AI Deep Learning: Convolutional Neural Networks (III)

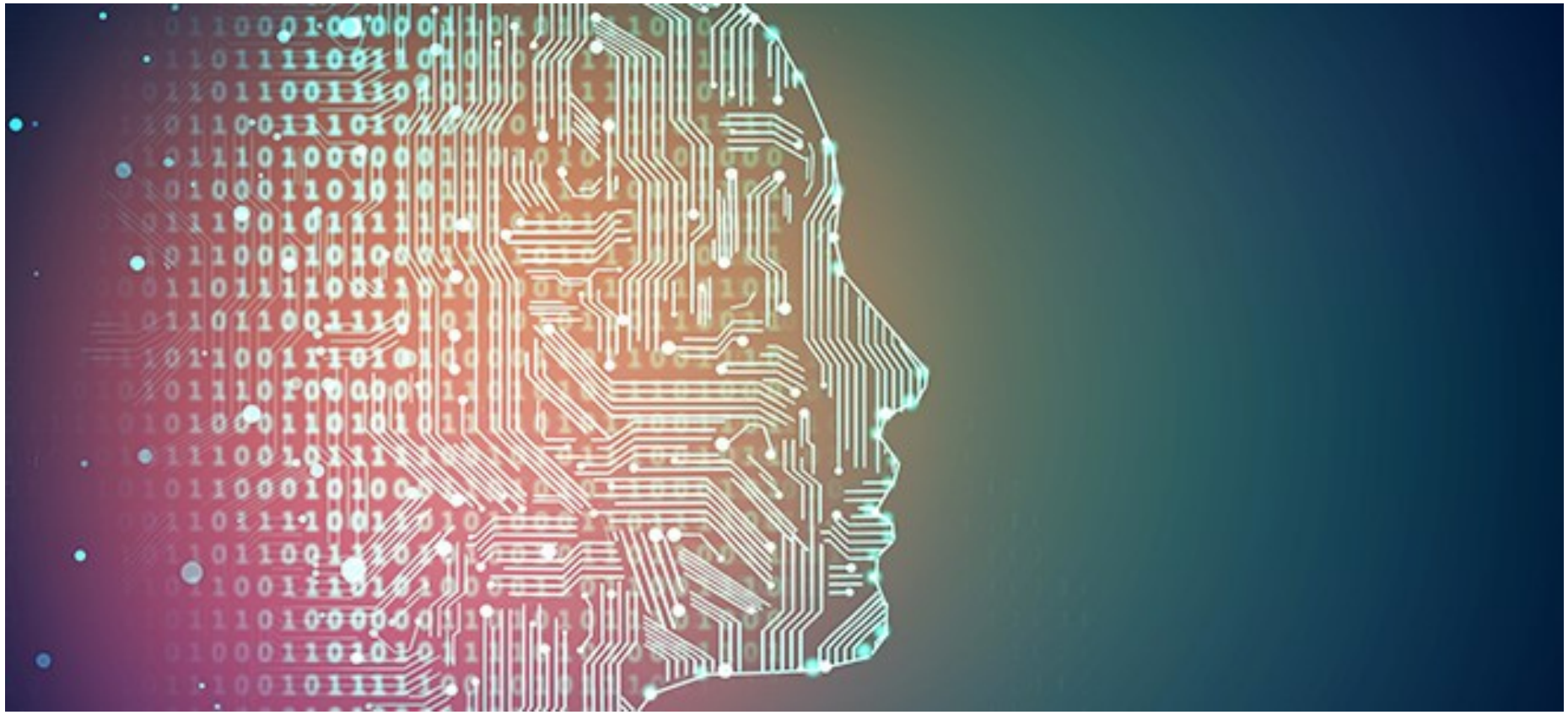
Deep Learning with Big Data

Thuan L Nguyen, PhD

AI Deep Learning: Convolutional Neural Networks (CNN)

