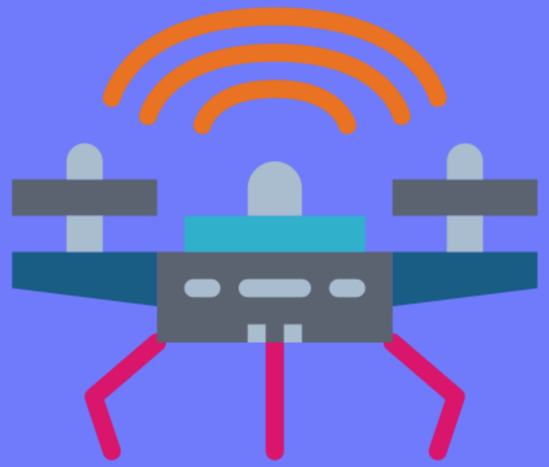
Section with Barbara

Week 5

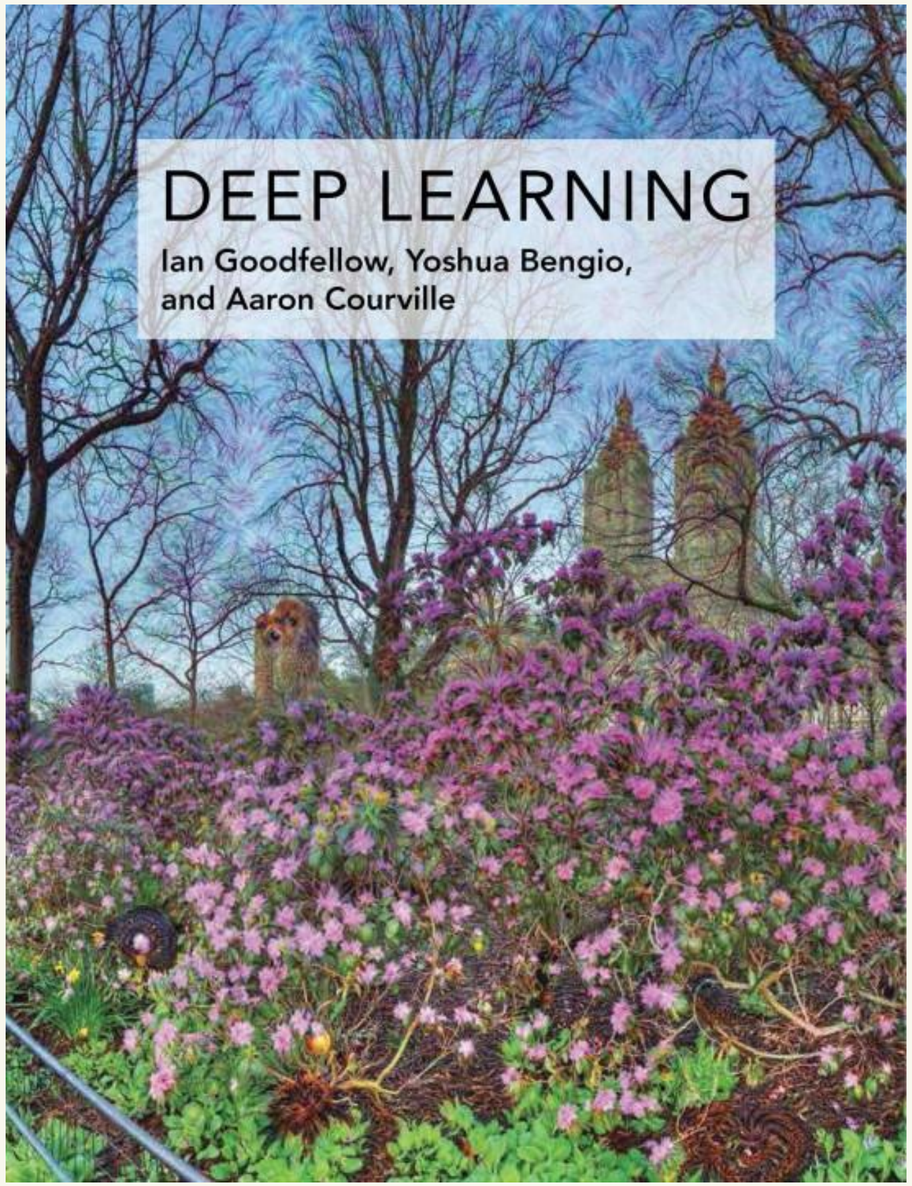
AI Stories

Projects

Neural Networks



AI Stories: NVIDIA "I am AI" Docuseries



