Artificial Intelligence and Machine Learning (AIML)

Attendance Code:
99014918

2023–24

Attendance Code:
99014918

- **So far**: Symbolic AI
  - Problems are defined by explicit rules and logic; deals with certainties and exact computational rules
  - Goal: find an **optimal solution** that satisfies all given constraints and rules
  - Techniques involve **search algorithms** & **logical reasoning**

- **From this lecture**: Machine Learning (ML)
  - Solution is not explicitly programmed; problems are defined by data & patterns within it
  - Goal: “learn” from data to make **predictions** or decisions (**classifications**)
  - Techniques involve adjusting **model parameters** that minimize prediction error

Attendance Code:
99014918

- **This lecture**: Introduction to ML
  - ML framework: problem definition
  - Sequential Gradient Descent (SGD)
    - Function, derivatives (**Gill**, Sections 5.1–5.4)
    - Vector notation (**Gill**, Sections 3.2–3.5)
- Example: linear regression

AIML

Attendance Code:
99014918

Machine learning (ML): overview

- Given some **training data** (and a proposed model), ML training finds optimal **model parameters** such that the **prediction error** is minimized
- **Training**: minimize an **error/loss function** $F(w)$ which depends upon the ML **model** and continuous w and the training data,

$$w^\star = \arg\min_{w' \in \mathcal{W}} F\left(w'\right)$$

- Typically, $\mathcal{W}$ is just D-dimensional Euclidean space $\mathbb{R}^D$, but special problems (such as clustering) have other optimization spaces
- Then use the trained parameters in the model, $w^\star$, to make predictions about new unseen **test data**
- ML algorithms usually categorized according to availability of labelled data: **supervised**, **unsupervised**, **self-supervised**, **transfer learning**

Image Classification

input

classifier

output

“cat”

“5”

Semantic Segmentation

“a label for each pixel”

Neural Networks

- Machine learning technique
- Often used for classification, semantic segmentation, and related tasks
- First ideas discussed in the 1950/60ies
- Theory work on NNs in the 1990ies
- Increase in attention from 2000 on
- Deep learning took off around 2010
- CNNs for image tasks from 2012 on

Neural Network

What is a **neuron**?

fundamental unit
(of the brain)

What is a **network**?

connected elements

neural networks are connected elementary (computing) units

Biological Neurons

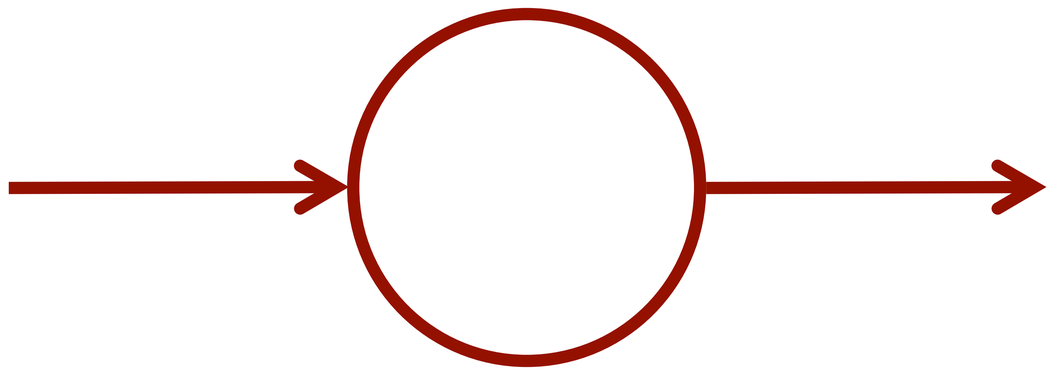
Biological neurons are the **fundamental units** of the brain that

- Receive sensory input from the external world or from other neurons
- Transform and relay signals
- Send signals to other neurons and also motor commands to the muscles

Artificial Neurons

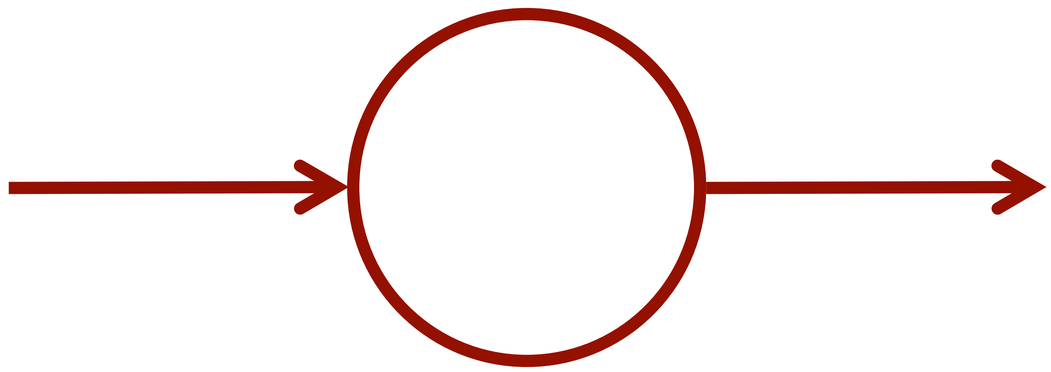
Artificial neurons are the fundamental units of artificial neural networks that

- Receive **inputs**
- **Transform** information
- Create an **output**



Neurons

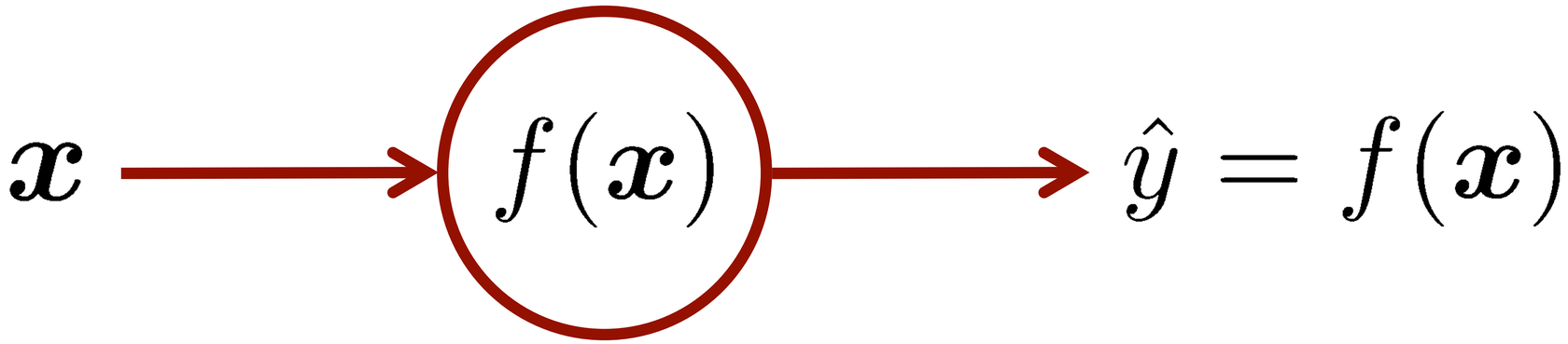
- Receive **inputs / activations** from sensors or other neurons
- **Combine / transform** information
- Create an **output / activation**



