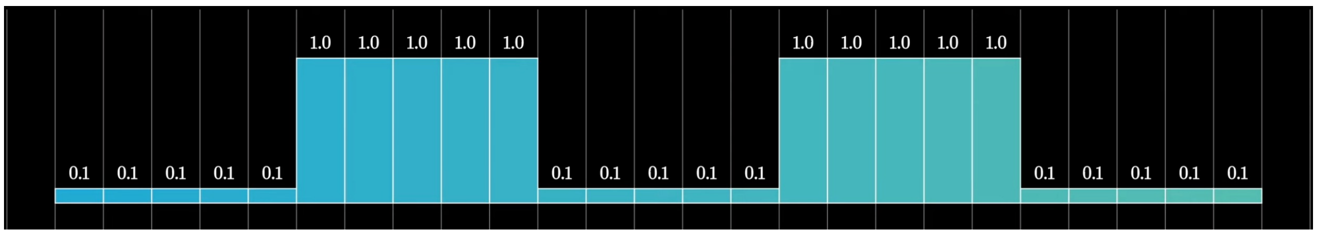
Deep Learning applied to EEG

Dung "Young" Truong & Arnaud Delorme

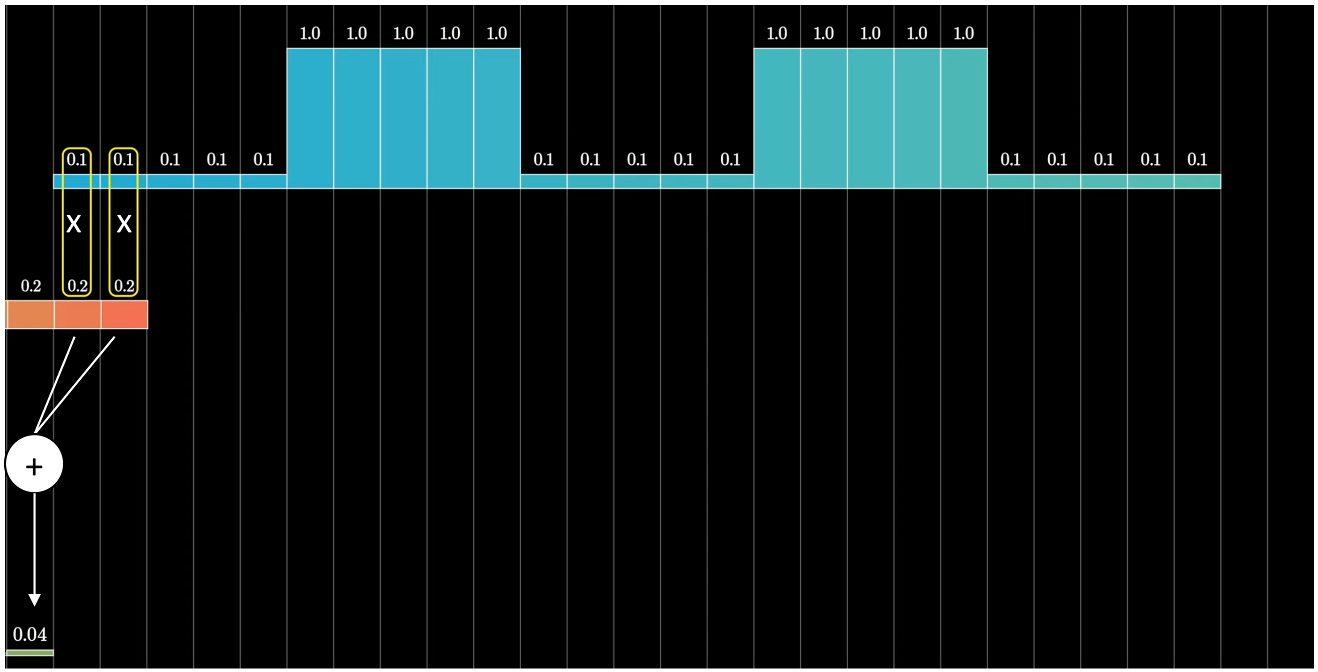
What kind of operation?



3Blue1Brown

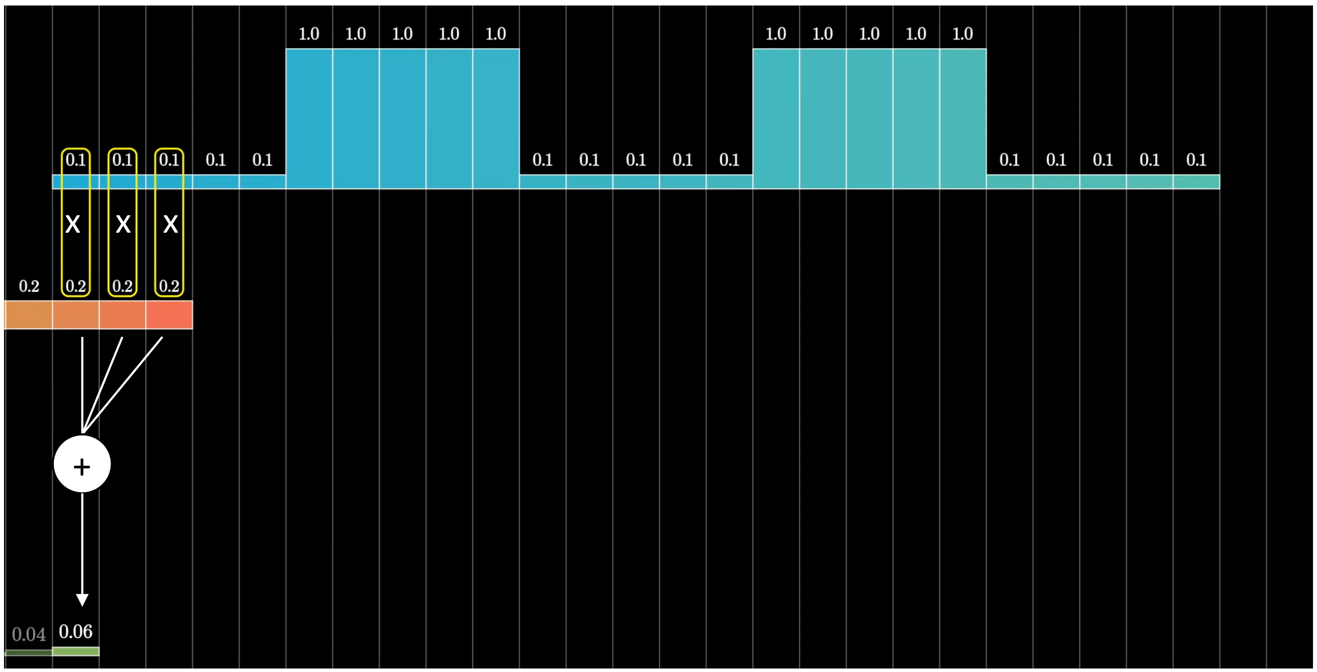
https://youtu.be/KuXjwB4LzSA

What kind of operation?



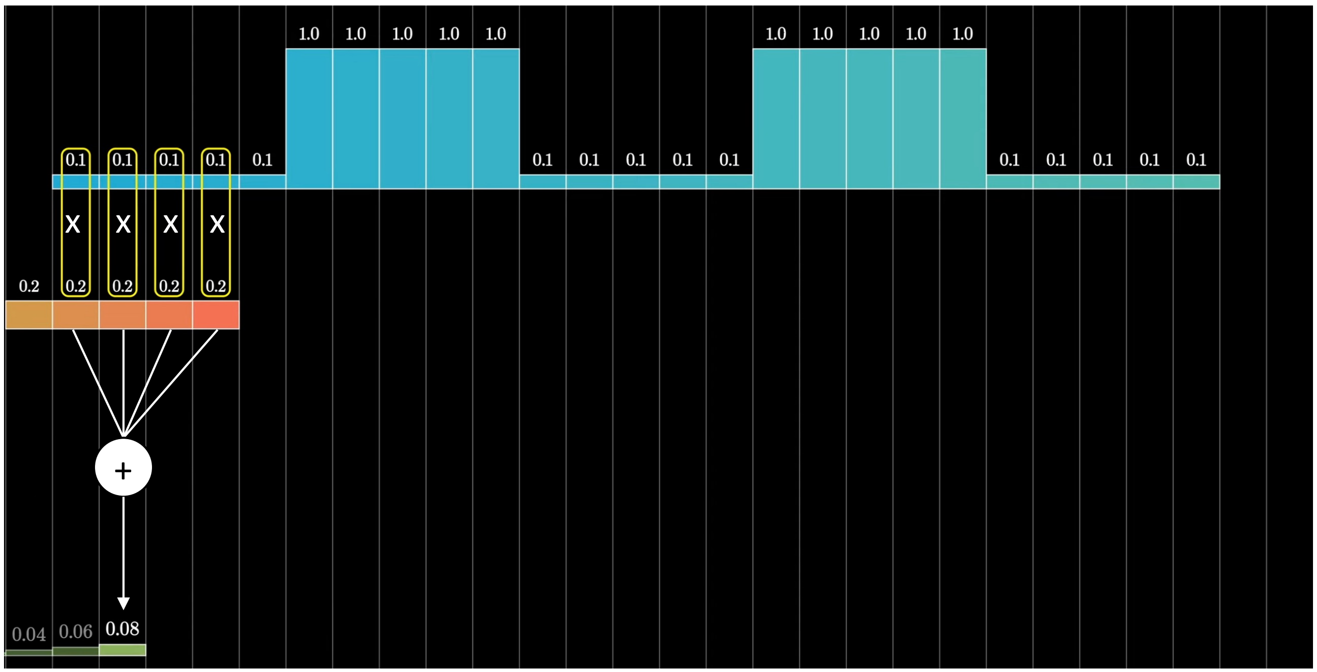
https://youtu.be/KuXjwB4LzSA

What kind of operation?



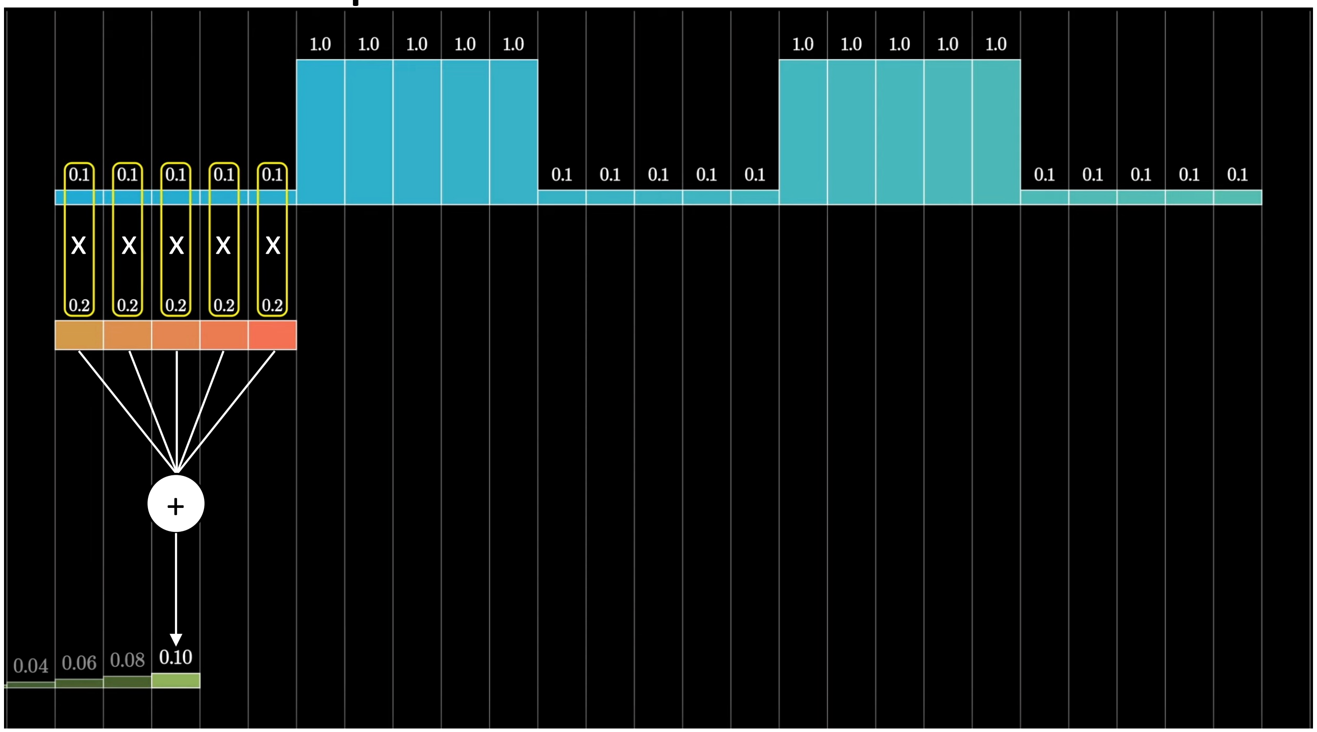
https://youtu.be/KuXjwB4LzSA

What kind of operation?



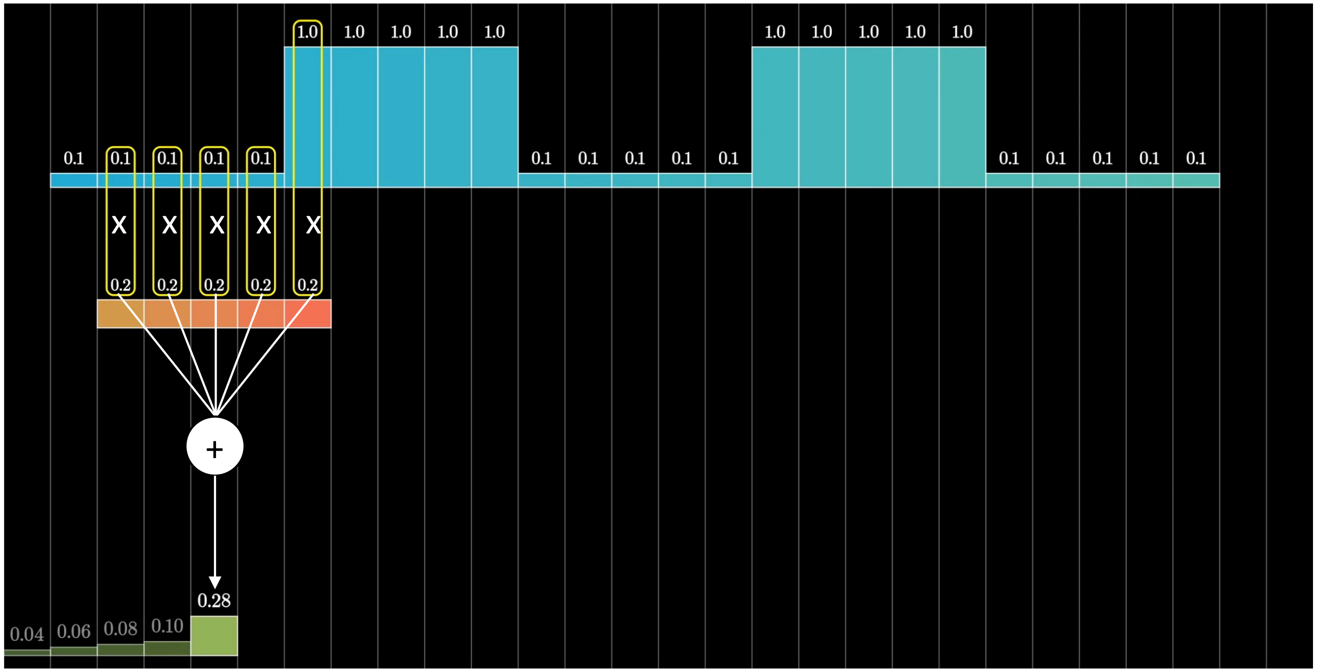
https://youtu.be/KuXjwB4LzSA

What kind of operation?



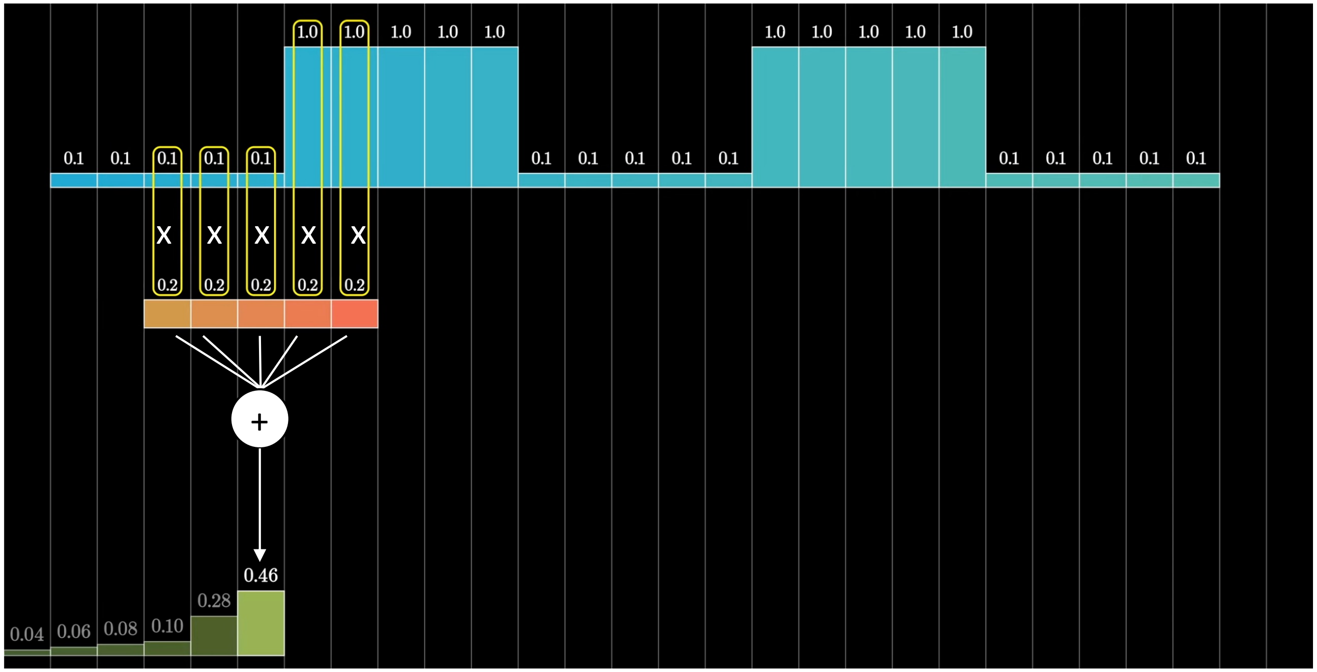
https://youtu.be/KuXjwB4LzSA

What kind of operation?



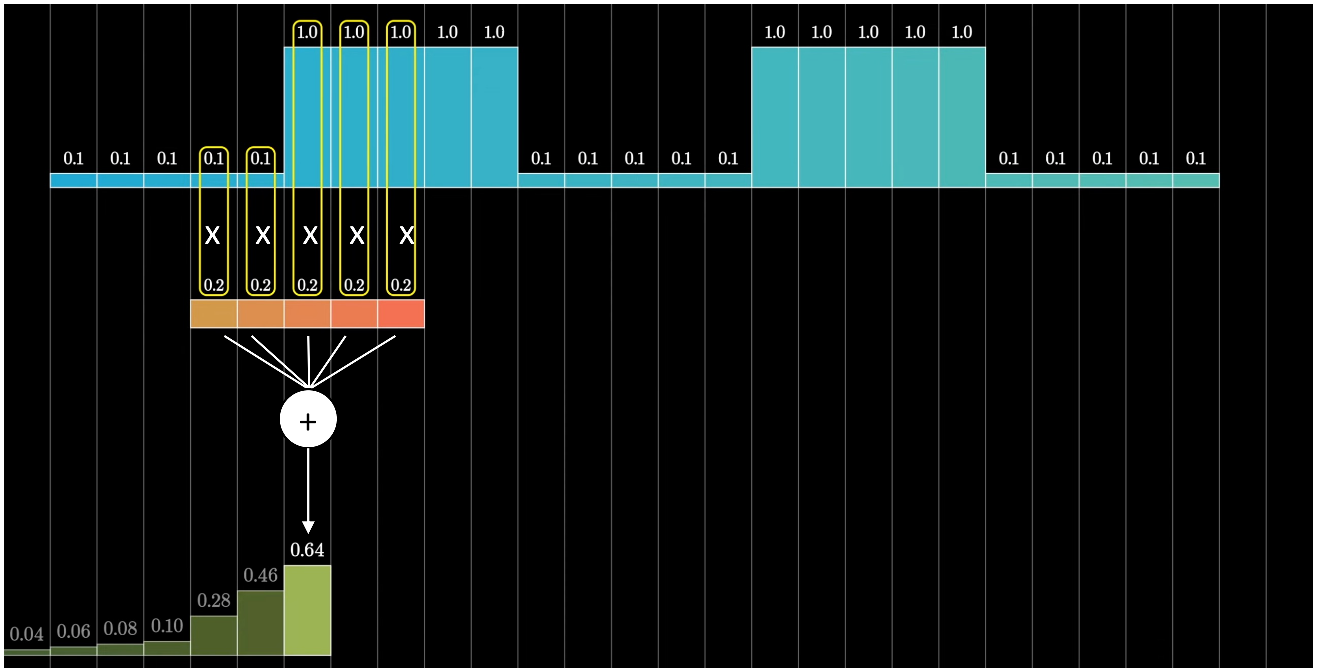
https://youtu.be/KuXjwB4LzSA

What kind of operation?



