Day 16, Oct-18, 2024

→ Continuing Convolutional Neural Networks Series.

Comparing Similar Fully Connected Network and CNN network Size.

Input layer	I/P	[(None, 1441)]
	O/P	[(None, 1441)]

↓

Dense	I/P	(None, 1441)
	O/P	(None, 100)

↓

Dense	I/P	(None, 100)
	O/P	(None, 200)

↓

Dense	Input	(None, 200)
	O/P	(None, 1)

FCN - 160K parameters.

I/p layer	I/P	(None, 1441, 1)
	O/P	None (1441, 1)

Conv 1D	I/P	None (1441, 1)
	O/P	None (1439, 16)

Max Pooling 1D	I/P	None (1439, 16)
	O/P	None (719, 16)

Conv 1D	I/P	(None, 719, 16)
	O/P	(None, 717, 32)

Maxpooling 1D	I/P	(None, 717, 32)
	O/P	(None: 358, 32)

Flatten	I/P	(None, 358, 32)
	O/P	(None, 11456)

Dense	I/P	(None, 11456)
	O/P	(None, 1)

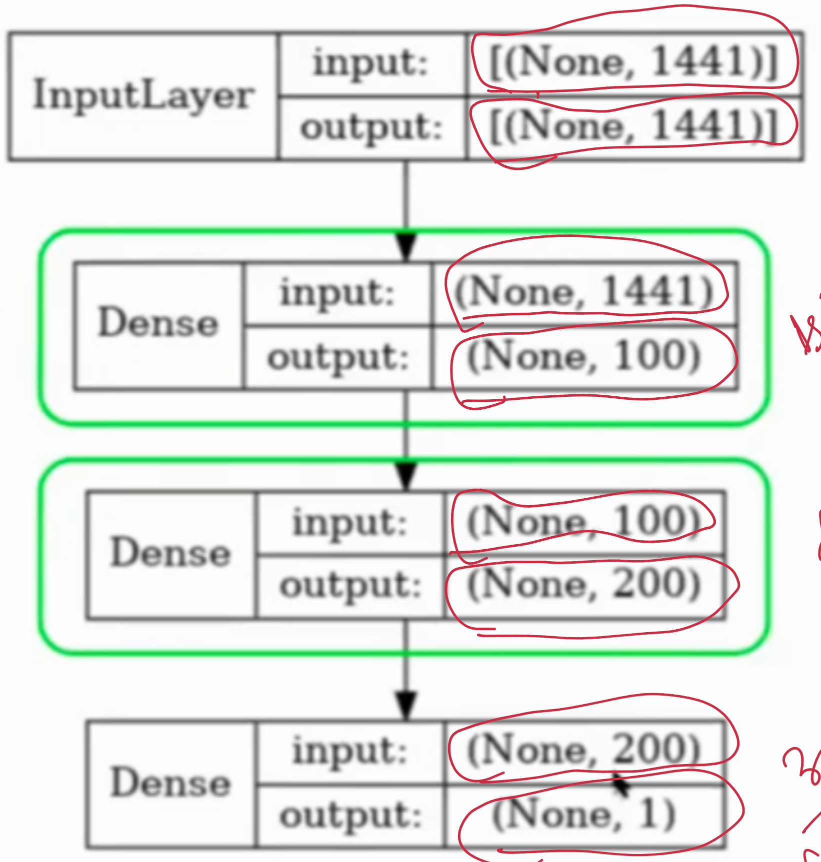
CNN - 13K paramete[rs]

Input Nodes → 1441
Output Nodes → 1441

(Input) 1441 reduces to (O/P) 100 in first Dense Layer
100 reduces to 200 in 2nd Dense layer

THIRD NEPAL WINTER SCHOOL in AI, 2021
NepAl Applied Mathematics and Informatics Institute for research

