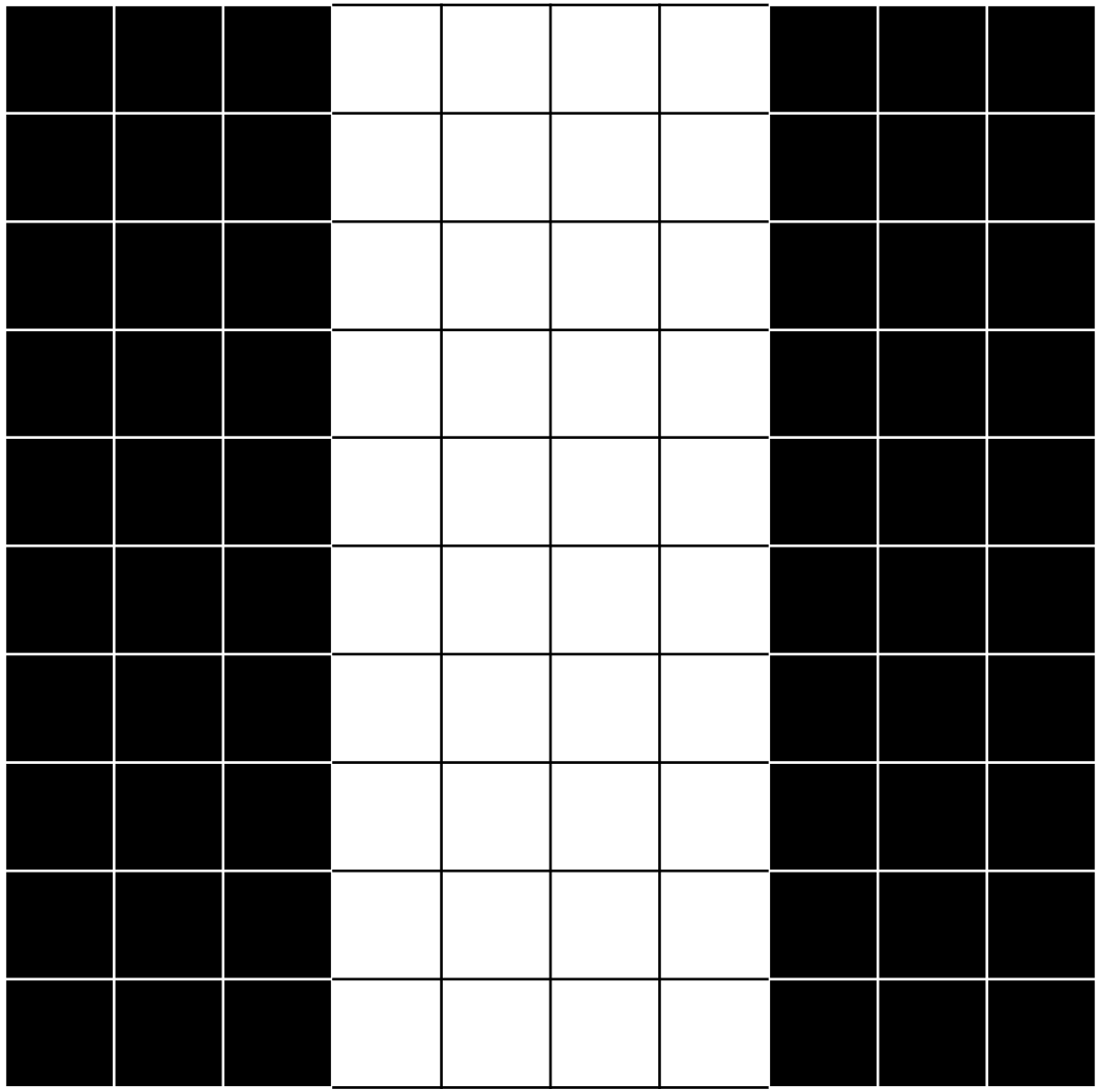






CAT

Convolution



-1	0	1
-1	0	1
-1	0	1

Convolution

X

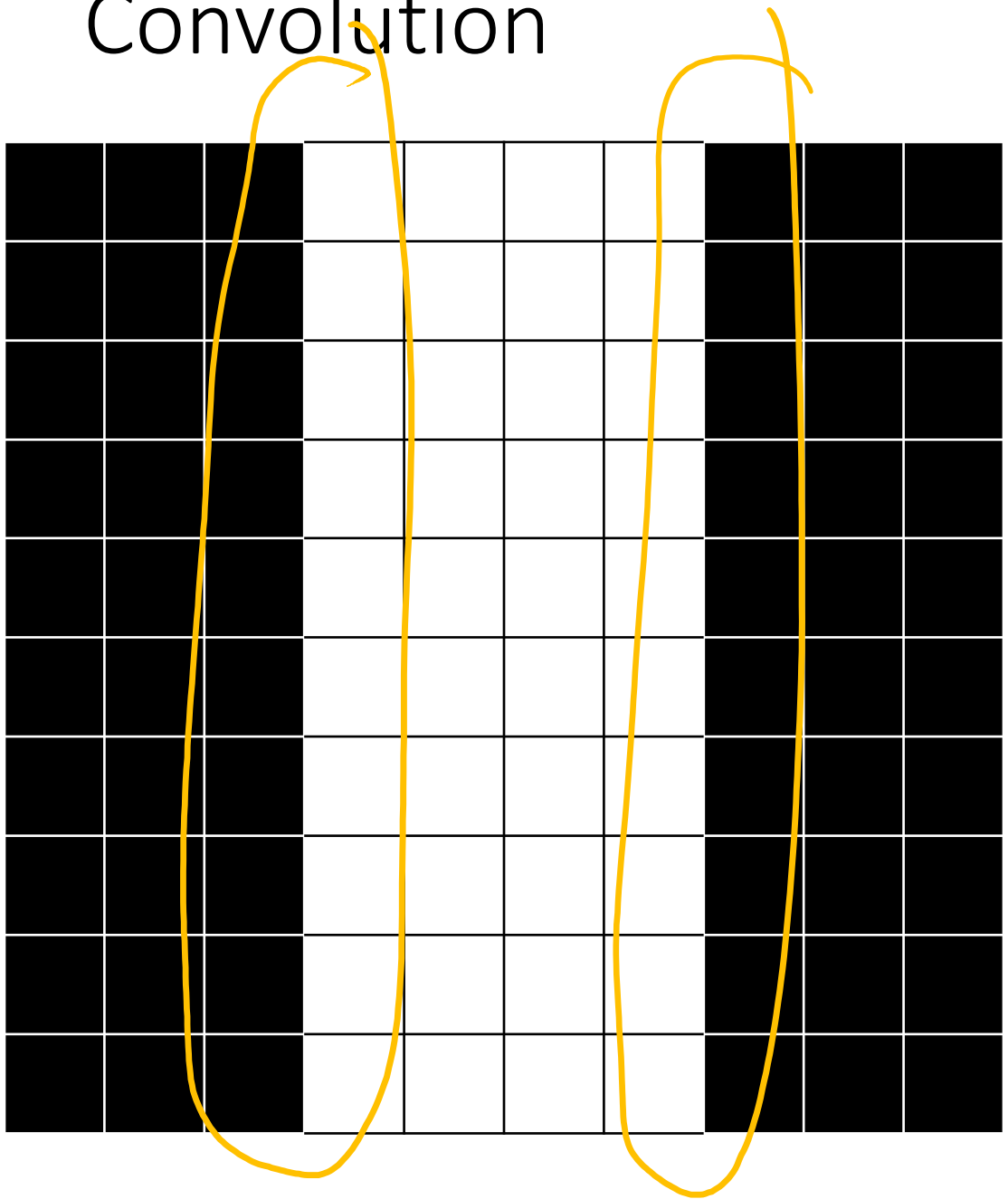
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0

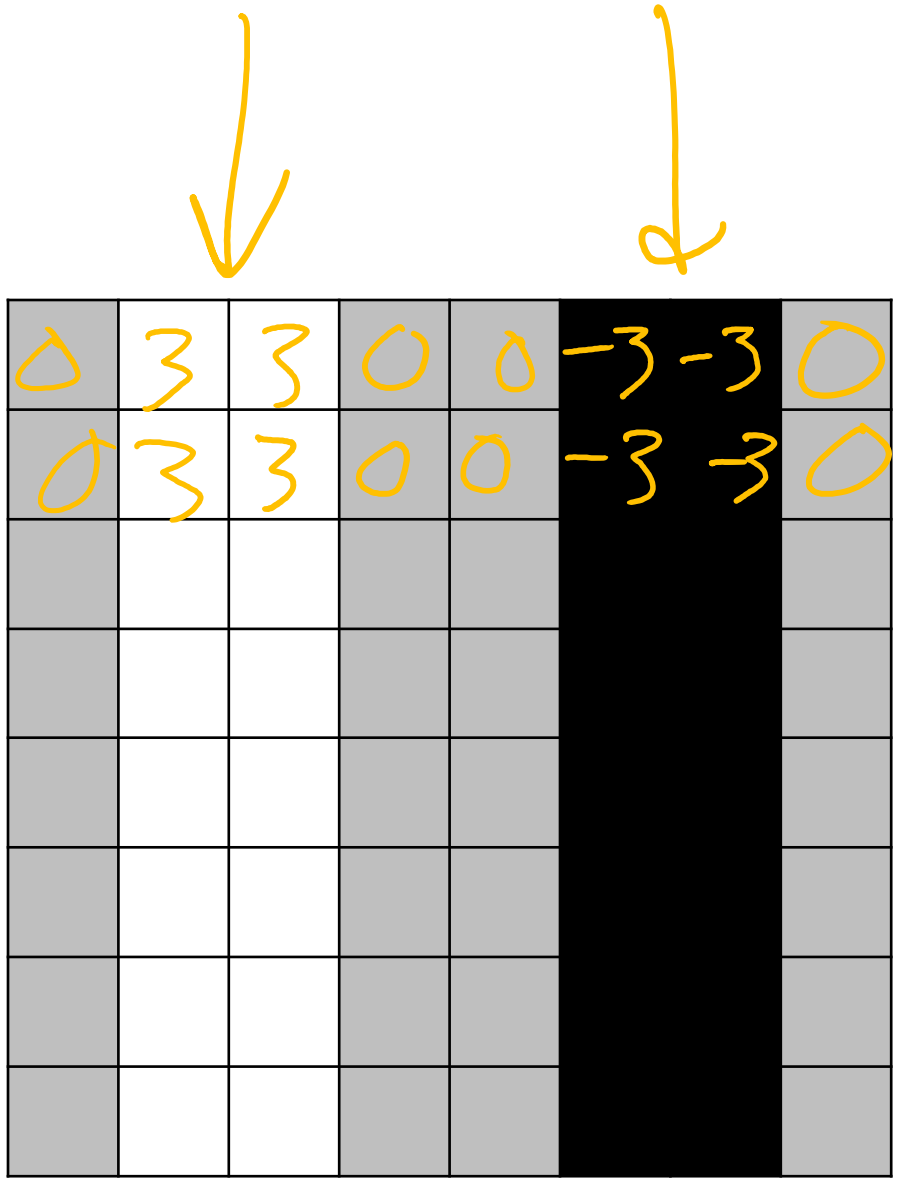
0	3	3	0
0			

W

-1	0	1
-1	0	1
-1	0	1

Convolution





-1	0	1
-1	0	1
-1	0	1

Convolution

Signal processing definition

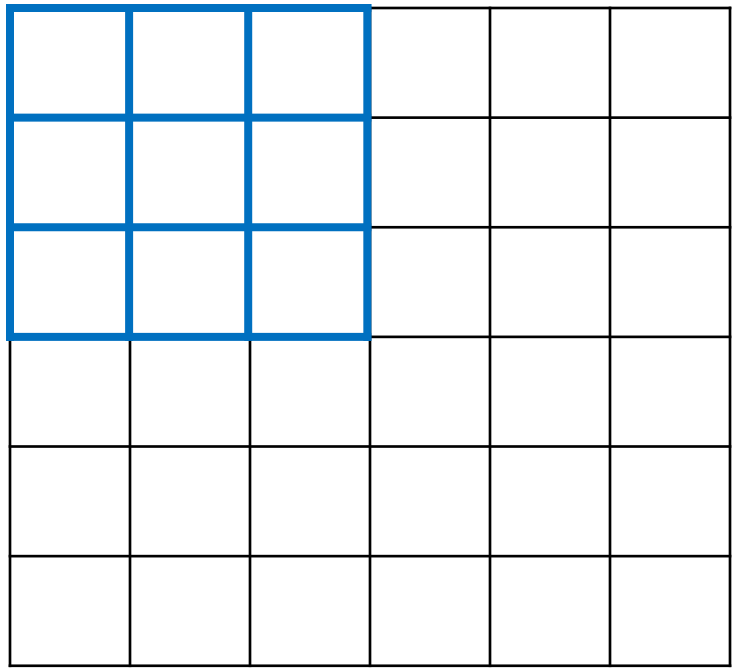
$$z[i,j] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} x[i-u, j-v] \cdot w[u, v]$$