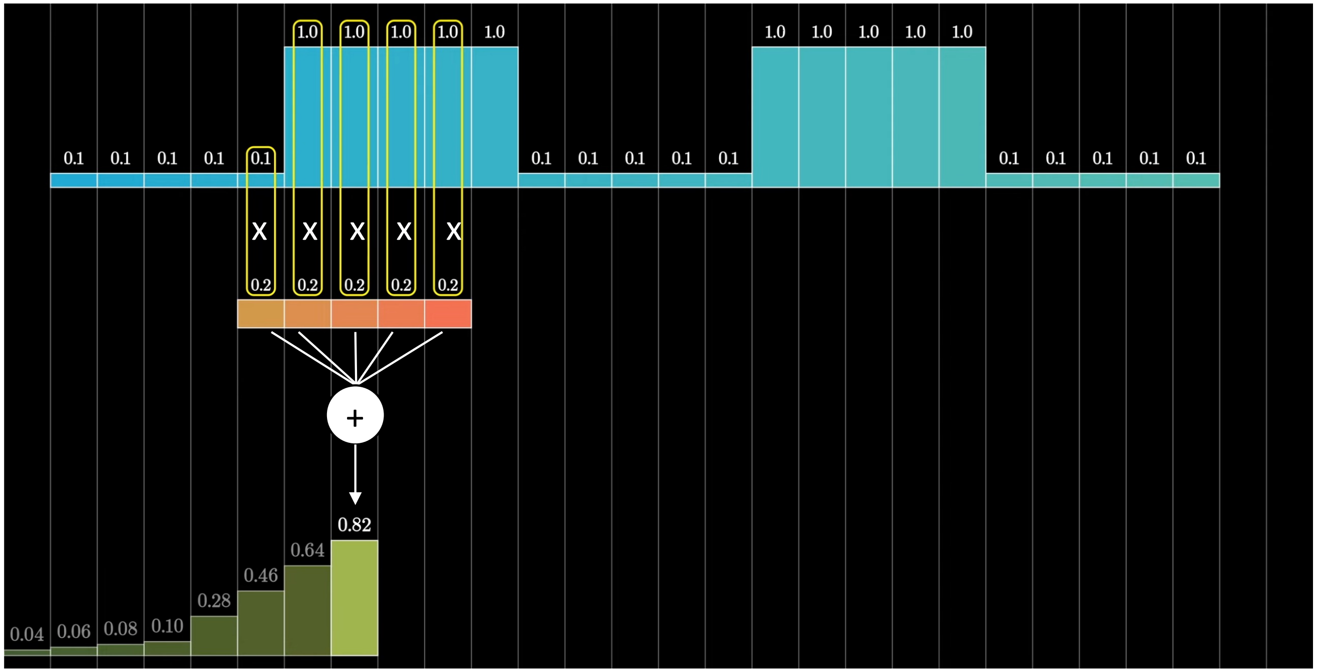
https://youtu.be/KuXjwB4LzSA

What kind of operation?



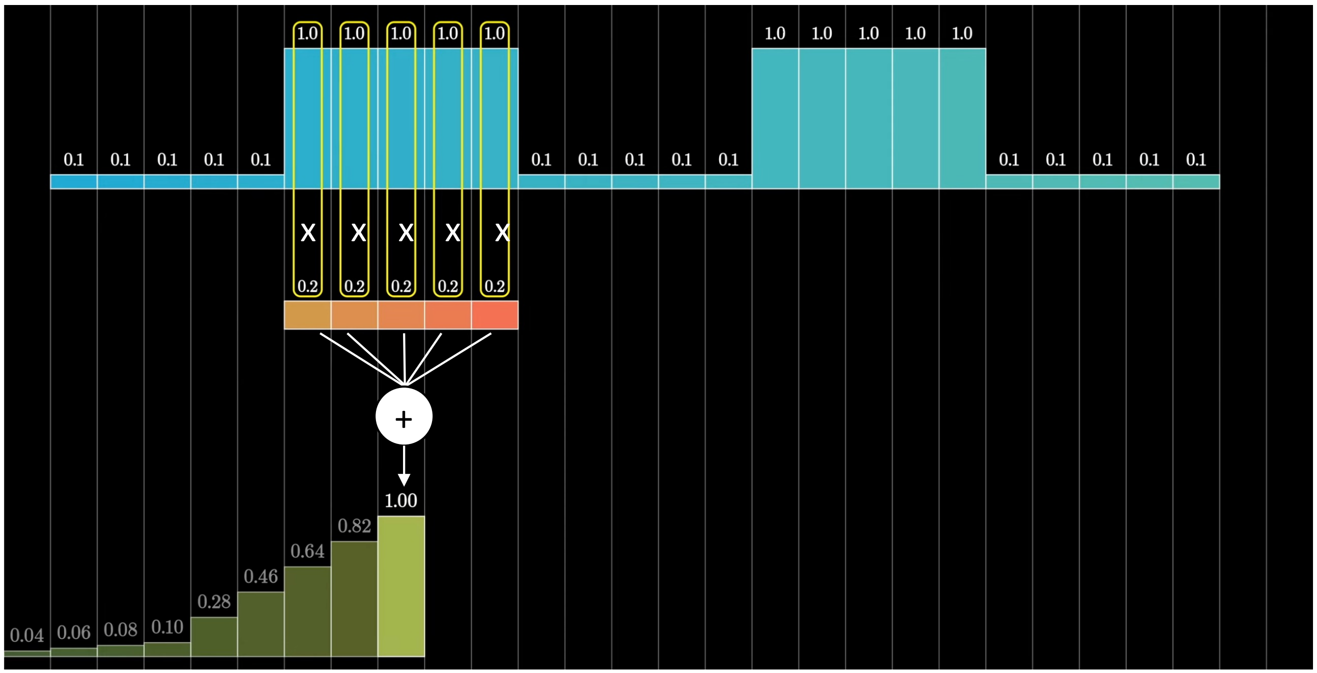
https://youtu.be/KuXjwB4LzSA

What kind of operation?



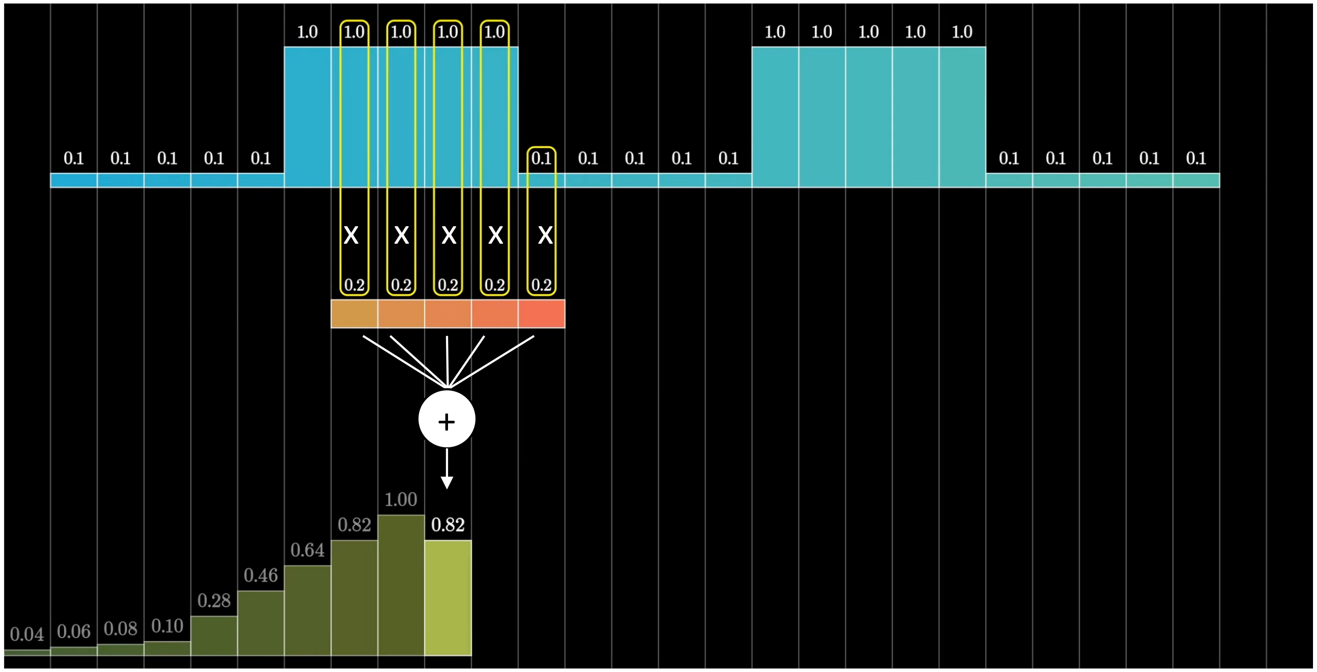
https://youtu.be/KuXjwB4LzSA

What kind of operation?



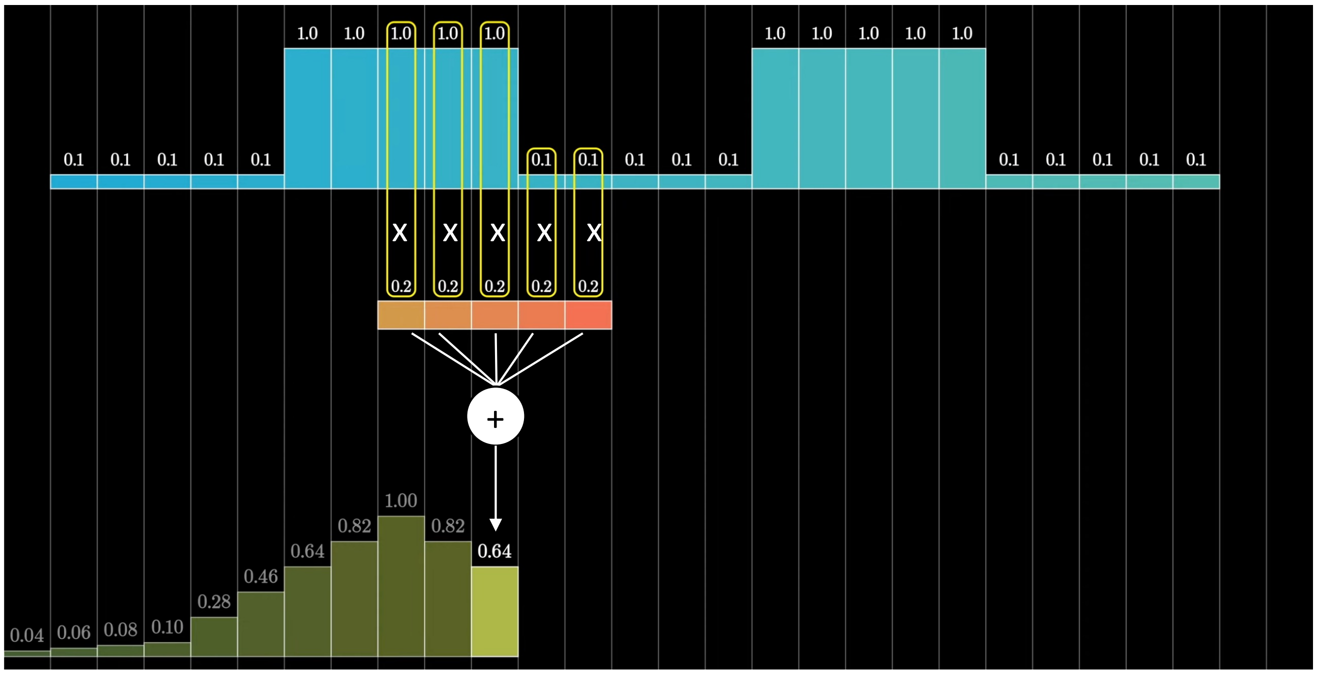
https://youtu.be/KuXjwB4LzSA

What kind of operation?



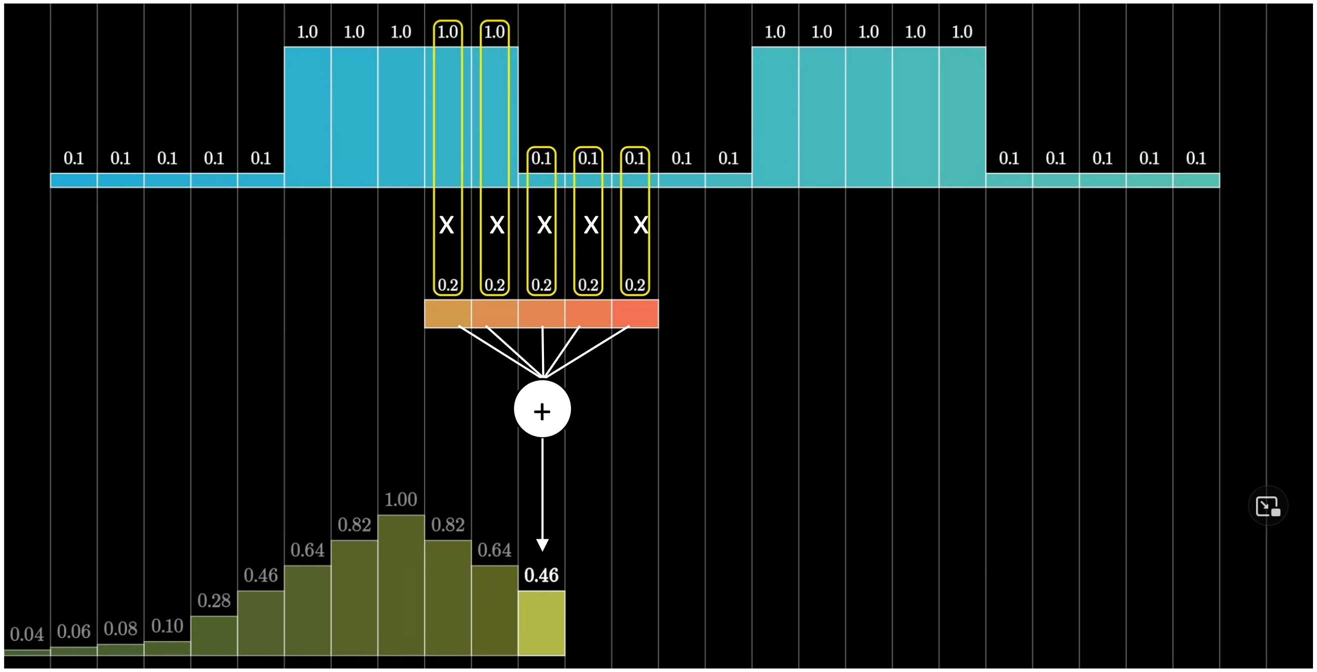
https://youtu.be/KuXjwB4LzSA

What kind of operation?



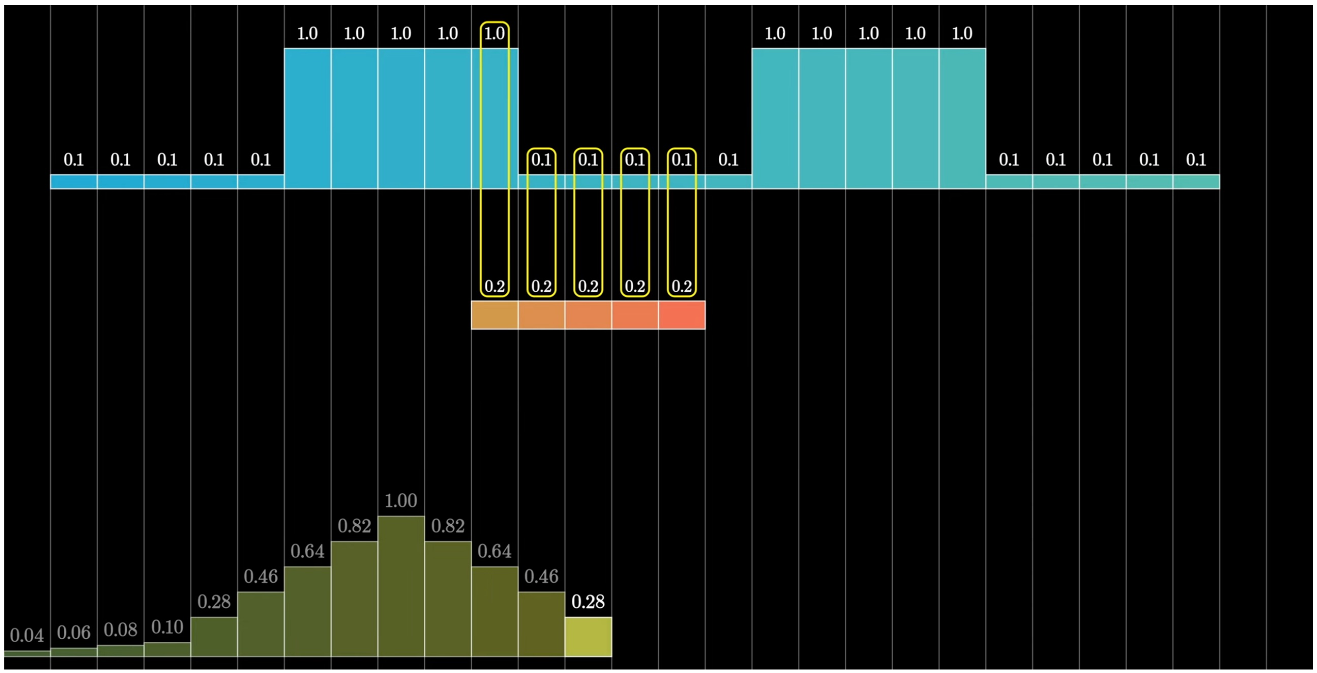
https://youtu.be/KuXjwB4LzSA

What kind of operation?



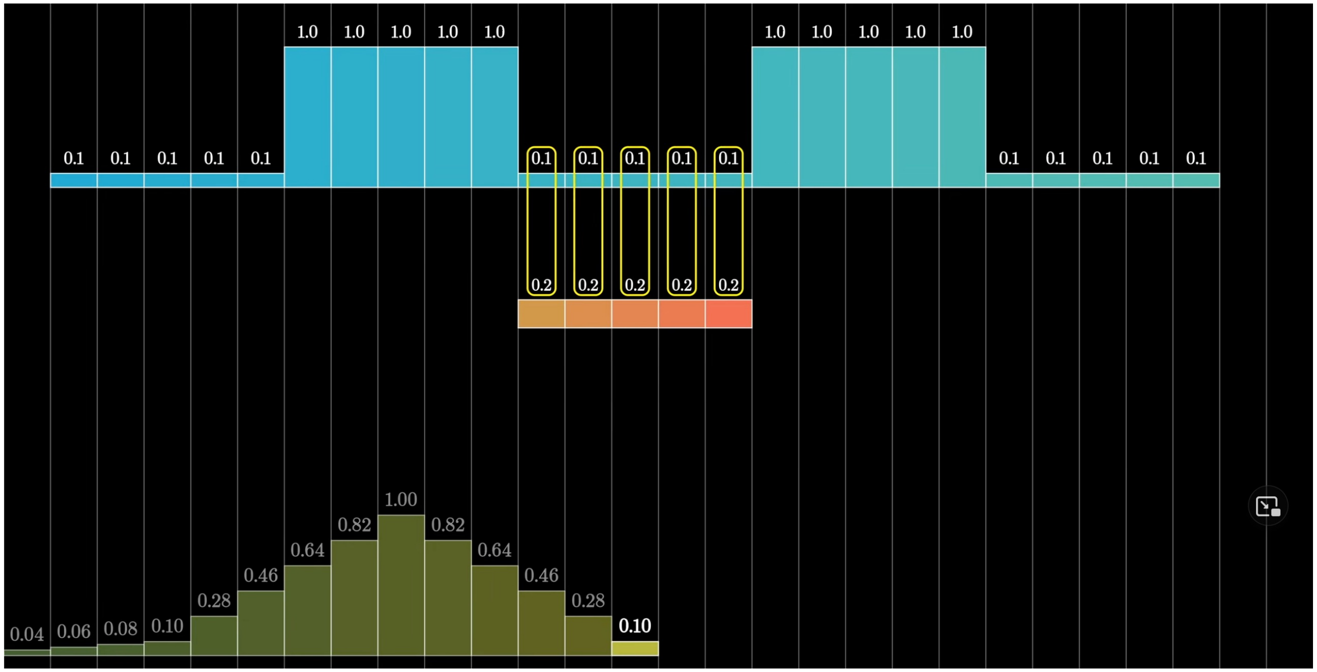
https://youtu.be/KuXjwB4LzSA

What kind of operation?



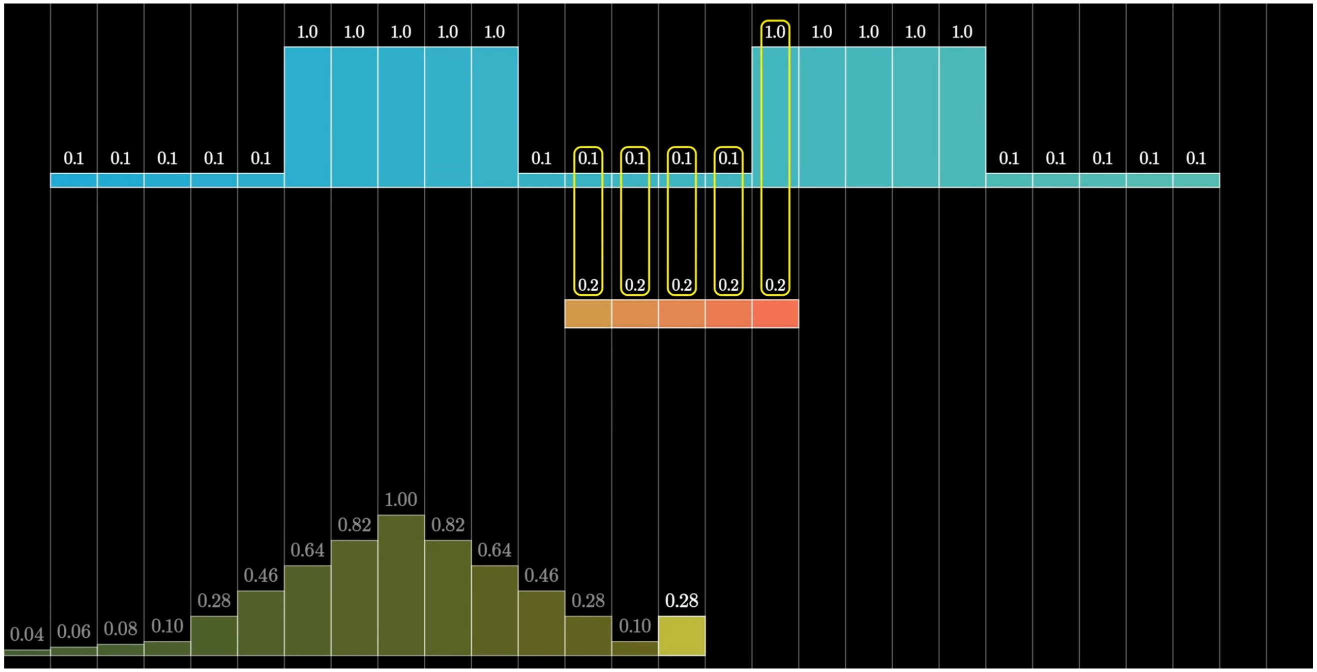
https://youtu.be/KuXjwB4LzSA

What kind of operation?