Comparing similar FCN vs CNN network sizes



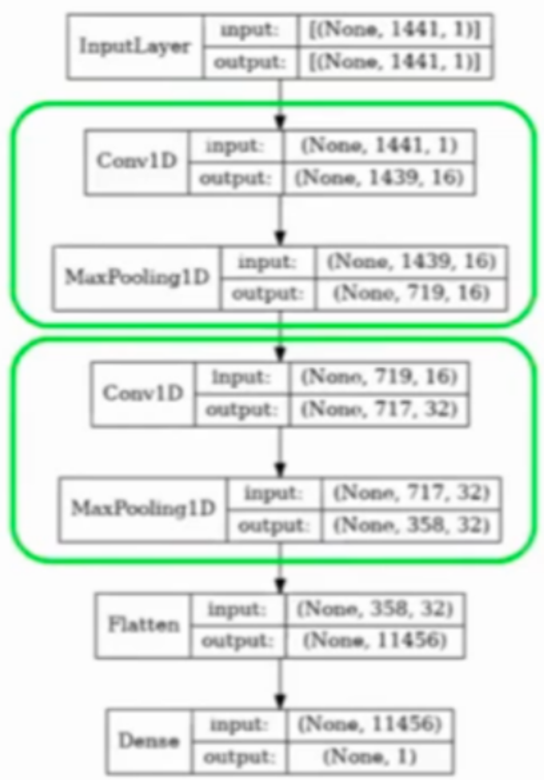
1st

2nd

3rd

1 final o/p

FCN - 160k parameters



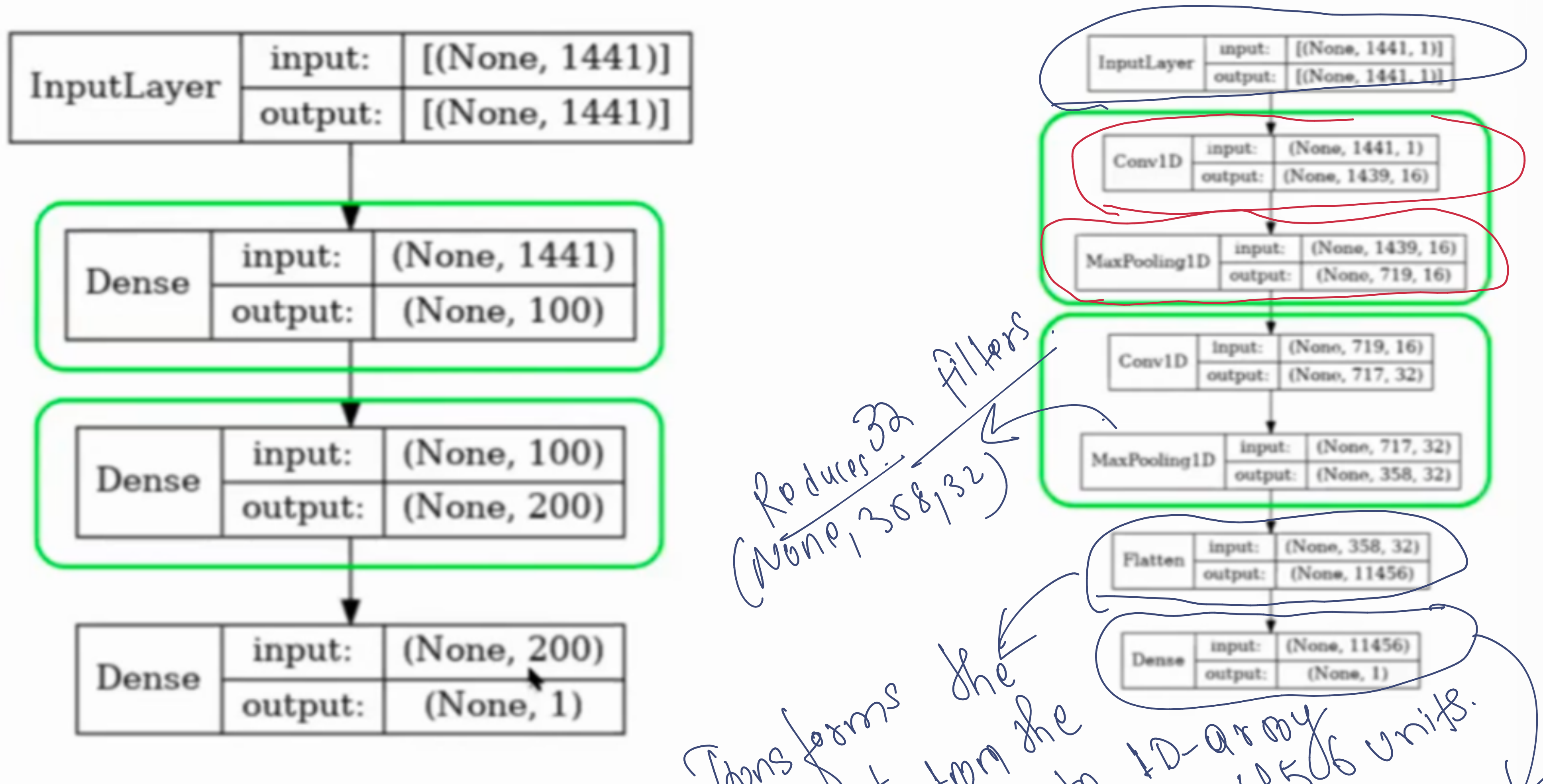
CNN - 13k parameters

nepalschool.naamii.org.np

[(None, 1441)] → None is the batch size and 1441 represents the number of features.

THIRD NEPAL WINTER SCHOOL in AI, 2021
NepAl Applied Mathematics and Informatics Institute for research

Comparing similar FCN vs CNN network sizes



FCN - 160k parameters

CNN - 13k parameters

nepalschool.naamii.org.np

Reduces 32 filters
(None, 358, 32)

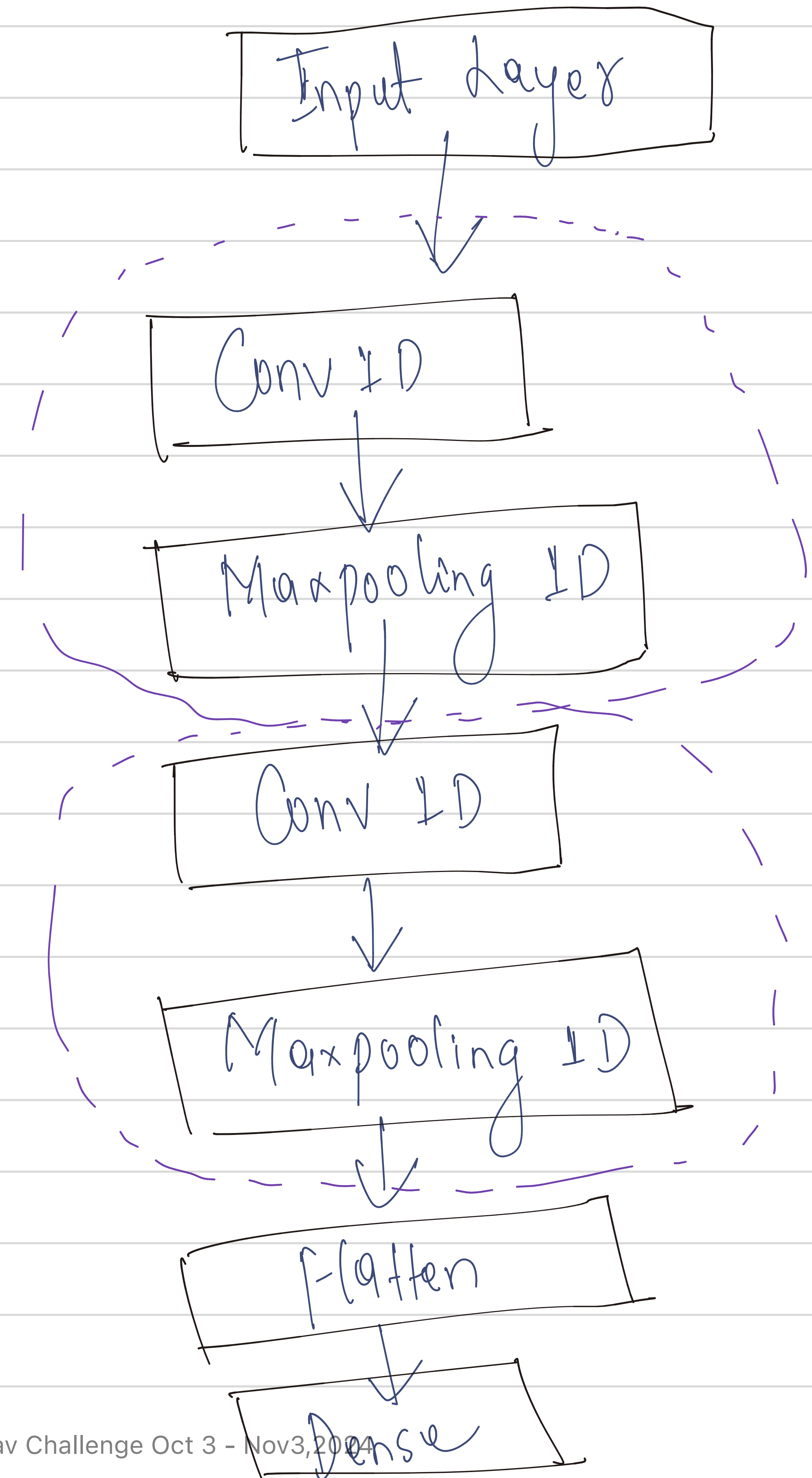
Transforms the 3D output from the CNN into 1D-array with 11,456 units.