-1	0	1
-2	0	2
-1	0	1

Relaxed definition

- Drop infinity; don't flip kernel

$$z[i,j] = \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} x[i+u, j+v] \cdot w[u, v]$$



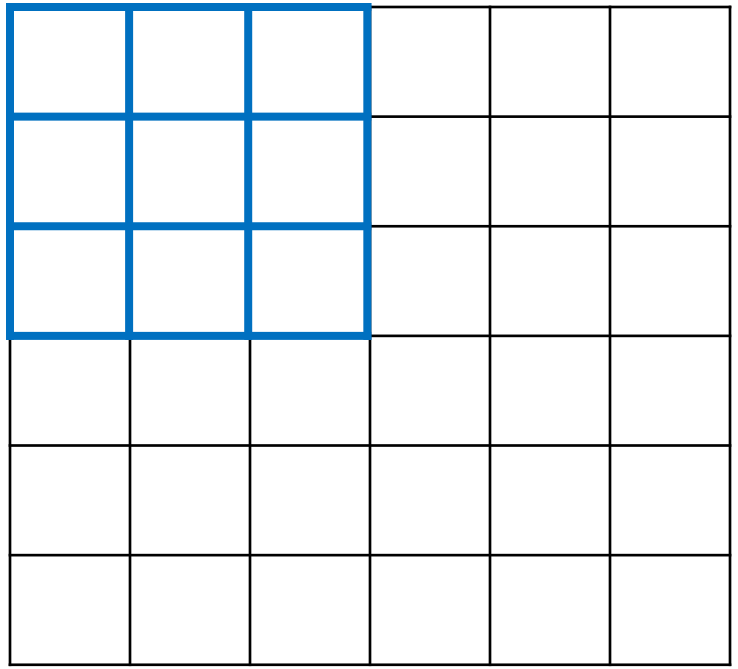
Convolution

Relaxed definition

$$z[i,j] = \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} x[i+u, j+v] \cdot w[u, v]$$

-1	0	1
-2	0	2
-1	0	1

```
for i in range(0, im_width - K + 1):
  for j in range(0, im_height - K):
    im_out[i,j] = 0
    for u in range(0, K):
      for v in range(0, K):
        im_out[i,j] += im[i+u, j+v] * kernel[u,v]
```



GPU!!

Convolution: Padding

0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0

0	2	2	0	0	-2	-2	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	3	3	0	0	-3	-3	0
0	2	2	0	0	-2	-2	0

Exercise: Which kernel goes with which output image?

Input

K1

-1	0	1
-2	0	2
-1	0	1

K2

-1	-2	-1
0	0	0
1	2	1

K3

0	0	-1	0
0	-2	0	1
-1	0	2	0
0	1	0	0

Im1

Im2

Im3

Exercise: Which kernel goes with which output image?