Neurons as Functions

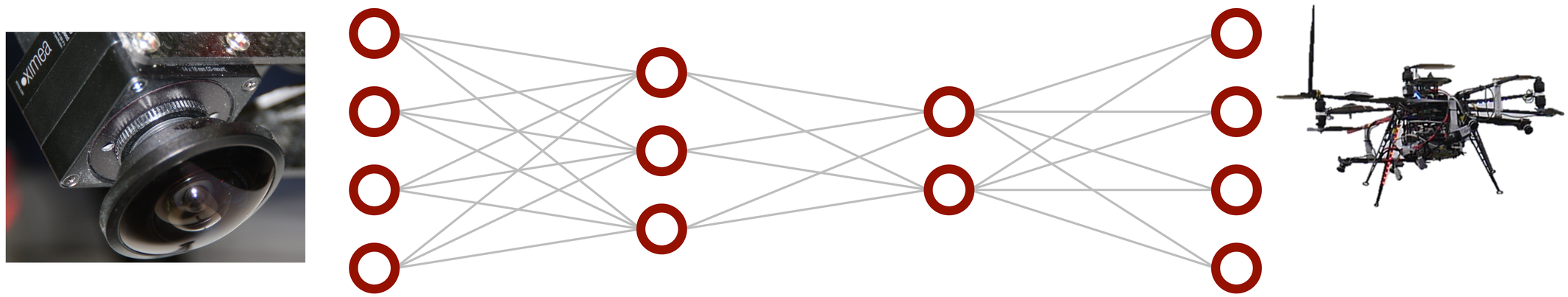
We can see a neuron as a function

- Input given by $\boldsymbol{x} \in \mathbb{R}^N$
- Transformation of the input data can be described by a function f
- Output $f(\boldsymbol{x}) = \hat{y} \in \mathbb{R}$



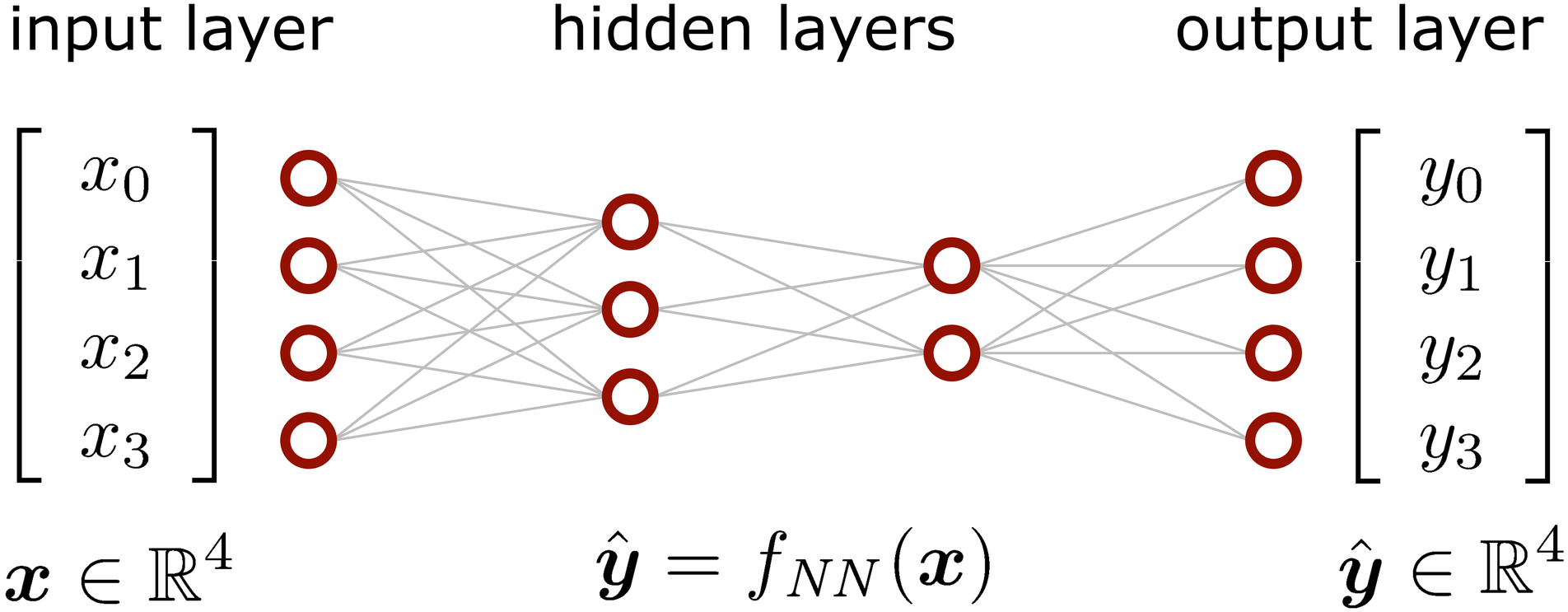
Neural Network

- NN is a network/graph of neurons
- Nodes are neurons
- Edges represent input-output connections of the data flow



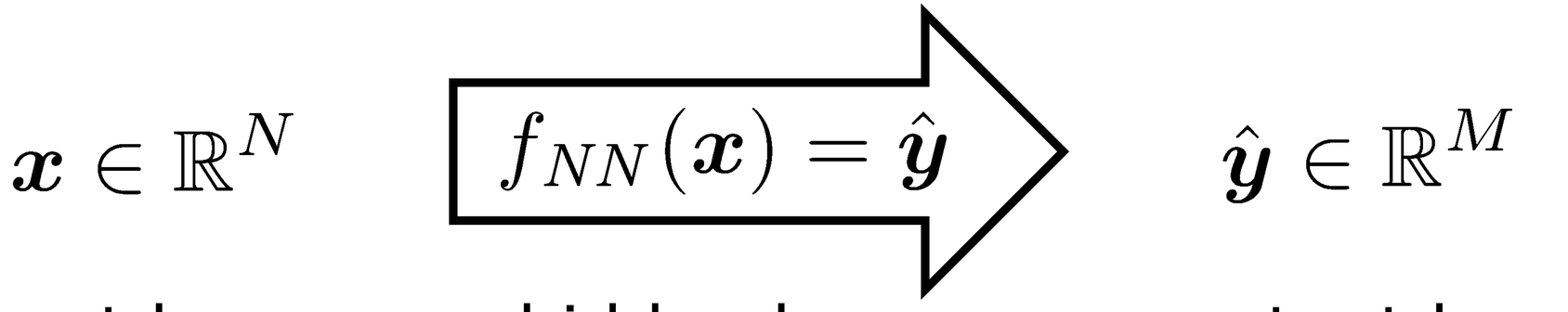
Neural Network as a Function

- The whole network is again a function
- Multi-layer perceptron or MLP is often seen as the “vanilla” neural network

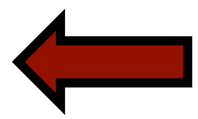


Neural Networks are Functions

- Neural networks are functions
- Consist of connected artificial neurons
- Input layer takes (sensor) data
- Output layer provides the function result (information or command)
- Hidden layers do some computations



Different Types of NNs

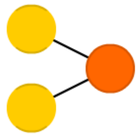
- Perceptron
- MLP – Multilayer perceptron 
- Autoencoder
- CNN – Convolutional NN
- RNN – Recurrent NN
- LSTM – Long/short term memory NN
- GANs – Generative adversarial network
- Graph NN
- Transformer
- …

