Remote
Online
Sessions *for*
Emerging
Seismologists

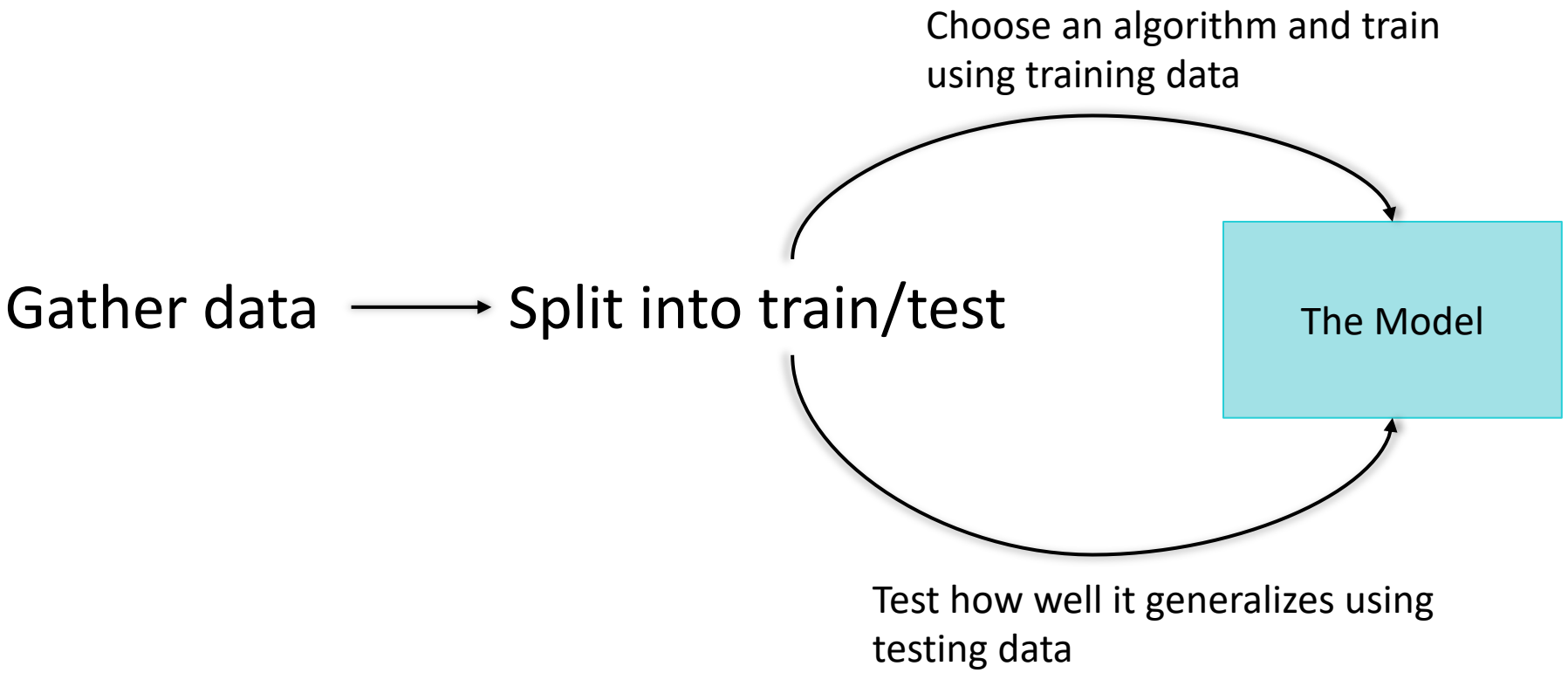
Introduction to Machine Learning

OMKAR RANADIVE

APPLIED SCIENTIST, ALCHERA LABS

Basics of Machine Learning

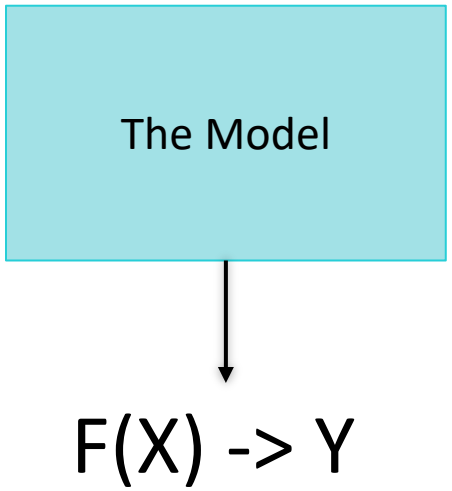
The Machine Learning Workflow



What exactly is “data” in context of ML?

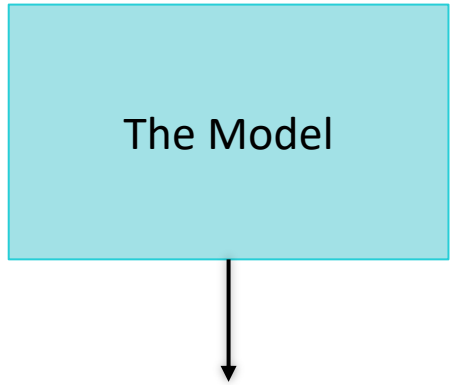
- Data samples should be made up of “features”
- These features should be good descriptors of what we are trying to model
- Example:
  - Features for House price prediction: Size of house, locality, number of bedrooms/bathrooms etc
- If these data samples have output labels associated with them, then it’s supervised learning
- If data is unlabeled, it is unsupervised learning

What does a model learn?



A model learns some function f, such that given an input X, it can map it to some output Y

What does a model learn?



Here learning corresponds to learning the values of these parameters (bias and intercept)

W0 + W1*X -> Y

(Linear Regression)

What does a model learn?

- This idea can be extended to multiple parameters
- These parameters will give weightage to the different features of the model

Example: Linear model with 3 features

$$\mathbf{Y = W0 + W1*X1 + W2*X2 + W3*X3}$$

How are these parameters learned?

- Define a loss function and try to minimize the loss

Example – Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = Mean Squared Error

n = Number of data points

Y_i = observed label values

$\hat{Y}_i$ = predicted label values

Putting everything together

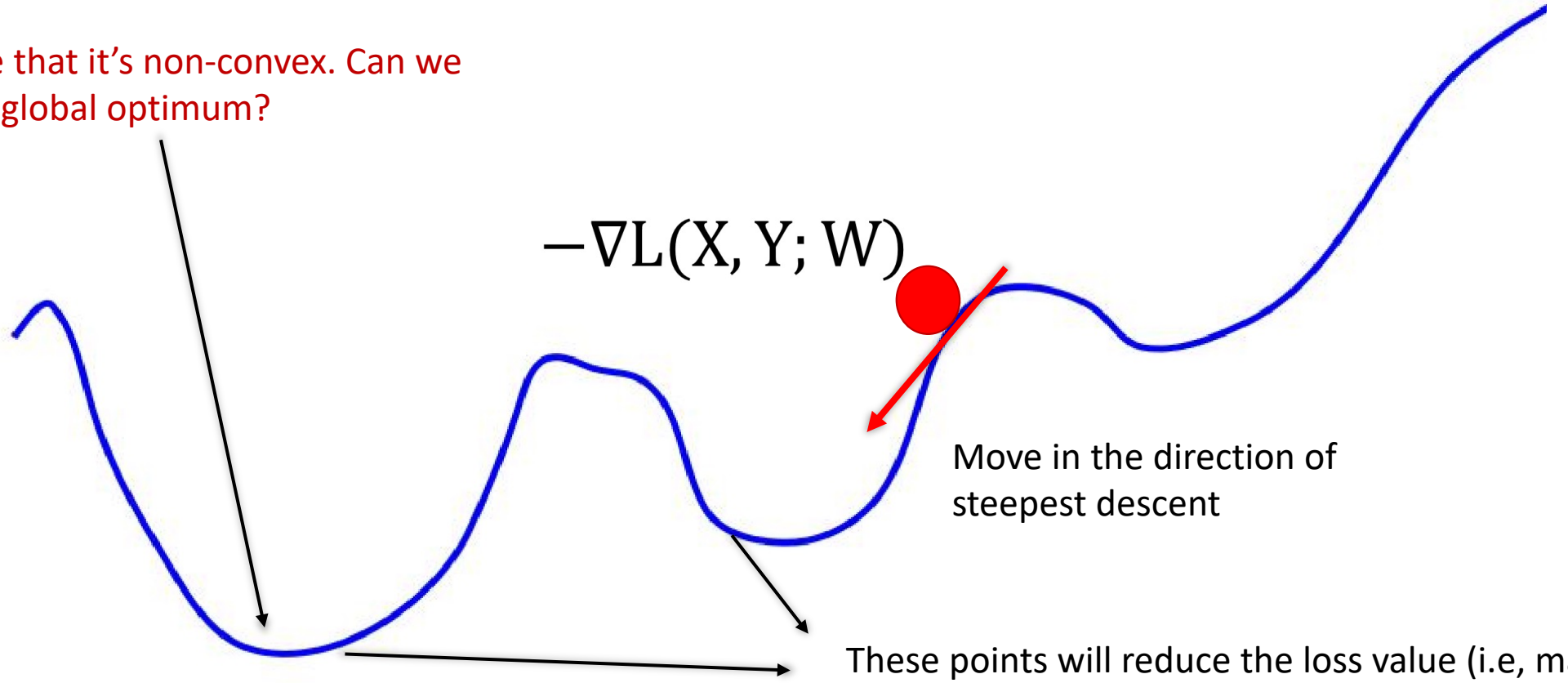
- Gather data X, labels Y, define a model M(W) and loss function L(X, Y; W)
- Initialize model parameters W
- Calculate the loss L w.r.t those parameters
- Update the parameters such that loss L is reduced

How?