Input

K1

-1	0	1
-2	0	2
-1	0	1

K2

-1	-2	-1
0	0	0
1	2	1

K3

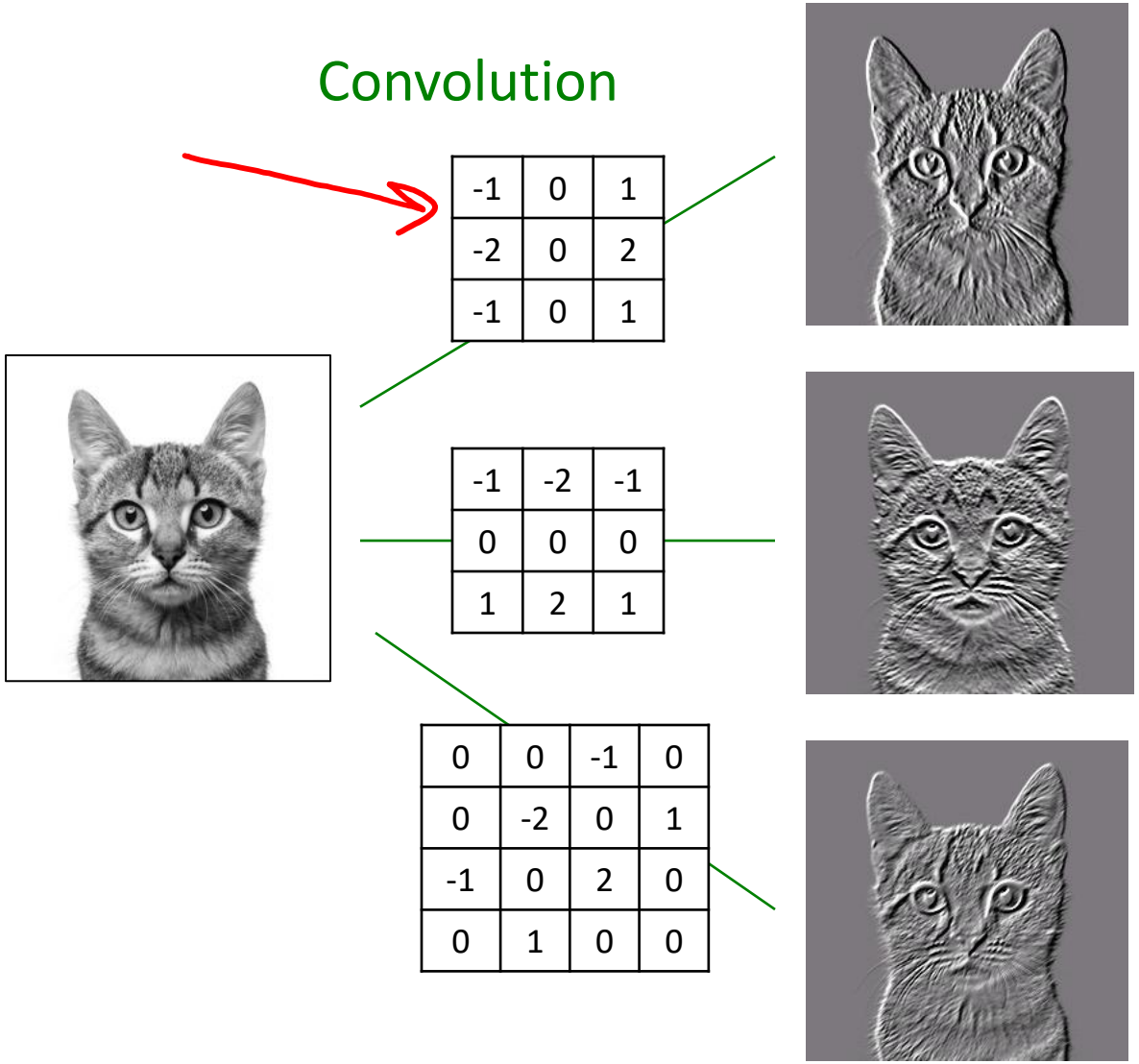
0	0	-1	0
0	-2	0	1
-1	0	2	0
0	1	0	0

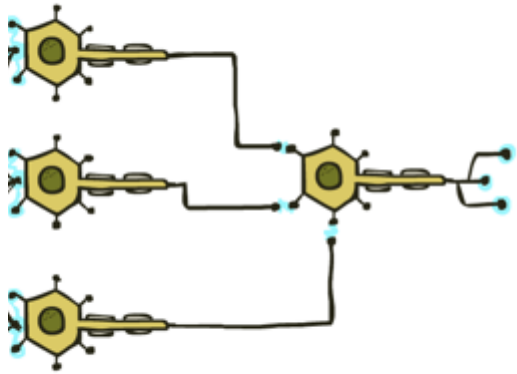
Im1

Im2

Im3

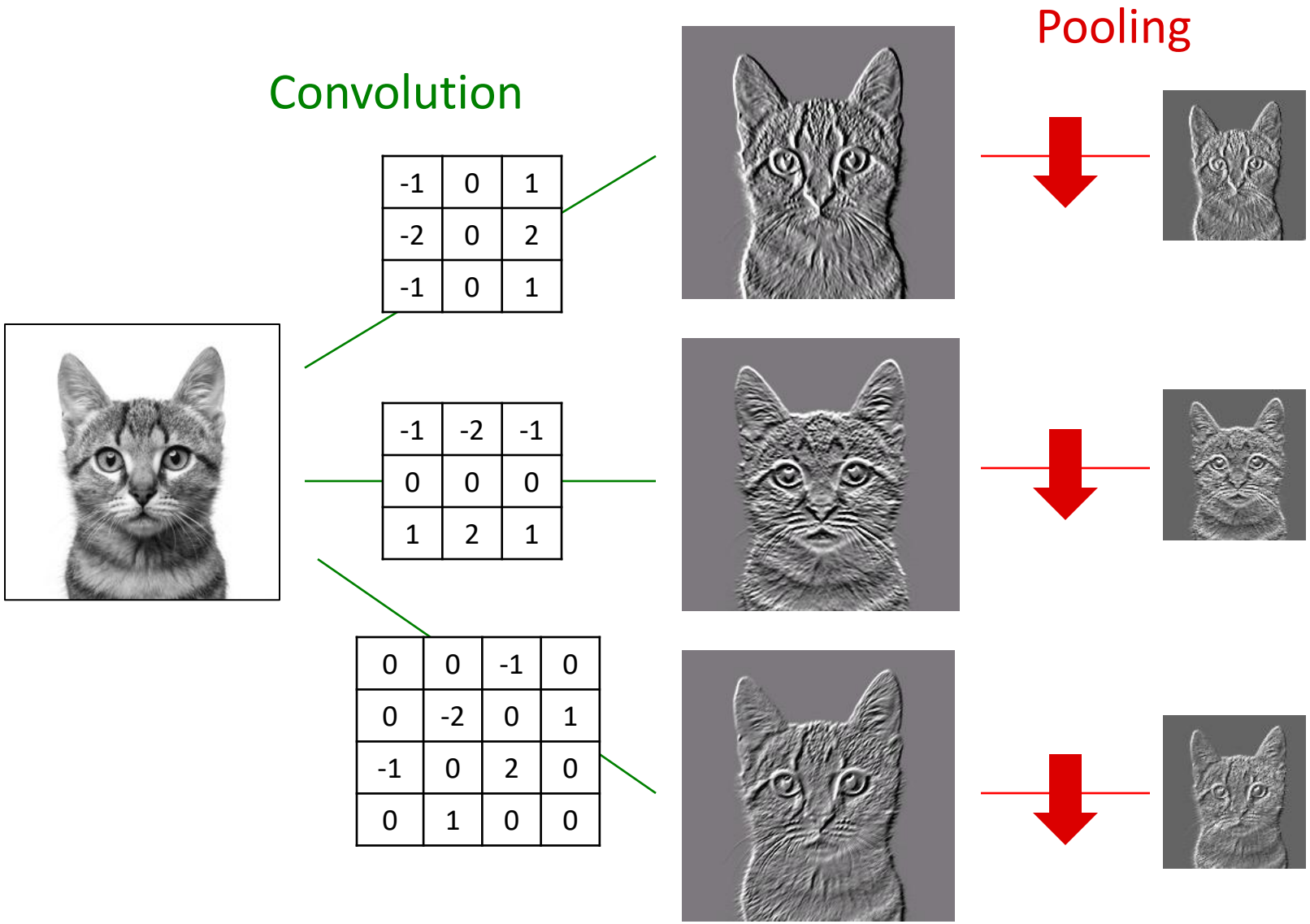
Convolutional Neural Networks





CAT

Convolutional Neural Networks



Convolution: Stride=2

0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0

.25	.25
.25	.25

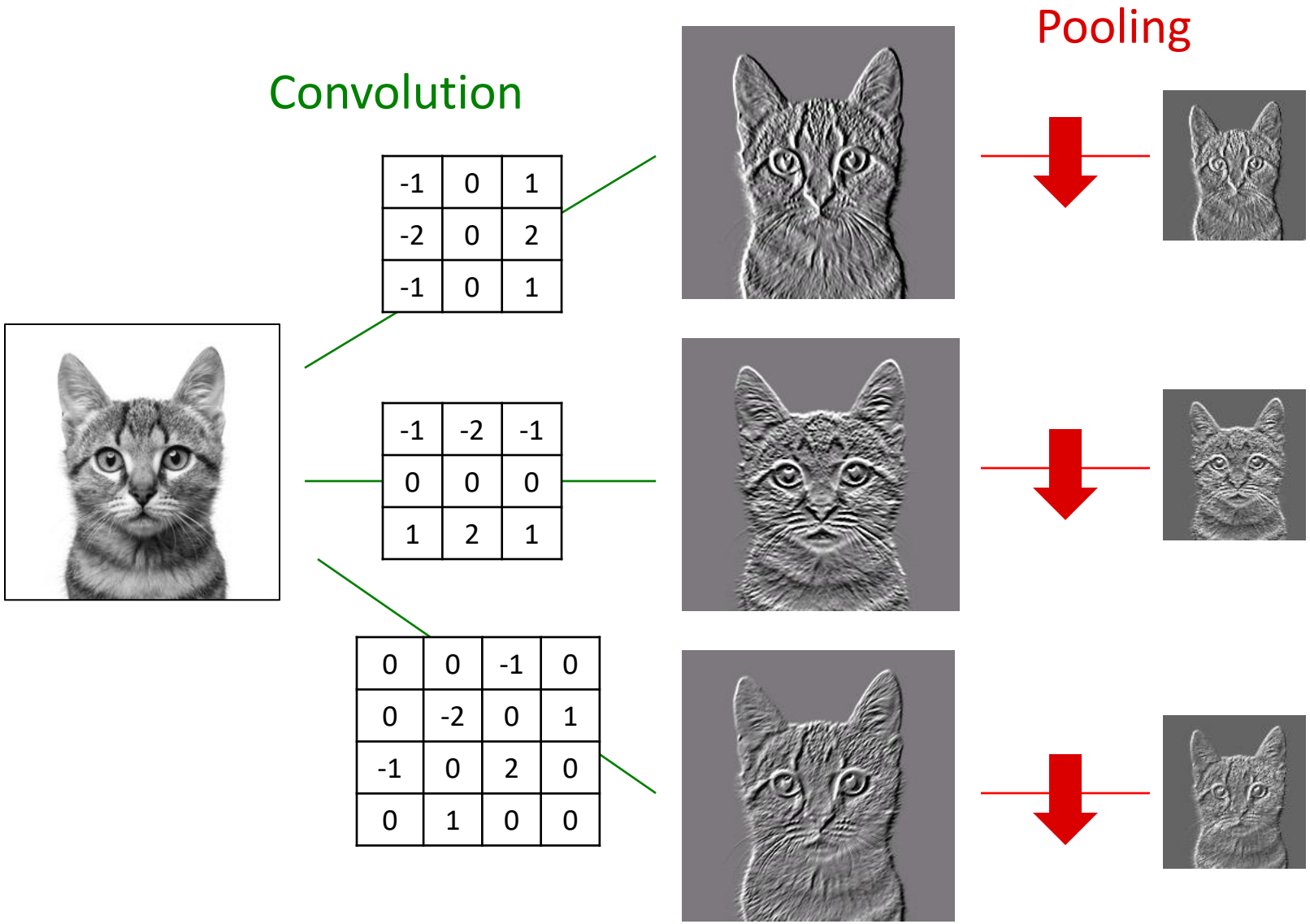
Stride: Max Pooling

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

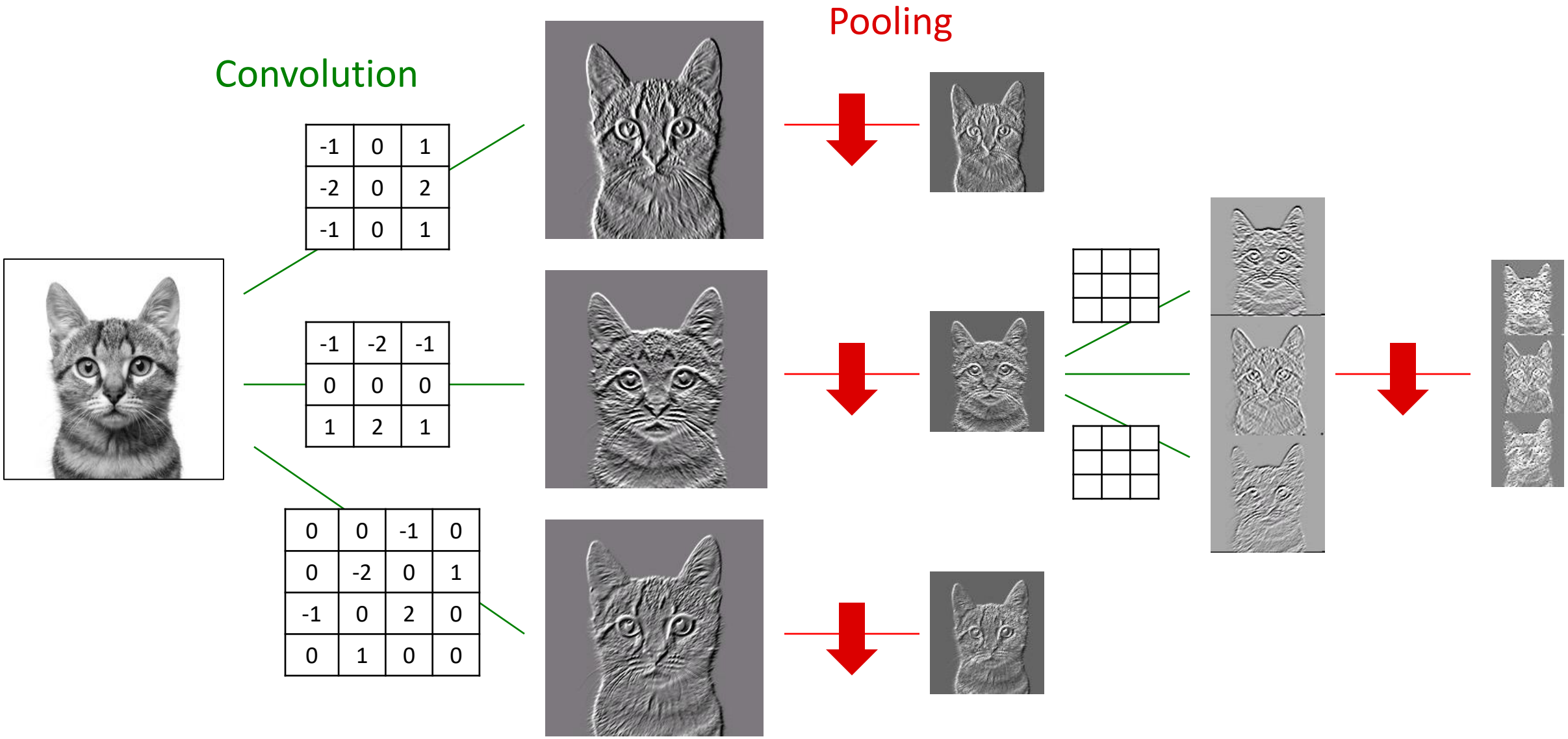
max pool with 2x2 filters
and stride 2

6	8
3	4

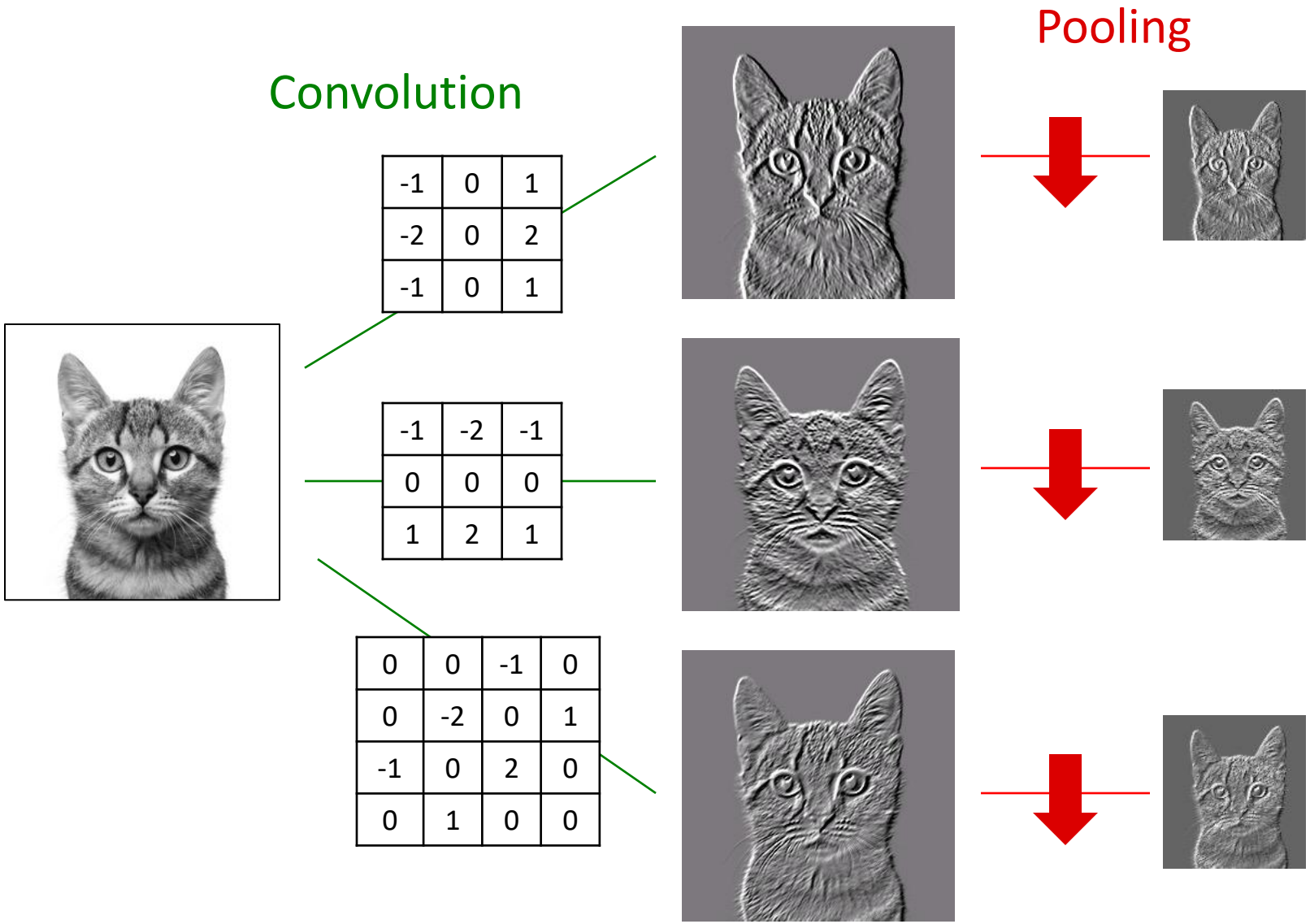
Convolutional Neural Networks



Convolutional Neural Networks



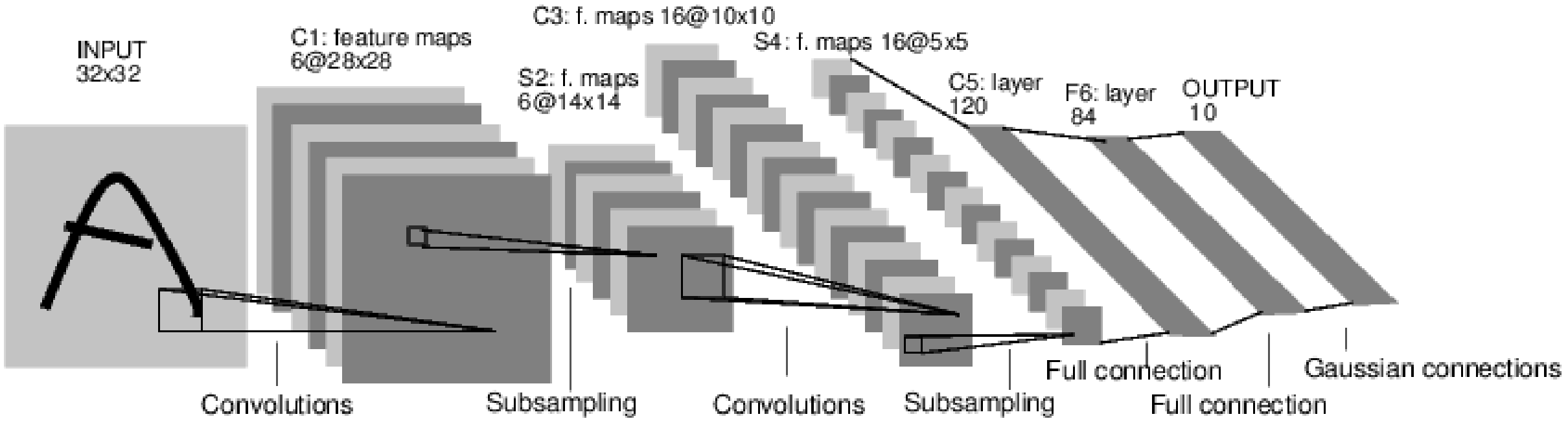
Convolutional Neural Networks



