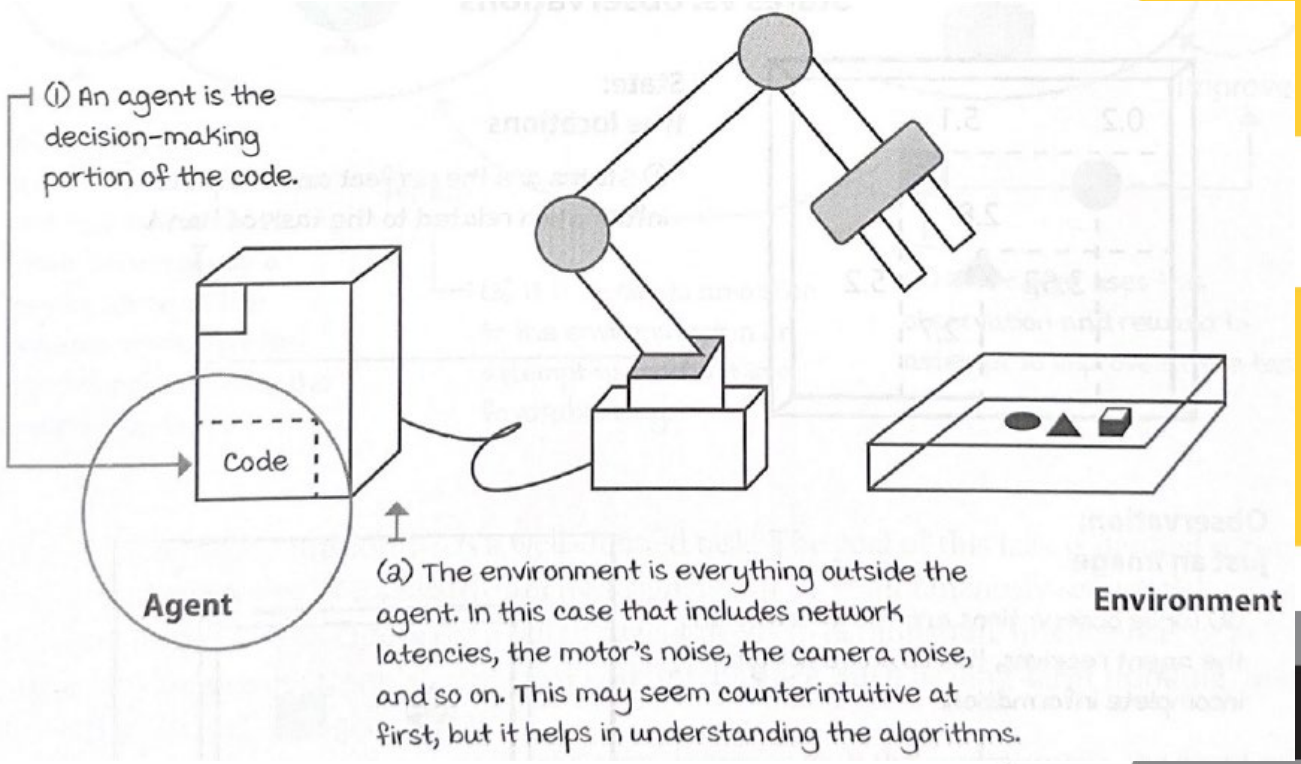
Introduction to Neural Networks

In RL

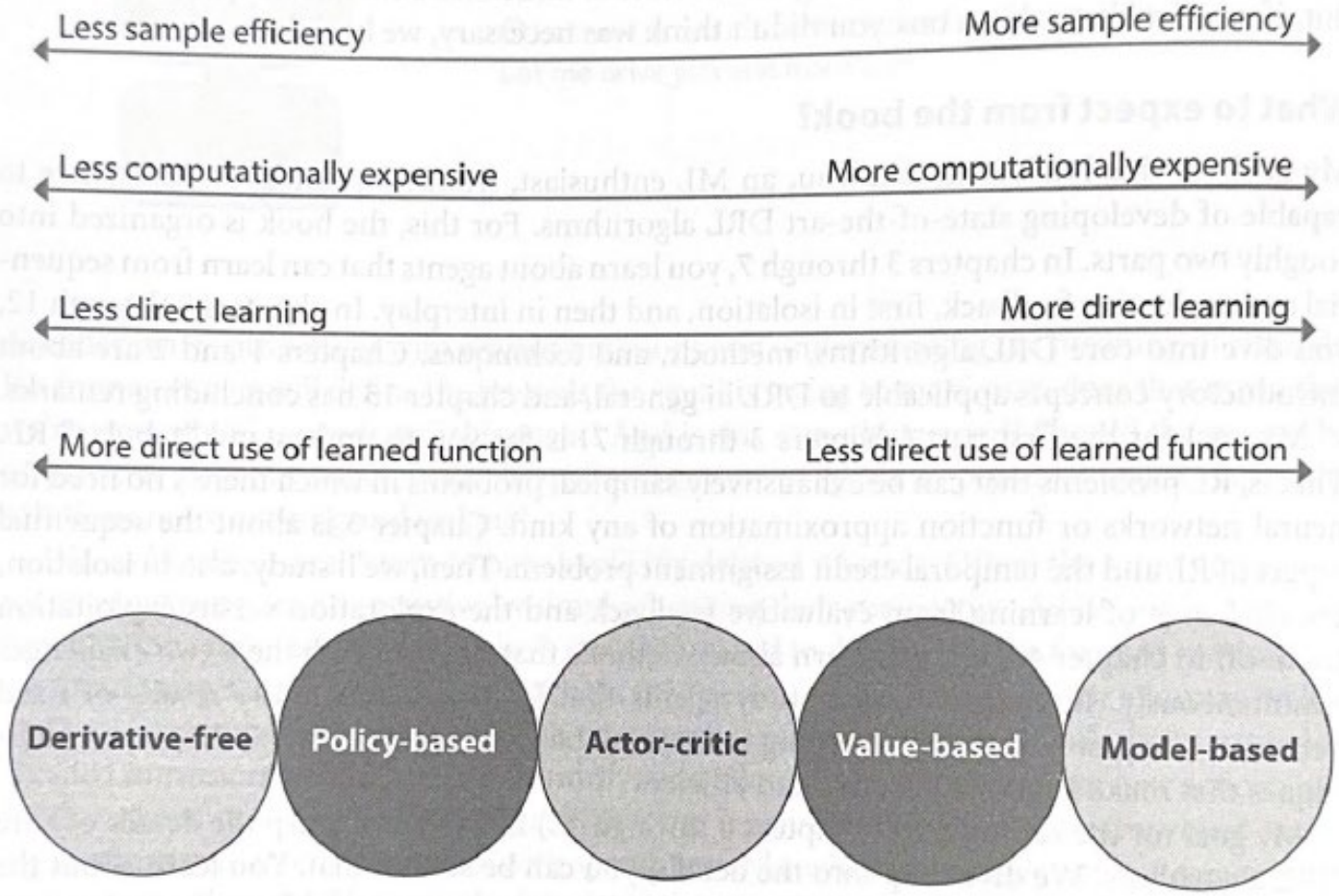
- Three approaches to learning:
  - Policies: map observations to actions
    - Depending on configuration:
      - Off-policy: Q-Learning
      - On-policy: SARSA
  - Models: learn the environment based on a model
    - Target model: goal of the agent
    - Base model: interaction of the agent with the environment (computationally intensive)
  - Value functions: learn to estimate the reward-to-go on mappings



- As more observations, more Q-values, more new states are generated:
  - Interaction of the RL agent and environment grows significantly.
  - An RL approach for more complex environments will be computationally expensive.