- Keep doing this until convergence

Gradient Descent

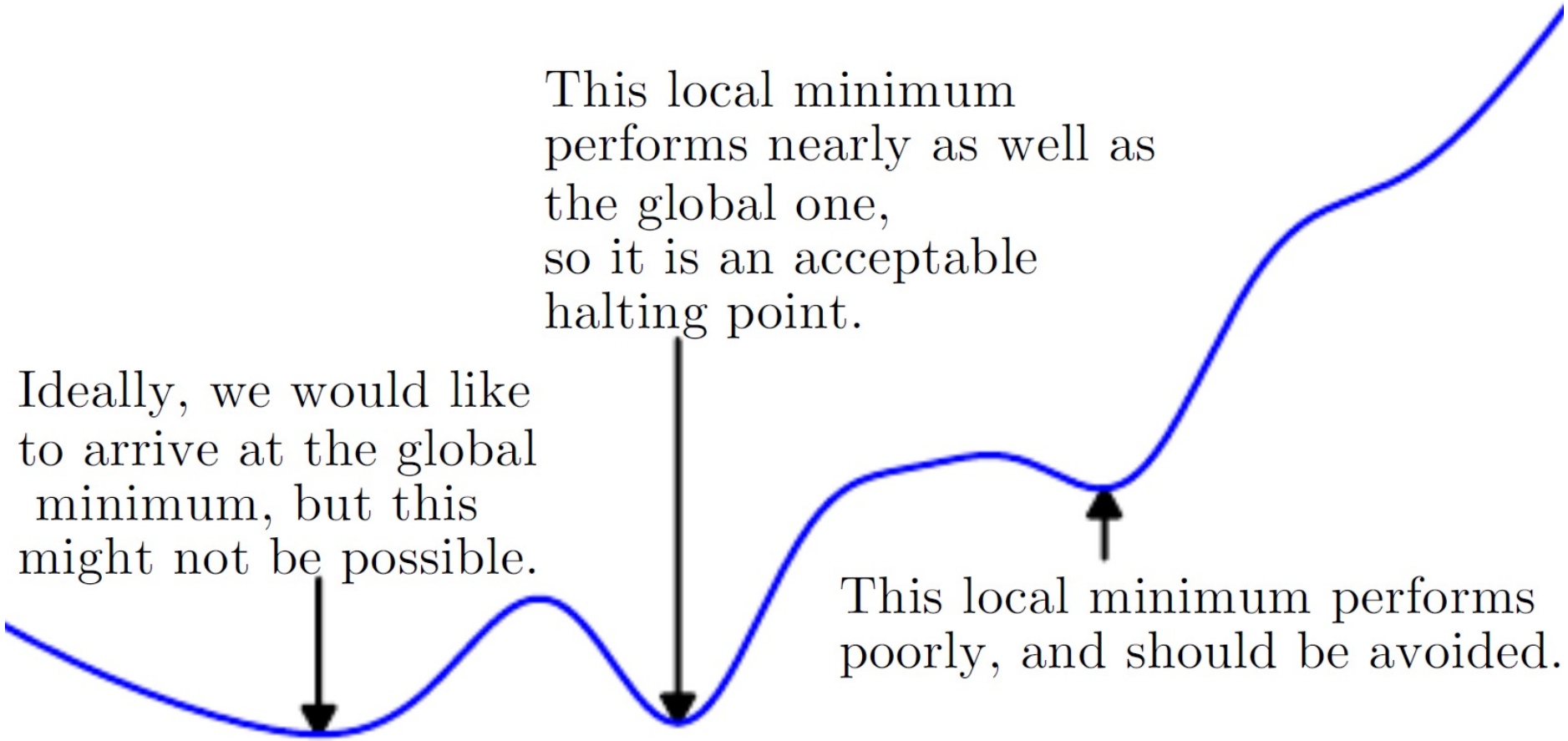
But notice that it's non-convex. Can we reach the global optimum?



Move in the direction of steepest descent

These points will reduce the loss value (i.e, make parameters fit our data better)

Gradient Descent



Main idea: Learn the distribution of the training data by minimizing the loss on it

Note: A model can only predict well on new (unseen) data if the data follows the training distribution

Example: The model won't work well if it is trained on *P* delay times recorded at the continent and used to predict *P* delay times at the bottom of the ocean

Traditional ML vs Neural Networks

Traditional Machine Learning

- Regression
- Logistic Regression
- Support Vector Machines
- ….

Neural Networks

- Fully Connected Networks
- Convolution Neural Networks
- Recurrent Neural Networks
- Generative Adversarial Networks
- ….

Neural Networks

Images used in this section are courtesy of Arden Dertat

What makes Neural Networks so special?

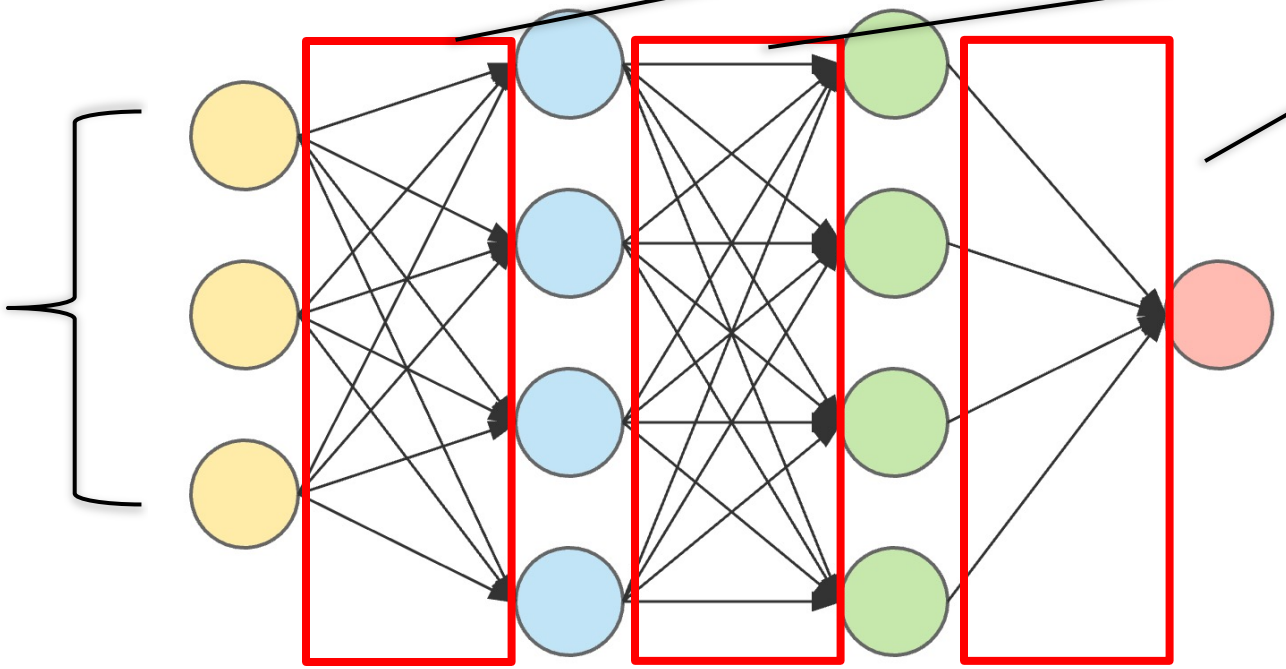
- They are universal function approximators; i.e., they can learn any function*
- They can be trained end-to-end

Fully Connected Network (FCN)

Every node is connected to every node in next layer (fully connected)

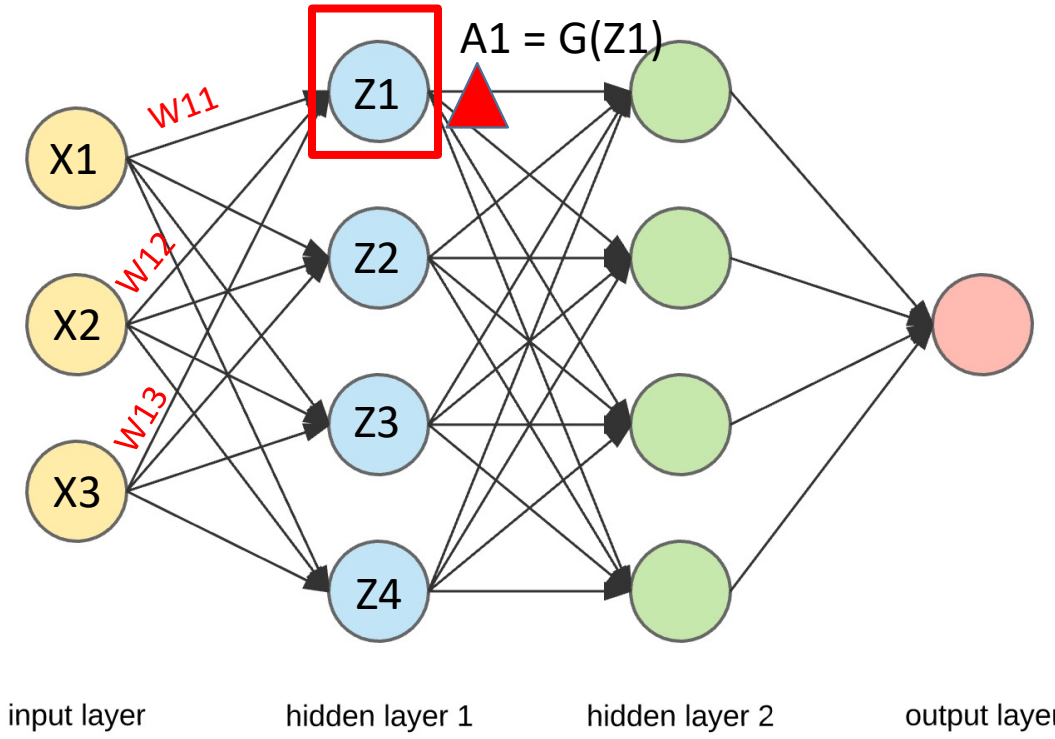
These weights/parameters (W) are learned by the model

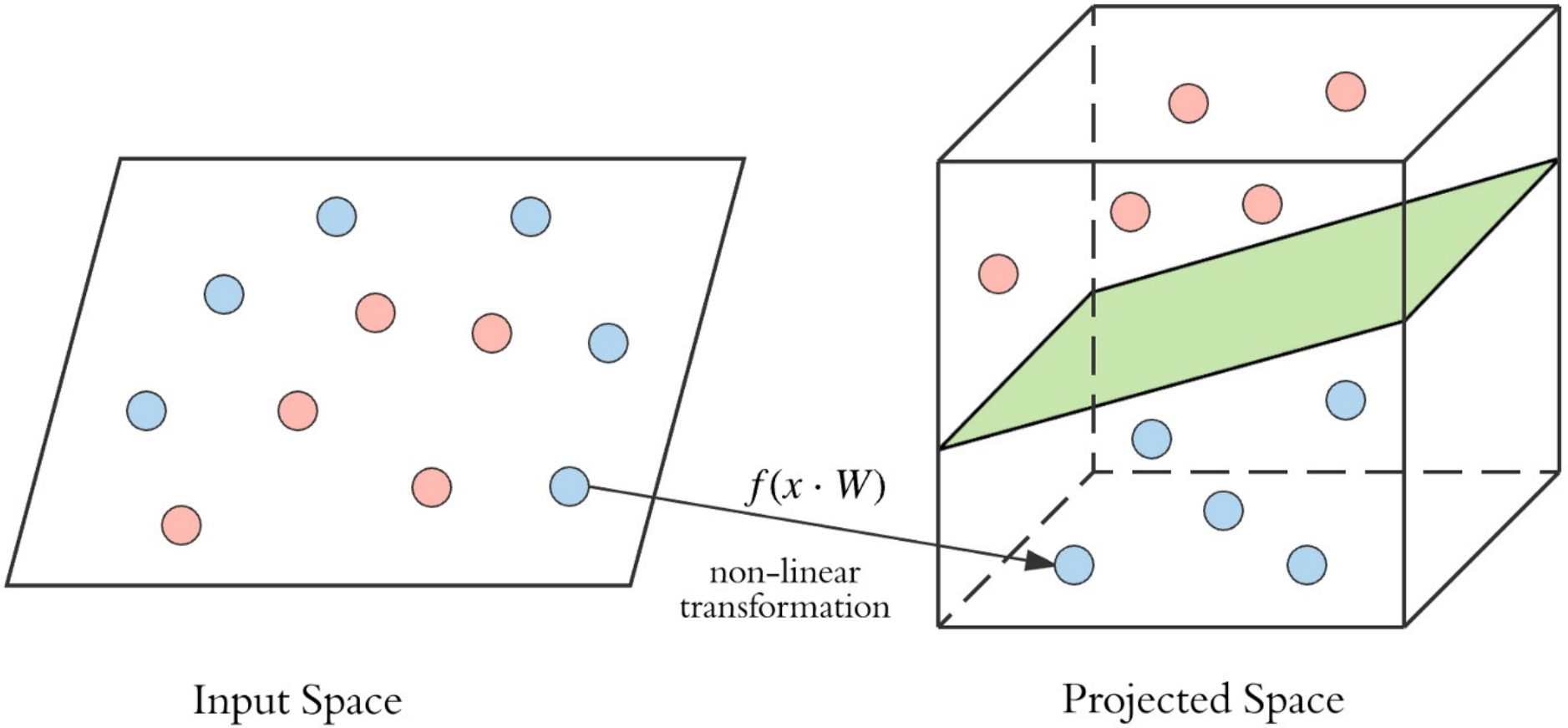
Input features (X1, X2, X3)