Convolutional Neural Networks

Lenet5 – Lecun, et al, 1998

- Convnets for digit recognition

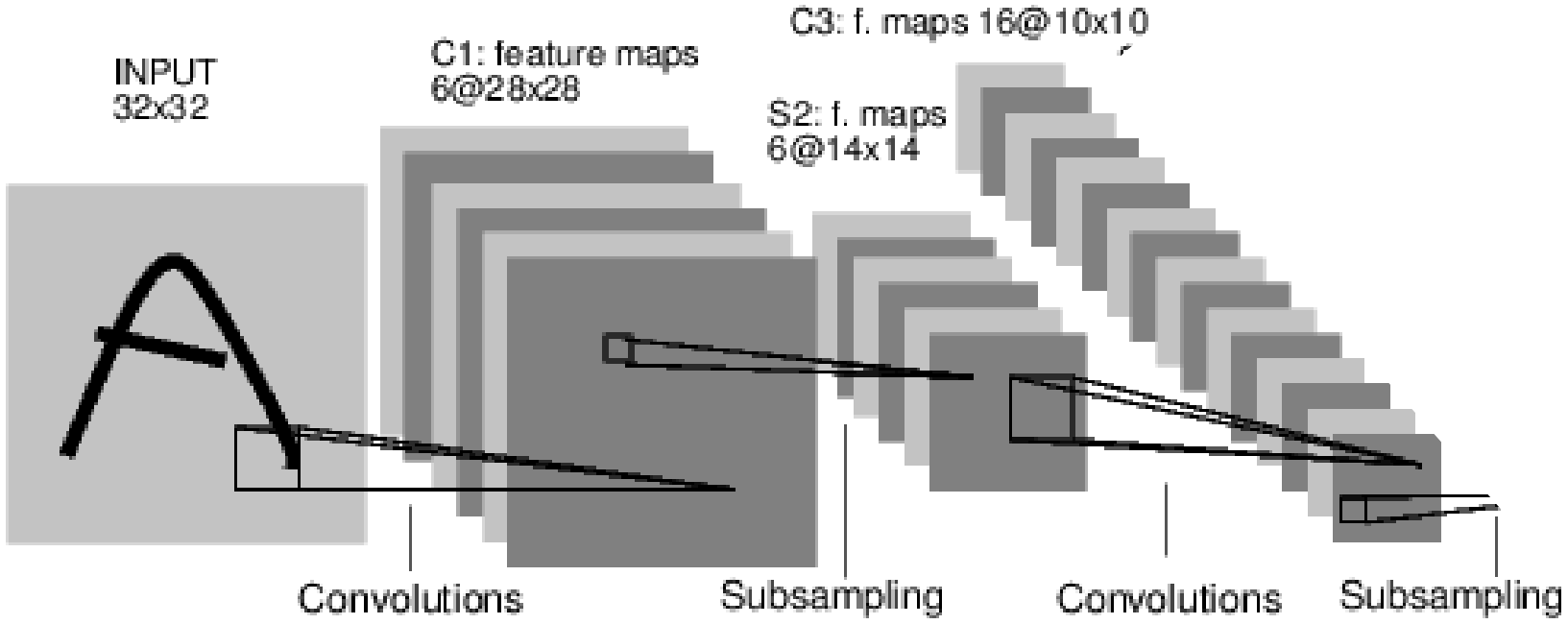


LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

Question:

How big many convolutional weights between S2 and C3?

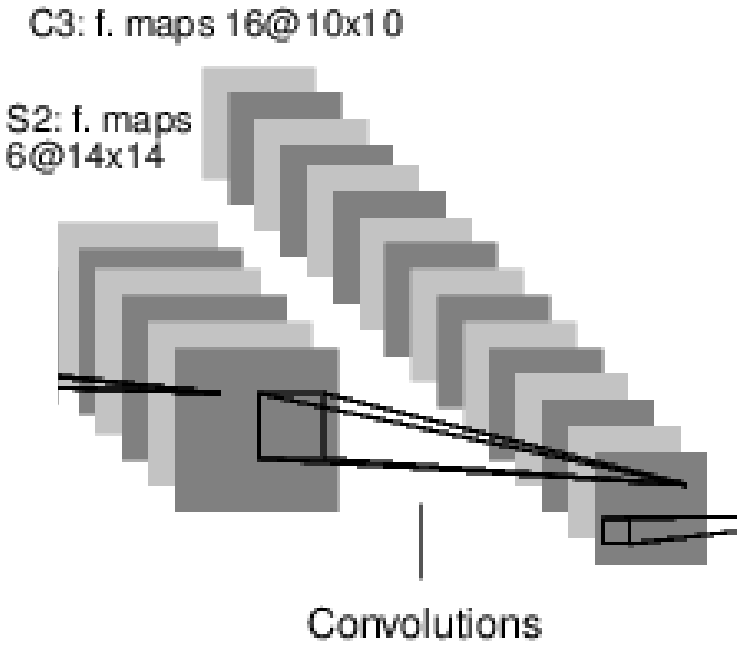
- S2: 6 channels @14x14
- Conv: 5x5, pad=0, stride=1
- C3: 16 channels @ 10x10



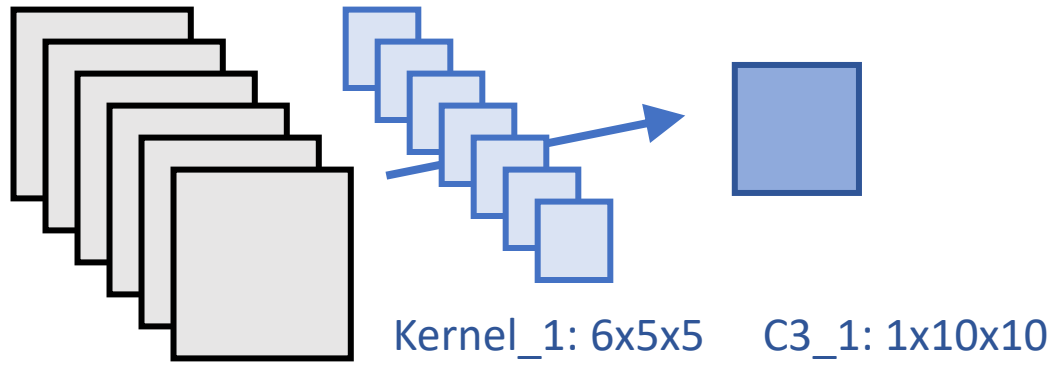
Question:

How big many convolutional weights between S2 and C3?

- S2: 6 channels @14x14
- Conv: 5x5, pad=0, stride=1
- C3: 16 channels @ 10x10



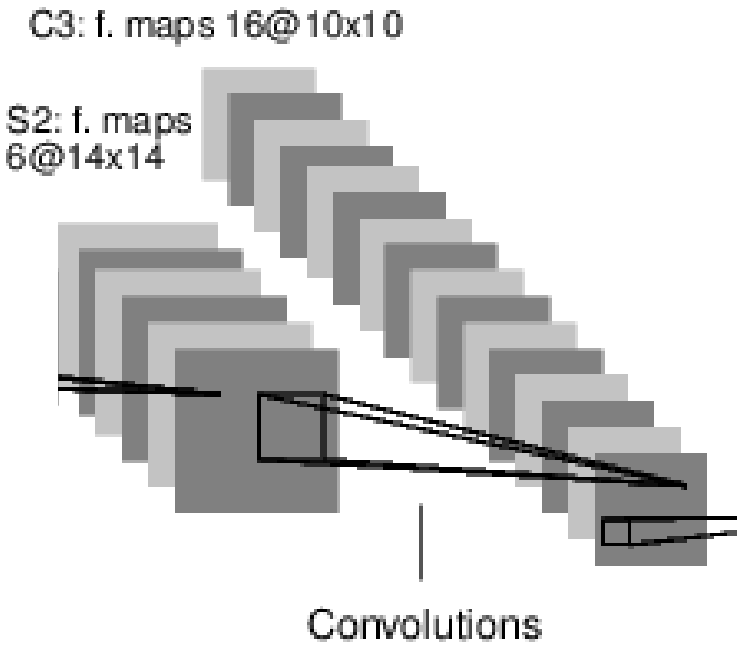
One image in C3 is actually the result of a 3D convolution



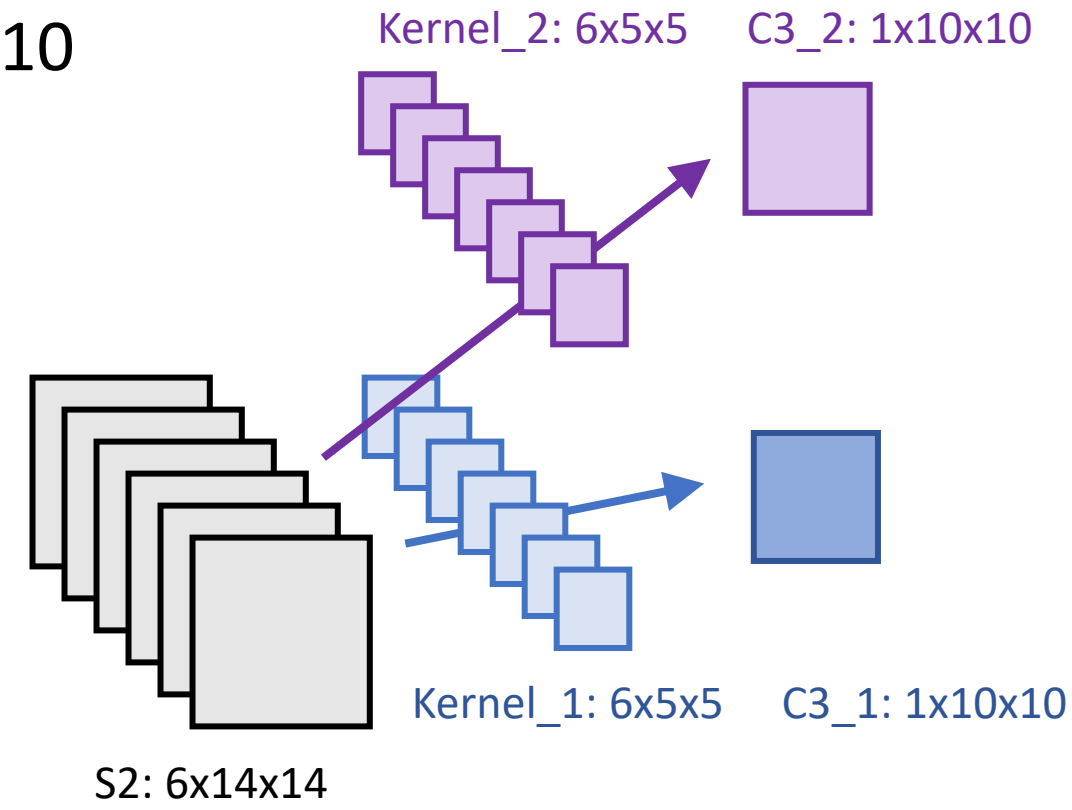
Question:

How big many convolutional weights between S2 and C3?