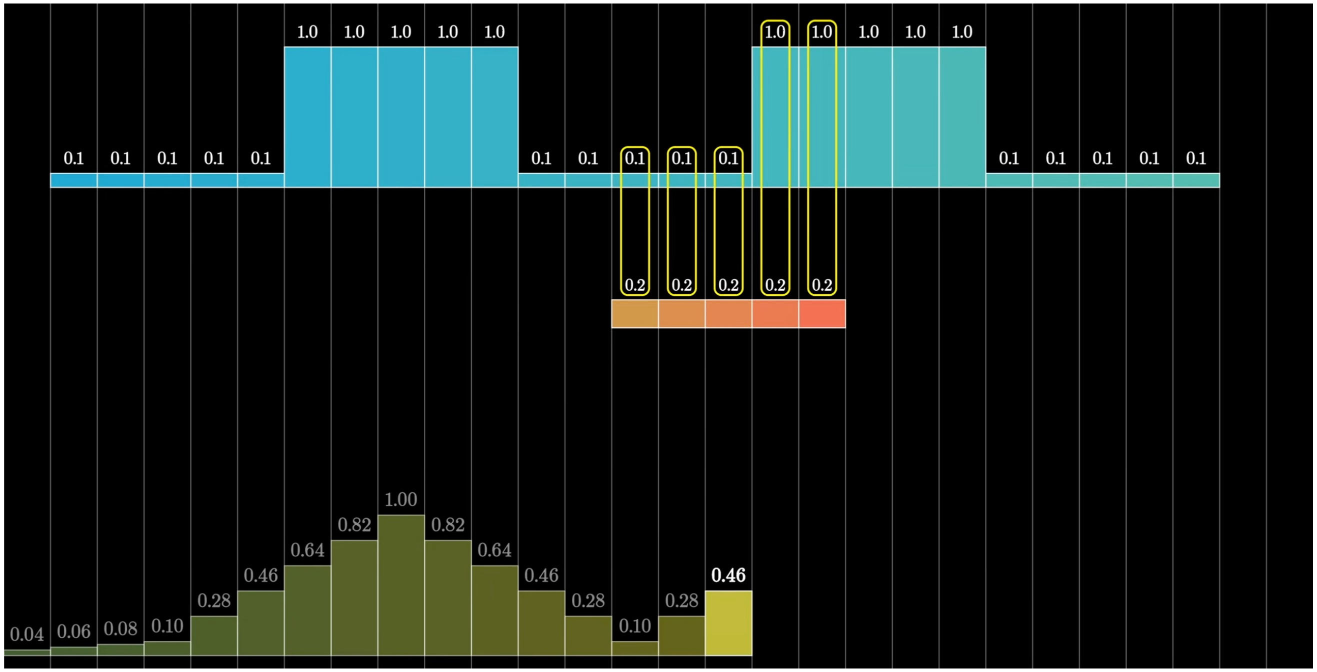
https://youtu.be/KuXjwB4LzSA

What kind of operation?



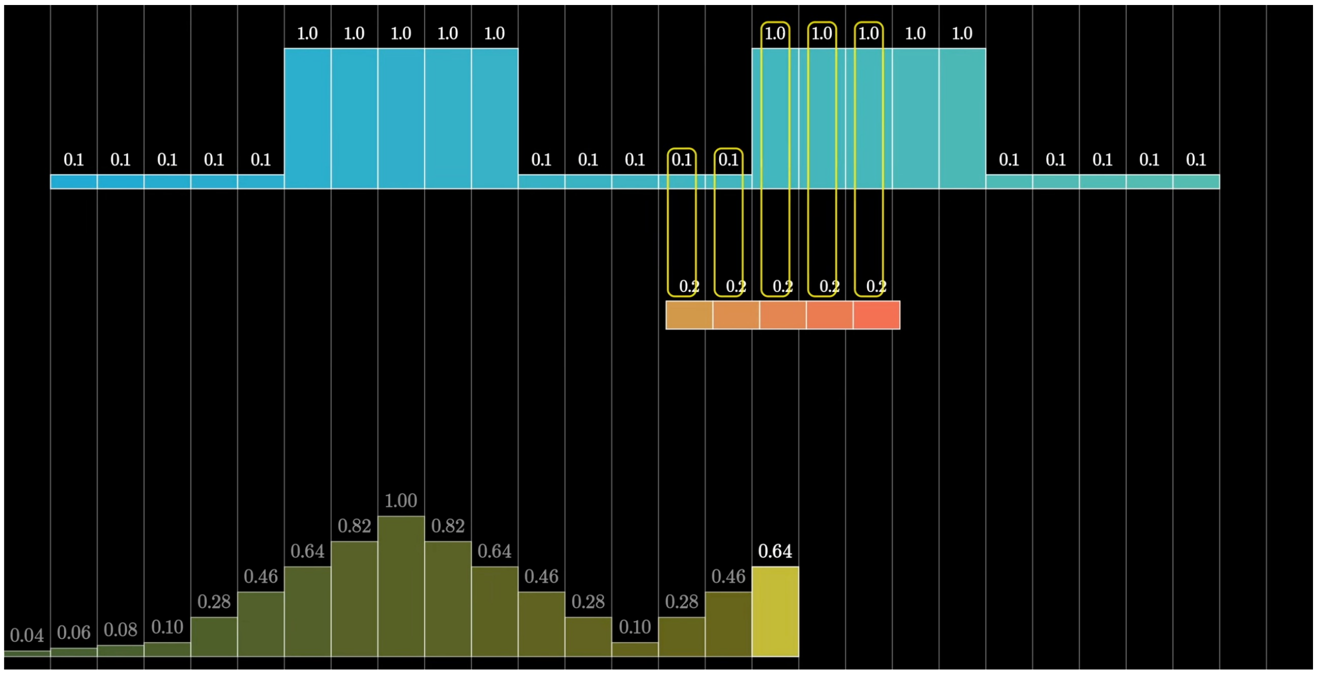
https://youtu.be/KuXjwB4LzSA

What kind of operation?



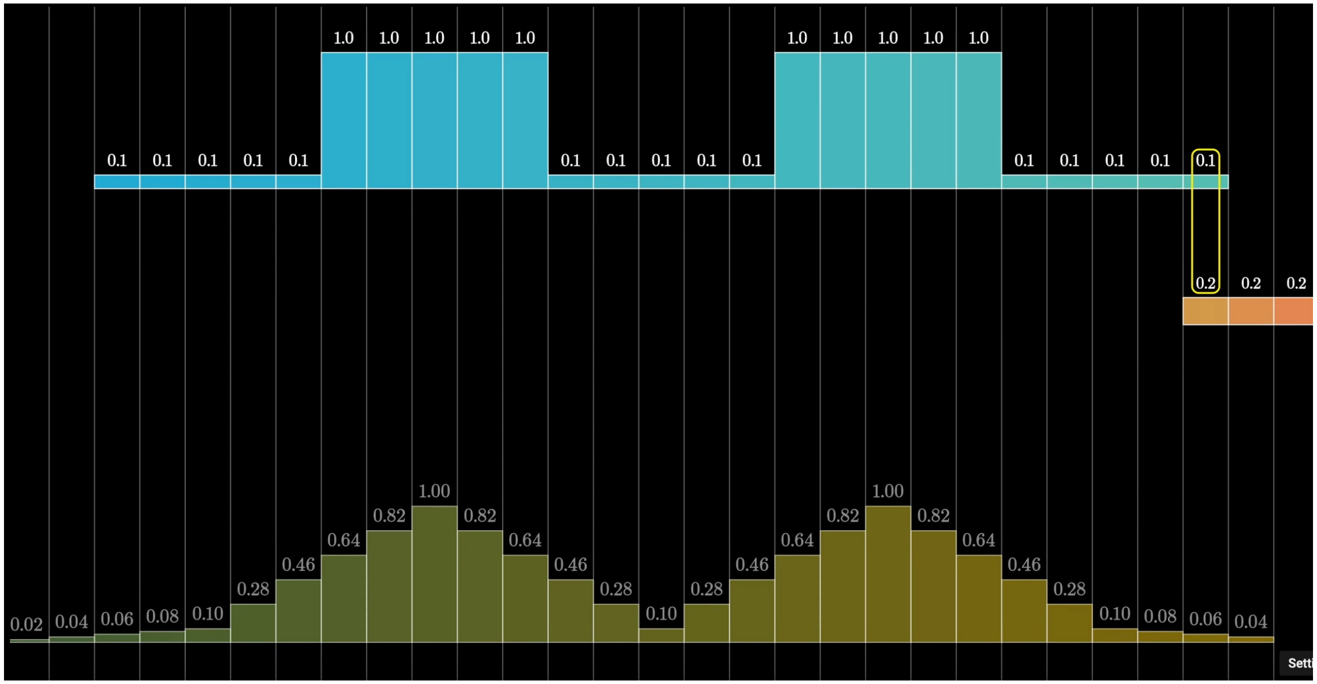
https://youtu.be/KuXjwB4LzSA

What kind of operation?



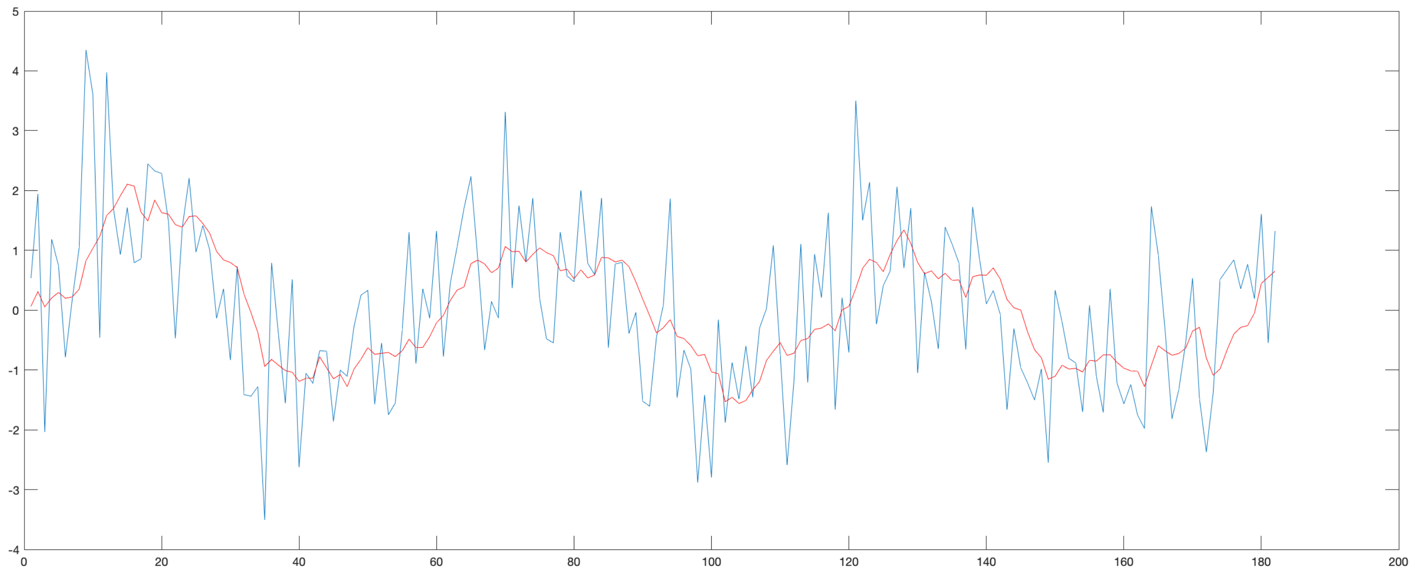
https://youtu.be/KuXjwB4LzSA

What kind of operation?



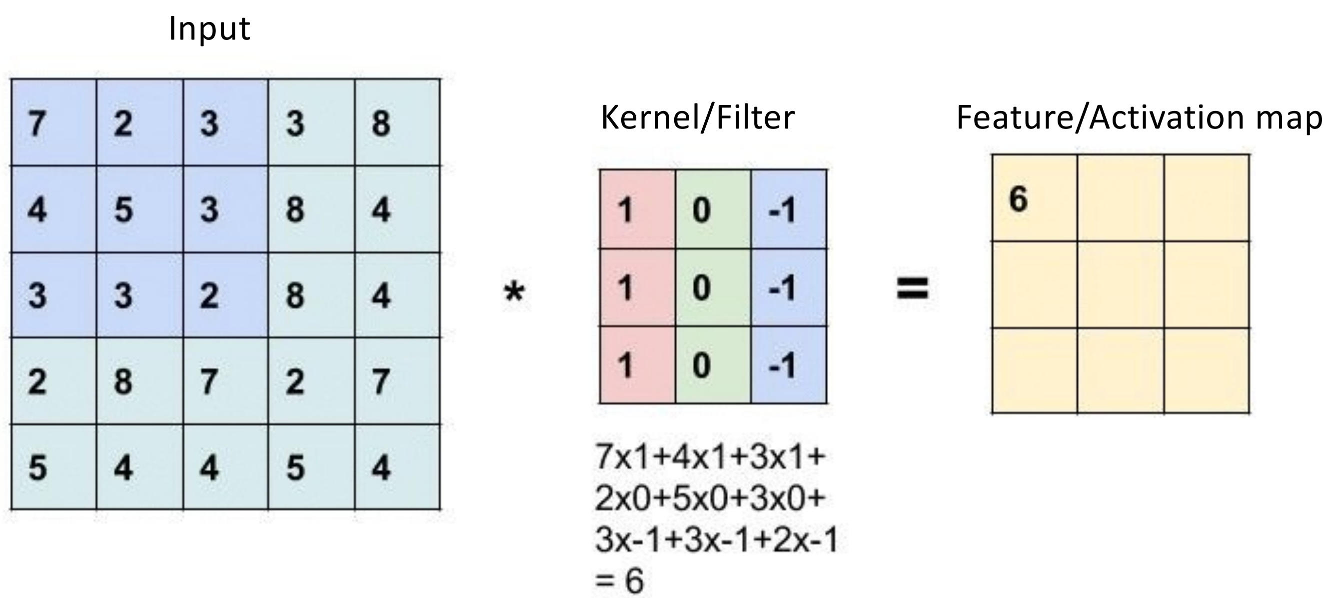
https://youtu.be/KuXjwB4LzSA

Moving average – 1D Convolution



filter = 1/8 * [1 1 1 1 1 1 1 1];

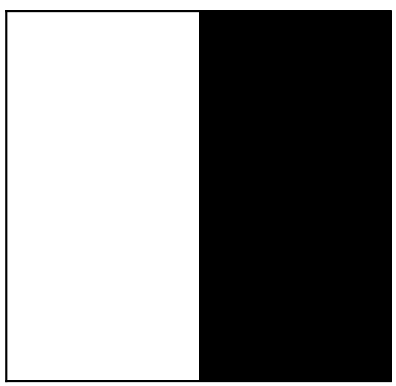
2D Convolution

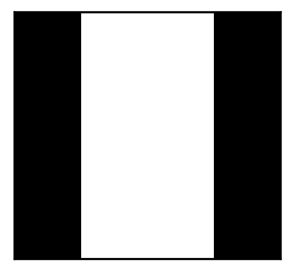


https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

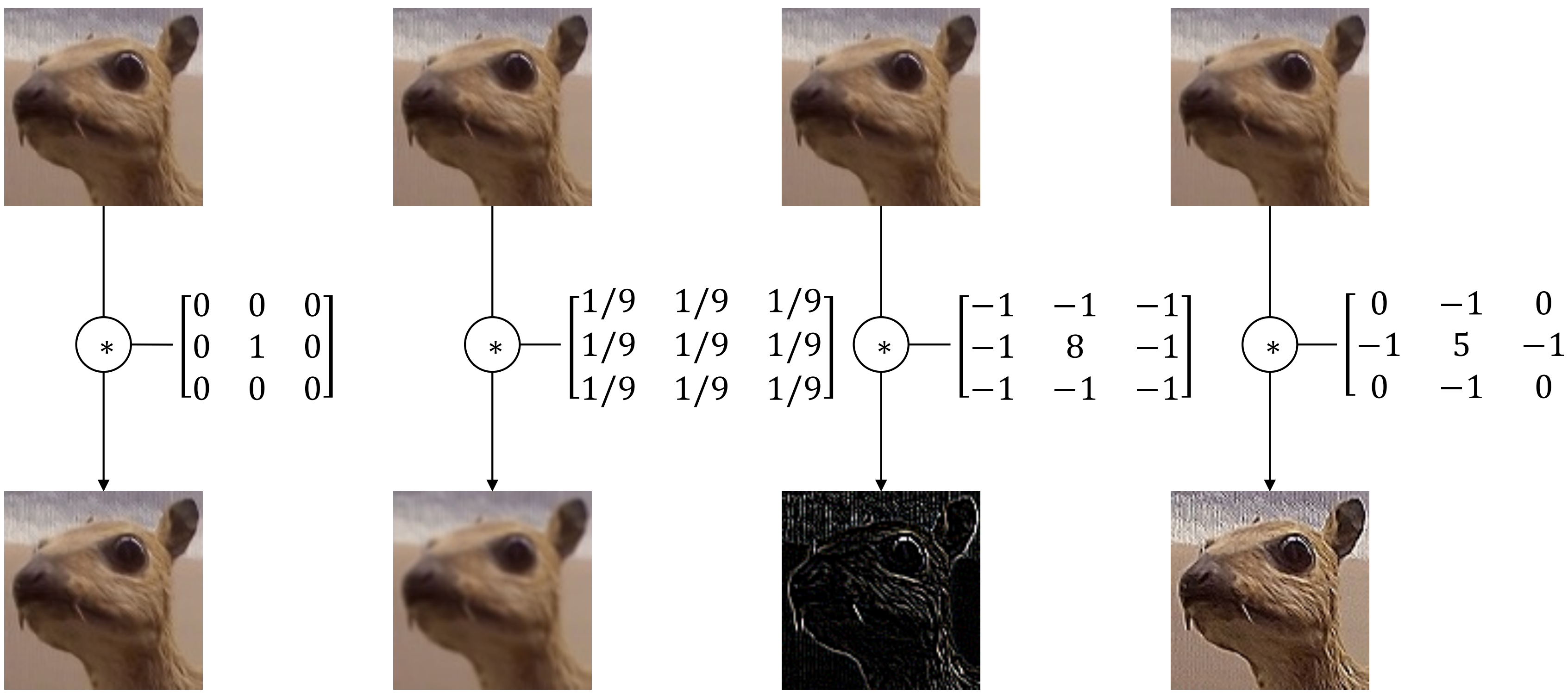
Edge detection filter

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}
*
\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}
=
\begin{bmatrix} 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \end{bmatrix}
$$





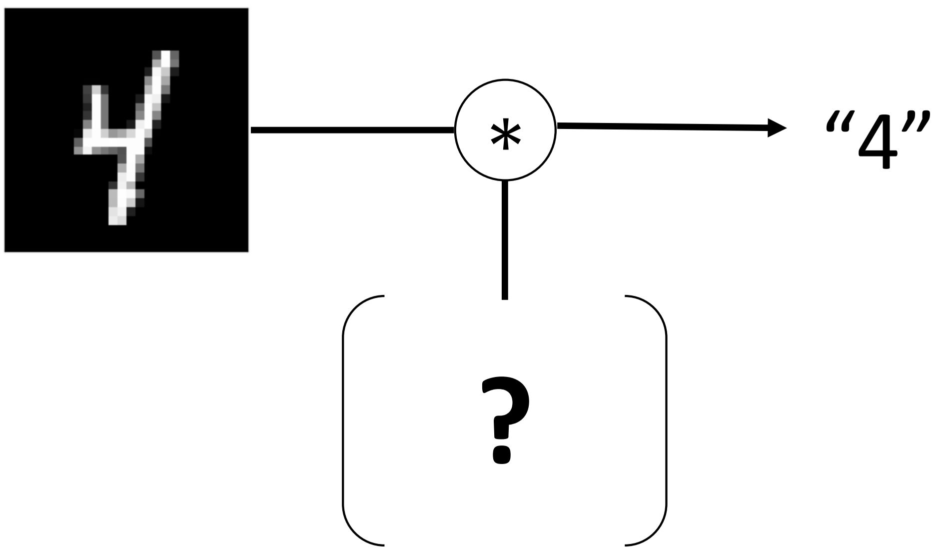
There are many types of filters



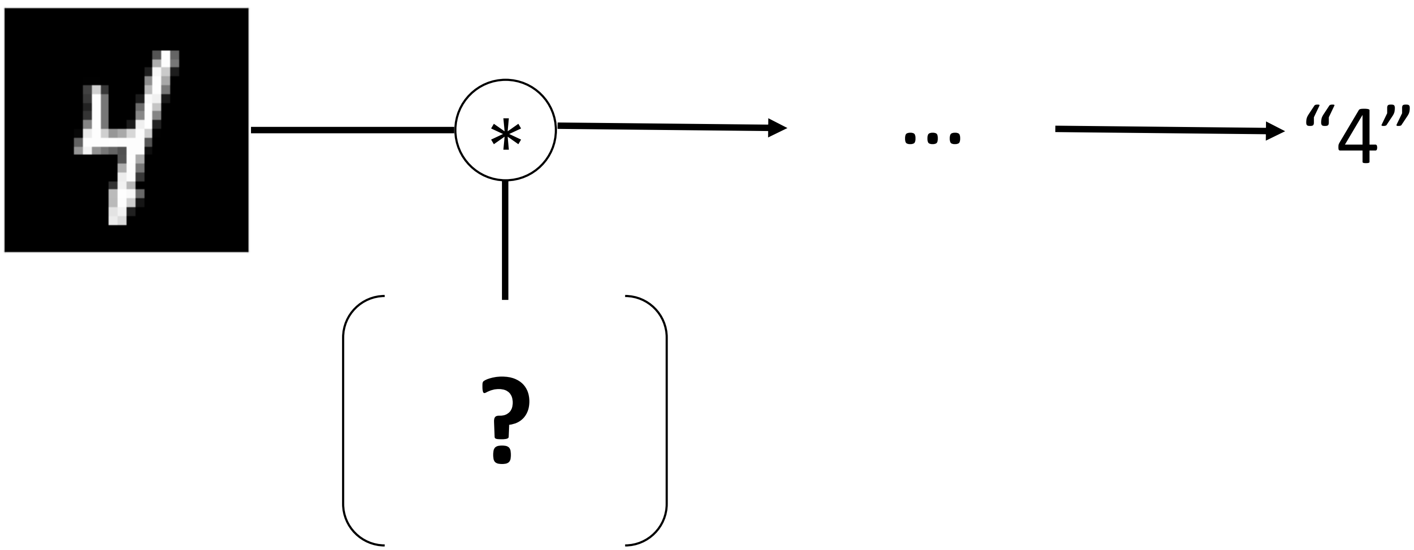
Identity Blur Edge detection Sharpen

https://en.wikipedia.org/wiki/Kernel_(image_processing)

Filter for digit detection?



