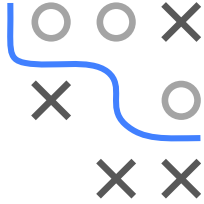
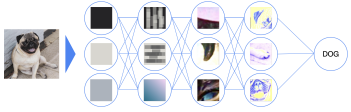
# Introduction to Machine Learning



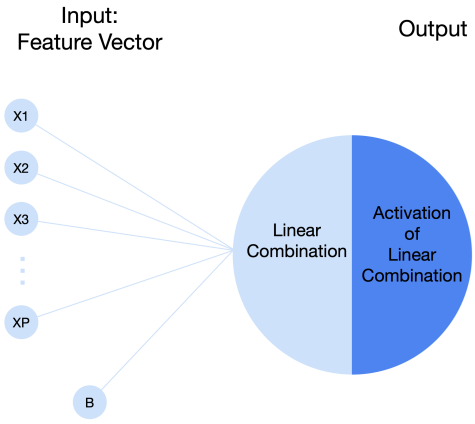
## Neural Networks
## In a Nutshell




**Learning goals**

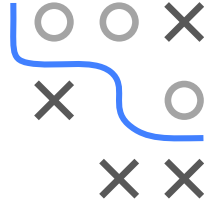
- Know basic computational unit
- Know basic architecture
- Understand Learning in NNs

# BASIC COMPUTATIONAL UNIT: PERCEPTRON



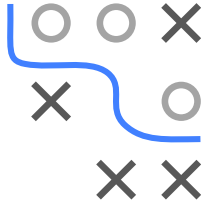

Output differs depending on activation function:

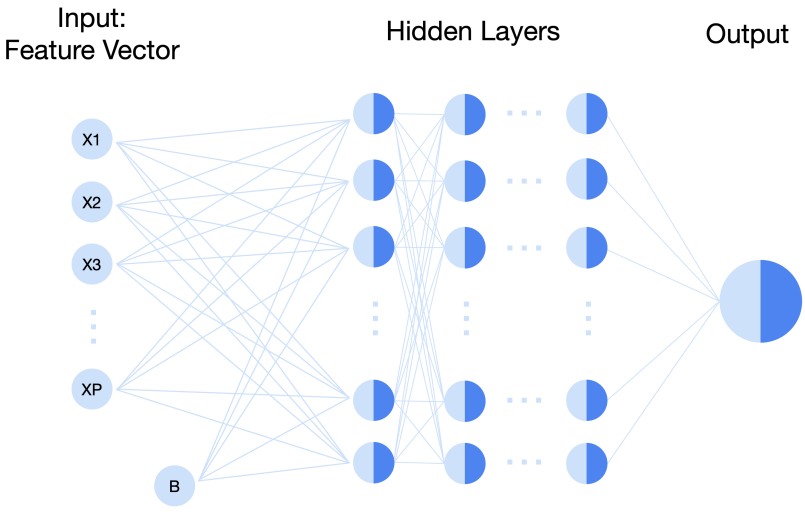
- **Identity function**: Perceptron represents linear regression
- **Logistic function**: Perceptron represents logistic regression
- Other activation functions possible



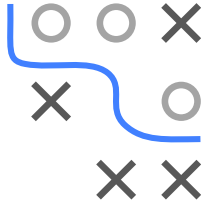
# BASIC ARCHITECTURE OF NN

A neural network is built by combination of multiple perceptrons:





# BASIC ARCHITECTURE OF NN



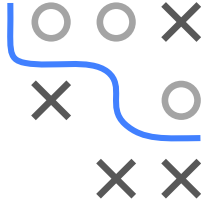
Hidden Layers:

- Output of hidden units serves as input for units in next layer
- Too many hidden layers or too many units per layer can lead to overfitting

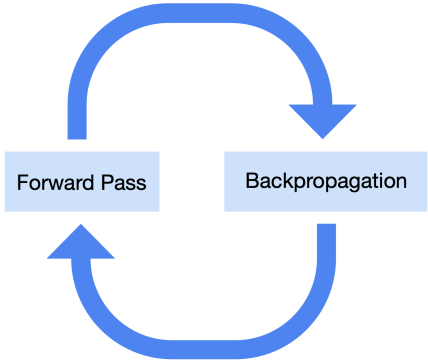
Output Layer:

- Number of output units depend on task
- Different activation functions for output layer and hidden layers possible