- S2: 6 channels @14x14
- Conv: 5x5, pad=0, stride=1
- C3: 16 channels @ 10x10



Each image in C3 convolved S2
convolved with a different 3D kernel

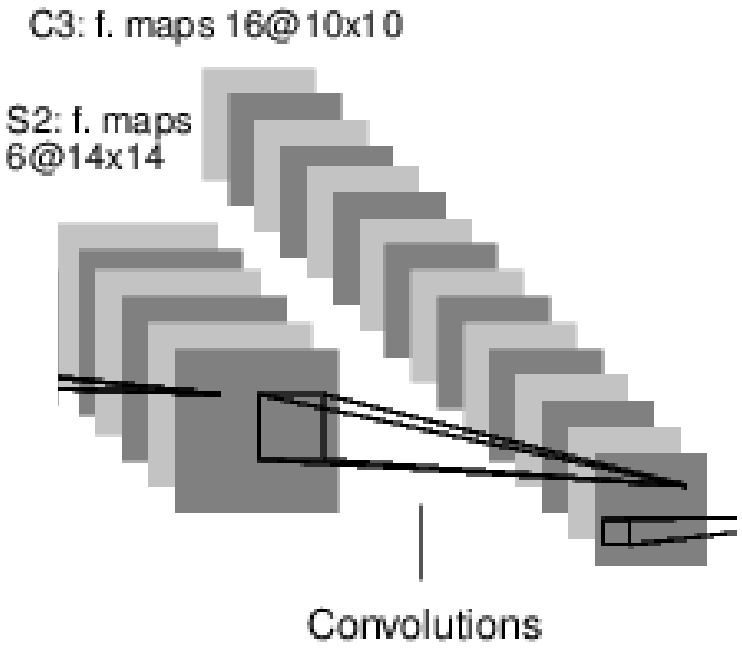


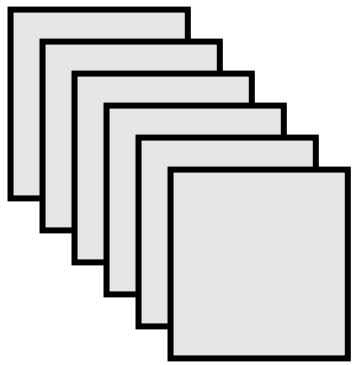
Question:

How big many convolutional weights between S2 and C3?

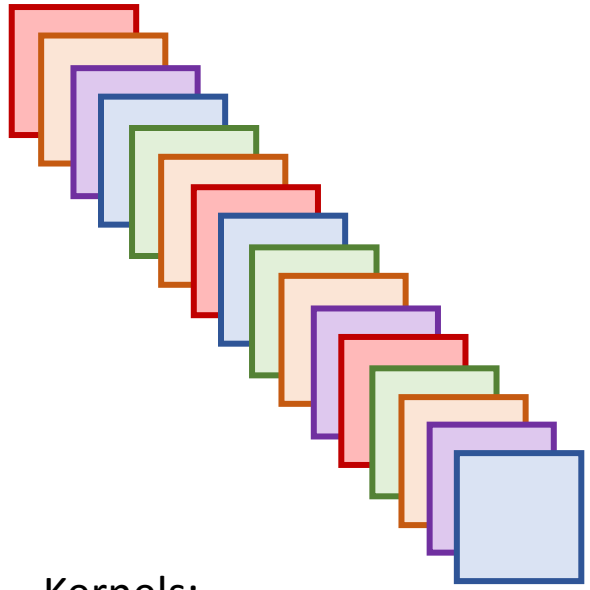
- S2: 6 channels @14x14
- Conv: 5x5, pad=0, stride=1
- C3: 16 channels @ 10x10

The 16 images in C3 are the result of doing 16 3D convolutions of S2 with 16 different 6x5x5 kernels. Assuming no bias term, this is 16x6x5x5 weights!





S2: 6x14x14



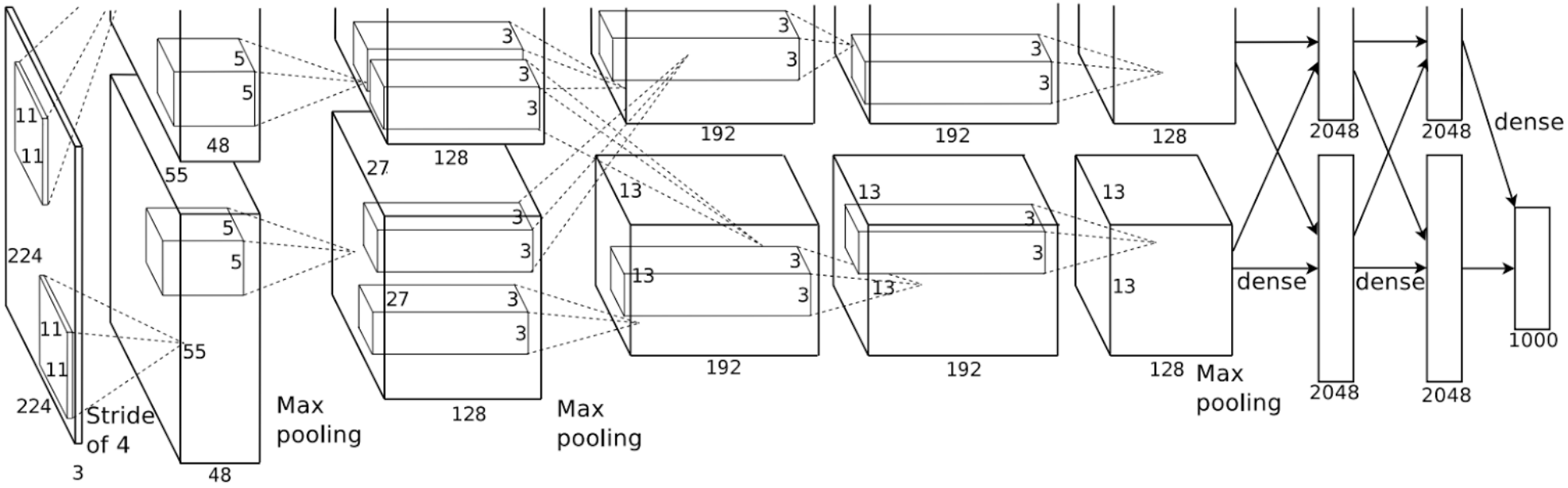
Kernels:
16@6x5x5

C3: 16@10x10

Convolutional Neural Networks

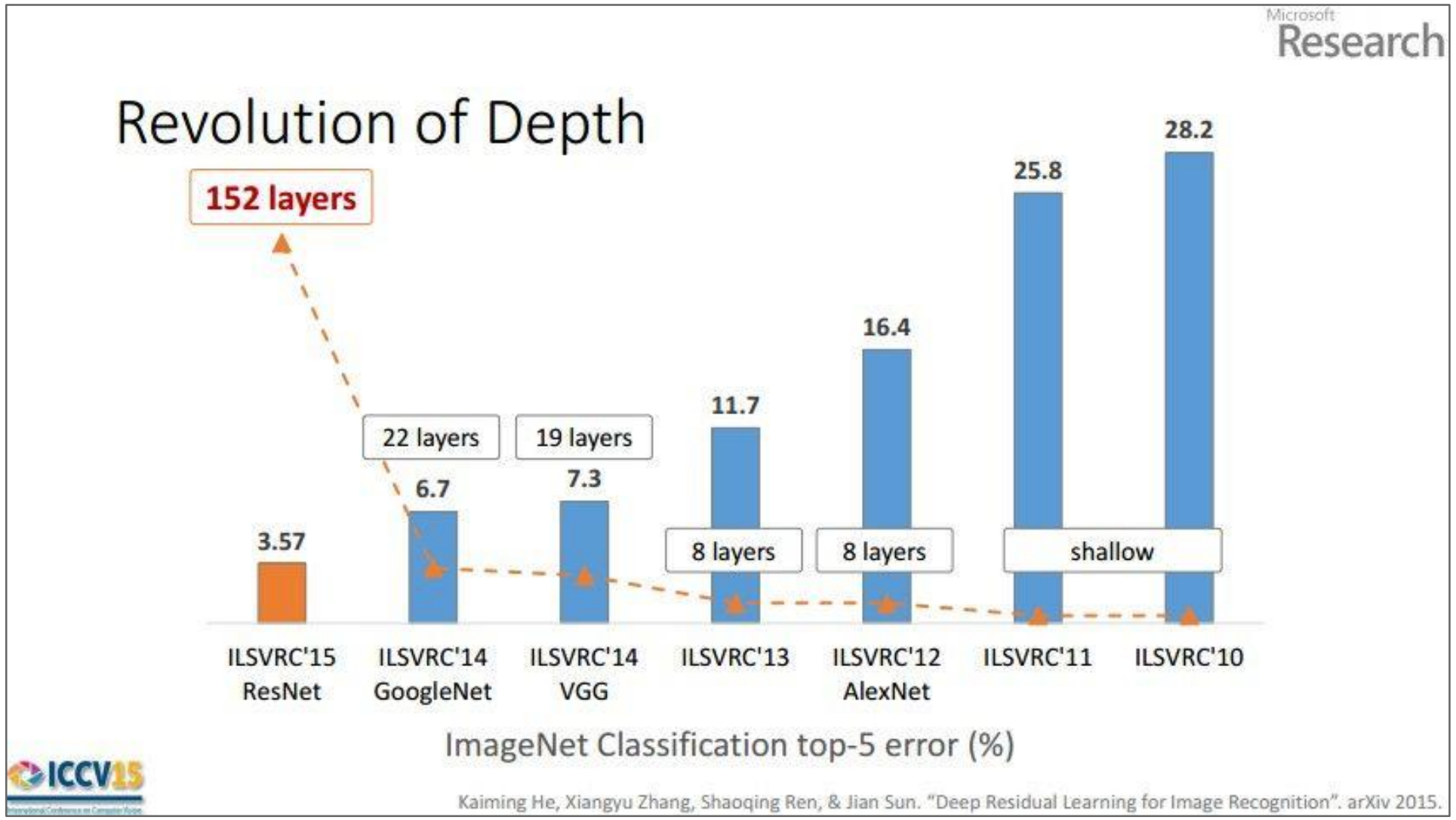
Alexnet – Krizhevsky, et al, 2012

- Convnets for image classification
- More data & more compute power



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." NIPS, 2012.

