3D Pattern Recognition Using Deep Neural Networks for Liquid Argon Time Projection Chambers (LArTPCs)

Kazuhiro Terao

SLAC National Accelerator Laboratory

Introduction

This **workshop's charge**:

This meeting will focus on the options of the magnet, comparison of the performance between the low-mass tracking options, electromagentic calorimeters, and gain better understanding of the scientifc potenial of the 3-d scintillator detector and the PRISM concept in DUNE.

Disclaimer: this talk does not contain any "result," but my research focus = "alternative" data reconstruction path using machine learning technique

+20 lbs.
after Ph.D

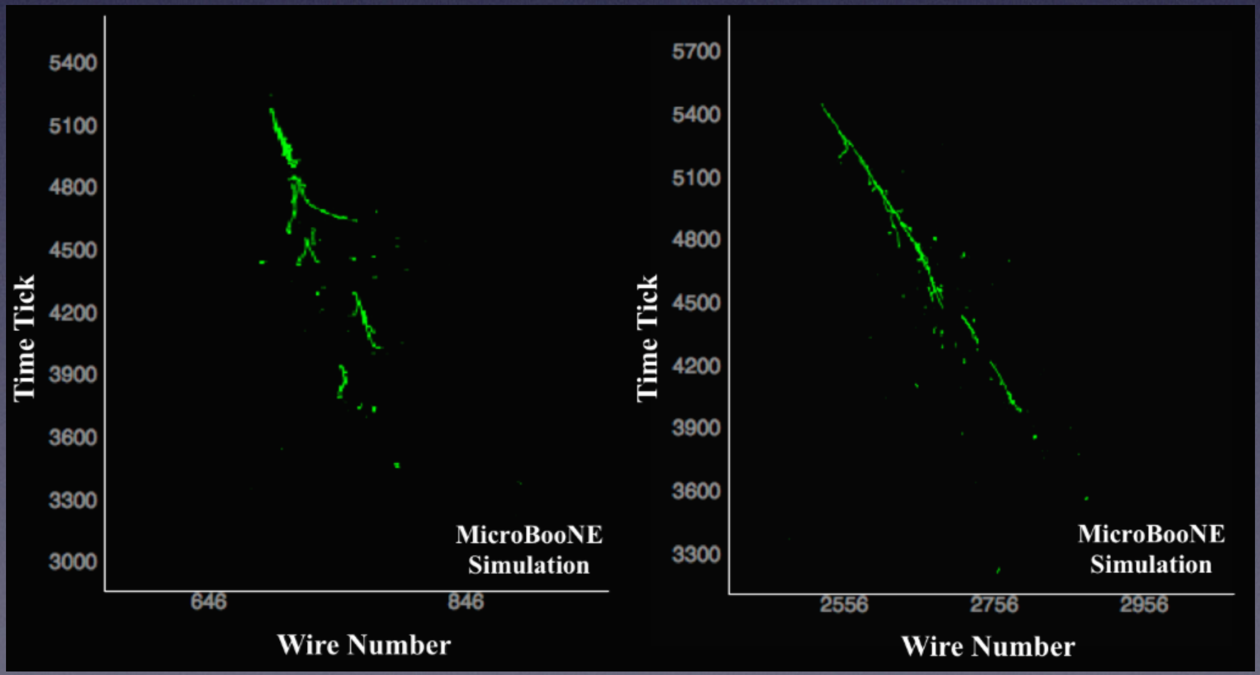
About me: **Kazuhiro Terao** (Kazu), 4 yrs in MicroBooNE, just joined SLAC and DUNE ND.

Interest: deep neural network (DNN) technique R&D for LArTPC detectors

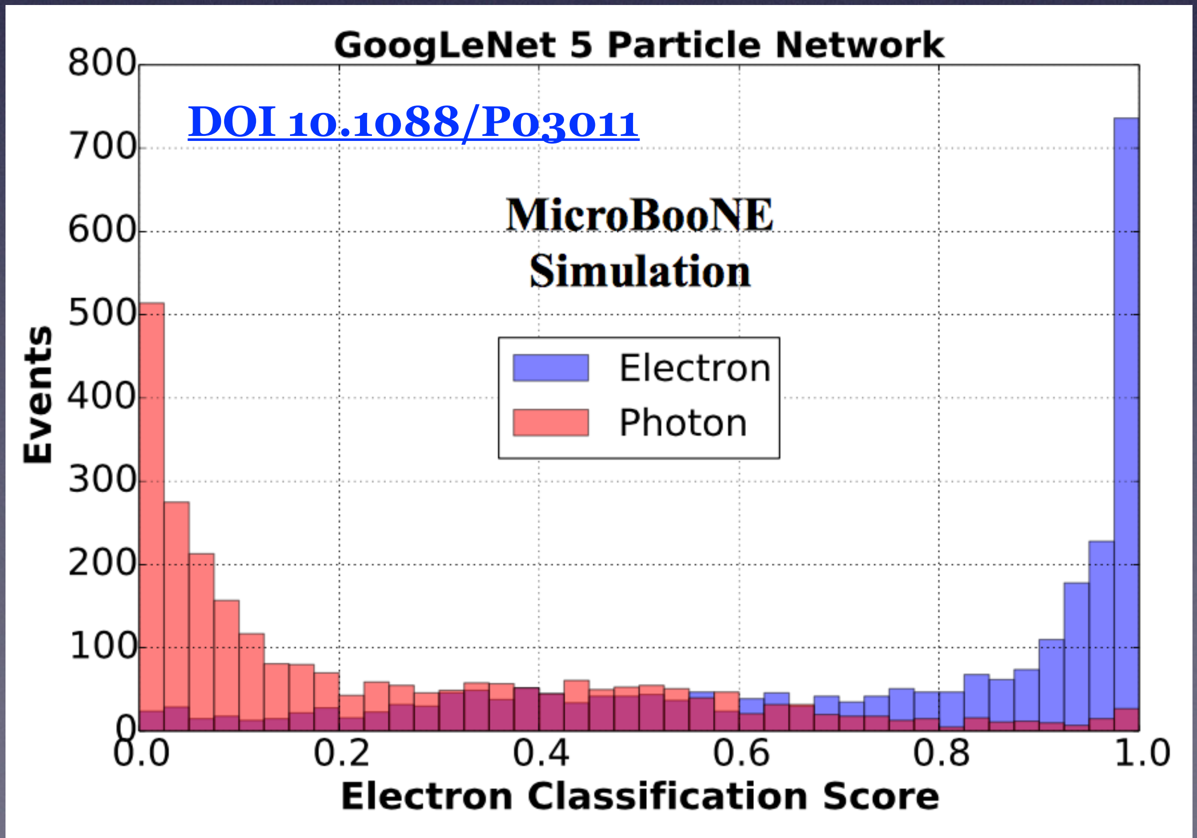
DNN for LArTPC Data Analysis

Why DNN?

- Modern solution for pattern recognition in computer vision (CV), the heart of LArTPC reconstruction
- Machine learning = natural support for algorithm optimization. Can combine many tasks (end-to-end).
- Works for LArTPC: demonstration in MicroBooNE



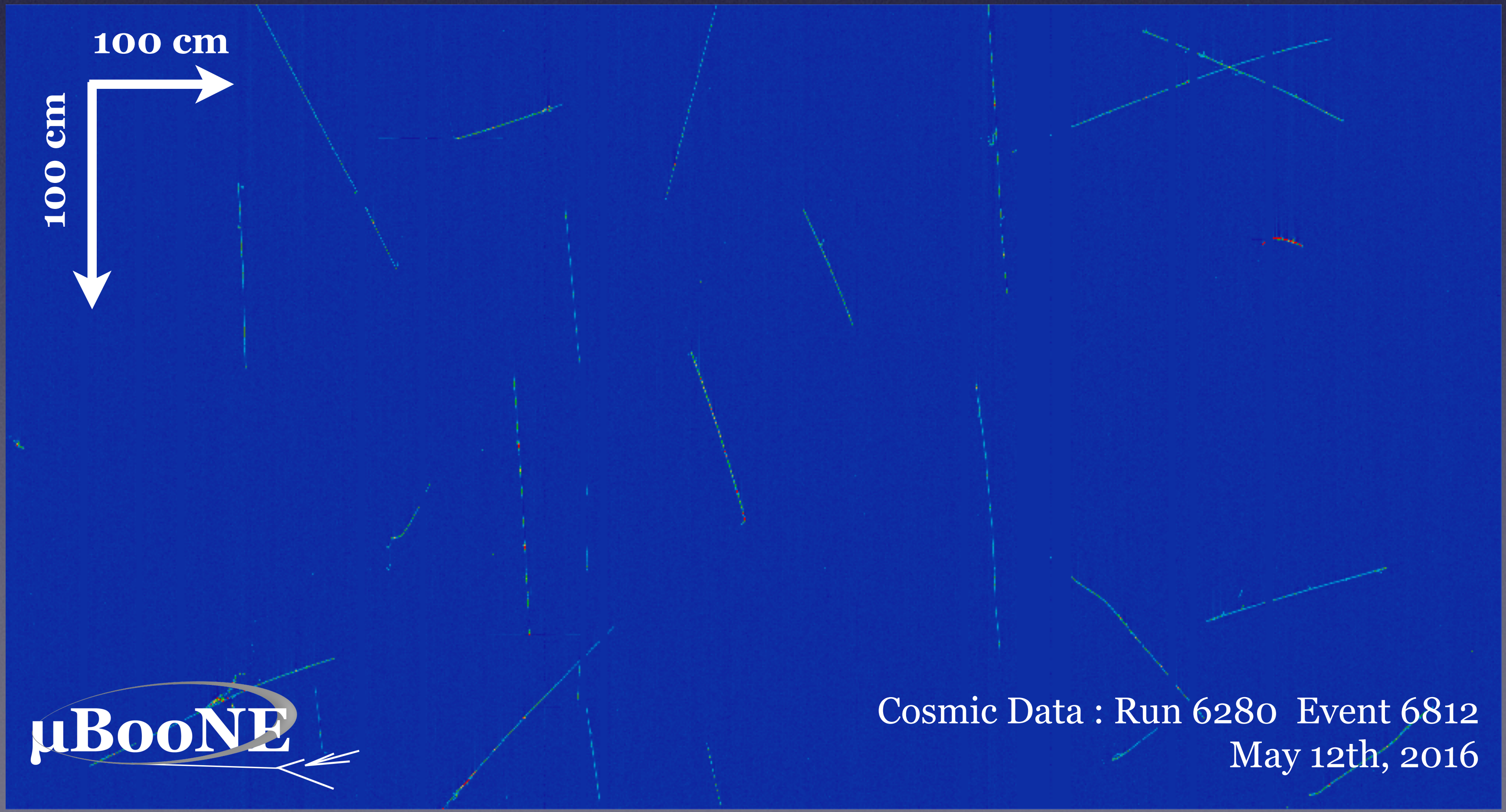
electron vs. gamma



DNN for LArTPC Data Analysis

Popular application: image classifier

- First applications in the field
  - **NoVA's** neutrino event classifier, **MicroBooNE's** signal (neutrino) vs. background (cosmic) classifier & particle ID
- **Concern:** A huge information reduction step (millions of pixels down to 1 variable!) makes DNN a big black box.