Fully Connected Network (FCN)

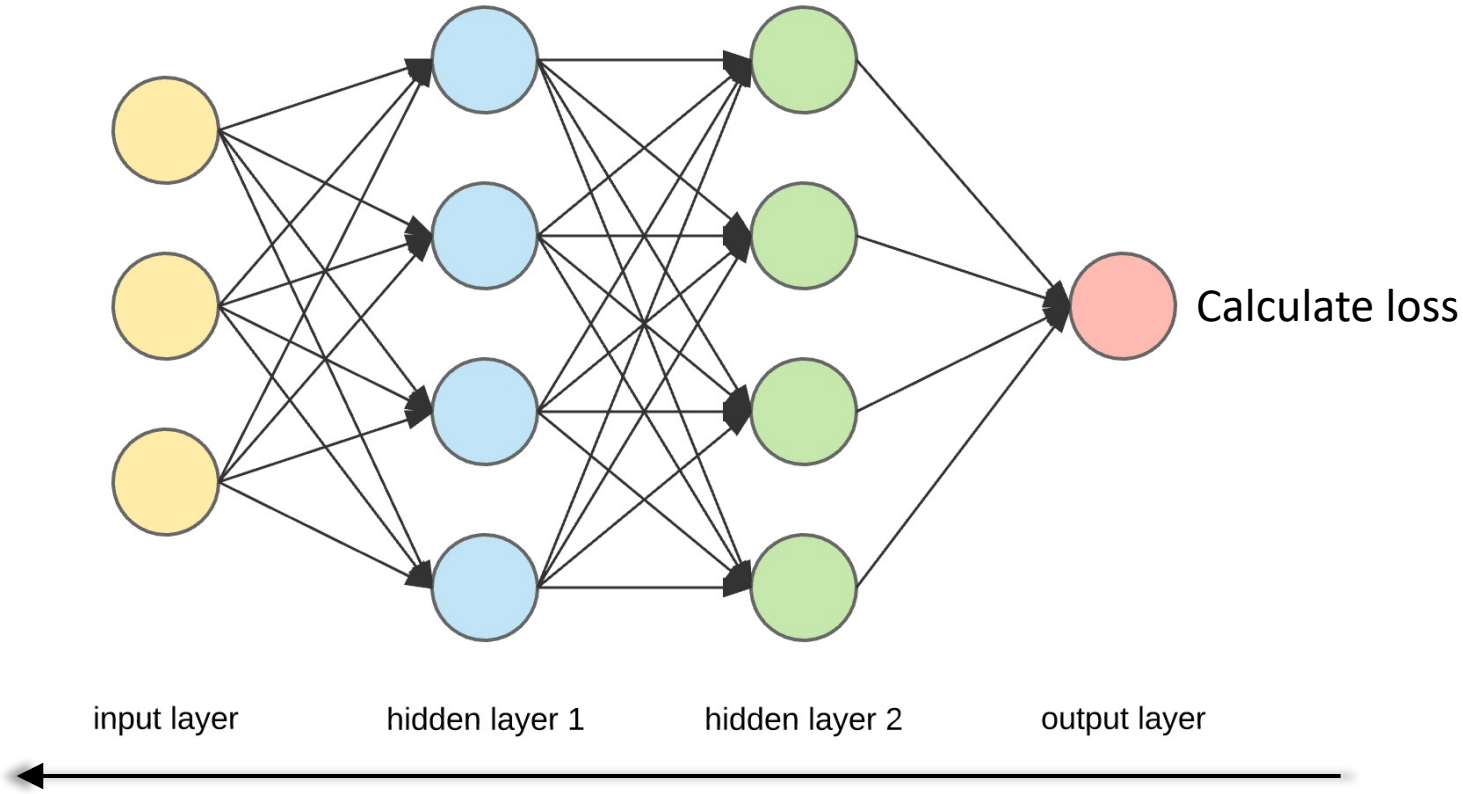
- Z1 = X1*W11 + X2*W12 + X3*W13
- A1 = G(Z1) where G is a non-linear function. Example – RELU, Sigmoid, tanh etc.
- These are called activation functions




$f(x \cdot W)$
non-linear
transformation
Input Space
Projected Space

Fully Connected Network (FCN)

Forward propagation, i.e., calculate values in a forward direction to get final output



Backward propagation (Learning phase): Flow gradient of the loss through all layers (chain rule of derivatives)

If FCNs are universal function approximators, why do we need other architectures like Convolution Neural Networks?

Convolution Neural Networks

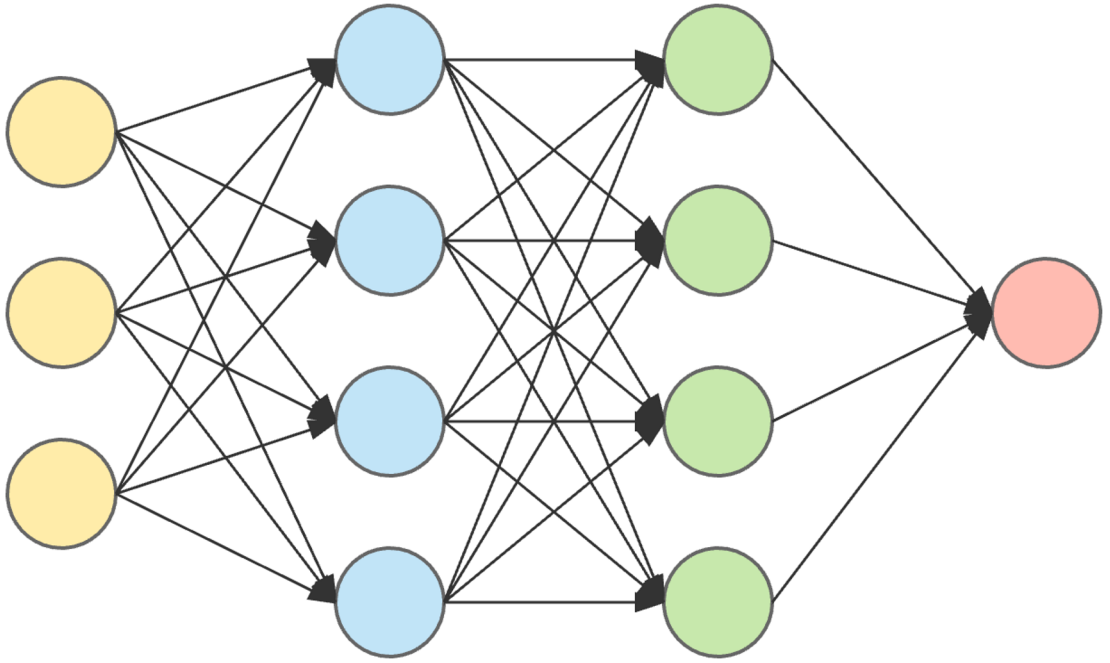
Some of the material in this section is adapted from Hung-Yi Lee's ML Course

Fully Connected Networks for Images

500 x 500

250,000 pixels

Inputs are pixels



input layer

250,000

hidden layer 1

100 nodes

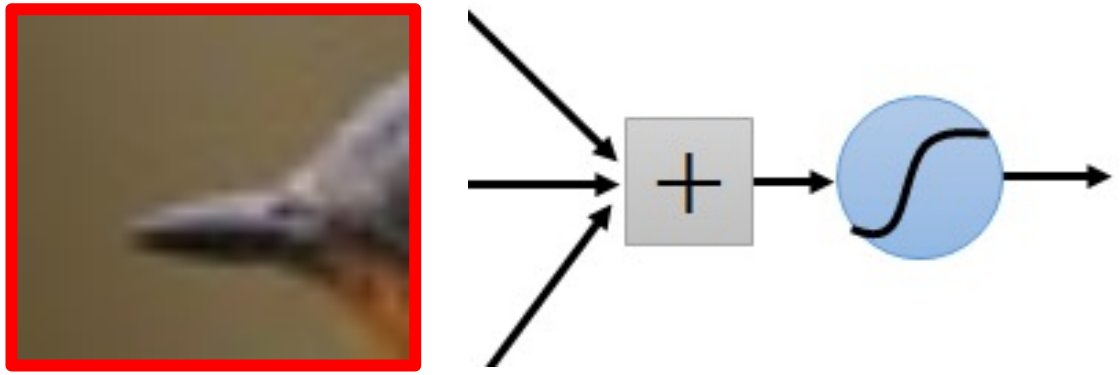
hidden layer 2

output layer

250k * 100 = 25 million weights

Why use CNNs?

- Some patterns are smaller than the whole image
- Neurons don’t need to see whole image to see the pattern



Why use CNNs?