CNNs for Image Recognition

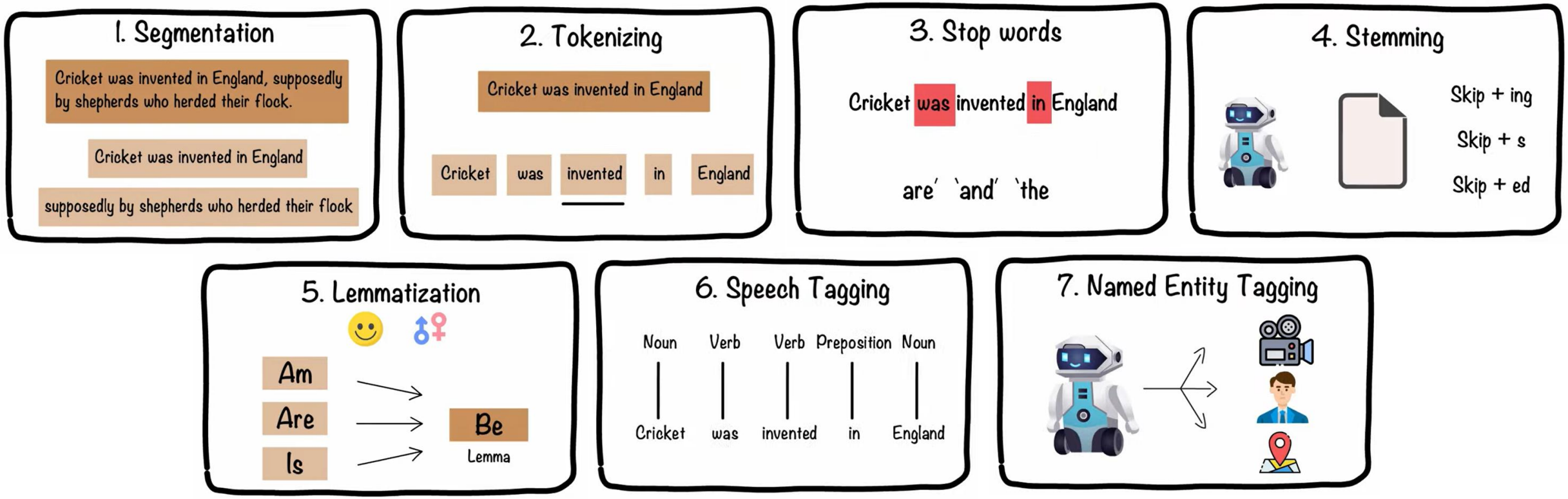


Slide from Kaiming He

Natural Language Processing

NLP Sub-tasks

Many different operations to help process language



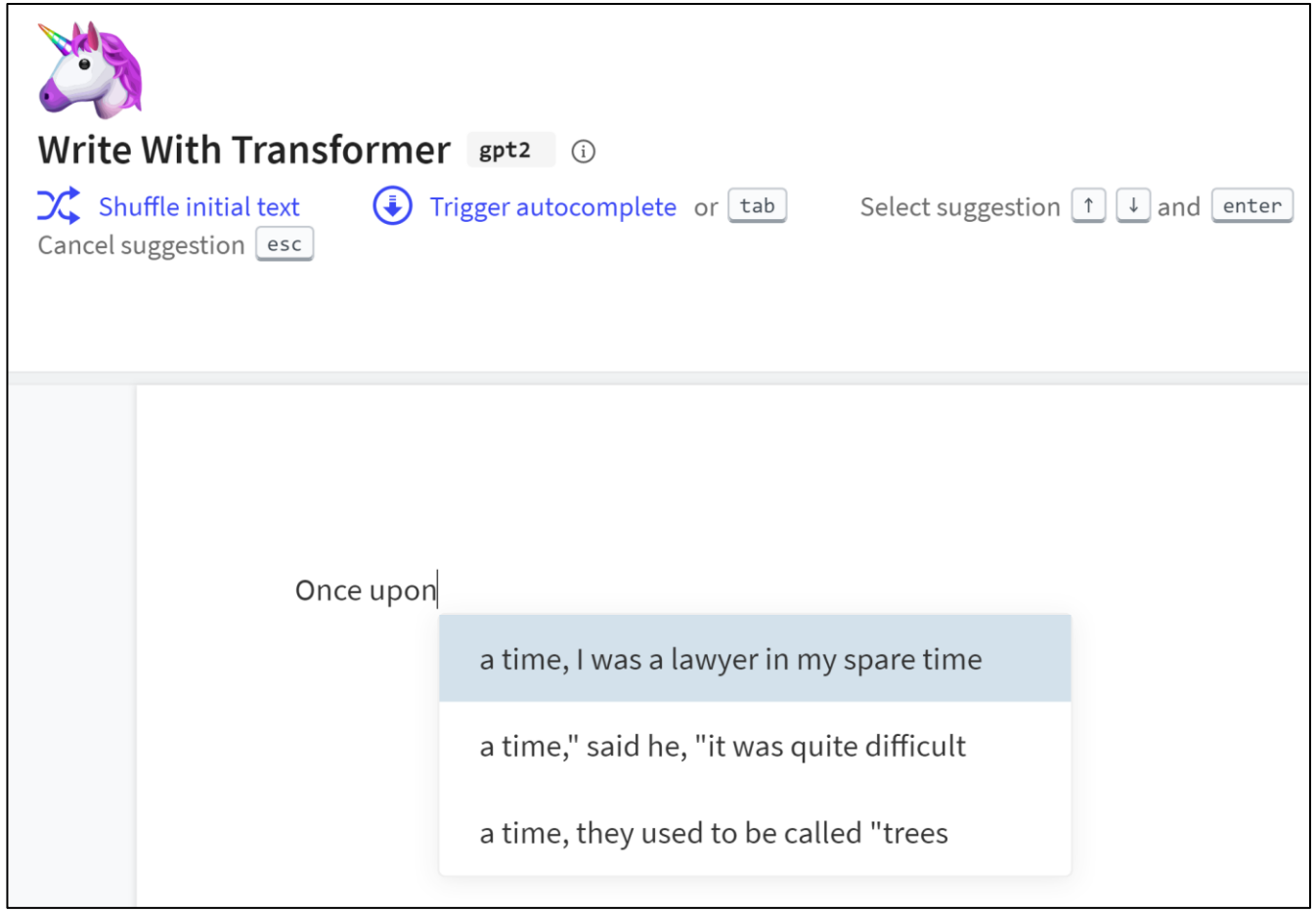
Images: https://www.youtube.com/watch?v=CMrHM8a3hqw

Language Models

Language Model

Generate highly probable next words

https://transformer.huggingface.co/doc/gpt2-large



Probabilistic language models

We haven't covered probability much yet, but with apologies for some forward references, a (probabilistic) language model aims at providing a probability distribution over every word, given all the words before it

$$P(\text{word}_i|\text{word}_1, \dots, \text{word}_{i-1})$$

E.g., you probably have a pretty good sense of what the next word should be:

- "Data science is the study and practice of how we can extract insight and knowledge from large amounts of"

$$P(\text{word}_i = \text{“data”}|\text{word}_1, \dots, \text{word}_{i-1}) = ?$$
$$P(\text{word}_i = \text{“pizza”}|\text{word}_1, \dots, \text{word}_{i-1}) = ?$$

Building language models

Building a language model that captures the true probabilities of natural language is still a distant goal

Instead, we make simplifying assumptions to build approximate tractable models

N-gram model: the probability of a word depends only on the $n-1$ words preceding it

$$P(\text{word}_i|\text{word}_1, \dots, \text{word}_{i-1}) \approx P(\text{word}_i|\text{word}_{i-n+1}, \dots, \text{word}_{i-1})$$

This puts a hard limit on the *context* that we can use to make a prediction, but also makes the modeling more tractable

"large amounts of data" vs. "large amounts of pizza"

Estimating probabilities

A simple way (but *not* the only way) to estimate the conditional probabilities is simply by counting

$$P(\text{word}_i|\text{word}_{i-n+1}, \dots, \text{word}_{i-1}) = \frac{\#(\text{word}_{i-n+1}, \dots, \text{word}_i)}{\#(\text{word}_{i-n+1}, \dots, \text{word}_{i-1})}$$

E.g.:

$$P(\text{“data”}|\text{“large amounts of”}) = \frac{\#(\text{“large amounts of data”})}{\#(\text{“large amounts of”})}$$

Probability Models

Example: Speech Recognition

“artificial ……”

Find most probable next word given “artificial” and the audio for second word.

Probability Models

Example: Speech Recognition

“artificial ……”

Find most probable next word given “artificial” and the audio for second word.

Which second word gives the highest probability?

$P(\textbf{limb} \mid \text{artificial}, \text{audio})$

$P(\textbf{intelligence} \mid \text{artificial}, \text{audio})$

$P(\textbf{flavoring} \mid \text{artificial}, \text{audio})$

<u>Break down problem</u>

n-gram probability * audio probability

$P(\textbf{limb} \mid \text{artificial}) * P(\text{audio} \mid \textbf{limb})$

$P(\textbf{intelligence} \mid \text{artificial}) * P(\text{audio} \mid \textbf{intelligence})$

$P(\textbf{flavoring} \mid \text{artificial}) * P(\text{audio} \mid \textbf{flavoring})$

N-gram Training

Where do the n-gram probabilities come from?

Google n-grams demo

NLP Can Be Huge

N-gram probabilities

Vocabulary size: 50,000

NLP Training

Self-supervised

Example: Jane Austen, *Pride and Prejudice*

Vanity and pride are different things, though the words are often used synonymously. A person may be proud without being vain. Pride relates more to our opinion of ourselves, vanity to what we would have others think of us.

NLP Training

Self-supervised learning (auto-regressive)

Example: Jane Austen, *Pride and Prejudice*

Vanity and pride are different things, though the words are often used synonymously. A person may be proud without being vain. Pride relates more to our opinion of ourselves, vanity to what we would have others think of us.

Examples

Random samples from language model trained on Shakespeare:

n=1: "in as , stands gods revenge ! france pitch good in fair hoist an what fair shallow-rooted , . that with wherefore it what a as your . , powers course which thee dalliance all"

n=2: "look you may i have given them to the dank here to the jaws of tune of great difference of ladies . o that did contemn what of ear is shorter time ; yet seems to"

n=3: "believe , they all confess that you withhold his levied host , having brought the fatal bowels of the pope ! ' and that this distemper'd messenger of heaven , since thou deniest the gentle desdemona ,"

n=7: "so express'd : but what of that ? 'twere good you do so much for charity . i cannot find it ; 'tis not in the bond . you , merchant , have you any thing to say ? but little"

This is starting to look a lot like Shakespeare… because it is Shakespeare

Slide: CMU, Zico Kolter

How do we pick n?

Hyperparameter

Lower n: less context, but more samples of each possible n-gram

Higher n: more context, but less samples

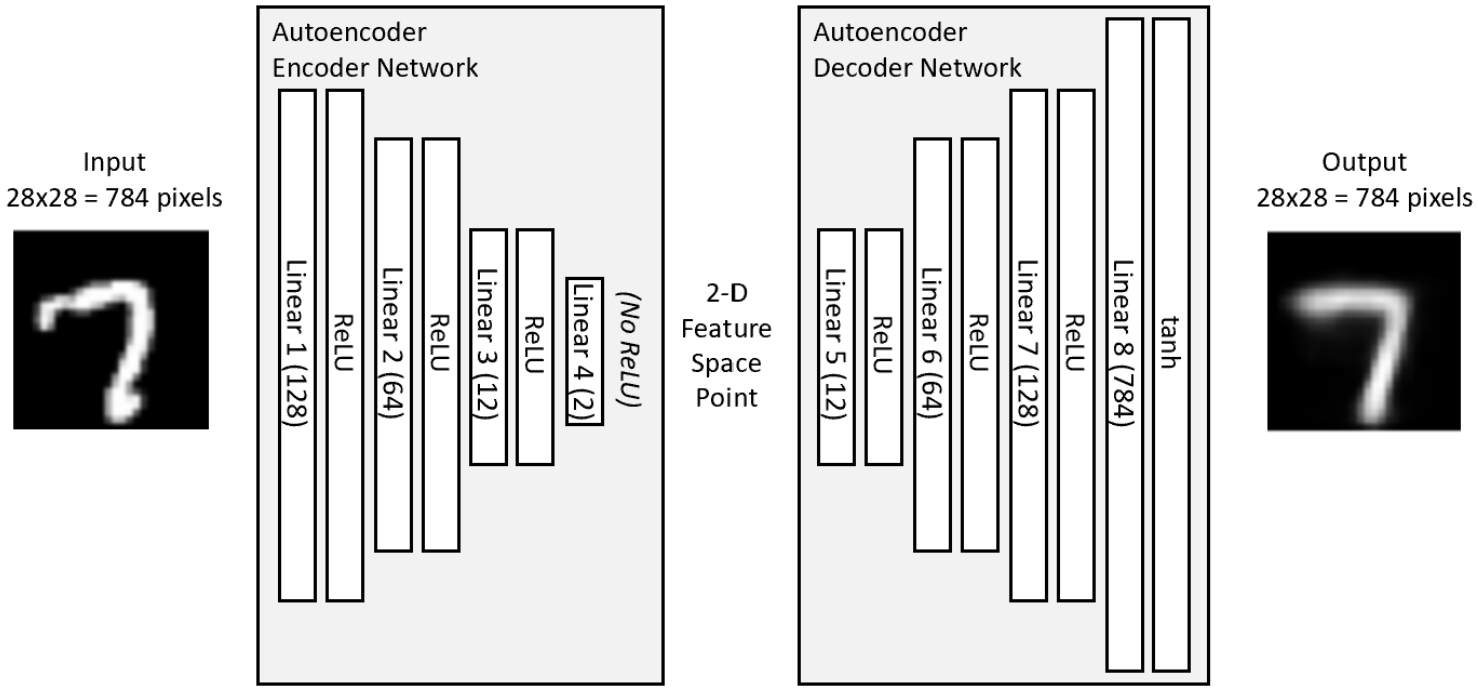
“Correct” choice is to use some measure of held-out cross-validation