Filter for digit detection?



Let machine learn

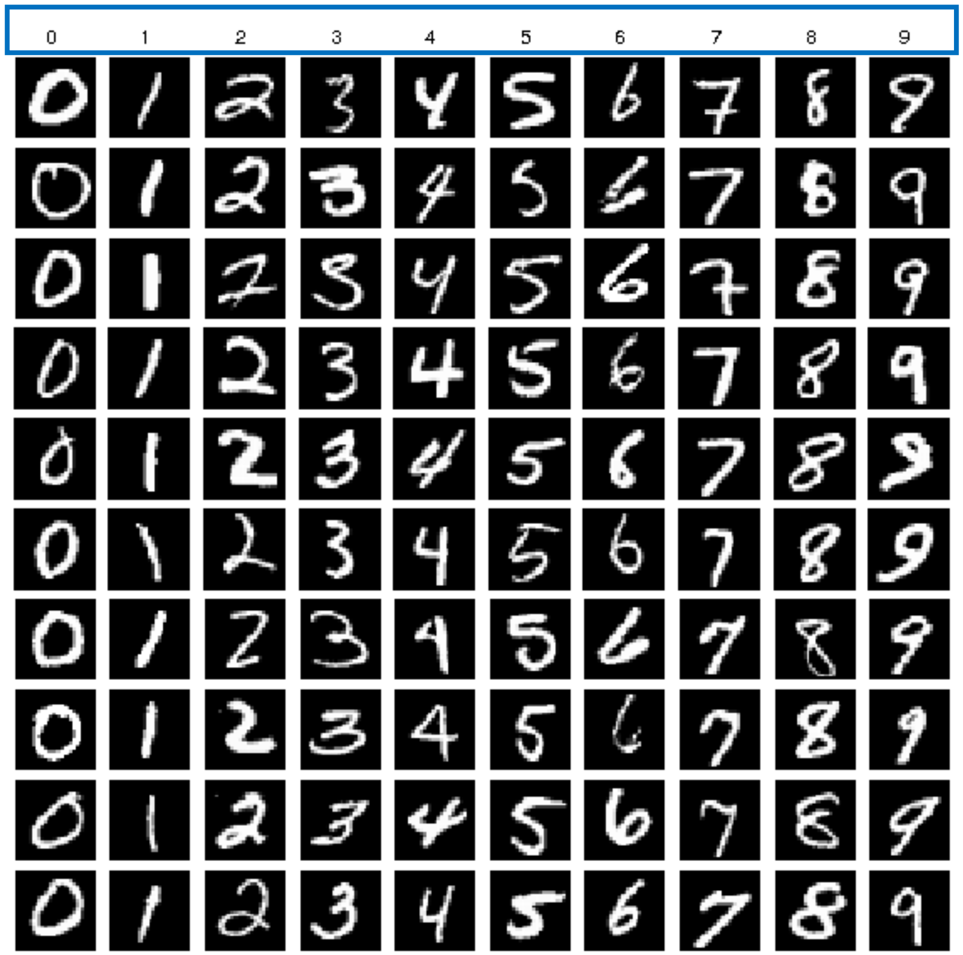
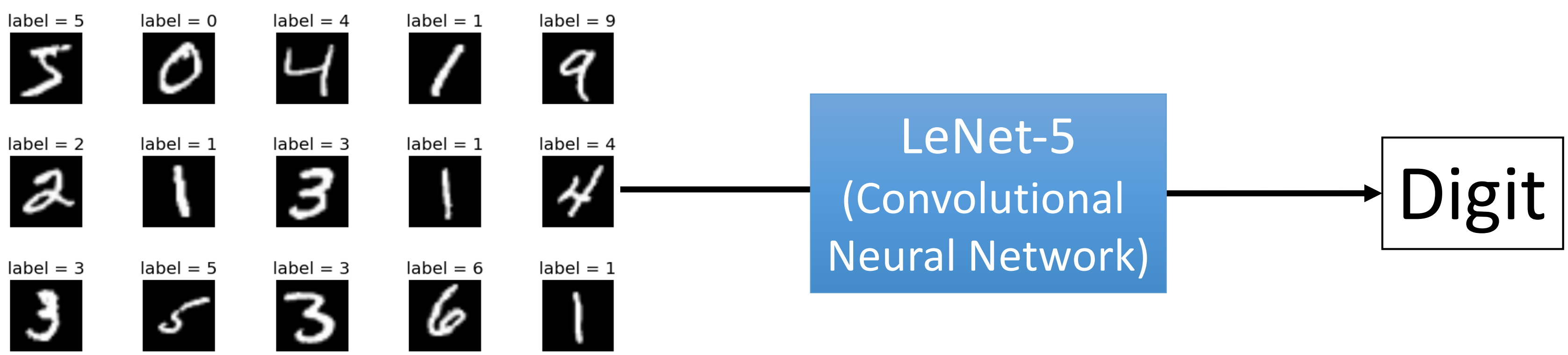


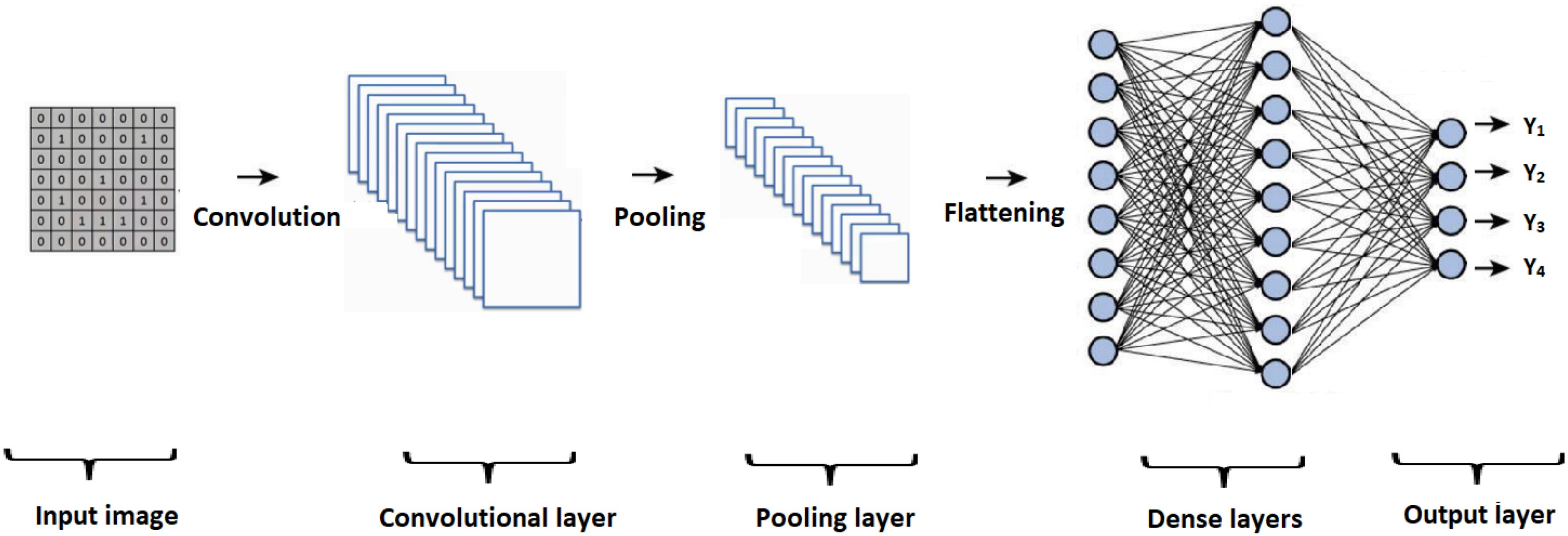
Image processing + Data + Neural network



MNIST dataset

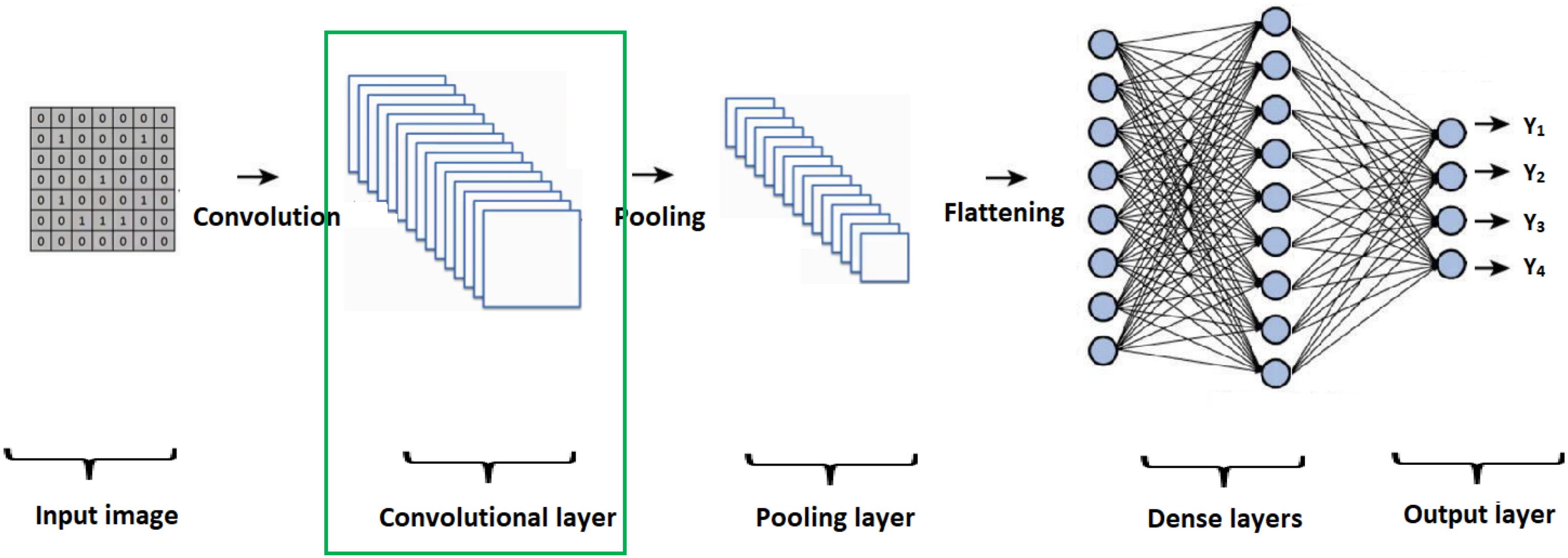
Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

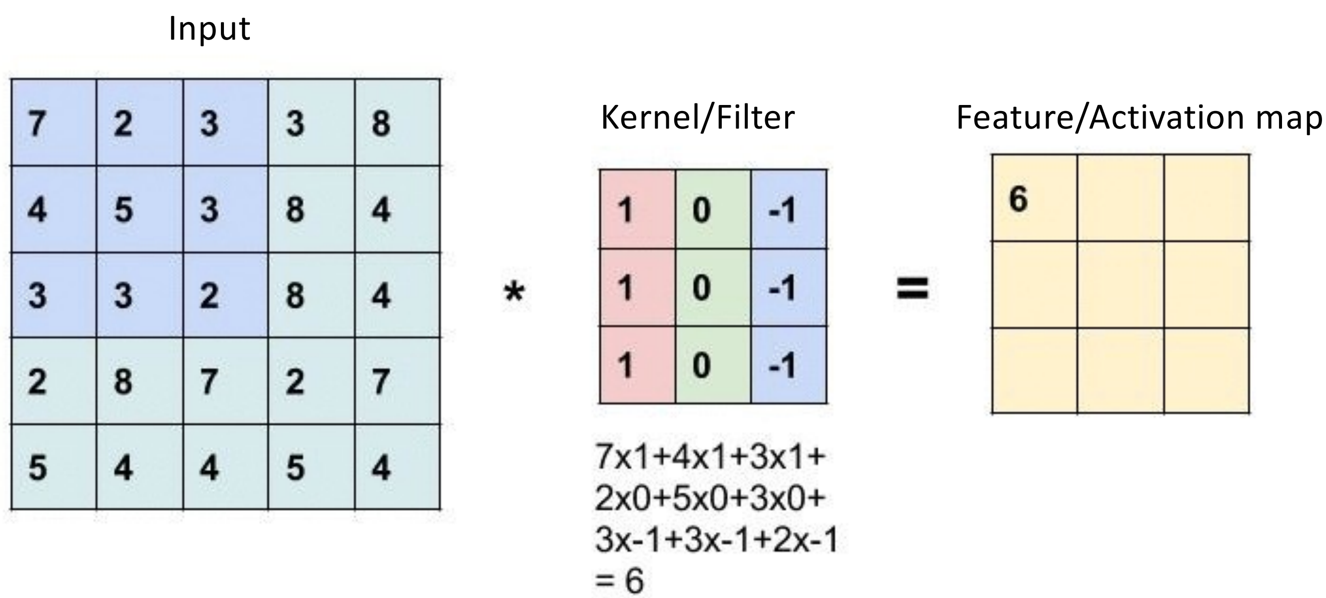
Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