- Same patterns appear in different regions

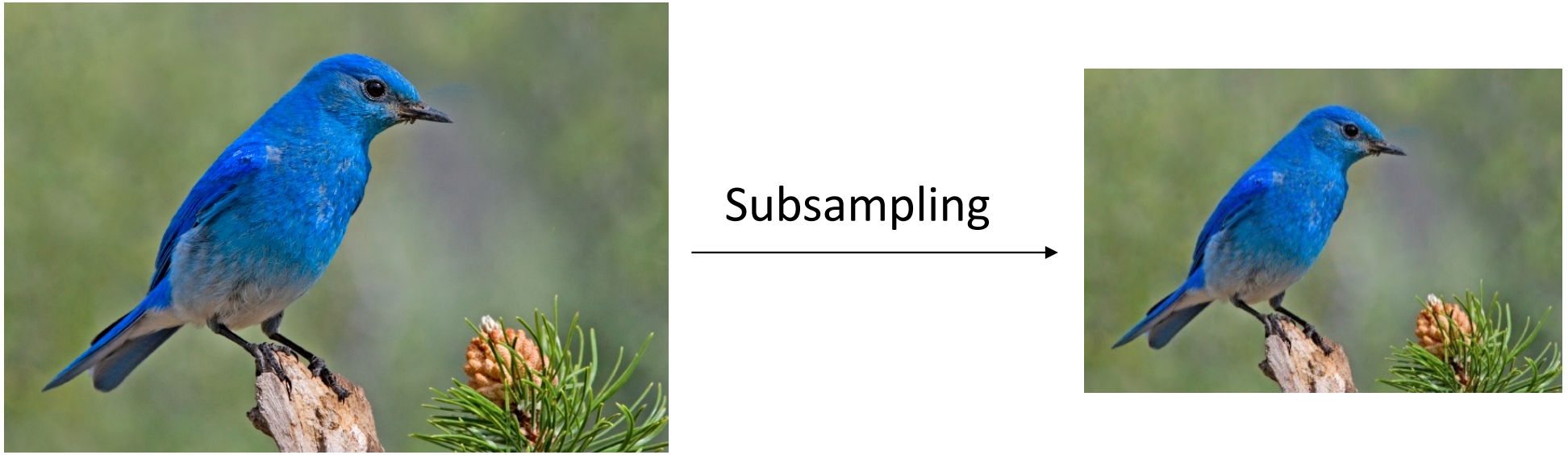
Upper left beak

Middle beak

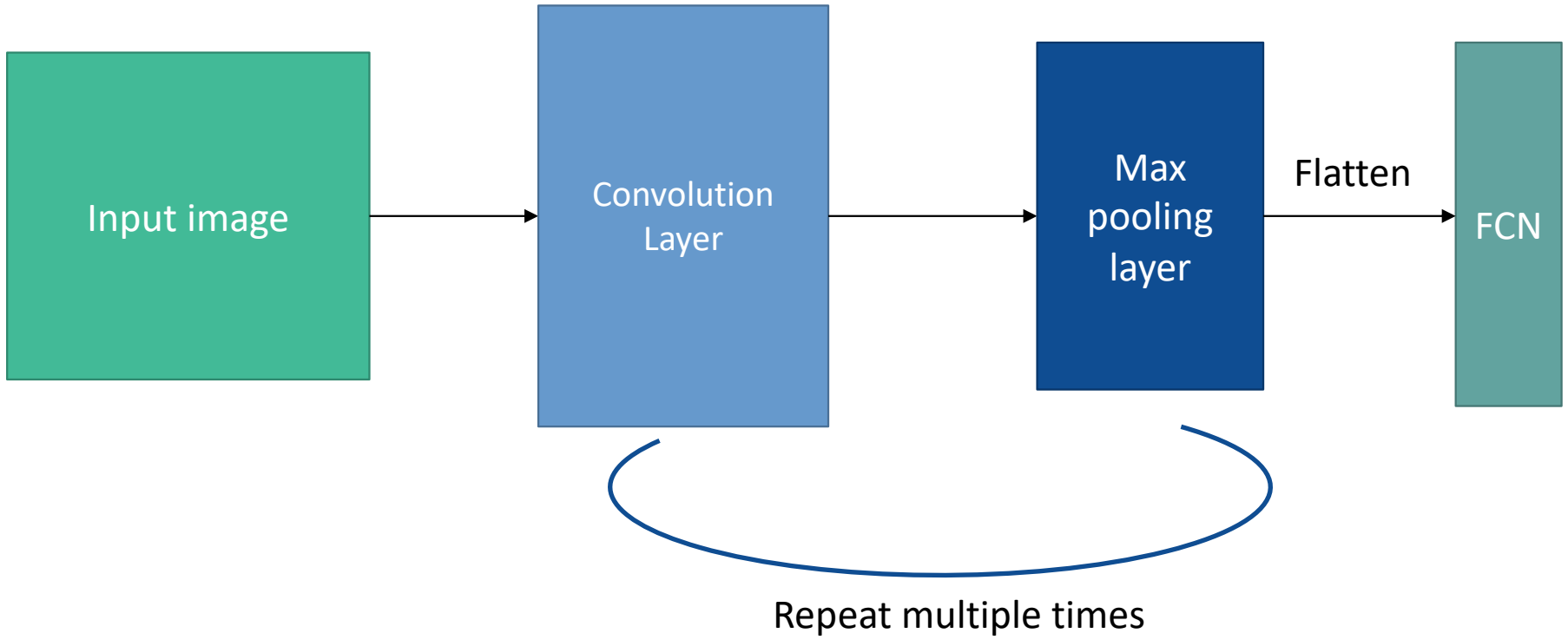
Can be detected using same set of parameters

Why use CNNs?

- Subsampling the image doesn’t change the object



The whole CNN



Main idea: A CNN leverages the properties of images

The convolution layer

- Takes advantage of –
  - Some patterns are smaller than the image
  - Same patterns appear in different regions
- Uses f x f filters to find these patterns (f < n)

What are these filters?

These filters were designed to hand-engineer features in the past

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

A CNN will find these filters by itself!

How?

Same idea as before; use gradient descent to minimize loss by updating parameters (filters)

Image courtesy of NASA

Applying filters

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

6 x 6 image

Filter matrices will be learned by the network

Different filter matrices will learn to extract different features

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1 Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2 Matrix

stride=1

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

6 x 6 image

3 -1

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

If stride=2

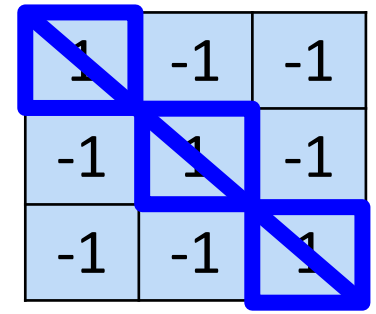
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Filter 1