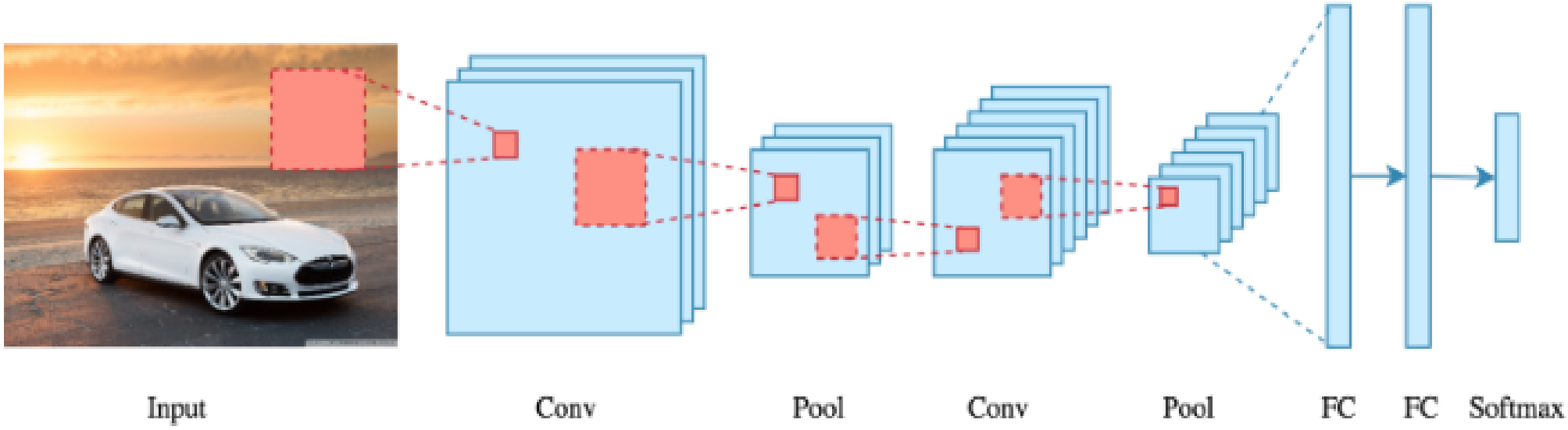
1. AI Deep Learning: Convolutional Neural Networks Overview
2. AI Deep Learning: Properties of Images
3. AI Deep Learning: Convolution Layers in Convolutional Neural Networks
4. AI Deep Learning: Pooling Layers in Convolutional Neural Networks
5. AI Deep Learning: Convolutional Neural Networks: Architectures: Overview
6. AI Deep Learning: Convolutional Neural Networks: Architectures: Layers
7. AI Deep Learning: Build and Train Convolutional Neural Networks

AI Deep Learning: Convolutional Neural Networks (CNN)



AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Overview

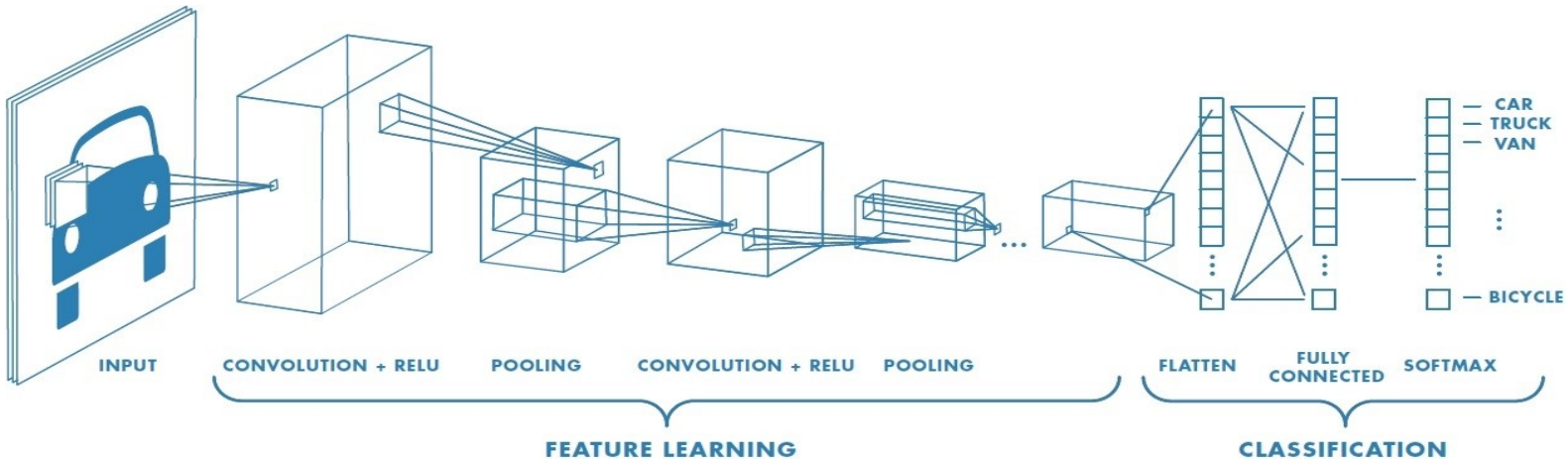


Sources: http://towardsdatascience.com

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers

- **CNN** van be viewed as the combination of two sub-networks:
  - First sub-network: **the feature-learning sub-network**:
    - This sub-network comprises layers that aim to learn or extract the features from inputs.
  - Second sub-network: **the fully-connected sub-network**:
    - That is very similar to a normal multi-layered neural network, i.e., multi-layer perceptron (MLP)
    - That comprises of fully-connected layers
    - That processes the outputs from the feature-learning sub-neural network and finishes the jobs