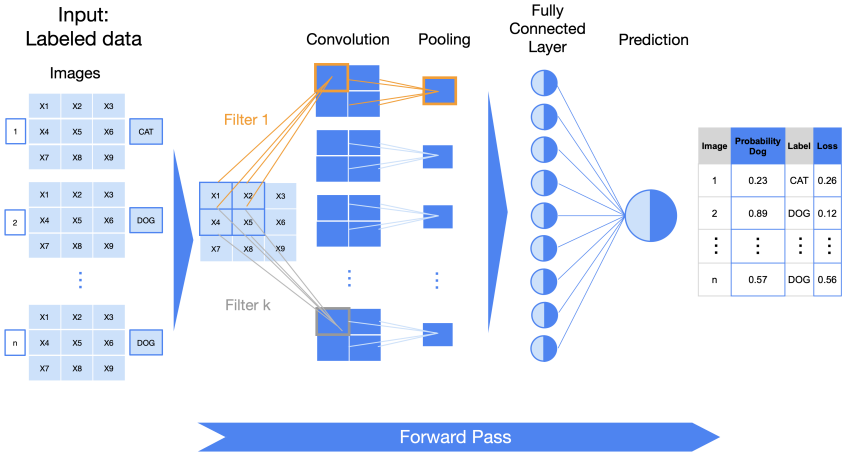
# LEARNING - IMAGE CLASSIFICATION TASK



For each Training Iteration:



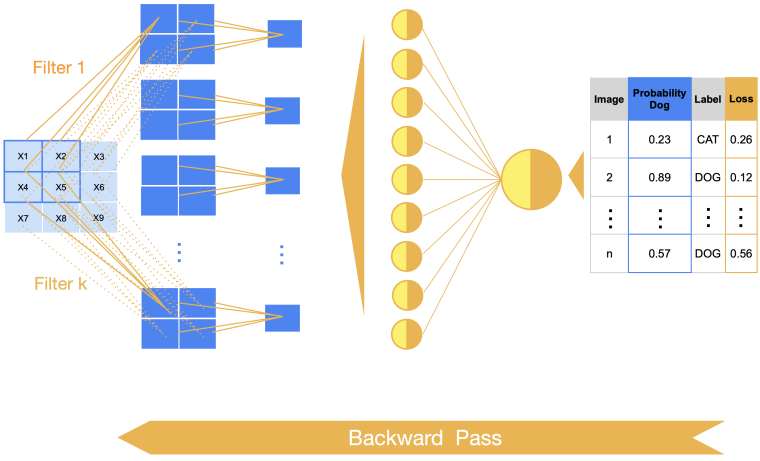
# LEARNING - IMAGE CLASSIFICATION TASK

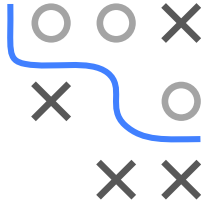



| Image | Probability Dog | Label | Loss |
|---|---|---|---|
| 1 | 0.23 | CAT | 0.26 |
| 2 | 0.89 | DOG | 0.12 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| n | 0.57 | DOG | 0.56 |

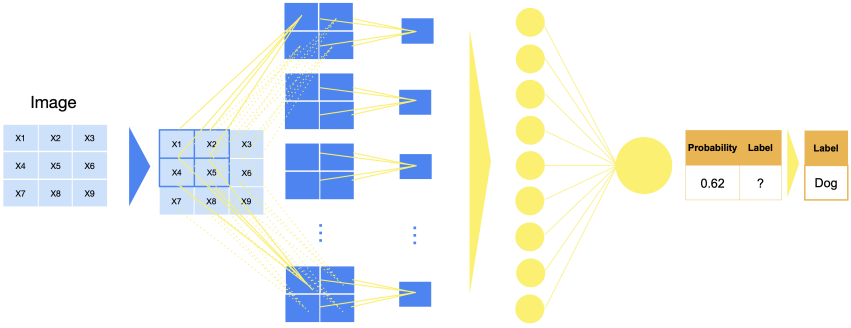
# LEARNING - IMAGE CLASSIFICATION TASK

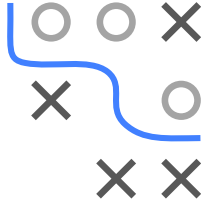
Compute update of each weight by backpropagation



# PREDICTION - IMAGE CLASSIFICATION TASK



# EFFECT OF HIDDEN LAYERS

- Learn more and more abstract representations
- Each layer adds degree of non-linearity