I/p (None, 1441, 1)
&
O/p.

(None, 1439, 16)
→ 16 Filters of size 1 to the input.

Reduces 11456 to 1 final o/p.

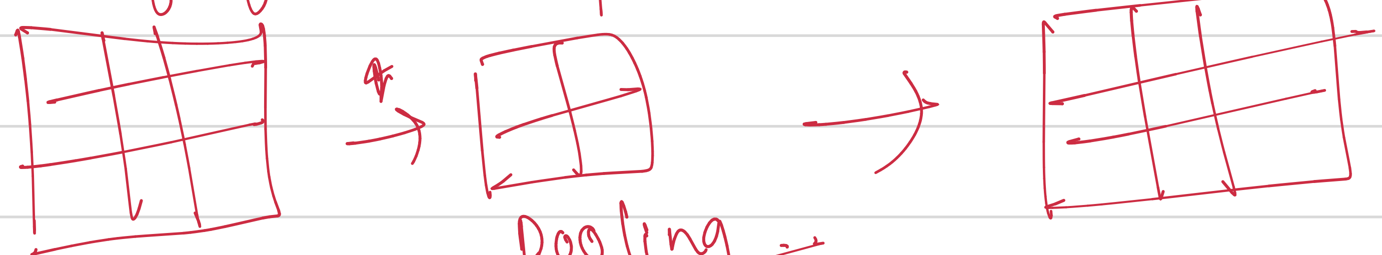
Input layer: [((None, 1441), 1)] means 1441 time steps (or features) and 1 channel (1D Convolutional for sequence data).



In CNN or ANN.

→ If function has parameters it must be differential function otherwise it is not possible to do backpropagation

Pooling: A key operation used to reduce the spatial dimension (width & height) of feature maps.



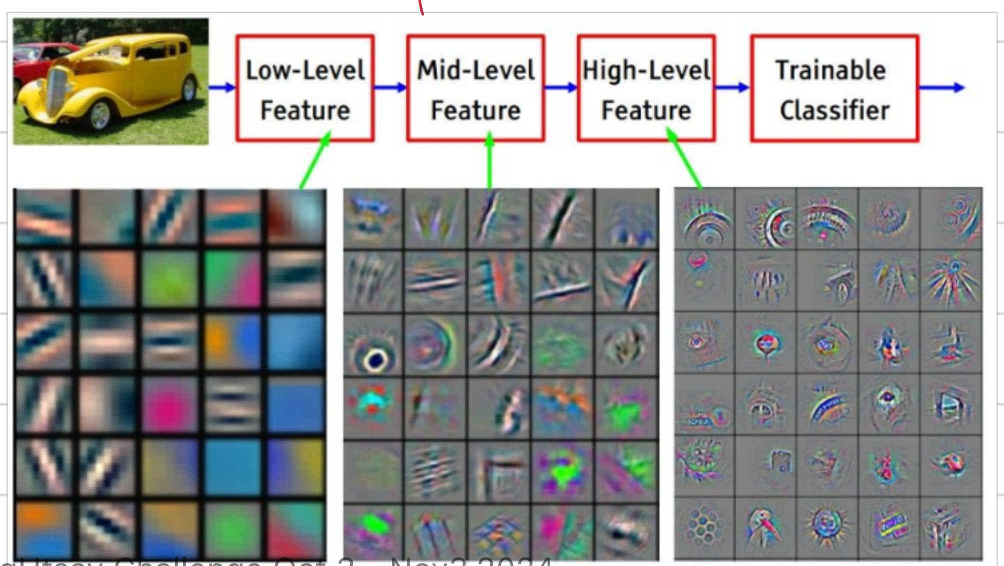
Pooling.

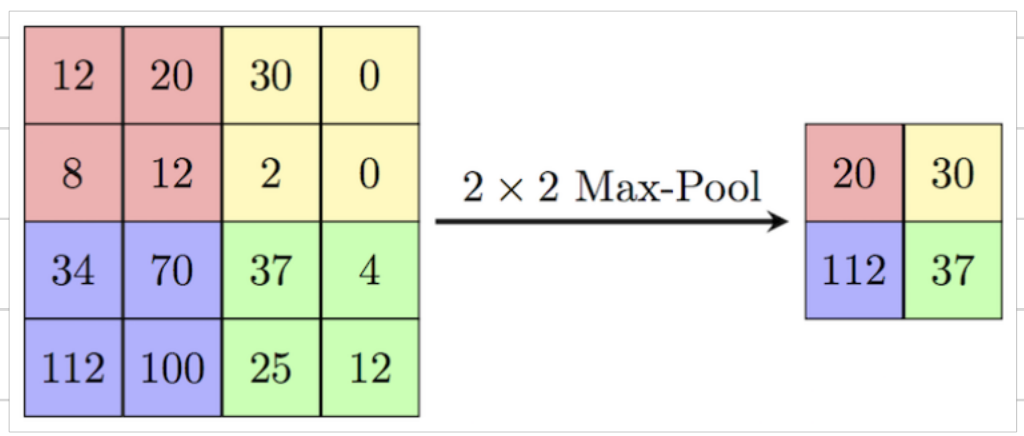
Max Pooling: for each window it selects max value can be 2x2 or 3x3

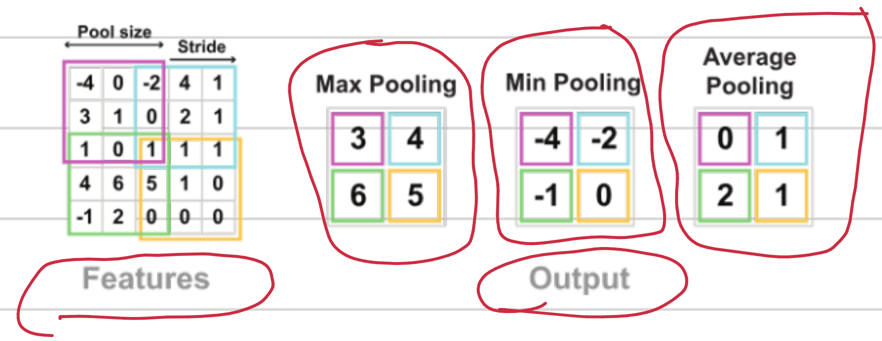
Why CNN's with 13K-parameters than 160K FCN parameters?

→ Pooling process for the entire Image (Kernel 3x3 or 2x2) window Sliding over the given Image Matrix + Gradient Descent and other Optimization techniques.