In practice: use $n = 3$ for large datasets (i.e., triplets) , $n = 2$ for small ones

Not-so-probabilistic language models

Reminder: Autoencoders

Encoding and decoding



Text processing with word2vec

Word embeddings

Skip-gram

```
score(word, <other words around it>)
```

Training data:

“The king sat on the throne”

“the queen sat on the throne”

“the banana is yellow”

“they sat on the yellow bus”

• king	• king
• sat	• sat
• throne	• throne
• queen	• queen
• banana	• banana
• yellow	• yellow
• they	• they
• bus	• bus

Attention/Transformers

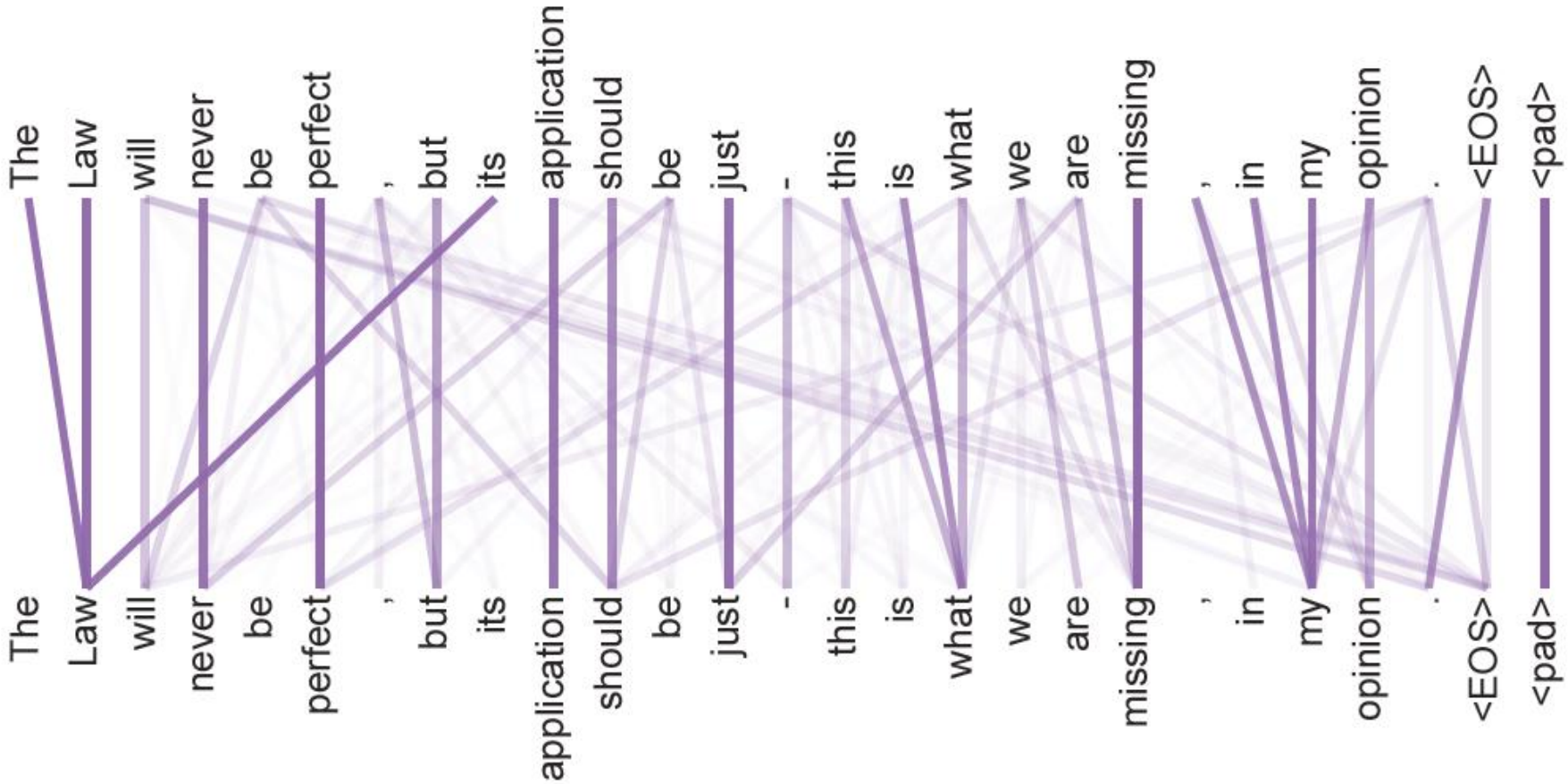


Figure: https://arxiv.org/abs/1706.03762

Attention/Transformers

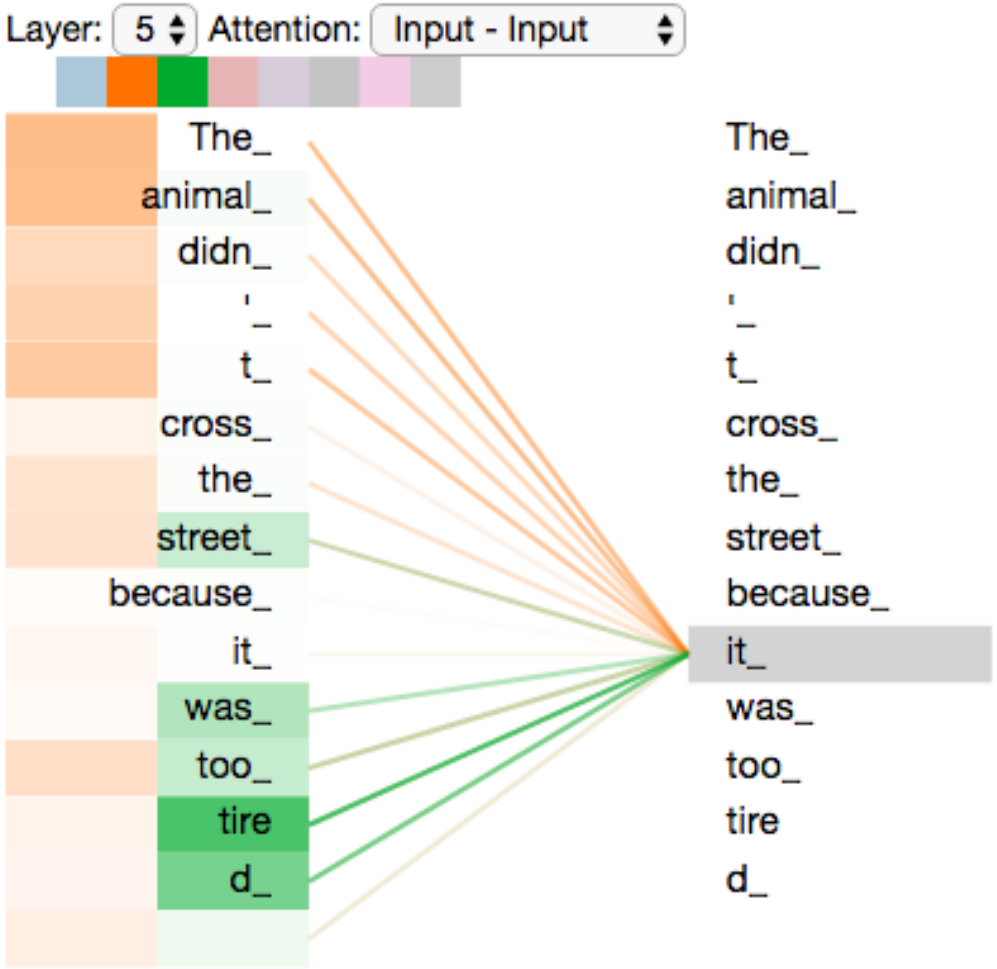


Figure: https://jalammar.github.io/illustrated-transformer/

Exploring GPT2

GPT-3 Language Model

Advanced language model

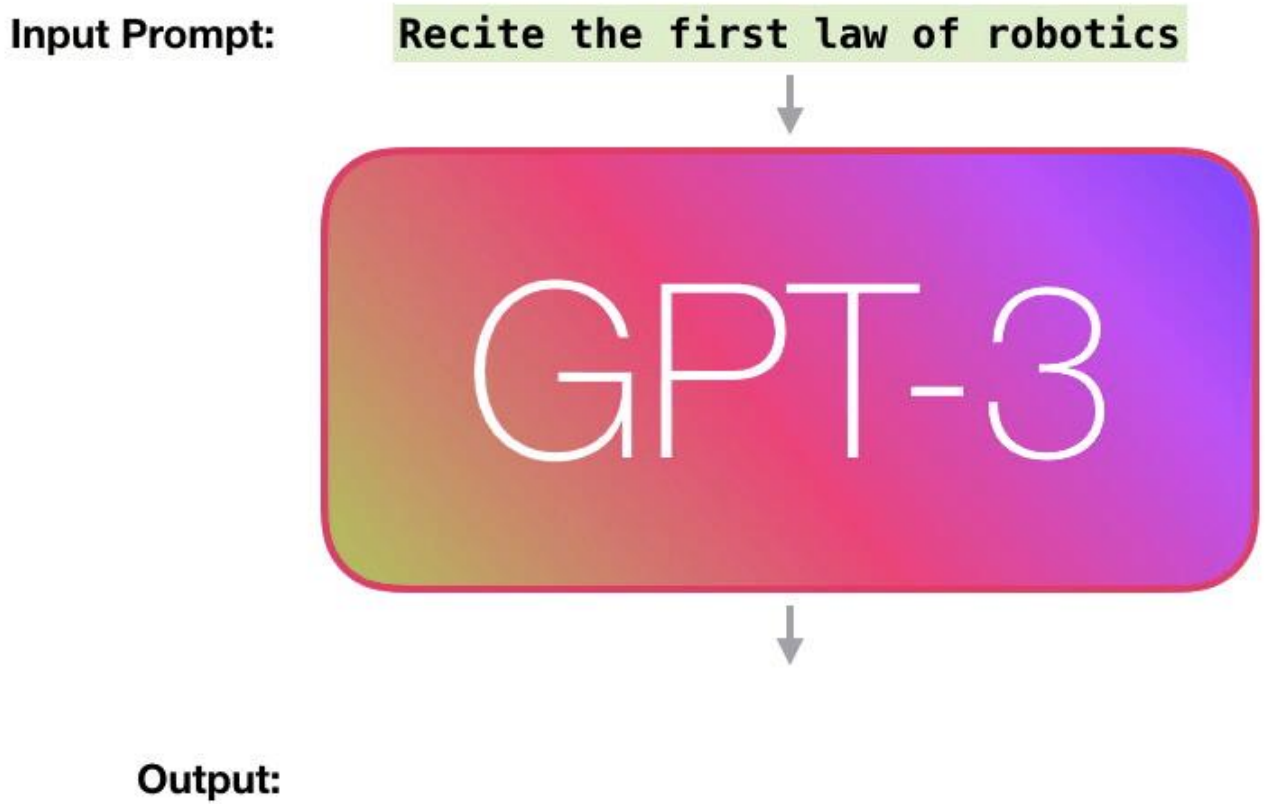


Image: https://jalammar.github.io/how-gpt3-works-visualizations-animations/

GPT-3 Language Model

State-of-the-art language model

- Trained with dataset of 300 billion tokens
- 175 billion parameters
- It was estimated to cost 355 GPU years and cost $4.6m