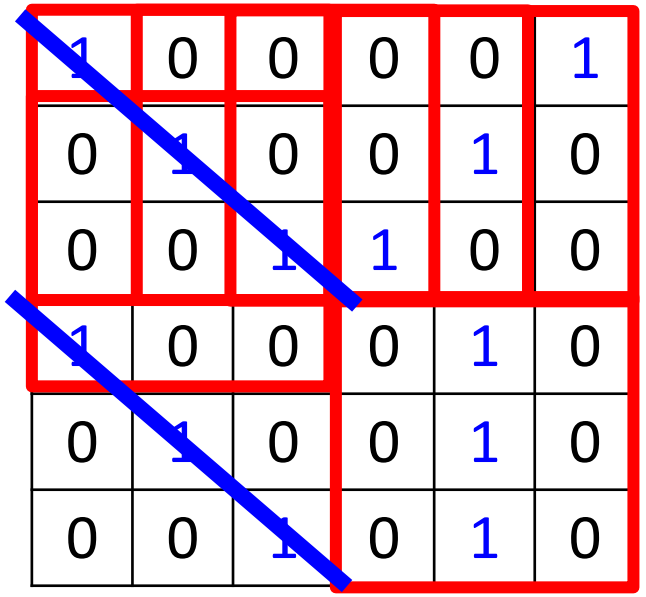
1	-1	-1
-1	1	-1
-1	-1	1

3 -3

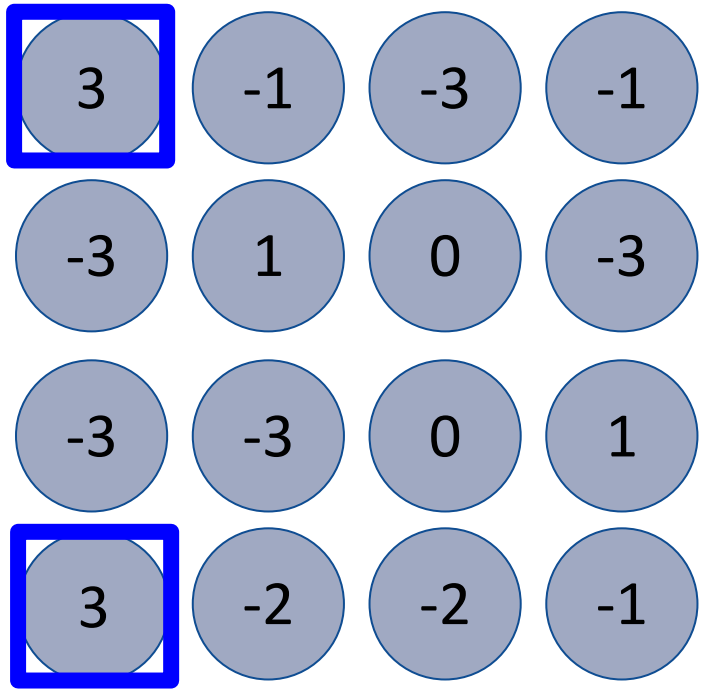


Filter 1

stride=1



6 x 6 image



Patterns are repeating

-1	1	-1
-1	1	-1
-1	1	-1

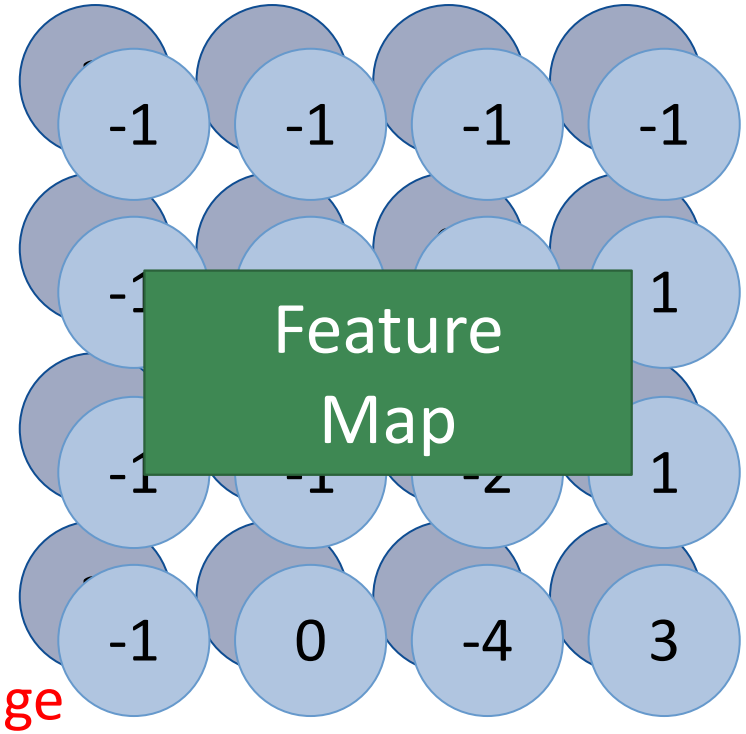
Filter 2

stride=1

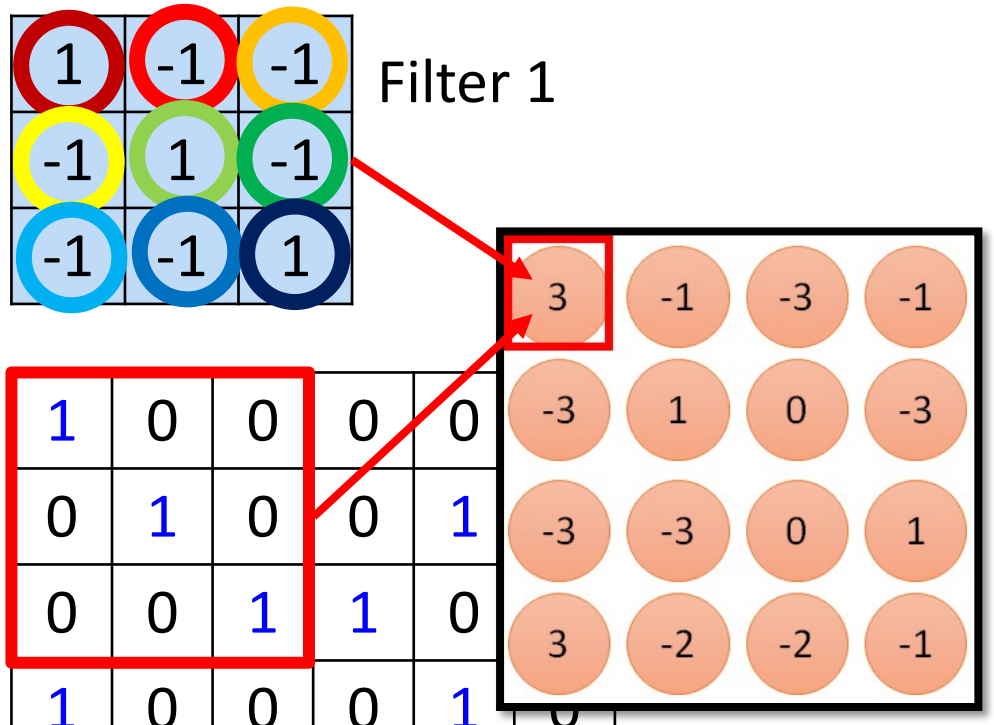
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

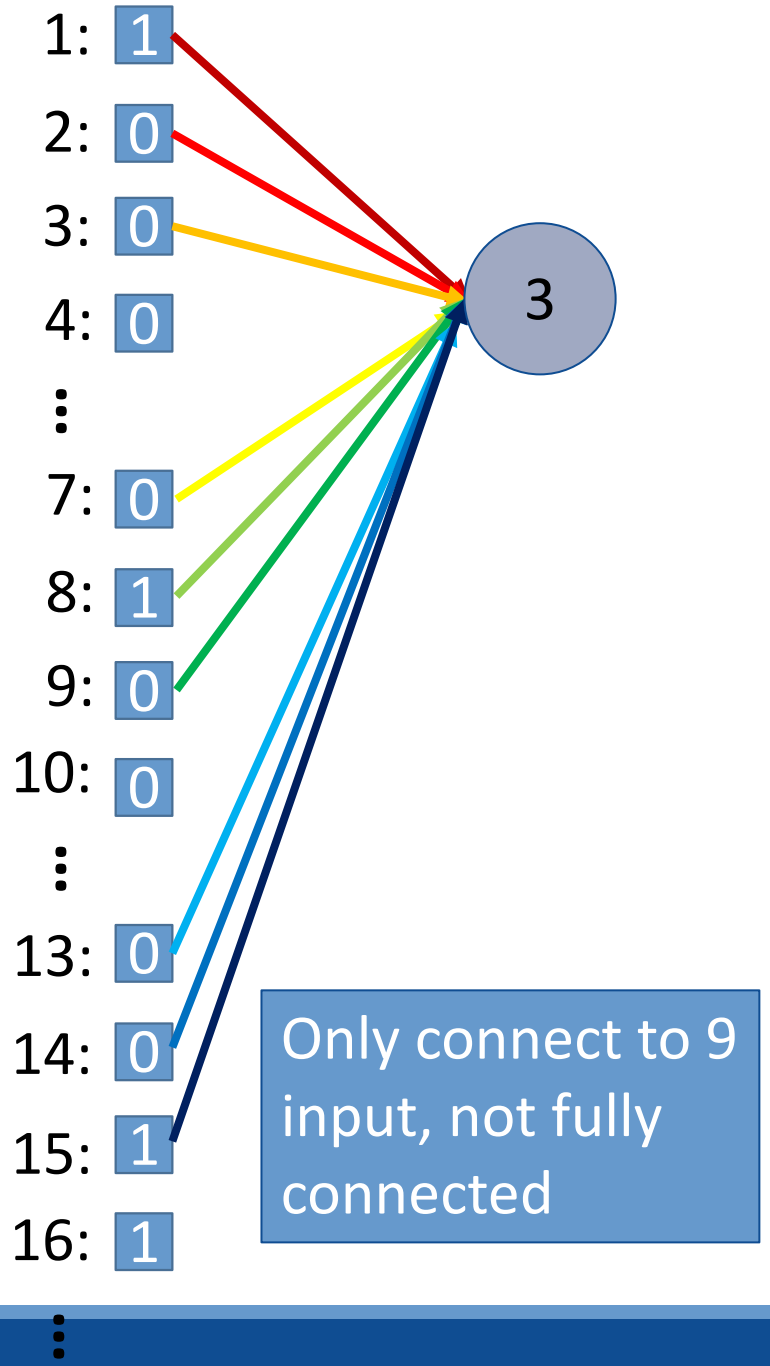
Do the same process for every filter



4 x 4 image

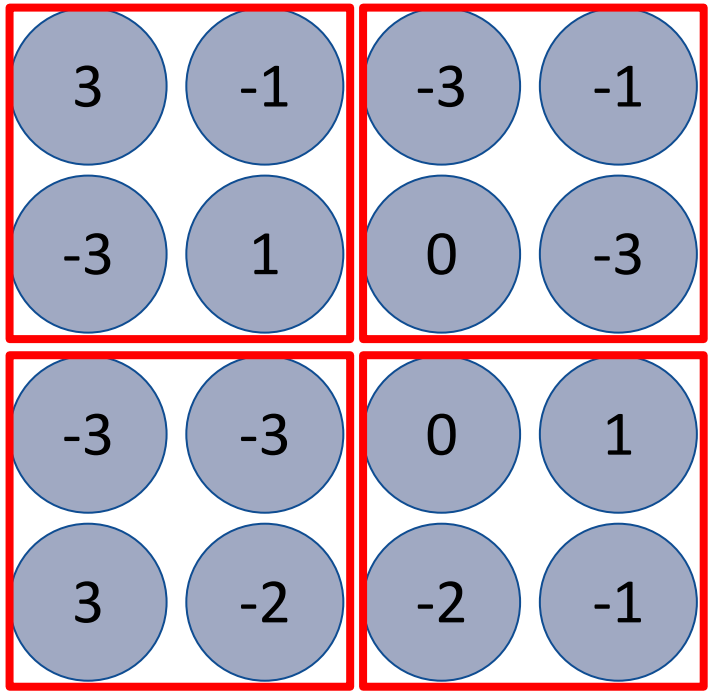


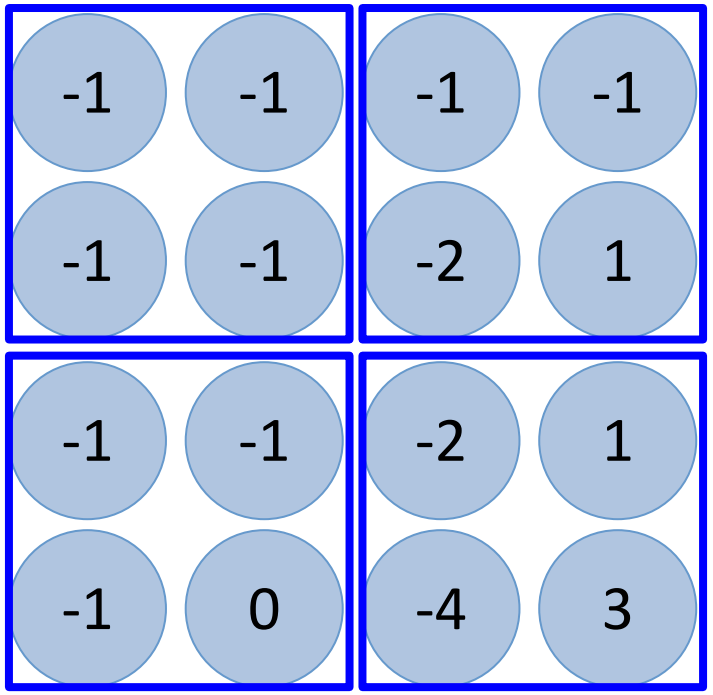
Less parameters!