A mostly complete chart of

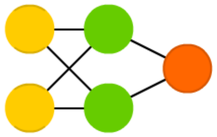
Neural Networks

Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool

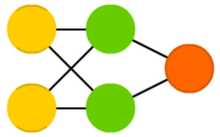
Perceptron (P)



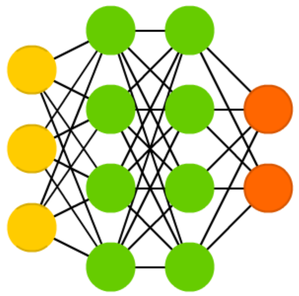
Feed Forward (FF)



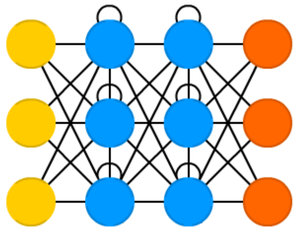
Radial Basis Network (RBF)



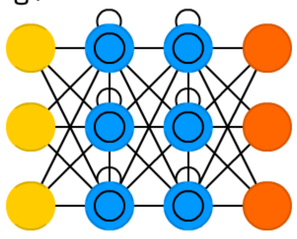
Deep Feed Forward (DFF)



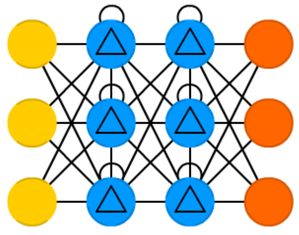
Recurrent Neural Network (RNN)



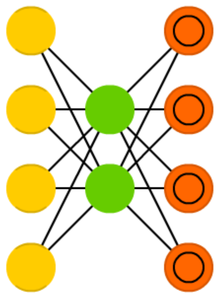
Long / Short Term Memory (LSTM)



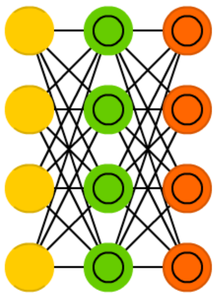
Gated Recurrent Unit (GRU)



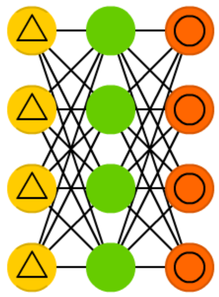
Auto Encoder (AE)



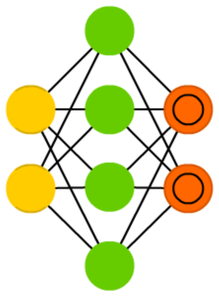
Variational AE (VAE)



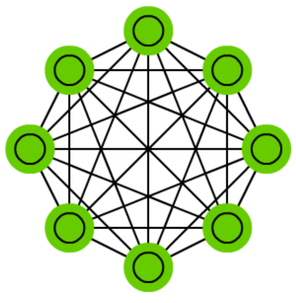
Denoising AE (DAE)



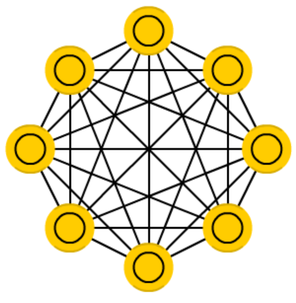
Sparse AE (SAE)



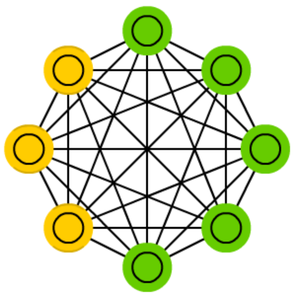
Markov Chain (MC)



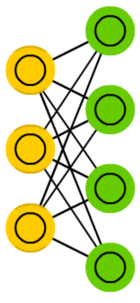
Hopfield Network (HN)



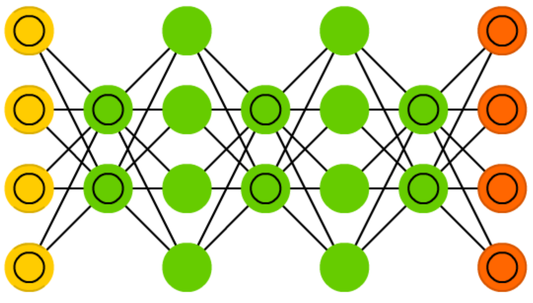
Boltzmann Machine (BM)



Restricted BM (RBM)

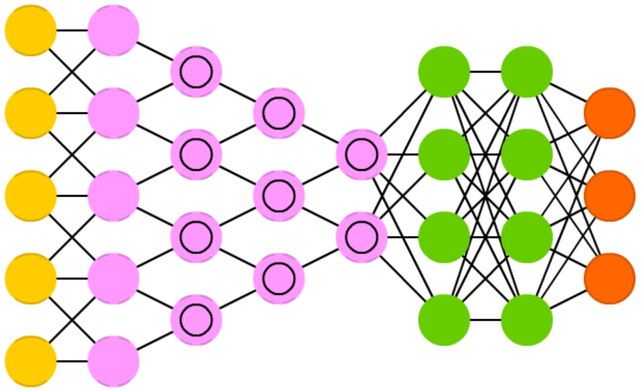


Deep Belief Network (DBN)

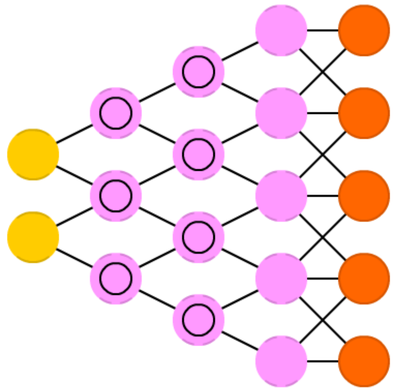


[Image courtesy: van Veen]

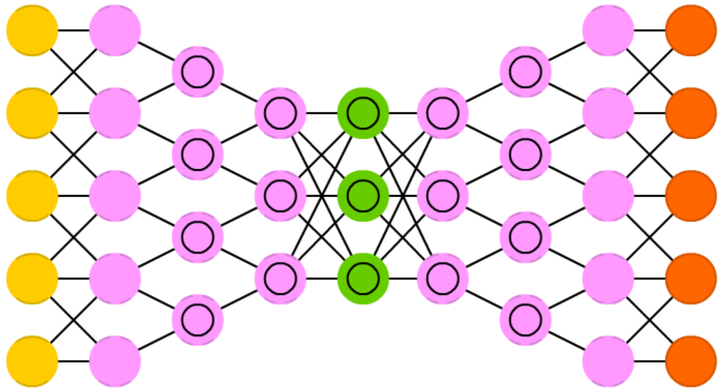
Deep Convolutional Network (DCN)



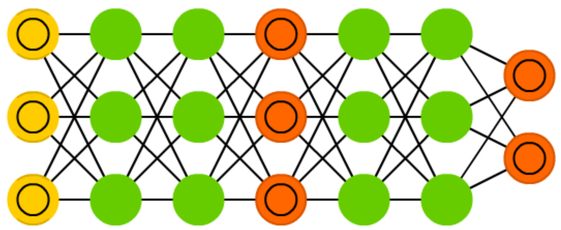
Deconvolutional Network (DN)