Convolutional layer

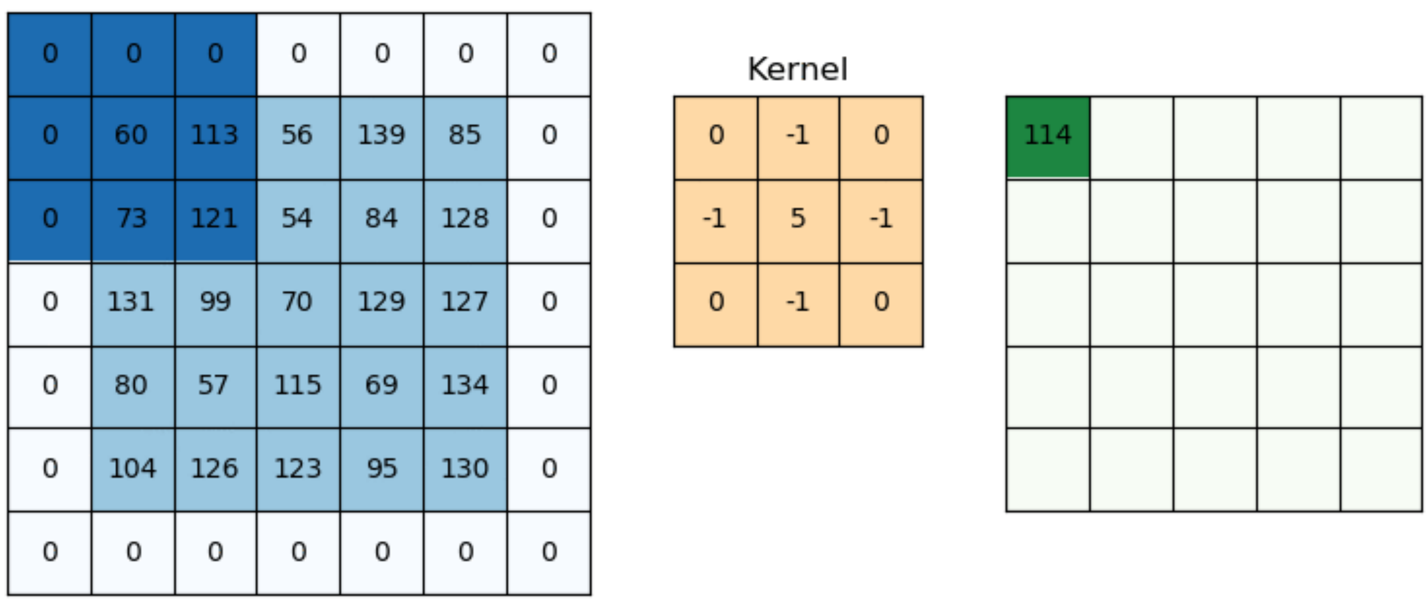
- Convolution operation



https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

Convolutional layer

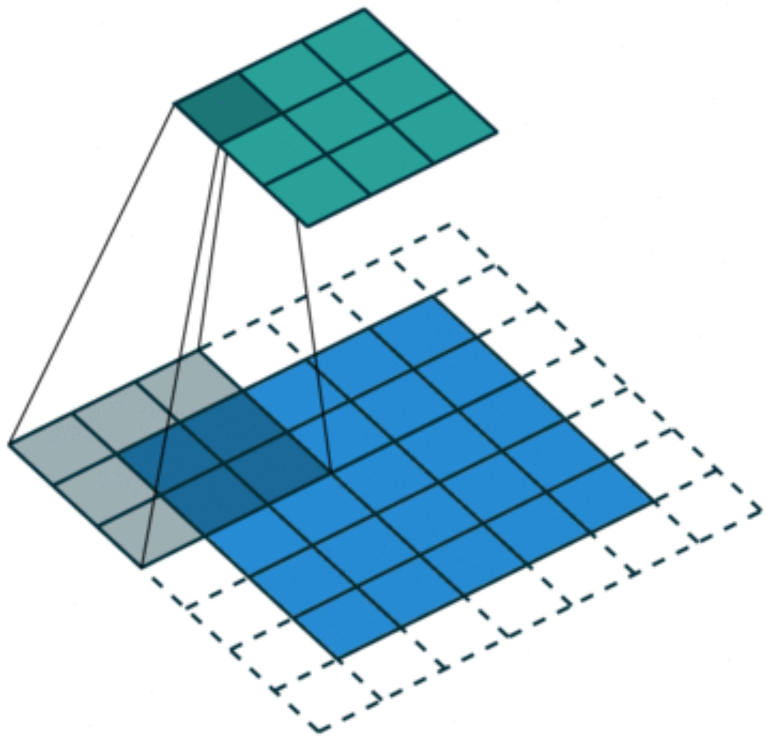
- Padding



https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

Convolutional layer

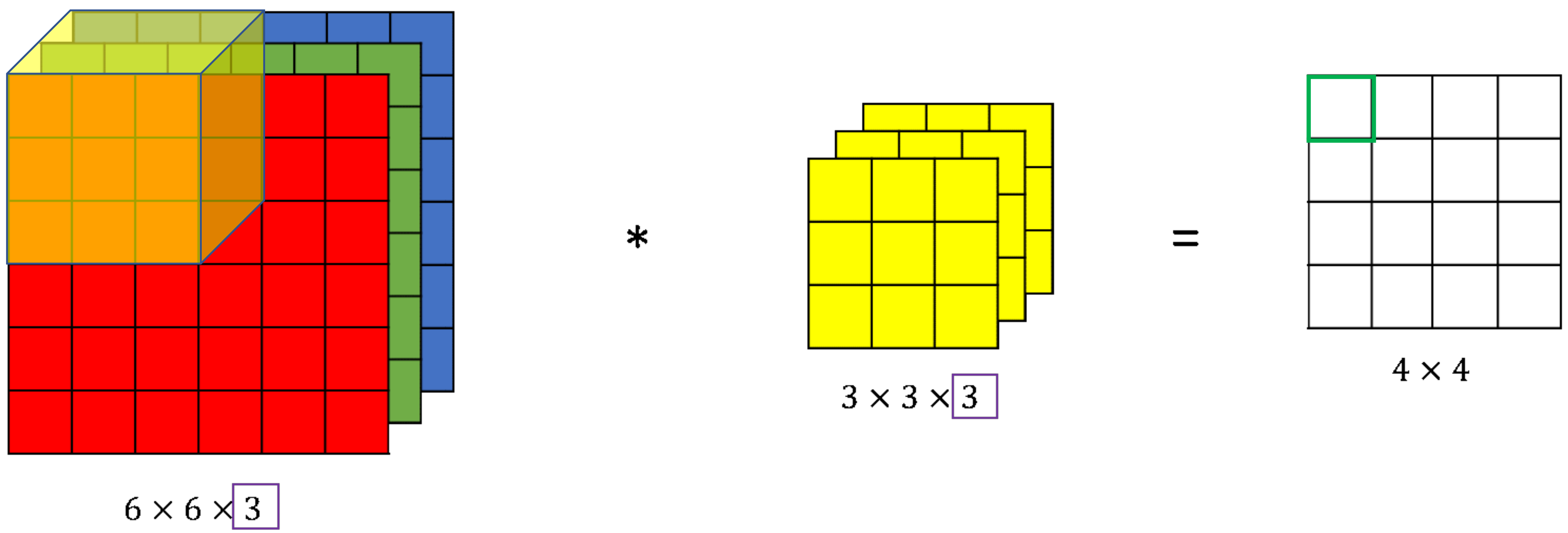
- Stride



Example: Padding = 1, Stride = 2

Convolutional layer

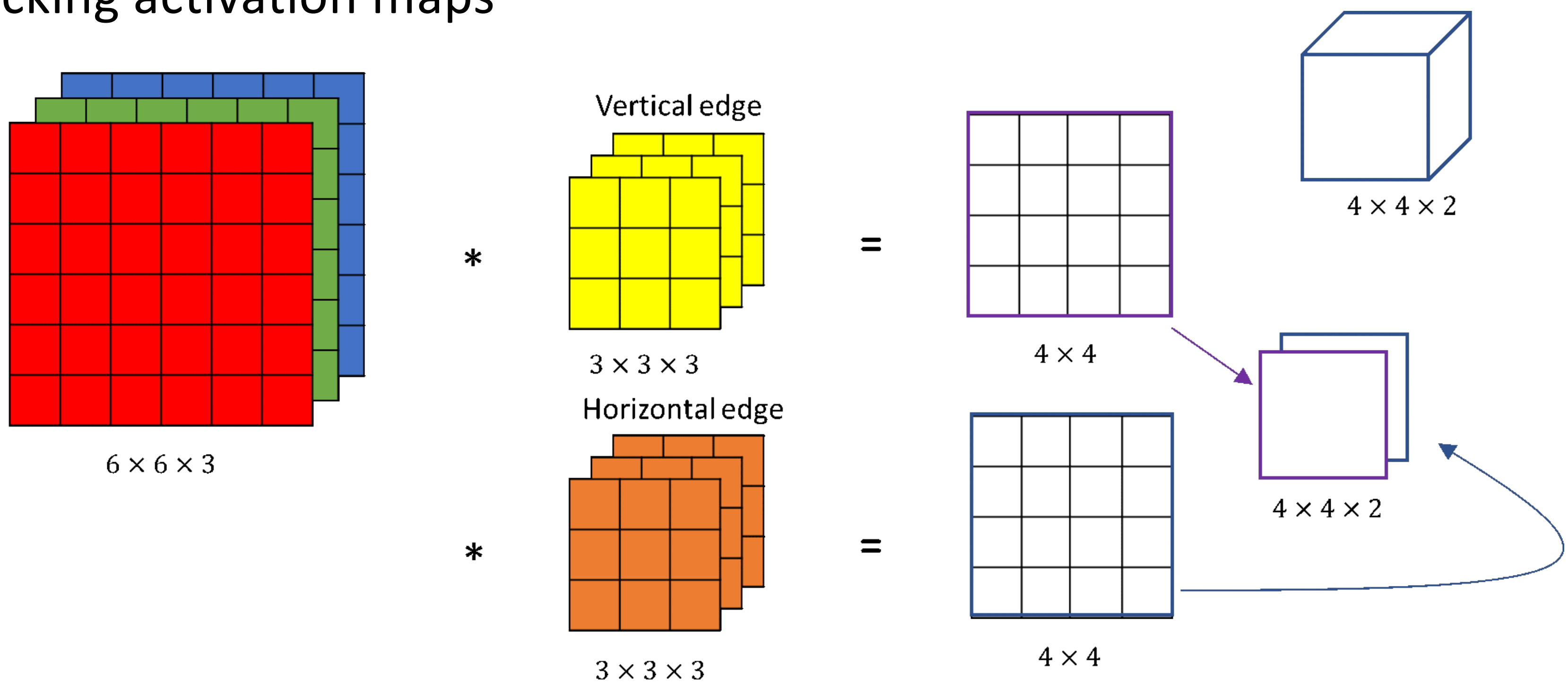
- Convolution over volume



Andrew Ng

Convolutional layer

- Stacking activation maps

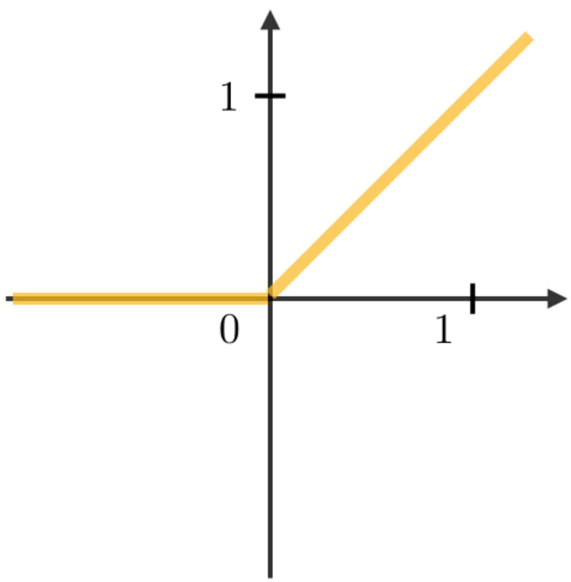


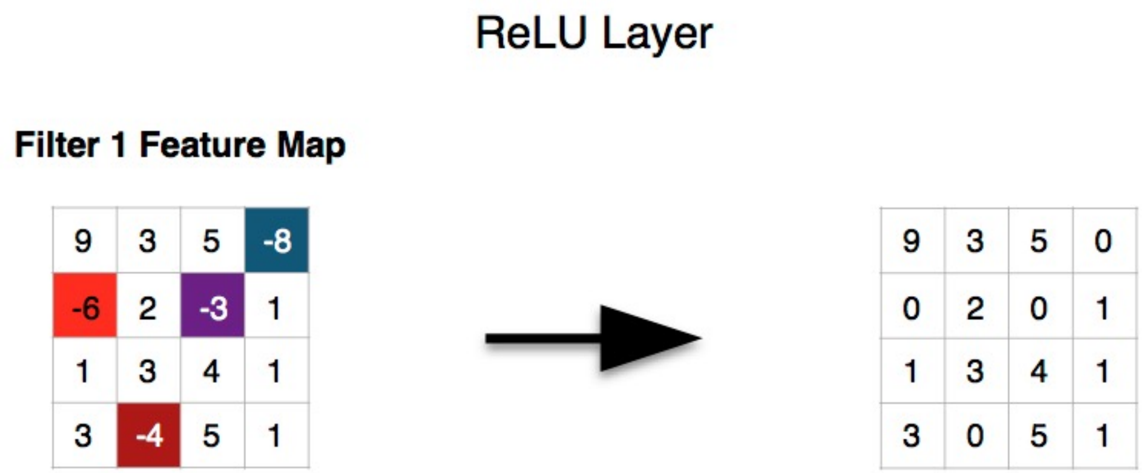
Andrew Ng

Activation functions

- Rectified Linear Unit (ReLU)

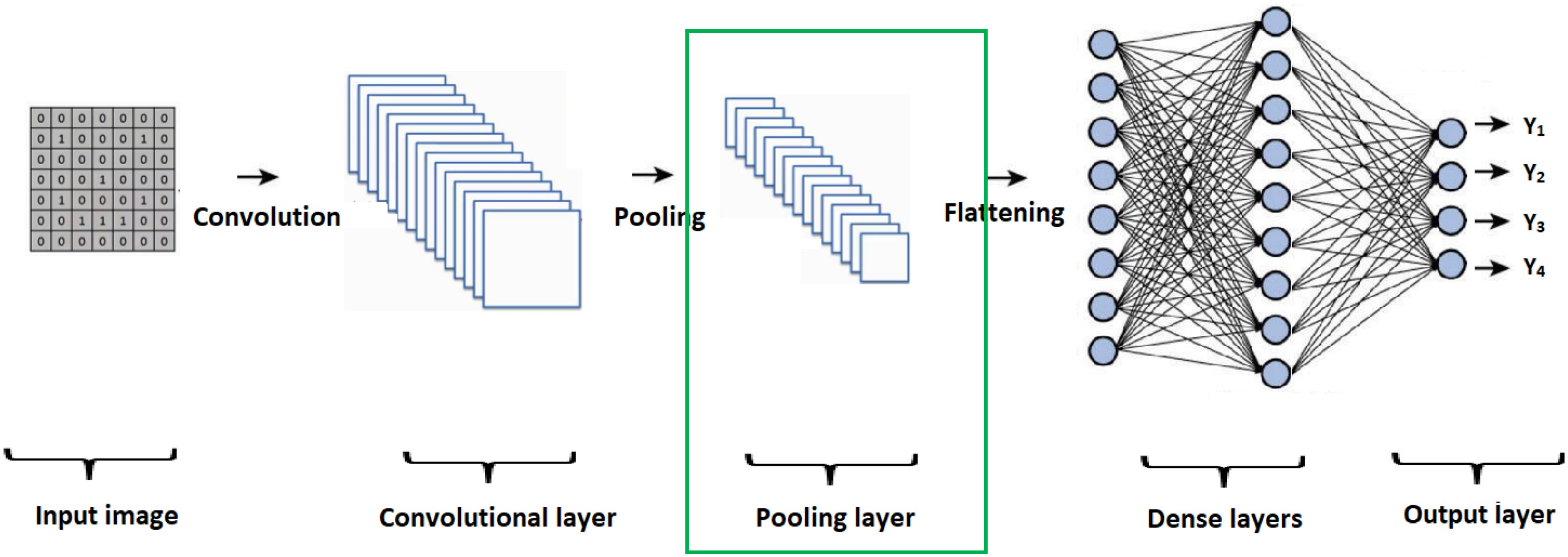
$$g(z) = \max(0, z)$$





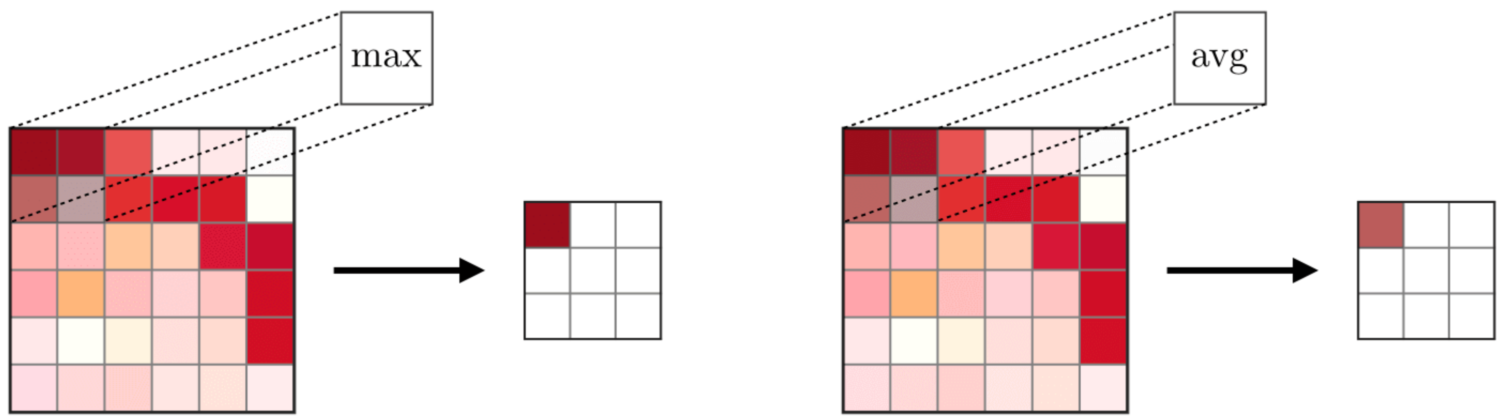
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

Convolutional Neural Network Architecture



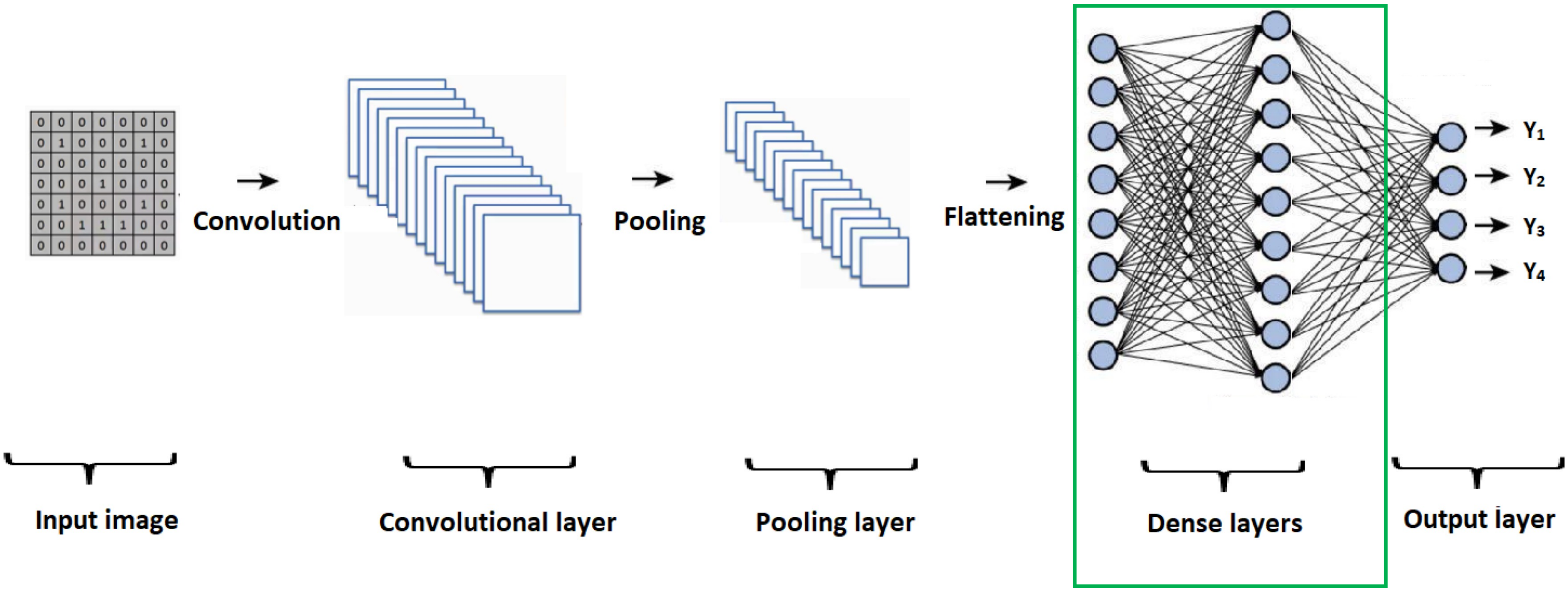
https://www.mdpi.com/2076-3417/10/4/1245/htm

Pooling layer



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

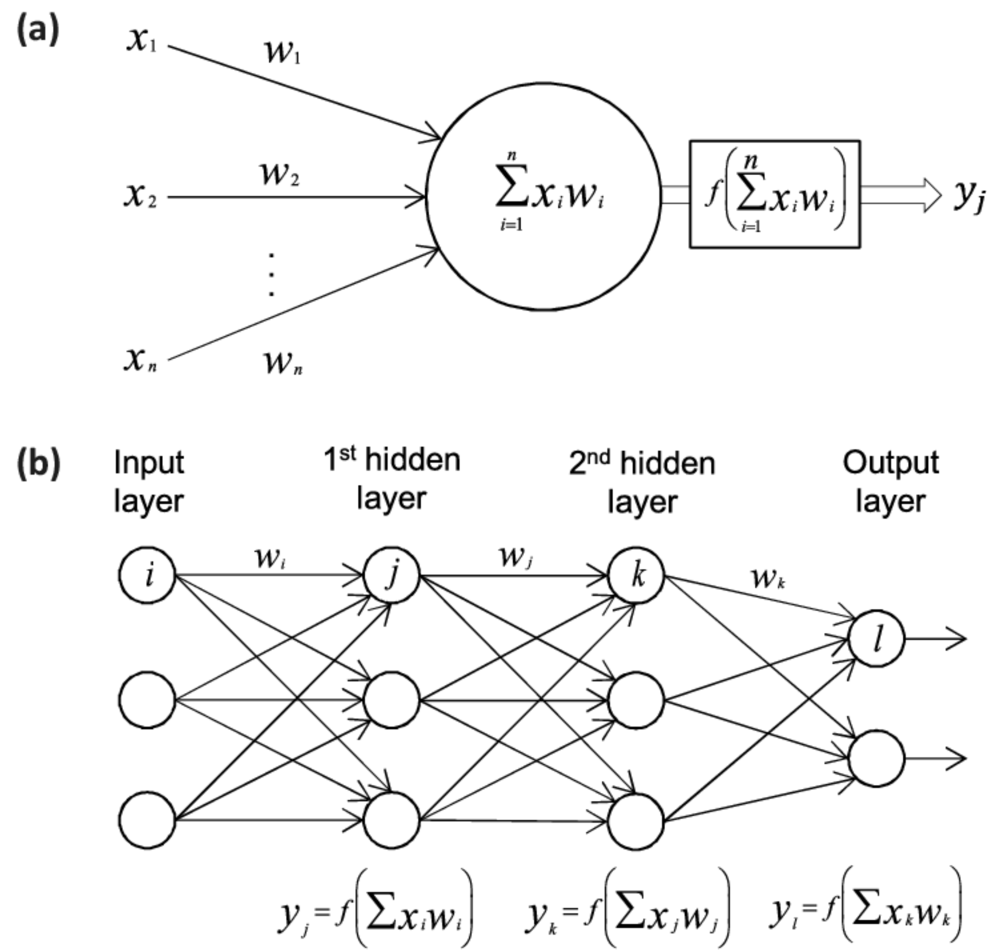
Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

Fully connected layer

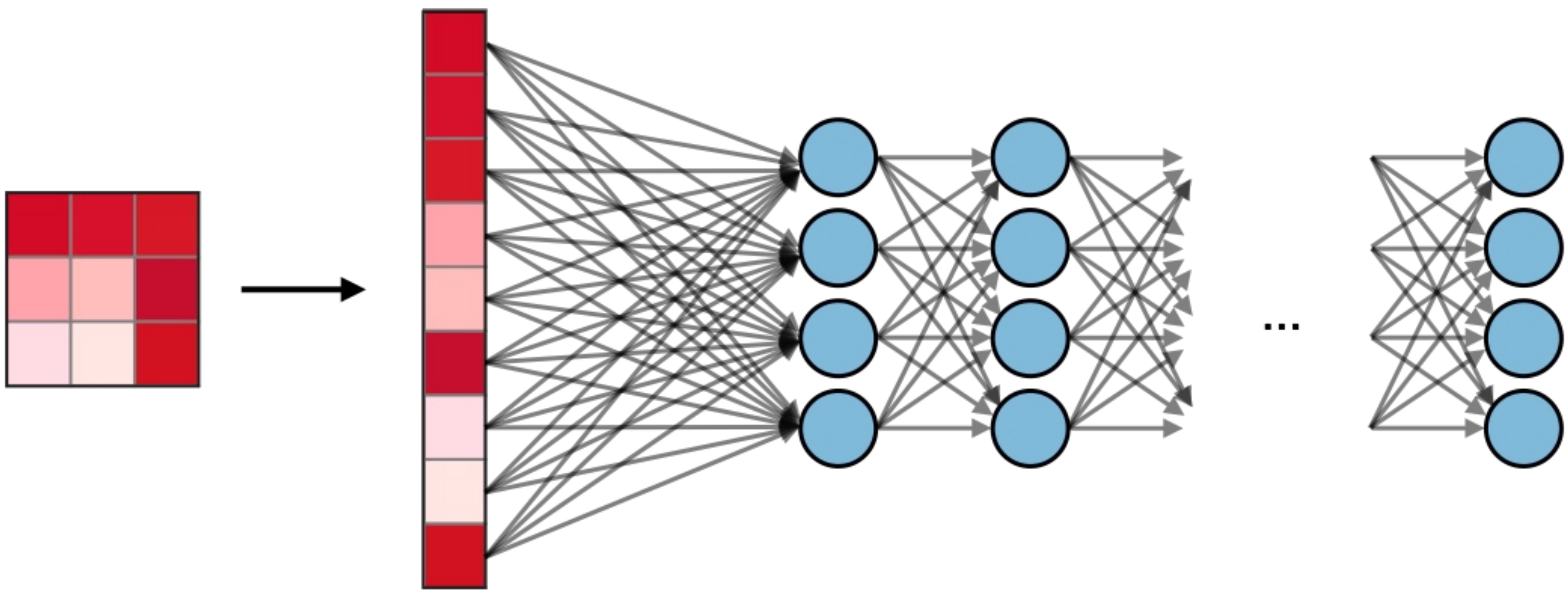
- Also known as dense layer
- Each input is connected to all hidden units



https://www.researchgate.net/figure/a-The-building-block-of-deep-neural-networks-artificial-neuron-or-node-Each-input-x_fig1_312205163

Fully connected layer

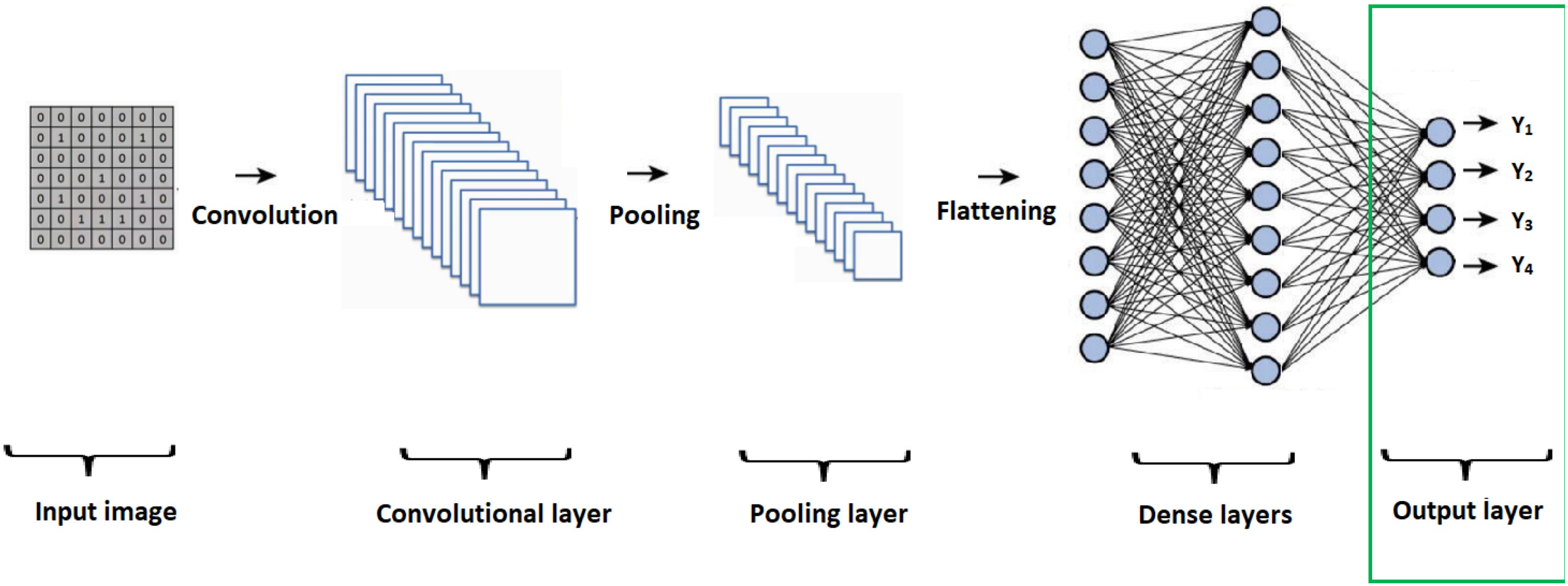
- Flattening



Flatten

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

Activation function for classification

- Softmax

$$\boxed{p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}} \quad \text{where} \quad \boxed{p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}}$$