MicroBooNE Collection Plane

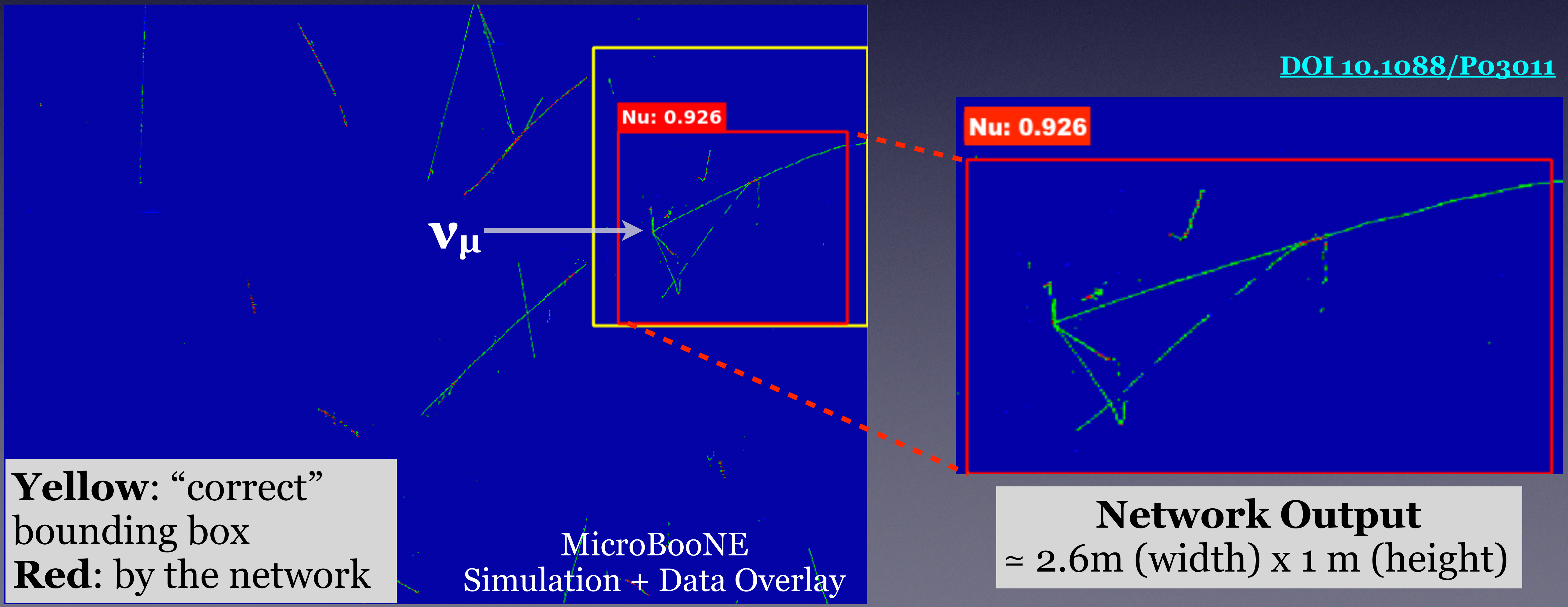
3456 wires x 9600 ticks
≃ 33e6 pixels (variables)

DNN for LArTPC Data Reconstruction

Reconstruction Using DNN

- **True strengths**: learns & extracts essential features in data for problem solving.
- Beyond image classification: can extract "features" in more basic physical observables, like "vertex location", "particle trajectory (clustering)", etc. … "**reconstruction**"!

DOI 10.1088/P03011



Yellow: "correct" bounding box
Red: by the network

Network Output
≃ 2.6m (width) x 1 m (height)

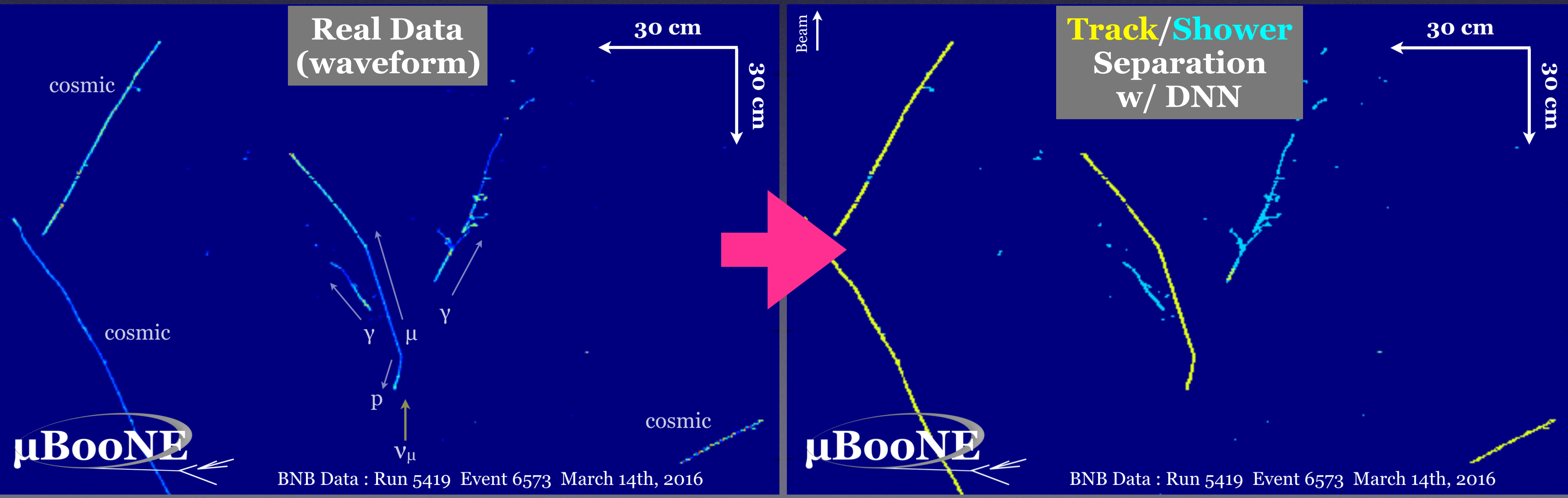
DNN for LArTPC Data Reconstruction

Development of chain

- Develop DNN to perform reconstruction step-by-step

Pre-processing (noise removal, etc) → Vertex Detection Particle Clustering → Particle Identification

- Data/simulation validation at each stage
- Whole chain optimization (end-to-end training) by combining multiple networks



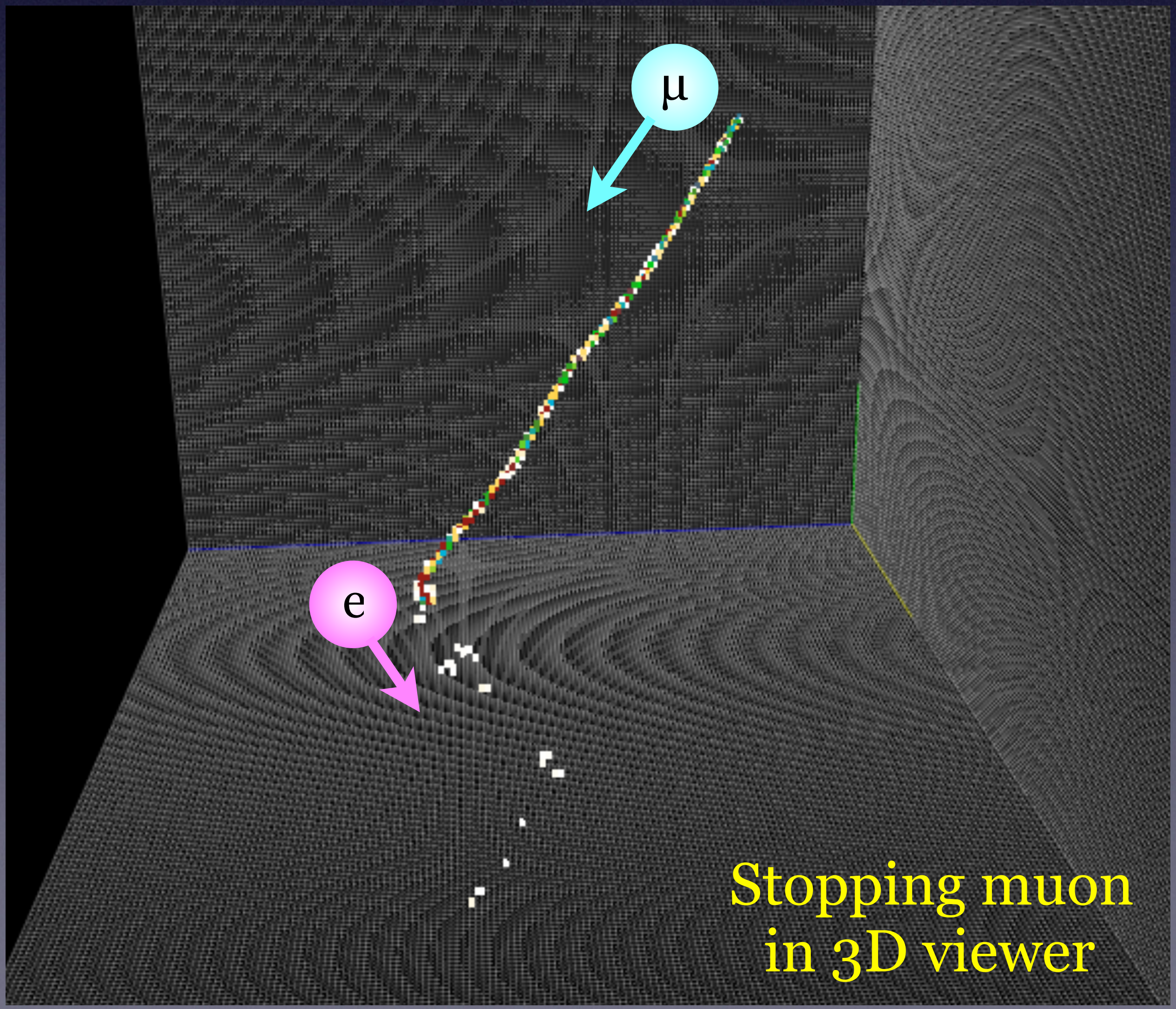
DATA CCπ^0 Candidate

Pixel-level analysis via custom CNN

Development Toward 3D Reconstruction

Current focus: 2 types of DNNs

- **Smoothing/Filtering**: makes a better 3D voxel (point) prediction, remove/fixes "ghost points"
- **3D Pattern Recognition**: find 3D interaction vertex + particle clustering of 3D charge depositions



Stopping muon in 3D viewer

Software Tools

LArCV ... standalone C++ software with extensive Python support for image and volumetric (2D/3D) data storage & processing. Fast data loading API to open source DNN softwares + Qt/ OpenGL based 2D/3D data visualization

DeepLearnPhysics ... github group supports cross-experiment software and DNN architecture development (link)

Current Status & Near Term Milestones

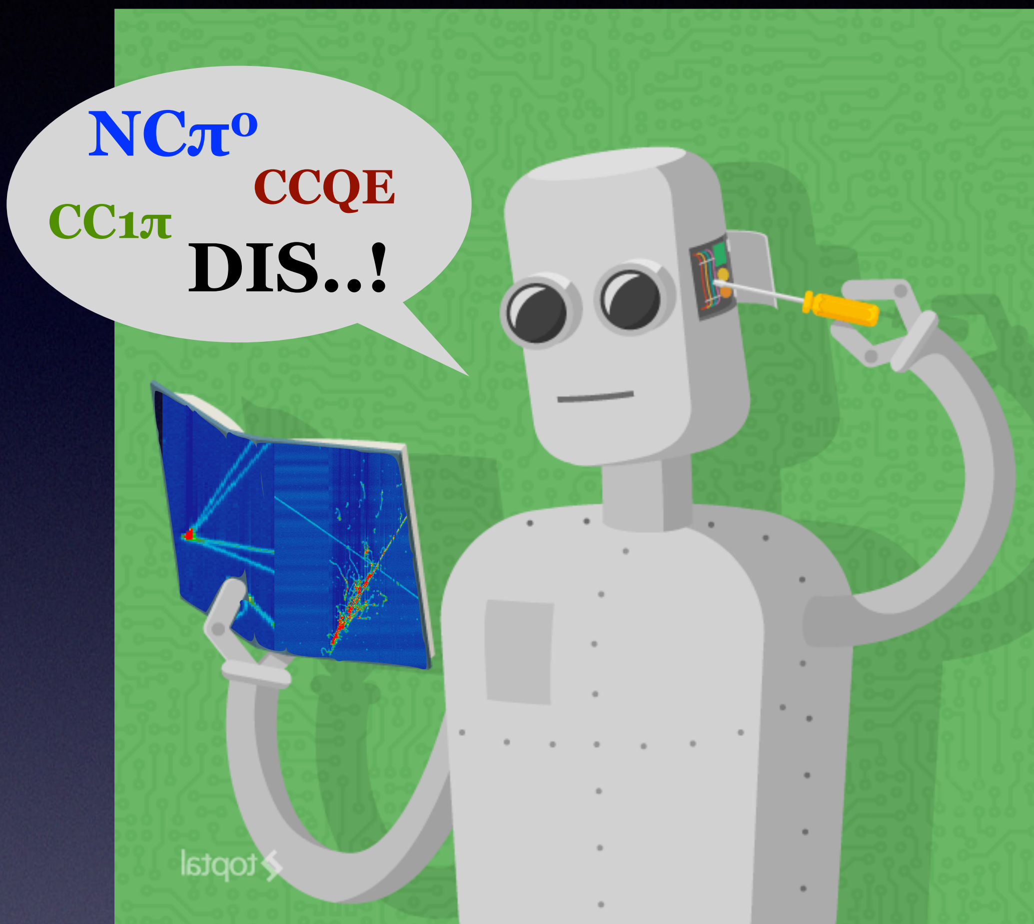
- **Finished 3D voxel data support**
  - Trained 3D DNN for single particle ID (same as UB paper) with 1cm cubic voxels for ≃ 2 m^3 volume (works)
- **3D vertex finding** with track/shower separation
  - Immediate target, training starts this week
- **3D voxel "smoothing"** network
  - Interest from wire detectors, clear path forward
  - Need to understand more for multiplex pixel detectors
- **3D particle clustering** network
  - Requires 3D object detection network to work first
  - After 3D vertex finding network