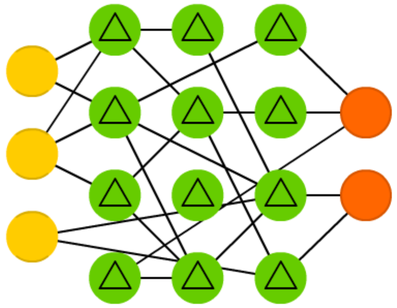
Deep Convolutional Inverse Graphics Network (DCIGN)



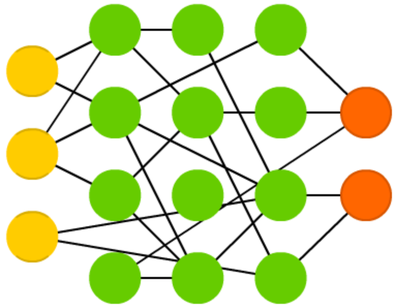
Generative Adversarial Network (GAN)



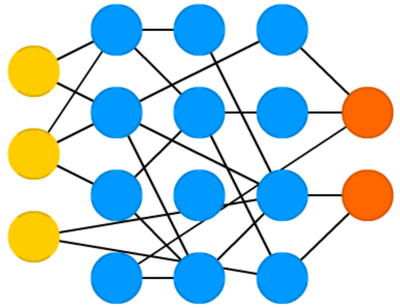
Liquid State Machine (LSM)



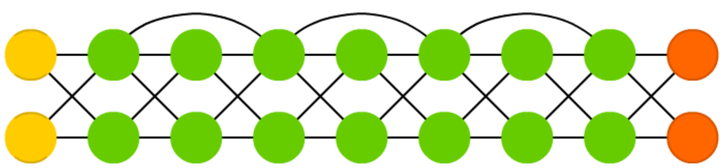
Extreme Learning Machine (ELM)



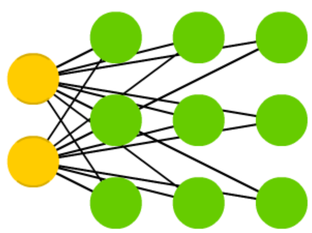
Echo State Network (ESN)



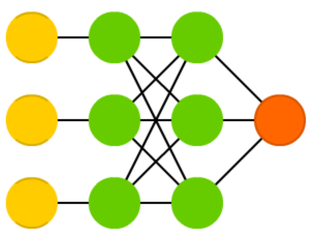
Deep Residual Network (DRN)



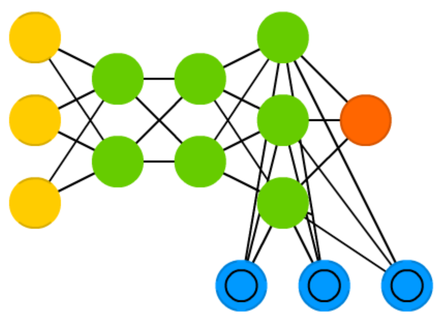
Kohonen Network (KN)



Support Vector Machine (SVM)

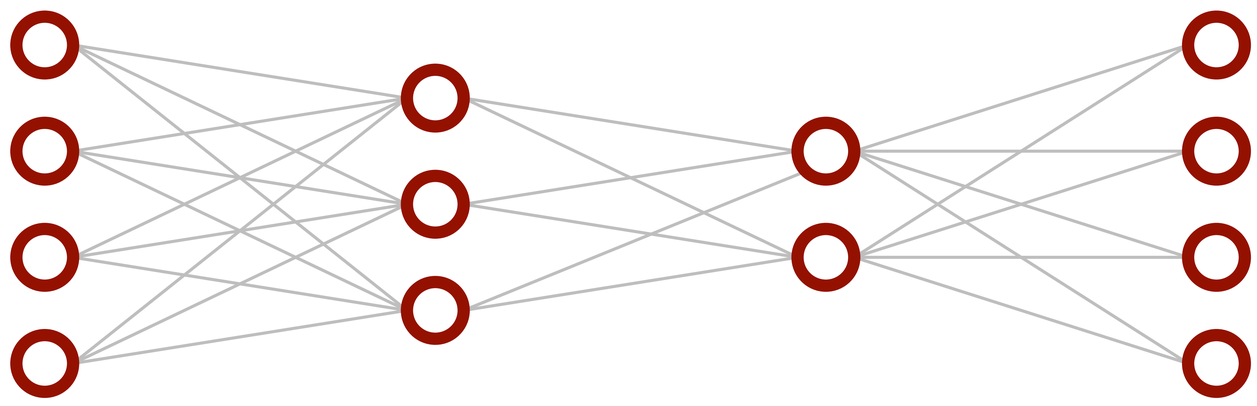


Neural Turing Machine (NTM)



[Image courtesy: van Ve

Multi-layer Perceptron (MLP)



