Tutorial 3: More Networks and More Areas

1. From Redshift to Classification: A Simple Change
2. Deepening the Redshift Net
3. Applying the Redshift Net to Gravitational Wave

Yu Wang

ICRANet / INAF / University of Rome

Nov 4th, 2021

Online

VIdeo Credit to Kurzgesagt – In a Nutshell

TUTORIAL 3: MORE NETWORKS AND MORE AREAS

Deep Learning Procedure

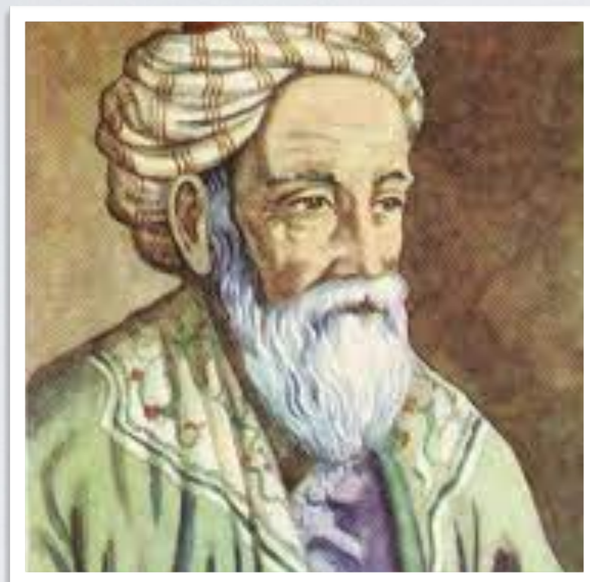
YU WANG

What can machine do ?

Input

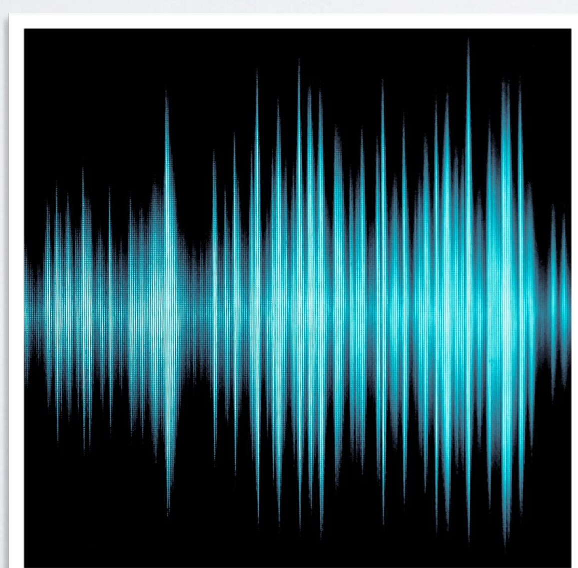
Output

Image Recognition



Omar Khayyam

Voice Recognition

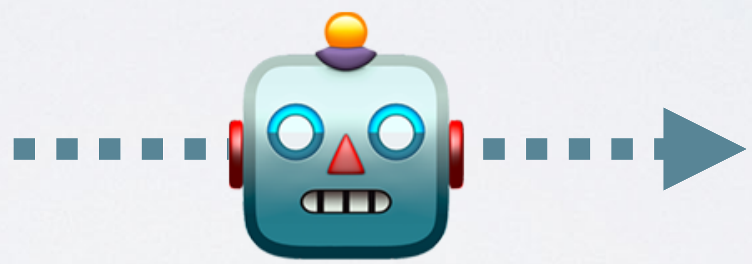


Isfahan is half the world

An answer from a given dataset

Data

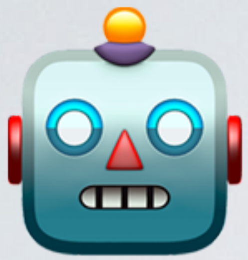
Answer



42

Image credit to Matrix

Machine as a *map* action.



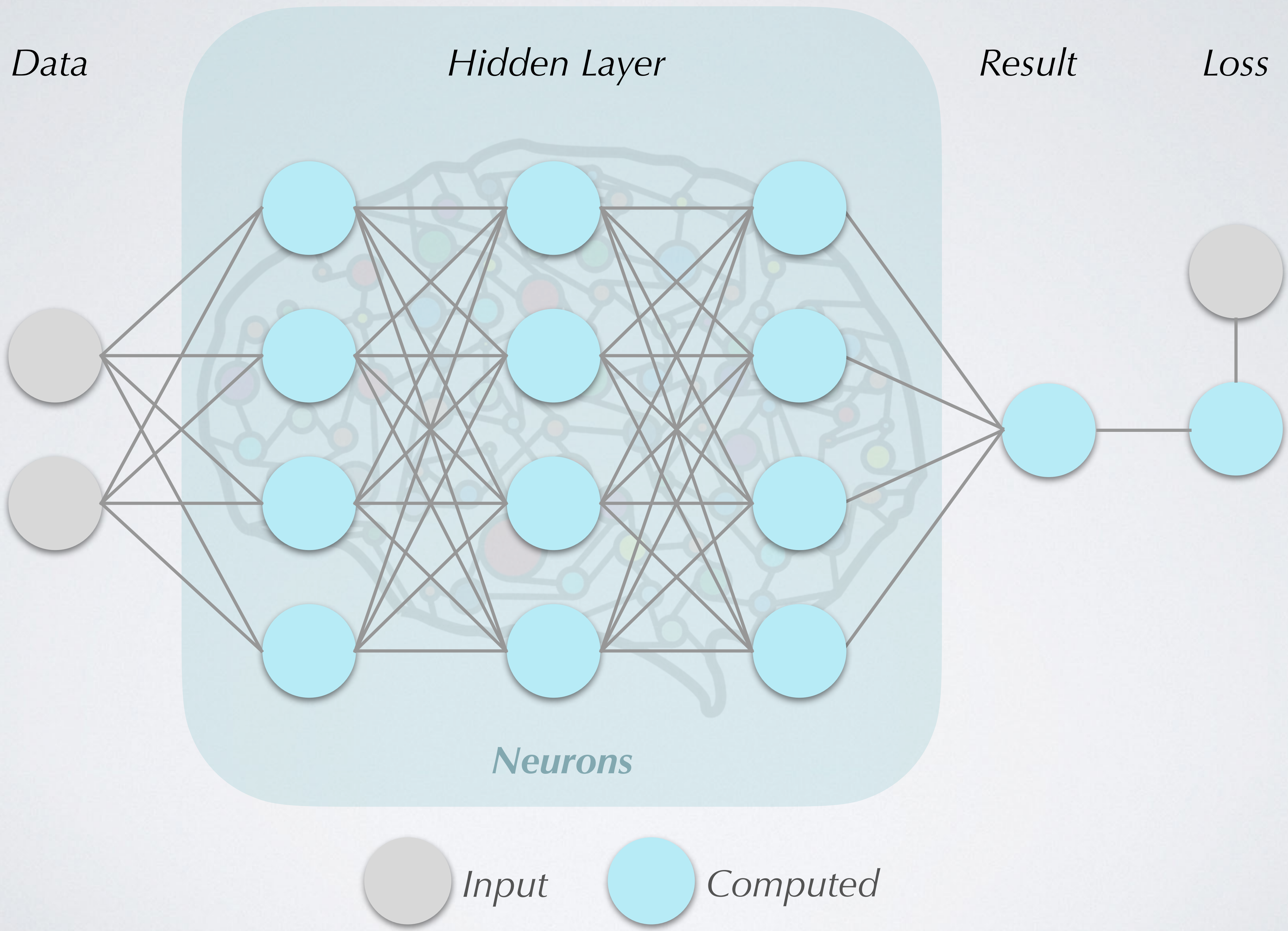
$$f\colon \text{data} \longmapsto \text{answer}$$