Only connect to 9 input, not fully connected

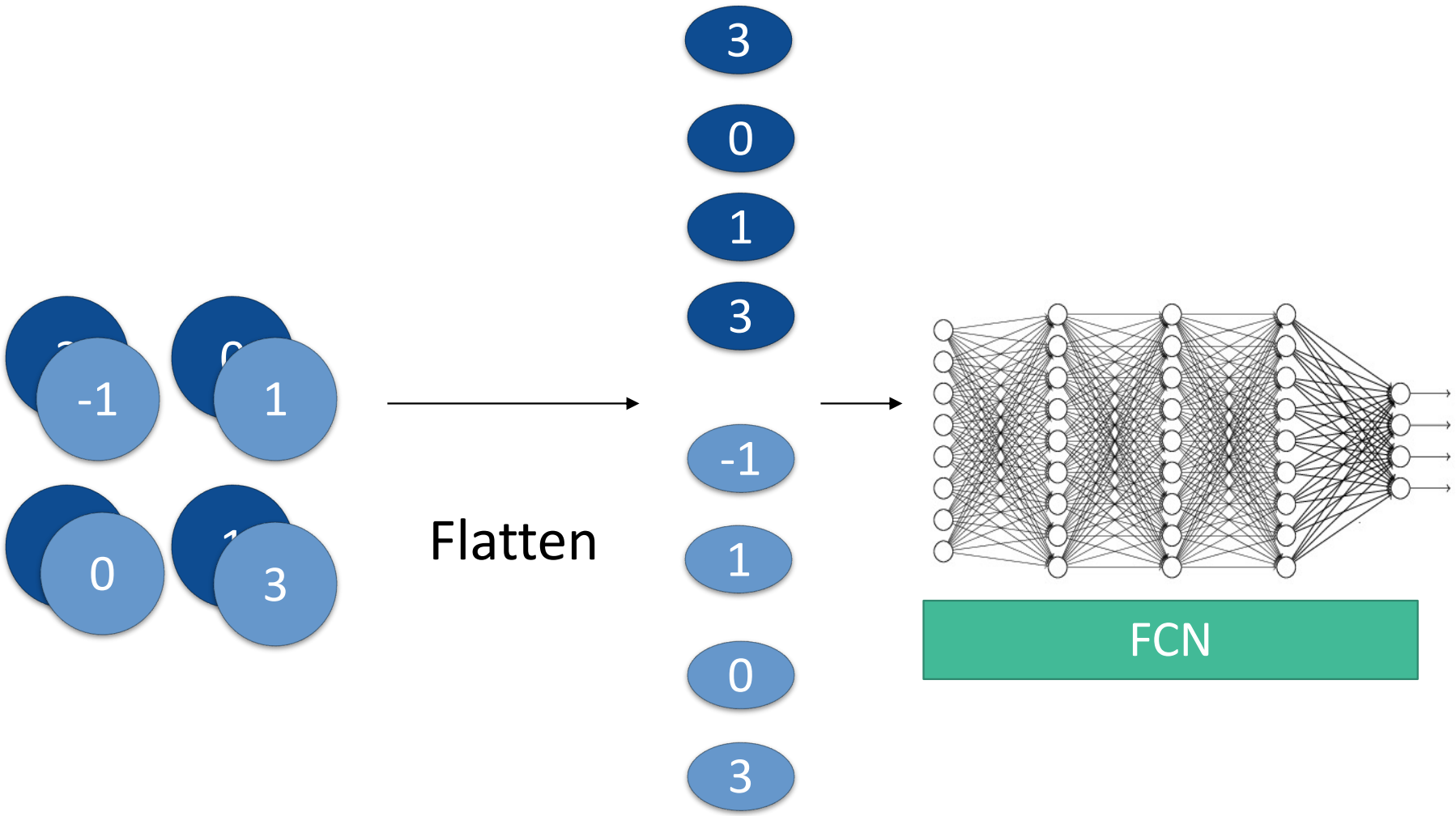
Max Pooling



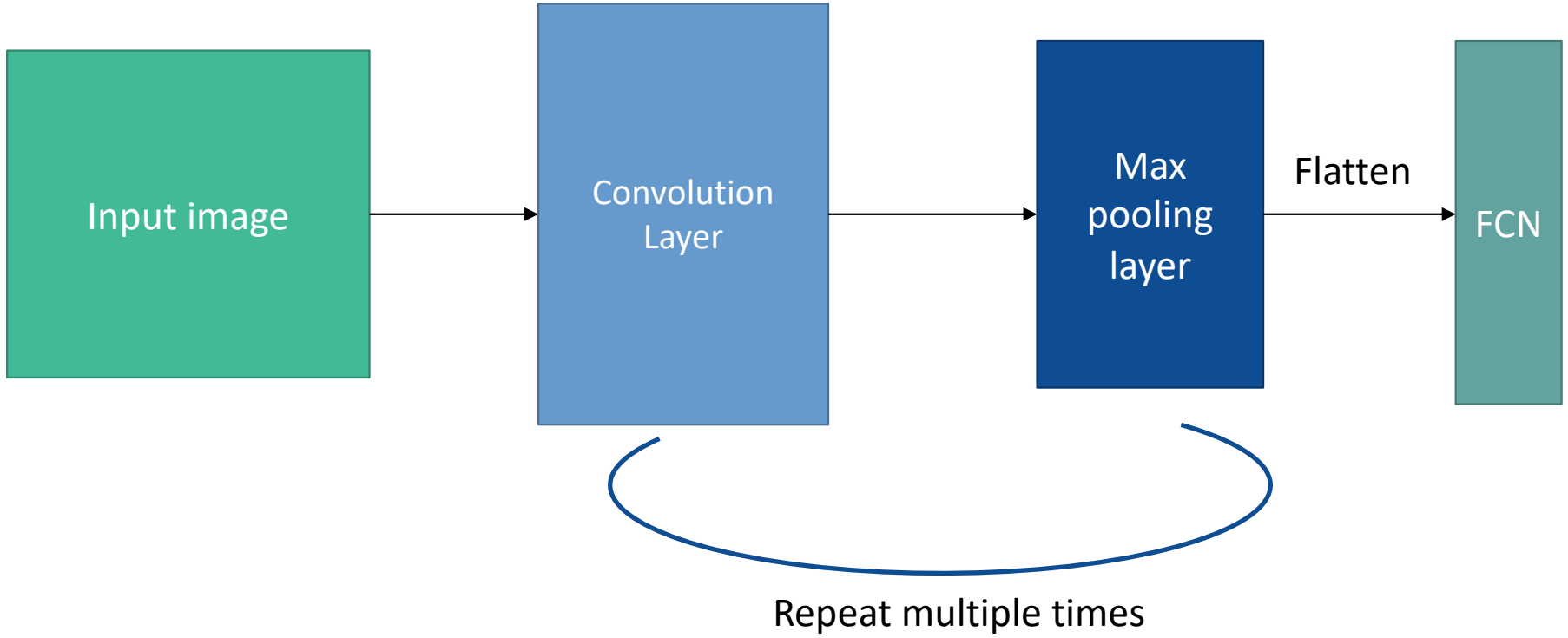


Down-sample the image; only keep the “sharpest” features (summary statistic)

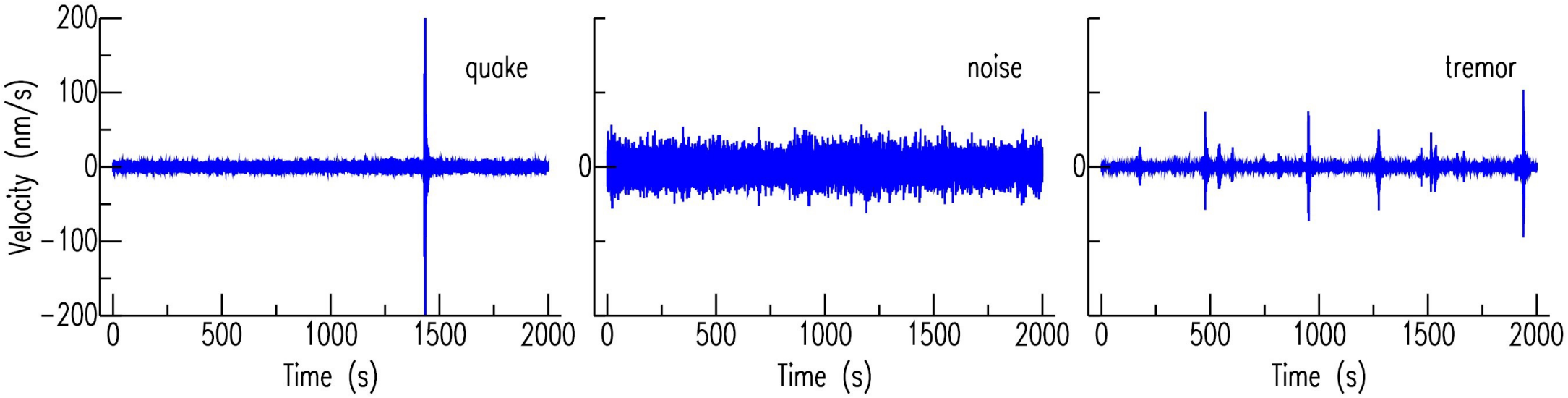
Flatten



The whole CNN



Use in seismology



Use 2d convolutions directly on images if patterns are clearly visible

Use in seismology

Use 1d convolutions over signals to detect patterns directly in the signal

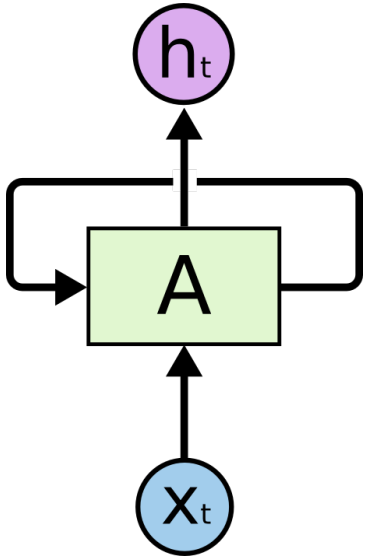
Example: A CNN can learn to detect an earthquake signal by itself by learning to extract useful features from the signal

Recurrent Neural Networks

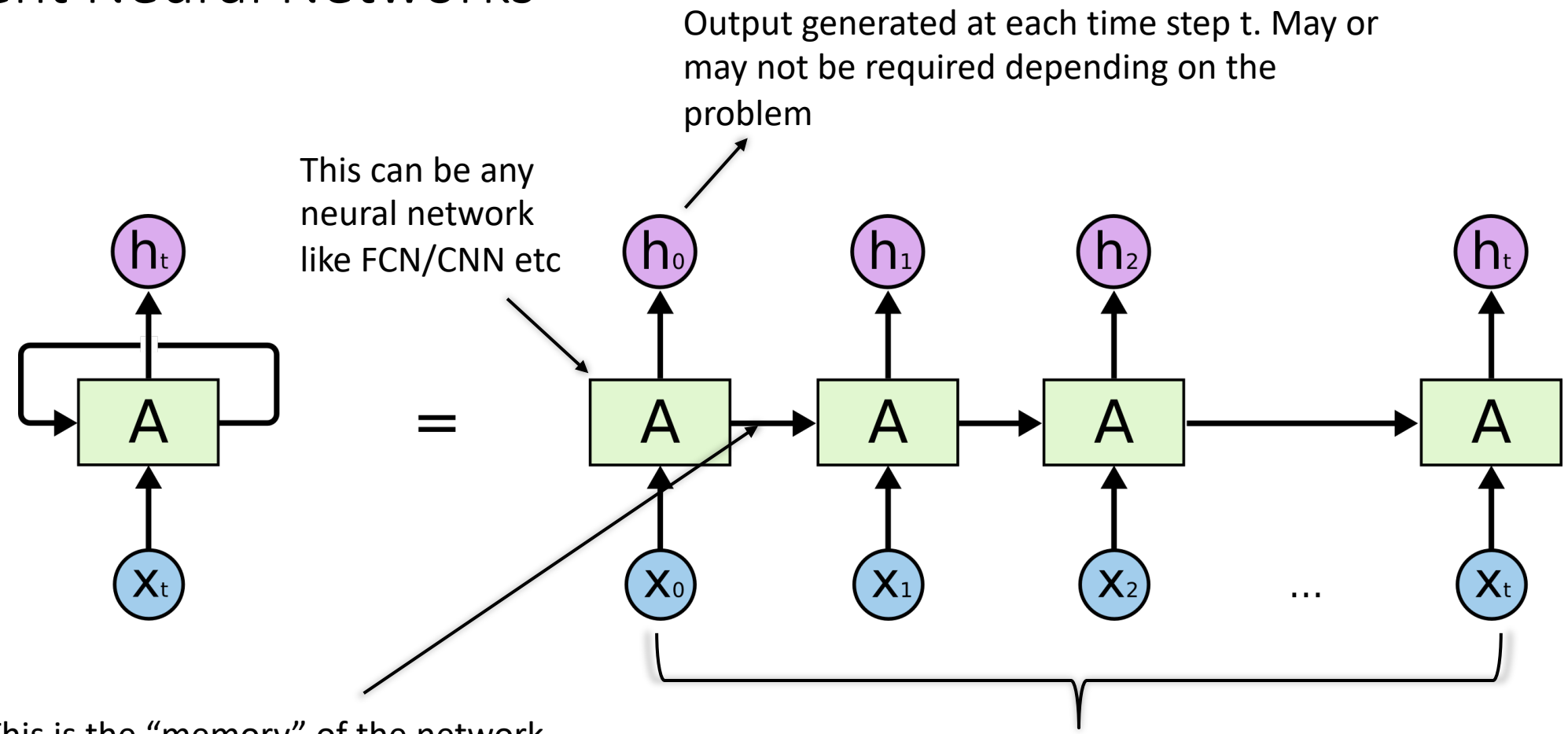
Images in this section are courtesy of Chris Olah