https://www.fast.ai/

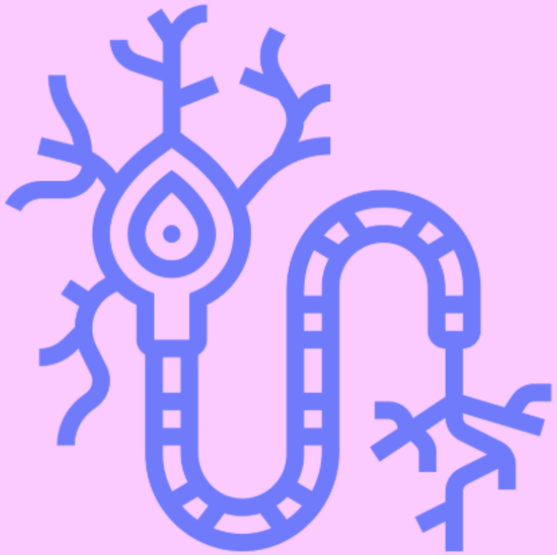
http://introtodeeplearning.com/

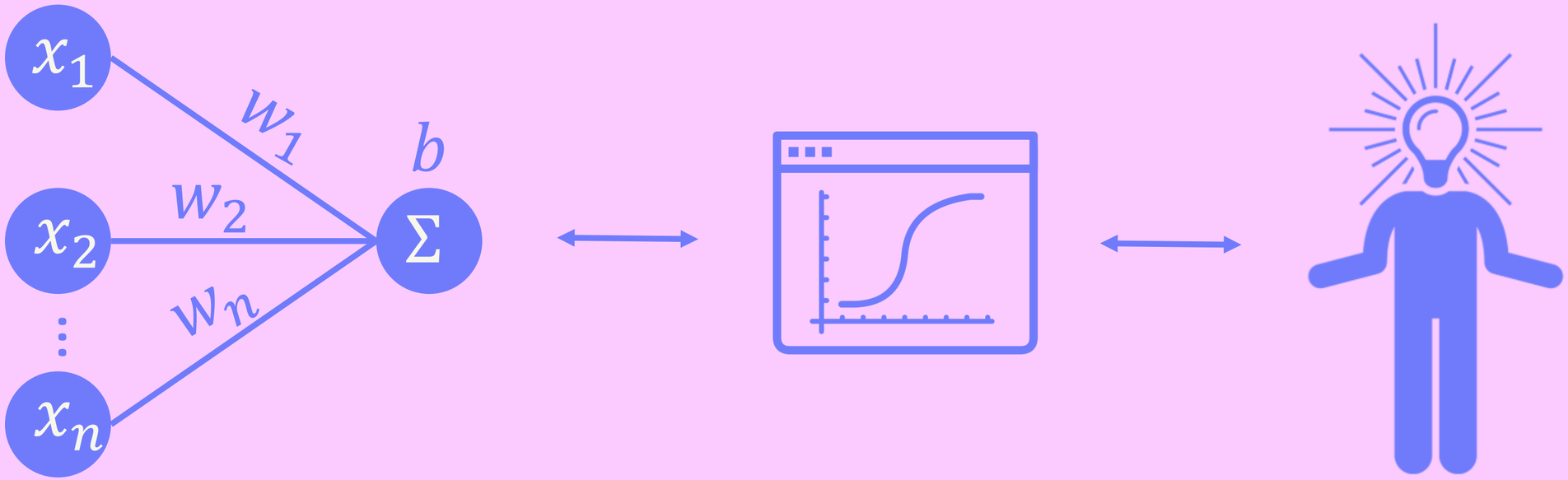
Bottomless wonders spring from simple rules…which are repeated without end