Deep learning:

f is constructed by many layers of neurons

DEEP LEARNING

A Simple Example - Fully Connected Net

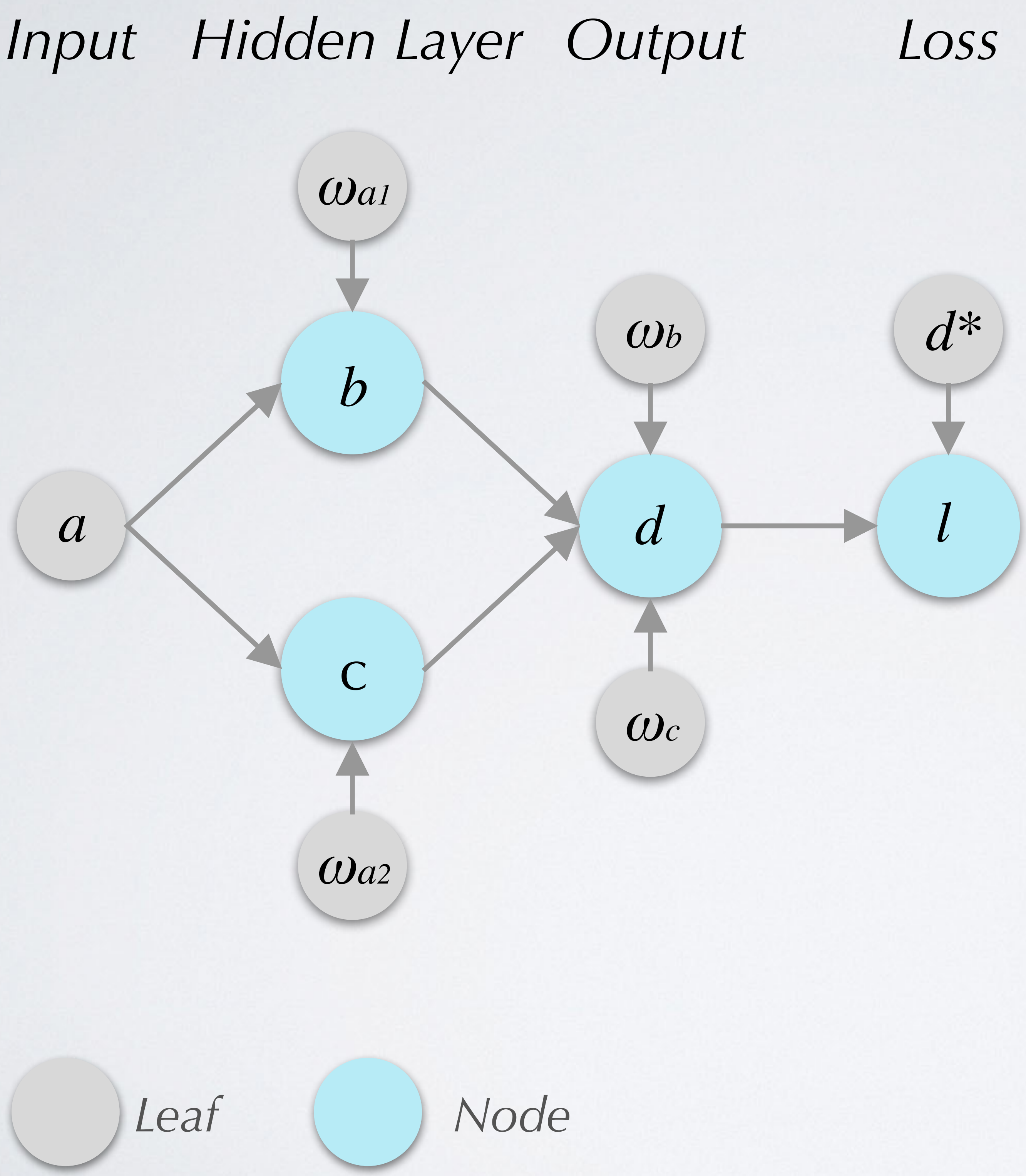


A Simple Example - Fully Connected Net

$$\overrightarrow{y} = A(\omega \cdot \overrightarrow{x} + b)$$

Output *Activation function* *Weight* *Input* *Offset*

A simple Net



Forward Propagation

$$x = A(x^T\omega)$$

$$e = loss(d, d^*)$$

$$b = \omega_{a1}a$$

$$c = \omega_{a2}a$$

$$d = \omega_b b + \omega_c c$$

$$e = d^* - d$$

* *For simplicity, the activation function (A) is ignored.*

Optimization

For a give set of ω^*, *the loss reaches 0:*

$$loss(d, d^*) = 0$$

When $\omega = \omega^*$

To find ω^*, *a simple and common way:*

Gradient descent

repeat until convergence

$$\omega_{t+1} = \omega_t - \eta \frac{\partial loss}{\partial \omega_t}$$

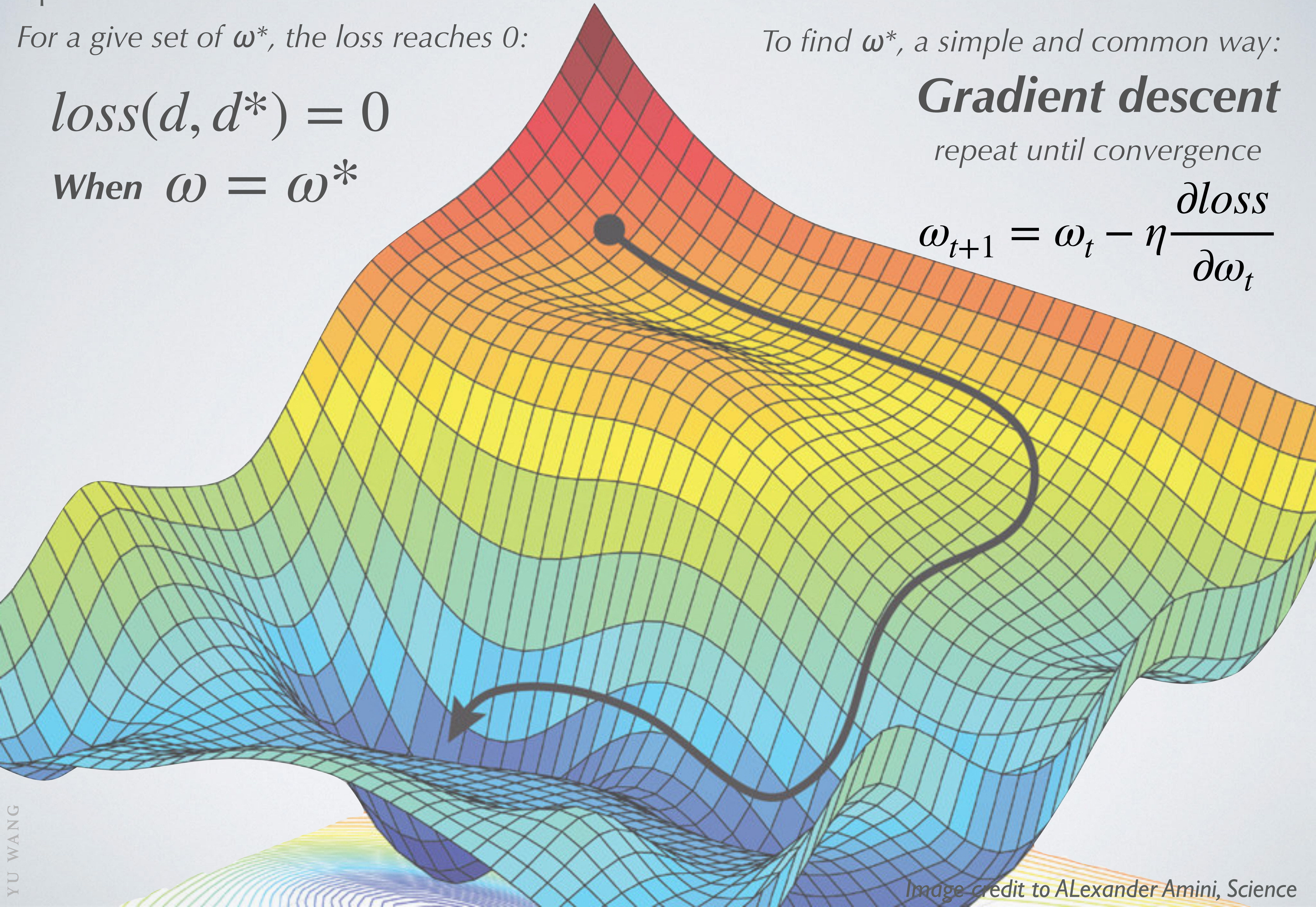
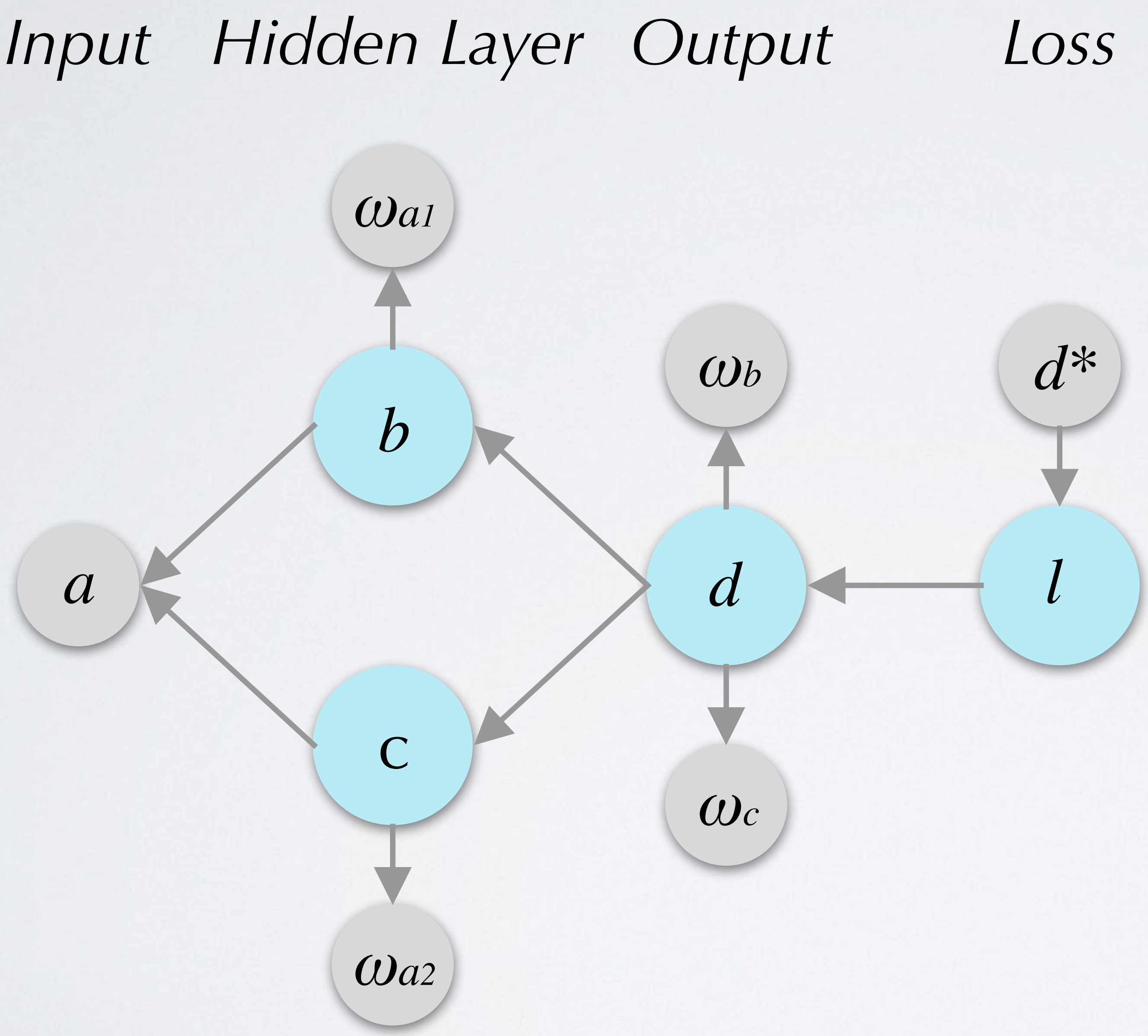


Image credit to ALexander Amini, Science

Chain Rule of Derivatives



$$\frac{\partial l}{\partial \omega_d} = \frac{\partial l}{\partial d}\frac{\partial d}{\partial \omega_d}$$

$$\frac{\partial l}{\partial \omega_c} = \frac{\partial l}{\partial d}\frac{\partial d}{\partial \omega_c}$$

$$\frac{\partial l}{\partial \omega_{a2}} = \frac{\partial l}{\partial d}\frac{\partial d}{\partial c}\frac{\partial c}{\partial \omega_{a2}}$$