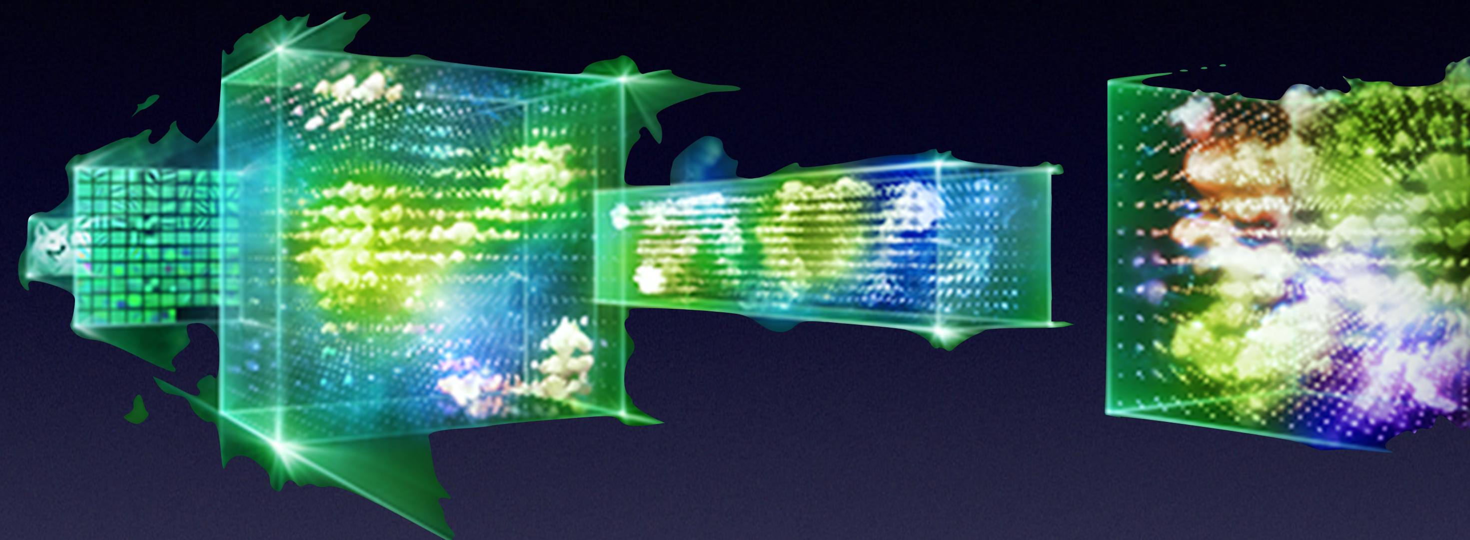
Plan to **benchmark performance with ArgonCUBE** (**LArPix/PixLAr**) data as we go. Plan to **utilize simulation tools by LBL** (**Dan** & **Chris**)

Thank you
for
your attention!

Any Questions

Back ups





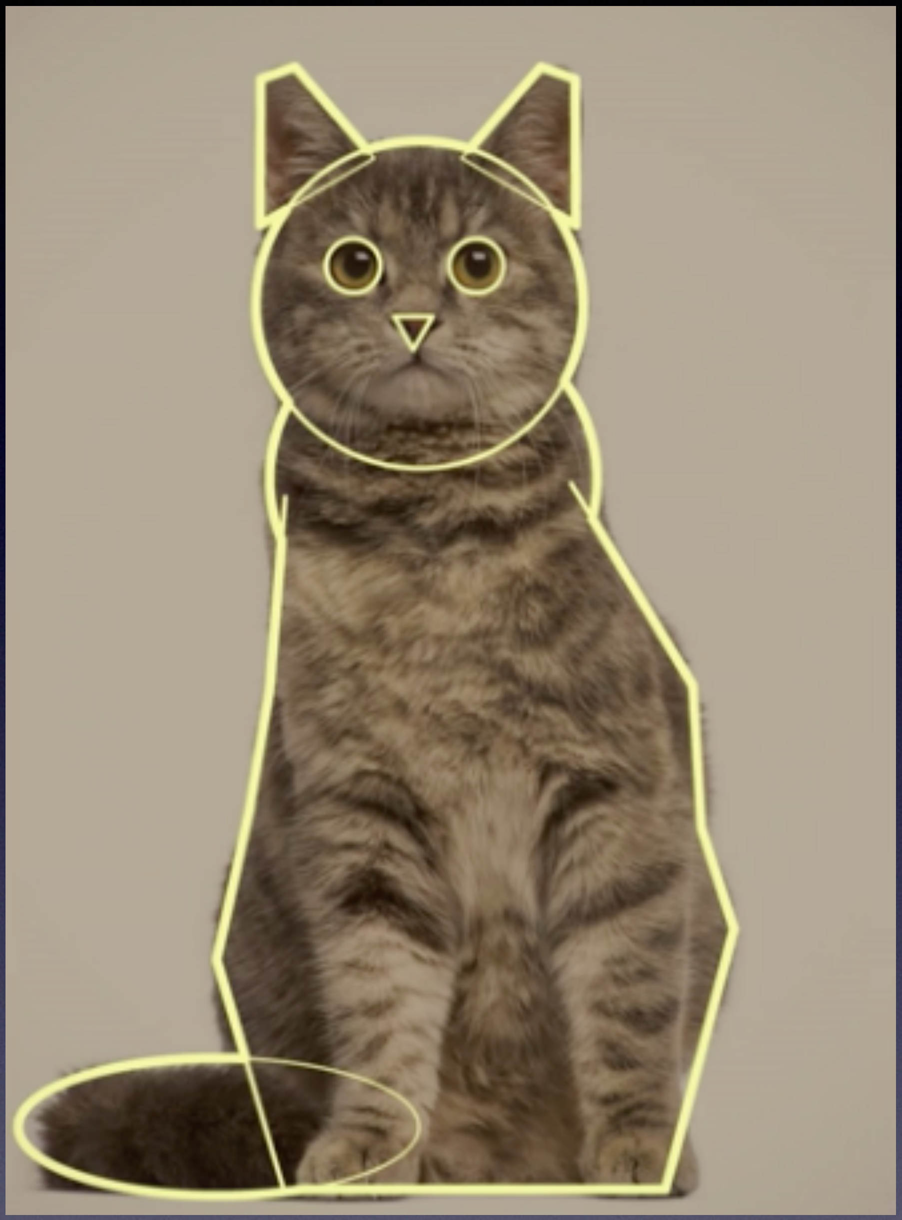
Convolutional Neural Networks

How Does It Work?

Image Analysis: Identifying a Cat

Taken from slides by Fei-Fei's TED talk

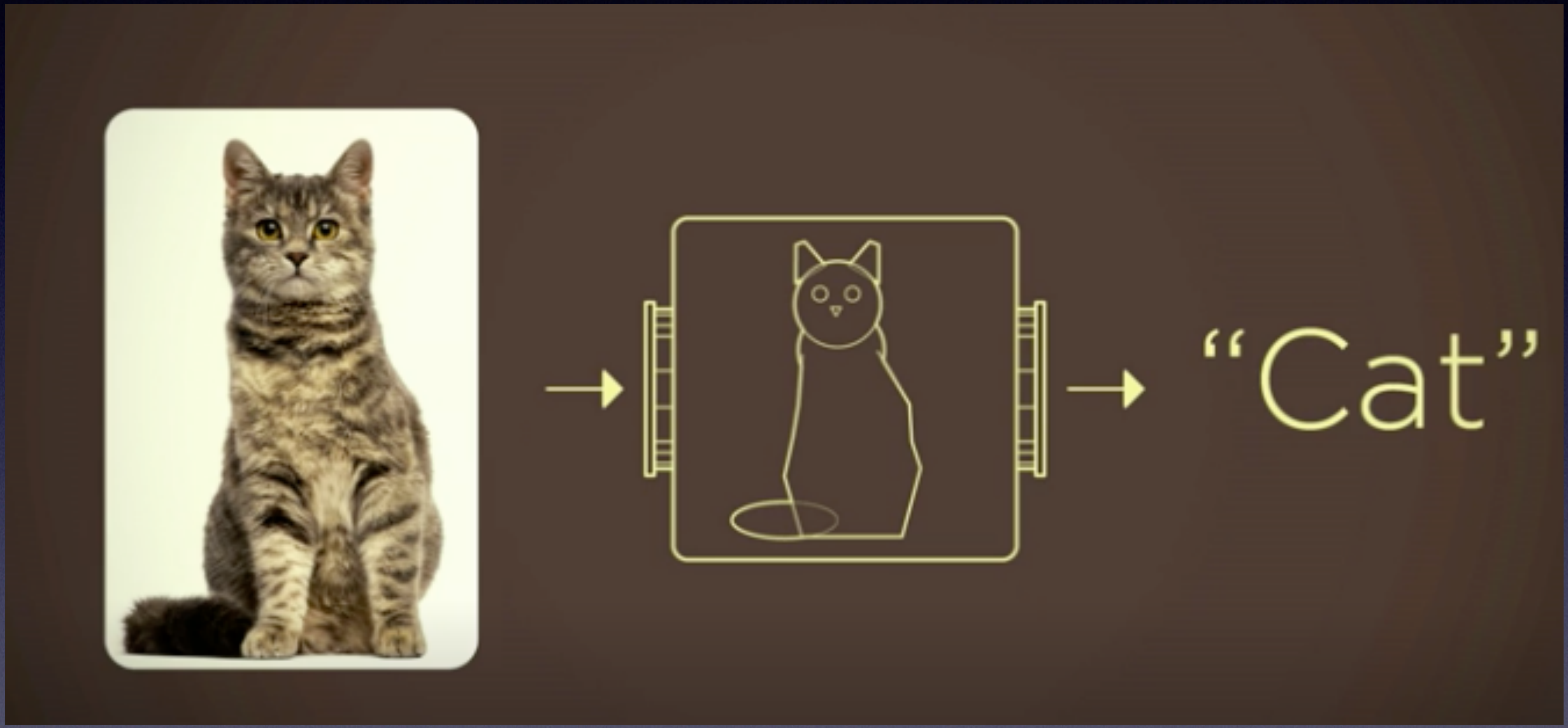
Image Analysis: Identifying a Cat



A cat = collection of certain shapes
(object modeling in early days)

Taken from slides by Fei-Fei's TED talk

Image Analysis: Identifying a Cat



A cat = collection of certain shapes
(object modeling in early days)

Taken from slides by Fei-Fei's TED talk

Image Analysis: Identifying a Cat

… how about this?

Take into account for a view point

Taken from slides by Fei-Fei's TED talk

Image Analysis: Identifying a Cat

... how about this?

... and maybe more shapes

Taken from slides by Fei-Fei's TED talk

Image Analysis: Identifying a Cat

… gets way worse …

… I (a human) am ***never taught exactly how cat should look like*** by anyone, but I somehow ***can recognize them really well***.