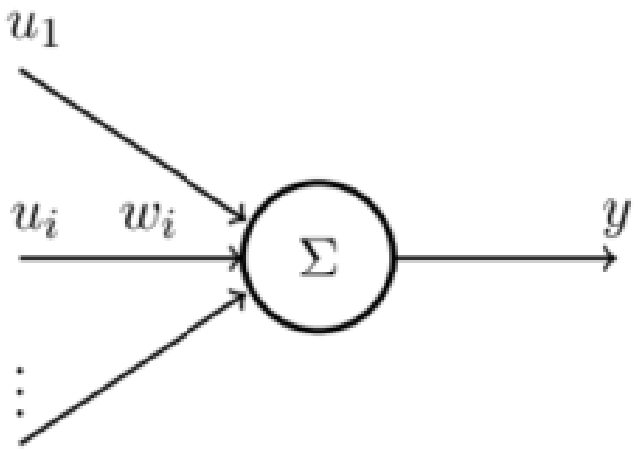
- Since RL is based on “trial – error”, an RL agent will have difficulties in more complex environments
  - The “prediction” section of Q-learning
    - Also based on “trial – error” by applying (state, action) from previous observations
- To solve this, implement ***approximations***:
  - Neurons

- More resources are needed depending on approach:



Neuron

- A neuron consists of:
  - An input, **u**
  - An output, **y**
  - An activation signal
  - A threshold, **θ**
  - A weight**, w**
  - *bias, **b**, can be part of the neuron, it can also be considered a hyperparameter



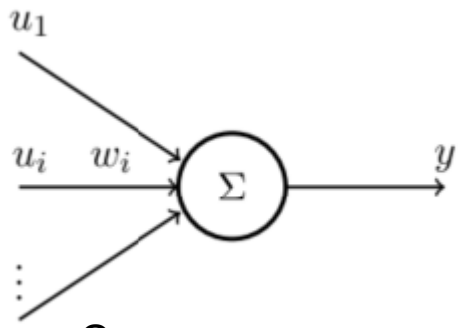
- Mathematically:

$$y = f(x), x = \sum_{i=1}^{n} w_i u_i$$

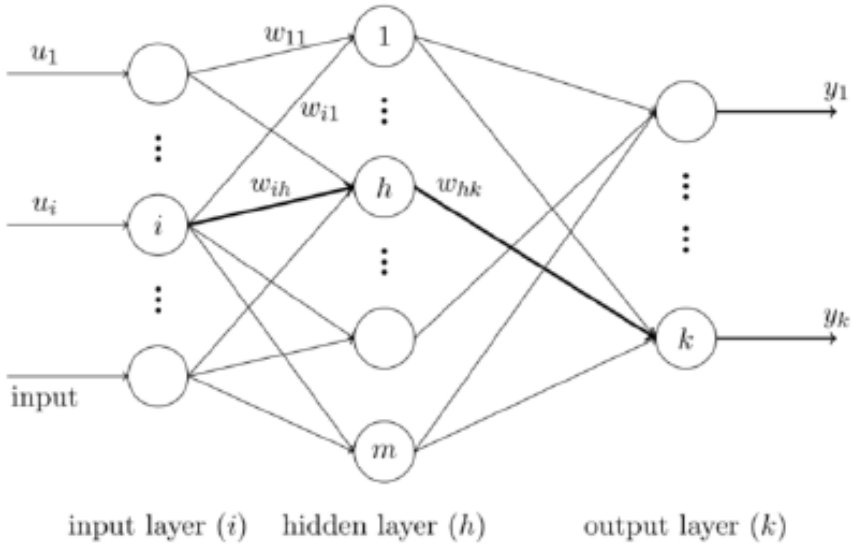
Where: x is a weighted sum, and f(x) an activation function depending on the parameter θ

Neuron

- When several neurons are configured in layers, and all neurons are connected to each other, it's known as a:
  - Fully connected neural network
    - In the simplest form, this is called: **Artificial Neural Network, ANN**
    - A model using an ANN with multiple (usually more than 2 or 3) NN layers is referred to as a model using *Deep Learning*.