Benoit Mandelbrot

artificial neuron

≈



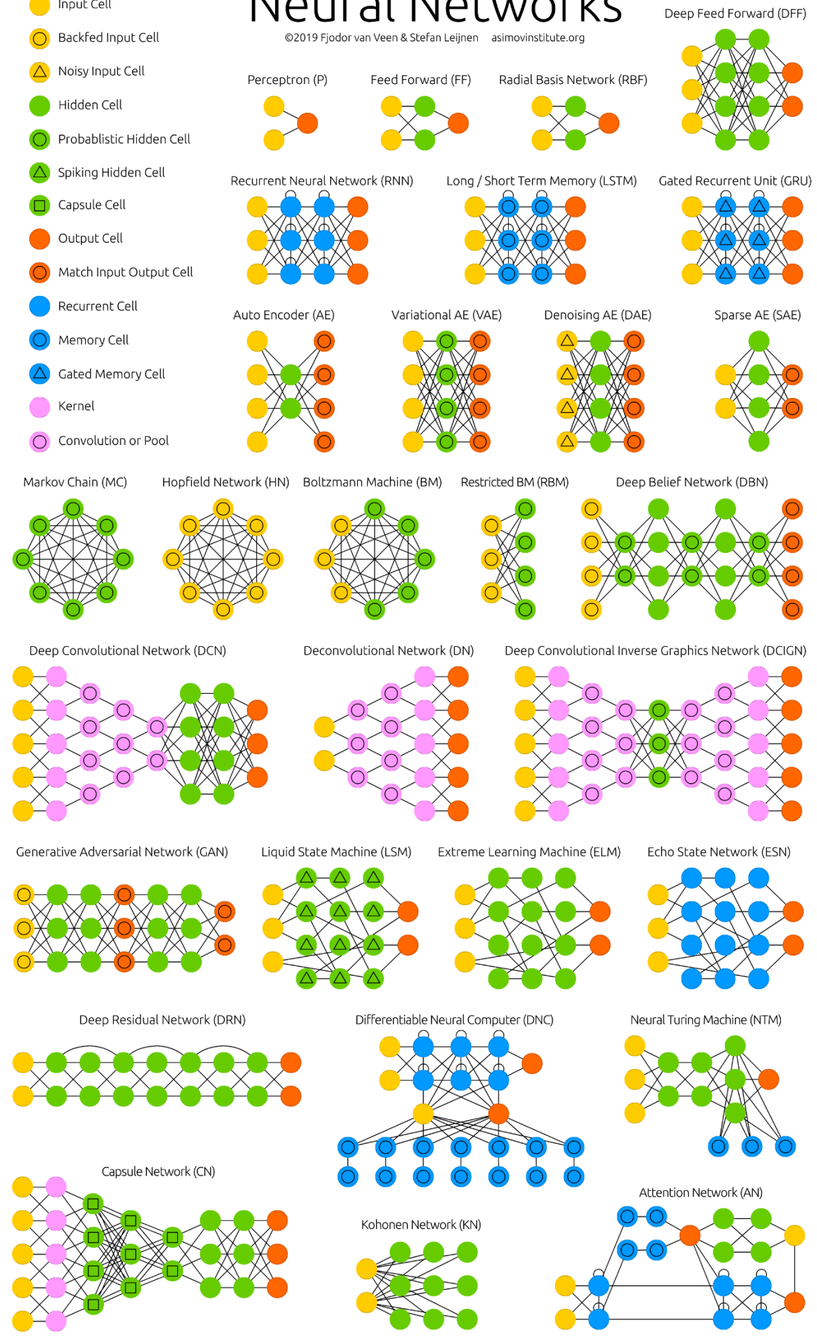

x_1
x_2
x_n
w_1
w_2
w_n
b
Σ

A mostly complete chart of

Neural Networks

 asimovinstitute.org

- Input Cell
- Backfed Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Gated Memory Cell
- Kernel
- Convolution or Pool



But do neural nets see the same way humans see?