Multi-layer Perceptron Seen as a Function

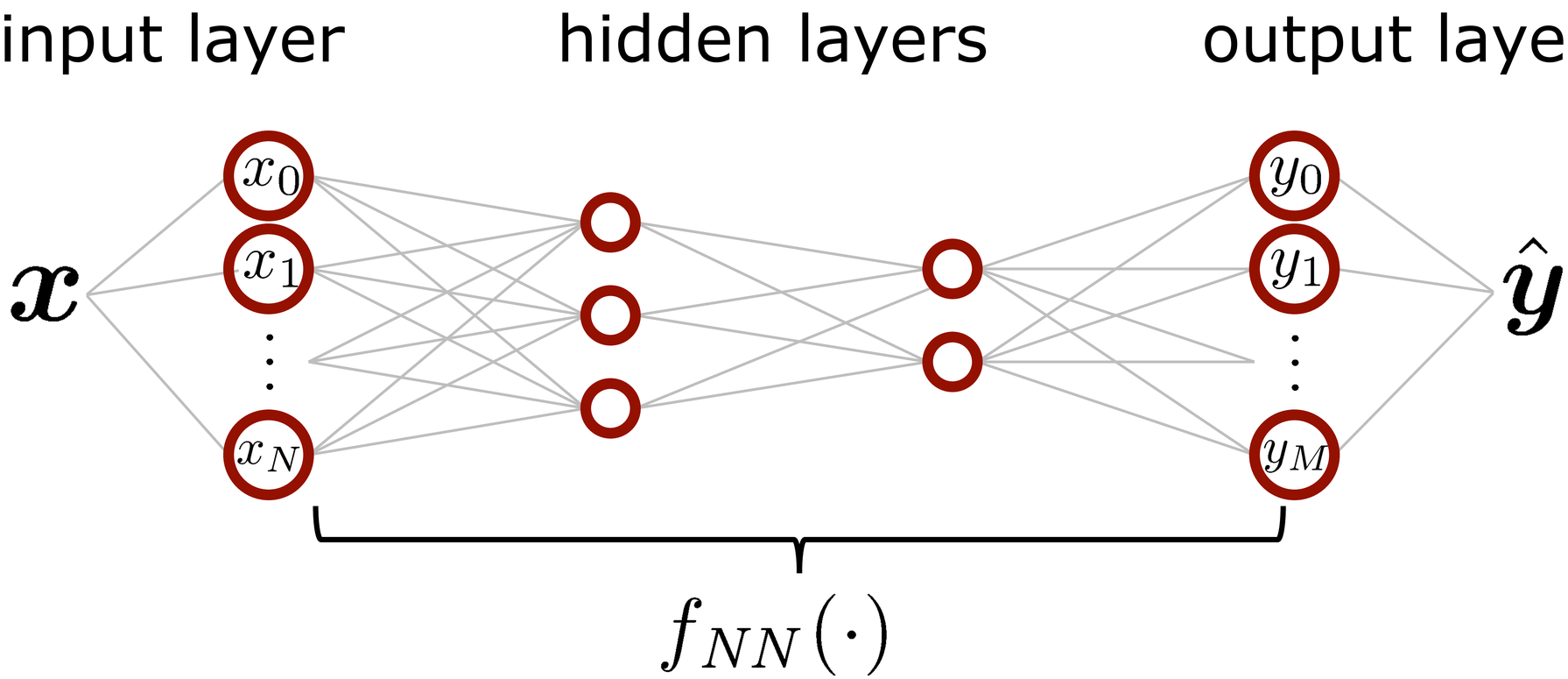
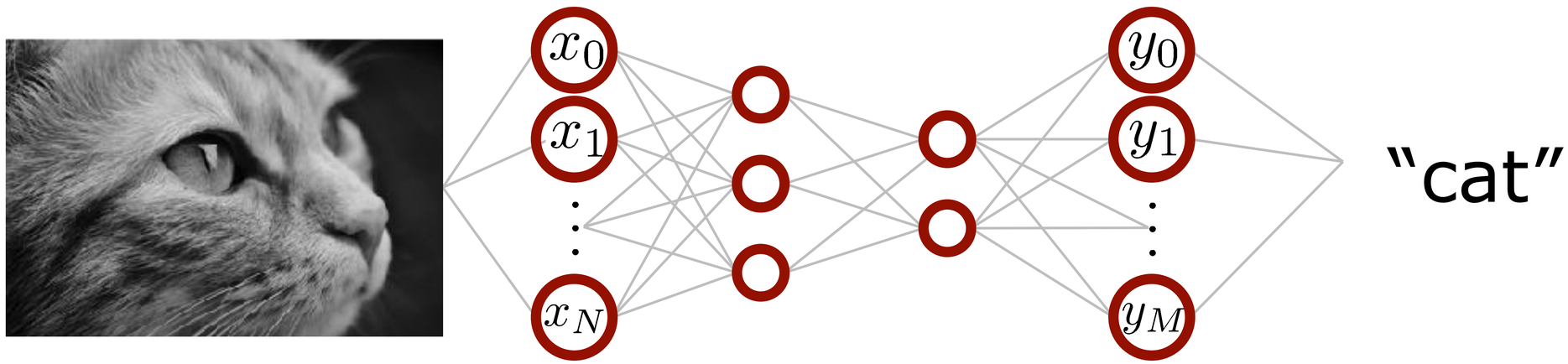


Image Classification Example



$\boldsymbol{x}$ — input image

$f_{NN}(\cdot)$ — function that maps images to labels

$\hat{\boldsymbol{y}}$ — label

What is the Network's Input?

An image consists of individual pixels.

image

What is the Network's Input?

An image consists of individual pixels.

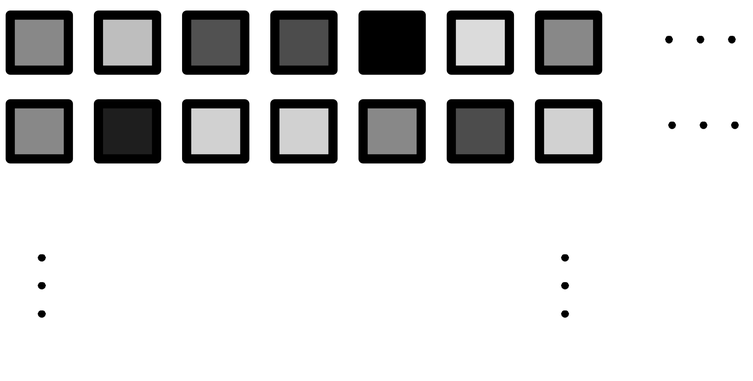
pixel intensities

image

Each pixel stores an intensity value.

What is the Network's Input?

pixel intensities

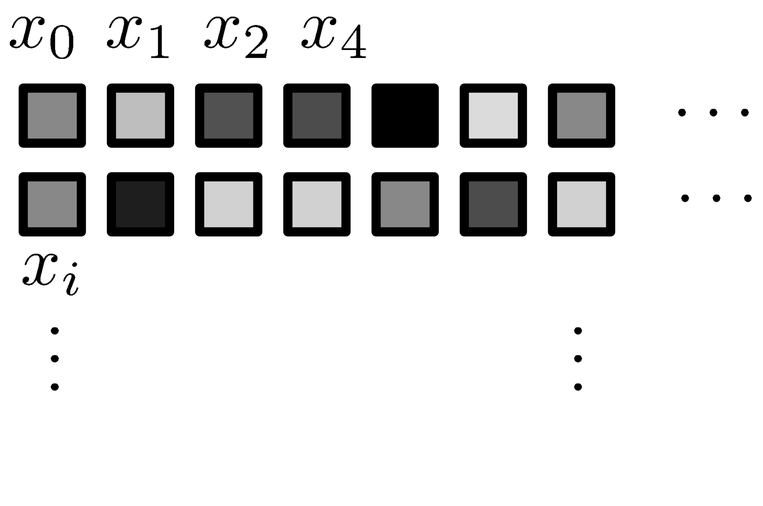


image

An image consists of individual pixels.

Each pixel stores an intensity value.

What is the Network's Input?

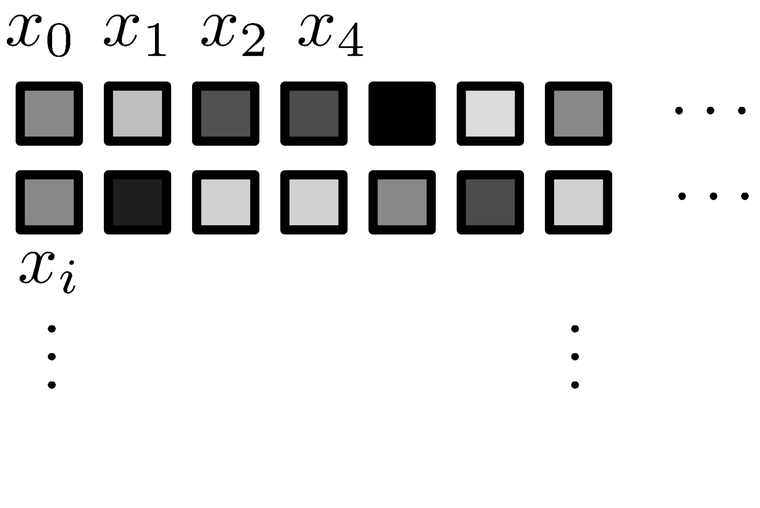


An image consists of individual pixels.

Each pixel stores an intensity value.

We have N+1 such intensity values.