Sources: http://www.mathworks.com

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers

In general, besides the input and output layer:

- There are three main types of layers in a convolutional neural networks,
  - **Convolution layers (CONV)**
  - **Pooling layers (POOL)**
  - **Fully-connected layers (FC)**

There are other layers that work hand-in-hand with the above main layers

For examples:

- Reshape/Flattening layers where multi-dimensional data are flattened into 1D data, i.e., a vector.
- ReLU layers that are actually where the activation functions process data progressively.
- Softmax layer where the last activation function (softmax) processes outputs from the fully-connected layer and produces the final outputs.
- Dropout layer where some inputs are randomly dropped before final outputs are produced.

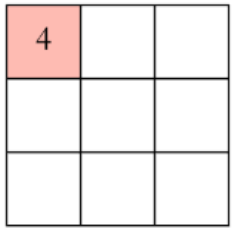
AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: Convolution Layer (CONV)

- The convolution layer is the core building block of a convolutional neural network.
  - That does most of its computational heavy lifting.

- Convolution is the first layer, after the input layer, of a convolutional neural network.
  - A convolution layer tries to extract higher-level features by replacing data for each (one) pixel with a value computed from the pixels covered by **the filter**, e.g., a 5x5 filter, centered on that pixel (all the pixels in that region).
  - Convolution **preserves** the relationship between pixels by **learning** image features using small squares of input data, i.e., the filter.