Feature Map are O/p of a Convolutional layer in CNN, represent the learned patterns or features extracted from the input data







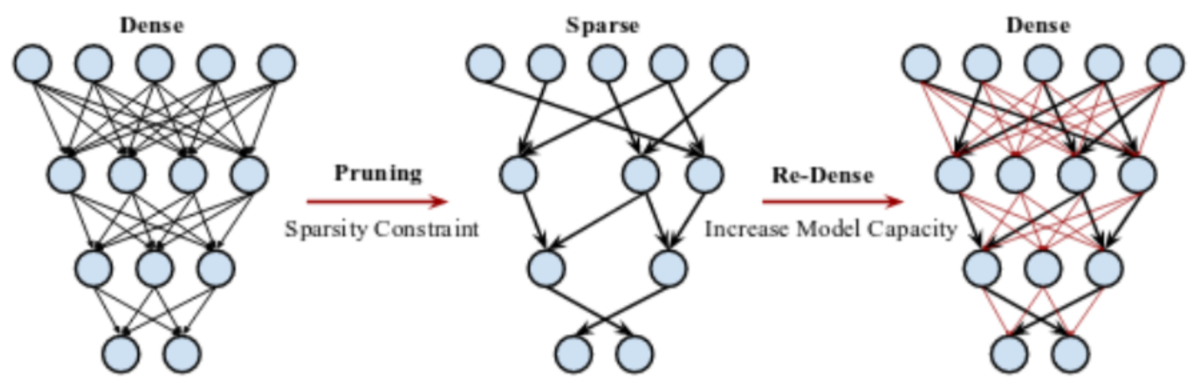
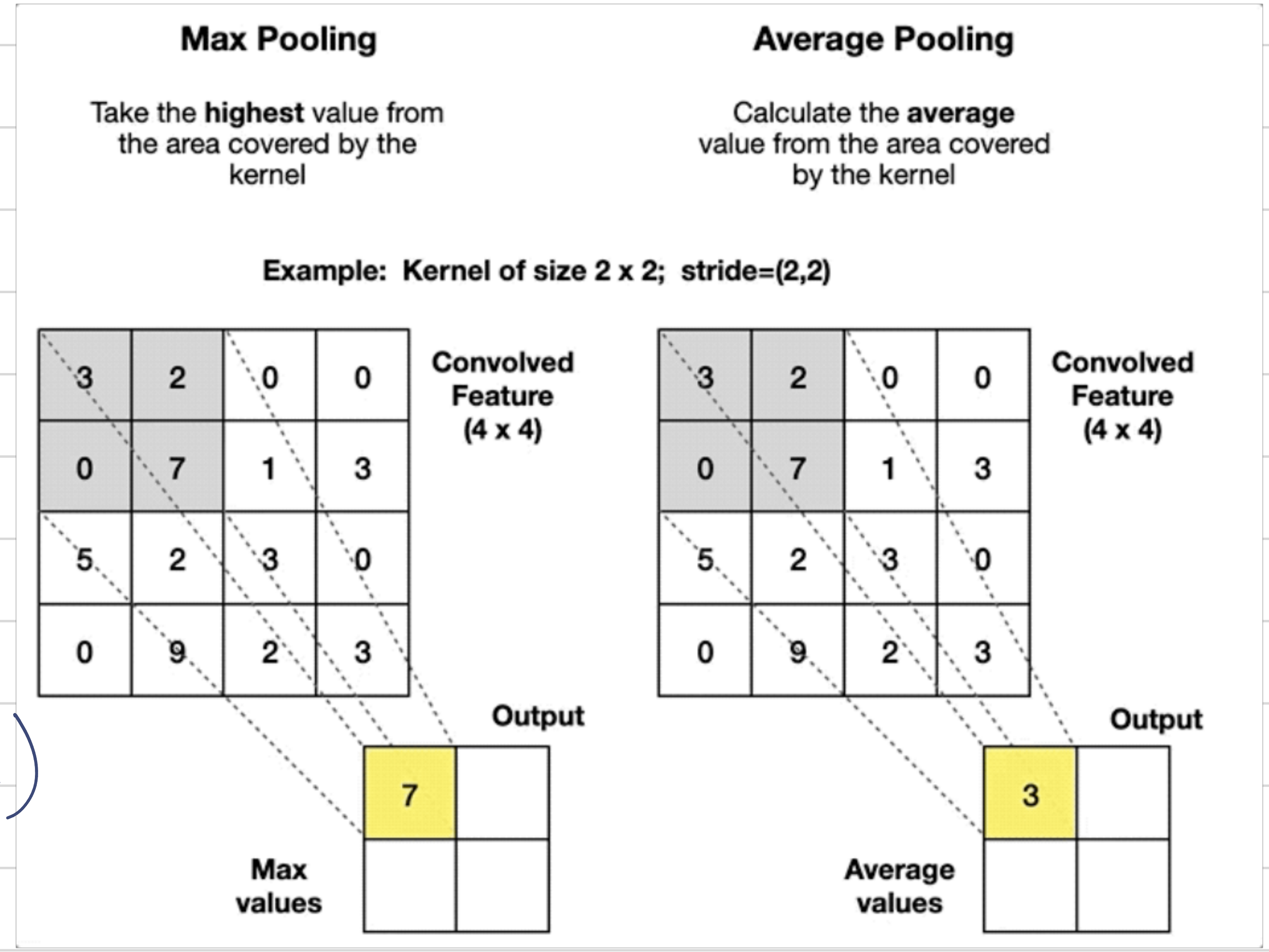
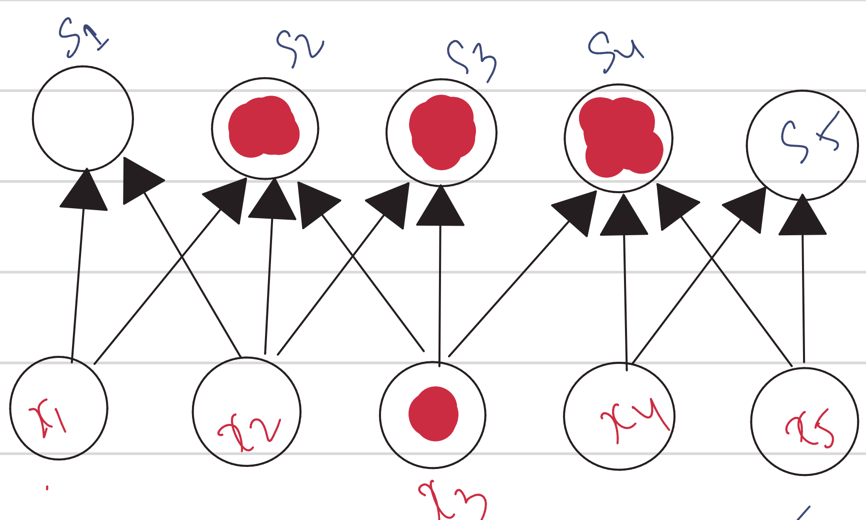
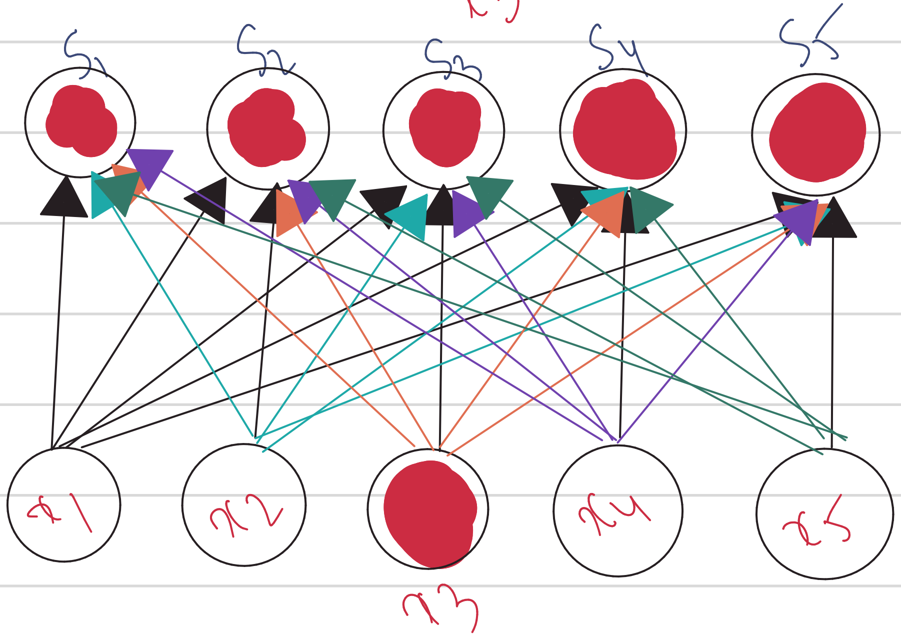