$$\frac{\partial l}{\partial \omega_{a1}} = \frac{\partial l}{\partial d}\frac{\partial d}{\partial b}\frac{\partial b}{\partial \omega_{a1}}$$

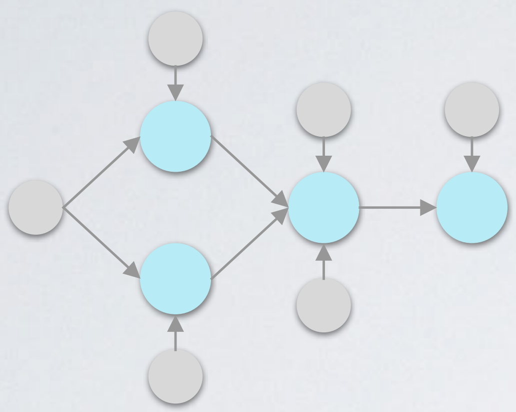
$$\Downarrow$$

$$\omega_{t+1} = \omega_t - \eta\frac{\partial l}{\partial \omega_t}$$

Procedure of Training a Net

$$normal(\text{gain} \times \sqrt{\frac{6}{\text{fan\_in} + \text{fan\_out}}})$$

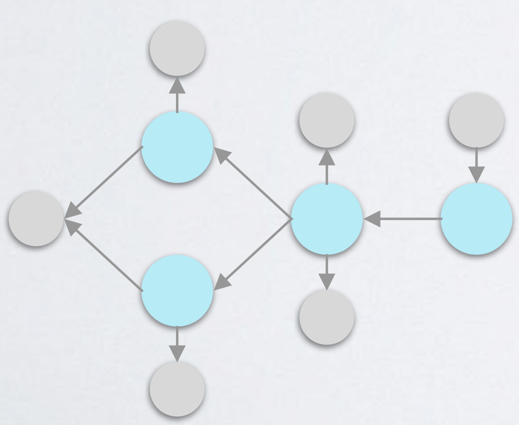
$$uniform(\text{gain} \times \sqrt{\frac{3}{\text{fan\_mode}}})$$



$$MSE(x) = E((x - x^*)^2)$$

$$H(p, q) = -\Sigma p(x) \log q(x)$$

$$D_{KL}(p||q) = \Sigma p(x) \ln \frac{Q(x)}{P(x)}$$



$$\frac{\partial loss}{\partial \omega} \rightarrow 0$$

0. Initialization

Initialize the weight and offset by random or specific methods, for e.g., Xavier normal, He uniform.

↓

repeat until convergence

1. Forward Propagation

The input features go throughout all the layers with lots of matrix computation and non-linear activation functions, and finally make a prediction.

↓

2. Compute the loss

Loss is the distance defined between the predicted label and the true label, for e.g., mean squared error, cross entropy, Kullback-Leibler divergence.

↓

3. Back Propagation

Loss back propagates all the net, and optimizes parameters by methods of, for e.g., stochastic gradient descent, RMSProp, Adam.