Taken from slides by Fei-Fei's TED talk

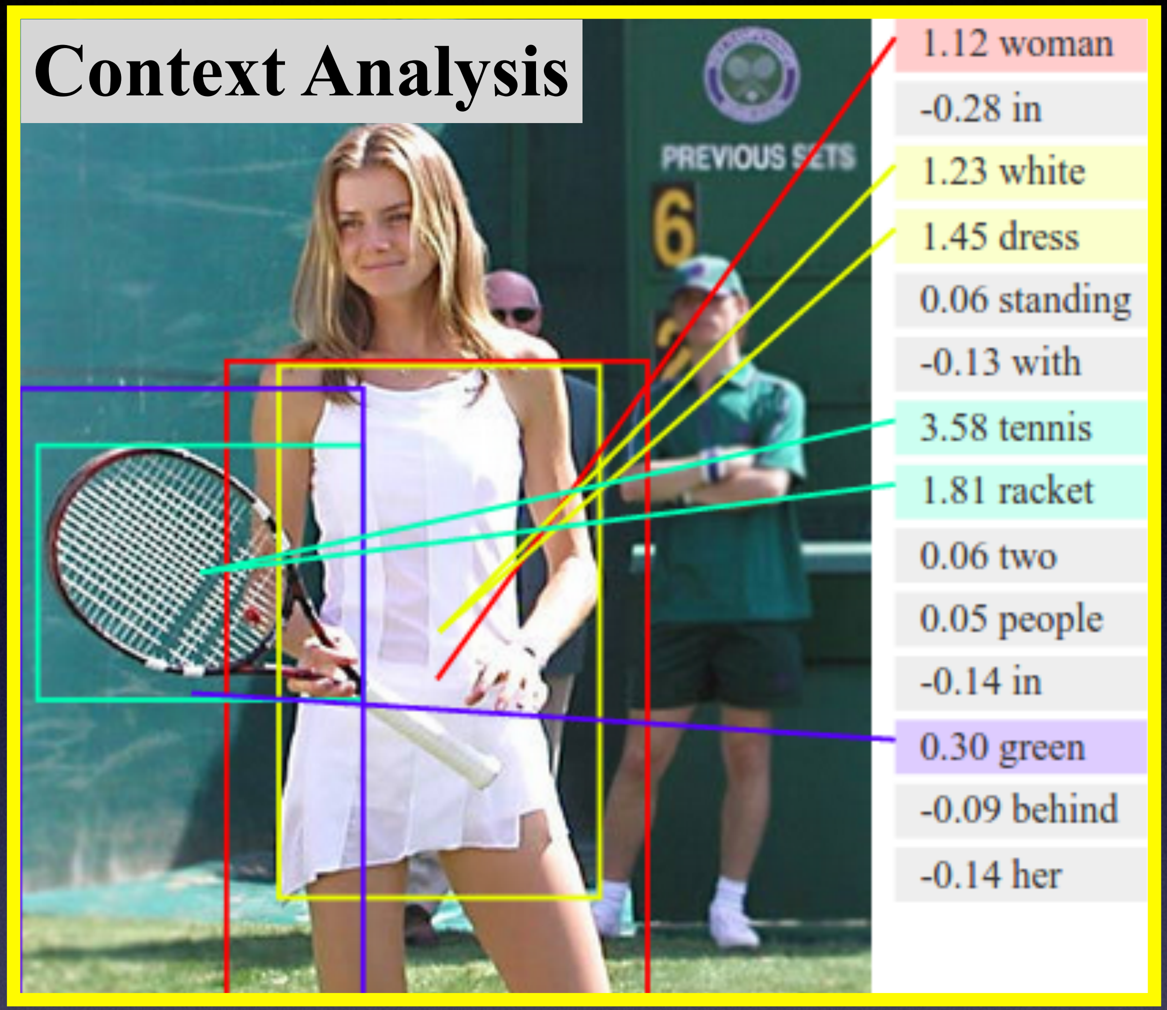
Image Analysis: Identifying a Cat

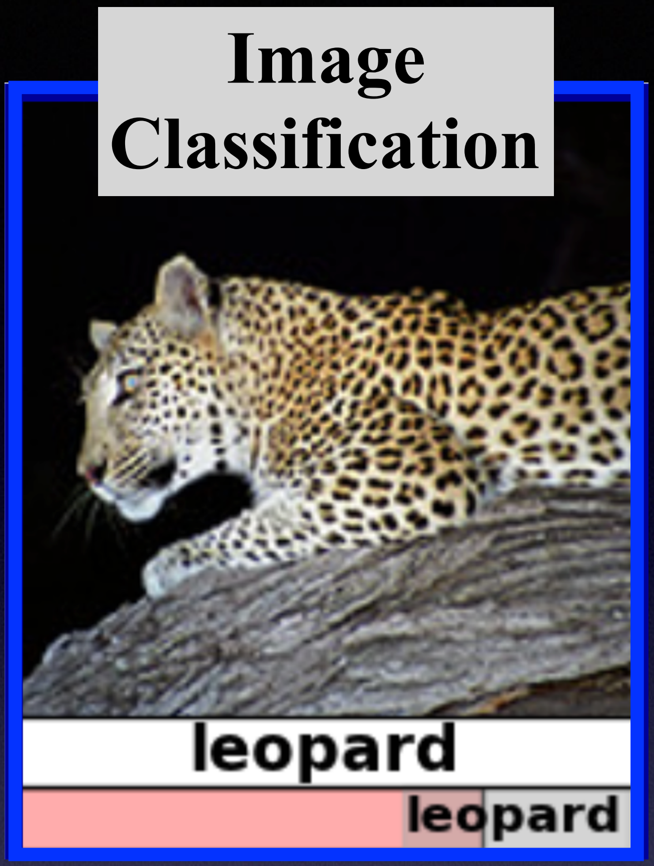
… gets way worse …

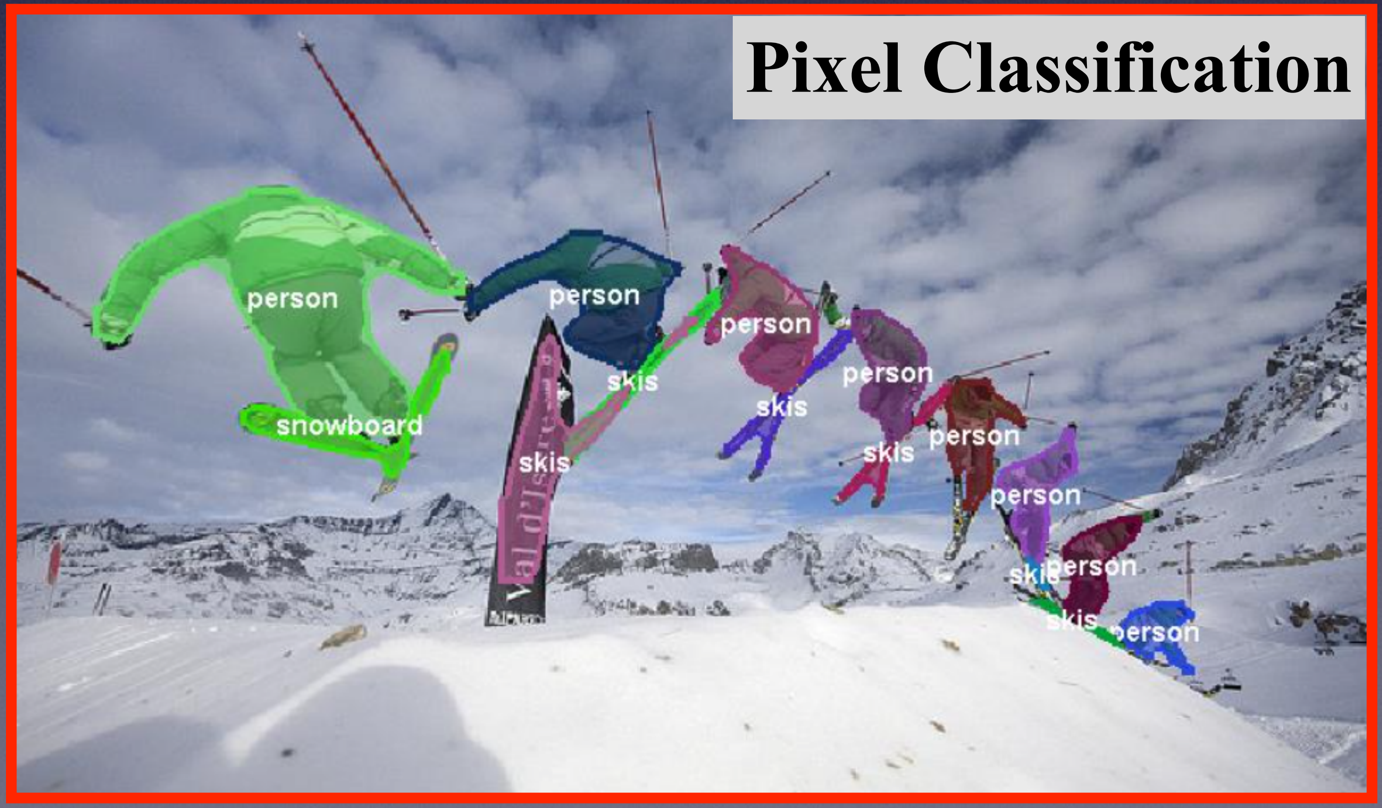
A breakthrough: a machine learning algorithm that forms (trains) itself by sampling a large set of data to "learn" how cat looks like (distribution)

Taken from slides by Fei-Fei's TED talk

Introduction to CNNs (I)







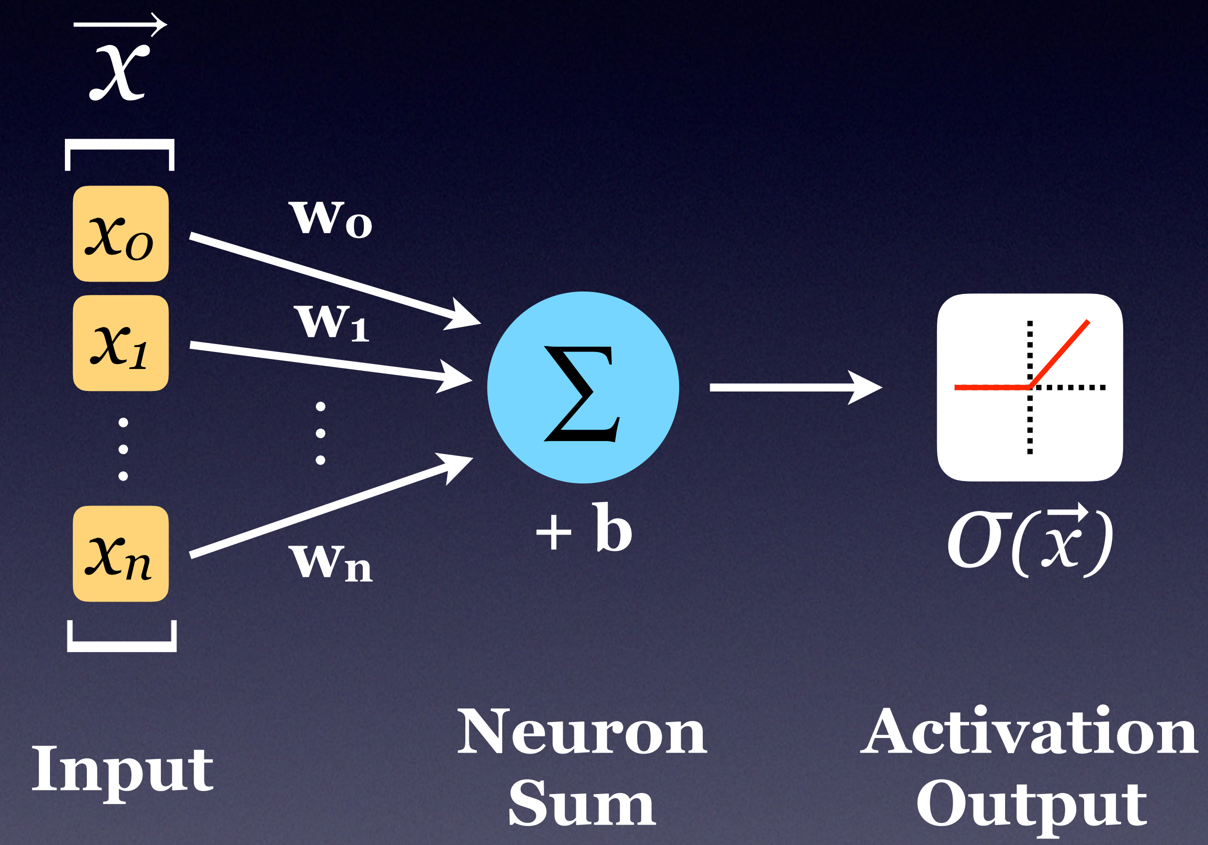
self-driving car,
image captioning,
playing a boardgame,
... and more!