Loss function

- Cross-entropy loss

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

y is a one-hot vector ([0,0,1,....,0])

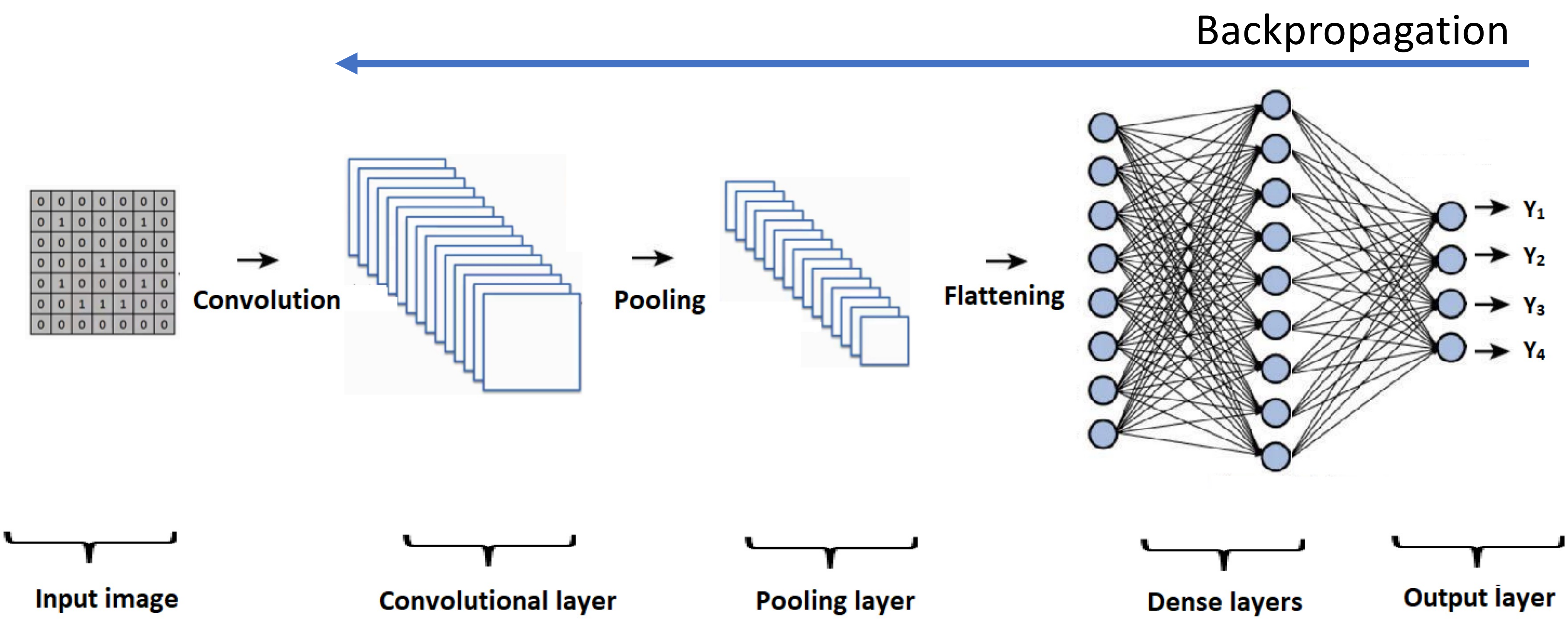
Gradient descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} L$$

Parameters

Learning rate

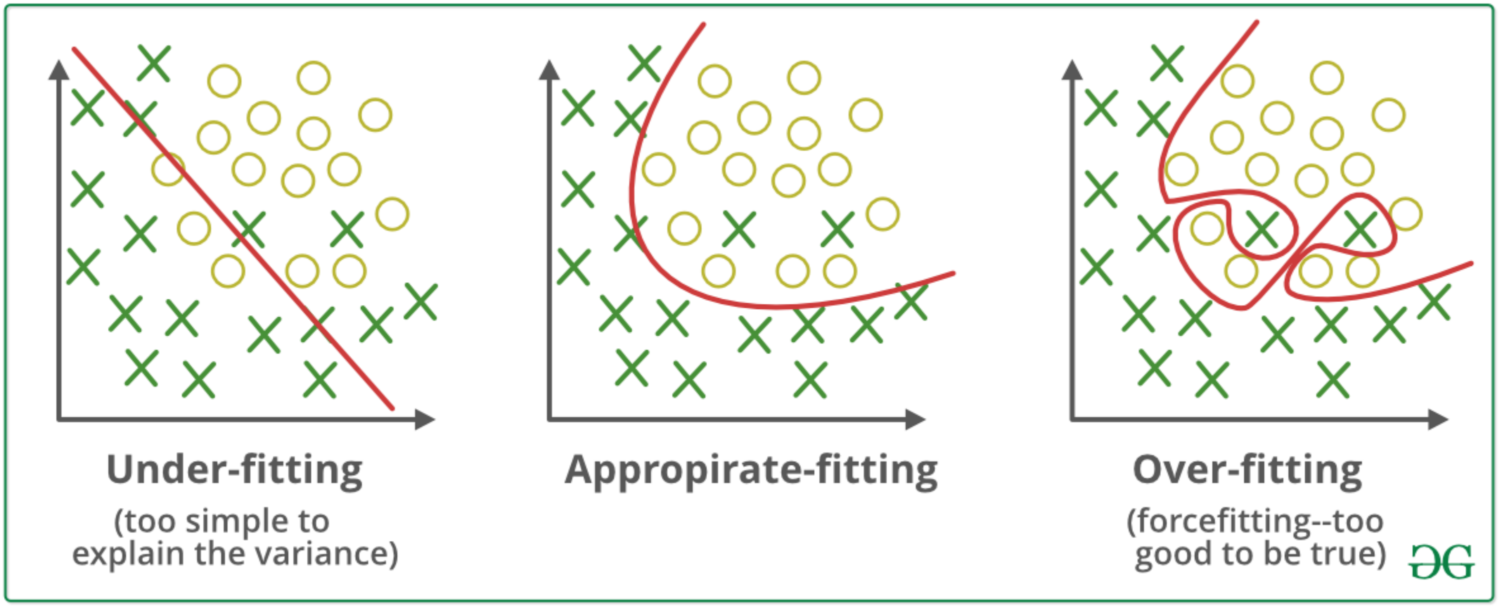
Training the network



https://www.mdpi.com/2076-3417/10/4/1245/htm

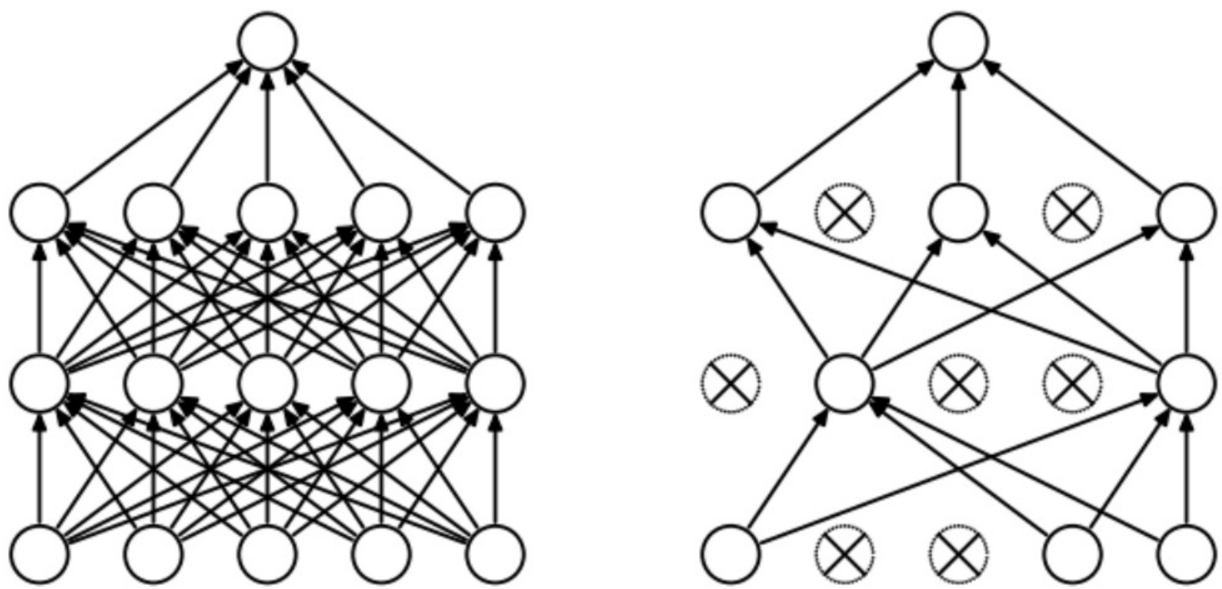
Dropout

- Overfitting



https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

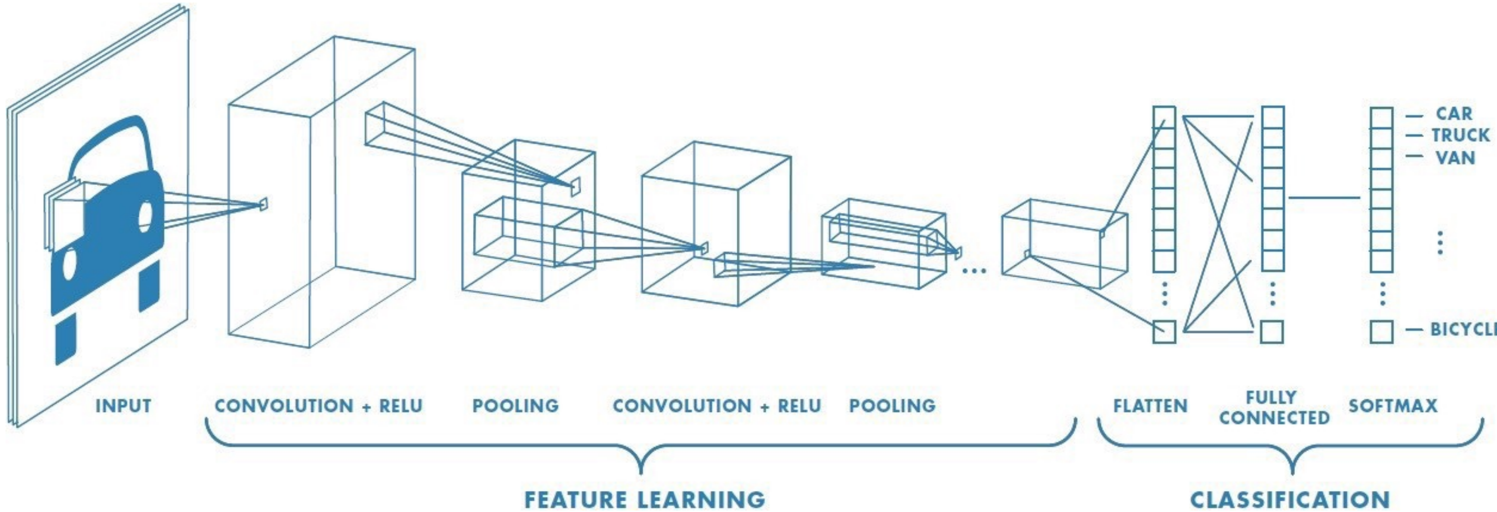
Dropout

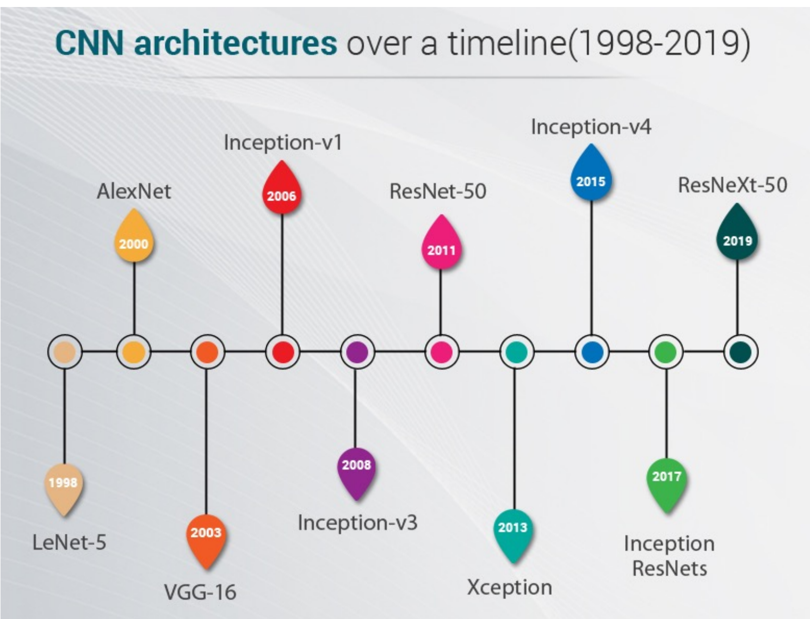


(a) Standard Neural Net

(b) After applying dropout.

https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5





IMDb

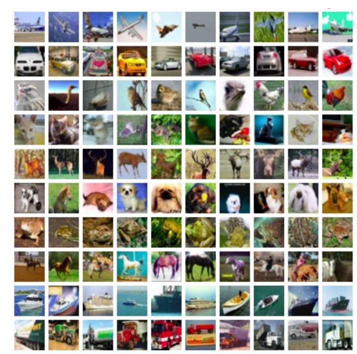
460,723 images

Wikipedia

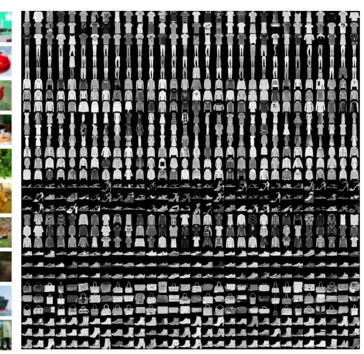
62,328 images

COCO 2020 Panoptic Segmentation Task

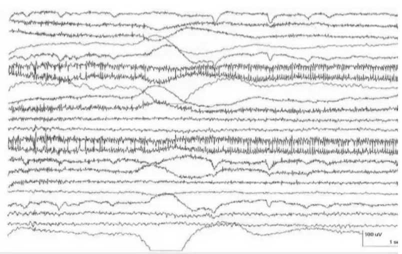
CIFAR-10



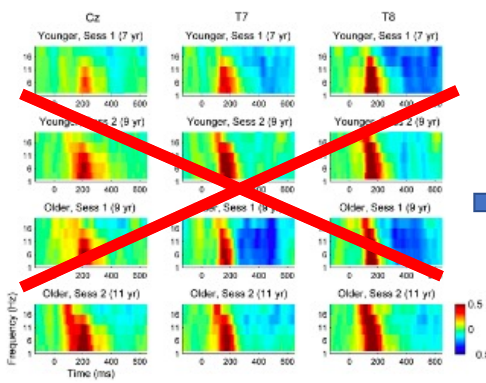
Fashion MNIST

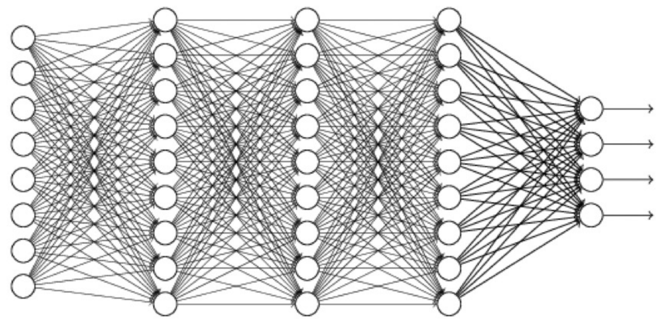


Applying to EEG



ERSP





Emotion/Traits

Raw EEG vs. Frequency-domain

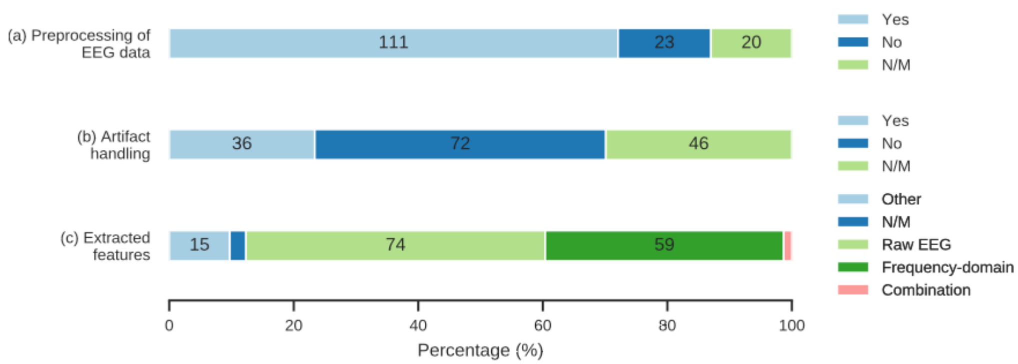
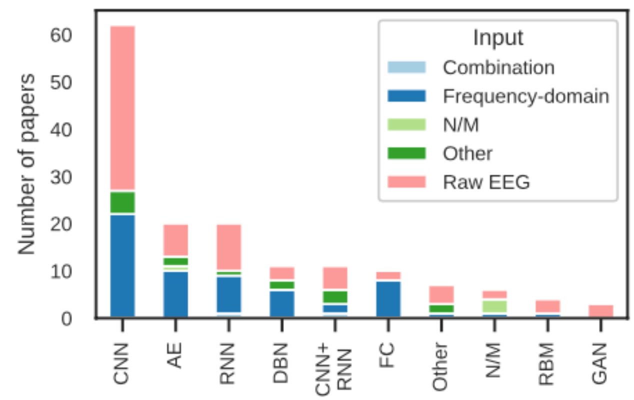


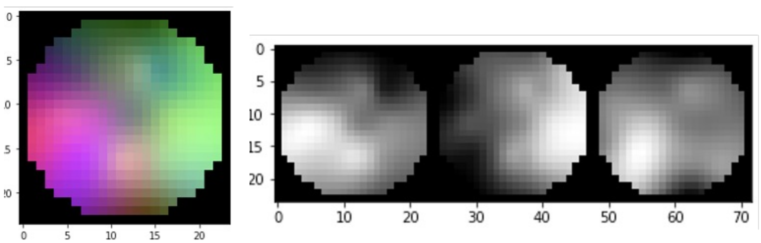
Figure 9. EEG processing choices. (a) Number of studies that used preprocessing steps, such as filtering, (b) number of studies that included, rejected or corrected artifacts in their data and (c) types of features that were used as input to the proposed models.



Distribution of input type according to the architecture category.

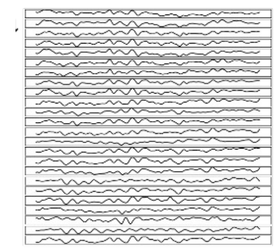
Roy Y. et al (2019)

Our approach

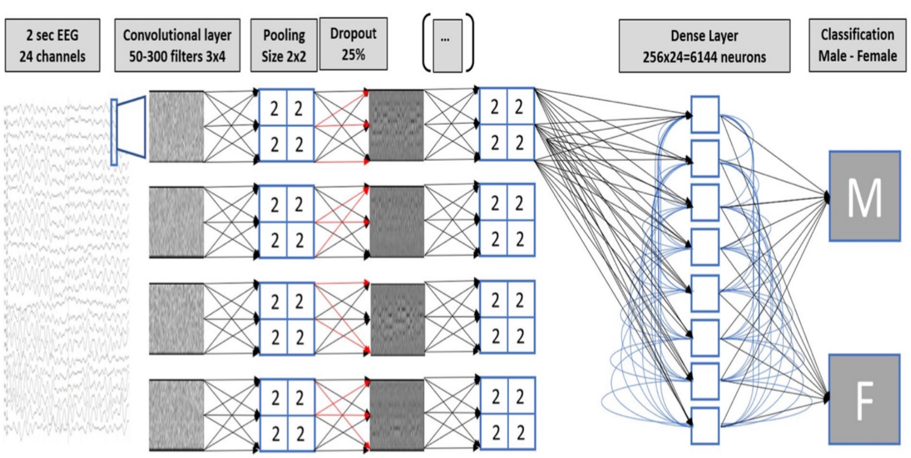


Spectral features

vs.

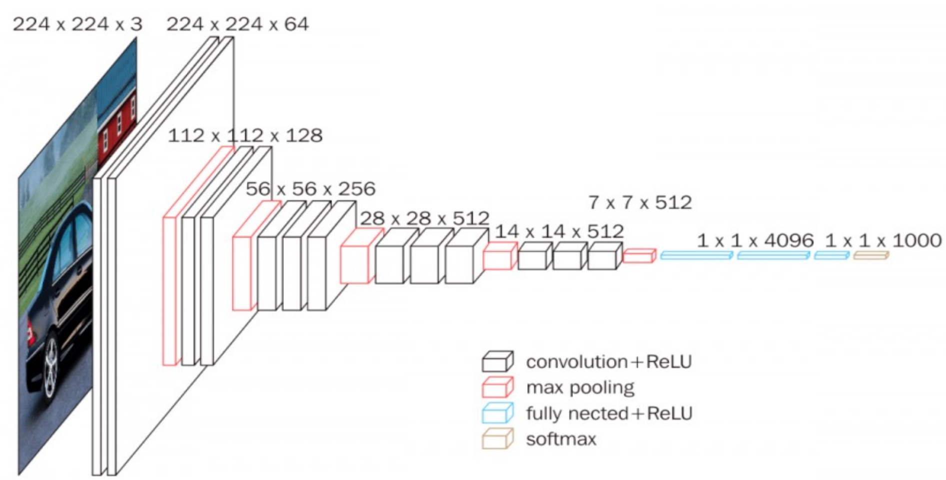


Raw EEG



van Putten et al. (2018)

vs.