↓

4. Convergence

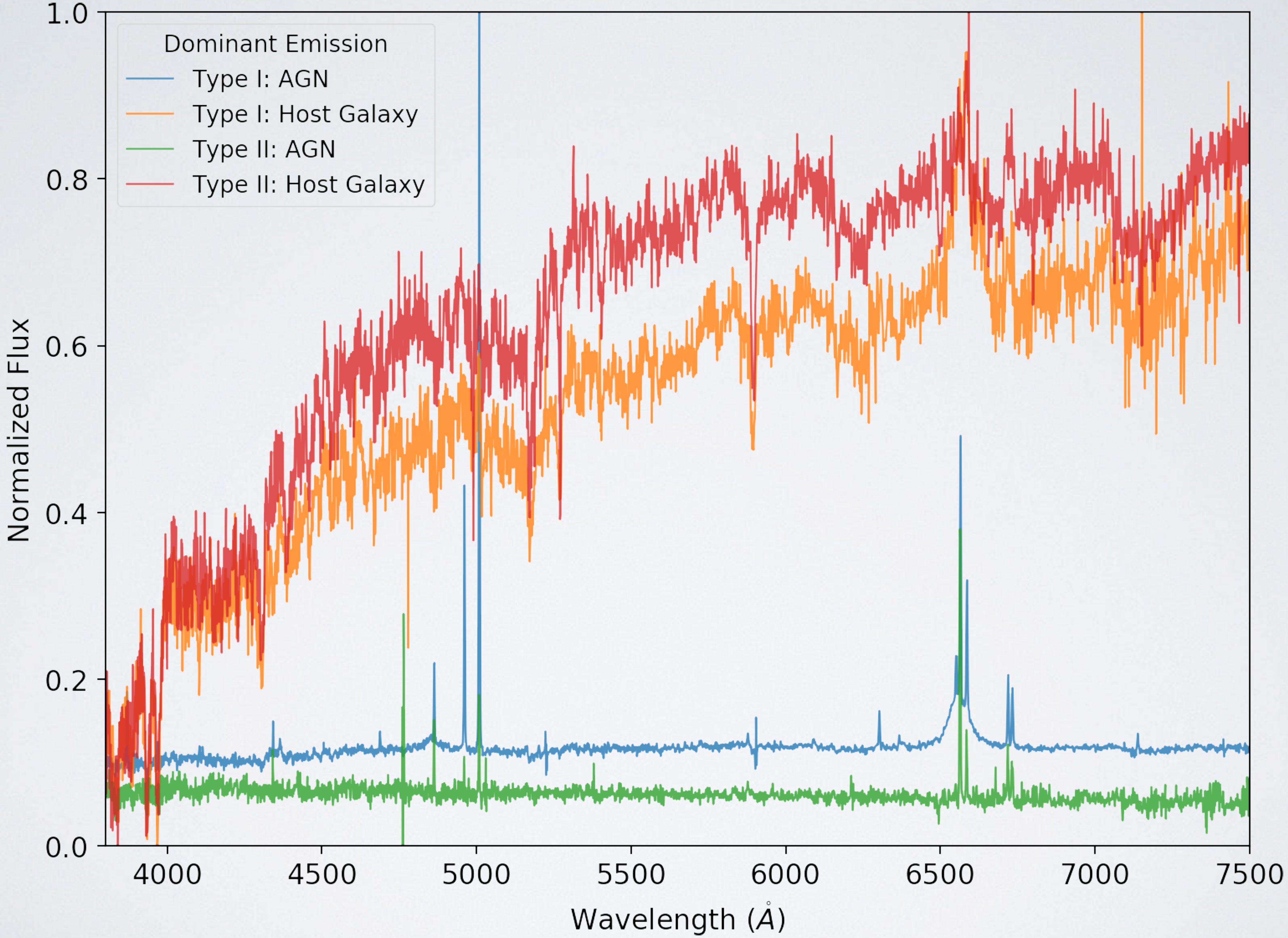
The loss function reaches a global minimal or a local minimal.

TUTORIAL 3: MORE NETWORKS AND MORE AREAS

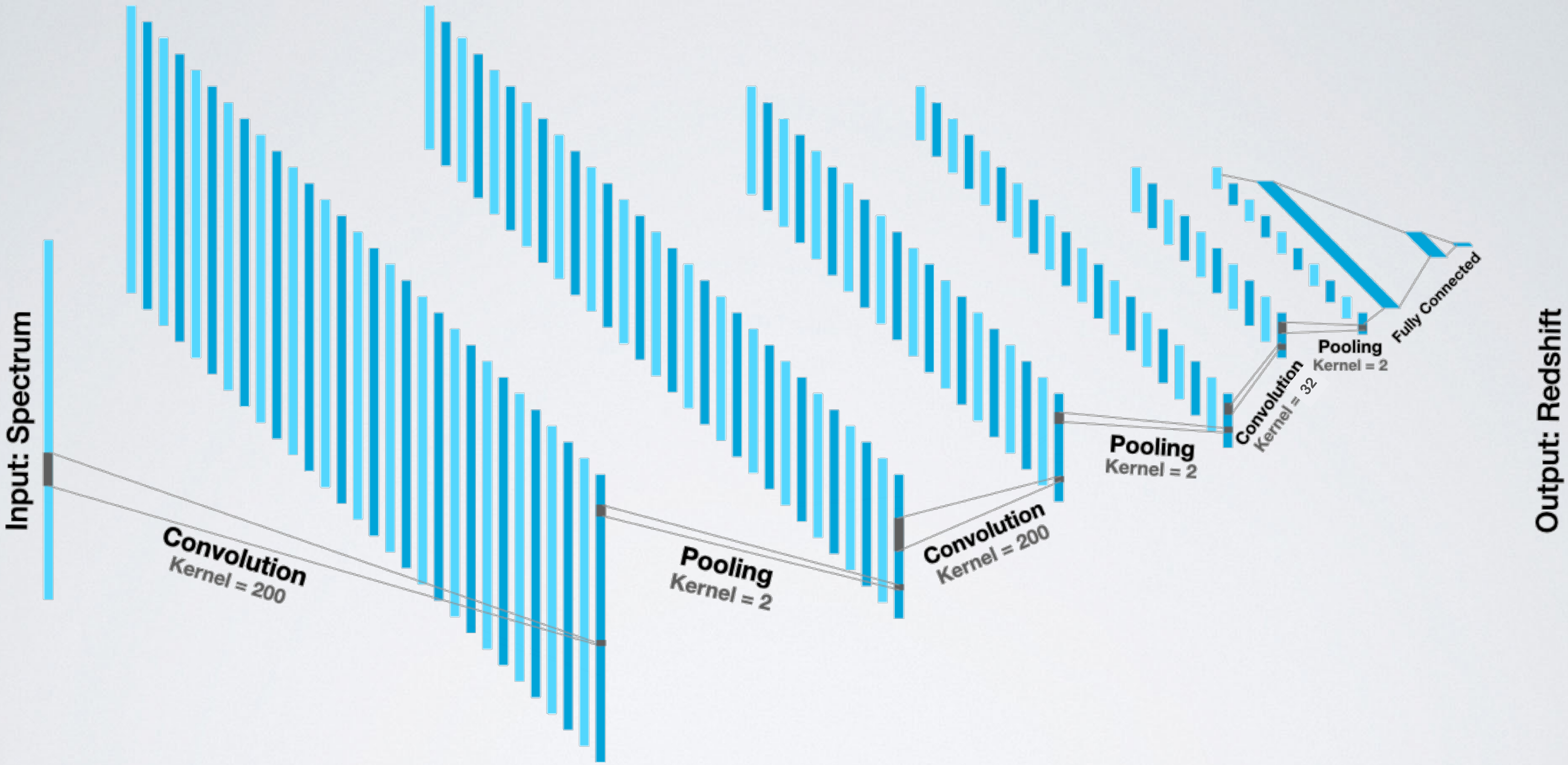
From Redshift to Classification: A Simple Change