Introduction to CNNs (II)

Background: Neural Net

The basic unit of a neural net is the ***perceptron*** (loosely based on a real neuron)

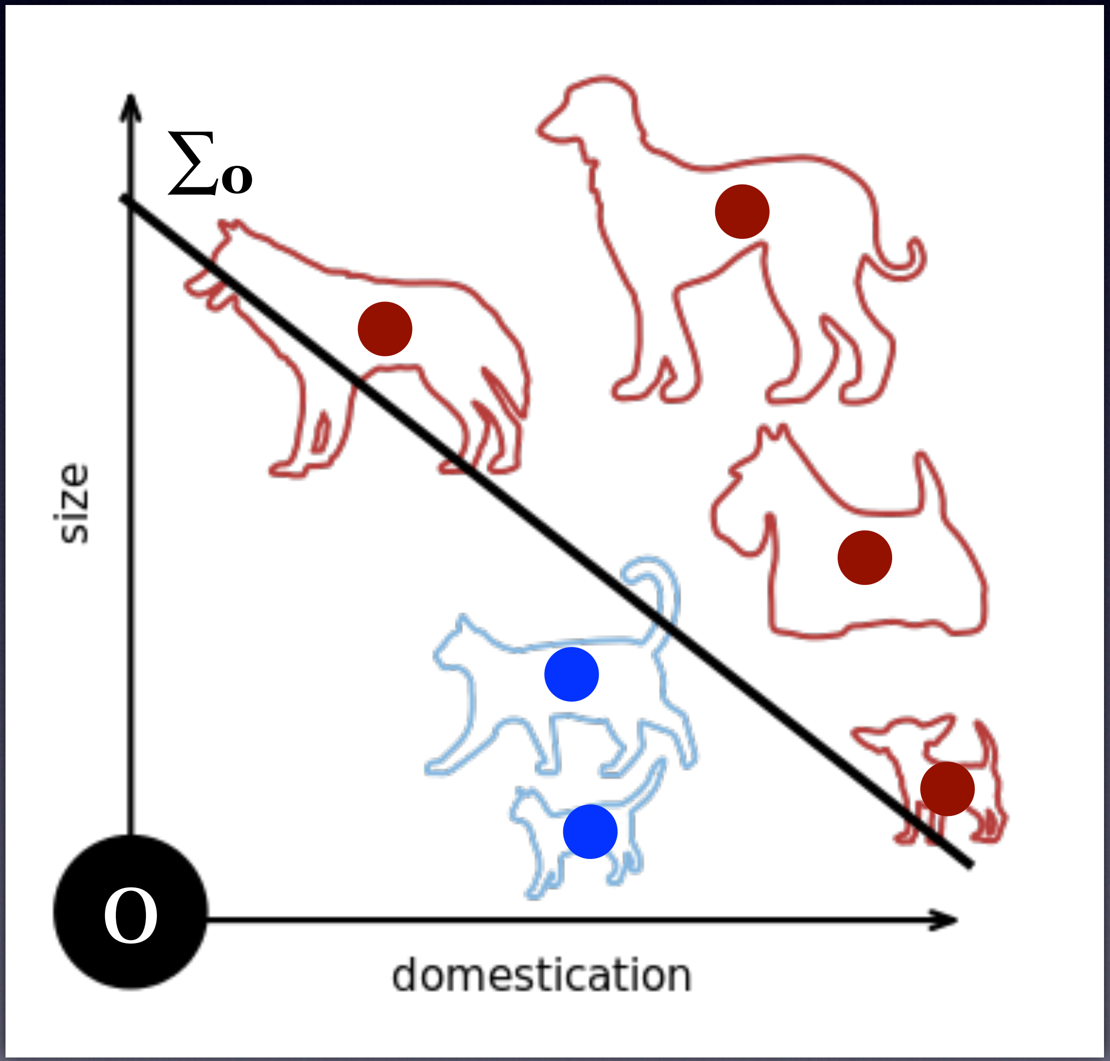
Takes in a vector of inputs (x). Commonly inputs are summed with weights (w) and offset (b) then run through activation.



$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

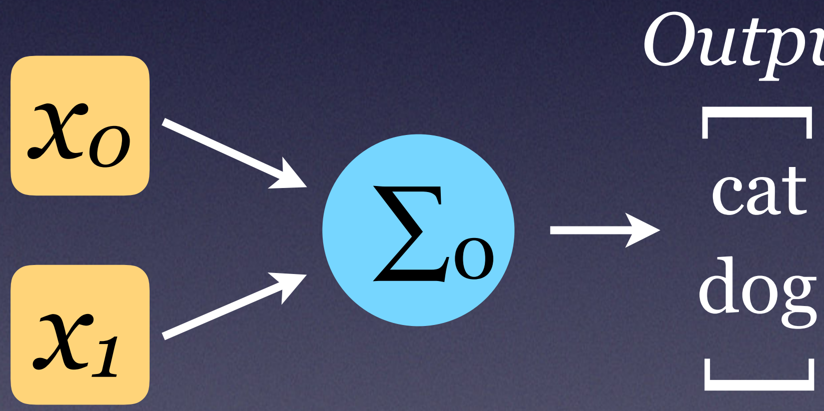
Introduction to CNNs (II)
Perceptron 2D Classification

Imagine using two features to separate cats and dogs



from [**wikipedia**]

$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

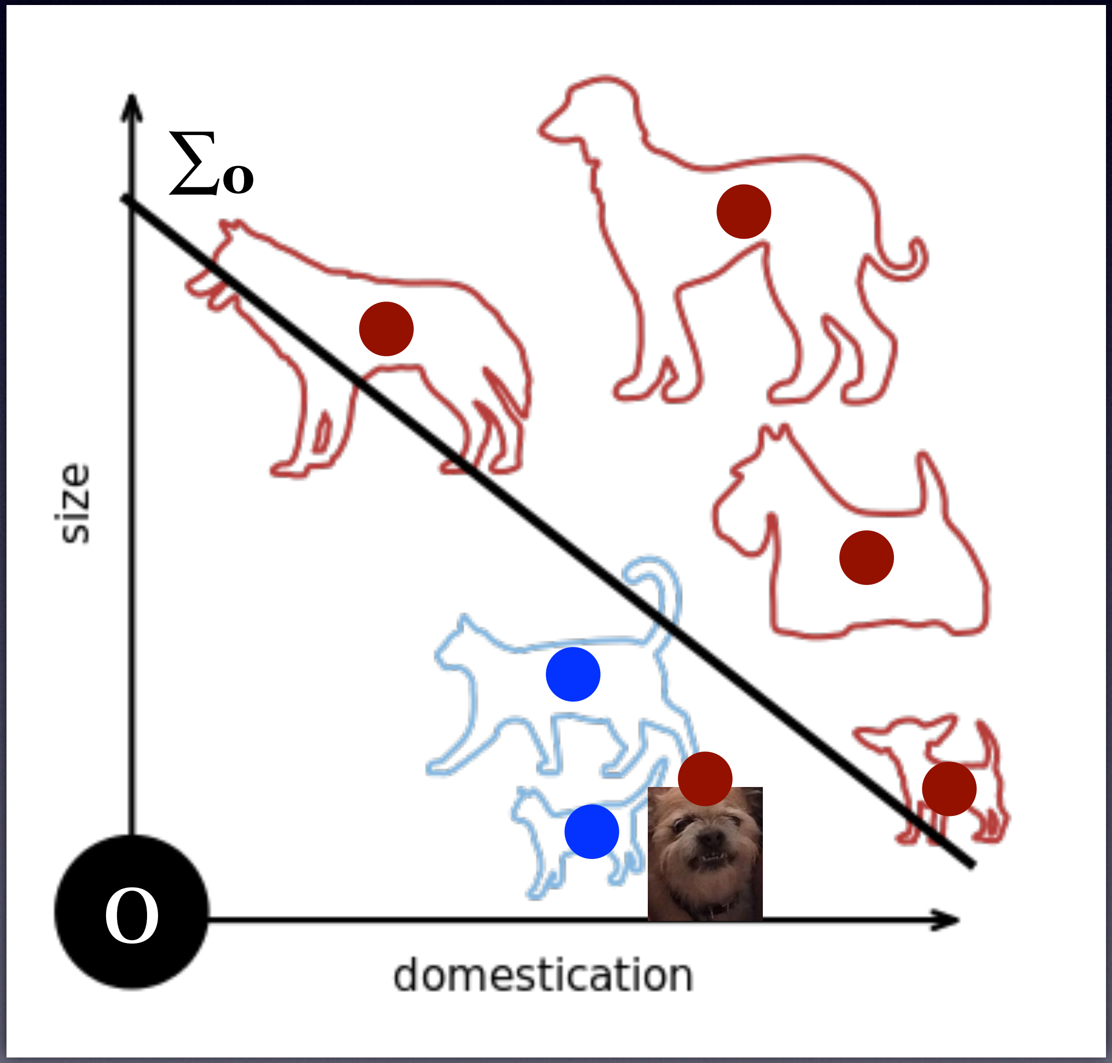


By picking a value for w and b, we define a boundary between the two sets of data

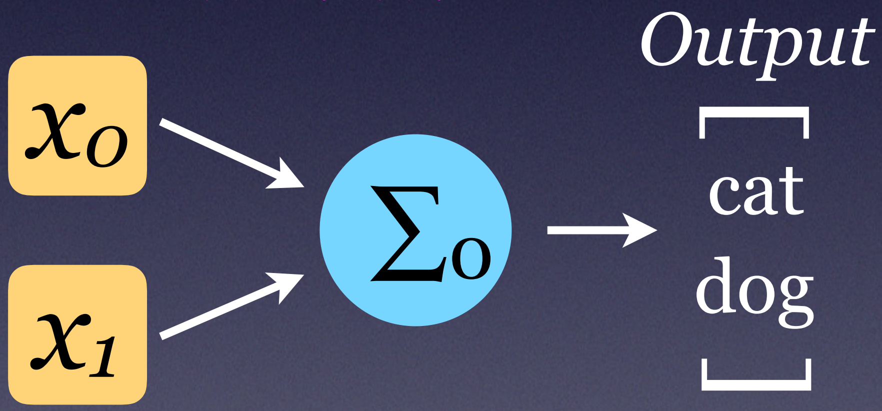
Introduction to CNNs (II)

Perceptron 2D Classification

Maybe we need to do better: assume new data point
(My friend's dog — small but not as well behaved)



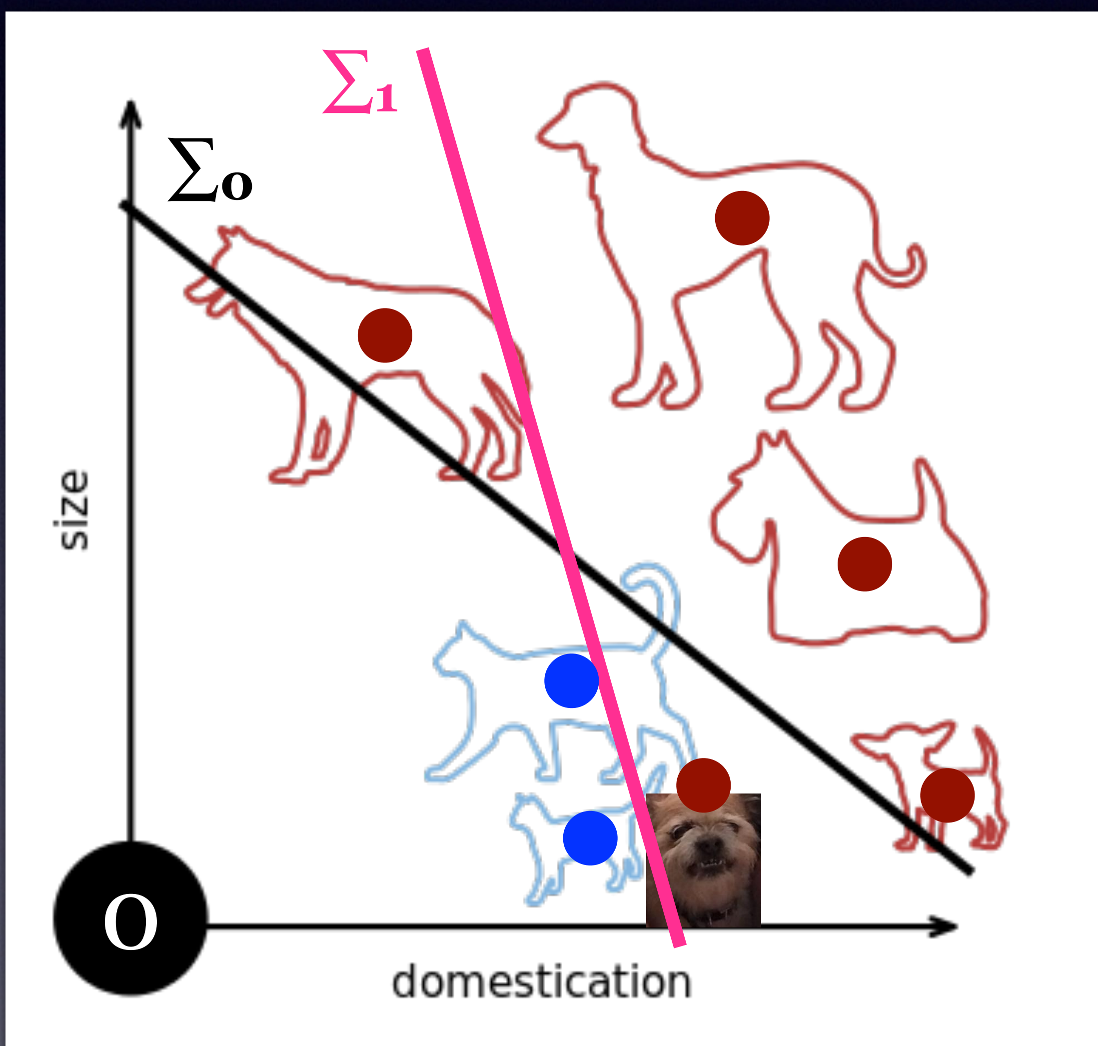
from [wikipedia]