https://distill.pub/2017/feature-visualization/

EXPERT LEARNING

SYSTEMS AND RULES

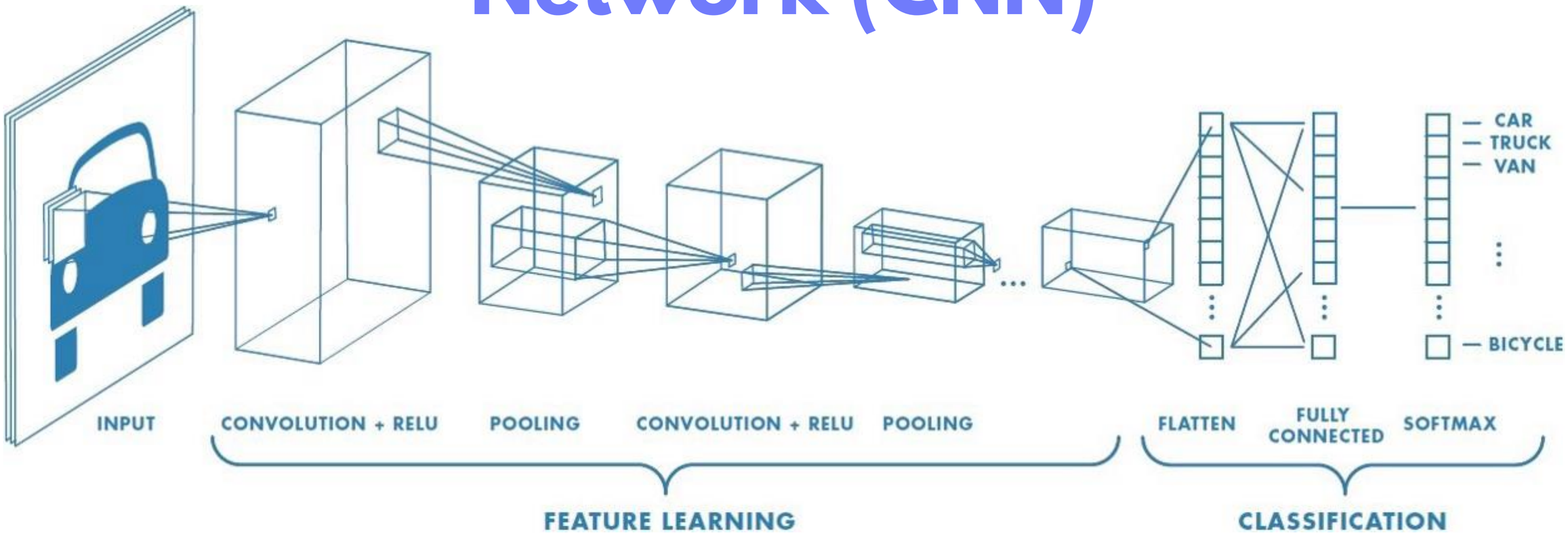
NOT FLEXIBLE
HIGH LEVEL
INTERPRETABLE