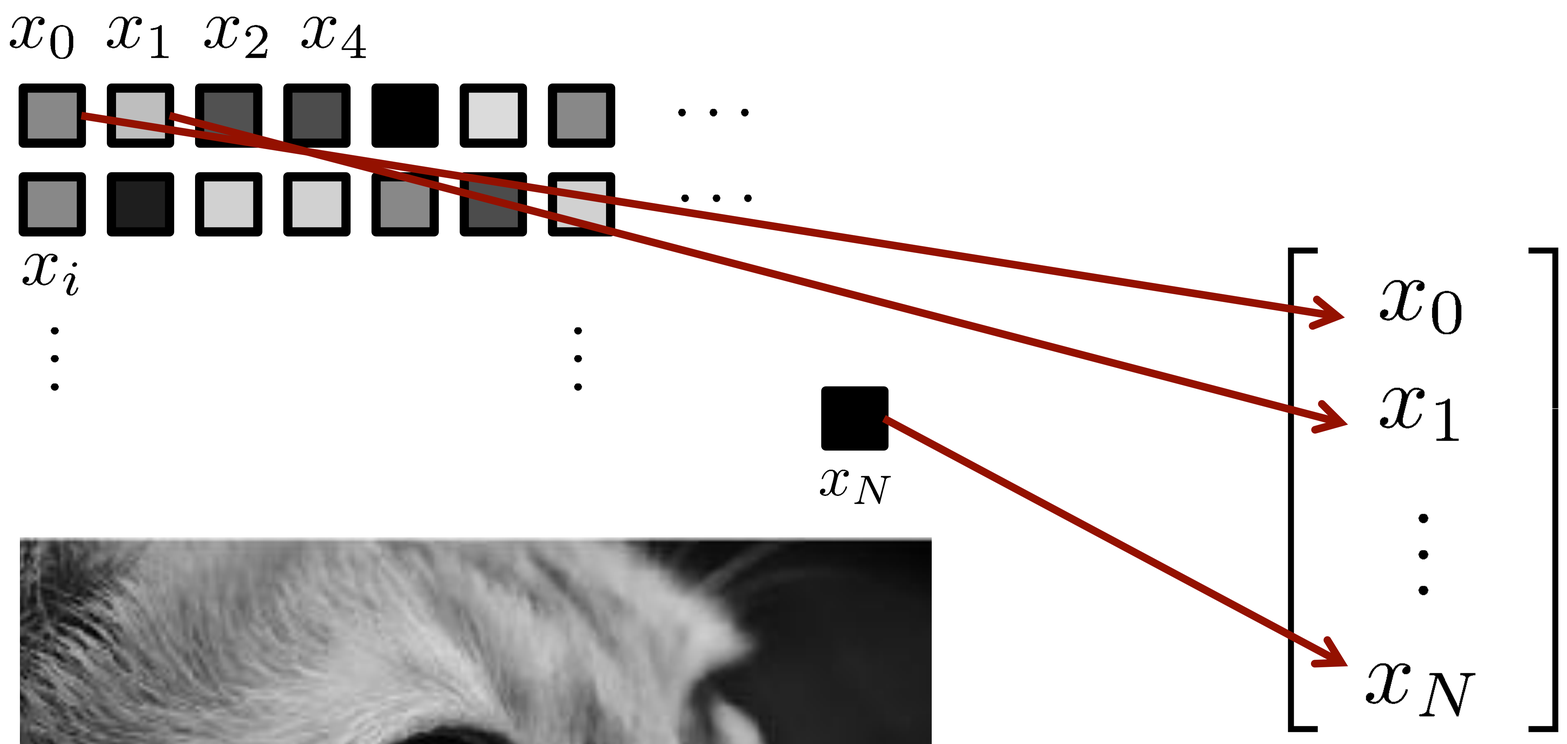
What is the Network's Input?



$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix}$$

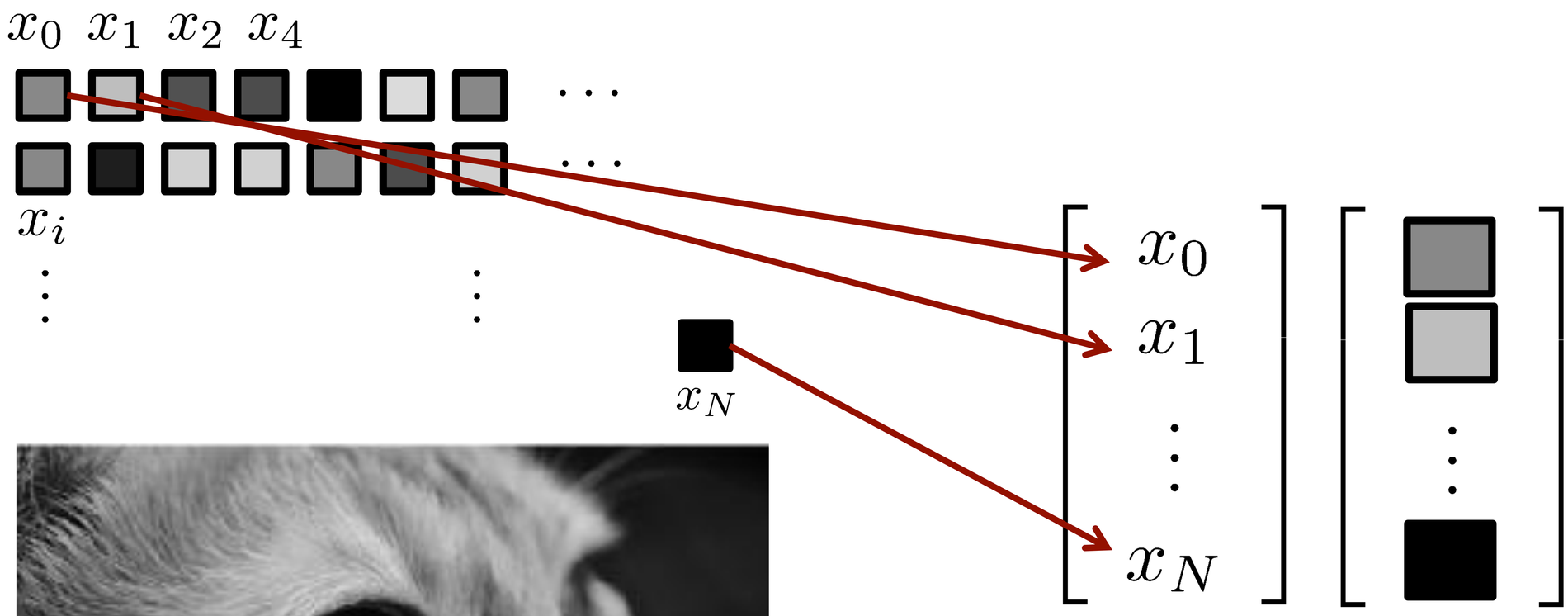
Arrange all the intensity values in a N+1 dim vector.

What is the Network's Input?