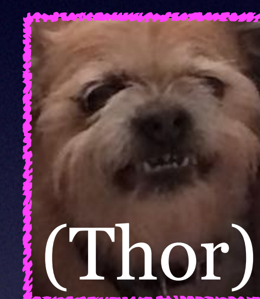
Introduction to CNNs (II)

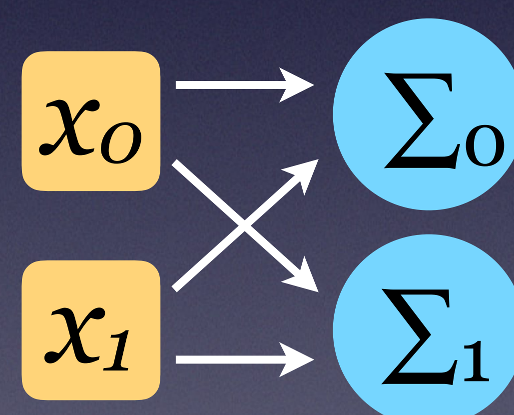
Perceptron 2D Classification

Maybe we need to do better: assume new data point (My friend's dog — small but not as well behaved)



from [wikipedia]()



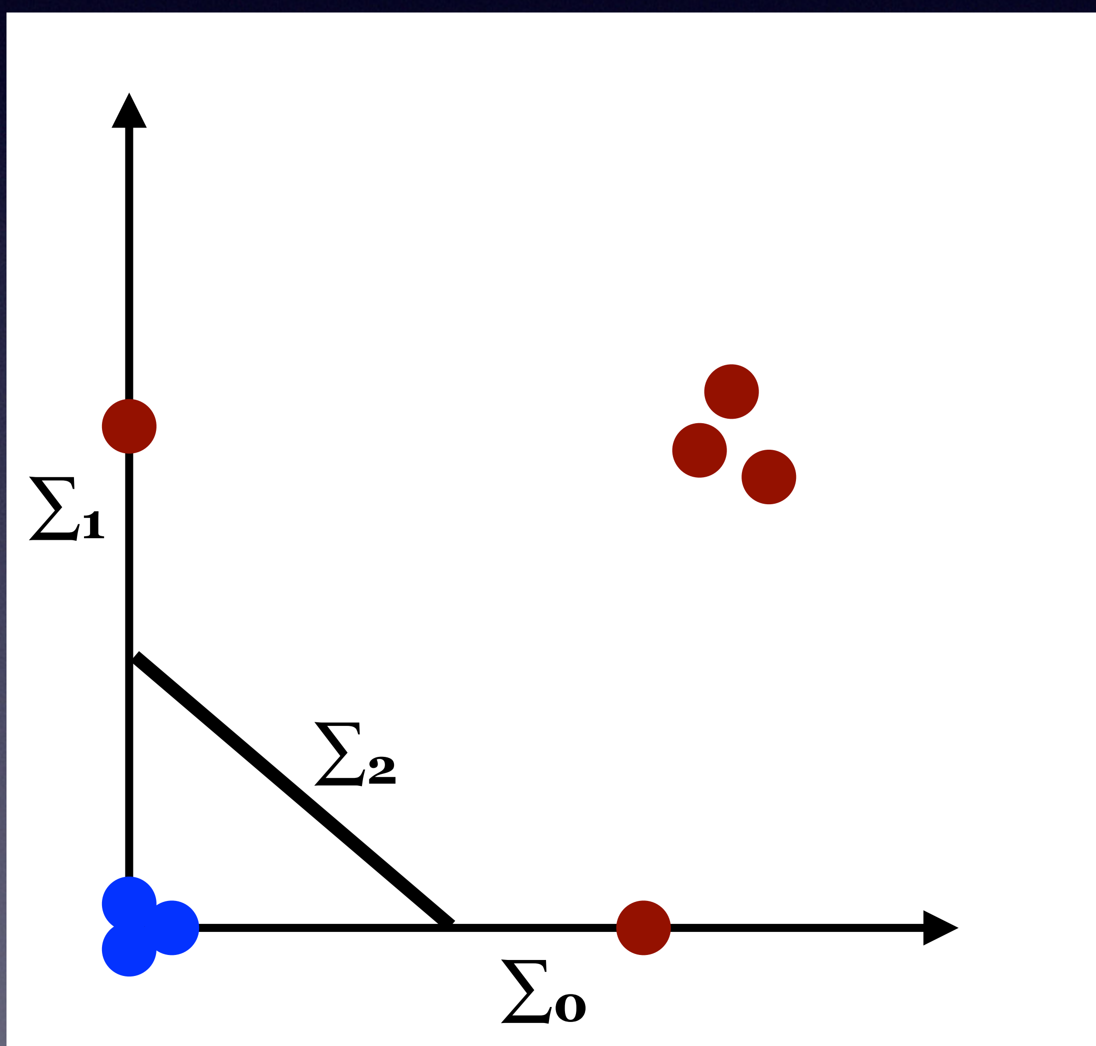


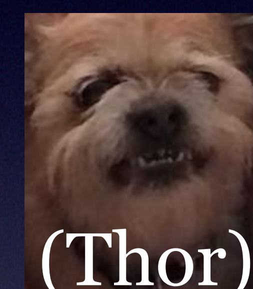
We can add another perceptron to help (but does not yet solve the problem)

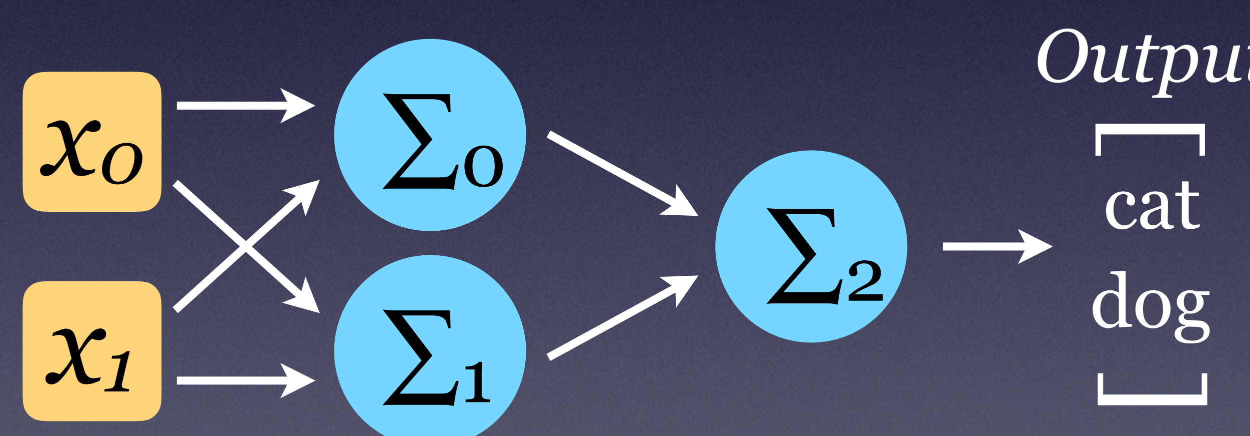
Introduction to CNNs (II)

Perceptron 2D Classification

Maybe we need to do better: assume new data point
(My friend's dog — small but not as well behaved)





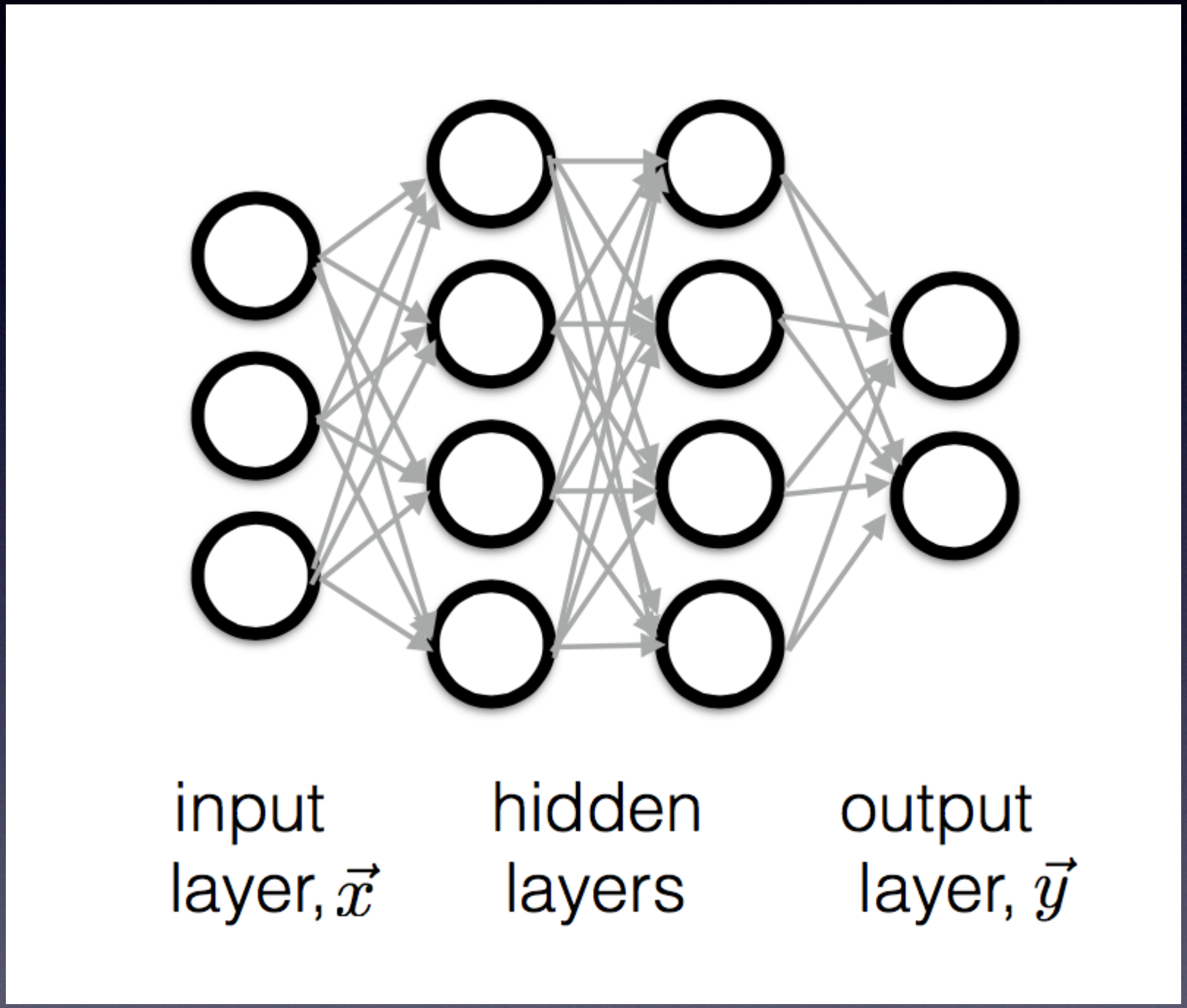


Another layer can classify based on preceding feature layer output

Introduction to CNNs (III)