Recurrent Neural Networks

- Architecture is designed to work well on sequential data
- Example use cases – Language Modeling/translation, Speech Recognition



Recurrent Neural Networks



Main takeaway: Different neural net architectures are designed to leverage different kinds of data but the core idea remains the same – Update parameters using gradient descent such that loss is minimized

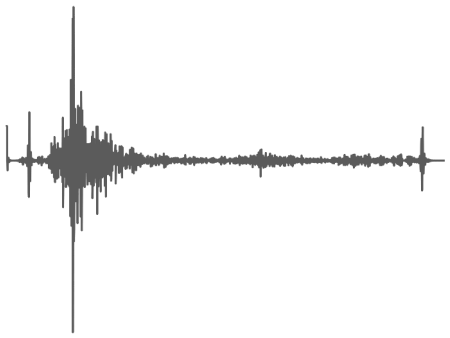
Applications

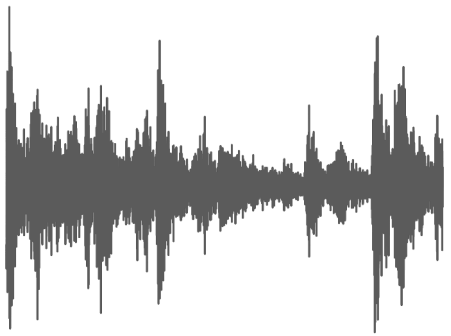
Earthquake Detective

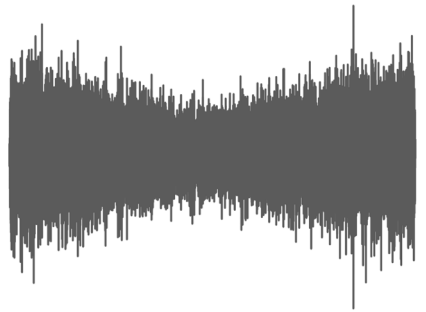
Ranadive, Omkar et al., “Applying Machine Learning to crowd-sourced data from Earthquake Detective”, NeurIPS’20.

Introduction

- Earthquake Detective is an online crowdsourcing platform on zooniverse.org
- Volunteers classify waveforms as - earthquakes, tremors, noise, none-of-the-above
- Volunteers are presented with signals from potentially triggered seismic events



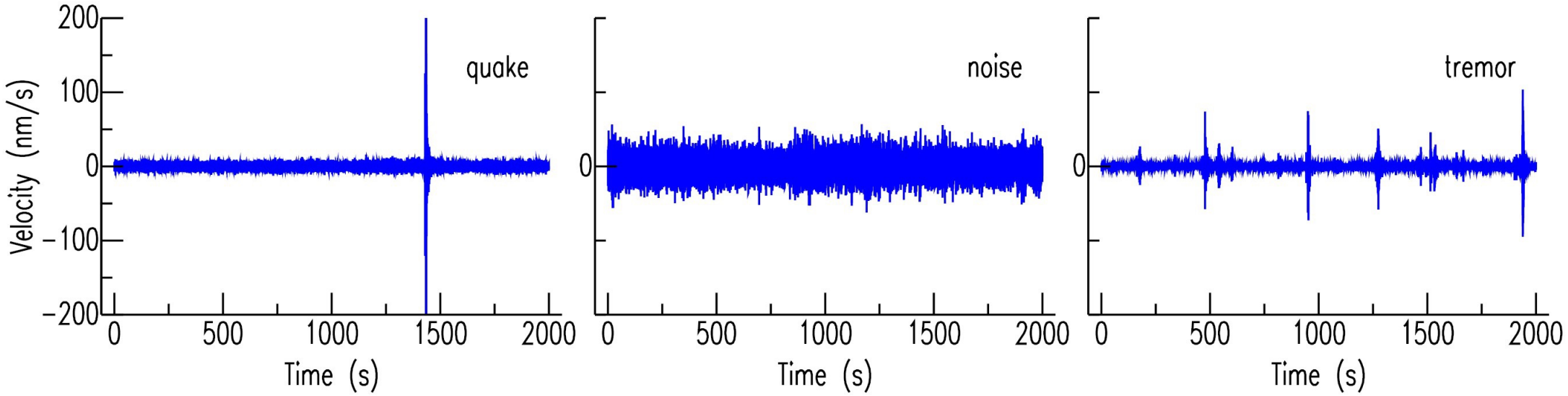




Potentially Triggered Seismic Events

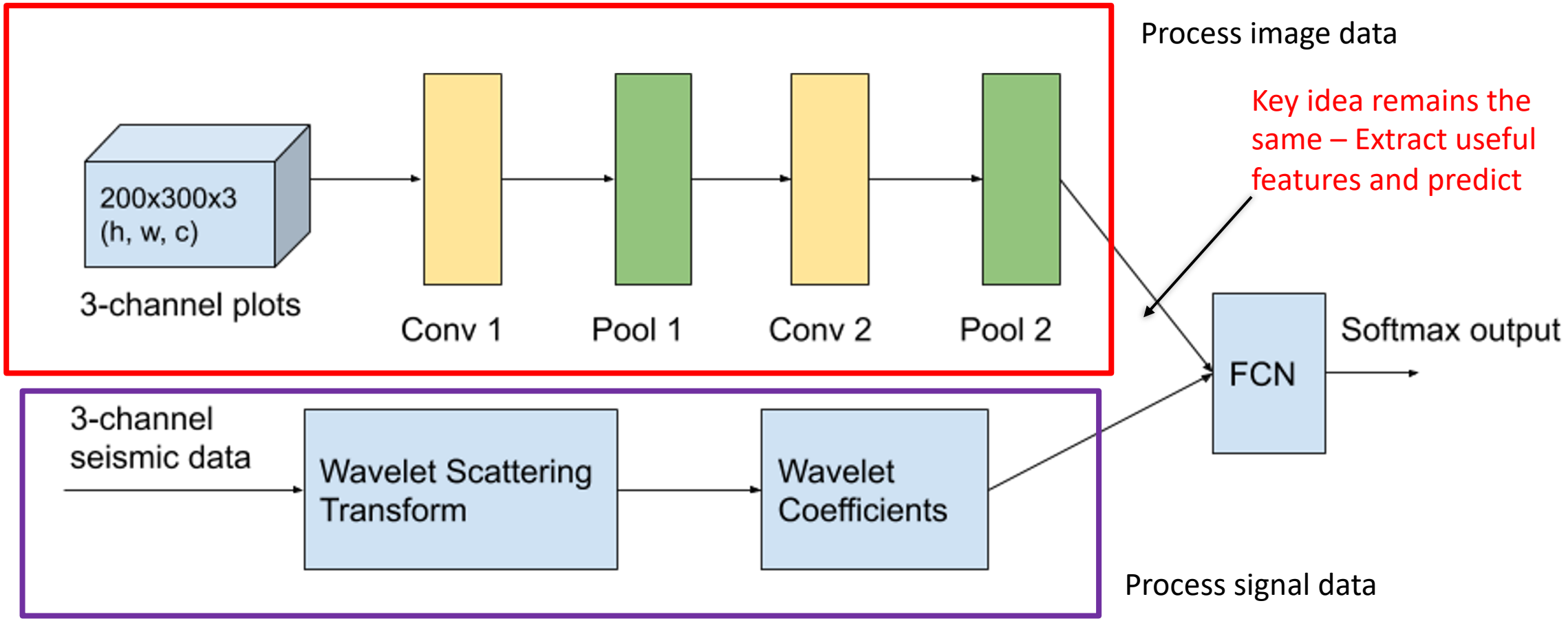
- Some local seismic events are triggered by strong seismic waves from remote earthquakes
- PT seismic events typically generate low-amplitude signals
- These signals typically have lower bandwidths
- A large database of template signals is rarely available
- The signals easily get buried in noise

Examples



The users are shown plots along with a sped up audio version of the plots

Model Architecture

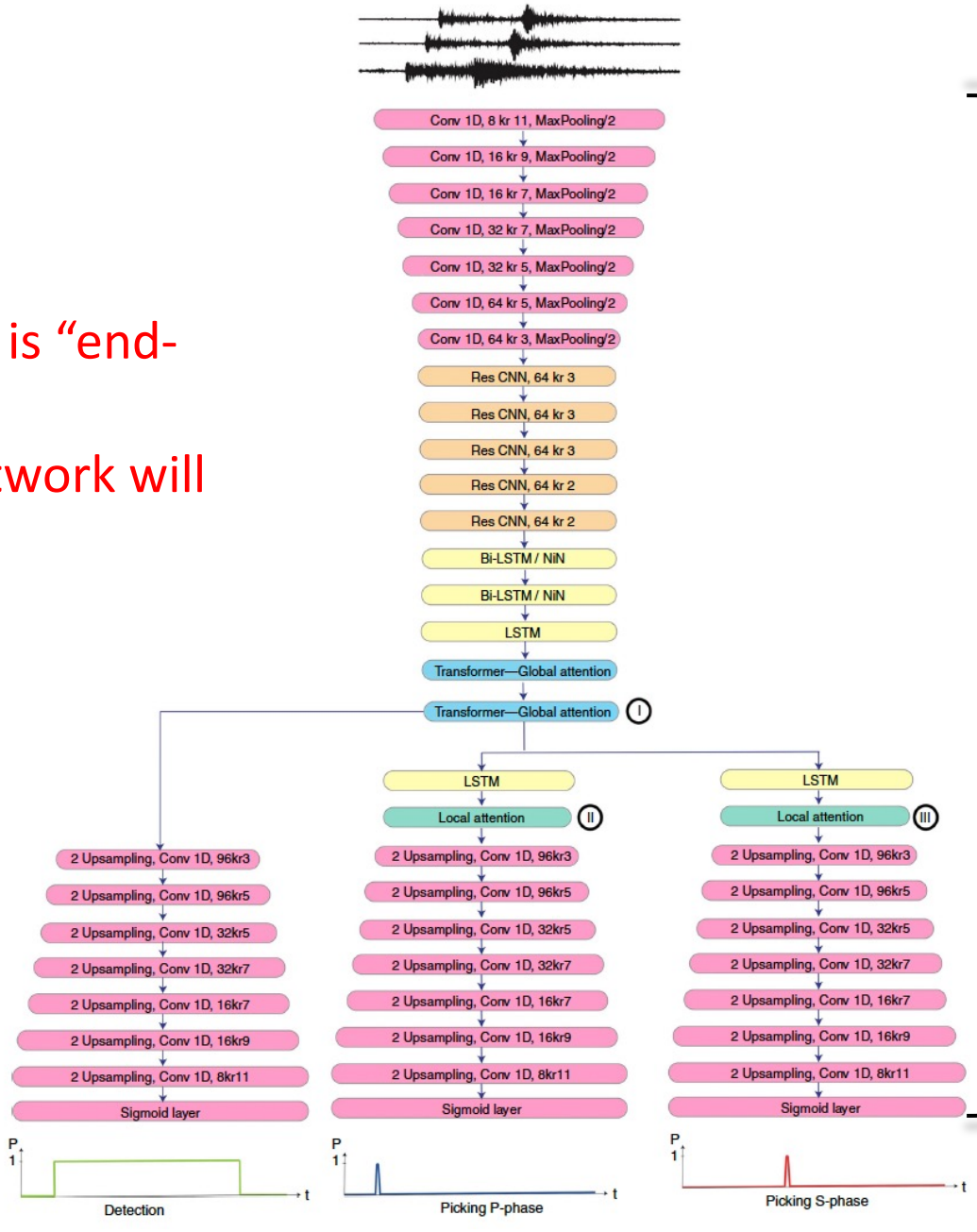


Earthquake Transformer

Mousavi, S.M. et al., “Earthquake transformer—an attentive deeplearning model for simultaneous earthquake detection and phase picking” Nat Commun 11, 3952 (2020).