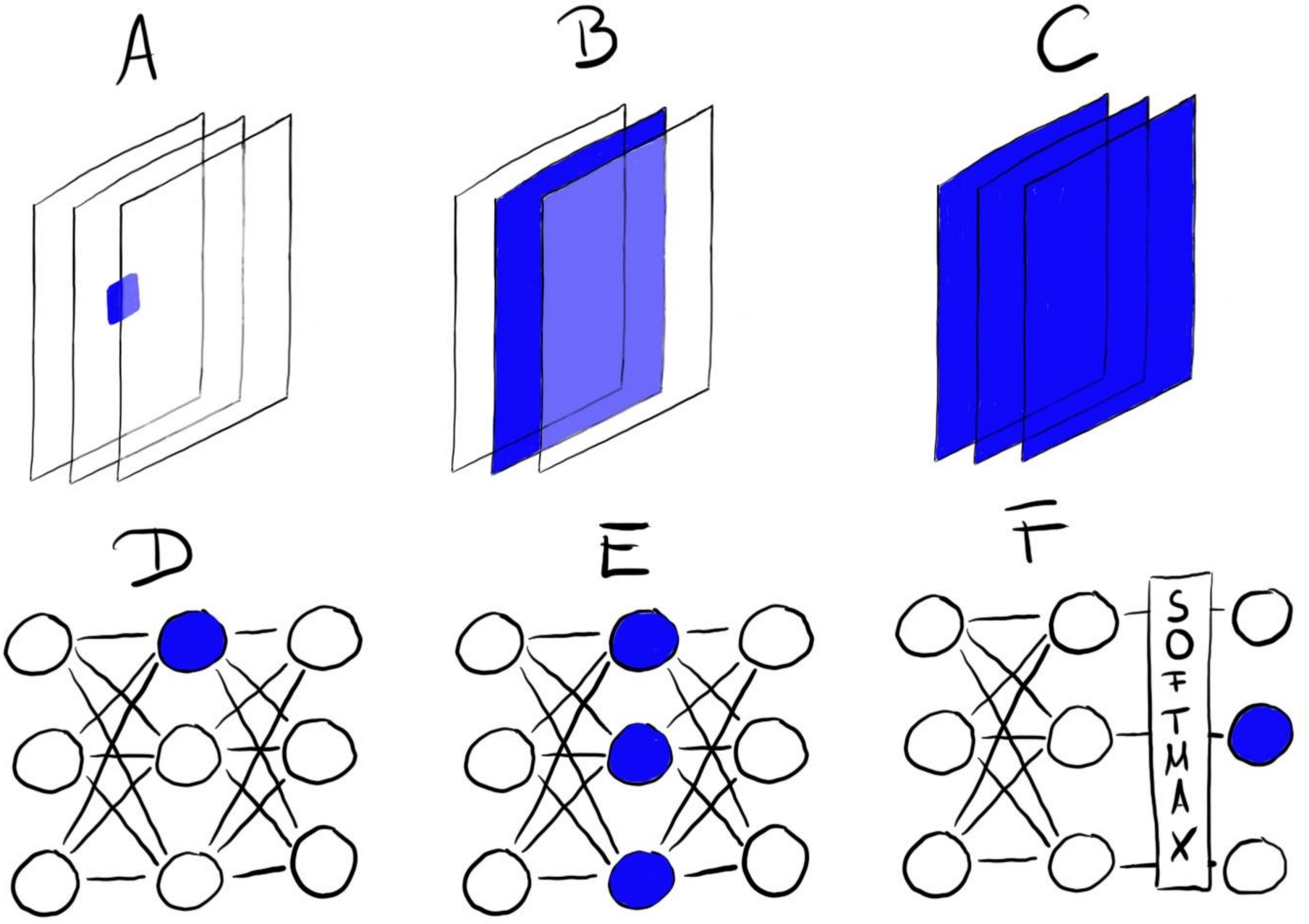
DEEP LEARNING

NEURAL NETWORKS

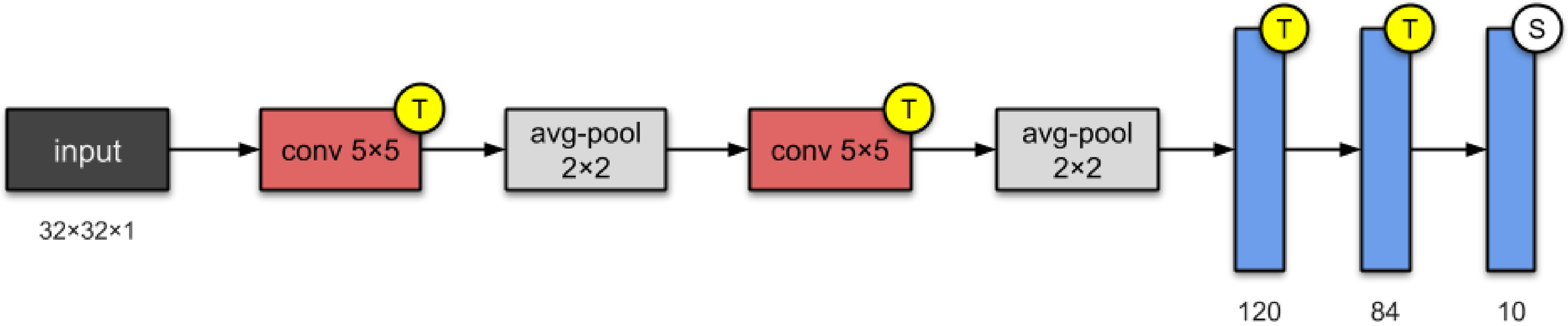
FLEXIBLE
LOW LEVEL
NOT INTERPRETABLE

Convolutional Neural Network (CNN)