One neuron



Neural network

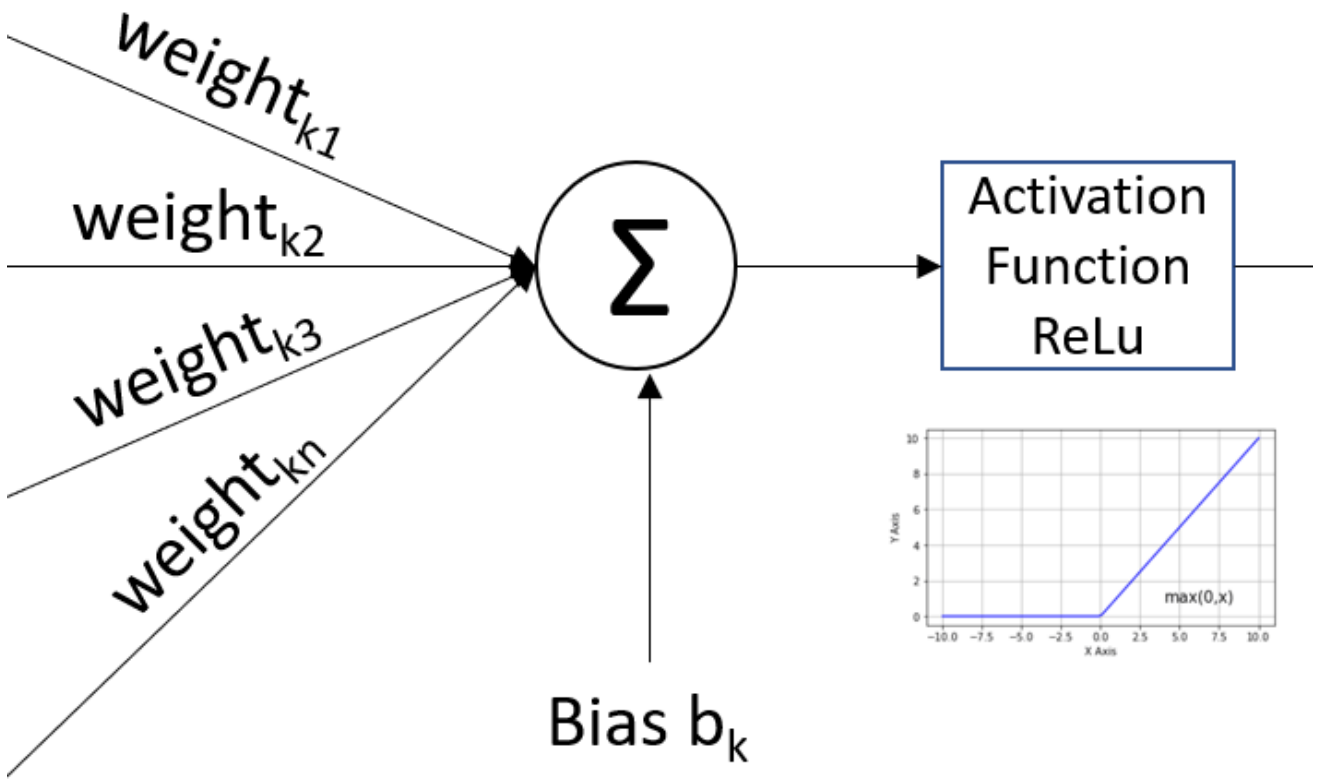
Layers

- In an ANN, in its basic structure: three layers
  - Input Layer:
    - Data coming from “the environment”
      - In DRL these will be observations from the agent
      - In Deep Learning:
        - Labeled data, i.e.
    - Depending on application:
      - Convolutional (& pooling layer): format the input data
        - Convolutional Neural Network, CNN
      - Recurrent: memory-based implementation
        - Recurrent Neural Network, RNN
  - Hidden Layer(s)
    - Where all approximations occur
  - Output Layer
    - Results of the approximations

Approximation

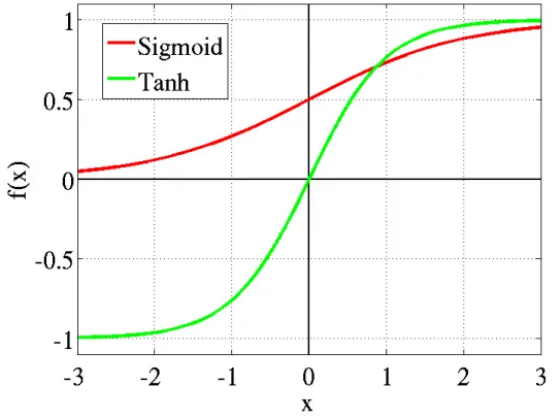
- In one neuron, calculations are occurring:
- Neurons in the hidden layer:
  - Input: weighted sum of neurons in previous layers
  - Weighted sum: input * weight + bias (bias can be optional, however, its is most likely to be included in the calculation)
  - Activation: check if the final weighted sum is within threshold parameters
    - If final weighted sum is within parameters:
      - Neuron remains active
    - If not:
      - Send weighted sum to next layer
      - Deactivate neuron
        - Neuron is no longer used on the next episode (epoch)
- Output layer:
  - Results of approximations:
    - transformed into formatted data