"Traditional neural net" in HEP

Fully-Connected Multi-Layer Perceptrons



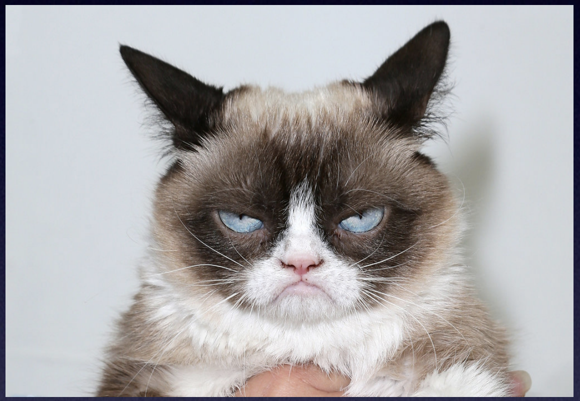
A traditional neural network consists of a stack of layers of such neurons where each neuron is ***fully connected*** to other neurons of the neighbor layers

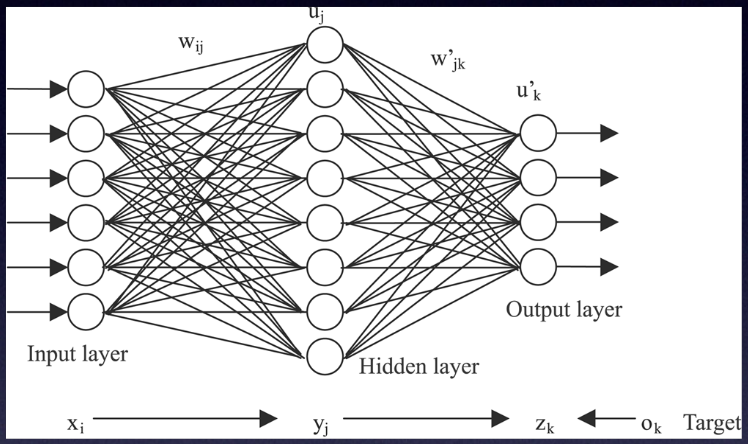
Introduction to CNNs (III)

"Traditional neural net" in HEP

Problems with it...

Feed in entire image

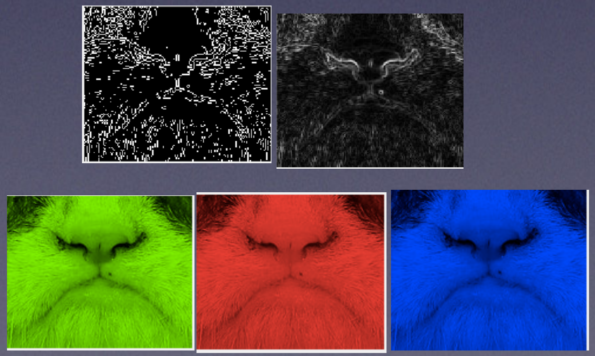


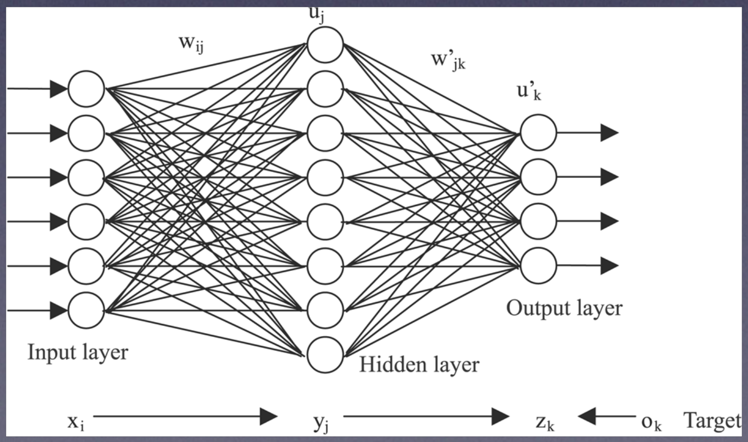


Cat?

Problem: scalability

Use pre-determined features



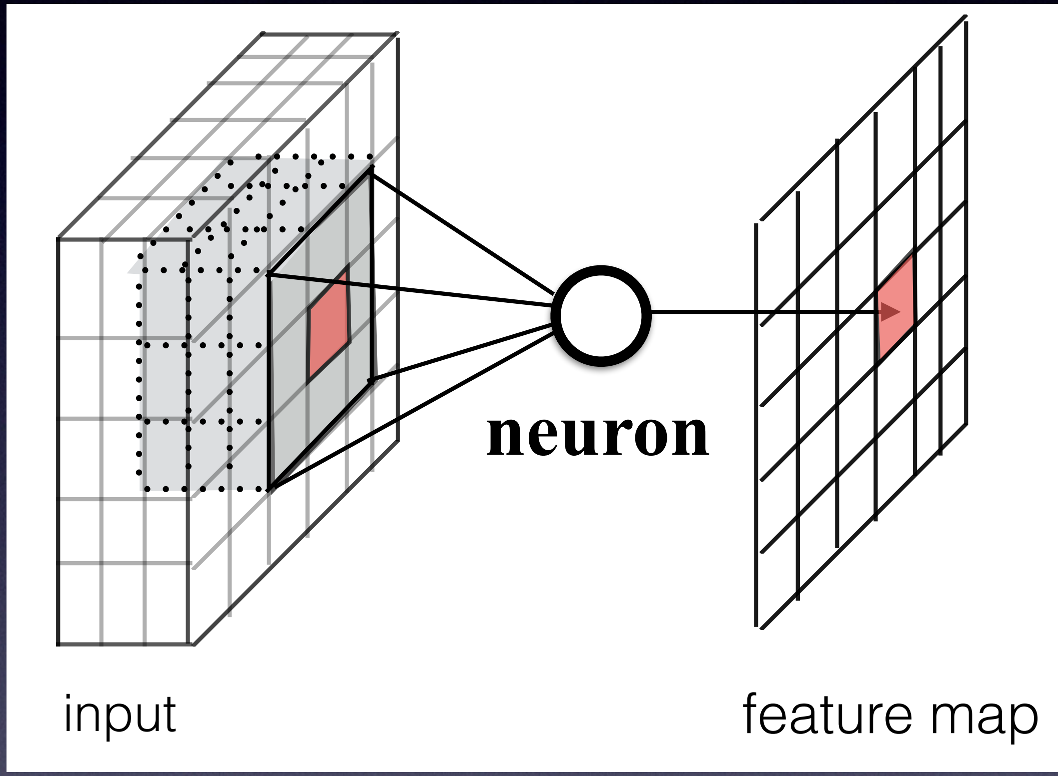


Cat?

Problem: generalization

Introduction to CNNs (III)

CNN introduce a ***limitation*** by forcing the network to look at only local, translation invariant features



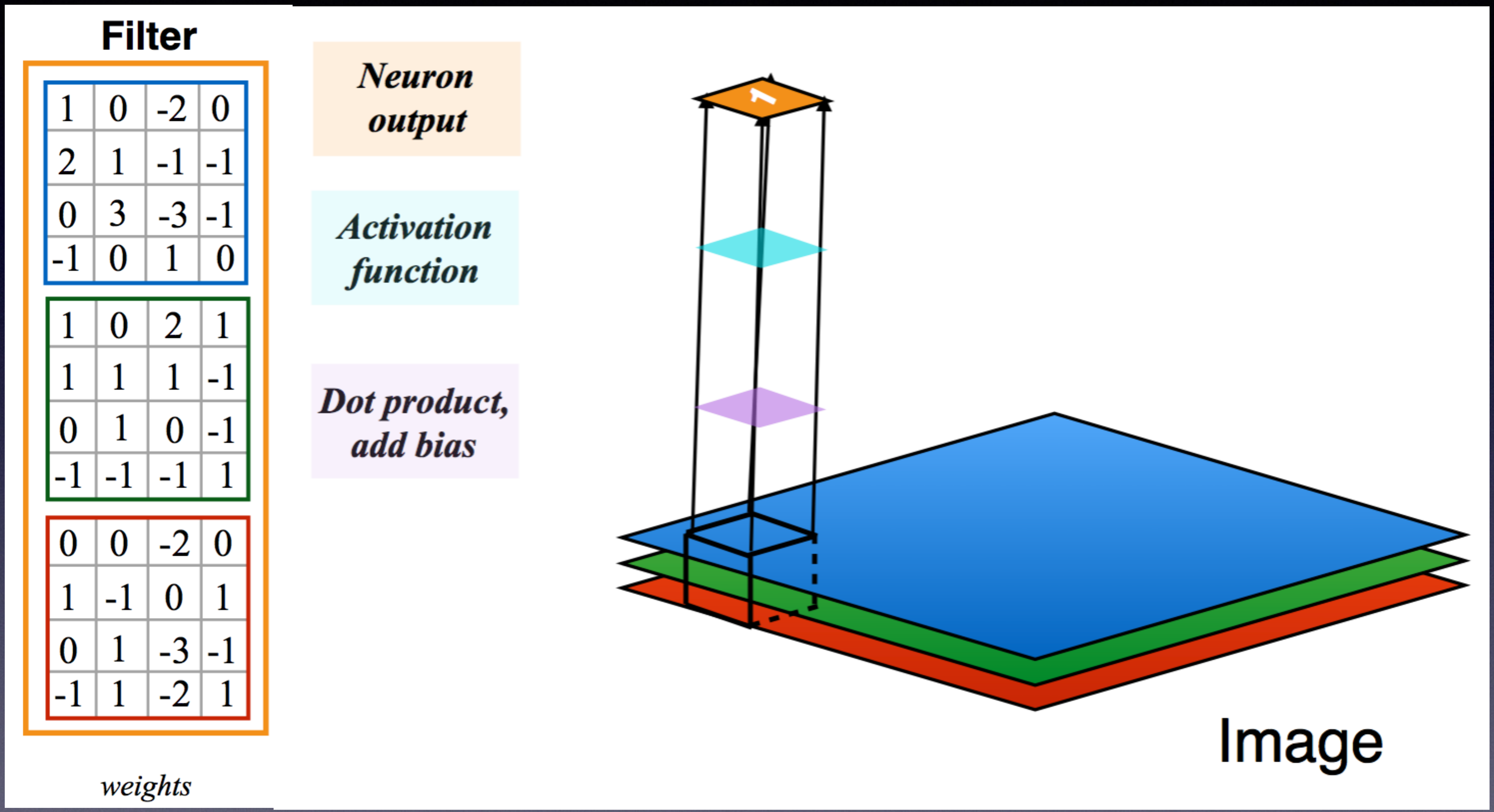
$$f_{i,j}(X) = \sigma\left(W_i \cdot X_j + b_i\right),$$

Activation of a neuron depends on the element-wise product of 3D weight tensor with 3D input data and a bias term

- Translate over 2D space to process the whole input
- Neuron **learns translation-invariant features**
- Applicable for a "homogeneous" detector like LArTPC