Input x Filter

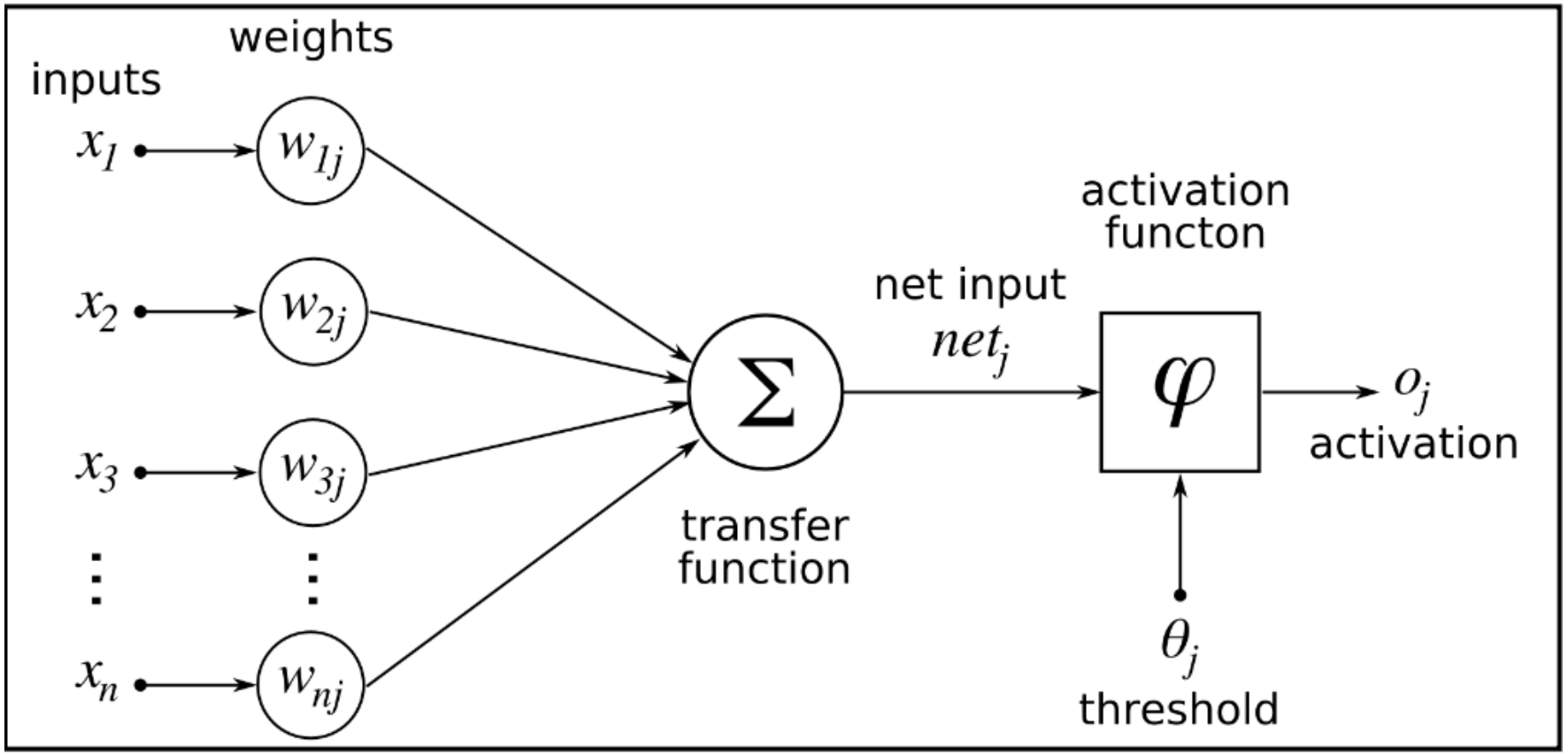


Feature Map

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: CONV: Activation Function - ReLU

- Let's review the simple McCulloch-Pitts neuron model:



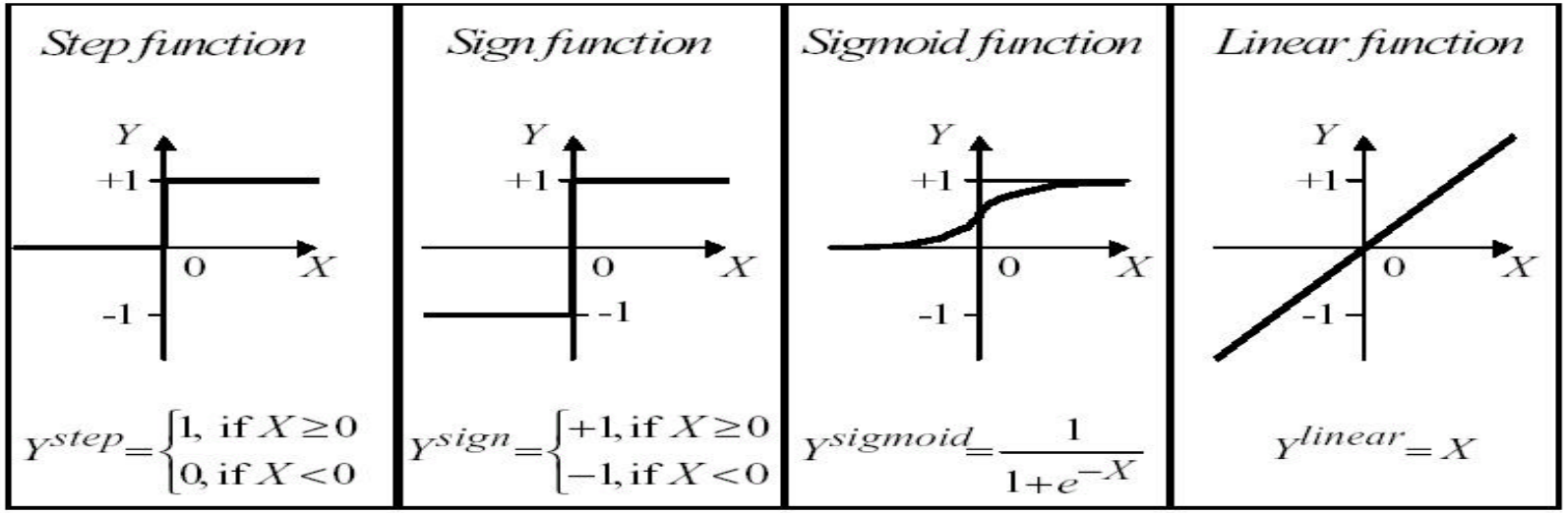
Sources: https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: CONV: Activation Function - ReLU

- Let's review the simple McCulloch-Pitts neuron model:
  - The inputs multiplied by the weights are summed up by the neuron.
  - The result is delivered as inputs to an activation function that processes the input and produces an output.

- Various functions can be used as activation functions in an artificial neural network. The following functions are commonly used:

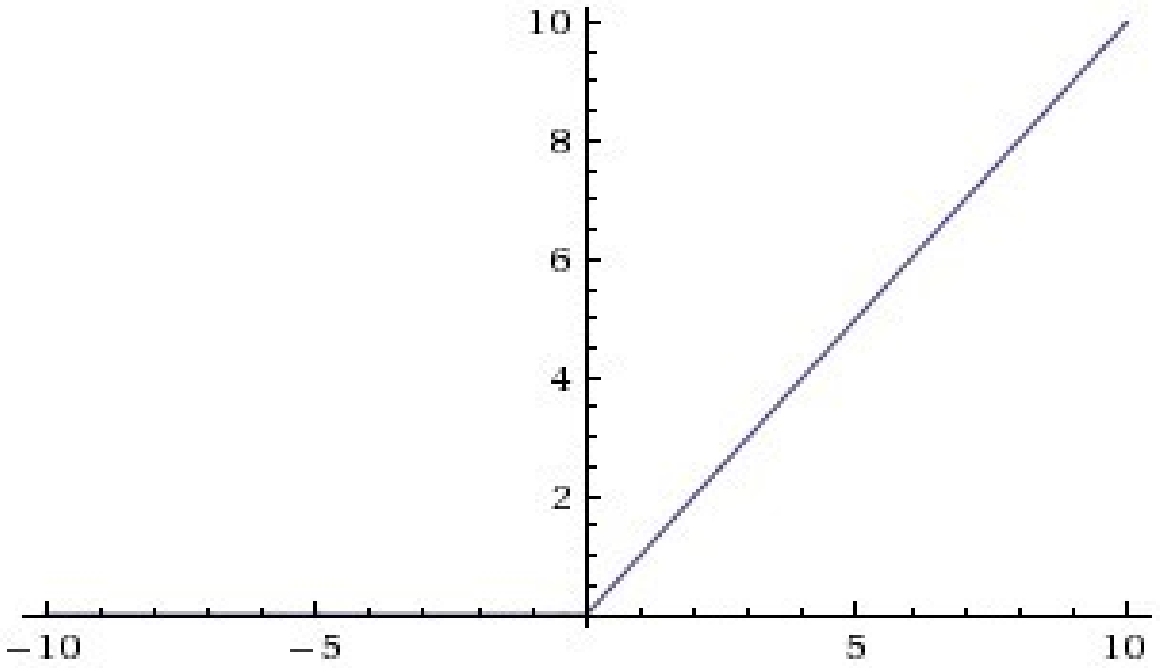


AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: CONV: Activation Function - ReLU

For convolutional neural networks:

- ReLU (Rectified Linear Unit) is one of the most used activation functions: y = max(0, x)
  - x >= 0 → y = x
  - x < 0 → y = 0



AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: CONV: Activation Function - ReLU

For **convolutional neural networks**:

- ReLU (Rectified Linear Unit) is important to convolutional neural networks.
  - Because it can introduce non-linearity into the network.

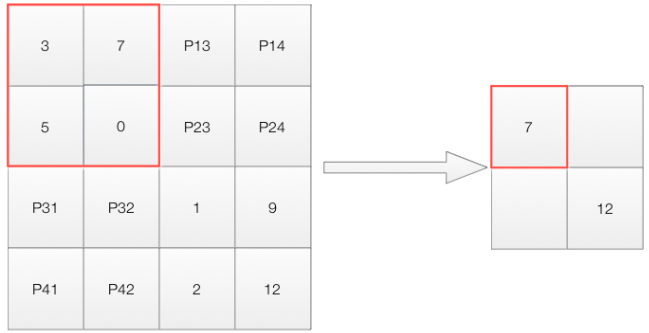
- **ReLU** can improve neural network performance by speeding up training:
  - The gradient computation is very simple (either 0 or 1 depending on the sign of x).
  - Also, the computational step of a ReLU is easy: any negative elements are set to 0.0 -- no exponentials, no multiplication or division operations.

- Gradients of logistic and hyperbolic tangent (*two other popular activation functions*) networks are smaller than the positive portion of the ReLU.
  - This means that the positive portion is updated more rapidly as training progresses.
  - However, this comes at a cost: The 0 gradient on the left-hand side has its own problem, "dead neurons," in which a gradient update sets the incoming values to a ReLU such that the output is always zero.
  - Modified ReLU units such as ELU (or Leaky ReLU, or PReLU, etc.) can ameliorate this.