Model Architecture

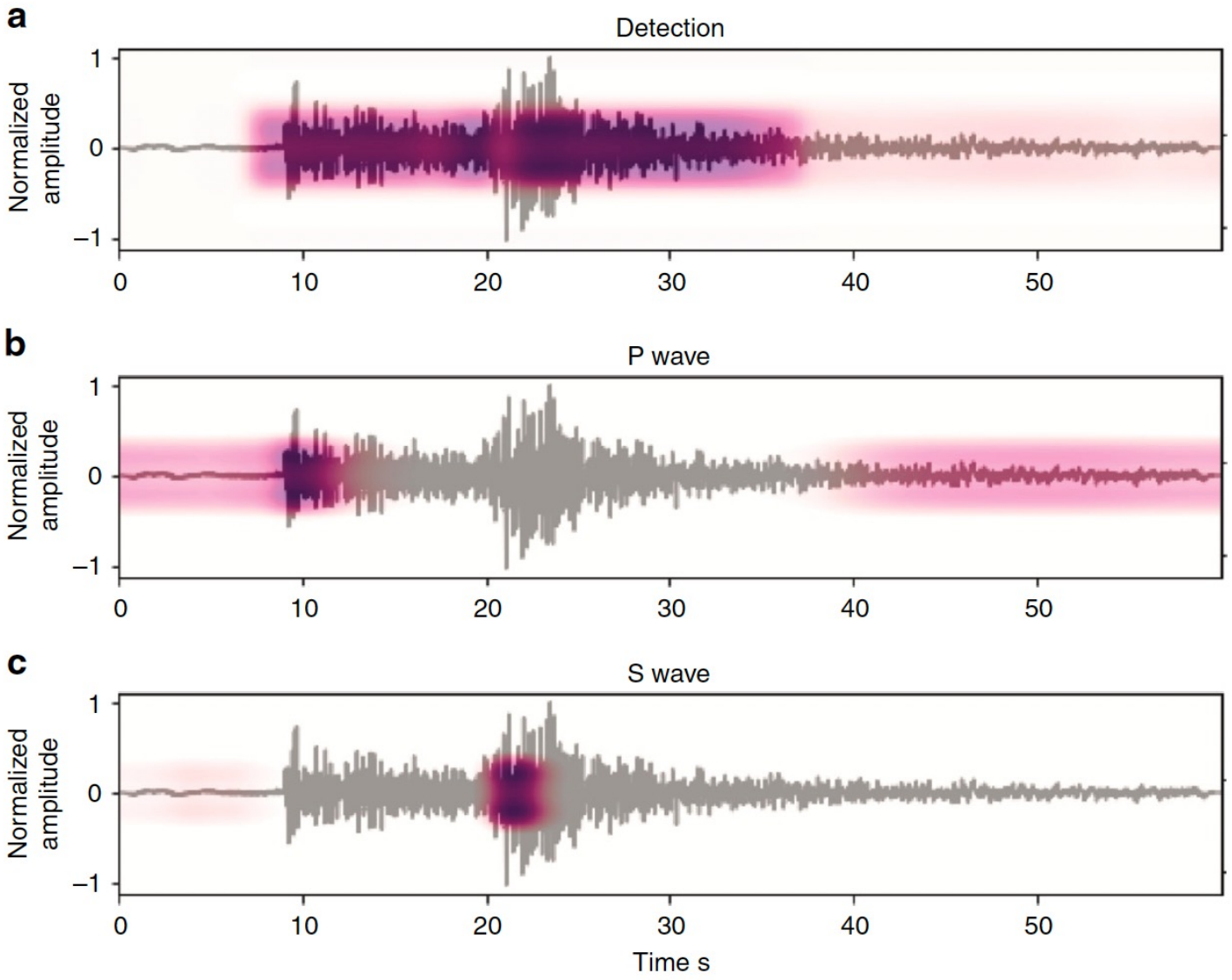
Main takeaway: So the key advantage of deep learning is “end-to-end” learning. No hand-engineering is required; network will learn everything by itself



Rest of the architecture is a combination of CNNs and RNNs and FCNs with some modifications.

Performs three tasks simultaneously

Attention


a
Detection
Normalized amplitude
1
0
−1
0
10
20
30
40
50
b
P wave
Normalized amplitude
1
0
−1
0
10
20
30
40
50
c
S wave
Normalized amplitude
1
0
−1
0
10
20
30
40
50
Time s

alchera

Wildfire Detection

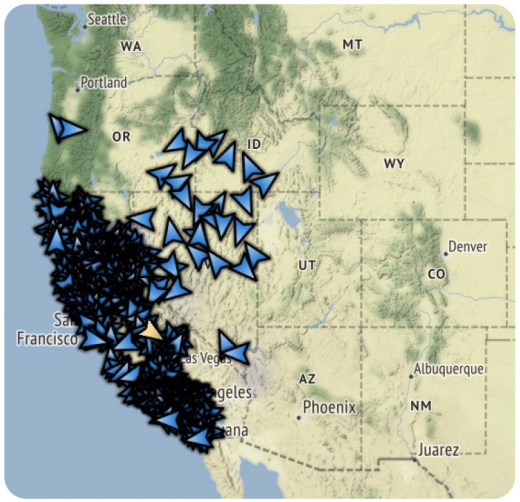
Alchera Inc, Wildfire Alert System

Alert Wildfire Cameras

- Began as a collaboration between Nevada Seismological Lab and Forest Guard Team
- Currently, it is a big network with **812+** PTZ cameras

Network Partners

Coverage Area

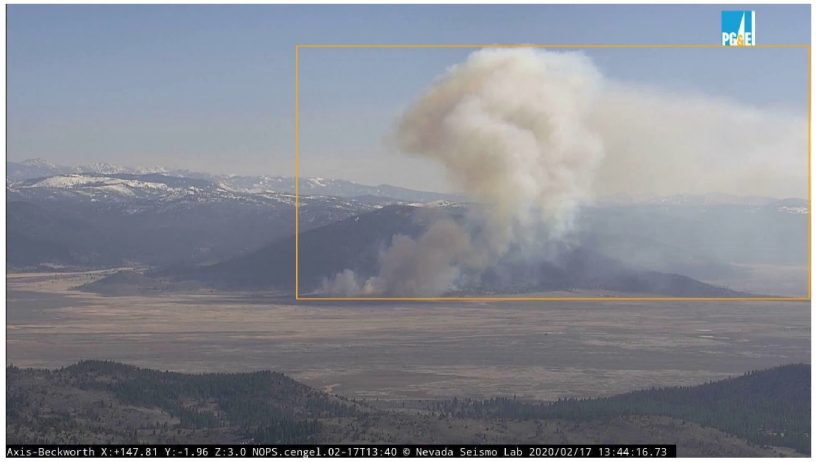


Cameras

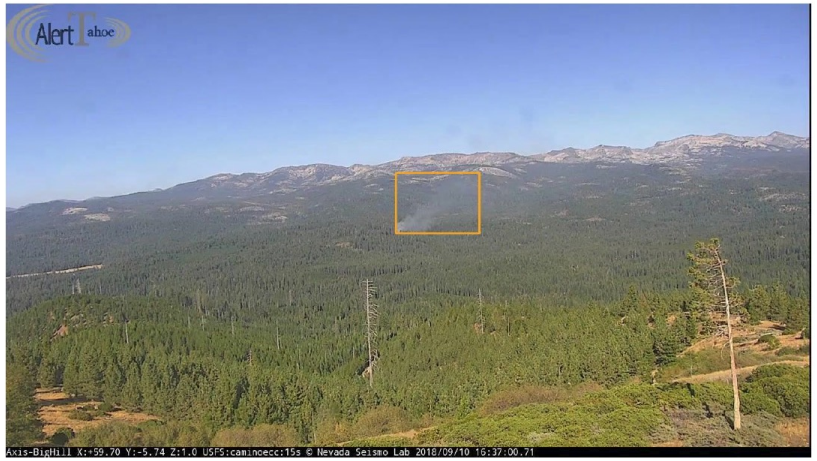
AXIS Q6055-E PTZ Network Camera

http://www.alertwildfire.org/

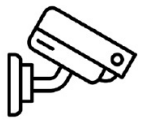
The Data


PG&E
Axis-Beckworth X:+147.81 Y:-1.96 Z:3.0 NOPS.cengel.02-17T13:40 © Nevada Seismo Lab 2020/02/17 13:44:16.73


EDISON
Axis-OatNorth1 X:-4.37 Y:-2.48 Z:5.6 VCFD:bheath:1236 © Nevada Seismo Lab 2019/06/09 12:39:00.51


Alert Tahoe
Axis-DigHill X:+59.70 Y:-5.74 Z:1.0 USFS:camincecc:15s © Nevada Seismo Lab 2018/09/10 16:37:00.71


Alert Tahoe
Axis-Zephyr X:-96.67 Y:+3.06 Z:1.0 System:Ready:2.9d © Nevada Seismo Lab 2017/11/06 11:32:10.60 Home



Firespotting Cameras

Send image data to the cloud
Provided by ALERT Wildfire

ALERT Wildfire

AI in the Cloud

Deep-learning algorithms **detects smoke**, sends wildfire alerts for confirmation

Human-in-the-Loop

1 person per 250+ cameras filters AI results for false alarms, **confirms alerts**

Immediate Alert

Human-confirmed fires populate on dashboard, triggers an **alert to users** by SMS, e-mail

alchera

Behind the scenes, it is still a form of convolution neural networks!

Conclusion

- Machine Learning is a powerful tool as it can learn the characteristics of data
- Similar architectures can be used to solve wide array of problems making it general purpose
- ML can greatly help expedite the scientific development process

Q/A

THANK YOU!