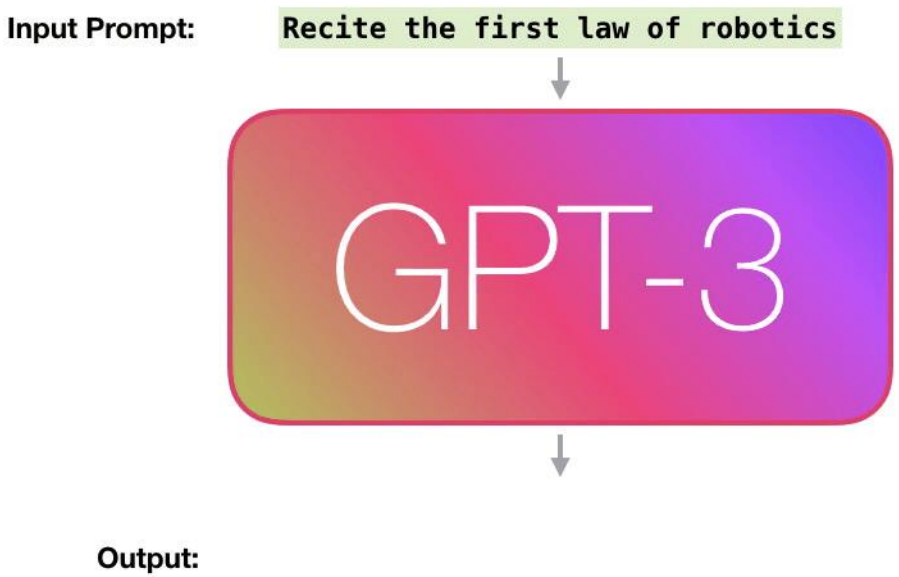
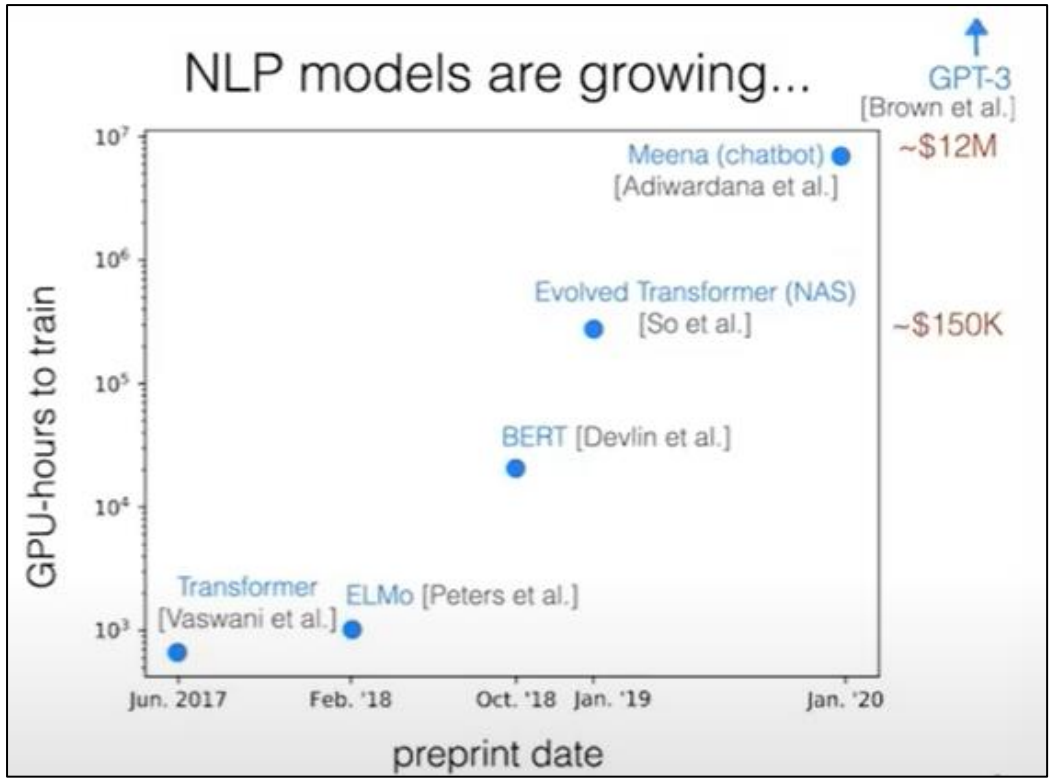


Image: https://jalammar.github.io/how-gpt3-works-visualizations-animations/

NLP Ethics

Environmental concerns

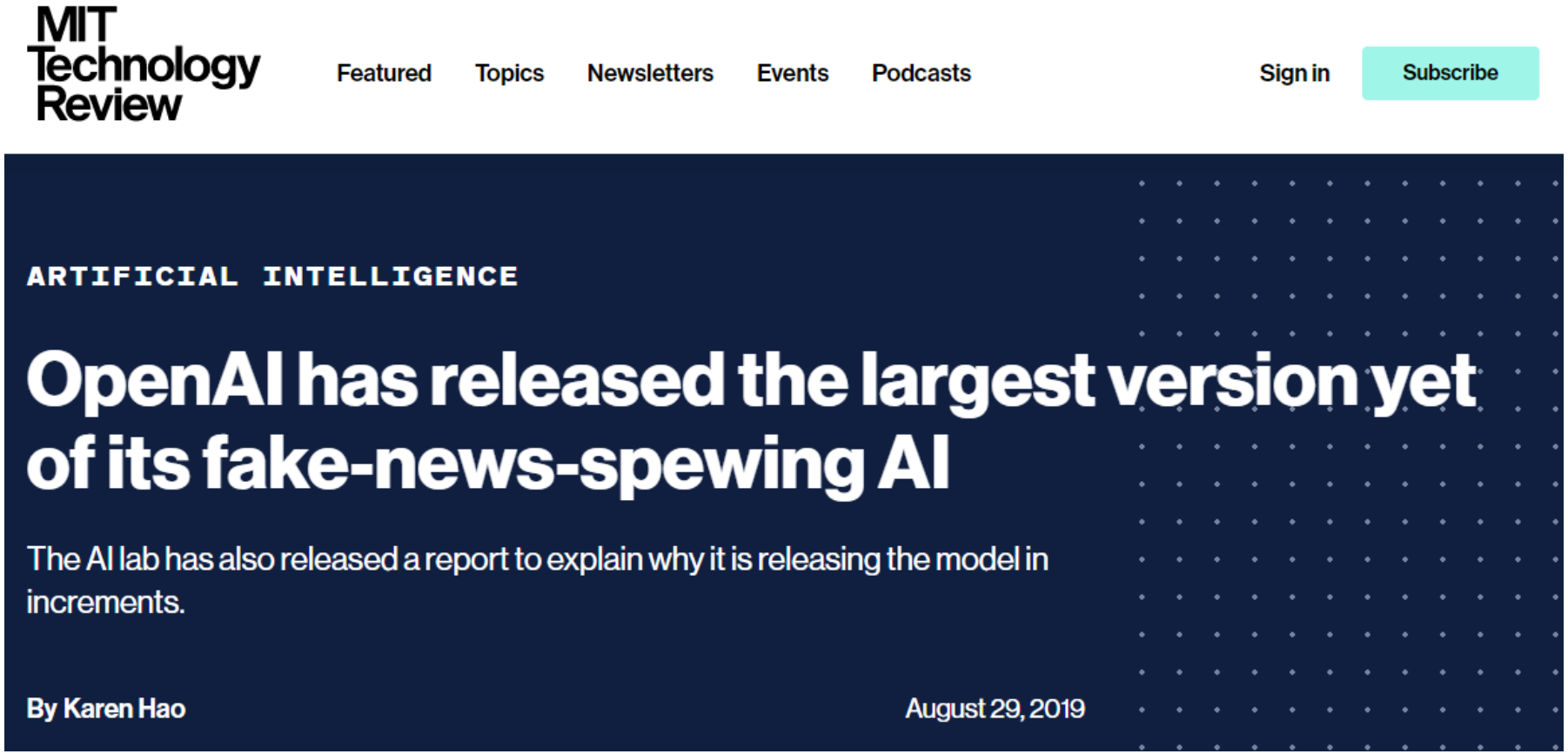


Emma Strubell
Assistant Professor
Language Technologies Institute
CMU

[Frontiers in Machine Learning: Climate Impact of Machine Learning]

NLP Ethics

Misuse



https://www.technologyreview.com/2019/08/29/133218/openai-released-its-fake-news-ai-gpt-2/

NLP Ethics

Bias

https://spectrum.ieee.org/tech-talk/artificial-intelligence/machine-learning/in-2016-microsofts-racist-chatbot-revealed-the-dangers-of-online-conversation

NLP Ethics

Bias

https://venturebeat.com/2021/06/10/openai-claims-to-have-mitigated-bias-and-toxicity-in-gpt-3/

NLP Ethics

Dangerous errors

AINEWS

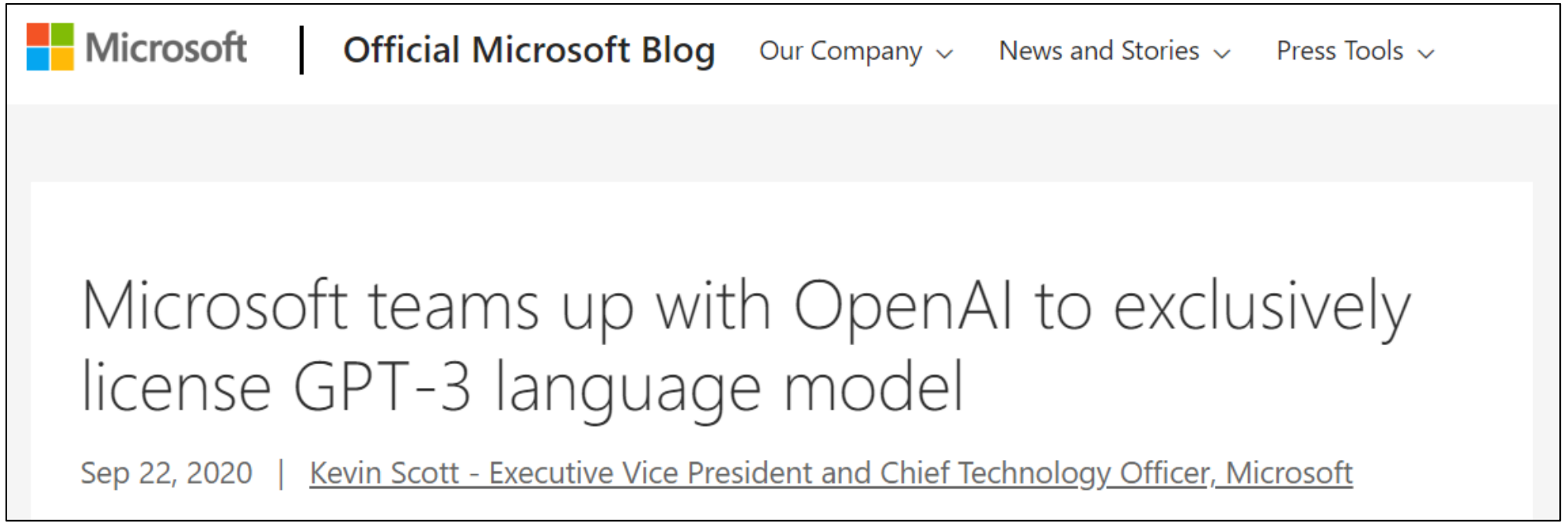
Medical chatbot using OpenAI's GPT-3 told a fake patient to kill themselves

By Ryan Daws | October 28, 2020 | TechForge Media
Categories: Chatbots, Healthcare,

https://artificialintelligence-news.com/2020/10/28/medical-chatbot-openai-gpt3-patient-kill-themselves/

NLP Ethics

Digital divide



https://blogs.microsoft.com/blog/2020/09/22/microsoft-teams-up-with-openai-to-exclusively-license-gpt-3-language-model/