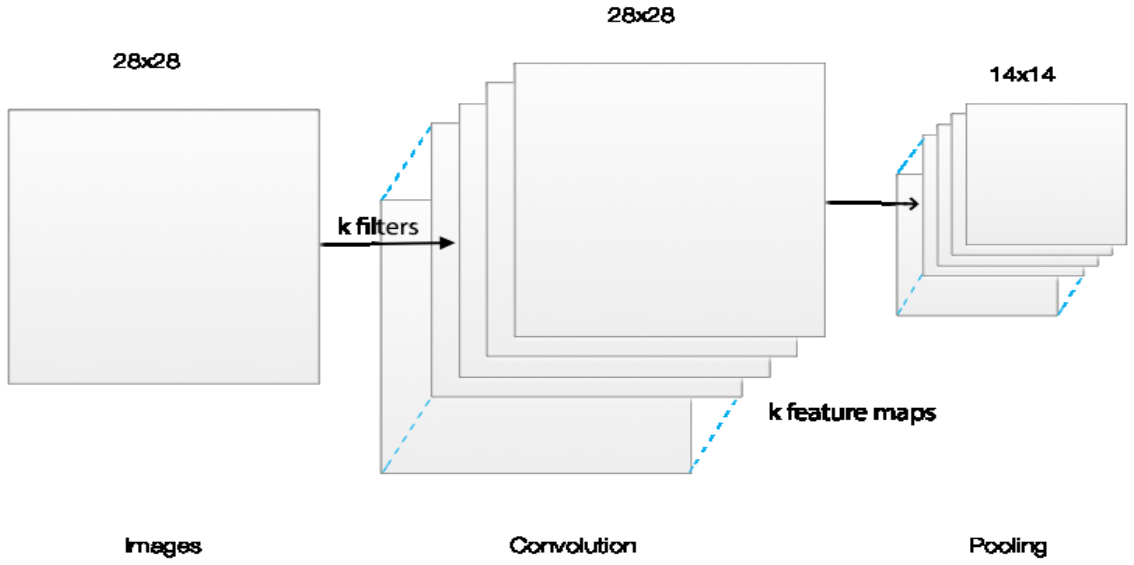
AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: Pooling Layer (POOL)

- The pooling or down-sampling layer is responsible for reducing the spatial size of the activation maps.
  - Pooling layers accept inputs from convolution layers and non-linearity activation to reduce the computational requirements progressively through the network as well as minimizing the risk of overfitting.

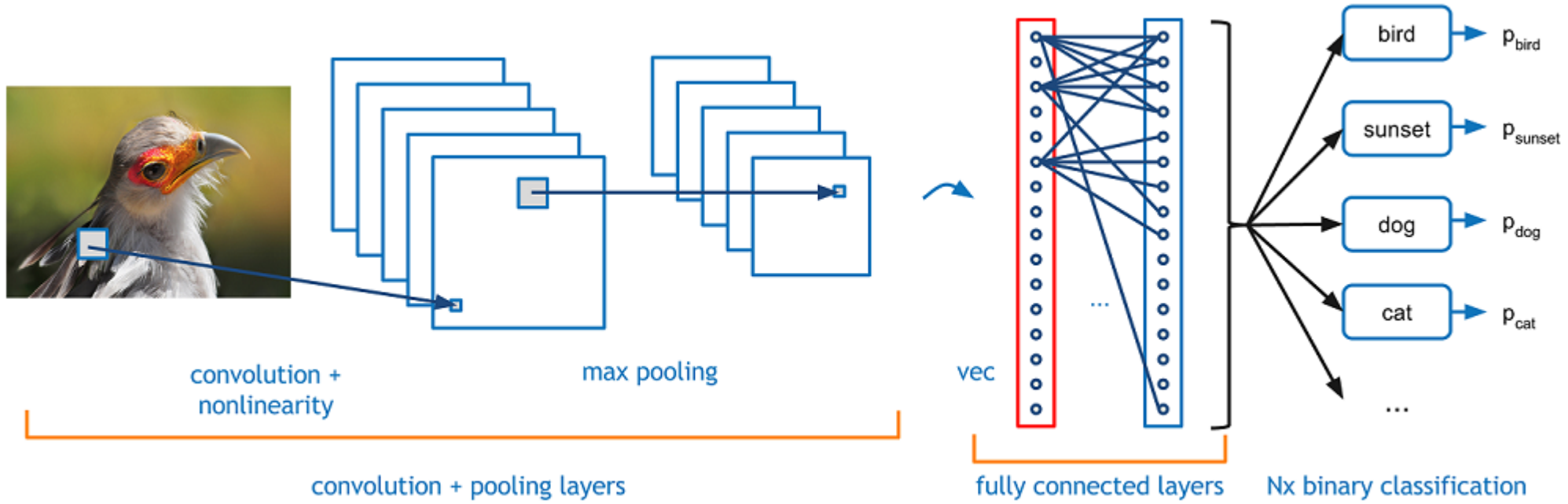


- For example:
  - To reduce the spatial dimension of a feature map, we apply maximum pool (max_pooling).
  - A 2x2 maximum pool replaces a 2x2 area by its maximum.
  - After applying a 2x2 pool, the spatial dimension is reduced from 4x4 to 2x2. (Filter size=2, Stride = 2)



AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures



Sources: Adit Deshpande

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Layers: Fully-Connected Layer (FC)

- After using a series of convolution layers to extract the spatial features of an image:
  - One or more fully connected layers are used to finish the jobs and produce the final outputs