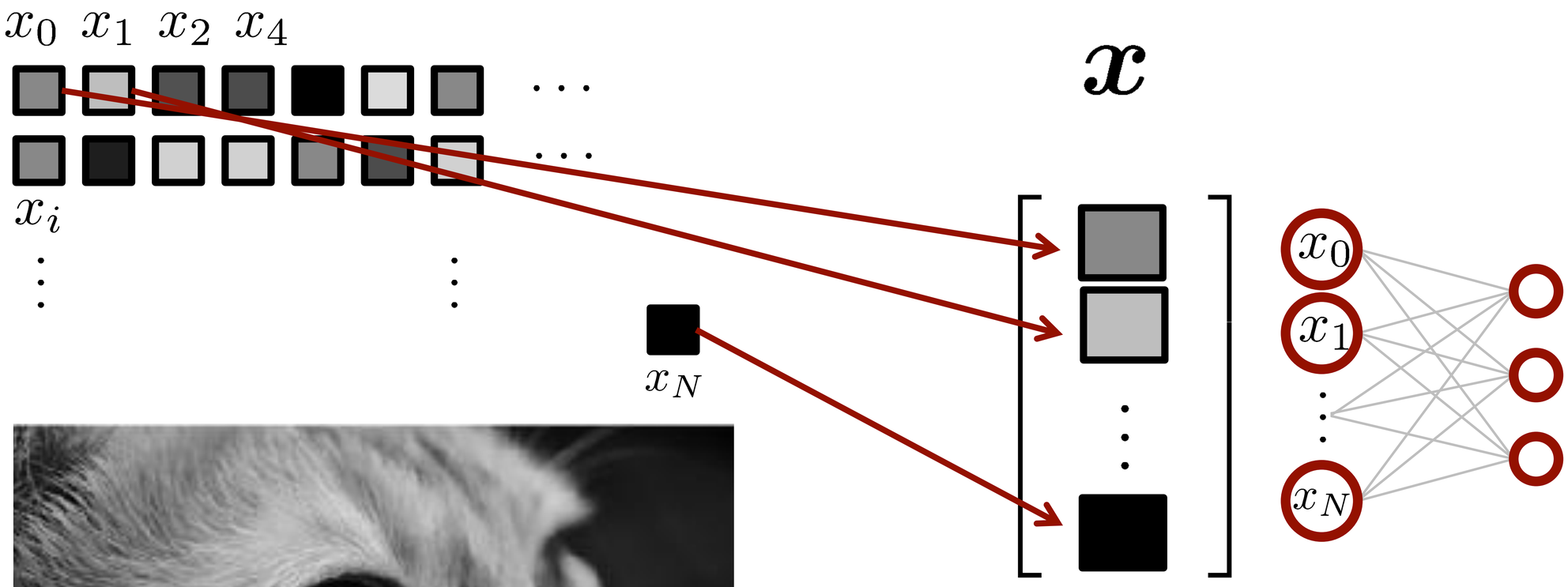
Arrange all the intensity values in a N+1 dim vector.

What is the Network's Input?



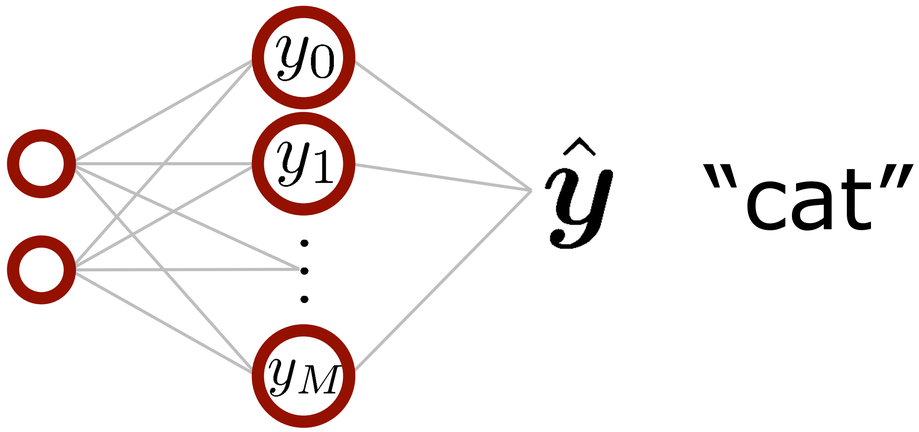
Arrange all the intensity values in a N+1 dim vector.

Input Layer of the Network

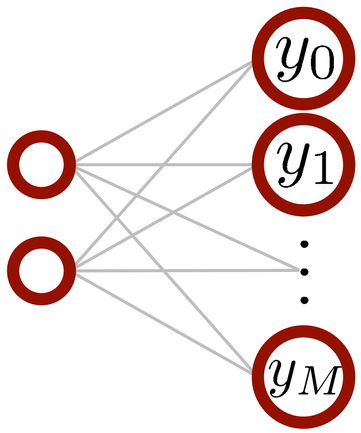


This vector is the input layer of our network!

What is the Network's Output?

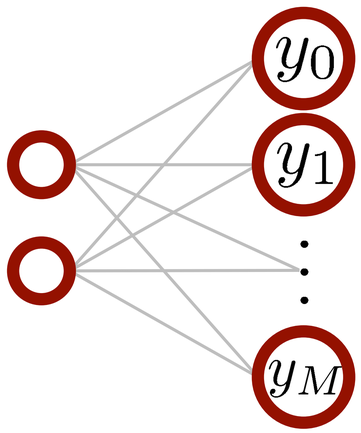


What is the Network's Output?



Is it a…
cat or a
dog or a
human or a
…?

What is the Network's Output?

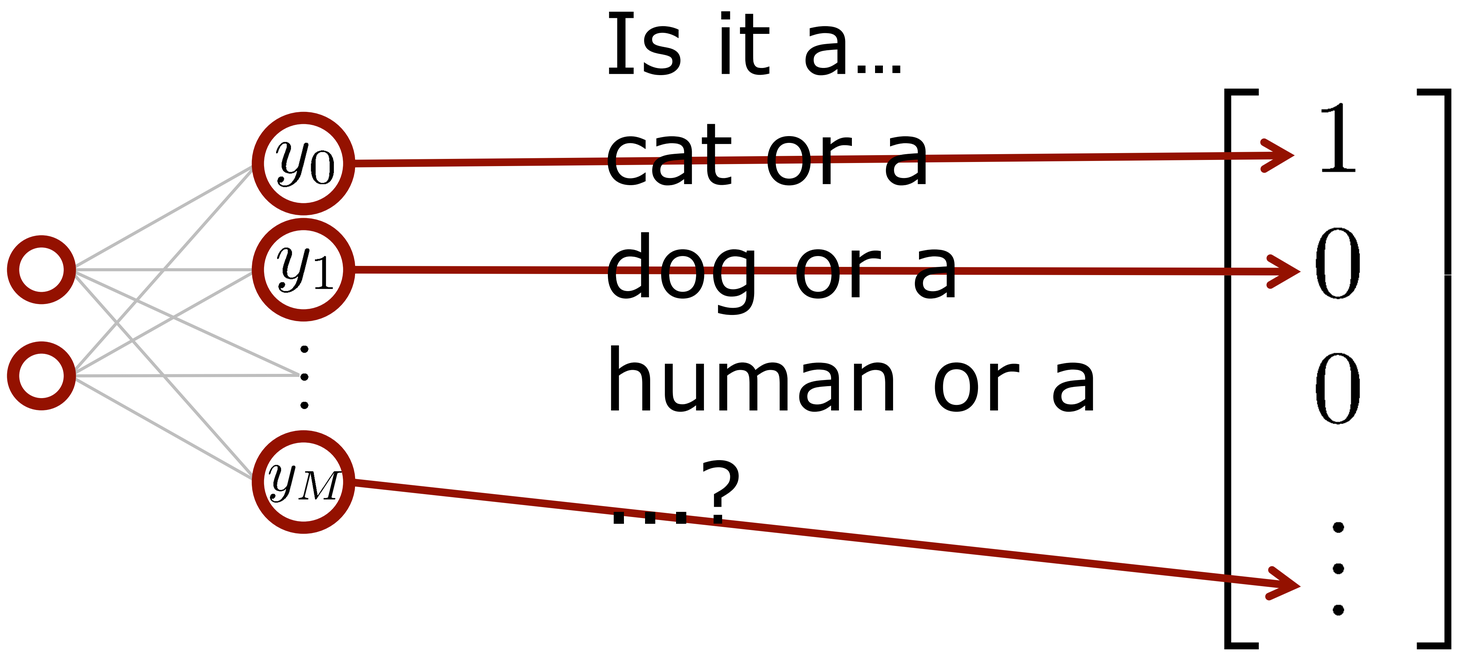


Is it a…
cat or a
dog or a
human or a
…?