Simonyan, K. and Zisserman A. (2014)

Data Data Data Data Data Data Data Data

CHILD MIND® INSTITUTE

Michael Milham, PI

- 128 channels EEG
- ~3,000 EEG datasets (planed 10,000)
- Tasks involving emotions (The Gift movie)
- Rest (eyes open and eyes closed)

Data

- 2224 subjects:
  - 787 female (35%)
  - Ages: min 5, max 22, mean 10
- Pre-processing following the paper:
  - Remove baseline
  - Filter 0.25-25Hz
  - Resample 128Hz
  - Re-reference to average mastoids
  - Epoching: eye-closed, 3 40-second blocks. Ignored first and last 3 seconds of each block
  - clean_rawdata – ASR (our)
  - Sub-select 24 channels
    - Fp1, Fp2, F7, F3, Fz, F4, F8, FC3, FCz, FC4, T3, C3, C4, T4, CP3, CPz, CP4, T5, P3, Pz, P4, T6, O1, Cz
  - Segment 2-second non-overlapping windows
    - → ~ 81 samples per subject

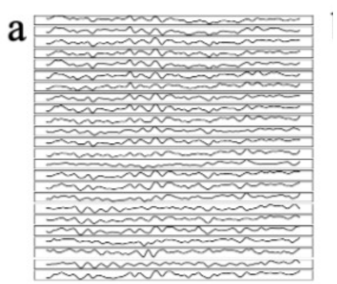
Data

- Sub-select 1574 participants (50% female) 24-channels
- 2-second extracted epochs eyes open and eyes closed
- 5 predictors: sex, handedness, eyes open/closed, age, segment count

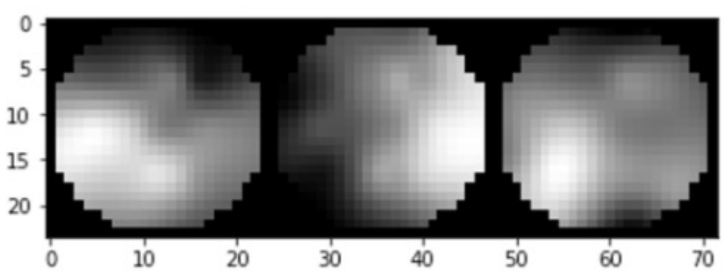
Categorical

Continuous

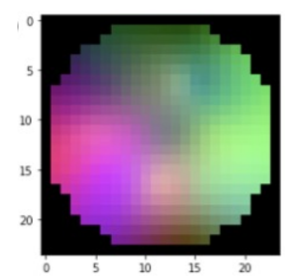
2-sec



Achromatic image

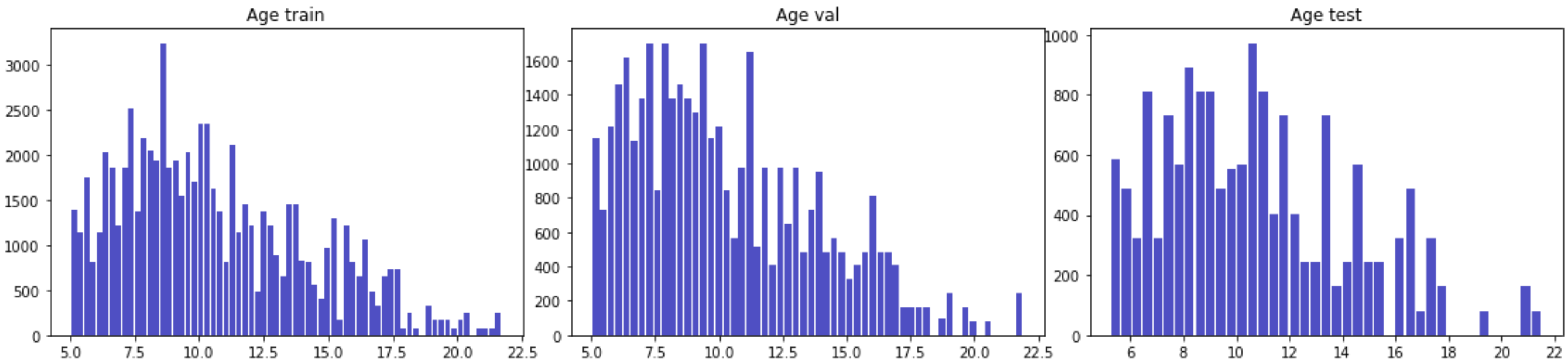


Chromatic image

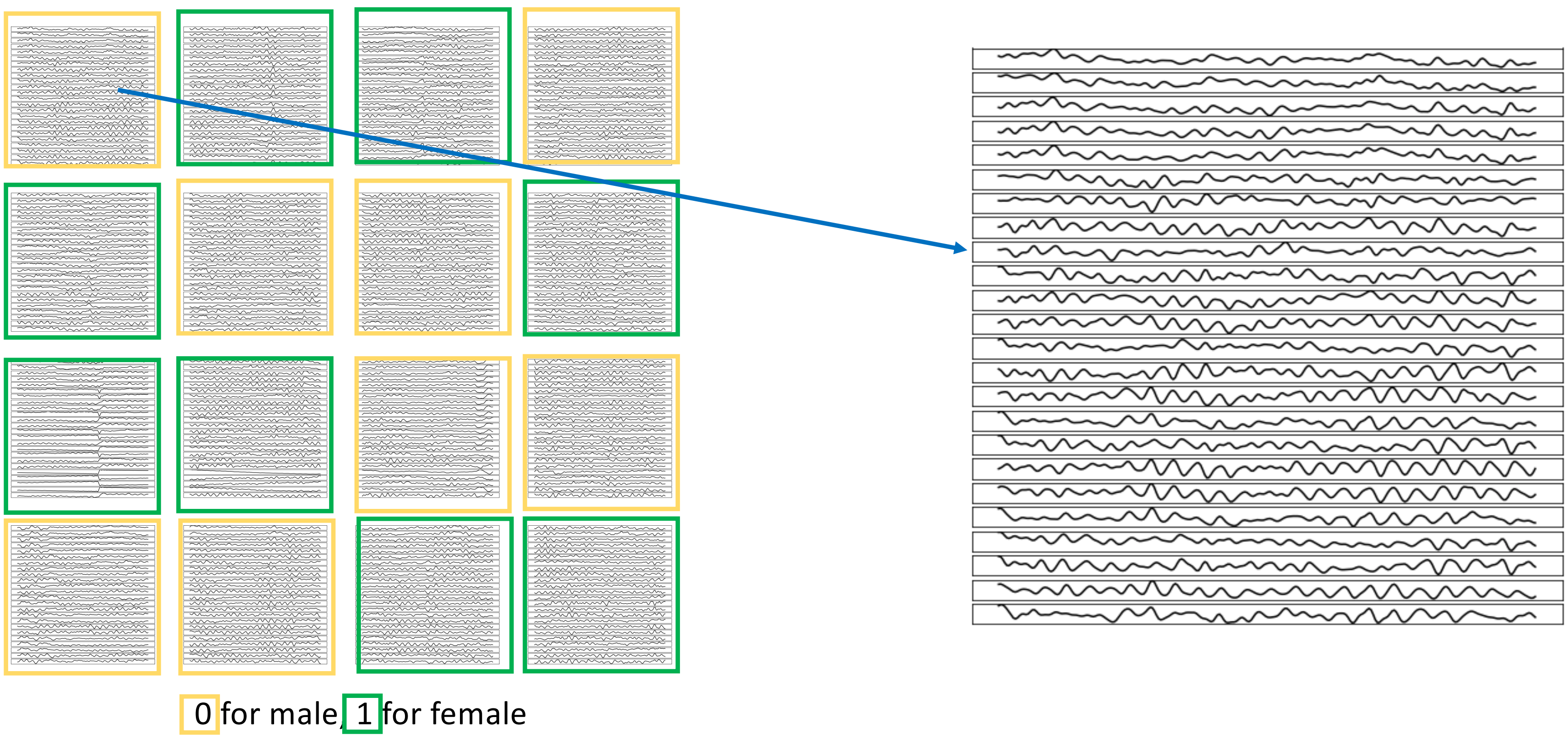


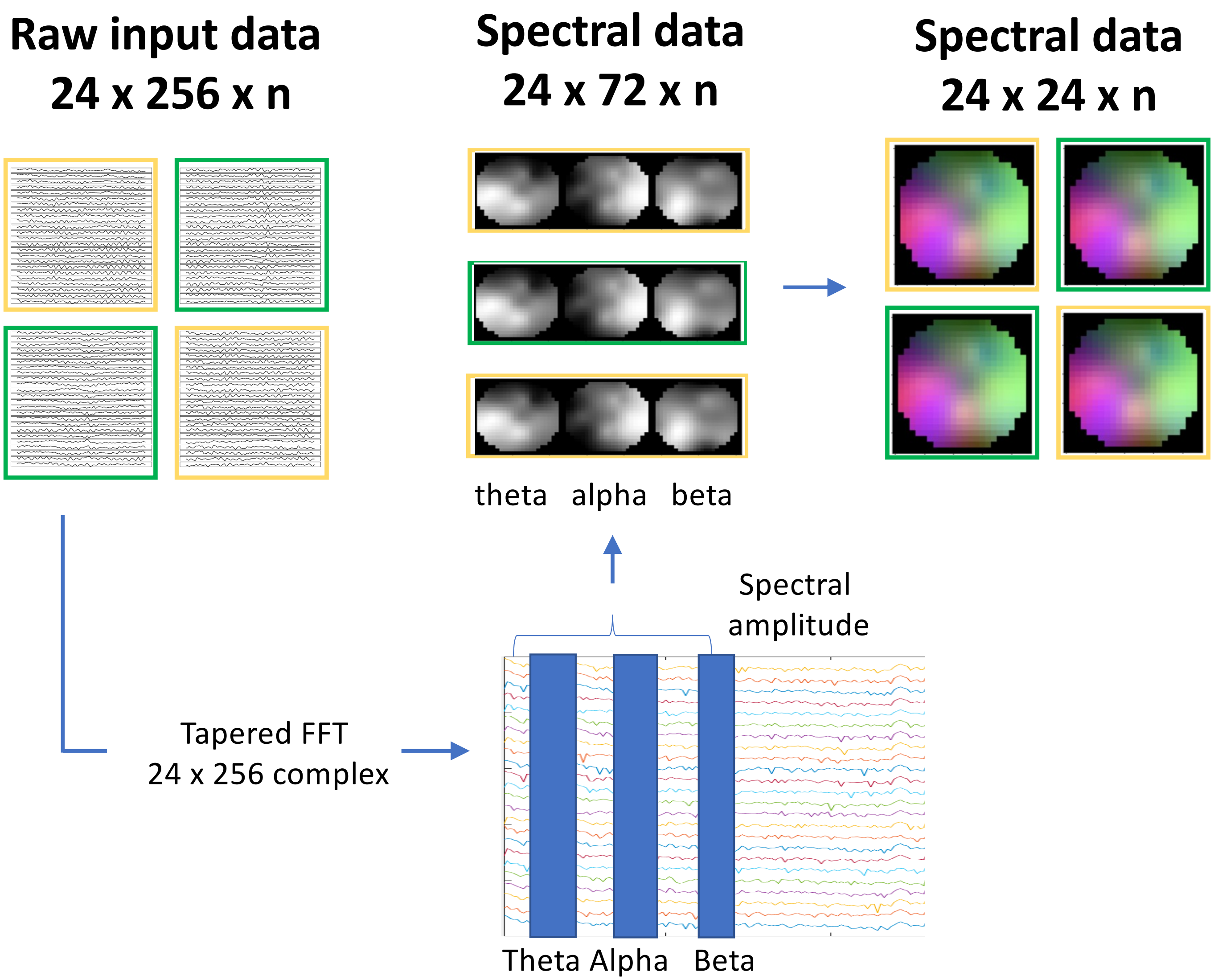
Data

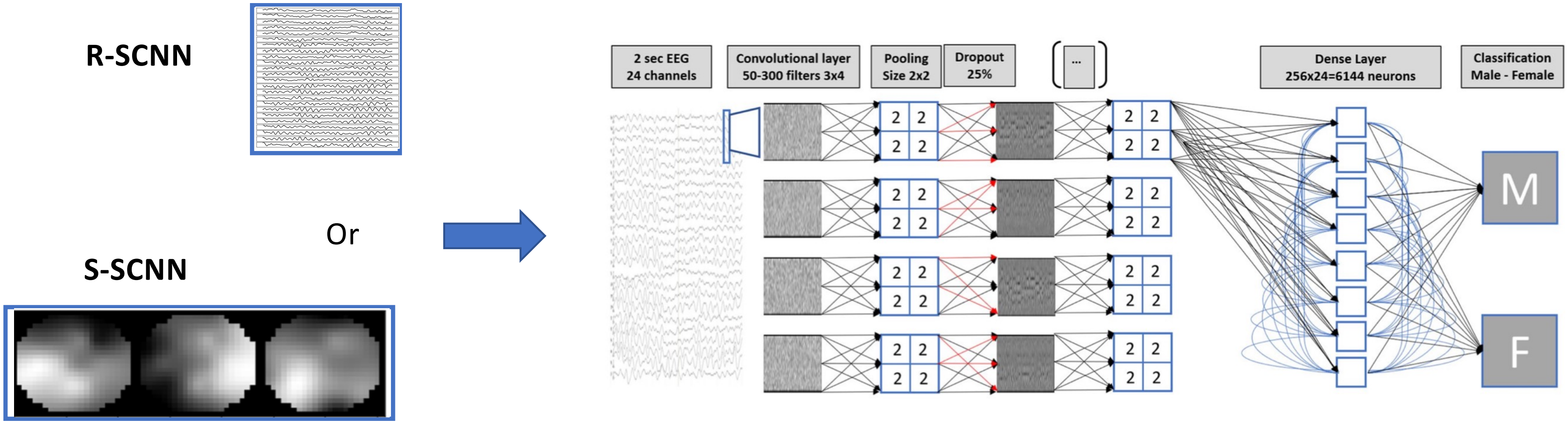
- 10-30-60 split
  - 885 subjects for training -> 71,381 samples
  - 492 subjects for validation -> 39,868 samples
  - 197 subjects for testing -> 15,925 samples



Input data 24 x 256 x n



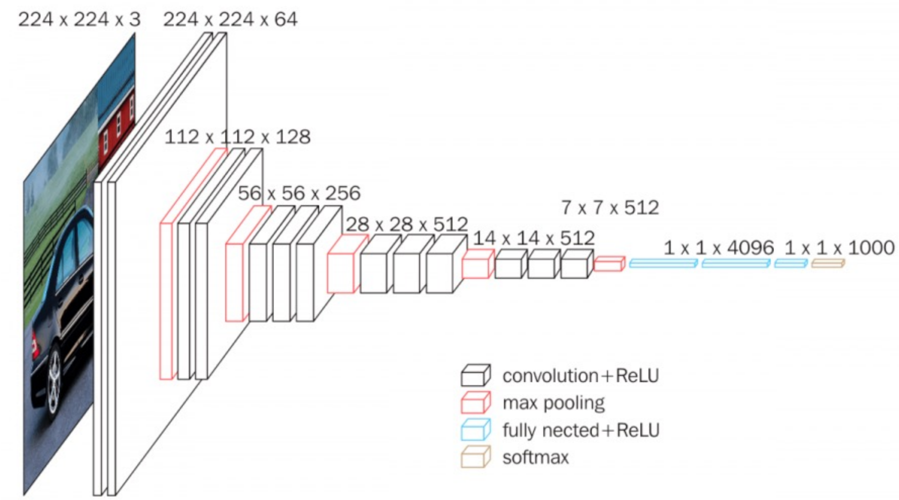

Raw input data
24 x 256 x n
Spectral data
24 x 72 x n
Spectral data
24 x 24 x n
theta alpha beta
Spectral amplitude
Tapered FFT
24 x 256 complex
Theta Alpha Beta



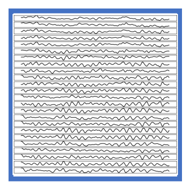
Layer	Filter size	# of filters/hidden units
Convolutional	3x3	100
MaxPooling Dropout (25%)		
Convolutional	3x3	100
MaxPooling Dropout (25%)		
Convolutional	2x3	300
MaxPooling Dropout (25%)		
Convolutional†	1x7	300
MaxPooling* Dropout (25%)		
Convolutional†	1x3	100
Convolutional†	1x3	100
Fully connected		6144
Fully connected		2
Softmax		

Modified VGG16

- Changed input size
- Keep the same scaling between layers
- Number of convolutions divided by 8 for each l
- Dropped layers 19 to 32
- Change number of output classes to 2
- Retrain the whole network

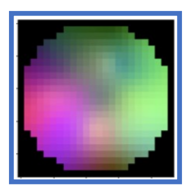


R-VGG



Or

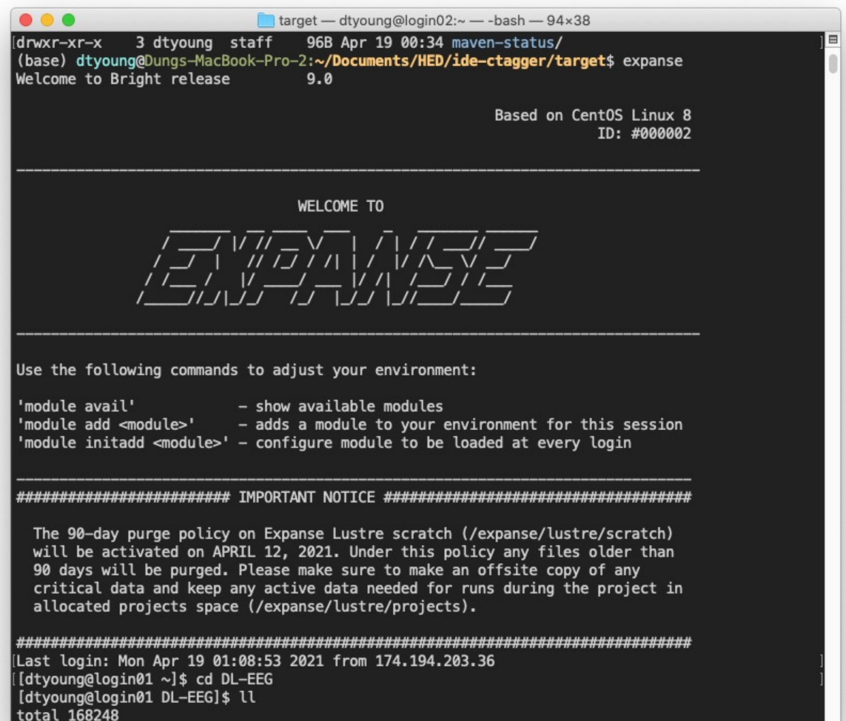
S-VGG



Layer	Filter size	# of filters/hidden units
Convolutional†	3x3	16
Convolutional	3x3	16
MaxPooling		
Convolutional	3x3	32
Convolutional	3x3	32
MaxPooling		
Convolutional	3x3	64
Convolutional	3x3	64
Convolutional	3x3	64
MaxPooling		
Fully connected		1024
Dropout (50%)		
Fully connected		1024
Dropout (50%)		
Fully connected		2
Softmax		

Training the network

- Computing environment



Training the network

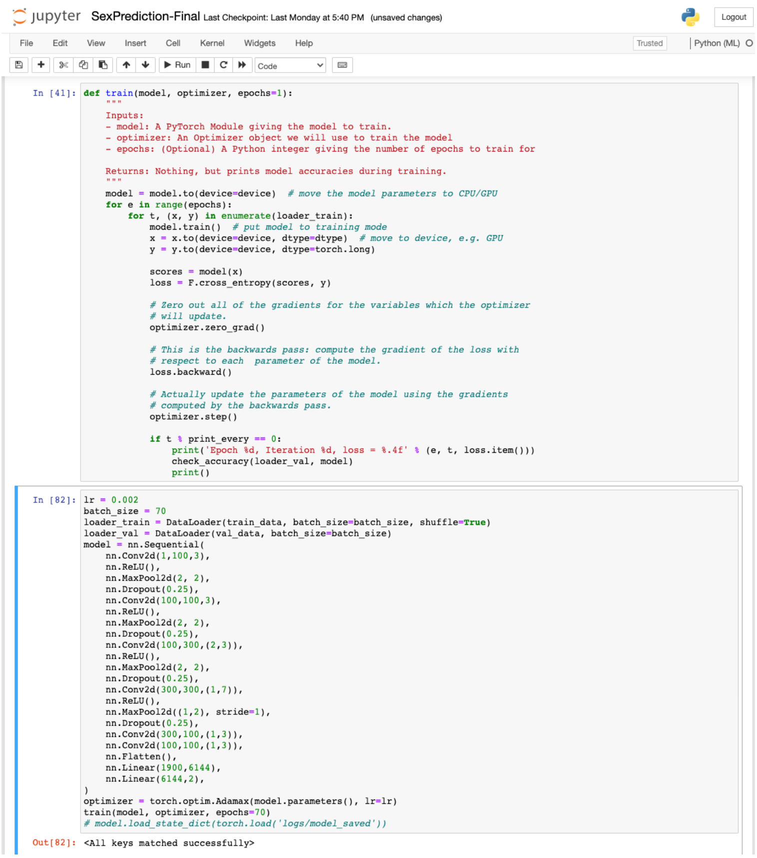
- Programming environment

python

v3.7.10

PyTorch

v1.3.1



jupyter SexPrediction-Final Last Checkpoint: Last Monday at 5:40 PM (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python (ML)

In [41]:

```
def train(model, optimizer, epochs=1):
    """
    Inputs:
    - model: A PyTorch Module giving the model to train.
    - optimizer: An Optimizer object we will use to train the model
    - epochs: (Optional) A Python integer giving the number of epochs to train for

    Returns: Nothing, but prints model accuracies during training.
    """
    model = model.to(device=device)  # move the model parameters to CPU/GPU
    for e in range(epochs):
        for t, (x, y) in enumerate(loader_train):
            model.train()  # put model to training mode
            x = x.to(device=device, dtype=dtype)  # move to device, e.g. GPU
            y = y.to(device=device, dtype=torch.long)

            scores = model(x)
            loss = F.cross_entropy(scores, y)

            # Zero out all of the gradients for the variables which the optimizer
            # will update.
            optimizer.zero_grad()

            # This is the backwards pass: compute the gradient of the loss with
            # respect to each  parameter of the model.
            loss.backward()

            # Actually update the parameters of the model using the gradients
            # computed by the backwards pass.
            optimizer.step()

            if t % print_every == 0:
                print('Epoch %d, Iteration %d, loss = %.4f' % (e, t, loss.item()))
                check_accuracy(loader_val, model)
                print()
```