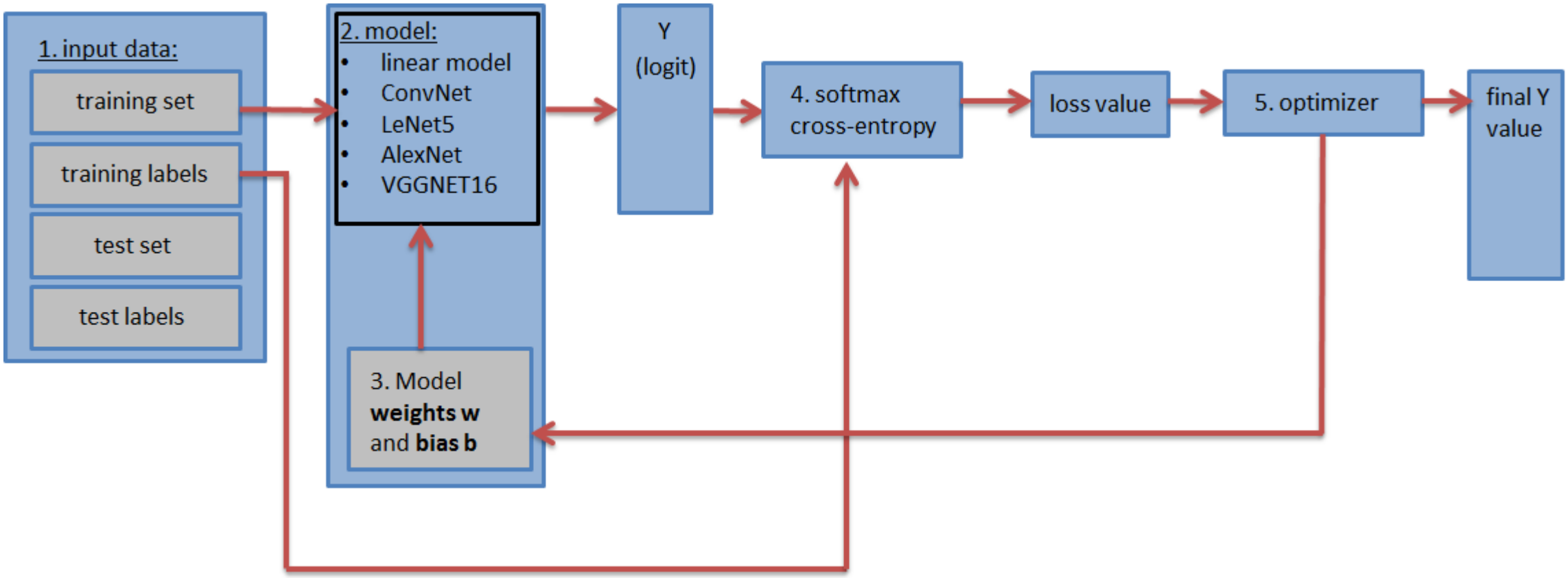
- First, the outputs of the convolution layers need to be reshaped or flattened into a vector:
  - For example, if the final features maps have a dimension of 4x4x512, we will flatten it to an array of 8192 elements.

- Then the data (the vector) is fed into the fully-connected layer(s).
  - That produces outputs that will be in turn fed into the final activation functions (Softmax).

- Some of the outputs of the activation functions (Softmax) will be dropped out randomly before the final outputs are produced.