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

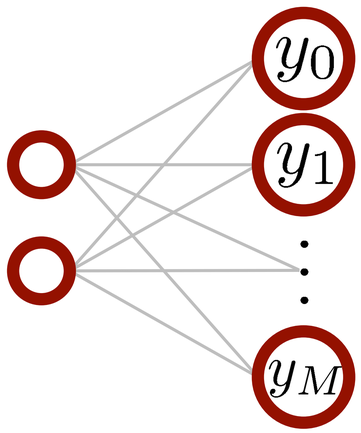
indicator vector

What is the Network's Output?



indicator vector

What is the Network's Output?

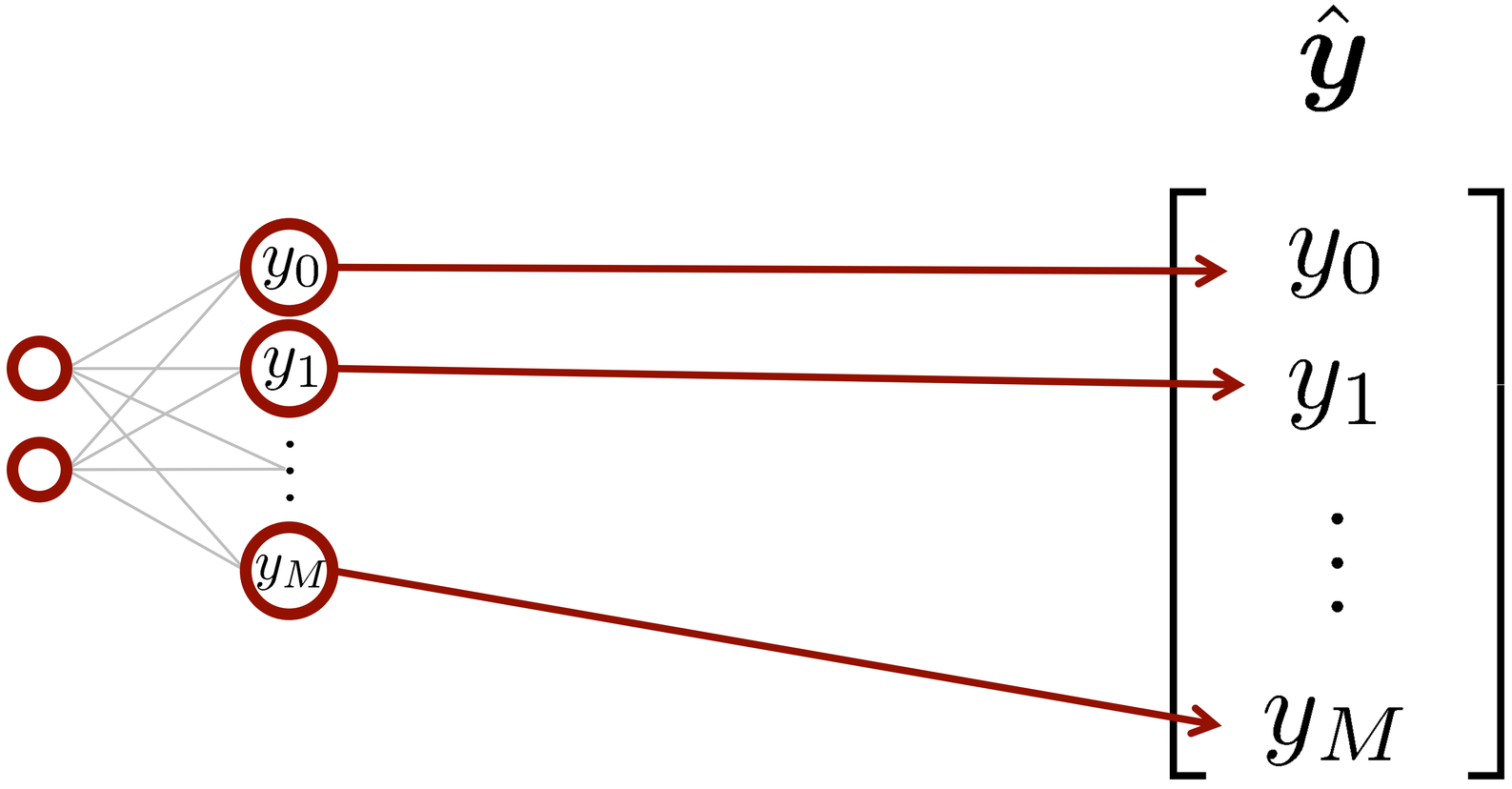


Is it a…
cat or a
dog or a
human or a
…?

$$\begin{bmatrix} 98\% \\ 1\% \\ 0.1\% \\ \vdots \end{bmatrix}$$

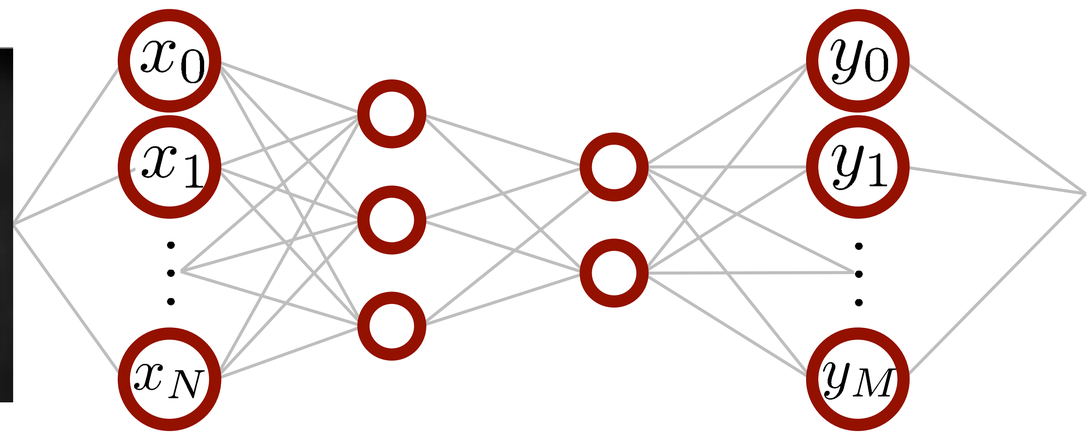
we are never certain…

Output of the Network



the output layer is vector indicating an activation/ likelihood for each label

Image Classification



"cat"

largest value

$\boldsymbol{x}$

$\hat{\boldsymbol{y}}$

pixels intensities are the values of the input layer

output layer is a vector of likelihoods for the possible labels

Next week:

Linear Regression

Gradient Descent

How training works