YU WANG

SDSS Data - One Dimensional

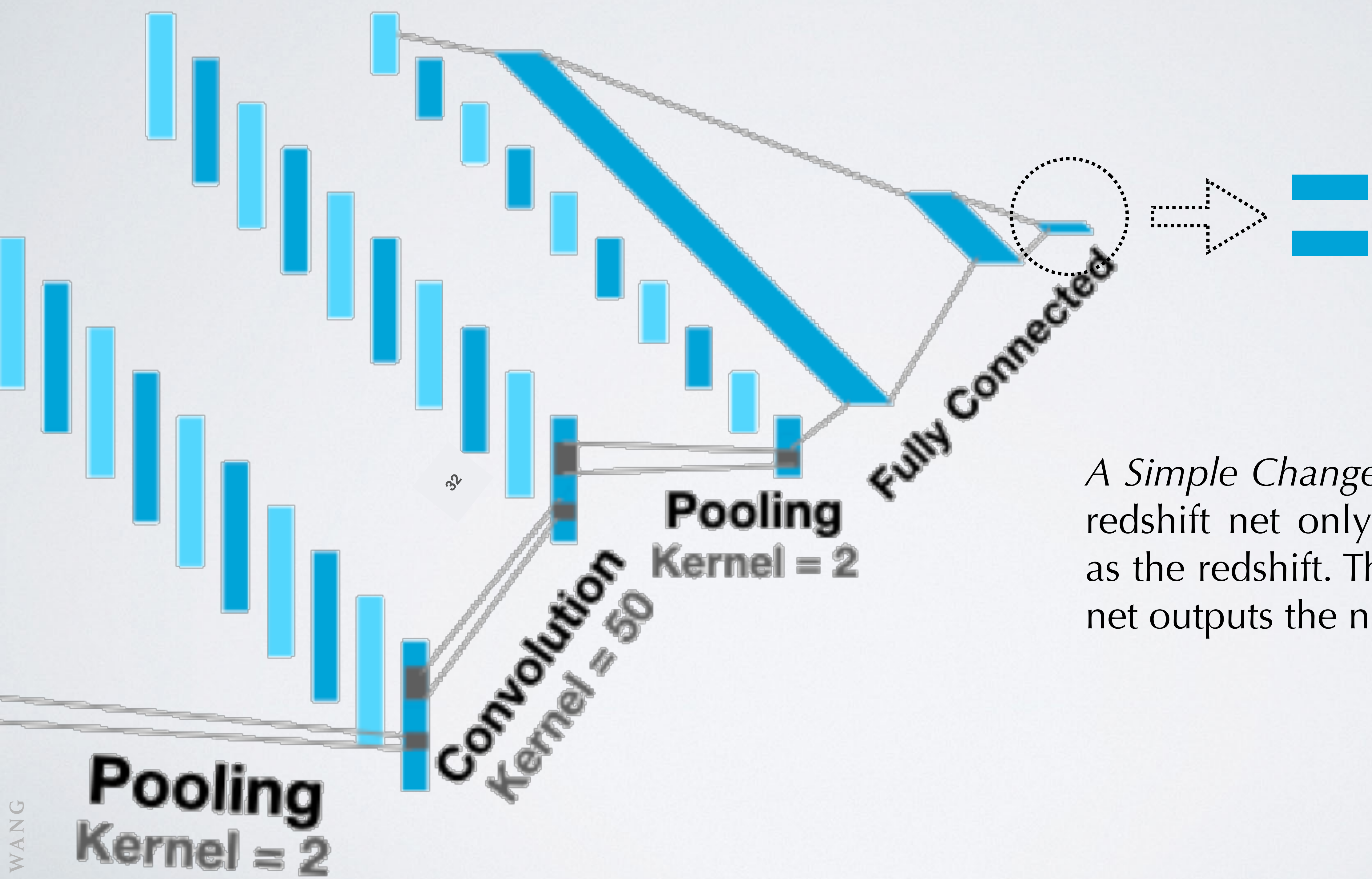


One Dimensional Convolutional Neural Network For Inferring Resfhit



Structure of one dimensional CNN. The spectrum of quasar is input as a one-dimensional array, which goes through the convolutional layer of kernel size = 200, 200, 32 respectively in order to search for the global and local pattern. The fully connected layers output the redshift.

One Dimensional Convolutional Neural Network For Classification



A Simple Change: The output of redshift net only has one value as the redshift. The classification net outputs the number.

One Dimensional Convolutional Neural Network For Classification

To standardize the output, that all values are between 0 and 1, and the sum of all equals to one. Done by passing to a LogSoftmax function

$$\text{LogSoftmax: } p(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right),$$

And we need to change the loss function. The result $q(x_i)$ (predicted classification) will be adopted and together with the labels $p(x_i)$ (real classification) to compute the cross entropy as the loss

$$\text{CrossEntropy: } H(p,q) = \sum_i p(x_i) \log q(x_i)$$

One Dimensional Convolutional Neural Network For Classification

Galaxy

0.28

Quasar

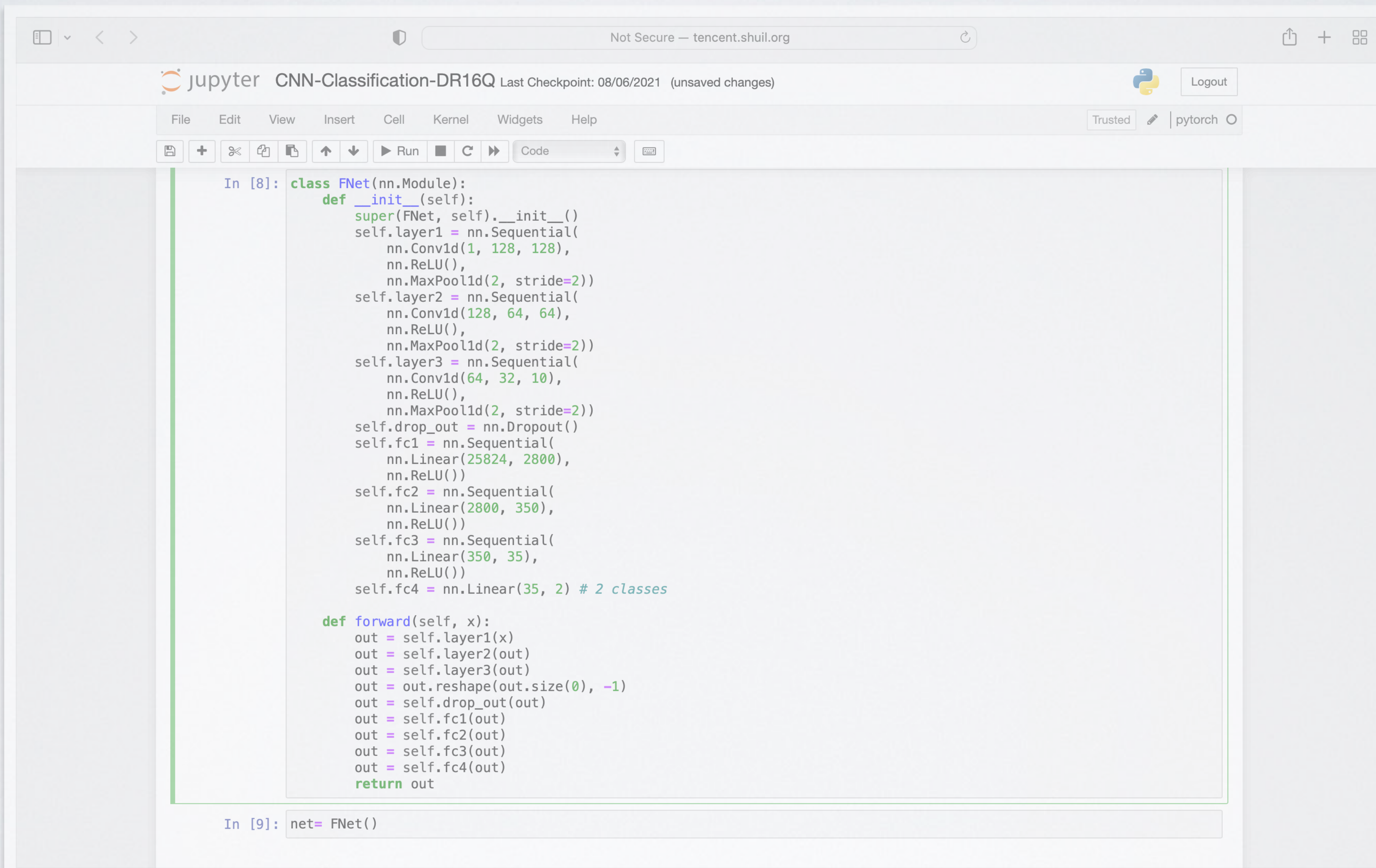
0.72

Quasar

The sum of all outputs equals to 1, the biggest value corresponds to the predicted class.