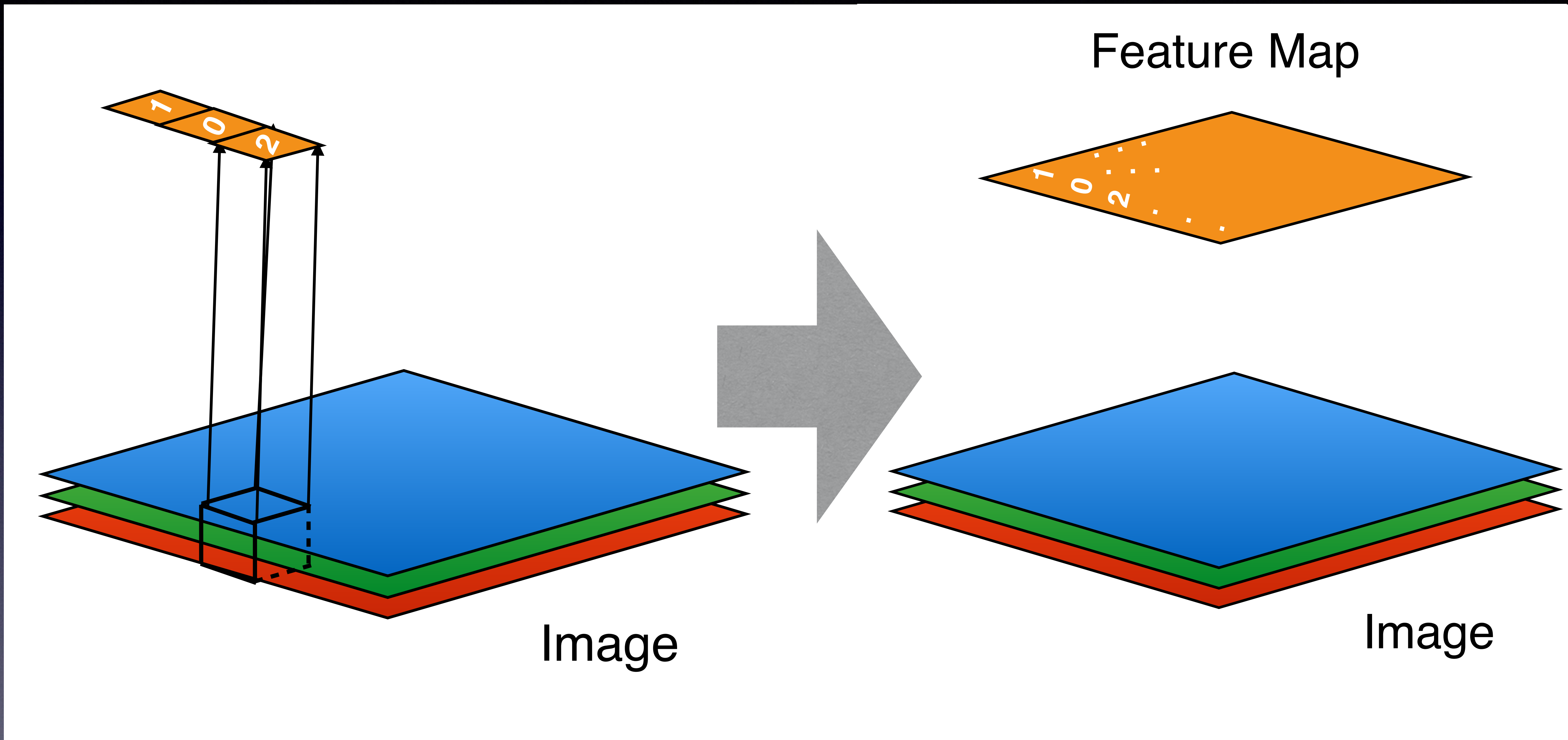
Want more details?
Feel free to ask me later!

Convolutional Neural Networks



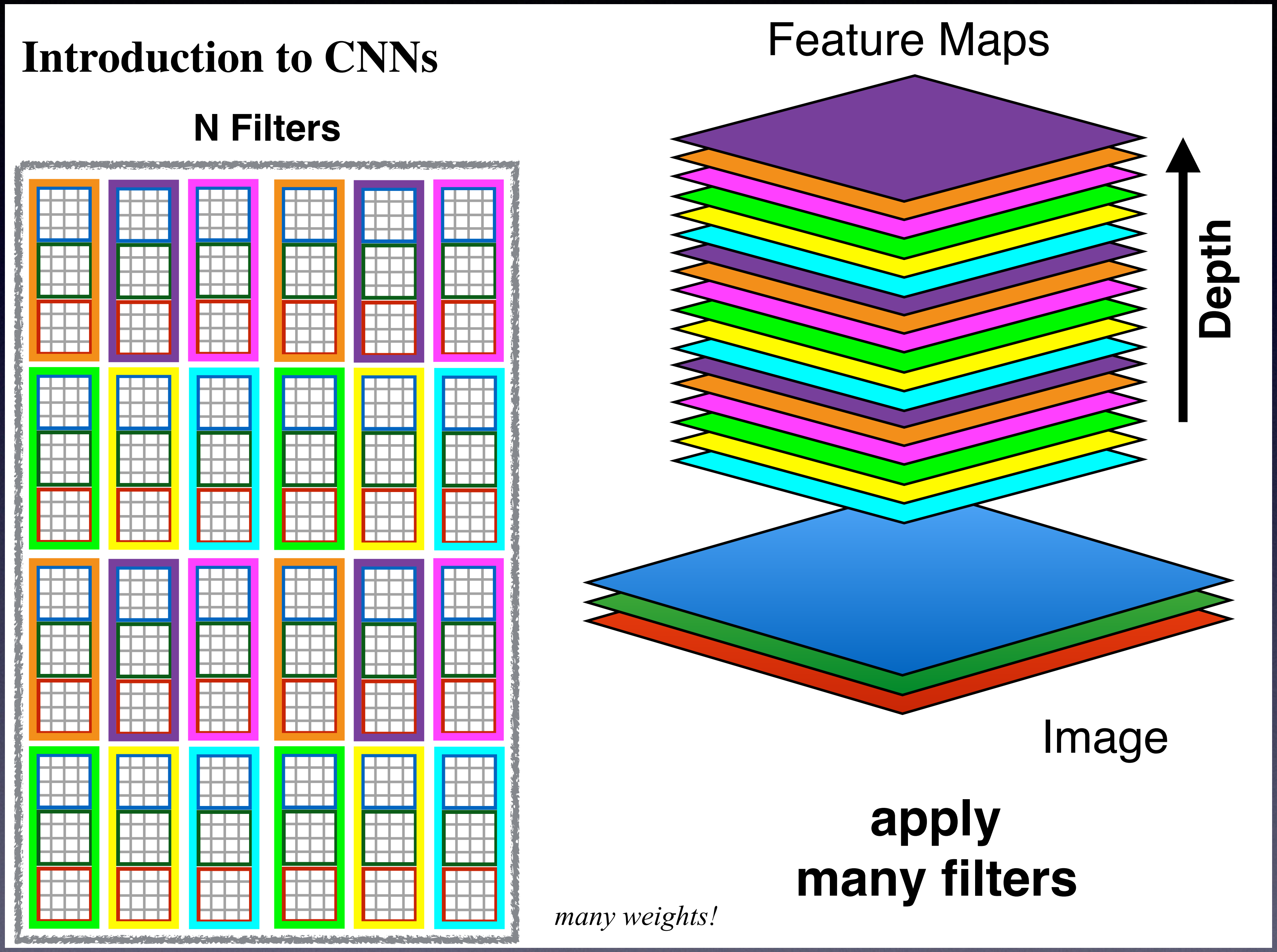
Toy visualization of the CNN operation

Convolutional Neural Networks



Toy visualization of the CNN operation

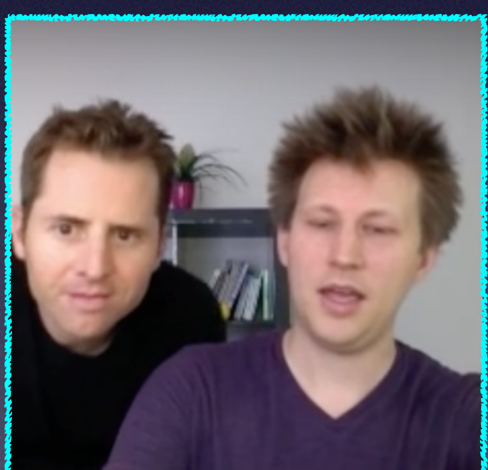
Convolutional Neural Networks

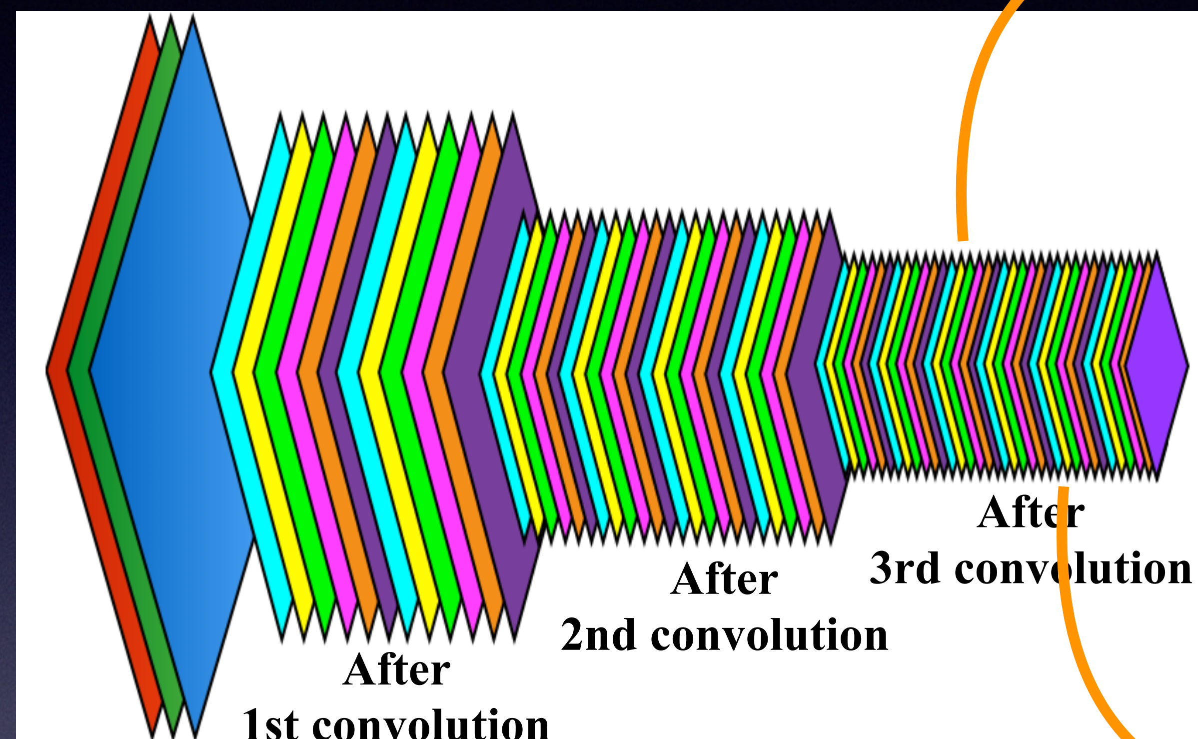


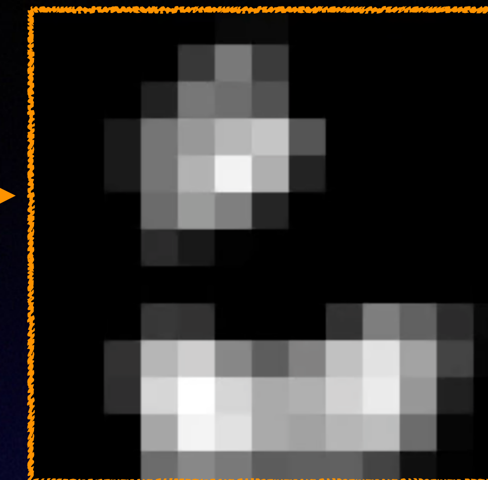
Toy visualization of the CNN operation

How Classification Network Works

Feature extraction by CNN

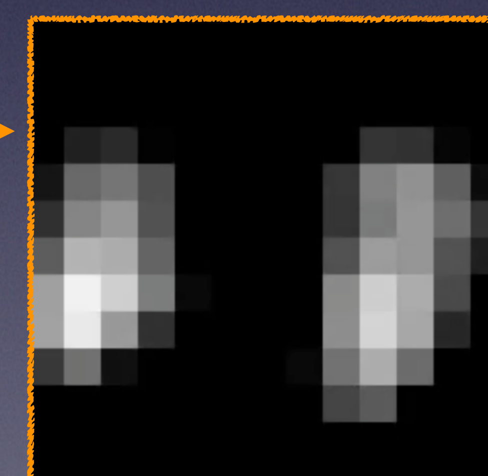






“Written Texts”
feature map

Feature extraction by CNN



“Human Face”
feature map

After steps of down-sampling,
“feature map” still preserves a rough object location information

How SSNet Works

Goal: recover precise, pixel-level location of objects

1. **Up-sampling**
   - Expand spatial dimensions of feature maps
2. **Convolution**
   - Smoothing (interpolation) of up-sampled feature maps