Figure 1: Dense-Sparse-Dense Training Flow. The sparse training regularizes the model, and the final dense training restores the pruned weights (red), increasing the model capacity without overfitting.



Src: deeplearningbook.org

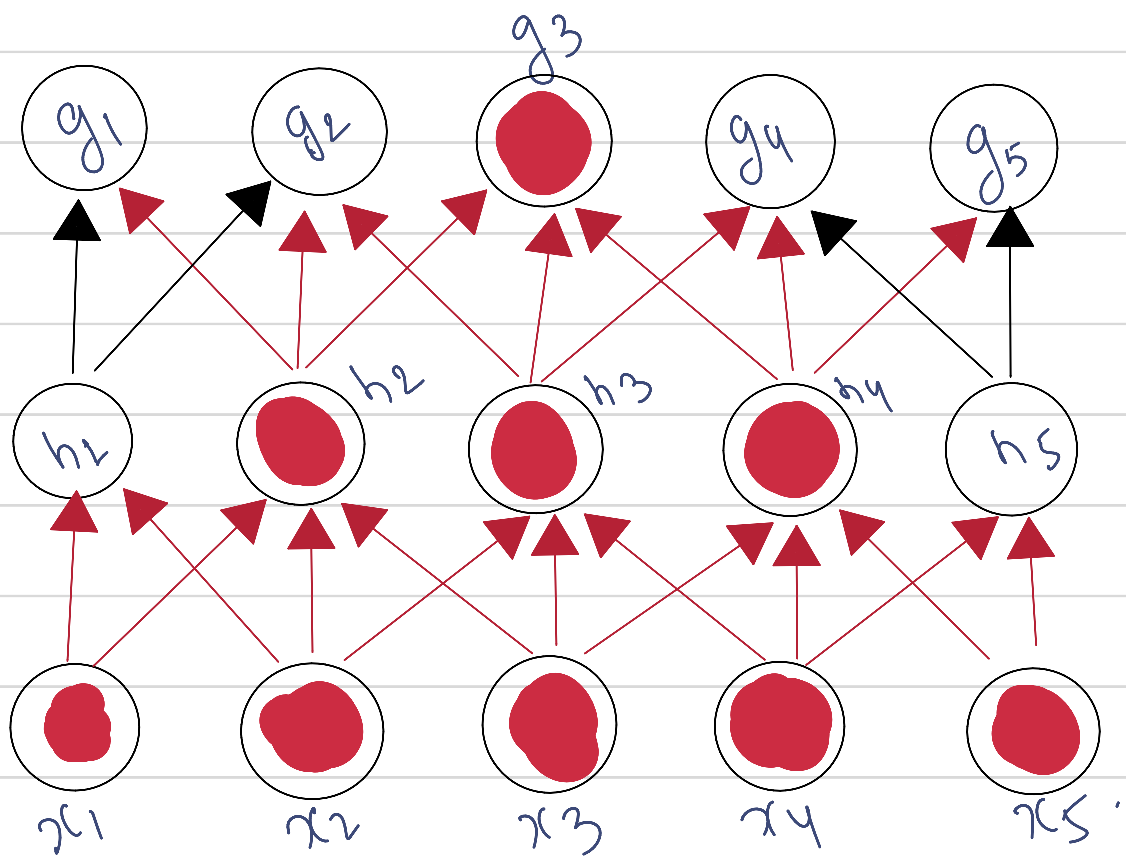




Top: Convolutional Connections - the input unit x_3 only influences the output units S_2, S_3 & S_5.

Bottom: Dense Connections - the input unit x_3 influences all the output units S_1, S_2, S_3 & S_4.

Receptive Field



As more layers are added into network, the outputs at the deeper layers are affected by more and more input.

h_3 is affected only by (x_2, x_3 and x_4)

However, g_3 is affected by all x's.

→ So, the intuition here is that as we go deeper our idea of sparsity doesn't work and all the features becomes globalization and this is what we call Receptive Field.

At the final dense layer the Receptive field must be dense

and high if not then the model's Performance is never optimized.

So, the receptive field at the final should be bottom-to-top and it must make sense at the final layer.

CNNs in Action: the filters:

Since + (left) & (-) at right

Edge detection /

✓ Edge Detector

-0.47	-0.24	-0.87

0.09	0.09	0.33

0.62	-0.09	-0.50

Smoothing

Three example filters in layer one of the earlier EKG example.