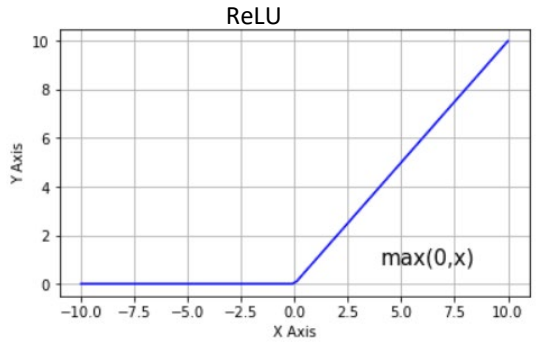
Hidden Layer



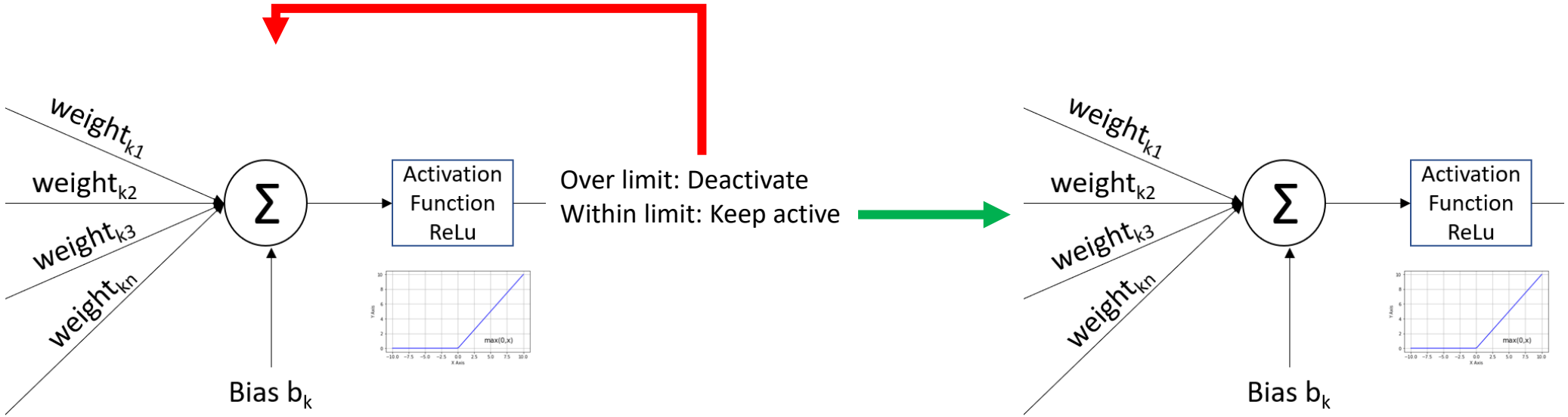
Activation

- Activation is used to decide whether the neuron in question should be kept active or not:
  - Methods:
    - Softmax: based on probabilities
      - Calculates a probability
    - Sigmoid: when the output needs to
      - based on probabilities
    - Tanh: similar to sigmoid, except that
      - Tanh will accept negatives
    - Rectified Linear Unit (ReLU):
      - f(x) is 0 when x is 0