FROM REDSHIFT TO CLASSIFICATION: A SIMPLE CHANGE

Demonstration in Jupyter notebook



https://github.com/YWangScience/Isfahan-workshop-2021/tree/main/code

TUTORIAL 3: MORE NETWORKS AND MORE AREAS

Deepening the Redshift Net

YU WANG

ResNet

If we count the latest neural network structures and refer to the winning networks of some recent machine learning competitions, we can see that most of the networks use the classical ResNet as a basis. This is especially true for networks targeting one-dimensional data, such as the gravitational wave prediction competition held at Kaggle just last month, where the top 3 networks all involved ResNet and did not make complex changes. So in this tutorial, we are going to demonstrate the ResNet.

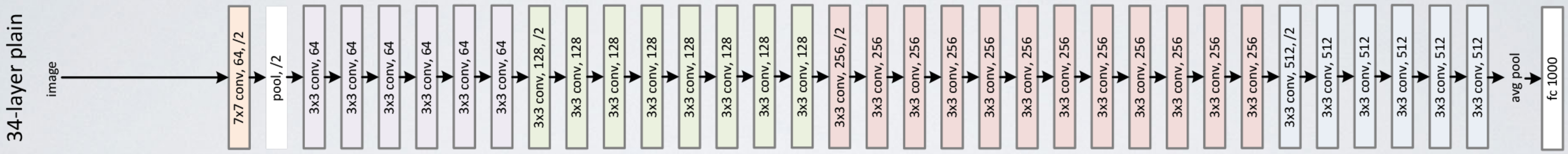
ResNet (Arxiv: 1512.03385)

Deep Residual Learning for Image Recognition

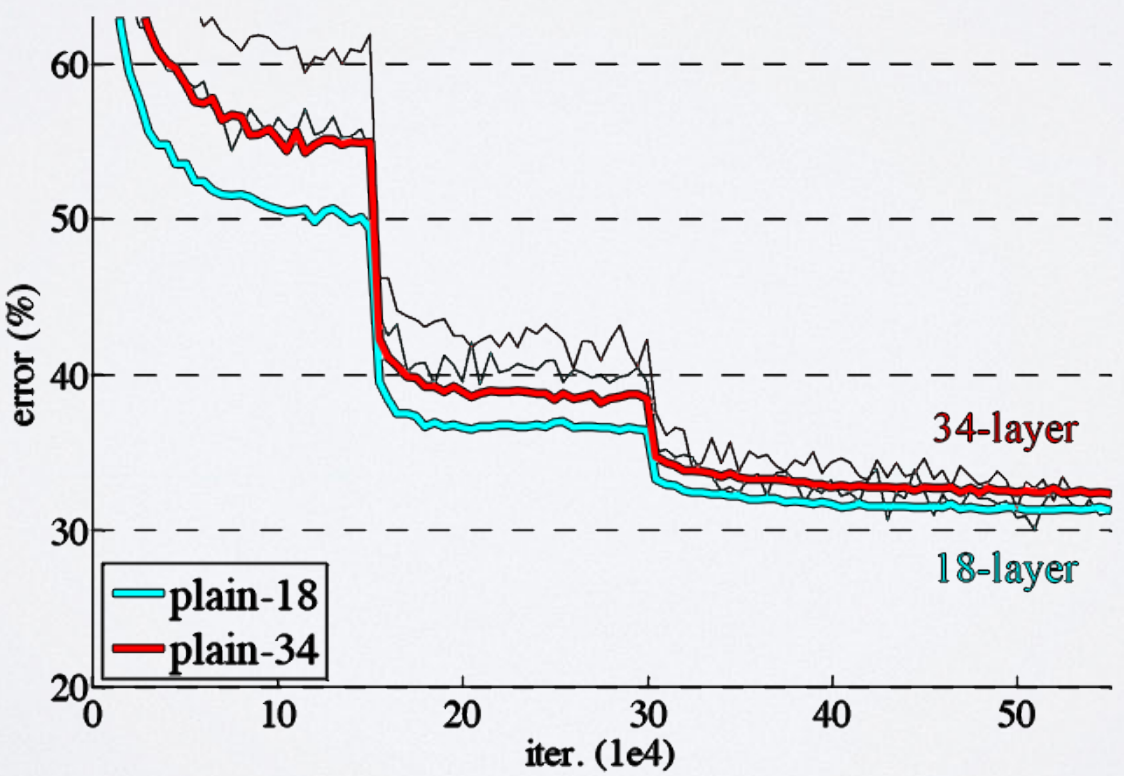
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

More Layers

The most straightforward idea to get more accurate predictions is to make the network deeper and wider to include more parameters and non-linear structures.

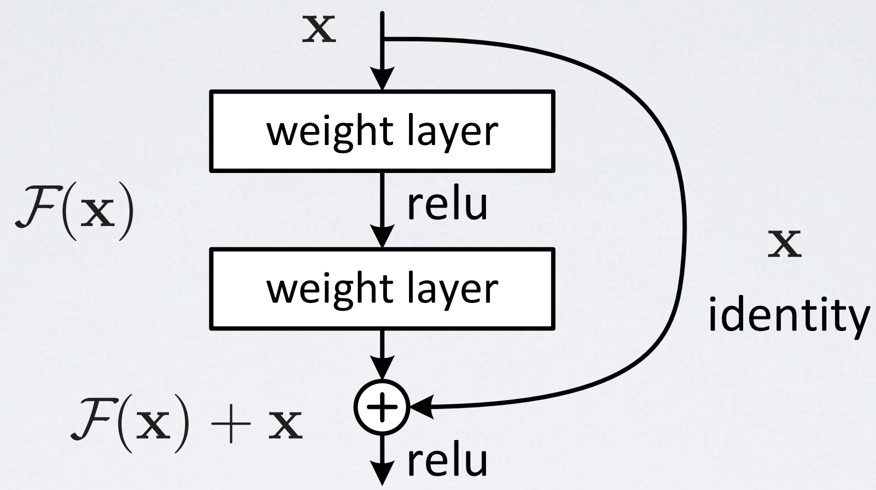


As the network depth increasing, accuracy gets saturated and then degrades rapidly, called the ***degradation*** *problem.*