-0.47	-0.24	-0.87

0.00	0.004	0.1

Some filters from the second layer. The 2nd layers' filters are actually of dimensions (3x16) since there are 16 channel outputs from the first layer. The 2nd layer filters therefore expand their influences across all channels. 2nd filter is almost 0.

→ 3x1 dimensions filter means they have 3 rows and 1 column

→ values in the filters represent the weights applied to input

→ Second filters from the Second layer –

3x16 → 3 rows and 16 columns (16 o/p channels of first layer)

and [0.00, 0.004, 0.1] means the filter has learned a weaker pattern from the data because weights are 0.00, 0.1 --

1st layer filters learn to highlight certain patterns within the input data by convolving across it.

It shows how the filters are used to extract features from the input data and glimpse of inner workings of CNNs.

Results

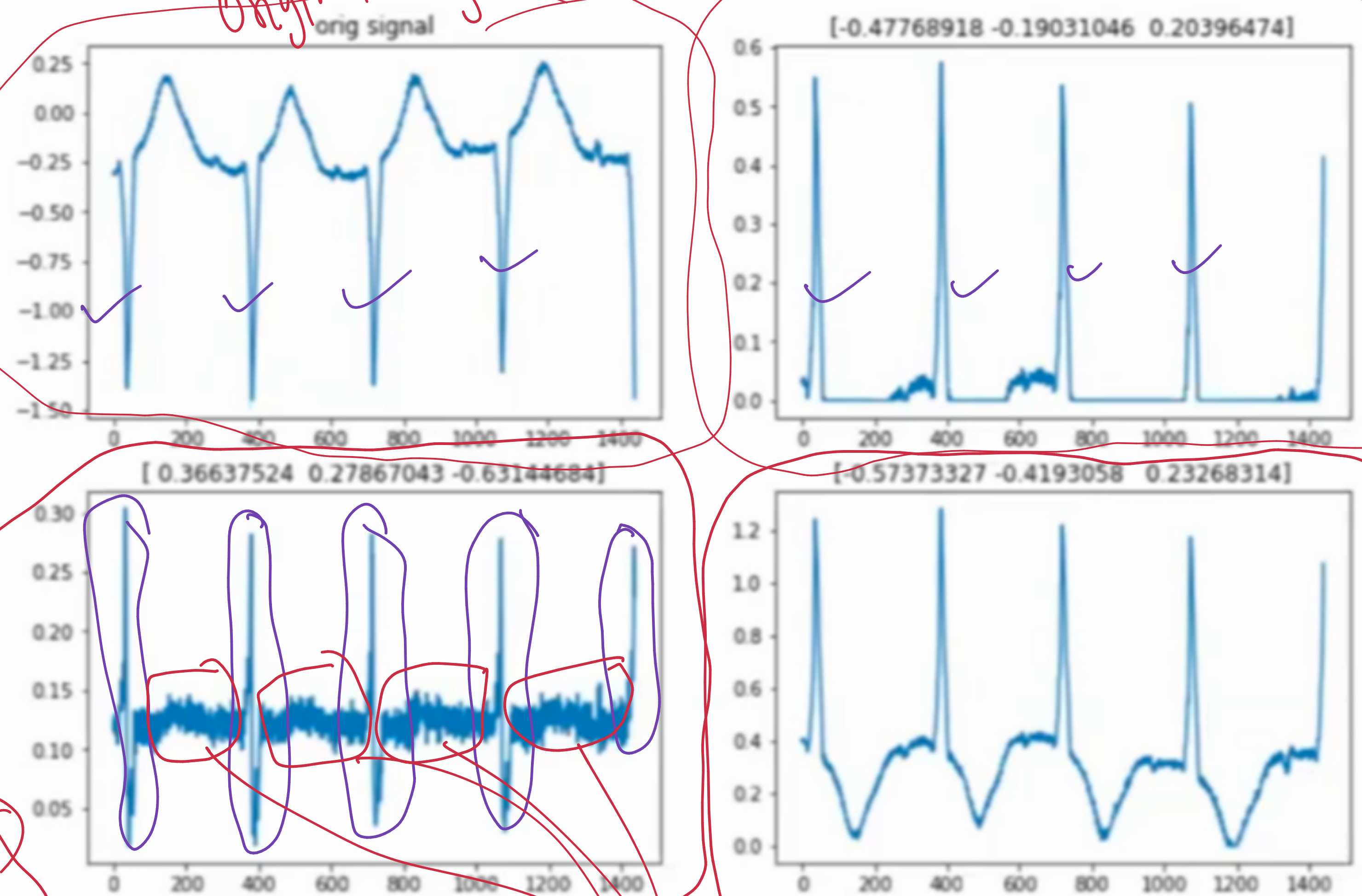
CNNs in action: activation (layer 1)

THIRD NEPAL WINTER SCHOOL in AI, 2021
NepAl Applied Mathematics and Informatics Institute for research

Here -0.4193058 and 0.23268314 can be Edge Detector

→ Flipping or Simply the Inverse Must have used ReLU?

Original Signal



$I(x)$

$I'(x)$

Edge Detection

Orig signal, filters and activations in the 1st layer.