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Overview



Sources: Ahmet Taspinar

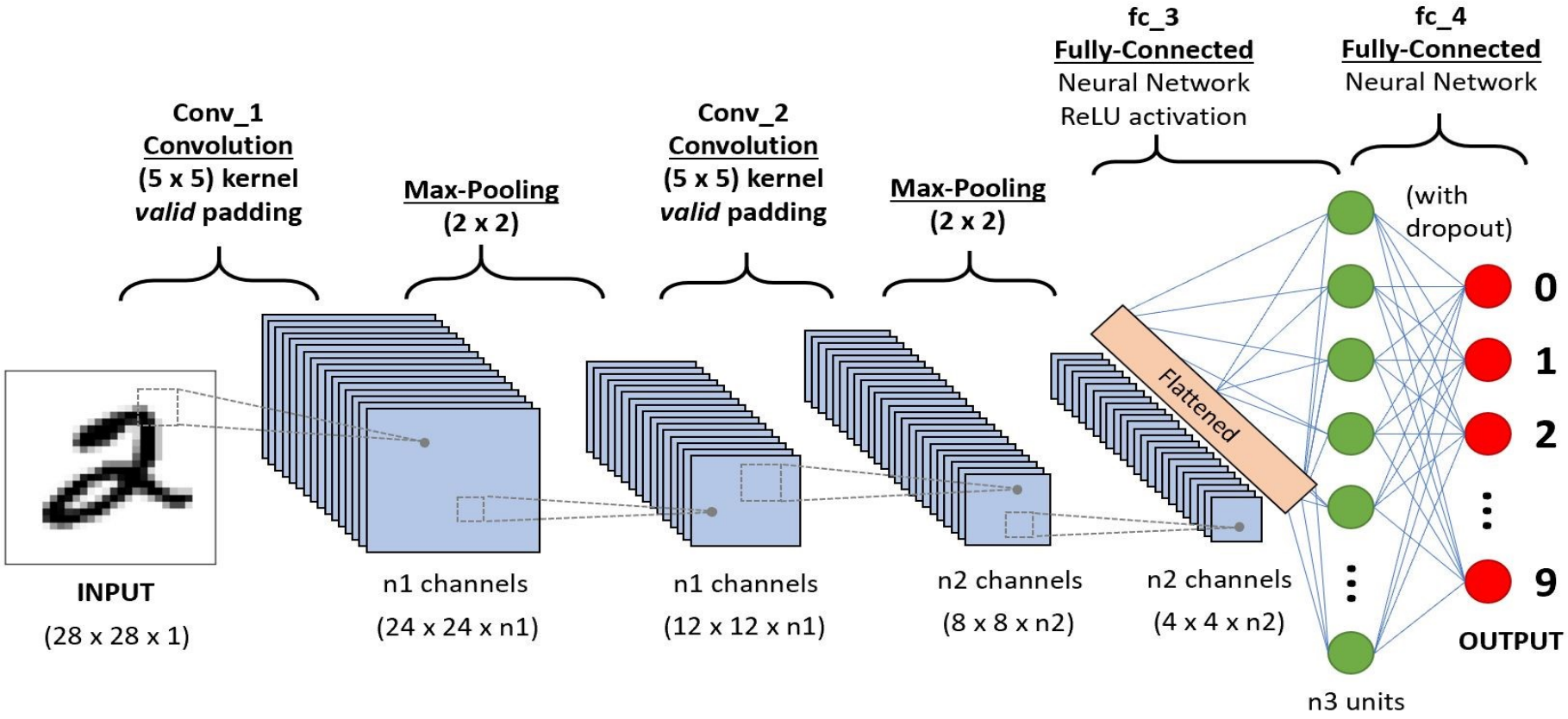
AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architectures: Overview

- The graph containing the Neural Network should contain the following components
  - The **input datasets**: the training and the test dataset and labels (and the validation dataset and labels).
    - The test and validation datasets can be placed inside a tf.constant().
    - The training dataset is placed in a tf.placeholder() so that it can be feeded in batches during the training (stochastic gradient descent).
  - The **artificial neural network model** with all of its layers.
  - The **weight matrices and bias vectors** defined in the proper shape and initialized to their initial values. (One weight matrix and bias vector per layer.)
  - The **softmax cross-entropy** and **loss value**:
    - The model has as output the logit vector (estimated training labels).
    - By comparing the logit with the actual labels, it is possible to calculate the loss value (with the softmax with cross-entropy function).
    - The loss value is an indication of how close the estimated training labels are to the actual training labels and will be used to update the weight values.
  - An **optimizer**, which will use the calculated loss value to update the weights and biases with backpropagation.

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Architecture: Layers



Sources: Ryan P. Marchildon

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Build and Train Deep Learning Convolutional Neural Networks

Two phases: Building the convolutional neural network and train the network

- Phase 1: Build the convolutional neural network
  - Step 1: Create the 1st convolution layer (CONV) that uses ReLU as the activation function
  - Step 2: Create the 1st pooling layer (POOL) that works hand-in-hand with the convolution layer
    - Repeat Step 1 and 2 until finishing all the convolution and pooling layers have been created
  - Step 3: Create a reshape/flattening layer to reshape output data from the last pooling layer
  - Step 4: Create a fully-connected (FC) layer that accepts outputs from the reshape layer as its inputs
    - Repeat Step 4 to create other fully-connected layer if necessary
  - Step 5: Create a dropout layer to drop some of its inputs that it receives.

AI Deep Learning: Convolutional Neural Networks (CNN)

CNN: Build and Train Deep Learning Convolutional Neural Networks

Two phases: Building the convolutional neural network and train the network

- Phase 1: Build the convolutional neural network (Cont.)
  - Step 6: Create the final output layer, an FC layer, that accepts the outputs from the dropout layer as its inputs and produces the final outputs.
  - Step 7: Create the loss function to compute softmax cross entropy between the labels and the predicted outputs
    - *i.e., Measures **the probability error** in discrete classification tasks in which the **classes are mutually exclusive** (each entry is in exactly one class)*
  - Step 8: Create an optimizer to optimize the network
  - Step 9: Create a AI deep learning trainer to train the network
- Phase 2: Train the convolutional neural network
  - Train the network using the AI deep learning trainer created above