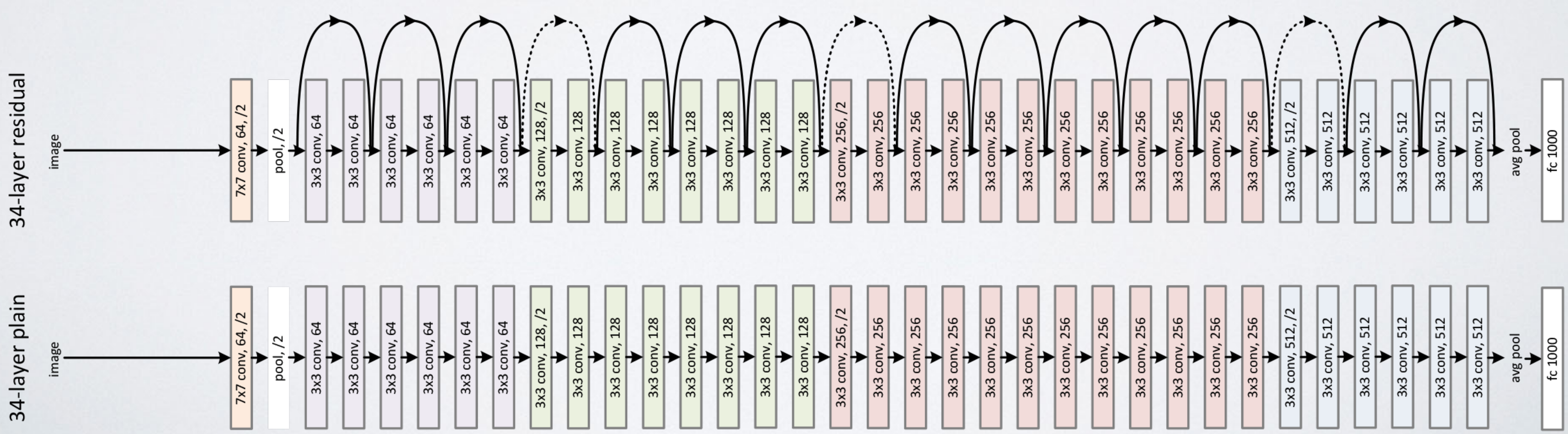


Shortcuts

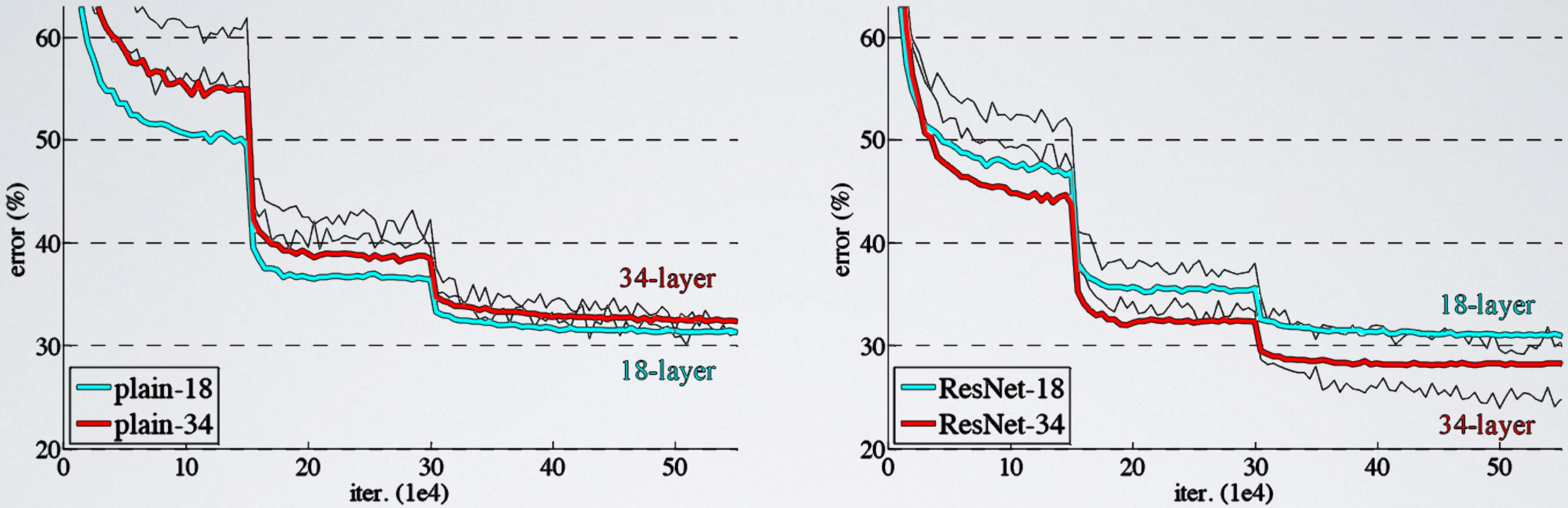
The shortcut connections simply perform ***identity*** *mapping, and their outputs are added to the outputs of the stacked layers*



Example of Residual network, shortcuts added.



Solving the degradation problem



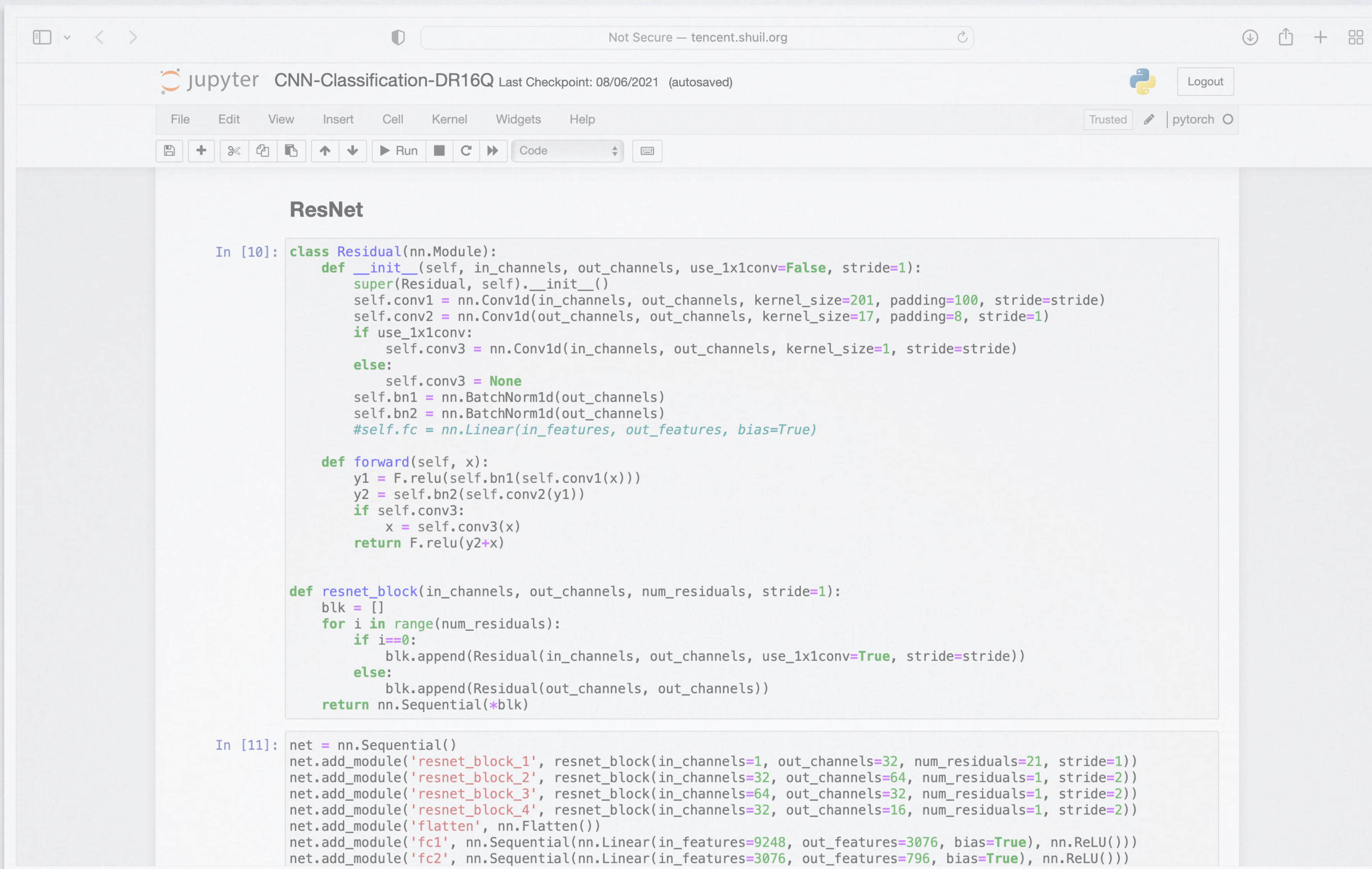
The error of 34 layers ResNet becomes smaller than the 18 layers network.