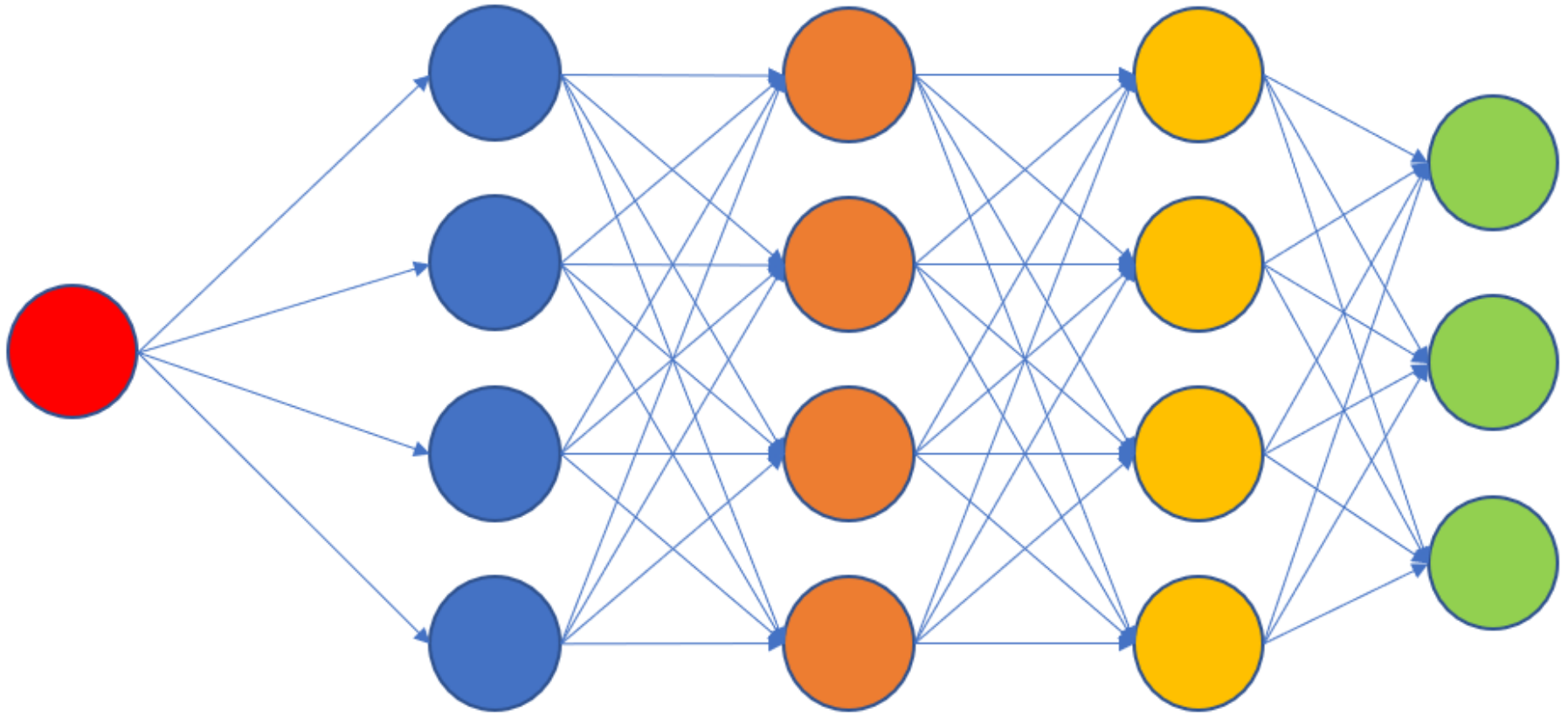


Activation


weight$_{k1}$
weight$_{k2}$
weight$_{k3}$
weight$_{kn}$
Σ
Activation Function ReLu
max(0,x)
Bias b$_k$
Over limit: Deactivate
Within limit: Keep active
weight$_{k1}$
weight$_{k2}$
weight$_{k3}$
weight$_{kn}$
Σ
Activation Function ReLu
max(0,x)
Bias b$_k$

- Before approximations begin:
  - Data needs to be “transformed” into proper neural input
    - Convolutional layer (& pooling layer):
      - Filtering
      - Pooling
      - Flatten & dense

Representation of a fully connected ANN



In CNN

THE UNIVERSITY OF
SOUTHERN MISSISSIPPI®