ReLU $[0, \infty)$

nepalschool.naamii.org.np

Edge Detector → derivative of $I(x)$

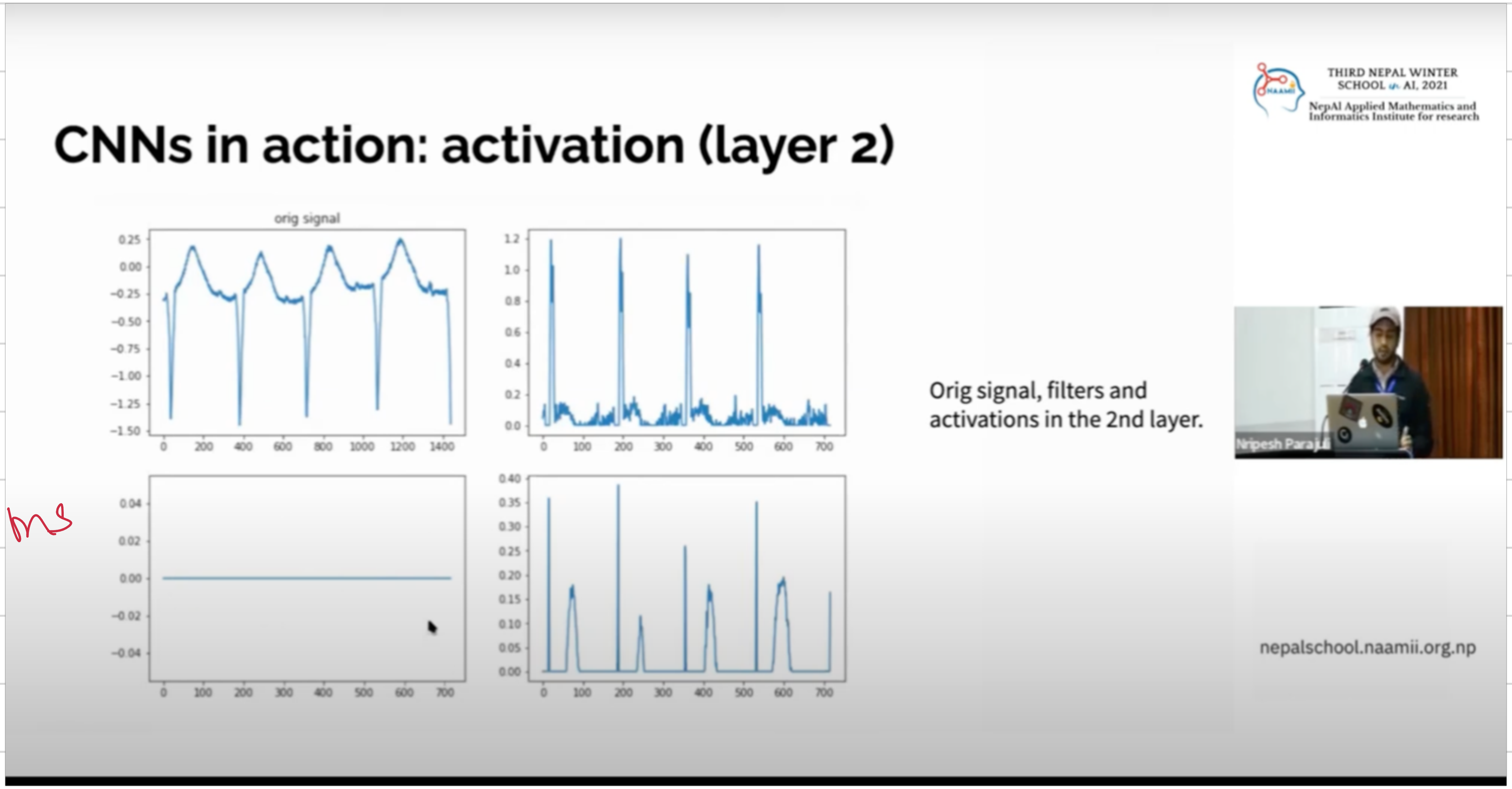
Noise: are the result of derivative.

$I(x) \rightarrow I'(x)$

Filters are responsive to n-number of Channels - Example 16- Channels from the first layer.

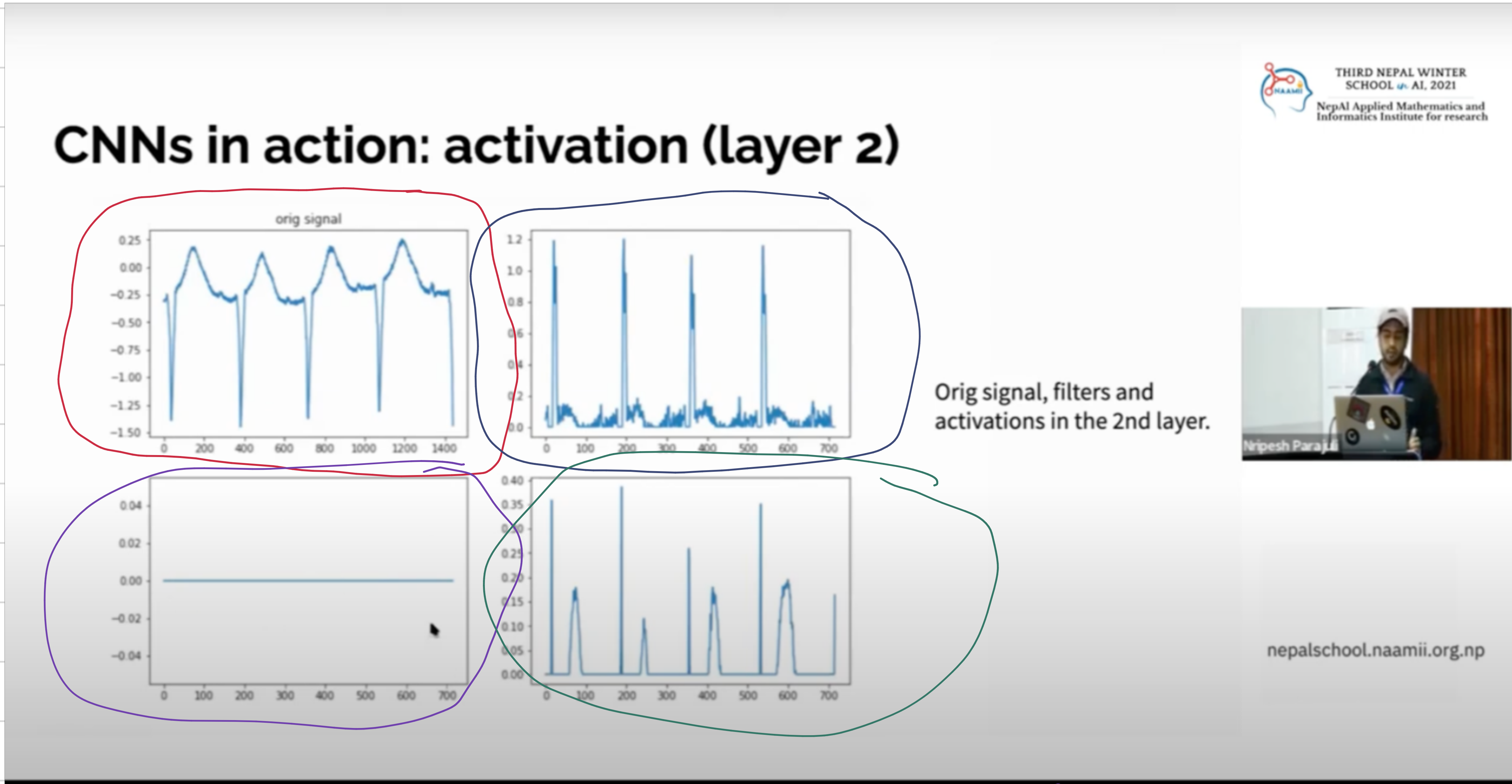
Original Signal

Original Signal applied the filters to extract features and activations present the output of filters.



Input Signal → Applied filters → Extract patterns

① Original Signal



② Filtering the Original Signal. Most prominent features detected by a particular field.