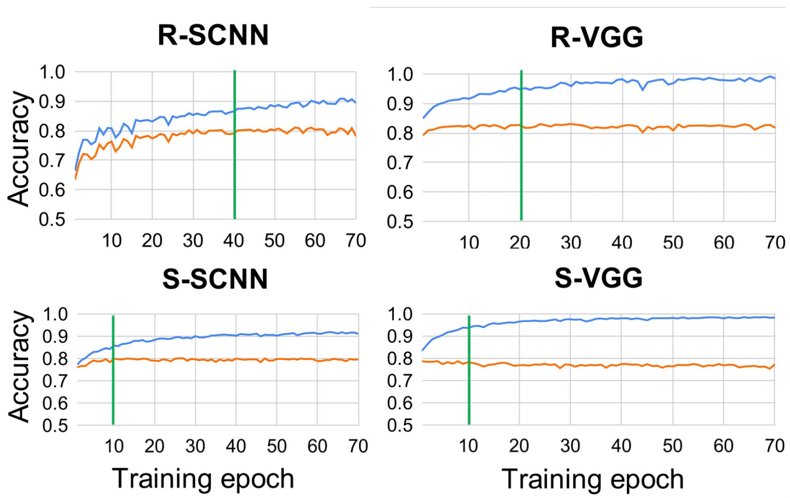
In [82]:

```
lr = 0.002
batch_size = 70
loader_train = DataLoader(train_data, batch_size=batch_size, shuffle=True)
loader_val = DataLoader(val_data, batch_size=batch_size)
model = nn.Sequential(
    nn.Conv2d(1,100,3),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.25),
    nn.Conv2d(100,100,3),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.25),
    nn.Conv2d(100,300,(2,3)),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.25),
    nn.Conv2d(300,300,(1,7)),
    nn.ReLU(),
    nn.MaxPool2d((1,2), stride=1),
    nn.Dropout(0.25),
    nn.Conv2d(300,100,(1,3)),
    nn.Conv2d(100,100,(1,3)),
    nn.Flatten(),
    nn.Linear(1900,6144),
    nn.Linear(6144,2),
)
optimizer = torch.optim.Adamax(model.parameters(), lr=lr)
train(model, optimizer, epochs=70)
# model.load_state_dict(torch.load('logs/model_saved'))
```

Out[82]:

```
<All keys matched successfully>
```

Results

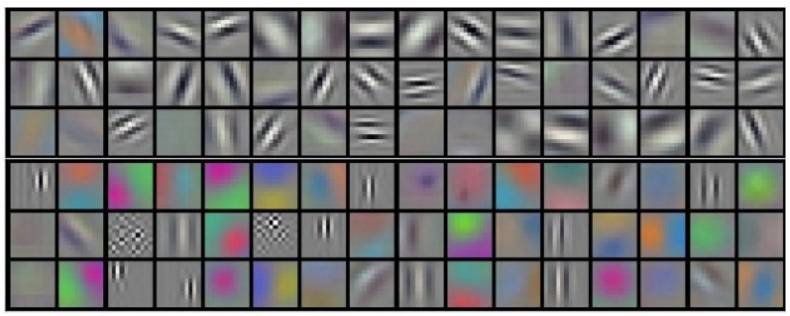


2-sec

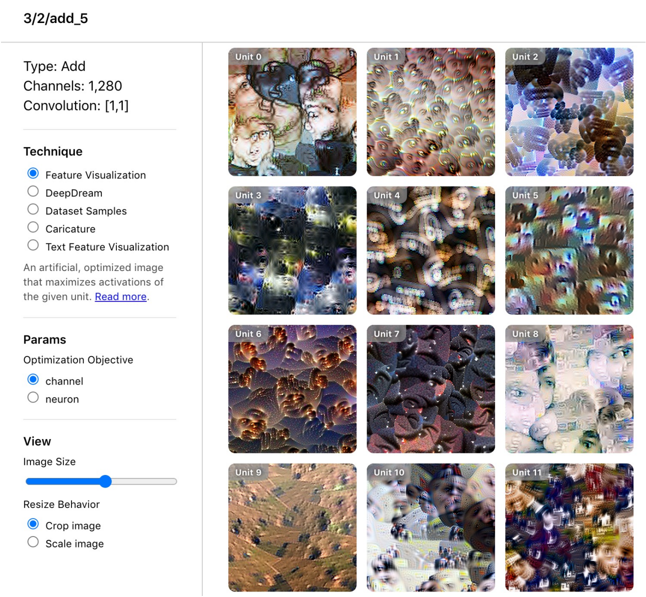
Model	Per-sample
R-SCNN	80.6 (79.7 to 81.5)
R-VGG	**83.1** (82.7 to 83.4)
S-SCNN	79.0 (78.7 to 79.3)
S-VGG	77.1 (76.8 to 77.4)

Truong, D., Milham, M., Makeig, S., and Delorme, A. Deep Convolutional Neural Network Applied to Electroencephalography: Raw Data vs Spectral Feature. Annu Int Conf IEEE Eng Med Biol Soc, 2021, pp. 1039-1042, doi: 10.1109/EMBC46164.2021.9630708.

Assessing features learned by the network



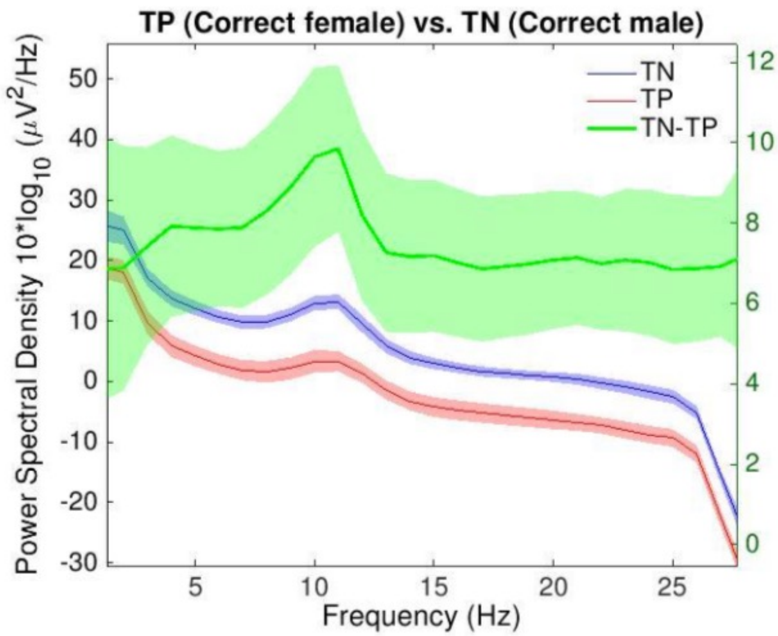
Krizhevsky et al., 2012



Best samples

- “Best” = gives highest activation in classification neurons
- Get samples of the top 20 subjects in validation set
- Best male samples show higher spectral power across frequencies, most notably the alpha band near 10Hz

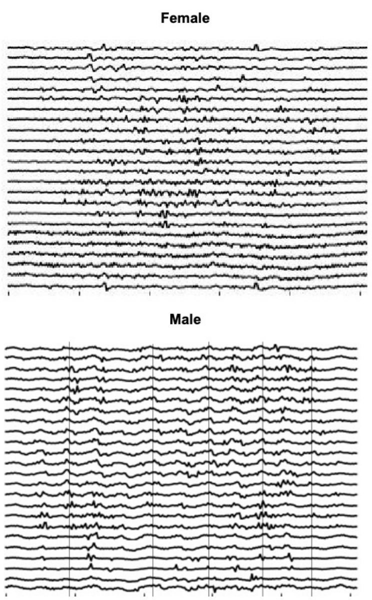
	Classified as female	Classified as male
Female sample	**True Positive (TP)**	False Negative (FN)
Male sample	False Positive (FP)	**True Negative (TN)**

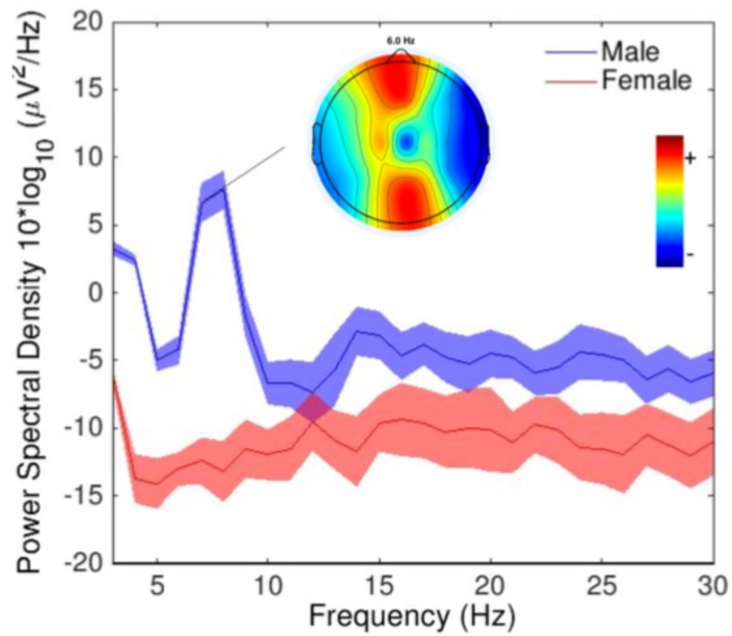


Truong, D., Makeig, S., & Delorme, A. (2021). Assessing learned features of Deep Learning applied to EEG. *arXiv preprint arXiv:2111.04309*.

Activation Maximization

- Synthesize the input that maximize the activation of the classification neurons
- Get 20 samples for each sex
- Best male samples show distinctly higher high theta power for male samples (6-8 Hz)

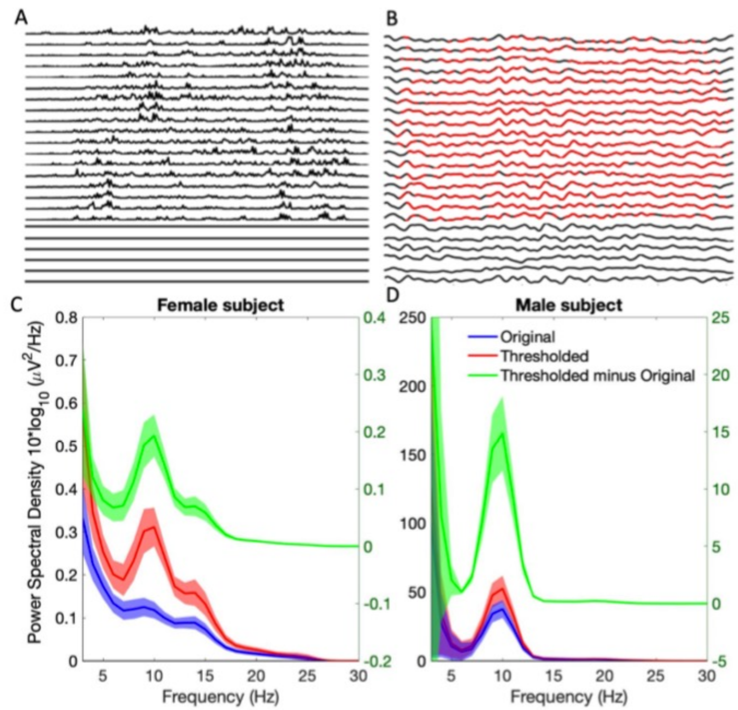




Truong, D., Makeig, S., & Delorme, A. (2021). Assessing learned features of Deep Learning applied to EEG. *arXiv preprint arXiv:2111.04309*.

Saliency map

- Back-project the gradient of the classification neuron to the input
- Magnitude of the gradients for each input value indicate the importance of that value to the neuron's activation
- Raw EEG samples contributing the least to the classification (gradients fell below the 30% quantile threshold) were removed then linearly interpolated using the remaining samples
- Thresholded samples showed higher alpha power, most notably near 10Hz



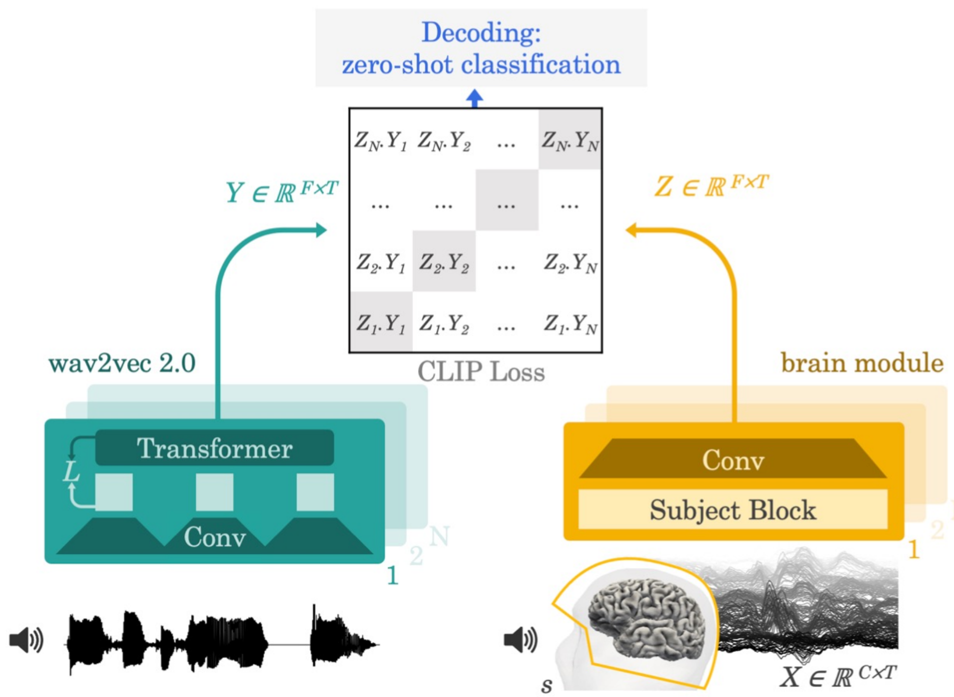
Truong, D., Makeig, S., & Delorme, A. (2021). Assessing learned features of Deep Learning applied to EEG. *arXiv preprint arXiv:2111.04309*.

Discussion

- Architecture design is not physiologically driven
- Not generalizable for different number of channels and sampling rate
- Could be a simple problem: boys move more → Train on normalized data

Decoding speech from non-invasive brain recordings

Alexandre Défossez[1,*], Charlotte Caucheteux[1,2], Jérémy Rapin[1], Ori Kabeli[1], and Jean-Rémi King[1,*]



Locating and Editing Factual Associations in GPT

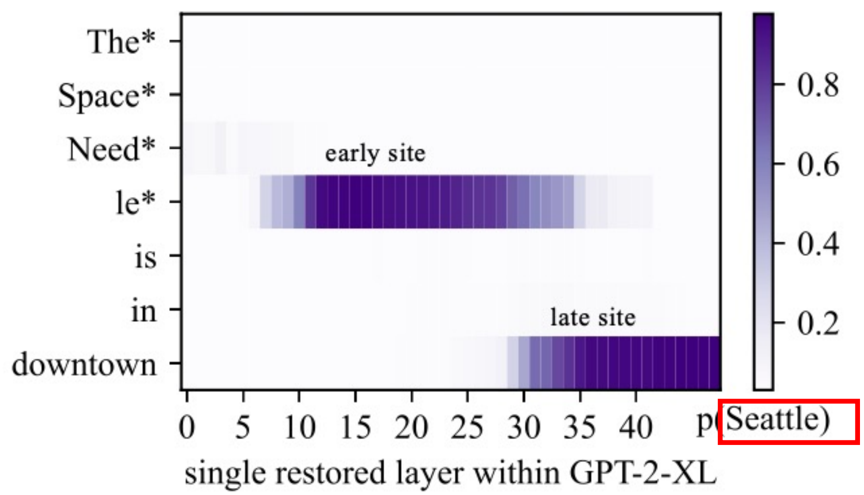
Kevin Meng*
MIT CSAIL

David Bau*
Northeastern University

Alex Andonian
MIT CSAIL

Yonatan Belinkov†
Technion – IIT

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



Locating and Editing Factual Associations in GPT

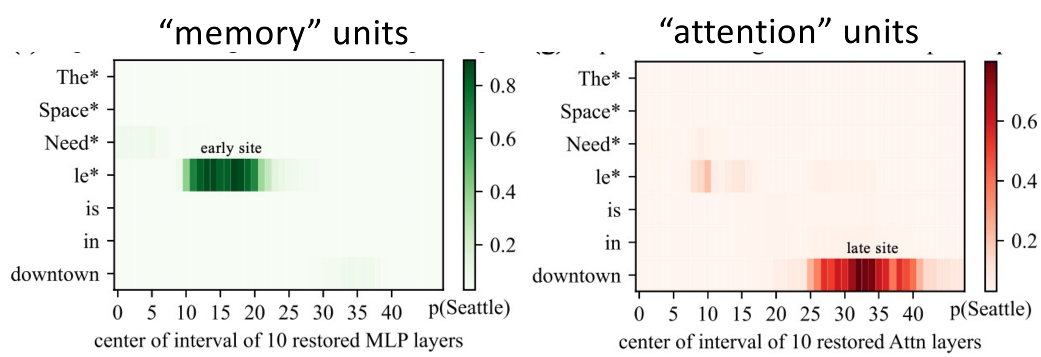
Kevin Meng* MIT CSAIL

David Bau* Northeastern University

Alex Andonian MIT CSAIL

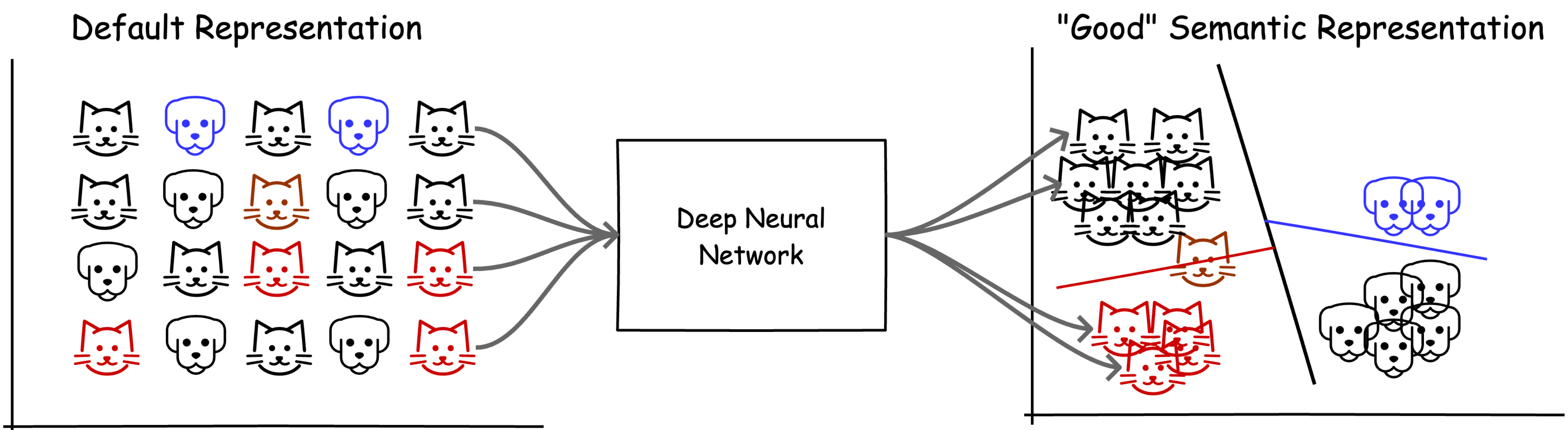
Yonatan Belinkov† Technion – IIT

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



Future work: representation learning

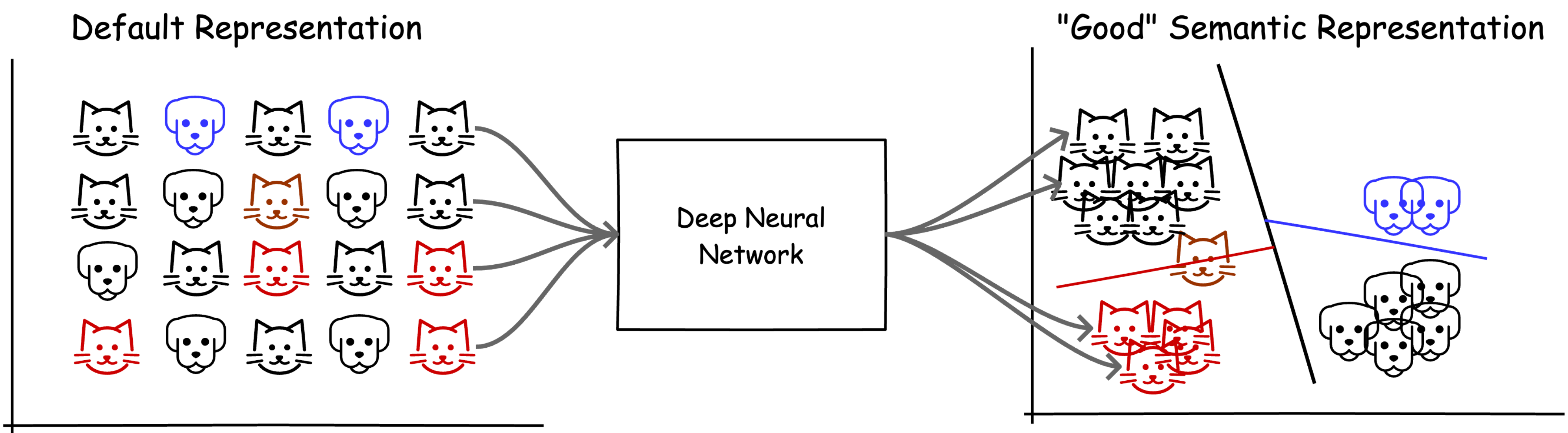
- DL models take data from the original space and map it to a "meaningful" representation



Cat by Martin LEBRETON, Dog by Serhii Smirnov from the Noun Project

Future work: representation learning

- DL models take data from the original space and map it to a "meaningful" representation
- Can we learn meaningful embeddings that are also generalizable?



Cat by Martin LEBRETON, Dog by Serhii Smirnov from the Noun Project

BIDS-DL plug-in

- Automated pipeline to convert HED-identified data segments from BIDS datasets to DL-ready dataset
- Host data on the cloud and make it streamable so that no data download/upload required

Truong, D., Sinha, M., Venkataraju, K. U., Milham, M., & Delorme, A. (2022) A streamable large-scale clinical EEG dataset for Deep Learning. The 44th International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC); July 11-15, 2022.

Thank you