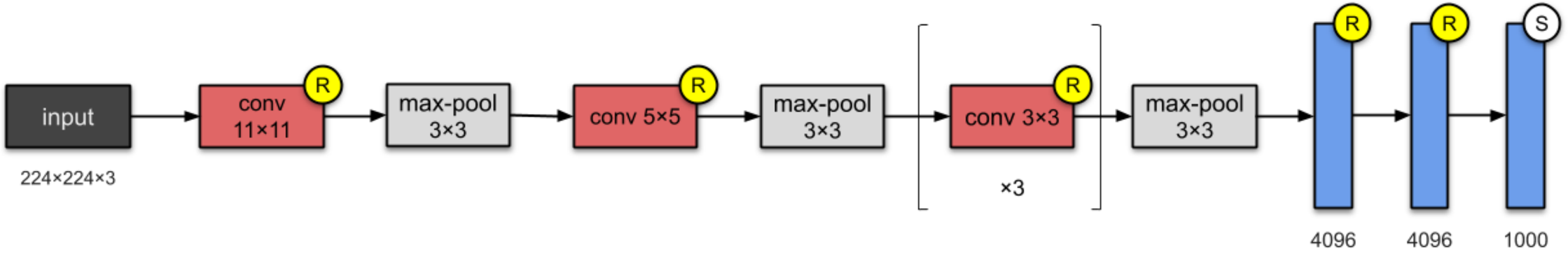



A
B
C
D
E
F
SOFTMAX

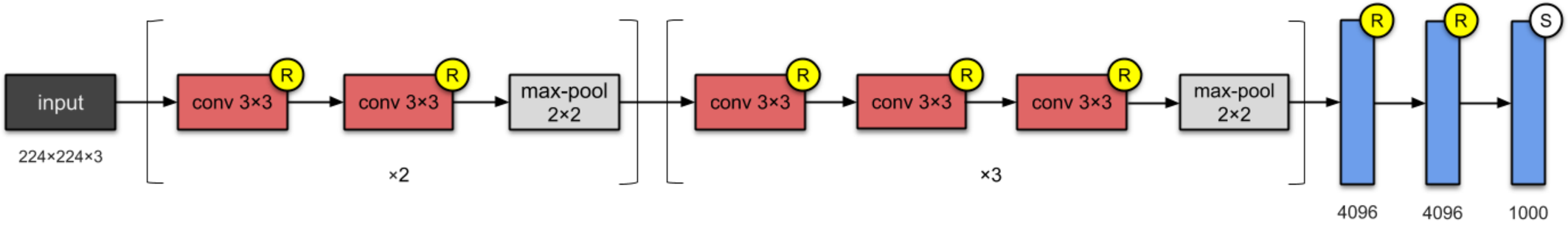
Le-Net 5


input
32×32×1
conv 5×5
T
avg-pool
2×2
conv 5×5
T
avg-pool
2×2
T
T
S
120
84
10

AlexNet


input
224×224×3
conv
11×11
R
max-pool
3×3
conv 5×5
R
max-pool
3×3
conv 3×3
R
×3
max-pool
3×3
R
R
S
4096
4096
1000

VGG-16


input
224×224×3
conv 3×3
R
conv 3×3
R
max-pool
2×2
×2
conv 3×3
R
conv 3×3
R
conv 3×3
R
max-pool
2×2
×3
R
R
S
4096
4096
1000

Inception Network
ResNet
Xception
Network in Network

projects

Another resource for datasets, if you want to do an exploratory project!

https://github.com/awesomedata/awesome-public-datasets

traffic


```
def load_data(data_dir):
    """
    Load image data from directory `data_dir`. Assume
    `data_dir` has one directory named after each category,
    numbered 0 through NUM_CATEGORIES - 1. Inside each
    category directory will be some number of image files.
    Return tuple `(images, labels)`. `images` should be a
    list of all the images in the data directory, where each
    image is formatted as a numpy ndarray with dimensions
    IMG_WIDTH x IMG_HEIGHT x 3. `labels` should be a list of
    integer labels, representing the categories for each of
    the corresponding `images`.
    """
```

Don’t make this complicated - cv2.imread() and cv2.resize() will be helpful here!

traffic


```
def get_model():
    """
    Returns a compiled convolutional neural network model.
    Assume that the `input_shape` of the first layer is
    `(IMG_WIDTH, IMG_HEIGHT, 3)`. The output layer should
    have `NUM_CATEGORIES` units, one for each category.
    """
```

Experiment to your heart’s fancy! I also encourage people to check out Batch Normalization!

If you want to see what the state-of-the-art accuracy is for traffic!

http://benchmark.ini.rub.de/

Click on the GTSRB tab

If you want to visualize what your CNN “saw” in each layer

https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html