③ Bottom-left plot: flat showing that the corresponding filter did not detect significant features from the original signal.

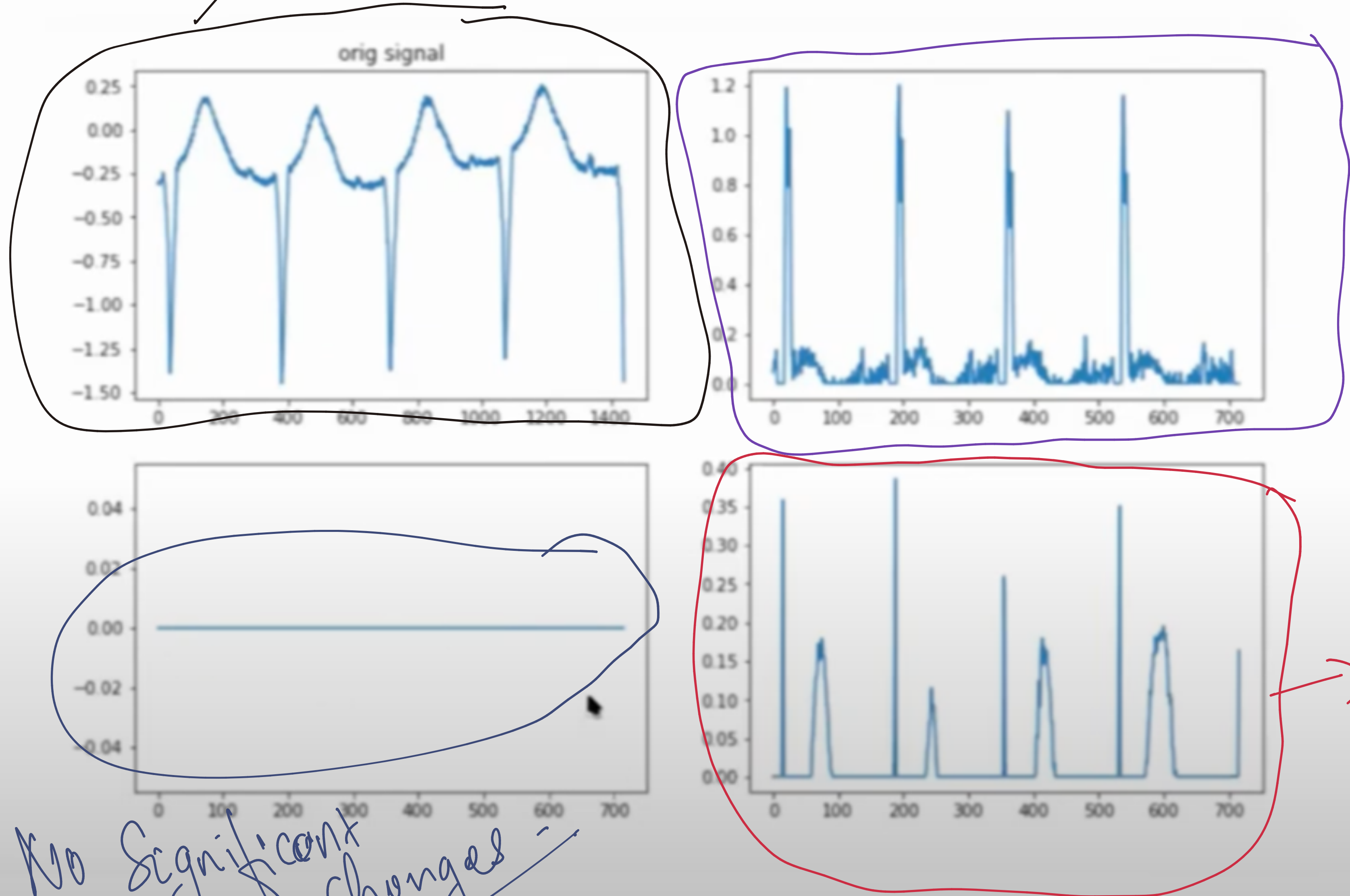
④ Bottom-Right plot:- Spikes with a different pattern, suggesting another feature extraction!

→ original Signals.

THIRD NEPAL WINTER SCHOOL in AI, 2021

NepAl Applied Mathematics and Informatics Institute for research

CNNs in action: activation (layer 2)



No Significant changes.

→ picked both +ve & -ve Signals.

Orig signal, filters and activations in the 2nd layer.

ReLU $[0, \infty) \rightarrow$ non-linear.

nepalschool.naamii.org.np

These are all plots derived from using Activations.

Knowledge Representation Methods:

① Propositional Logic: $\neg, \wedge, \vee,$

→ $\neg P(x)$: x is a dog.

→ $(P \wedge Q)$: P is a water conductor of electricity
Q is a copper " " "

Water and Copper conductor of electricity

→ $(P \vee Q)$: P is a cat pet animal
Q is a dog pet animal.

Cat or dog is a pet animal.

$\Rightarrow$: if it rains then the road gets wet

$\Leftrightarrow$ A number is prime if and if it has GCD 1 and Co-factor of 1 dividing by itself none than other number.

② Propositional Logic vs Predicate Logic vs FOPL

Propositional Logic → simple, inference, semantic & expressive with limitations like AND, OR, NOT, $\Rightarrow$, $\Leftrightarrow$

Predicate Logic: Extension of Propositional Logic - with and without quantifiers.

$\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$, $\exists$, $\forall$

First Order Predicate Logic is extension of Predicate Logic that can be used for objects in the domain.

Example: $\forall x\,(p(x) \wedge (\exists y\,(Q(y) \vee \neg R(x,y))))$

So, Above FOPL can be read as -

$\forall x \Rightarrow$ the universal existential of 'x' objects.

$p(x) \Rightarrow$ Any p is an object.

$\exists y \Rightarrow$ there exists y

$V \Rightarrow$ OR (union operation)

$\neg \Rightarrow$ NOT operator

$R(x,y) \Rightarrow$ Two objects

So, FOPL can be used to express Complex Statements, Show relations and we can state that -

$\neg$ Brother (leftLeg(Richard), John)

Richard left leg is not John's leg and Both are Brother . Richard or

John is King.

$\neg King(Richard) \Rightarrow King(John)$

if Richard is not King then John also not a King.

$\forall x (P(x) \wedge (\exists y (Q(y) \vee \neg R(x,y))))$

The universal existance of x if $p(x)$ and there exists property y which is $Q(y)$ and not $R(x,y)$ that means - y has 'no' any relation R with $Q(y)$ and $R(x,y)$.

Final.

For all x, if x has property P, then there exists a y such that either y has property Q or x and y don't have relation R.