**Why residual structure works? This is an unanswered question, many papers proposed the explanation, we will discuss if we have time.*

** Also pay attention to the efficient net, which studies model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Arxiv: 1905.11946*

DEEPENING THE REDSHIFT NET

Demonstration in Jupyter notebook



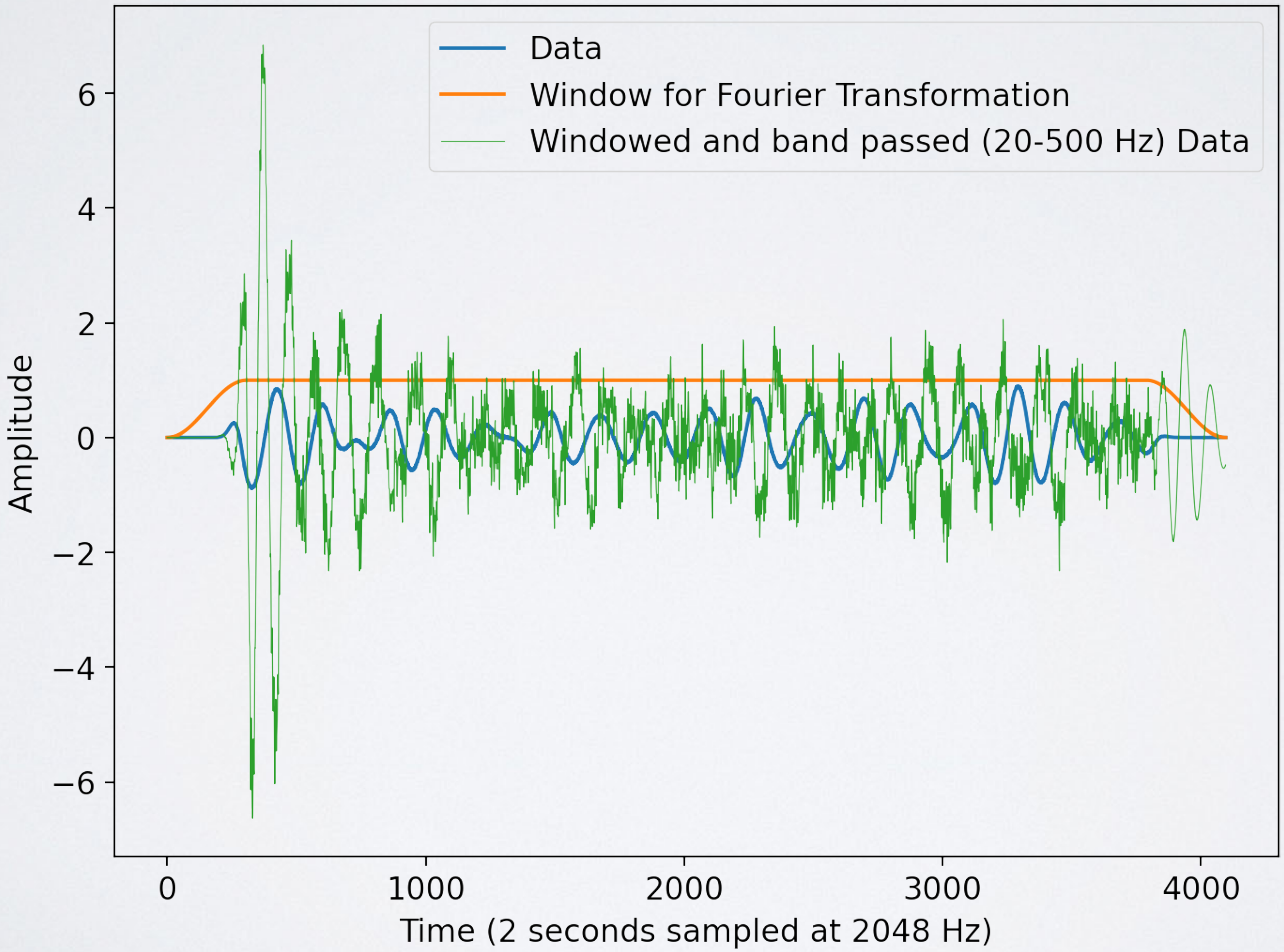
https://github.com/YWangScience/Isfahan-workshop-2021/tree/main/code

TUTORIAL 3: MORE NETWORKS AND MORE AREAS

Applying the Redshift Net to Gravitational Wave

Gravitational Wave Data - One Dimensional

Taking the competition of G2Net Gravitational Wave Detection hosted by European Gravitational Observatory as an example.
Details: https://www.kaggle.com/c/g2net-gravitational-wave-detection/



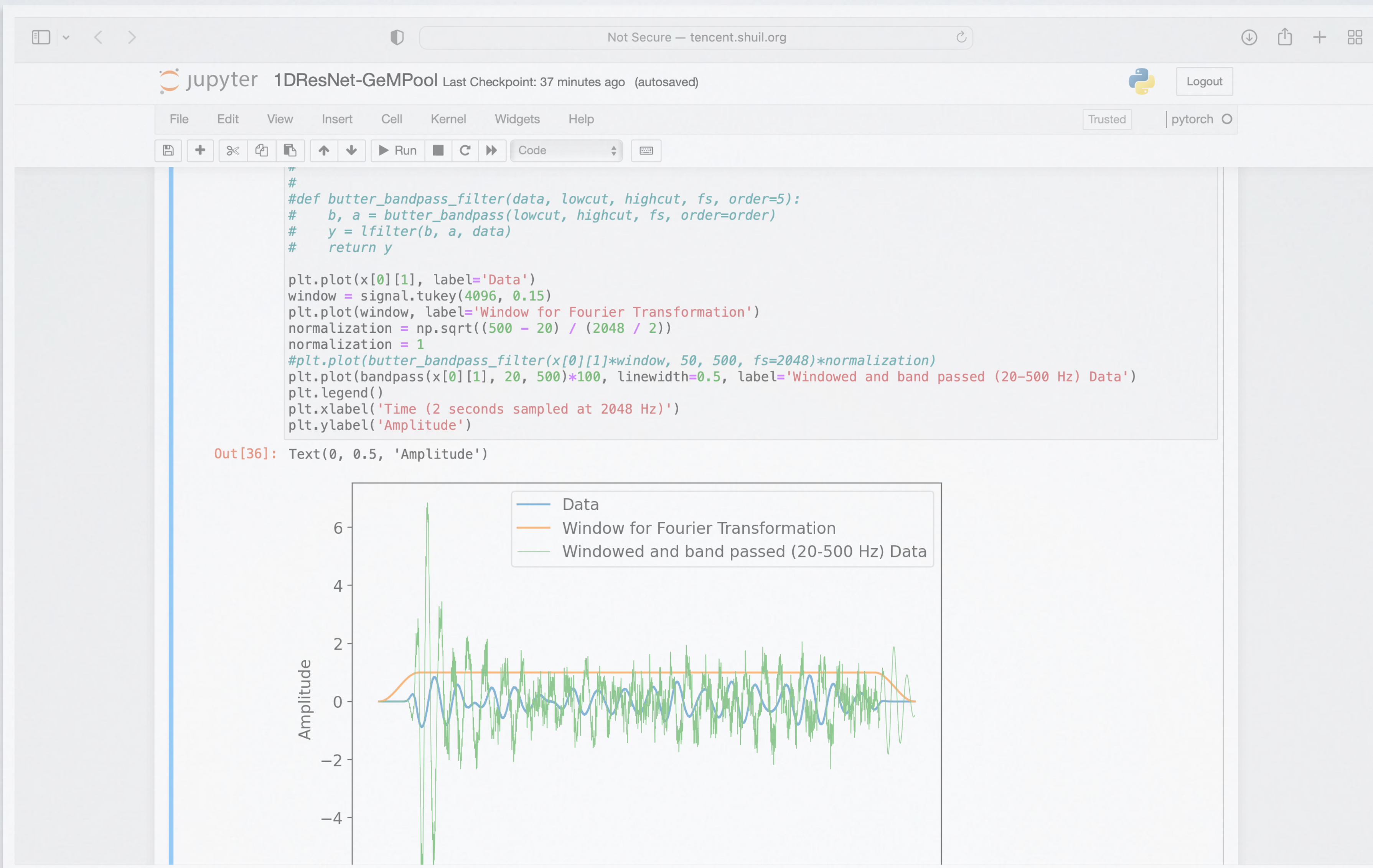
Gravitational Wave Data - One Dimensional

	Input	Output
Redshift	one dimensional data	redshift value
Gravitational Wave	one dimensional data	Is true signal?

We can adopt the same network for redshift detection, simply change the size of input length, and the loss function.

APPLYING THE REDSHIFT NET TO GRAVITATIONAL WAVE

Demonstration in Jupyter notebook



https://github.com/YWangScience/Isfahan-workshop-2021/tree/main/code

THANKS YOU

JOIN THE DISCUSSION CHANNEL ON SLACK

http://shorturl.at/tvyL1